# Chunked Data Shapley: A Scalable Dataset Quality Assessment for Machine Learning

Andreas Loizou
*Database and Knowledge Systems Lab*
*School of ECE, National Technical University of Athens*
Athens, Greece
antreasloizou@mail.ntua.gr

Dimitrios Tsoumakos
*Database and Knowledge Systems Lab*
*School of ECE, National Technical University of Athens*
Athens, Greece
dtsouma@mail.ntua.gr

## Abstract

As the volume and diversity of available datasets continue to increase, assessing data quality has become crucial for reliable and efficient Machine Learning analytics. A modern, game-theoretic approach for evaluating data quality is the notion of *Data Shapley* which quantifies the value of individual data points within a dataset. State-of-the-art methods to scale the NP-hard Shapley computation also face severe challenges when applied to large-scale datasets, limiting their practical use. In this work, we present a Data Shapley approach to identify a dataset's high-quality data tuples, *Chunked Data Shapley (C-DaSh)*. *C-DaSh* scalably divides the dataset into manageable chunks and estimates the contribution of each chunk using optimized subset selection and single-iteration stochastic gradient descent. This approach drastically reduces computation time while preserving high quality results. We empirically benchmark our method on diverse real-world classification and regression tasks, demonstrating that *C-DaSh* outperforms existing Shapley approximations in both computational efficiency (achieving speedups between 80× – 2300×) and accuracy in detecting low-quality data regions. Our method enables practical measurement of dataset quality on large tabular datasets, supporting both classification and regression pipelines.

## CCS Concepts

• **Computing methodologies → Machine learning**; **Machine learning approaches**; *Supervised learning*; *Unsupervised learning*; *Ensemble methods*; • **Information systems → Specialized information retrieval**; **Data cleaning**; **Data analytics**.

## Keywords

Data Quality, Data Shapley, Machine Learning, Data Selection, Data-Centric AI

## 1 Introduction

Data grows exponentially, in volume as well as variety and velocity [12]. As data increases at this pace in data centres, organisations struggle with the performance of their decision-making algorithms, one promising reason is the data quality. Selecting the "right" data for the used of an data-driven algorithm (such as machine learning) can improve the algorithm's performance efficiency [18].

The plethora of available data within the organisation's data centres can cause the identification of high-quality data to be a real struggle. Using the human factor to identify high-quality data is time-consuming, as well as drop the performance of the data-driven algorithm. Automating data selection using AI techniques has been gamy tracing the data context is named as Data-Centric Artificial Intelligence (AI) [17, 18, 33]. Data-Centric AI techniques besides automate the data selection, improve the selection of the high quality data and the performance of a data-driven ML algorithm. Also, because data quality is described on different dimensions using Data-Centric AI can focus on the specific dimension's that matters [14].

What constitutes **Data Quality** can vary depending on different contexts and applications. Previous studies have focused on traditional data quality issues, such as low accuracy due to poor data quality, dataset completeness with respect to real-world scenarios, and consistency among the data tuples within a dataset [4, 32]. More recent studies define low data quality with respect to the size, order, and distribution of the data [11]. Others emphasize on more nuanced dimensions, such as bias and fairness, which can result to discriminatory results in machine learning models trained on suboptimal datasets [3, 24]. A recent study [21] categorises data quality in six different dimensions: representation, completeness, feature accuracy, target accuracy, uniqueness and target class balance. Data Quality can be depended on the data tuple features, such as corruption or missing, which is a problem that Data Shapley works try to resolve [27, 28]. As a data-centric AI approach, **Data Shapley** quantifies the importance of individual data tuples in a data-driven learning algorithm, identifying its quality and value by their contribution to overall model performance [31, 33]. However, computing Data Shapley is an NP-hard problem, and various approximations have been designed to improve its computational efficiency [10, 30]. Data Shapley approximations remain computational expensive, often require hours to find each data tuple quality value for a whole dataset with thousands of lines working in parallel mode [10]. In a Data Shapley approximation study [20], they mentioned that a critical component in making these approximations practical is the selection of the most informative subset combinations, which can significantly influence both accuracy and efficiency.

For instance having a repository like MIMIC-III, which contains daily patient reports from Beth Israel Deaconess Medical Center [15], it is crucial to identify the missing or corrupted data tuples, as they can negatively impact ML algorithms. Previous studies [10, 20], showed that approximate methods can be used to detect low-quality data tuples. However, these methods still suffer from high computational complexity, often requiring hours to days. This makes their application to large-scale, real-world scenarios impractical.

In order to enable practical assessment of dataset quality, we propose a Data Shapley approximation method called *Chunked Data Shapley (C-DaSh)*. This method aims to close the gap between the computational efficiency and identify the quality of each data tuple accurately. *C-DaSh*, instead of evaluating the contribution of each individual data tuple, groups data tuples into chunks and computes the contribution of each chunk to the performance of a data-driven ML algorithm. By incorporating a subset selection prior to the Data Shapley computation, we identify the most optimal subsets which enhances the ability of our method to identify low-quality data chunks. Unlike most existing Data Shapley approximation methods that are limited to either classification or regression/time-series tasks, our approximation method is designed to work effectively for both tasks. In comparison to prior baseline work by [10], which introduces two Data Shapley approximations methods, the Gradient Shapley(G-Shapley) and Truncated Monte Carlo Shapley (TMC-Shapley), our approach demonstrates improved accuracy in detecting low-quality data and achieves considerably improved performance.

The main contributions of our work can be summarised as follows:

- We introduce a novel Data Shapley approximation algorithm, *C-DaSh*, that partitions datasets into chunks and computes quality scores per chunk instead of individual data tuples. Using our subset selection process to identify the most informative data chunks, our method effectively reduces computational complexity from exponential to manageable levels with no degradation in accuracy.
- We extend the applicability of our method to support both classification and regression/time-series tasks, demonstrating versatility across different domains.
- We provide an experimental evaluation across multiple real-world datasets with varied data quality issues (noise, missing values, label corruption), comparing our method against two established Data Shapley approximation techniques. Results show that *C-DaSh* can identify data quality more accurately, achieving impressive speedups ranging from 80× to 2300×. We also provides insights into optimal chunk sizes and subset configurations, balancing accuracy and efficiency for real-world deployments.

Our work is structured as follows: Section 2 presents the Data Shapley problem formulation, along with two baseline approximations. Section 3 outlines the motivation behind our proposed approach, based on the limitations of existing methods and emerging requirements. It also introduces our proposed method, *C-DaSh*, and the adopted subset selection strategy. In section 4, we provide a comprehensive experimental evaluation, comparing *C-DaSh* against two baseline methods, and exploring trade-offs between

different configuration settings of *C-DaSh*. Related work in the area of data quality, both with and without the Data Shapley approach, is described in Section 5. Section 6 concludes the paper with key findings and directions for future research.

## 2 Problem Formulation

Consider a large collection of tabular datasets without any information related to it. Each dataset $D$ comprises data tuples $x_j$ represented as $D = (x_1, x_2, x_3, \ldots, x_n)$. Each data tuple $x_j$ has the same number of features with the other data tuples, as well as be on the same domain. For a given Machine Learning (ML) algorithm $\mathcal{A}$ (such as Multilayer Perception, SVM, etc.) we aim to assess the "quality" of the data tuples and distinguish between the low data quality with the high ones.

Data Shapley [27, 28], rooted in game theory, offers a principled way to evaluate the contribution or "quality" of each data tuple as a player. Having a performance metric $M$ such as Mean Square Error, Accuracy, F-1 score, etc., and a learning algorithm $\mathcal{A}$, finds for each data tuple its quality value, find its contribution among all the available subset combinations using all the rest data tuples from the dataset $D$. Data Shapley value $ds_j$ for a data tuple $x_j$ and a performance metric $M$ can be computed as:

$$ds_{x_j}^M = \sum_{Z \subseteq D \setminus \{j\}} \frac{|Z|! \cdot (|D| - |Z| - 1)!}{|D|!} \left( M(Z \cup \{j\}) - M(Z) \right) \quad (1)$$

where $Z$ is a subset with data tuples (set of players) from dataset $D$. When a subset $Z$ consists all the rest data tuples except $x_j$ is named as the "grand coalition". After the Data Shapley algorithm, is applied on $D$, we have a $DS = (ds_1, ds_2, ds_3, \ldots, ds_n)$, where $n$ is the total number of tuples in dataset $D$. Low $ds_j$ values denote low-quality tuples and vice-versa. However, Data Shapley is an NP-hard problem [8], with a complexity of $O(2^n)$, making it impossible to identify the quality of a large-scale dataset.

Truncated Monte Carlo Shapley [10] is a Data Shapley approximation that tries to speedup the calculation of Data Shapley values. Firstly, this method applies a random permutation on the tuples of dataset $D$. For each Monte Carlo iteration, is applied for each data tuple, the Data Shapley equation to find its marginal contribution on the ML algorithm $\mathcal{A}$. For each data tuple $x_j$ the subset $Z$ consists all the permuted data tuples before the $x_j$, such as $Z = (x_1, \ldots, x_{j-1})$. Multiple Monte-Carlo permutations with the same procedure are repeated until the *truncation* is met. Truncation is a strategy used to assess whether the performance of a learning algorithm $\mathcal{A}$ on a subset $Z$, as measured by the performance metric $M$, meets a predefined tolerance threshold. It helps determine when the marginal contributions of all data tuples are sufficiently accounted for. The final Data Shapley value is the average from all Monte-Carlo permutations marginal contribution of the specific tuple. However, this approximation may speedup the runtime efficiency, with complexity $O(m \cdot n)$ (in practice $m \approx 3n$, so $O(n^2)$) where $m$ is the total number of Monte-Carlo permutations until the truncation is met. However, for a large-scale dataset with thousands of tuples is still not efficient.

Gradient Shapley [10] is another Data Shapley approximation. Using the Stochastic Gradient Descent (SGD) [2] on algorithm $\mathcal{A}$ when is applicable can use the knowledge from the subset data

tuples and distinguish the high-quality data tuples from the lower one more accurately. This approximation uses randomly selected batches with random permutations, similarly to the Truncated Monte Carlo Shapley. Using the random subsets update the model hyperparameters. For each data tuple applies the Data Shapley equation using the model hyperparameters information from the SGD which gained it from the previous data tuples. Using that information find for $x_j$, its marginal contribution on the algorithm $\mathcal{A}$ and the performance metric $M$. The subset $Z$ for each data tuple $x_j$ is all the previous data tuples where update the model hyperparameters using SGD, such as $Z = (x_1, \ldots, x_{j-1})$. The complexity time in this approximation is $O(n \cdot T \cdot d)$, where $d$ is the model hyperparameters dimensionality and $T$ is the number of gradient descent steps (how many times random permutations are performed). This approximation may improve the runtime efficiency compared to the exact Data Shapley computation, but for large-scale datasets will not be efficient.

## 3 Methodology

### 3.1 Motivation

Determining a multi-dimensional "quality" score for each data tuple in a large-scale dataset capturing aspects such as noise, completeness, and more is a highly challenging task. It involves addressing various additional complexities and circumstances, including what performance metric $M$ you must couple with learning algorithm $\mathcal{A}$ and how you efficient is the quality value of the identification on the low-quality data tuples [10]. Data Shapley has the ability to integrate a lot of data quality dimensions, but its high complexity makes it difficult for large-scale datasets. Data Shapley approximation methods have been introduced that try to improve the runtime efficiency, without affecting the identification of the low data quality. However, their practicality is limited to datasets with a ten of thousands of tuples [10]. Nowadays, datasets comprise hundreds to millions data tuples, making it difficult to apply these approximations as they would require months to execute to find the "right" high-quality data tuples for an ML algorithm. Our motivation stems from these limitations. We aim to design and implement a novel approximation of the Data Shapley that not only scales to large datasets but also provides an interpretable, high-precision distinction between the low-quality data tuples and the high-quality among a large dataset. Computing per tuple Data Shapley values is computationally expensive, our approach leverages chunk-based strategy assigning a quality value to each chunk. Our approximation method incorporates SGD [2], to more accurately estimate the contribution of each chunk. Additionally, we employ an smart subset selection strategy to enhance the identification of low quality chunks. Together, these techniques significantly reduce computational overhead while preserving the integrity and precision of the data quality assessment.

### 3.2 Chunked Data Shapley

*3.2.1 Data Quality Measurement.* To reduce the computational complexity of calculating the Data Shapley value for an entire dataset $D$, we partition the dataset into $c$ chunks of the same size $l$, where the total amount of chunks $c \ll n$, as well as the chunk size

---

**Algorithm 1** Chunked Data Shapley

---

**Require:** Dataset $D$, Machine Learning algorithm $\mathcal{A}$, performance metric $M$, chunk size $l$, total number of subsets $k$, threshold $th$, Constant variable $C$

1: **Initialize** Split dataset **D** into **c** chunks of size **l**
2: **Initialize** Chunks Data Shapley quality values **DS** as $ds_j = 0$ for $j = 1, \ldots, c$
3: **while** Truncation Criterion is not Met **do**
4:     **Initialize k** Subsets from dataset **D**
5:     **Execute** subset selection on $\mathcal{S}$ using threshold **th**
6:     **Initialize $w_0$** ← Random Parameters
7:     **for** each chunk $ch_j \in D$ for $j = 1, ..., c$ **do**
8:         **Include** Subsets $Z \subseteq \mathcal{S} \setminus \{j\}$
9:         $w_j(Z) = w_{j-1} - \eta_j \sum_{x \in Z} \nabla \mathcal{L}(w_j, x)$
10:        $ds_{ch_j}^M = \sum_{Z \subseteq \mathcal{S} \setminus \{j\}} \frac{C}{|D| \cdot |Z|} (M(w_j(Z \cup \{j\})) - M(w_j(Z)))$
11:    **end for**
12: **end while**
13: **Having** Chunk Data Shapley Quality values $DS = (ds_1, ds_2, ds_3, \ldots, ds_l)$
14: **Return** Dataset quality values $DS$

---

$l \ll n$. For each chunk $ch_j$ from the set $(ch_1, ch_2, ch_3, \ldots, ch_c)$ we assign a corresponding data quality score value $ds_j$. Upon completing the approximation process for all chunks, we obtain a Data Shapley quality score vector $DS = (ds_1, ds_2, ds_3, \ldots, ds_c)$. Organising the dataset into distinct chunks, non-overlapping multiple data tuples, we significantly speed up computation, as the number of required subset evaluations is greatly reduced. Before computing the Data Shapley values, we perform a subset selection step to identify the most informative and representative subsets $\mathcal{S}$ for evaluation. As highlighted in [20], careful subset selection leads to more accurate quality estimates for individual data tuples. A detailed explanation of our subset selection strategy is provided in Subsection 3.2.2.

First we partition the dataset $D$ into chunks with equal size of $l$. Once the subset selection step (described at subsection 3.2.2) is performed, we apply the Data Shapley computation over the selected subsets of chunks by using the knowledge information from the SGD [2] optimisation function. We use the SGD optimisation function to retrieve from the model hyperparameters the chunks data tuples knowledge information and identify their true quality value score more accurately. Previous Data Shapley studies [10, 30] using SGD achieved that and identify the low-data quality and the high one more accurately. For each chunk $ch_j$, we compute its data quality value by considering all subsets $Z \in \mathcal{S}$ that do not contain chunk $ch_j$. Using the subsets $Z$, we create a model checkpoint $w_j$ for chunk $ch_j$, which includes all the data tuples informations from $Z$. One complete single run of our method results in a set of model checkpoints $W = (w_1, w_2, \cdots, w_c)$, where $c$ is the total number chunks. The Shapley value calculation for chunk $ch_j$, over a performance metric function $M$ is calculated as follows:

$$ds_{ch_j}^M = \sum_{Z \subseteq \mathcal{S} \setminus \{j\}} \frac{C}{|D| \cdot |Z|} \left( M(w_j(Z \cup \{j\})) - M(w_j(Z)) \right) \quad (2)$$

where for the selected subsets $Z \subseteq \mathcal{S}$ do not contain data tuples from chunk $ch_j$ are used to calculate its quality value. When chunk

---

**Algorithm 2** Subset Selection

---

**Require:** Chunks $ch$, total number of subsets $k$, Machine Learning algorithm $\mathcal{A}$, performance metric $M$, threshold $th$

1: **Initialize** randomly $k$ subsets $\mathcal{S}$
2: **for** each chunk $ch_j \in D$ for $j = 1, ..., c$ **do**
3:    **while** $ch_j$ is more than 25% in $Z \in \mathcal{S}$ **do**
4:       **Initialize** new subsets $Z$ for the $\mathcal{S}$
5:    **end while**
6: **end for**
7: **for** each Subset $Z_j \in \mathcal{S}$ for $j = 1, ..., k$ **do**
8:    **Get** $M_j$ from $\mathcal{A}$ using Subset $Z_j$
9:    **while** $M_j$ perform worst than the $th$ **do**
10:       **Initialize** new Subset $Z_j$ for the $\mathcal{S}$
11:       **Check** that none $ch$ is appear more than 25% in $Z \in \mathcal{S}$
12:       **Get** $M_j$ from $\mathcal{A}$ using new subset $Z_j$
13:    **end while**
14: **end for**
15: **Return** Subsets $\mathcal{S}$

---

$ch_j$ is not included in any subset, we use all the subsets from $\mathcal{S}$. First we obtain the data information knowledge from $Z$ from the performance metric $M$, and after that we update the model hyperparameters and create the new model checkpoint $w_j$ for chunk $ch_j$ and get its performance for metric $M$. The model checkpoint is calculated using the previous checkpoint $w_{j-1}$ and the optimised loss function using SGD $\mathcal{L}$, for all the chunks in subset $Z$ as:

$$w_j(Z) = w_{j-1} - \eta_j \sum_{x \in Z} \nabla \mathcal{L}(w_j, x) \tag{3}$$

where $\eta_j$ is the learning rate at chunk $ch_j$. This approach, using the model checkpoint information from all the previous data tuples, optimises the knowledge of the current chunk of data tuples to identify if it contains low-quality data tuples. This approximation with the combination of the optimal subset selection leads to an accurate Shapley value and identifies more easily the data chunks that contain low-quality data tuples, as well as require fewer iterations for computation the Shapley value. Similarly to the existing approaches a lower value for $ds_j$ means less valuable is for the ML algorithm $\mathcal{A}$. Algorithm 1 outlines the complete procedure, taking as input a collection a dataset with $n$ tuples, a machine learning algorithm $\mathcal{A}$, a performance metric $M$, and the chunk size $l$. The procedure is repeated until the truncation criterion is satisfied, following a similar approach to the work of [10].

*3.2.2 Subset Selection.* For Data Shapley approximations to come closer to the actual value of the original algorithm, it is essential to select an "optimal" amount of subsets. Initially, $k$ subsets are randomly chosen, ensuring that each new subset contains data points not present in more than twenty-five percent of the previously selected subsets [20]. The number of subsets $k$ is much smaller than the number of chunks $k << c$, as well as the length of the dataset ($k << n$). This reduces redundancy and helps the algorithm better capture diverse data contributions. To ensure that we select the subset with the highest quality of data tuples, using the algorithm $\mathcal{A}$ over all the subsets $\mathcal{S}$, we capture for each subset how well it performs over the metric $M$. Then, for each subset $j$ where his metric from the function $M$ is lower than the threshold $th$ for classification

task and higher for regression, we make the creation of the subset until achieved better performance than the threshold $th$. Algorithm 2, depict the subset selection method taking as input all the chunks $ch$, a machine learning algorithm $\mathcal{A}$, a performance metric $M$, and a threshold $th$.

## 4 Experimental Evaluation

Our proposed Data Shapley approximation method, *C-DaSh*, is evaluated against two baseline approximation algorithms from [10], available in an open-source repository. For all experiments involving a classification task, we employ a Multi-Layer Perceptron (MLP) [1, 25, 26] with two hidden layers, and for the regression task experiments, a similarly structured MLP tailored accordingly. The learning rate $\eta$ for all methods using SGD was set to 0.001. The parameters settings for the approximation algorithms are the same as those used in [10]. In our work, the setting for the threshold **th** for the subset selection has been set at 0.5 for the classification experiments and 25 for the regression experiments. The threshold for classification was selected because it corresponds to the mean of the accuracy range (0 to 1). For regression, the threshold was chosen the average RMSE after computed the algorithm $\mathcal{A}$ using the entire training dataset. First, we discuss the efficiency of removing low-quality data tuples from the datasets and compare MLP's prediction accuracy. These low-quality data tuples exhibit different such us noise, label corruption, missing data, etc. To evaluate prediction performance, we train the MLP model five times for each experiment, recording the prediction accuracy and reporting the average result. Using the Local Outlier Factor (LOF), we demonstrate the efficiency of our method in identifying outliers in large scale datasets. Therefore, we compare the computational efficiency of our approximation algorithm with the baseline methods from [10]. Relative speedup was calculated using the formula $S = \frac{T_1}{T_2}$, where $T_1$ is the execution time of the baseline method and $T_2$ is the execution time of *C-DaSh* with the optimal subset selection. We demonstrate the effectiveness of *C-DaSh* through its ability to identify low-quality chunks in regression datasets where the temporal order plays a critical role.

All experiments were executed in a AWS EC2 server with 48 vCPUs of AMD EPYC 7R32 processors at 2.40GHz, and four A10s GPUs with 24GB of memory each, $192GB$ of RAM memory, and $1.5TB$ of storage, running Ubuntu 24.4 LTS.

### 4.1 Datasets

**Table 1: Dataset properties for experimental evaluation**

| Algorithm | Dataset Name | # Files | # Tuples |
|---|---|---|---|
| Classification | Adult [5] | 1 | $40K$ |
| | Bank Marketing [23] | 1 | $60k$ |
| | MIMIC-III [15] | 1 | $55k$ |
| Regression | Air Quality [7] | 12 | $35K$ |
| | Household Power Consumption [13] | 5 | $22k$ - $52k$ |
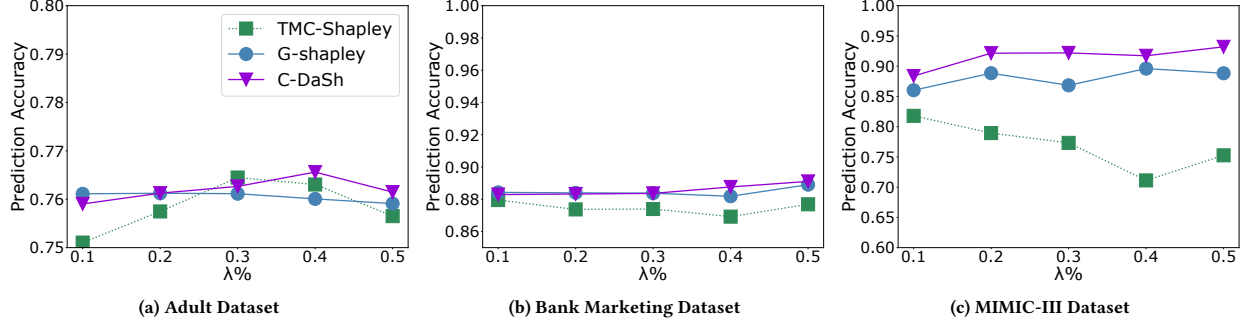
**Figure 1: Compare C-DaSh with G-Shapley and TMC-Shapley prediction accuracy by removing low data quality from datasets for classification task.**

We evaluated *C-DaSh* using five real-world datasets: Three for classification tasks and two for regression tasks. The Adult [5] dataset contains approximately forty thousand individuals with socio-economic features, and poses a binary classification problem: Predicting whether an individual's income exceeds a certain amount. It contains a substantial number of low-quality data tuples, often affected by missing or corrupted values. The Bank Marketing [23] dataset consists of data from a Portuguese banking institution, collected through direct marketing campaigns conducted via phone calls. It includes around sixty thousand individuals in a single file, with the classification goal of predicting whether a client will subscribe to a term deposit. The issue with the Bank dataset is that there are a lot of outliers in the data tuples. The MIMIC-III dataset, from Beth Israel Deaconess Medical Centre, is one of the largest available medical datasets, with over forty thousand patients from 2001 to 2012. After preprocessing, we created a single dataset to classify whether a patient has sepsis, following an approach similar to [29]. As noted in [29], the MIMIC-III dataset suffers from missing data tuples. The Air Quality [7] dataset comprises twelve separate datasets, each from a different nationally controlled air-quality monitoring site. Each dataset contains hourly air pollutants. The regression task is to predict the air quality. The primary data quality issue of this dataset is the high volume of missing data tuples. The Household Power Consumption (HPC) [13] dataset contains electric power usage measurements from a household in Sceaux, France. It includes five yearly datasets, each with between $22k$ and $52k$ data tuples recorded at one-minute intervals. The data quality problem in this dataset is the presence of missing values during certain periods of the day. Categorical columns across all datasets were converted to numerical format using one-hot encoding.

## 4.2 Baseline Comparison

*4.2.1 Prediction accuracy and outlier detection evaluation.* Figure 1 visualises the comparison of our proposed approximation to the G-Shapley and TMC-Shapley approximations on three classification datasets, since these two baselines apply only to classification tasks. In this experiment, we obtained the data Shapley values, and using *C-DaSh* we remove the bottom $\lambda\%$ (ranging from 0.1 to 0.5) of data chunks based on their Data Shapley values, where each chunk has a size of 256. The number of subsets $k$ used in our approximation

method was set to 50. For the G-Shapley and TMC-Shapley the $\lambda\%$ lowest Shapley values of individual data tuples are removed. Figure 1 for the three datasets depicts the accuracy (ranging from 0 to 1, where higher is better) as a performance metric $M$ of the ML algorithm $\mathcal{A}$ on the y-axis. On the axis-x, the percentage of removed data based on the lowest Shapley values is depicted.

Figure 1a shows that for the Adult dataset, our proposed method, *C-DaSh* has a low improvement impact in prediction accuracy as some of low-quality data are removed. It achieves slightly higher accuracy compared to both G-Shapley and TMC-Shapley across all levels of data removal. For example, with just 10% of the lowest-quality data tuples removed, our method have 1% higher accuracy than TMC-Shapley and almost the same prediction accuracy with G-Shapley. This small improvement stems from optimal subset selection, which allows our method to effectively target the low-quality chunks while preserving high-quality data. A drop in performance typically indicates that most low-quality data tuples has already been eliminated, and further removal begins to affect chunks which containing high-quality data tuples.

Figure 1b illustrates that, for the Bank marketing dataset TMC-Shapley for $\lambda$ equal to or lower than 40%, the accuracy slightly drops, suggesting that these methods mistakenly identify high-quality tuples as low-quality. In contrast, *C-DaSh*, as well as G-Shapley has a slight improvement until 30% has same performance without any up-down trend. That's show that both high-quality and low-quality data tuples are removes. For $\lambda$ value higher or equal with 40%, *C-DaSh* has higher prediction accuracy with a small difference. This further demonstrates that optimal subset selection assists our method minimise the removal of valuable data while effectively filtering out low-quality data, thereby preserving the training performance of algorithm $\mathcal{A}$.

Figure 1c demonstrates that, for the MIMIC-III dataset, *C-DaSh* surpass both G-Shapley and TMC-Shapley in accuracy by approximately 7% and 15%, respectively. TMC-Shapley exhibits a significant drop in performance accuracy up to $\lambda$ equal with 40%, indicating that it misclassifies high-quality tuples as low-quality. G-Shapley performs better in overall, however for $\lambda$ between 20% and 30% purges high-quality data features, while at higher $\lambda$, it begins to

(a) Gaussian Noise as low data quality



(b) Label Corruption as low data quality
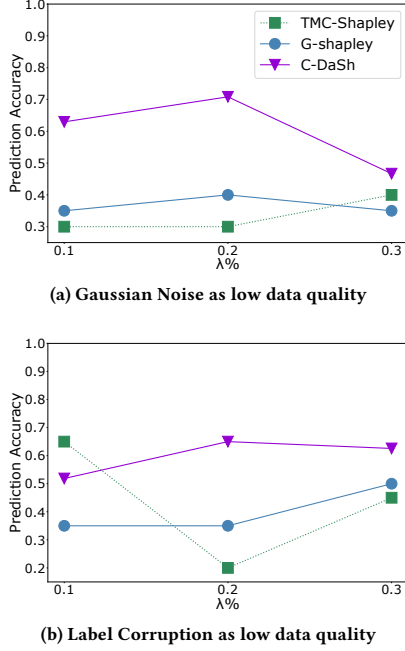
Figure 2: Compare C-DaSh with G-Shapley and TMC-Shapley prediction accuracy on Adult Dataset with manually annotated different low data quality issues.



(a) Adult Dataset



(b) MIMIC-III Dataset

Figure 3: Compare prediction accuracy between C-DaSh and a Chunk Average method where uses the quality values from G-Shapley or TMC-Shapley on Adult and MIMIC Datasets by remove the $\lambda\%$.

identify low-quality data tuples more effectively. *C-DaSh* consistently achieves higher accuracy than both baselines due to its optimal subset selection, which helps identify low-quality data chunks and retain high-quality ones for training algorithm $\mathcal{A}$ across all $\lambda$ values.

Figure 2 illustrates two different low-quality data scenarios for the Adult dataset, Gaussian noise added to 20% of the data tuples (Figure 2a) and label corruption where 20% of the labels were flipped (Figure 2b). In both experiments, our goal is to exclude data tuples with missing values for better evaluation approximation algorithms' ability to identify and handle specific data quality issues. In both Figures, the x-axis represent the percentage ($\lambda\%$) of removed data tuples, while the y-axis reports the prediction accuracy of the learning algorithm $\mathcal{A}$. At $\lambda$ equal to 20%, *C-DaSh* successfully removes most of the low-quality data tuples and outperforms both G-Shapley and TMC-Shapley by 30% and 40% respectively. However, G-Shapley identifies some of the high-quality data tuples until $\lambda$ equals 20%, but for higher $\lambda$ values mistakenly classifies some high-quality data tuples as low-quality, and that leads to a significant drop in prediction accuracy. In the label corruption from Figure 2b, TMC-Shapley identifies more low data quality data tuples at $\lambda$ equals with 10% and outperforms *C-DaSh* by almost 12%. However, for higher $\lambda$ values, *C-DaSh* consistently outperforms both methods.

We designed a second method in which we chunk the datasets evaluate the ability of chunking as a method, using either the G-Shapley or TMC-Shapley approximation. The overall quality value of a chunk is computed by averaging for each chunk all of its tuple Shapley scores. As in our proposed *C-DaSh* method, each chunk
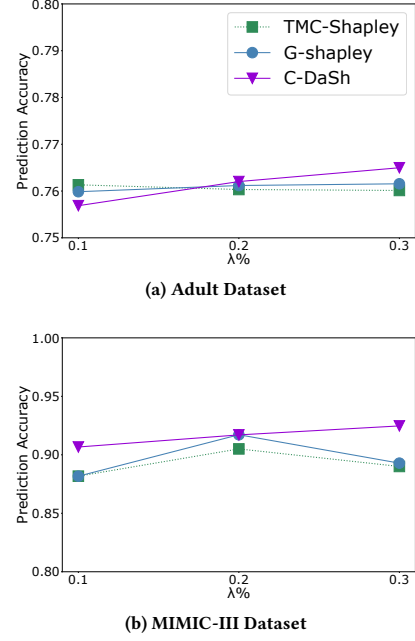
is assigned a quality value based on one of the two approximation methods. To evaluate how accurately *C-DaSh* identifies chunk quality score value, we compare it with the chunked version of both approximation methods. All the three methods use the same chunks containing equal number of data tuples (chunk size of 250 data tuples for the Adult and MIMIC-III datasets). Figure 3 presents the prediction accuracy results from these experiments. The y-axis of both sub-figures depicts the prediction accuracy, while the x-axis indicates the percentage of removed low-quality chunks (ranging from 10% to 30%). For the Adult dataset (Figure 3a), when 10% of the chunks were removed, both baseline methods slightly outperformed *C-DaSh*, with less than a 0.5% improvement. Furthermore, at higher removal percentages, *C-DaSh* outperformed the chunk-averaging with the data quality values based on the baselines' approximations. This indicates that *C-DaSh* can more precisely identify low-quality chunks with more precision compare to the baseline averaging method. For the MIMIC-III dataset (Figure 3b), *C-DaSh* consistently achieved higher accuracy than the chunk averaging using the TMC-Shapley across all removal percentages. While G-Shapley at $\lambda$ equals with 20%, achieved prediction accuracy comparable to *C-DaSh*. In contrast, *C-DaSh* continued to identify low-quality chunks more effectively. However, both of baseline chunking methods either performs similarly (G-Shapley), or better (TMC-Shapley) than *C-DaSh* on both datasets. This demonstrates that employing chunking is beneficial for the dataset, as it enables more precise identification of low-quality data chunks.

**Table 2: Data Shapley Approximations Outlier Detection Average Error over $\lambda\%$ percentage of removing low data quality data tuples**

| Dataset Name | DS Approximation Method | $\lambda = 0.1$ | $\lambda = 0.2$ | $\lambda = 0.3$ |
|---|---|---|---|---|
| Adult | C-DaSh | **1.260** | 1.262 | 1.266 |
| | TMC-Shapley | 1.274 | 1.312 | 1.327 |
| | G-Shapley | 1.275 | 1.312 | 1.327 |
| Bank Marketing | C-DaSh | **1.046** | **1.046** | 1.049 |
| | TMC-Shapley | 1.096 | 1.076 | 1.068 |
| | G-Shapley | 1.096 | 1.076 | 1.068 |
| MIMIC-III | C-DaSh | **1.148** | 1.150 | 1.153 |
| | TMC-Shapley | 1.348 | 1.340 | 1.303 |
| | G-Shapley | 1.348 | 1.340 | 1.303 |

Table 2 presents the absolute outlier factor scores obtained using the Local Outlier Factor (LOF) [6] method for the three classification datasets of Adult, Bank Marketing, and MIMIC-III. LOF detects anomalies by comparing a point's local density to that of its neighbours, flagging points with significantly lower density as outliers. In this experiment, we applied our proposed *C-DaSh* method and compare it against G-Shapley and TMC-Shapley approximations. For each method, we removed the lowest-quality data either chunks (for *C-DaSh*) or individual tuples (for the baselines) at removal rates of $\lambda$ equal with 0.1, 0.2, and 0.3. A lower absolute LOF score indicates a more effective removal of outlier data tuples. In Table 2, bold values represent the best-performing method for each dataset and $\lambda$ value. Our proposed method consistently outperformed both of G-Shapley and TMC-Shapley across all datasets. This demonstrates that chunk-based subset selection in *C-DaSh* provides a more robust mechanism for identifying and removing low-quality or outlier data. Also, is notably at $\lambda$ equal with 0.1, *C-DaSh* achieved the lowest absolute LOF score for all the datasets, suggesting that most outliers are captured early in the removal process. As $\lambda$ increases, the absolute LOF score increases, indicating that some high-quality tuples are removed. This suggests that the most uninformative or noisy data are identified and eliminated very early. Additionally, compared to G-Shapley, which also leverages information from SGD for the Shapley value estimation, *C-DaSh* consistently demonstrates superior with chunk-based evaluation strategy.

*4.2.2 Speedup Evaluation.* Table 3 depicts the speedup achieved between *C-DaSh* and baseline methods [10] (G-Shapley and TMC-Shapley). For each dataset, the chunk size that achieved the highest speed over the two baseline methods is highlighted in bold. Our *C-DaSh* method, which uses data chunks rather than individual tuples, achieves significant speedups in computation. Finding the contribution of chunks, rather of individual data tuple, highly improves computational efficiency among all datasets. Increasing the chunk size up to 250 data tuples leads to improved computational efficiency. This improvement occurs because fewer chunks exist and the subset selection requires less time, allowing more effective identification of high-quality data chunks that contribute positively to the learning algorithm. However, when the chunk size exceeds

**Table 3: C-DaSh Speedup over the G-Shapley and TMC-Shapley computational efficiency**

| Dataset Name | Chunk Size | Speedup over G-Shapley ($\times$) | Speedup over TMC-Shapley ($\times$) |
|---|---|---|---|
| Adult | 50 | 23.6 | 413.11 |
| | 100 | 31.76 | 555.88 |
| | 150 | 41.14 | 720 |
| | 250 | **108** | **1890** |
| | 500 | 25.71 | 450 |
| | 1000 | 24.68 | 432 |
| Bank Marketing | 50 | 36.56 | 618.75 |
| | 100 | 75.36 | 1277.41 |
| | 150 | 77.6 | 1315.37 |
| | 250 | **80.68** | **1365.51** |
| | 500 | 49.26 | 833.68 |
| | 1000 | 48.24 | 816.49 |
| MIMIC-III | 50 | 49.31 | 467.12 |
| | 100 | 74.22 | 853.61 |
| | 150 | 112.5 | 1293.75 |
| | 250 | **200** | **2300** |
| | 500 | 53.73 | 617.912 |
| | 1000 | 36.54 | 420.30 |

250 tuples, the computational efficiency of our *C-DaSh* drops, This is because chunks are more likely to contain both high-quality and bigger proportion of low-quality data tuples, making it harder to identify the highest-quality chunks during subset selection.

## 4.3 Chunk Size Evaluation



**(a) Adult Dataset**



**(b) Bank Marketing**
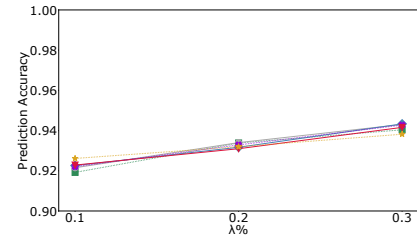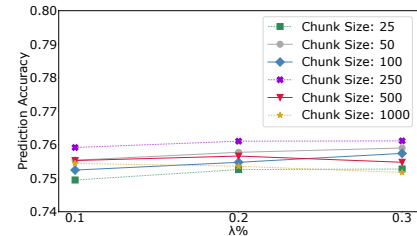
**Figure 4: Prediction Accuracy comparison over different chunk size by removing the $\lambda\%$ lowest data quality chunks.**

In this experiment, we evaluated the impact of varying chunk sizes on the performance of our proposed approximation method, *C-DaSh*, using the Adult and Bank Marketing datasets. We varied the chunk size from 50 to 1000 data tuples to assess its effect on the accuracy of the machine learning algorithm $\mathcal{A}$. For all the experiments, we set the number of subsets $k$ to 50. We removed the lowest-quality data chunks, with $\lambda$ ranging from 0.1 to 0.3. Figure 4 presents the results of this experiment. Figure 4a corresponds to the Adult dataset, while Figure 4b shows results for the Bank Marketing dataset. In both sub-figures, the y-axis represents the prediction accuracy, while the x-axis shows the $\lambda$ percentage of removed low-quality data tuples. For the adult dataset (Figure 4a), we observe that for chunk sizes between 50 and 256 data tuples, accuracy did not significantly change as more low-quality chunks were removed. However, for chunk size equal to 256 our method achieved its peak accuracy with a chunk size around 0.5% of the total dataset size. As the chunk size increased to 500 and 1000 tuples, the performance of the algorithm $\mathcal{A}$ slightly drops. This declines is due to valuable data tuples being included in the removed chunks, which negatively impacts the algorithm's learning process. For the Bank Marketing dataset (Figure 4b), at $\lambda$ equal to 0.1, a chunk size of 1000 tuples achieves the highest accuracy, with a small margin over the next best size. This suggests that for larger chunk sizes can effectively identifies low-quality data tuples, although the accuracy gains may be marginal compared to smaller chunks. However, the differences between the prediction accuracy and chunk sizes remains relatively small. This depicts *C-DaSh*, using the subset selection, effectively identifies low-quality chunks without affecting the prediction accuracy. Based on Figure 4 and Table 3, a chunk size of approximately 250 data tuples offers the most balanced trade-off between speedup and prediction accuracy.

## 4.4 Subset Size Evaluation

Figure 5 illustrates the experiment to evaluate *C-DaSh* under varying subset sizes on the Adult (Figure 5a) and MIMIC-III (Figure 5b). In this experiment, the subset size $k$ ranged from 25 to 500, while the chunk size was fixed at 256 data tuples for both datasets, because the trade-off between the chunk size and the speedup. In both sub-figures, the x-axis represents the $\lambda\%$ of the removed low data quality chunks, and the y-axis the prediction accuracy. The results indicate that increasing the subset size $k$ had minimal impact to the impact on prediction accuracy for both datasets, as the accuracy remained close. This suggests that the subset selection step in *C-DaSh*, effectively helps identify the most optimal subsets and improve the prediction accuracy regardless the $k$ size. Notably, a subset size $k$ of 50 achieves the best performance, with accuracy only 0.05% below the second-best result for the Adult dataset, and 1% lower for MIMIC-III. Overall, these results highlight that *C-DaSh* is robust across varying subset sizes and can deliver high-accuracy performance even with relatively few subsets.

## 4.5 Experimental evaluation on regression datasets

Figure 6 illustrates the performance of *C-DaSh* on two regression datasets, demonstrating its effectiveness in regression tasks. For this experiment, we used Root Mean Square Error (RMSE) as the



(a) Adult Dataset



(b) MIMIC-III Dataset
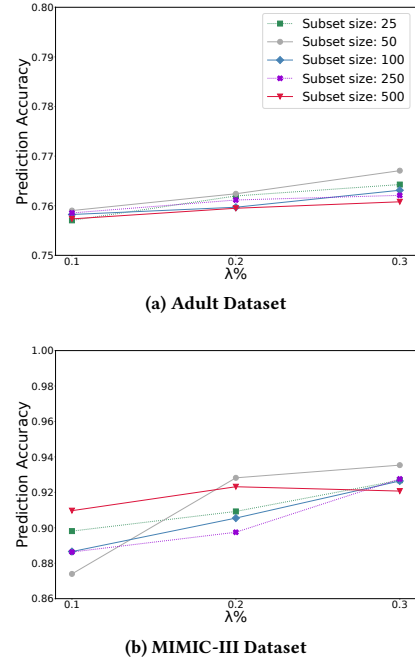
Figure 5: Prediction Accuracy comparison on *C-DaSh* over different subset size by removing the $\lambda\%$ lowest data quality chunks.



(a) HPC Dataset  (b) Air Quality
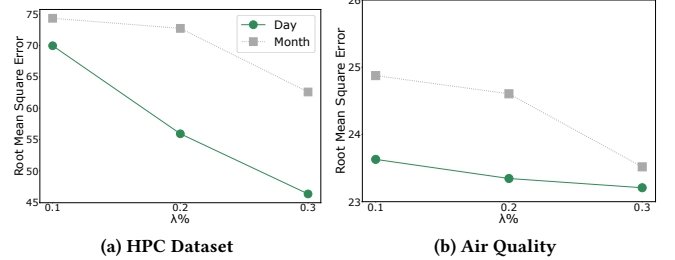
Figure 6: *C-DaSh* prediction accuracy on Regression Dataset for daily and monthly chunks by removing the $\lambda\%$ lowest data quality chunks.

performance metric, as it effectively captures both the magnitude and direction of prediction errors. Figure 6a shows the prediction accuracy of the HPC dataset, and Figure 6b for the Air Quality dataset. For each dataset from the $n$ available datasets from both Air quality and HPC, we removed $\lambda\%$ of the chunks with the lowest quality value $DS_j$. In both sub-figures, the x-axis shows the percentage of removed low-quality data chunks, and the y-axis shows the RMSE (lower is better) prediction error loss function. Since regression datasets are sensitive to temporal order, we preserved this order by constructing chunks based on daily or monthly intervals, ensuring the natural structure of the data remained intact. For both experiments, the number of subsets $k$ was fixed at 50. The results show that our method for both daily and monthly effectively identifies and removes low-quality chunks, leading to reduced RMSE as $\lambda\%$ increases. Furthermore, we observe that for a daily chunk

size, our algorithm performs better and can identify the days that include missing or corrupted data. In contrast, for monthly chunks, the error is higher because removing a chunk may eliminate both low-quality and high-quality data tuples. On this experiment, we evaluate only *C-DaSh*, as the baseline methods are not applicable to regression algorithms $\mathcal{A}$, with the exception of logistic regression, which is primarily used for classification tasks.

## 5 Related Work

The goal of data quality methods is to examine dataset content and select the "right" data for training a Machine Learning algorithm instead of boosting the algorithm itself. Shapley value [28] has been used to select the identical or valuable data tuples for the training of Machine Learning algorithm. However, because of the exponential computational complexity time of Shapley value calculation, approximations studies are suggested to reduce complexity time without affecting the accuracy of detecting the low-quality data tuples with traditional data errors (such as outliers, corrupted data, missing values, etc). Study from [10], proposed two approximations (Truncated Monte Carlo Shapley and Gradient Shapley) to improve the identification of low-quality data tuples. Identifying the most optimal subsets to improve the quality of Shapley values has been suggested and implemented from [20]. Their method uses a kernel Hilbert space in the permutation functions and checks the connections between permutations and the hypersphere, tries to identify the highest-quality subsets. On lower-dimensional problems, their method has been shown to be more effective than the higher-dimensional problems. 2-D Shapley [16] is a Data Shapley approximation where, based on the two-dimensional counterfactual calculation, they abstract the block valuation problem into a two-dimensional cooperative game. Their proposed axioms for the block valuation results to a unique representation of the Shapley value for each block. In [30], noted the high complexity time issue on the Data Shapley and its approximation methods when applied to different subsets. To address this, they proposed an approximation that uses first or second-order Taylor expansions and gradient dot-products or gradient-Hessian products between training and validation data, to compute the Shapley value for each data tuple. A distributional approach to Data Shapley computation was proposed in the study by [9], where the Shapley value of a data tuple is defined with respect to an underlying data distribution. A Multi-Modal Data Shapley approach was proposed by [19], introducing the *Shapley Value-based Contrastive Alignment* (Shap-CA) method to align image and text modalities using a Data Shapley approximation. In their framework, each image-text pair is considered as a player, and its vector embedding is extracted using a Large Language Model (LLM). Contrastive alignment is then applied to align the embeddings of the paired objects. The Shapley value measures the marginal contribution of each image-text pair to the overall vector embedding representation alignment. A higher Shapley value indicates a stronger semantic alignment and greater mutual contribution between the image and text components.

Other previous studies tried to identify low-quality data tuples without using the Data Shapley value. Commet [22] optimises ML algorithms through a step-by-step recommendation that identify and clean low-quality data features with data errors (e.g., missing values, Gaussian noise, categorical shift, and scaling), while maximising data cleaning efficiency under resource constraints. The study by [21] demonstrates six data quality dimensions relevant to ML and AI algorithms and proposes various data quality measurement metrics to asses the quality of each data tuple on each of these issues. The study aims to demonstrate the relationship between the data quality dimensions and the main impact on the algorithm performance. Our work builds on the concept of Data Shapley by partitioning the dataset into equally sized chunks and computing a quality score for each chunk, rather than for each individual data tuple as done in previous Data Shapley approximation methods. Additionally, we significantly reduce computational complexity by approximating the Data Shapley values using only $k$ selected subsets of data chunks, where each chunk contains $l$ data tuples ($l \ll n$).

## 6 Conclusions and Future Work

In this work, we addressed the challenge of assessing the "quality" of a dataset's data tuple. We introduced a novel data Shapley approximation method, termed *C-DaSh*, which estimates the quality of individual data chunks. By incorporating a subset selection step before the data Shapley iterations, our approach effectively identifies the most suitable quality subsets. Leveraging information retrieved from stochastic gradient descent (SGD) enhances our approximation to identify the low-quality data tuples more accurately. Our method can be applied to both classification and regression tasks in machine learning models that use SGD as the optimisation algorithm. Empirically, we demonstrated that our approach outperforms two recent approximations, G-Shapley and TMC-Shapley [10], in identifying low-quality data tuples across various data quality issues, while also achieving significant speedups from 80× to 2300×. Additionally, we explored the trade-offs between execution time and accuracy over different chunk sizes, and analysed how subset size impacts the precision of detecting low-quality data. For future work, we aim to improve our subset selection strategy by introducing a data structure that groups low-quality data tuples in the same chunks, thereby enhancing the precision of quality identification. Furthermore, we plan to extend our method to support *multi-modal datasets* that combine textual descriptions (e.g., captions or subtitles) with images or videos, enabling the evaluation of data quality across different modalities.

## 7 GenAI Usage Disclosure

Generative AI tools such as Grammarly and ChatGPT were used in some sections simply to improve the text's flow and address grammar and syntax mistakes.

## Acknowledgments

## References

[1] Luis B Almeida. 2020. Multilayer perceptrons. In *Handbook of Neural Computation*. CRC Press, C1–2.
[2] Shun-ichi Amari. 1993. Backpropagation and stochastic gradient descent method. *Neurocomputing* 5, 4-5 (1993), 185–196.

[3] Aki Barry, Lei Han, and Gianluca Demartini. 2023. On the impact of data quality on image classification fairness. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2225–2229.

[4] Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. 2009. Methodologies for data quality assessment and improvement. *ACM computing surveys (CSUR)* 41, 3 (2009), 1–52.

[5] Barry Becker and Ronny Kohavi. 1996. Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20.

[6] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 93–104.

[7] Song Chen. 2017. Beijing Multi-Site Air Quality. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5RK5G.

[8] Xiaotie Deng and Christos H Papadimitriou. 1994. On the complexity of co-operative solution concepts. *Mathematics of operations research* 19, 2 (1994), 257–266.

[9] Amirata Ghorbani, Michael P. Kim, and James Zou. 2020. A Distributional Framework for Data Valuation. arXiv:2002.12334 [cs.LG] https://arxiv.org/abs/2002.12334

[10] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*. PMLR, 2242–2251.

[11] Ioannis Giannakopoulos, Dimitrios Tsoumakos, and Nectarios Koziris. 2018. A Content-Based Approach for Modeling Analytics Operators. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 227–236.

[12] Benjamin T Hazen, Christopher A Boone, Jeremy D Ezell, and L Allison Jones-Farmer. 2014. Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. *International Journal of Production Economics* 154 (2014), 72–80.

[13] Georges Hebrail and Alice Berard. 2006. Individual Household Electric Power Consumption. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C58K54.

[14] Johannes Jakubik, Michael Vössing, Niklas Kühl, Jannis Walk, and Gerhard Satzger. 2024. Data-centric artificial intelligence. *Business & Information Systems Engineering* 66, 4 (2024), 507–515.

[15] Alistair E W Johnson, David J Stone, Leo A Celi, and Tom J Pollard. 2018. The MIMIC Code Repository: enabling reproducibility in critical care research. *Journal of the American Medical Informatics Association* 25, 1 (2018), 32–39.

[16] Zhihong Liu, Hoang Anh Just, Xiangyu Chang, Xi Chen, and Ruoxi Jia. 2023. 2d-shapley: A framework for fragmented data valuation. In *International Conference on Machine Learning*. PMLR, 21730–21755.

[17] Andreas Loizou, Nikolaos Andriotis, and Dimitrios Tsoumakos. 2025. VEnOM: A Vector Embedding Operator Modelling Framework. In *Proceedings of the 19th ACM International Conference on Distributed and Event-based Systems*. 245–248.

[18] Andreas Loizou and Dimitrios Tsoumakos. 2025. Analytics Modelling over Multiple Datasets Using Vector Embeddings. In *Proceedings of the 36th International Conference on Database and Expert Systems Applications (DEXA 2025)*. Springer, 237–253.

[19] Wen Luo, Yu Xia, Shen Tianshu, and Sujian Li. 2024. Shapley Value-based Contrastive Alignment for Multimodal Information Extraction. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 5270–5279.

[20] Rory Mitchell, Joshua Cooper, Eibe Frank, and Geoffrey Holmes. 2022. Sampling permutations for shapley value estimation. *Journal of Machine Learning Research* 23, 43 (2022), 1–46.

[21] Sedir Mohammed, Lukas Budach, Moritz Feuerpfeil, Nina Ihde, Andrea Nathansen, Nele Noack, Hendrik Patzlaff, Felix Naumann, and Hazar Harmouch. 2025. The effects of data quality on machine learning performance on tabular data. *Information Systems* 132 (2025), 102549.

[22] Sedir Mohammed, Felix Naumann, and Hazar Harmouch. 2025. Step-by-Step Data Cleaning Recommendations to Improve ML Prediction Accuracy. In *International Conference on Extending Database Technology (EDBT)*.

[23] S. Moro, P. Rita, and P. Cortez. 2014. Bank Marketing. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5K306.

[24] Evaggelia Pitoura. 2020. Social-minded measures of data quality: fairness, diversity, and lack of bias. *Journal of Data and Information Quality (JDIQ)* 12, 3 (2020), 1–8.

[25] MARIUS-CONSTANTIN Popescu, VALENTINA Balas, ONISIFOR Olaru, and NIKOS Mastorakis. 2009. The backpropagation algorithm functions for the multilayer perceptron. In *Proceedings of the 11th WSEAS International Conference on Sustainability in Science Engineering*. 28–31.

[26] Frank Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* 65, 6 (1958), 386.

[27] Alvin E Roth. 1988. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press.

[28] Lloyd S Shapley et al. 1953. A value for n-person games. (1953).

[29] Charithea Stylianides, Andria Nicolaou, Waqar Aziz Sulaiman, Christina-Athanasia Alexandropoulou, Ilias Panagiotopoulos, Konstantina Karathanasopoulou, George Dimitrakopoulos, Styliani Kleanthous, Eleni Politi, Dimitris Ntalaperas, et al. 2025. AI Advances in ICU with an Emphasis on Sepsis Prediction: An Overview. *Machine Learning and Knowledge Extraction* 7, 1 (2025), 6.

[30] Jiachen T Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. 2024. Data shapley in one training run. *arXiv preprint arXiv:2406.11011* (2024).

[31] Jiachen T. Wang, Tianji Yang, James Zou, Yongchan Kwon, and Ruoxi Jia. 2024. Rethinking Data Shapley for Data Selection Tasks: Misleads and Merits. arXiv:2405.03875 [cs.LG] https://arxiv.org/abs/2405.03875

[32] Richard Y Wang and Diane M Strong. 1996. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems* 12, 4 (1996), 5–33.

[33] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. 2025. Data-centric artificial intelligence: A survey. *Comput. Surveys* 57, 5 (2025), 1–42.