A Comprehensive Study of Shapley Value in Data Analytics

Hong Lin*
The State Key Laboratory of Blockchain and Data Security, Zhejiang University
honglin@zju.edu.cn

Shixin Wan*
The State Key Laboratory of Blockchain and Data Security, Zhejiang University
wansx@zju.edu.cn

Zhongle Xie*†
The State Key Laboratory of Blockchain and Data Security, Zhejiang University
xiezl@zju.edu.cn

Ke Chen*
The State Key Laboratory of Blockchain and Data Security, Zhejiang University
chenk@zju.edu.cn

Meihui Zhang
School of Computer Science & Technology, Beijing Institute of Technology
meihui_zhang@bit.edu.cn

Lidan Shou*†
The State Key Laboratory of Blockchain and Data Security, Zhejiang University
should@zju.edu.cn

Gang Chen*
The State Key Laboratory of Blockchain and Data Security, Zhejiang University
cg@zju.edu.cn

ABSTRACT

Over the recent years, Shapley value (SV), a solution concept from cooperative game theory, has found numerous applications in data analytics (DA). This paper presents the first comprehensive study of SV used throughout the DA workflow, clarifying the key variables in defining DA-applicable SV and the essential functionalities that SV can provide for data scientists. We condense four primary challenges of using SV in DA, namely computation efficiency, approximation error, privacy preservation, and interpretability, disentangle the resolution techniques from existing arts in this field, then analyze and discuss the techniques w.r.t. each challenge and the potential conflicts between challenges. We also implement *SVBench*, a modular and extensible open-source framework for developing SV applications in different DA tasks, and conduct extensive evaluations to validate our analyses and discussions. Based on the qualitative and quantitative results, we identify the limitations of current efforts for applying SV to DA and highlight the directions of future research and engineering.

PVLDB Reference Format:
Hong Lin, Shixin Wan, Zhongle Xie, Ke Chen, Meihui Zhang, Lidan Shou, and Gang Chen. A Comprehensive Study of Shapley Value in Data Analytics. PVLDB, 18(9): 3077 - 3092, 2025.
doi:10.14778/3746405.3746429

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/zjuDBSystems/SVBench.

*Also affiliated with Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security.
†Zhongle Xie and Lidan Shou are the corresponding authors.

Proceedings of the VLDB Endowment, Vol. 18, No. 9 ISSN 2150-8097.
doi:10.14778/3746405.3746429

1 INTRODUCTION

Data analytics (DA), exploring data to mine insightful information for problem-solving, has garnered significant attention in industry and academia over the past few years [52, 114]. The global DA market was valued at USD 64.99 billion in 2024 and is projected to reach USD 402.70 billion by 2032, with a compound annual growth rate of 25.5% from 2024 to 2032 [105]. A typical DA workflow generally follows three key stages: (1) **Data Fabrication** [46, 95], encompassing tasks such as *data collection (DC)* [7, 20] to identify, retrieve, and transfer data from diverse data sources to the analysis platform, and *data orchestration (DO)* [82, 170] to cleanse and transform data to align with downstream analytical requirements. (2) **Data Exploration**, including *data valuation (DV)* to preserve high-value, refined data, and *data mining (DM)* to uncover patterns and insights using techniques such as Machine Learning (ML). (3) **Result Reporting**, involving *result interpretation (RI)* [12, 18, 29] to render the analytical outcomes comprehensible, and *result trading (RT)* to bargain and exchange DA derivatives such as trained ML models in data marketplaces [19, 157].

Recently, data scientists have applied Shapley value (SV), a method derived from cooperative game theory for fairly distributing the total gains generated by the coalition of all players [113], to numerous tasks throughout the DA workflow. Figure 1 depicts a series of examples applying SV to the tasks analyzing medical images, where the analytical objects include pixels, images, image sets, ML models, etc. The application purposes can be summarized into four categories: (1) **pricing**, to determine the net worth of analytical objects for trading, such as buying image datasets in the DC task and selling well-trained models in the RT task; (2) **selection**, to select qualified and important analytical objects for exploration, for example, selecting pixels important to reduce training losses to learn models in the DO task; (3) **weighting**, to assign reasonable weights to analytical objects collected from multiple sources for valid fusion of those objects, for instance, weighting local models collected in the DM task, where federated learning (FL) [70, 85] is used to protect privacy within medical images, for fusing as a valid global model whose accuracy exceeds a certain threshold; (4)

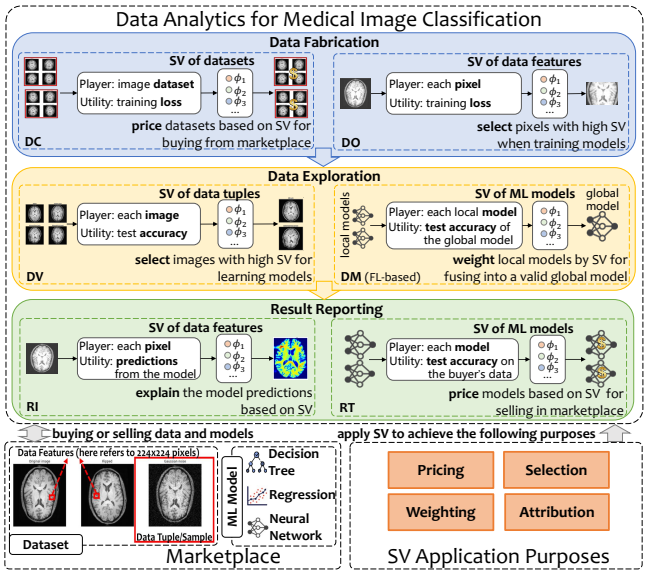


Figure 1: An example of using Shapley values ($\phi_1, \phi_2, \phi_3, \cdots$) throughout the data analytics workflow.

attribution, to explain data exploration outputs, e.g., explaining how pixels impact the model predictions in the RI task.

To better understand the usage of SV in the DA domain, several surveys have been proposed [12, 17, 65, 94, 100, 108, 122, 128]. However, they do not cover the entire life cycle of DA, leaving a gap in a holistic guidance for applying SV to diverse DA tasks. As summarized in Table 1, prior works predominantly concentrated on ad-hoc SV implementations, mainly addressing computation efficiency instead of other challenges such as interpretability. Furthermore, they analyze SV applications or computing algorithms as monolithic units rather than decomposing them into reusable building blocks, which is a limitation that stifles the development of modular extensible frameworks from a DA application perspective. For instance, existing tools like SHAP [76] (tailored to RI tasks) and DataShapley [43] (bound to DV tasks) are constrained by rigid task-specific configurations, sacrificing flexibility and universality. These gaps leave practitioners ill-equipped to navigate SV's foundational design principles (such as cooperative game modeling) or resolve conflicting challenges (e.g., approximation error vs. efficiency) when adapting SV to new DA tasks.

In this work, we endeavor to bridge existing gaps through an in-depth survey analyzing the application of SV across the entire spectrum of DA. By synthesizing the insights, this paper contributes to not only a deeper understanding of the SV's potential to enhance DA but also the implementation of a modular extensible framework for SV application development, laying the groundwork for both the practical implementation of SV in real-world DA systems and the advancement of academic research in this exciting field. We expect this paper to be a helpful resource for both newcomers to the field and seasoned experts seeking a consolidated and systematic update on current developments.

The contributions of this paper are outlined as follows:

Survey	Application Purposes				Solutions For				Units of Analysis	Framework
	pric.	sele.	weig.	attr.	eff.	err.	priv.	int.		
[100, 128]	✔	✖	✖	✖	✔	✖	✖	✖	Applications or Algorithms	✖
[108]	✖	✔	✖	✔	✔	✔	✖	✖		✖
[78]	✔	✔	✖	✔	✔	✔	✖	✖		✖
[122]	✖	✔	✖	✖	✔	✖	✔	✖		✖
[12, 17, 94]	✖	✖	✖	✔	✔	✖	✖	✖		✖
[65]	✔	✔	✖	✔	✔	✖	✖	✖		✖
ours	✔	✔	✔	✔	✔	✔	✔	✔	Techniques	✔

Table 1: Comparison with related works. Our work focuses on the techniques (or called building blocks of SV applications and algorithms) solving four challenges of SV: computation efficiency (eff.), approximation error (err.), privacy preservation (priv.), and interpretability (int.).

- We present the first comprehensive survey of SV applied throughout the DA workflow, clarify the key variables in defining DA-applicable SV, and reveal the essential functionalities that SV can provide for DA. We also condense four technical challenges of using SV in DA, analyze and discuss the building blocks of solutions w.r.t. each challenge and potential conflicts between challenges. (§3)
- We propose *SVBench*, a modular and extensible open-source framework for developing SV applications. *SVBench* integrates abundant techniques addressing key challenges of using SV in DA and allows a flexible usage of these techniques to build up new algorithms. The developers, as well as the academicians, could extend the framework with flexible APIs for succeeding studies on SV. (§4)
- We conduct extensive evaluations to consolidate our analyses and discussions, involving possible technical combinations for efficient computation, trade-offs in handling different challenges, and key factors affecting mainstream SV interpretations. Experiment results also validate the usability and modularity of *SVBench*. Through experiments, we reveal limiting factors of existing efforts and highlight future research and engineering directions. (§5)

2 PRELIMINARIES

This section introduces a general definition of the SV used in DA. For a more accessible presentation, we start by introducing the cooperative game, an important concept used in SV definition.

A cooperative game is composed of *a player set* and *a utility function* that defines the utility of each coalition (i.e., a subset of the player set). The formal definitions are stated as follows.

Definition 1. ***Player set, coalition, and cooperative game.*** *Let $\mathcal{N} = \{p_1, \cdots, p_n\}$ be a finite set of players. A coalition is a nonempty subset $\mathcal{S} \subseteq \mathcal{N}$ and the grand coalition is $\mathcal{N}$ itself. A cooperative game, denoted by $C(\mathcal{N}, U)$, consists of a player set $\mathcal{N}$ and a utility function $U(\cdot)$ that maps each coalition to a scalar value, i.e., $U : \mathcal{P}(\mathcal{N}) \to \mathbb{R}$, where $\mathcal{P}(\mathcal{N})$ is the power set of $\mathcal{N}$ and $U(\emptyset) = 0$.*

For any $\mathcal{S} \subseteq \mathcal{N}$, $U(\mathcal{S})$ represents the sum of the expected utility that the members of $\mathcal{S}$ can achieve through cooperation and is available for distribution among the members of $\mathcal{S}$.

SV is a method in cooperative game theory designed to fairly allocate the overall utility generated by the collective efforts of all players within a game. SV assigns a value to each player in the game, based on the player's marginal contribution to each possible coalition's utility, especially considering the case where the player is not part of the coalition. Intuitively, SV captures the essence of how much one coalition's utility increases (or decreases) with the inclusion of a new player, providing a fair and quantifiable measure of each player's influence on the game's overall utility.

SV has already been widely applied in numerous DA tasks modeled as cooperative games. According to the application purpose, existing works fall into 4 high-level categories, i.e., pricing, selection, weighting, and attribution, as presented in §1. Within each category, SV applications can be further classified into 8 finer-grained subcategories based on the definition of player and utility in the cooperative game, as shown in Figure 2. Through a comprehensive review of these applications, we formalize a general definition of SV used in DA as follows. A detailed discussion of SV applied for different purposes will be presented in the next section.

Definition 2. ***Shapley value in data analytics.*** *Given a task modeled as $C(\mathcal{N}, U)$, where each player $p_i \in \mathcal{N}$ is an analytical object and the utility $U(\cdot)$ refers to an analytical outcome or the outcome evaluation score, the Shapley value ϕ_i is a numerical value representing the weighted average of marginal contributions made by the player p_i to $U(\mathcal{S})$ produced by each coalition $\mathcal{S} \subseteq \mathcal{N} \backslash \{p_i\}$.*

$$\phi_i = \sum_{\mathcal{S} \subseteq \mathcal{N} \backslash \{p_i\}} \underbrace{\frac{|\mathcal{S}|!(n - |\mathcal{S}| - 1)!}{n!}}_{weight\ factor} \underbrace{[U(\mathcal{S} \cup \{p_i\}) - U(\mathcal{S})]}_{marginal\ contribution} \quad (1)$$

$$= \sum_{O \in \pi(\mathcal{N})} \frac{1}{n!} \underbrace{[U(P(O, p_i) \cup \{p_i\}) - U(P(O, p_i))]}_{marginal\ contribution}. \quad (2)$$

Equations 1 and 2 show the mathematical formulation of SV from different perspectives. Equation 1 is given based on Definition 2, while Equation 2 is given from the perspective of expectation calculation. The two formulas are interchangeable in SV computation. However, Equation 2 would be preferred if a developer expects to compute SV using expectation calculation techniques in mathematical statistics (even if not originally devised for SV, e.g., Monte Carlo methods). In Equation 2, $\pi(\mathcal{N})$ is the set of all possible permutations of players in the grand coalition, and $P(O, p_i)$ is the set of predecessors of player p_i in a specific permutation $O \in \pi(\mathcal{N})$.

3 THE SHAPLEY VALUE IN DATA ANALYTICS

In this section, we first introduce cooperative game modeling, a critical step to apply SV to DA tasks (§3.1), and then discuss the challenges of using SV in DA and the corresponding solutions (§3.2).

3.1 Cooperative Game Modeling

The fundamental step of applying SV for a DA task is cooperative game modeling. The core is to properly match the player and the utility to the elements in the task. Figure 2 summarizes existing combinations of the player and utility in DA tasks and their associated SV application purposes.

From the figure, we can see that the player is taxonomized into four types: (1) the data *feature*, (2) the data *tuple* (or called sample)

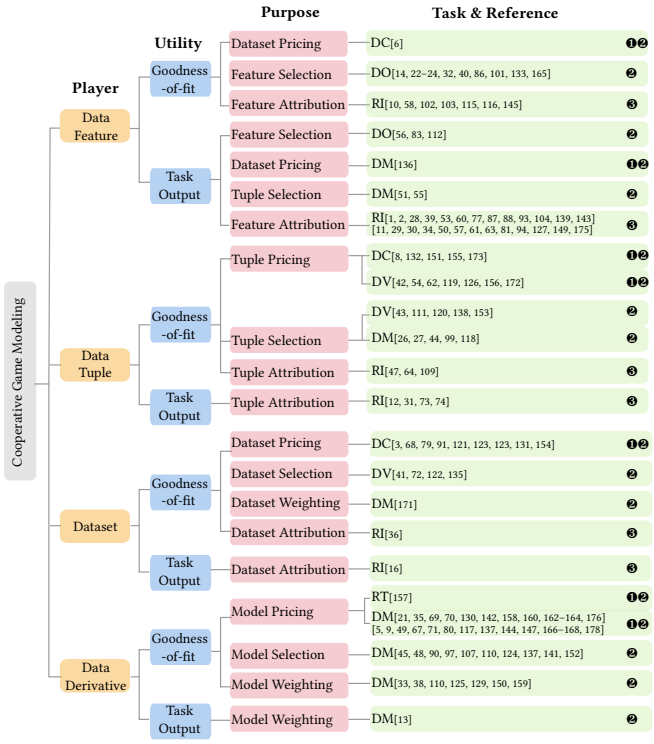


Figure 2: Cooperative game modeling for applying SV in the DA domain (❶, ❷, ❸ refer to the three functionalities summarized in Finding 3).

composed of several features and label(s), (3) the *dataset* composed of several data tuples, or (4) the data *derivative*, e.g., the ML model trained on several datasets.

Finding 1. Players may differ from the analytical object. While players in SV typically align with the analytical objects of a DA task, they can also represent sub-components of the analytical object for specialized objectives.

Discussion. Figure 1 demonstrates how the players align with the analytical objects of each task: in DO, the players are pixels (i.e., data features); in DV, they are images (i.e., data tuples); in DC, they correspond to image sets (namely datasets); in RT, players are ML models (data derivatives). Such alignments ensure that SV quantifies contributions at the granularity of the task's core analytical object.

In contrast, Wang et al. [136] defined the players as the sub-components of the analytical object in a multi-group data valuation task. The task is to evaluate contributions to Cervical cancer prediction across five disjoint groups, where each group's dataset contains three unique data features (e.g., age) with no feature overlap between groups. Here, the player is the data feature, a sub-component of the core analytical object (each group's dataset). The flexibility in defining players as either the analytical object or its sub-components underscores the adaptability to DA task-specific interpretability needs.

The utility function has two kinds of outputs: (1) the *goodness-of-fit score* between the DA task's outputs and ground truth facts, including but not limited to test accuracy, training loss, confidence

score, etc., (2) the *task outputs* themselves, more specifically, predictive outputs from the ML models learned in DA tasks or the answers to queries/questions on the data used in the DA workflow.

Finding 2. Utility definitions are task-dependent. The choice of utility is dictated by the objectives of the DA task. When maximizing or minimizing goodness-of-fit scores (e.g., accuracy), the utility directly reflects those scores. Conversely, when explaining task outputs or quantifying data impacts on those outputs, the utility is defined by the outputs themselves.

Discussion. Figure 1 exemplifies this duality. In the DV task, utility is tied to test accuracy to optimize a medical image classifier, while in the RI task, utility derives from predictive outputs to explain how pixel-level variations influence results. Notably, a single player-utility pairing, such as defining players as ML models and utility as test accuracy, can serve multiple DA tasks (e.g., DM and RT in the figure). Similarly, one task may leverage SV for diverse purposes: the DM task, for instance, can use SV both to weight local models and to assign pricing incentives, ensuring higher model quality while aligning contributions with economic rewards. This adaptability reveals SV's capacity to address heterogeneous analytical goals through context-aware utility design.

Finding 3. The real-world functionality of applying SV lies in three aspects: (1) to construct fair marketplaces for data and related products, incentivizing data sharing; (2) to improve the quality of DA outputs while reducing the economic cost; (3) to transfer DA outputs into actions to solve real-world problems.

Discussion. Figure 1 exemplifies the three functionalities that different DA tasks target to achieve. DC and RT tasks seek to provide helpful results to construct fair marketplaces for data in the medical domain. DV and DM tasks strive to improve the test accuracy of a medical image classification model while reducing the learning cost. The RI task intends to transfer predictions from a medical image classification model into actions assisting diagnosis. As summarized in Figure 2, to serve the first functionality, the purpose of applying SV is typically pricing. For the second functionality, the purposes of applying SV include pricing, selection, and weighting. To fulfill the last functionality, the purpose of applying SV is primarily attribution. Demanders can design SV applications that satisfy their specific requirements according to Figure 2 and Findings 1-3.

3.2 Challenges and Solutions of Applying SV

In this section, we taxonomize existing studies on applying SV in the DA domain according to the challenges they tackle, namely, *computation efficiency*, *approximation error*, *privacy preservation*, and *interpretability*. Subsequently, we conduct a thorough analysis and discussion of the countermeasures, highlighting key takeaways and remaining questions that we will study further through experiments in §5.1–§5.4. For more details of the techniques summarized in §3.2.1–§3.2.4, please refer to our technical report [66].

3.2.1 Computation Efficiency. Based on Definition 2 and Equation 1, the total cost of SV computation is determined by $N_{uc} \times T_{uc}$, where N_{uc} is the total number of utility computations, and T_{uc} is the average time cost of every utility computation. To compute exact SV, the algorithm generally enumerates all possible coalitions of players in the cooperative game by *iterations*, with each iteration computing the utility on one coalition.

Exact SV computing algorithm is known costly [12, 17, 108], with the number of iterations reaching an exponential level, i.e., $N_{uc} = 2^n$. Thus, the computation complexity of exact SV is $O(2^n)$. In DA tasks, the quantity of players (i.e.,n) is much higher than expected, resulting in the prohibitive computation cost. Moreover, as highlighted by Jethani et al. [53], the total cost of SV computing in DA can be further increased when DA tasks involve ML models, since each iteration typically necessitates retraining these models, associated with non-trivial expenses. For example, the DO task in Figure 1 analyzes 224×224 players (pixels), thus $N_{uc} = 2^{224\times224}$. Depending on the task's model size and computation resources, T_{uc} can reach hours to months [25], much longer than milliseconds in traditional game theory.

To efficiently compute SV, the research focus has shifted from the exact computation to the approximate algorithms. Abundant approximate SV computing algorithms have been proposed and most existing surveys [12, 17, 65, 94, 100, 108, 122, 128] classified the algorithms into two categories: task-agnostic, i.e., covering general usage for different DA tasks, and task-specific, i.e., relying on certain assumptions about DA tasks.

Different from previous surveys, our paper disentangles the vital techniques that can be used flexibly to develop new efficient SV computing algorithms from existing applications. Generally, these techniques fall into two categories: (1) iteration reduction for reducing N_{uc}, and (2) ML speedup for reducing T_{uc}.

Table 2 summarizes the iteration reduction techniques disentangled from task-agnostic algorithms, which can be outlined into three types: (1) sampling-based, to estimate SV using randomly-sampled coalitions or permutations, involving Monte Carlo (MC), regression (RE), multilinear extension (MLE), group testing (GT), compressive permutation sampling (CP), (2) truncation (TC), to avoid computing the marginal contribution of new players who join a coalition unnecessarily, (3) fitting-based, to train a predictive ML model whose outputs are the estimations of the targeted SVs, including SV predictor learning (PL).

Finding 4. For task-agnostic iteration reduction, sampling-based techniques gain the leading place in practical applications. Each sampling-based technique can be deployed independently as a base SV computing algorithm.

Discussion. Among the sampling-based iteration techniques, MC gains the widest application. Theoretically, GT and CP can further reduce the complexity of SV computation, but these two techniques do not gain much wider application than MC. We speculate that GT is limited by its assumption on the correctness of mirroring the difference between the SVs of any two players, while CP is limited by its strong reliance on the sparse SV assumption. The other sampling-based iteration techniques, RE and MLE, are also relatively limited in practical applications compared with MC. These two techniques are heuristic and the corresponding big-O notations are not given by existing arts.

Compared with sampling-based techniques, fitting-based techniques can generate SV at a much lower complexity. However, these techniques work effectively only with access to abundant ground-truth SVs (or high-quality surrogates of the ground-truth) for training the SV predictor, and the training incurs extra cost.

Tech.	Main Idea	Complexity	Extra Cost	Use Scenarios
MC	sampling	$O(n^2 \log(n))$	✖	DC[3, 68, 91, 121, 123, 154, 173], DO[24, 40, 83, 112], DV[42, 43, 54, 62], EL[13, 107], FL[21, 35, 69, 90, 136, 141], RI[88]
RE		/	✖	RI[28, 53, 76]
MLE		Polynomial	✖	EL[107], RI[88, 92]
GT		$O(\sqrt{n}\log(n)^2)$	✖	DV[54], FL[141]
CP		$O(n \log\log(n))$	✖	DV[54]
TC	truncation	/	✖	DC[8, 79, 131], DV[79, 99, 153], FL[44, 70, 70, 71, 71, 159, 160, 160]
PL	fitting	$O(1)$	✔	RI[53]

Table 2: Summary of iteration reduction techniques disentangled from *task-agnostic* SV computing algorithms. Based on learning paradigms, our work further classifies the DM tasks having applied SV into five types: semi-supervised learning (SSL), active learning (AL), continuous learning (CL), ensemble learning (EL), and federated learning (FL).

Finding 5. The truncation-based technique tends to be employed in conjunction with a sampling-based technique.

Discussion. Many works [8, 24, 43, 70, 79, 99, 131, 153, 160, 173] have attempted the combination of *MC+TC* to develop hybrid[1] SV computing algorithms. As a flexible technique, TC is also compatible with other sampling-based iteration reduction techniques, e.g., RE, MLE, GT, and CP. However, these combinations have not been evaluated yet, since prior works analyze SV applications or computing algorithms as monolithic units rather than decomposing them into reusable building blocks. Later in §5.1, we will evaluate the performance of TC integrated with different sampling-based techniques to validate its flexibility.

Table 3 summarizes the iteration reduction techniques disentangled from task-specific algorithms, including linear-based, tree-based, K-nearest-neighbor(KNN)-based, deep-neural-network(DNN)-based, uniform division, and influence function.

Finding 6. For task-specific iteration reduction, leveraging simple-structured ML models or loss-bounded learning algorithms is the key to lowering the complexity of SV computation. However, with the rising popularity of large-scale pre-trained models in the DA domain, the usability of this solution direction might be compromised.

Discussion. Although some task-specific techniques are designed based on DNN, the complexity of those techniques is determined by the number of hidden layers in the model and the number of parameters in each layer. Therefore, if a DA task relies on large-scale pre-trained models (such as GPT-4 [96]), those techniques may not work efficiently to generate accurate SV approximation results in this case.

Table 4 summarizes the ML speedup techniques: gradient approximation (GA), test sample skip (TSS), and model appraiser (MA). All these techniques, disentangled from task-specific algorithms, are heuristic (thus not given with big-O notations). GA expedites the

[1]In this paper, we call the SV computing algorithm adopting different techniques for efficient computation as the *hybrid* algorithm.

Tech.	Main Idea	Complexity	Use Scenario
Linear-based	model structure	$O(n)$	RI[76]
Tree-based		$O(TLD^2)^*$	RI[75, 89, 161, 169]
KNN-based		$O(nlog(n))$	DV[139], SSL[27], AL[44], CL[118]
DNN-based		/	RI[4, 140]
Uniform division	stable learning	$O(1)$	DV[54]
Influence function	smooth utility	$O(n)$	DV[54]

* T: number of trees; L: maximum number of leaves in a tree; D: maximum tree depth.

Table 3: Summary of iteration reduction techniques disentangled from *task-specific* SV computing algorithms.

Tech.	Objective	Main Idea	Use Scenarios
GA	training speedup	replace gradients from costly multi-step computations with easy-to-obtain surrogates	DV[54], AL[44], FL[125, 130, 158]
TSS	inference speedup	evaluate on only ambiguous test data whose predictive results vary across models	FL[176]
MA		train a model whose outputs are estimations of the given model's performance score	DC[132], DV [42], RT[157]

Table 4: Summary of ML speedup techniques.

model training needed at each time of utility computation, while TSS and MA accelerate model inference processes.

Finding 7. ML speedup techniques are compatible with iteration reduction techniques.

Discussion. There are many works on hybrid SV computing algorithms integrating these two types of techniques. An example is *MC+GA*, utilized by both DV tasks selecting high-quality data tuples from the UK Biobank dataset to train logistic regression [43] and FL tasks selecting high-quality local models to generate the global model for image classification [9, 70, 160]. *MC + GA + TSS* is another typical combination designed for tasks with the players being ML models [176]. We note that integrating ML speedup with iteration reduction is well-suited for tasks, like DV or FL, which have (one of) the following characteristics. The first characteristic is that computing a utility involves the costly multi-step gradient descent for model training. Another characteristic is that the utility computation relies on numerous test data samples or complicated models, e.g., pre-trained models with billions of parameters.

Though heaps of hybrid algorithms have been proposed, providing empirical usage of SV, a question remains open for rigorous demanders: **Can the hybrid SV computing algorithms always ensure higher efficiency than the algorithm using only one of the integrated techniques?** Concluding this subsection with the question, we attempt to answer this question later in §5.1.

3.2.2 Approximation Error. SV approximate computation, though faster than exact computation, introduces the variance caused by the randomness in sampling the player coalitions and the bias caused by incomplete exploration of all the possible coalitions. Hence, it poses a new challenge that the approximate SV may not

Tech.	Main Idea	Compatible With	Use Scenarios
stratified	sample permutations or coalitions from disjoint strata proportionally	MC, RE, MLE, GT, CP	DC[151], DO[57], DV[54, 146, 172], FL[67, 97], RI[57, 58, 88]
antithetic	sample negatively correlated permutations or coalitions		FL[97],RI[28, 88, 92]
kernel-based	sample permutations with good distributions relative to kernels	MC, CP	RI[88]

Table 5: Summary of variance reduction techniques.

Tech.	Objective	Lightweight	Rigorous	Use Scenarios
NPM	exposure elimination	✔	✖	RI [15]
HE		✖	✔	FL[176]
SMPC		✖	✔	DC[132]
QT	inference prevention	✔	✖	RI[77]
DR		✔	✖	RI[77]
DP	both	✔	✔	DV[139, 146], RI[15, 77], RT[68]

Table 6: Summary of privacy protection techniques.

be an accurate and unbiased estimation of the exact SV, and thus may fail to serve its expected application purposes. The key to reducing the SV approximation error is variance reduction. Table 5 summarizes the techniques falling into three types: stratified, antithetic, and kernel-based.

Finding 8. Stratified and antithetic techniques are compatible with sampling-based iteration reduction techniques, regardless of the sampling objects being coalitions or permutations. The two techniques can be integrated seamlessly with MC, RE, MLE, GT, or CP. However, the kernel-based technique is designed exclusively for permutation sampling, e.g., in algorithms based on MC or CP.

Discussion. The stratified technique is applicable regardless of the sampling objects, because the coalitions can be divided into disjoint strata based on the number of players included in each coalition, and the permutations can be stratified according to the position of a player. The antithetic technique also works for sampling the two objects, since each coalition $\mathcal{S}$ has a negatively correlated counterpart $\mathcal{N}\backslash\mathcal{S}$, and any two permutations are negatively correlated if the order of players is completely reversed. Existing kernel-based technique, in contrast, only defines the distance and similarity between permutations, and thus cannot work when the sampling object is a coalition.

Finding 9. Reducing the approximation error of SV might compromise its computation efficiency.

Discussion. The mainstream algorithms for computing approximate SV rely on sampling. However, according to the law of large numbers [148], no matter which sampling strategy is adopted, it is inevitable to sample more coalitions and compute their utilities to reduce approximation error, which, on the other hand, increases computation complexity. This leaves a riddle: **Given a sampling strategy, how to strike a balance between SV approximation error and computation efficiency?** Similarly, we conduct certain experiments in §5.2 to bridge this gap.

3.2.3 Privacy Preservation. Applying SV in DA can raise privacy concerns when the data in analysis contain sensitive or personal information [139]. The privacy issues come from two aspects: (1) computing SV in DA, especially in distributed tasks like FL, requires exposure of data or data derivatives such as ML models; and (2) attackers can leverage SV to infer private and sensitive information about individuals in the dataset when the SV is reported by private data owners themselves or a cloud service. The current research has explored the potential of SV for feature inference attacks (FIA) [77], to reconstruct private data by deducing the features of those data using the feature's SV, and membership inference attacks (MIA) [139], to detect the presence or absence of data samples in a private dataset using the sample's SV.

Studies of SV-driven privacy issues are aimed at two kinds of objectives: (1) exposure elimination, which safeguards raw data and derivatives during SV computation, and (2) inference prevention, which handles privacy inference attacks. As cataloged in Table 6, countermeasures include non-perturbation masking (NPM), homomorphic encryption (HE), and secure multiparty computation (SMPC) for the first objective, quantization (QT) and dimensionality reduction (DR) for the second objective, and differential privacy (DP) which serves both.

Finding 10. Lightweight privacy protection techniques, including NPM, QT, DR, and DP, balance efficiency and protection, while rigorous techniques, containing HE and SMPC, prioritize security at a higher computation cost. To achieve both privacy preservation and secure computation, hybrid schemes that combine HE or SMPC with QT, DR, or DP are recommended.

Discussion. Lightweight and rigorous theoretical privacy guarantees are two primary factors affecting the selection of privacy protection measures for SV. For applications related to cross-institutional highly-sensitive data (e.g., cross-hospital medical images in Figure 1), rigorous techniques like HE or SMPC are needed. While if computing resources are constrained (e.g., in applications deployed across edge devices), the lightweight NPM, QT, or DR techniques are preferred. We note that DP, possessing the lightweight property and rigorous theoretical guarantees simultaneously, is applicable to both cases.

Finding 11. Adopting privacy-preserving measures may compromise SV's computation efficiency and effectiveness.

Discussion. The impacts on the efficiency can originate from three factors: (1) the noise (introduced by measures such as DP and NPM) which may slow down computation convergence, (2) the extra time cost needed for encrypting and decrypting data (introduced by measures such as HE), and (3) the extra time cost for multiparty interactions (needed by measures such as SMCP). Several studies [68, 77, 132, 139, 146, 176] have attempted to achieve a compromise between SV's computation efficiency and privacy preservation. Their key idea is to combine privacy-preserving measures with hybrid SV computing algorithms which improve efficiency by integrating techniques summarized in §3.2.1. For example, Tian et al. [132] adopted SMCP on top of a hybrid scheme that computes SV using both MC and MA techniques, ensuring the efficiency of SV computation on data held by different data owners while eliminating the need to expose data before buyers pay for them.

Tech.	Interpretations
Utility-based	Altering the value of data features with high SV tends to incur more changes in model predictions [59–61]. High-valued data tuples result in more increase in test accuracy [43, 54, 119, 120].
Characteristic-based	Data tuples with low SV are inclined to be noisy data with false labels [54], outliers [43], or corruptions [43]. ML models with high SV are those that have high test accuracy and certainty [107, 157].
Counterfactual	Given a dataset, there exists a minimum set of tuples in this dataset such that transferring the found set from this dataset to another dataset can flip the direction of the inequality between the SVs of those two datasets [121].

Table 7: Summary of SV interpretations.

Privacy protection can also influence the effectiveness of SV. The mainstream arts rely on the scaling [33, 38, 110, 125, 129, 150, 159] or ranking [43, 54, 62, 86, 132, 177] of final SV results to perform pricing, selection, weighting, and attribution in DA. However, the scaled SV results or the ranking results can be easily altered [15] when integrated with techniques like DP, QT, and DR. We notice a scarcity of quantitative results for answering the question: **Can a balance be achieved between the effectiveness of privacy protection with the effectiveness of SV?** Therefore, we offer insights into this question with the experimental study in §5.3.

Overall, we advocate a deeper cost-benefit study analyzing the marginal gains in privacy protection against the computation cost and effectiveness loss of SV, both qualitatively and quantitatively. Besides, a dynamic adjustment on the efficiency, privacy, and effectiveness of SV according to DA task settings, system constraints, user requirements, etc., merits further investigation.

3.2.4 Interpretability. In addition to the aforementioned challenges, applying SV to DA also faces the trouble of how to properly translate the obtained SVs to exact actions (e.g., adding or deleting a data sample, normalizing data features in some dimensions, etc.) in the DA workflow [61].

For understandable SV interpretations, researchers have relied on three paradigms: (1) utility-based, pointing out that players with high SV are those who have more impact on the overall utility of the targeted DA tasks; (2) characteristic-based, seeking to reveal the relationship between the intrinsic characteristics of each player and its SV; (3) counterfactual, aiming to find out the minimum change between two players that can flip the direction of the inequality between the SVs of those players. Table 7 summarizes the interpretations produced by three paradigms.

Finding 12. The utility-based interpretation paradigm is the most universal for interpreting SV in DA. In contrast, the characteristic-based paradigm needs sufficient expert knowledge to elaborate intrinsic characteristics of data, and the counterfactual explanation needs costly computations.

Discussion. With requirements on expert knowledge or extra computations, the characteristic-based paradigm and the counterfactual explanation did not gain a wider application than the utility-based paradigm. Despite the popularity of utility-based interpretations, many works [59, 61, 81, 174] have claimed that the player with a larger SV may have less influence on the overall utility of the associated DA task, which contradicts the mainstream interpretations. However, these works studied only the cases defining data features as the player and thus cannot fully answer the following questions: **Can the SVs of the four types of players in DA be correctly interpreted by the mainstream paradigm? If cannot, why? Is there any general reason applicable to all four types of players?** We endeavor to answer these questions through a comprehensive evaluation in §5.4.

3.2.5 Summary of Findings. Firstly, our findings, summarized from a wide range of the arts, provide *general and comprehensive* guidelines for academicians and engineers to study and develop SV applications when confronted with DA tasks. Secondly, our findings analyze the arts in this field through *a finer-grained perspective*. We analyze the pros and cons of the resolution techniques disentangled from complete algorithms in addressing four challenges of using SV in DA (Findings 4, 6, 10, 12), propose the possible combinations of the techniques for developing new algorithms (Findings 5, 7, 8), and discuss the potential conflicts between different challenges, e.g., computation efficiency vs. approximation error (Finding 9) or privacy preservation (Finding 11). Finally, as a consequence of the finer-grained understanding of SV, our findings can instruct the design of a *modular and extensible* framework for developing SV applications, presented in the next section.

4 *SVBENCH*

In this section, we propose *SVBench*, a modular and extensible framework for developing SV applications for DA tasks, aiming to bridge the gap in a unified library supporting flexible integrations of state-of-the-art countermeasures to different challenges of SV.

Overview. As shown in Figure 3, *SVBench* consists of a configuration loader, a sampler, a utility calculator, a convergence checker, and an output aggregator. The configuration loader loads the SV computing parameters specified by the users. The sampler generates the coalitions or permutations of players based on the configured sampling strategy. The utility calculator takes the sampled coalitions or permutations as the input to a utility function and drives the computation. When users specify an efficiency optimization strategy, the utility calculator will use that strategy to accelerate the computation. The convergence checker determines whether to terminate the SV computation based on the convergence criterion specified in the configuration. An *iteration* of SV calculation is conducted starting from the sampler and ending at the convergence checker. Once the convergence criterion is not met, another iteration will be initiated as demonstrated in the figure (with dashed arrow). The output aggregator generates the final SV of each player. If users specify privacy protection measures, the aggregator will execute those measures before reporting the final results.

Usage Instructions. Users of *SVBench*, including seasoned engineers and researchers seeking updates on the building blocks of SV applications and newcomers to this field, need to initialize the framework in three steps. Firstly, *cooperative game modeling*, in which the users properly define the players and the utility function of their targeted DA task. *SVBench* suggests users perform this step according to Figure 2 and also allows new definitions of player and utility not included in this figure. Secondly, *SV computation*,

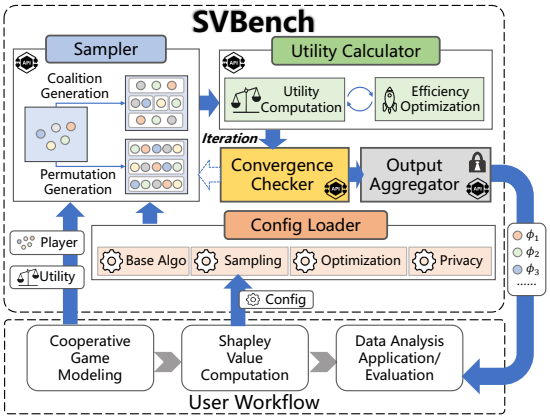


Figure 3: The *SVBench* overview.

Config	Parameters (Default setting is in italics.)
Base Algo	*MC*, RE, MLE, GT, CP, user-specific
Sampling	None, *random*, stratified, antithetic, user-specific
Optimization	*None*, TC, GA, TC+GA, GA+TSS, TC+GA+TSS, user-specific
Privacy	*None*, DP, QT, DR, user-specific

Table 8: Configuration parameters used by *SVBench*.

in which users specify the configuration parameters listed in Table 8. The default settings are marked in the table. Thirdly, *data analysis application/evaluation*, in which users utilize SV results to perform pricing, selection, weighting, or attribution on data or data derivatives according to the specific DA task requirements.

Use Case. Take the DV task in Figure 1 as an example. Suppose the user plans to use *SVBench* to implement a hybrid SV algorithm, which combines MC and TC techniques with DP for privacy protection, to generate SVs of 6000 images from a 'MedicalImg' dataset in DV. The user defines 6000 images as the players and a 'DV-MI' function, which takes images as inputs to train a classification model and outputs the test accuracy of this model, as the utility function. Next, the user configures *SVBench* with the following parameters: task='DV', dataset='medicalImg', player='tuple', utility_function='DV-MI', base_algo='MC', optimization_strategy='TC', privacy_protection_measure='DP', then invokes the SV computing function in *SVBench* to obtain SVs. After that, the user selects the images assigned with the top 50% SV results to learn the classification model. More use cases can be found at GitHub [2].

Extension Supports. Besides implementing the mainstream techniques for computing SV, we provide several APIs in the modules of *SVBench* (marked by an API icon in Figure 3) for users to extend this framework. One can configure the module he expects to extend by a user-specific parameter at the SV computation step and submit the new functions corresponding to that module. For example, in the above use case, the user submits a user-specific utility function namely DV-MI and configures the utility calculator module by setting utility_function='DV-MI'. Similarly, the user can also configure a user-specific aggregator module by submitting a new privacy protection function (e.g., namely newMaskSV, which takes original SV computing results as inputs and outputs masked SV results) and setting privacy_protection_measure='newMaskSV'. *SVBench* will check the legitimacy and validity of the received functions and use the valid functions to execute the operations in the corresponding modules. Moreover, with the user permission, *SVBench* will embed the valid new functions into their corresponding modules to provide more development choices for future use.

Configuration of *SVBench*				Section Index
Base Algo	Sampling	Optimization	Privacy	
MC / RE$^\spadesuit$ / GT$^\spadesuit$/ MLE$^\spadesuit$ / CP$^\spadesuit$	random	None / TC / GA / TC+GA / GA+TSS /TC+GA+TSS	None	§5.1
MLE	random/ stratified/ antithetic	None	None	§5.2
	random	None	DP / QT / DR	§5.3
	random	None	None	§5.4

Table 9: The configuration of *SVBench* for implementing different SV computing algorithms in §5.1–§5.4.

Summary. The application of SV in DA requires to be flexible in engineering implementation, yet most existing works overlook such demands. The application should feature *a modular architecture with configurable parameters*, empowering engineers to tailor the usage of SV according to task-specific requirements. Moreover, the application should possess the ability to be *parallelized and disaggregated*. In this manner, it can leverage diverse computing architectures, including multi-core CPUs, GPUs, and distributed computing paradigms, to enhance scalability and efficiency. With these considerations, we propose *SVBench*. While *SVBench* has demonstrated usability, modularity, and flexibility through the success of implementing dozens of algorithms in the next section, we anticipate future extensions to further enhance its potential to develop efficient, secure, and effective SV applications in DA.

5 EVALUATION

In this section, we use *SVBench* to implement multiple SV computing algorithms, including both the base algorithms that have been studied by prior works and the hybrid algorithms with novel combinations of SV computing techniques. Using the implemented algorithms, we not only validate the usability, modularity, and flexibility of *SVBench* but also study the following four sets of evaluations in order to answer the aforementioned questions:

- §5.1 compares the efficiency of five base SV computing algorithms with several hybrid algorithms, answering the question highlighted at the end of §3.2.1.
- §5.2 investigates the relationship among the computation efficiency, approximation error, and the effectiveness of SV, solving the problem proposed at the end of §3.2.2.
- §5.3 examines the effectiveness of existing measures for preventing SV-driven attacks and the impacts of the measures on the effectiveness of SV, tackling the problem left in §3.2.3.
- §5.4 explores the relationship between the SVs of the four types of players in DA and the overall utility of their associated tasks, offering insights to the question bold in §3.2.4.

Table 9 summarizes the detailed configurations specified for each algorithm. We note that this work is the first attempt to combine

the ♠-tagged base algorithm with the five optimization techniques. We use MLE as the base algorithm in §5.2–§5.4, since it generally achieves better efficiency and accuracy performance than the other base algorithms in §5.1 and varying base algorithms would not influence conclusions in those subsections.

For the generality of findings from experiments, we select six canonical datasets, Adult [84], Tic-Tac-Toe (Ttt) [84], Bank [84], Dota2 [84], Wind [37, 134], 2Dplanes [37, 134], from a large number of literature to conduct four types of DA tasks – result interpretation (RI), data tuple valuation (DV), dataset valuation (DSV), and federated learning (FL), which enable evaluations of SV for different types of players defined in the current DA domain. More details of task settings and the code are available at GitHub [2].

5.1 Computation Efficiency

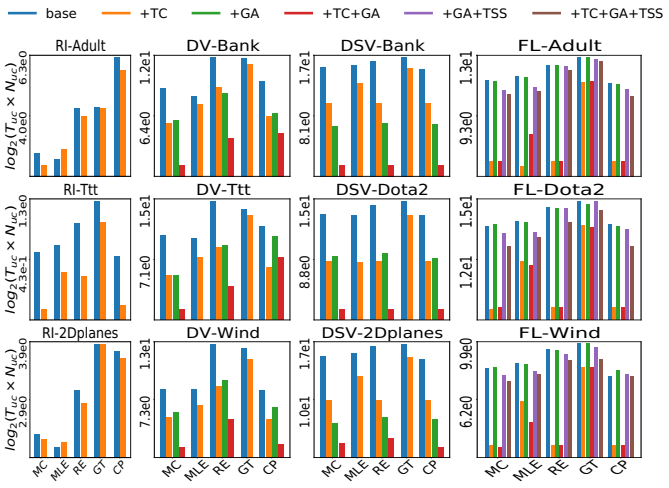
This section investigates the efficiency performance of five base SV computing algorithms (MC, RE, MLE, GT, CP) and the hybrid algorithms, each of which selects an optimization strategy from Table 8 and integrates this strategy with a base algorithm. To control the time cost of each experiment, we follow previous work [70] to monitor the approximation stability by $\Delta\hat{\phi} = \frac{1}{5n}\sum_{m=1}^{5}\sum_{i=1}^{n}|\frac{\hat{\phi}_i^{e}-\hat{\phi}_i^{e-m\times n}}{\hat{\phi}_i^{e}}|$ and terminate the approximation when the convergence criterion ($\Delta\hat{\phi} < \tau$) is satisfied, where $\hat{\phi}_i^{e}$ is the approximate SV of player p_i after e times of utility computation, τ is the convergence threshold set to 0.05 for all tasks. We measure the efficiency performance by the total time cost $N_{uc} \times T_{uc}$, and show the computation complexity N_{uc} and the approximation error $\epsilon = 1 - \frac{\sum_{i=1}^{n}\hat{\phi}_i\cdot\phi_i}{\sqrt{\sum_{i=1}^{n}\hat{\phi}_i^2}\cdot\sqrt{\sum_{i=1}^{n}\phi_i^2}}$ [70], as a reference in the results.

Figure 4 presents the efficiency results. The number of bars differs across tasks due to (1) no applicable GA/TSS techniques for RI tasks and no TSS technique applicable for DV tasks; (2) MLE/GT incompatible with the GA technique devised for DV tasks.

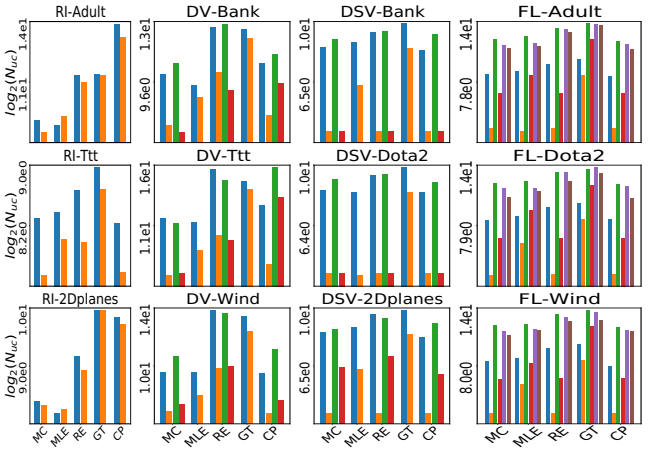
As denoted by blue bars, MLE generally achieves the leading efficiency performance (lower $T_{uc} \times N_{uc}$ and N_{uc}) among the five base algorithms. The major reason is that, unlike MLE, algorithms such as RE, GT, and CP must solve auxiliary optimization problems (e.g., weighted least squares in RE, feasibility constraints in GT, or convex objectives in CP) in the computation, introducing an extra cost. In particular, when it is hard to solve the corresponding optimization problem for the DA task (e.g., RI-Adult, RI-2Dplanes, and DV-Ttt), prolonged convergence will appear. Although CP can converge faster than MLE in some tasks, e.g., RI-Ttt, DV-Wind, FL-Wind, it has a strong dependency on the sparse SV assumption and performs well only when that assumption is satisfied (to some extent). We also notice that MC can outperform MLE occasionally (e.g., in RI-Ttt, DSV-2Dplanes, FL-Dota2). However, MLE generally achieves higher accuracy (lower ϵ) than MC under identical convergence criteria, due to its closed-form integral expression of SV (see our technical report [66] for details), enabling more deterministic computation with high precision (e.g., via Gaussian quadrature).

By comparing bars of different colors, we note that integrating TC reduces the SV computation complexity (N_{uc}) and thus reduces the total time cost ($N_{uc} \times T_{uc}$), in most cases. TC can reduce N_{uc}

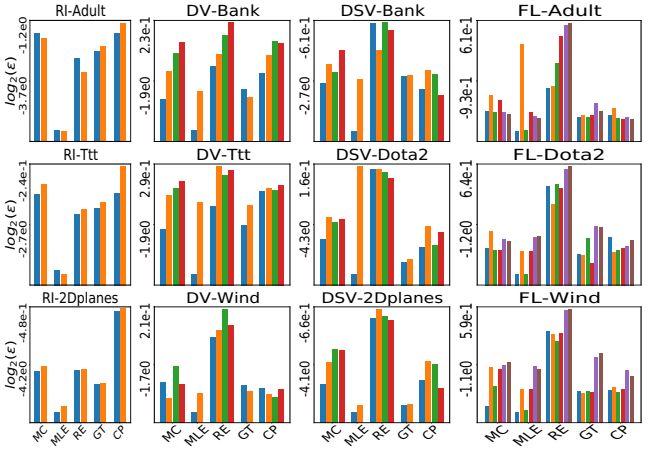
[2] https://github.com/zjuDBSystems/SVBench.



(a) The total time cost, $T_{uc} \times N_{uc}$.



(b) The computation complexity, N_{uc}.



(c) The approximation error measured by ϵ.

Figure 4: Efficiency performance (x-axis: the base SV computing algorithm). The smaller the three metrics, the more efficient and accurate the computation.

because it avoids computing the marginal contribution of new players who join a coalition unnecessarily, particularly when that coalition's utility $U(\mathcal{S})$ is close to the overall utility achieved by the grand coalition $U(\mathcal{N})$, for example $U(\mathcal{S}) > 90\%U(\mathcal{N})$. However, if few coalitions meet such a condition, TC would pose trivial positive impacts on SV computation efficiency. To intensify TC's effects in reducing cost, setting a looser condition, e.g., $U(\mathcal{S}) > 80\%U(\mathcal{N})$, is a potential solution, but this may enlarge the approximation error. Therefore, a thorough tuning of the truncation condition is necessitated when adopting TC.

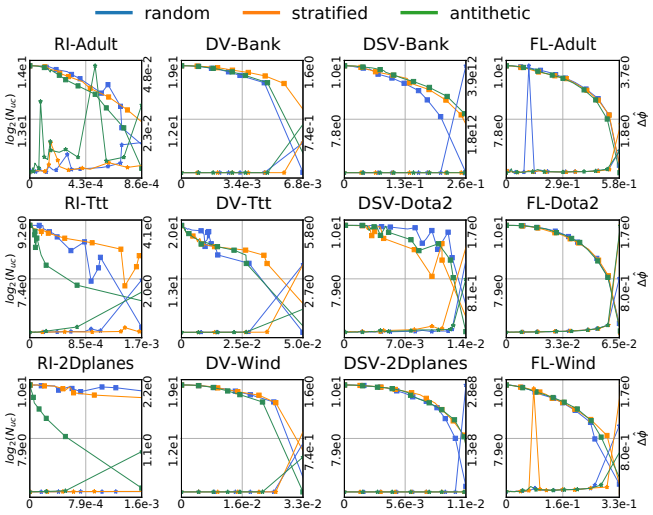
By comparing Figure 4(a) and Figure 4(b) vertically, we can tell that GA and TSS techniques, though they may contribute negatively to reducing N_{uc}, are very helpful in reducing the average time cost of utility computation (T_{uc}). GA and TSS can reduce T_{uc} because they use fewer training batches and fewer test data samples in marginal contribution estimation, respectively. However, these operations may enlarge the variance of approximate SVs in different iterations of utility computations, leading to an increasing number of utility computations (N_{uc}) for convergence. When using GA or TSS together with TC, their negative impacts on computation complexity can be mitigated.

Research Direction 1: Exploration on innovative hybrid SV computing algorithms. Our evaluations show that the hybrid SV computing algorithms integrating multiple efficiency optimization techniques perform better than the algorithms using only one of the integrated techniques in most cases. We highly recommend *combining TC with a base SV computing algorithm* in all DA tasks and *activating GA and TSS* when the utility computation (e.g., in FL) needs costly model training or evaluation. For the base algorithm, it is recommended to choose MLE (or MC). In summary, innovative hybrid SV computing algorithms with extensive experiments are anticipated for efficient SV computation in the literature.

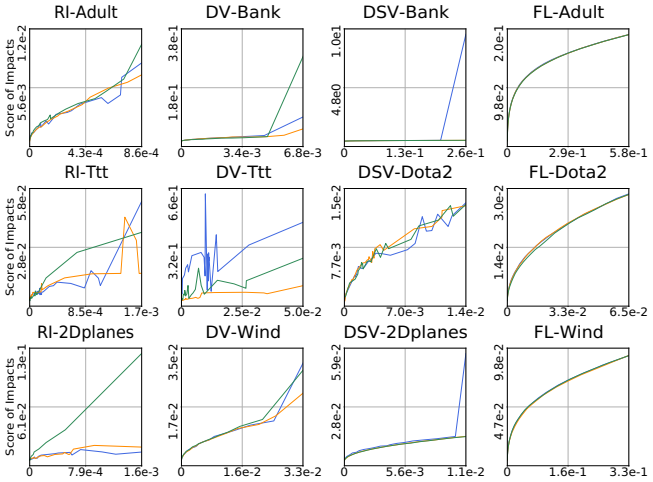
5.2 Approximation Error

This section explores the relationship among the approximation error, computation efficiency, and the effectiveness of SV in different DA tasks. We compare the performance of approximating SV using three widespread sampling techniques, *random sampling*, *stratified sampling*, and *antithetic sampling*. The computation complexity is still measured by N_{uc}. The impact of approximation error on the effectiveness of SV is quantified by the score $\sum_{i=1}^{n}(\frac{\hat{\phi}_i}{\sum_{i=1}^{n}\hat{\phi}_i} - \frac{\phi_i}{\sum_{i=1}^{n}\phi_i})$. We also report $\Delta\hat{\phi}$, the stability of approximate SVs defined in the previous section, as a reference.

Figure 5(a) presents the relationship between the approximation error of three SV computing algorithms and the computation efficiency and stability, while Figure 5(b) presents the impacts of approximation error on the effectiveness of SV. The square-tagged lines in Figure 5(a) show that, in most tasks, the larger the SV approximation error, the smaller the computation complexity. This is due to far fewer utility values needing to be computed when a larger tolerance is given to the approximation error, regardless of the strategy for sampling coalitions, as discussed in §3.2.2. However, as shown in the star-tagged lines in Figure 5(a), the approximate SVs with a large error tend to be generated when the computation is far from an ideal convergence status (where $\Delta\hat{\phi}$ goes below a trivial



(a) Left y-axis&square-tagged lines report N_{uc}. Right y-axis&star-tagged lines report $\Delta\hat{\phi}$. The lower the two values, the more efficient and stable the SV approximation.



(b) The smaller the score, the better effectiveness the approximate SV has.

Figure 5: Impacts of approximation error (x-axis: the ϵ value).

threshold value). Those SVs may lead to a larger error in the scaled SV results, as shown in Figure 5(b), indicating the potential ineffectiveness of using approximate SV for pricing, selection, weighting, and attribution in DA. All these results consolidate the necessity to strike a balance among the approximation error, computation complexity, and the effectiveness of SV. To achieve such a balance, we highlight the following future directions.

Research Direction 2: Investigation on the runtime dynamic tuning of convergence criterion when using sampling-based approximation techniques. Take the criterion $\Delta\hat{\phi} < \tau$ and the DV task in Figure 1 as an example. A small threshold (e.g.,$\tau = 10^{-5}$) can be set at the start of SV approximation. Then, the task can *periodically check* whether image samples ranked in the top 50% based on the latest approximate SV produce more accurate classification models than those produced by images ranked in the

bottom. Once satisfied, the approximation can be terminated to save computation cost.

Research Direction 3: Exploration on lightweight fitting-based approximation techniques. The fitting-based methods search for a mathematical relationship $\hat{\phi}_i = G(E(p_i))$, where $E(\cdot)$ encodes the player p_i into a characteristic vector, e.g., encoding the image in Figure 1 into a vector composed of the image's pixel values and label indexes, and $G(\cdot)$ generates the unbiased approximate SV, e.g., the SV of the image. Once $G(E(p_i))$ is determined in a DA task, the approximate SV can be generated by $O(1)$ complexity for new players in that task, significantly mitigating the conflicts between SV computation efficiency and approximation error. The major obstacle against the feasibility of learning $G(E(\cdot))$ at an affordable cost is *the collection of a sufficient amount of ground-truth SVs (or high-quality surrogates of the ground-truth)*. This problem is highly expected to be solved with the surge in real-world SV applications.

5.3 Privacy Protection

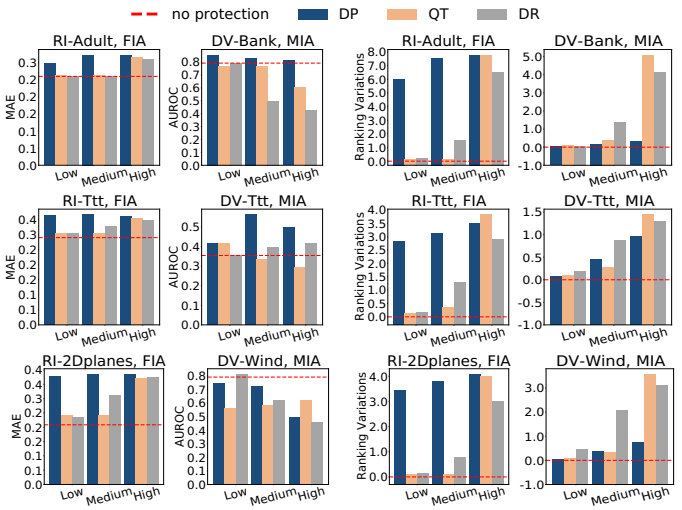
This section studies the effectiveness of three popular privacy protection techniques (*DP*, *QT*, *DR*) for preventing SV-driven attacks, including *FIA* [77], which may occur in RI tasks, and *MIA* [139], which may occur in DV tasks. We also study the impacts of the three techniques on SV's effectiveness.

The implementation of two attacks and three countermeasures is based on previous papers [77, 139]. We tune the strength of privacy protection from low to high by varying (1) the standard deviation of noise generated by DP from 0.1, 0.5 to 0.9, (2) the number of distinctive discrete Shapley values produced by QT from $0.9n$, $0.5n$ to $0.1n$, and (3) the number of $\vec{\phi}$'s dimension reduced by DR from $0.1n$, $0.5n$ to $0.9n$. We measure the performance in preventing SV-driven FIA by the MAE metric used by Luo et al. [77] and measure the effectiveness of preventing SV-driven MIA by the AUROC score used by Wang et al. [139]. As both RI and DV tasks rely on the ranking of final SVs to find the top important data features and samples, the impact of privacy protection techniques on SV effectiveness is measured by the variance in SV ranking results before and after protection.

Figure 6 shows the results of preventing the two attacks. In most cases, the stronger the strength of privacy protection techniques, the more effective those techniques are. However, stronger privacy protection renders a larger impact on SV ranking, affecting more on the effectiveness of SV for pricing, selection, weighting, or attribution. All these results consolidate the necessity of a trade-off between the prevention of SV-driven privacy leakage and the effectiveness of SV for decision-making in DA.

For preventing FIA, DP is the most effective, while QT and DR need tuning the privacy protection strength to a high level for an apparent MAE enlargement. Despite this, DP renders a much larger impact on SV ranking (thus affects more on SV's effectiveness) under the setting of low or median privacy protection strength. These phenomena are mainly because DP adds noise to SVs of all features no matter how the privacy protection strength is varied, while QT and DR alter SVs of only 10% (or 50%) data features when the strength is set to a low (or median) level.

For preventing MIA, QT and DR are effective in most cases, while DP leads to AUROC larger than the results achieved without



(a) The larger the MAE or the smaller the AUROC, the less privacy the SV exposes.

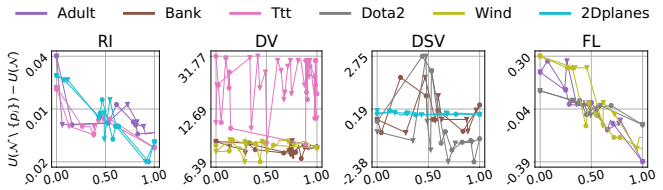
(b) The smaller the variance, the less the impact on SV effectiveness.

Figure 6: Preventing SV-driven FIA and MIA (x-axis: privacy protection strength).

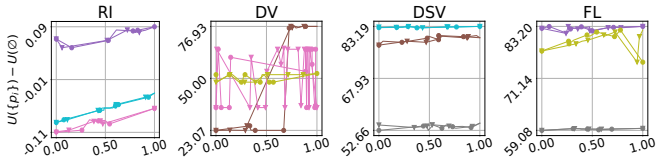
privacy protection in some cases, e.g., DV-Bank, DV-Ttt. Moreover, different from the results of preventing FIA, enhancing the privacy protection strength may not result in better effectiveness (lower AUROC) of the three techniques in preventing MIA. For example, in DV-Wind, QT performs the worst when setting a high privacy protection strength. These outcomes stem from the inability of QT, DR, and DP to rigorously enforce indistinguishability between two Shapley value distributions: the IN distribution, computed when the target sample is included in the dataset, and the OUT distribution, computed when the target sample is excluded [139]. Specifically, due to randomness, DP may inject markedly dissimilar noise into SVs used for generating IN and OUT distributions and thus enlarge the difference between the two distributions, making the success of MIA much easier. QT and DR map the original SVs into a new value space, in which the difference between SVs used for generating IN and OUT distributions might be more distinguishable, negatively affecting the efficacy in preventing SV-driven MIA.

Research Direction 4: Exploration on innovative techniques for tackling SV-driven privacy issues. For practical usage of existing privacy protection measures, we suggest *adjusting the strength parameter of the chosen measure*, such as the standard deviation for noise generation in DP, *to its median value* to achieve the privacy-effectiveness balance. Meanwhile, we strongly advocate for *innovative privacy protection measures* that prevent SV-driven attacks without compromising the computation efficiency and effectiveness of SV.

Research Direction 5: Exploration on new attack patterns. Plenty of SV-driven privacy issues remain unexplored. Except for FIA and MIA, malicious adversaries may utilize SV to launch other types of attacks, such as the *model extraction attack* [106], reconstructing a model learned in the DA workflow by creating a substitute model that behaves very similarly in SV-based evaluations



(a) The variation of the overall utility in different DA tasks caused by *removing* players.



(b) The variation of the overall utility in different DA tasks caused by *adding* players.

Figure 7: Removing or adding players (x-axis: SV of the removed/added player after min-max normalization). Circle-tagged lines are depicted based on exact SV, while triangle-tagged lines are based on approximate SV.

to the model under attack. Another example is the *data property inference attack* [106], extracting the properties implicitly encoded as features of the dataset.

5.4 SV Interpretations

This section examines the correctness of the mainstream SV interpretation paradigm, i.e., utility-based, in explaining SVs of different types of players defined in the current DA domain. This paradigm points out that the larger the SV of a player, the more the impacts that the player could pose on the DA task's overall utility. Therefore, we observe the change in the overall utility of DA tasks caused by removing or adding players.

Figure 7 presents our observed results. These results show that, in all four types of DA tasks, the impacts of removing or adding a player on the task overall utilities (i.e., $U(\mathcal{N} \setminus \{p_i\}) - U(\mathcal{N})$ or $U(\{p_i\}) - U(\emptyset)$) fluctuate as the SV of the removed or added player, no matter whether the SV is exact or approximate. The fluctuating results contradict the mainstream SV interpretations, probably because the latter rely mainly on the average marginal contribution while neglecting the variance of the marginal contributions of each player. Given a player in practical DA tasks, the variance of its marginal contributions might be much larger than its average contribution. For example, in DV-Wind, the variance of marginal contributions of a player ranges approximately from 11.83 to 18.00, which is much larger than the player's average marginal contribution, ranging from 0.06 to 1.43. Similarly, in DSV-2Dplanes, the variance ranges from 13.37 to 13.50, also larger than the average value in a range of $[0.13, 0.22]$. In both cases, we do not observe a strong correlation between SV and the player's contribution to a specific coalition (e.g., $\mathcal{N}$ in Figure 7(a) or $\emptyset$ in Figure 7(b)). All these results indicate the necessity to consider the variance of marginal contributions when using the mainstream utility-based SV interpretations to make decisions on pricing, selection, weighting, and attribution of data.

Research Direction 6: In-depth investigation on key factors affecting SV interpretations. SV may not be correctly interpreted or fail to serve its desired application purposes due to two points. One is *the difference between the exact SV and the approximate SV*. The other one is *the mismatch between the DA task and the implicit assumption of SV*. For example, DV-Wind or DSV-2Dplanes, as discussed above, mismatching with the assumption that the average contribution is appropriate to fairly distribute the overall utility of the task among players. Another example is the RI task in Figure 1, where pixels in an image interact mutually to influence predictions on the label of this image, mismatching with the assumption that the contribution of any individual player to the cooperative game is independent of its interactions with other players. Future works mining the deep understanding of these factors under more types of DA tasks to enhance SV's effectiveness are anticipated.

Research Direction 7: Exploration on supports for complicated DA tasks. The real-world DA tasks are complicated, containing *mutually dependent players* [1, 2, 22, 39, 93] and *real-time dynamic updates* on the player set [126, 155]. For example, the RI task in Figure 1, where the influence of one pixel on the prediction from the image classification model tends to interrelate with nearby pixels. Besides, the dynamic real-time updates of the player set [126, 155] are common in analytical tasks built upon the databases [98, 155, 173]. However, most existing SV applications are ill-suited for these tasks due to the assumptions that the players are independent and the player set is static and immutable. Although conditional SV [17] and dynamic SV [173] have been devised for complicated tasks, these new types of SV do not treat the mutual dependency among players and their real-time dynamic updates *as a single interrelated problem*, thus are limited in practical applications. It is expected that future work can devise more SV applications under the complicated yet realistic DA task settings.

6 CONCLUSION

This paper comprehensively studied the Shapley value applied throughout the data analytics workflow. We summarized the critical variables (i.e., the player and the utility function) in designing SV applications for DA and clarified the essential functionalities of SV for data scientists. We condensed the technical challenges of applying SV in DA and discussed the related arts, qualitatively and quantitatively. The conclusions of experimental evaluations based on our development framework, *SVBench*, support our findings in a synthetic review. At last, we identified the limitations of current efforts and offered insights into the directions of future work.

ACKNOWLEDGMENTS

This work was funded by National Key Research and Development Program of China (Grant No: 2022YFB2703100), the Pioneer R&D Program of Zhejiang (No. 2024C01021), and Zhejiang Province "Leading Talent of Technological Innovation Program" (No. 2023R5214). Meihui Zhang was funded by National Natural Science Foundation of China (U2441237) and the Open Research Fund of The State Key Laboratory of Blockchain and Data Security, Zhejiang University. We also appreciate anonymous reviewers for their valuable feedback and thank Jiaqi Chai for carefully proofreading the manuscript.

REFERENCES

[1] Kjersti Aas, Martin Jullum, and Anders Løland. 2021. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artif. Intell.* 298 (2021), 103502.

[2] Kjersti Aas, Thomas Nagler, Martin Jullum, and Anders Løland. 2021. Explaining predictive models using Shapley values and non-parametric vine copulas. *Dependence Modeling* 9, 1 (2021), 62–81.

[3] Anish Agarwal, Munther Dahleh, and Tuhin Sarkar. 2019. A marketplace for data: An algorithmic solution. In *EC*. 701–726.

[4] Marco Ancona, Cengiz Öztireli, and Markus H. Gross. 2019. Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Value Approximation. In *ICML*. PMLR, 272–281.

[5] Meriem Arbaoui, Mohamed-El-Amine Brahmia, Abdellatif Rahmoun, and Mourad Zghal. 2024. Optimizing Shapley Value for Client Valuation in Federated Learning through Enhanced GTG-Shapley. In *IWCMC*. 1528–1533.

[6] Santiago Andrés Azcoitia, Costas Iordanou, and Nikolaos Laoutaris. 2023. Understanding the Price of Data in Commercial Data Marketplaces. In *ICDE*. 3718–3728.

[7] Santiago Andrés Azcoitia and Nikolaos Laoutaris. 2022. A Survey of Data Marketplaces and Their Business Models. *SIGMOD* 51 (2022), 18 – 29.

[8] Santiago Andrés Azcoitia, Marius Paraschiv, and Nikolaos Laoutaris. 2022. Computing the relative value of spatio-temporal data in data marketplaces. In *SIGSPATIAL '22*. 1–11.

[9] Zahra Batool, Kaiwen Zhang, and Matthew Toews. 2022. FL-MAB: client selection and monetization for blockchain-based federated learning. In *SAC '22*. 299–307.

[10] Daniel Beechey, Thomas MS Smith, and Özgür Şimşek. 2023. Explaining reinforcement learning with shapley values. In *ICML*. 2003–2014.

[11] Joao Bento, Pedro Saleiro, André F. Cruz, Mario Figueiredo, and Pedro Bizarro. 2021. TimeSHAP: Explaining Recurrent Models through Sequence Perturbations. In *KDD '21*.

[12] Leopoldo E. Bertossi, Benny Kimelfeld, Ester Livshits, and Mikaël Monet. 2023. The Shapley Value in Database Management. *SIGMOD* 52 (2023), 6 – 17.

[13] Giovanna Bimonte, Maria Russolillo, Han Lin Shang, and Yang Yang. 2024. Mortality models ensemble via Shapley value. *Decis. Econ. Finance* (2024), 1–29.

[14] Joost Bosker, Marc Gürtler, and Marvin Zöllner. 2024. Machine learning-based variable selection for clustered credit risk modeling. *Journal of Business Economics* (2024), 1–36.

[15] Aso Bozorgpanah, Vicenç Torra, and Laya Aliahmadipour. 2022. Privacy and Explainability: The Effects of Data Protection on Shapley Values. *Technologies* 10, 6 (2022). https://doi.org/10.3390/technologies10060125

[16] Andreas Brandsæter and Ingrid K Glad. 2024. Shapley values for cluster importance: How clusters of the training data affect a prediction. *Data Mining and Knowledge Discovery* 38, 5 (2024), 2633–2664.

[17] Hugh Chen, Ian C. Covert, Scott M. Lundberg, and Su-In Lee. 2023. Algorithms to estimate Shapley value feature attributions. In *IJCAI*, Vol. 5. 590–601.

[18] Hugh Chen, Scott M Lundberg, and Su-In Lee. 2022. Explaining a series of models by propagating Shapley values. *Nat. Commun.* 13, 1 (2022), 4512.

[19] Lingjiao Chen, Paraschos Koutris, and Arun Kumar. 2019. Towards Model-based Pricing for Machine Learning in a Data Marketplace. In *SIGMOD '19*. 1535–1552.

[20] Jack C.P. Cheng, Qiqi Lu, and Yichuan Deng. 2016. Analytical review and evaluation of civil information modeling. *Autom. Constr.* 67 (2016), 31–47.

[21] Ziwen Cheng, Yi Liu, Chao Wu, Yongqi Pan, Liushun Zhao, and Cheng Zhu. 2024. PoShapley-BCFL: A Fair and Robust Decentralized Federated Learning Based on Blockchain and the Proof of Shapley-Value. In *NIPS*. 531–549.

[22] Carlin Chun Fai Chu and David Po Kin Chan. 2020. Feature Selection Using Approximated High-Order Interaction Components of the Shapley Value for Boosted Tree Classifier. *IEEE Access* 8 (2020), 112742–112750.

[23] Shay Cohen, Gideon Dror, and Eytan Ruppin. 2007. Feature Selection via Coalitional Game Theory. *Neural Comput.* 19, 7 (2007), 1939–1961.

[24] Shay B Cohen, Gideon Dror, and Eytan Ruppin. 2005. Feature selection based on the shapley value. In *IJCAI*. 1–6.

[25] Ben Cottier, Robi Rahman, Loredana Fattorini, Nestor Maslej, and David Owen. 2024. The rising costs of training frontier AI models. *arXiv* (2024).

[26] Christie Courtnage. 2022. A Systematic Study of Semi-Supervised Learning Based on Shapley Value Data Valuation. https://www.diva-portal.org/smash/get/diva2:1697410/FULLTEXT01.pdf

[27] Christie Courtnage and Evgueni Smirnov. 2021. Shapley-value data valuation for semi-supervised learning. In *DS 2021*. 94–108.

[28] Ian Covert and Su-In Lee. 2020. Improving kernelSHAP: Practical shapley value estimation via linear regression. *arXiv:2012.01536* (2020).

[29] Alfredo Cuzzocrea, Qudrat E. Alahy Ratul, Islam Belmerabet, and Edoardo Serra. 2023. Attribution Methods Assessment for Interpretable Machine Learning. In *SEBD*.

[30] Konstantinos Demertzis, Lazaros Iliadis, Panagiotis Kikiras, and Elias Pimenidis. 2022. An explainable semi-personalized federated learning model. *Integr. Comput.-Aided Eng.* 29, 4 (2022), 335–350.

[31] Daniel Deutch, Nave Frost, Benny Kimelfeld, and Mikaël Monet. 2022. Computing the Shapley value of facts in query answering. In *SIGMOD '22*. 1570–1583.

[32] Hongbin Dong, Jing Sun, and Xiaohang Sun. 2021. A multi-objective multi-label feature selection algorithm based on shapley value. *Entropy* 23, 8 (2021), 1094.

[33] Vaidotas Drungilas, Evaldas Vaičiukynas, Linas Ablonskis, and Lina Čeponienė. 2023. Shapley Values as a Strategy for Ensemble Weights Estimation. *Applied Sciences* 13, 12 (2023).

[34] Alexandre Duval and Fragkiskos D Malliaros. 2021. Graphsvx: Shapley value explanations for graph neural networks. In *ECML PKDD 2021*. 302–318.

[35] Zhenan Fan, Huang Fang, Zirui Zhou, Jian Pei, Michael P Friedlander, Changxin Liu, and Yong Zhang. 2022. Improving fairness for data valuation in horizontal federated learning. In *ICDE*. 2440–2453.

[36] Eitan Farchi, Ramasuri Narayanam, and Lokesh Nagalapatti. 2021. Ranking Data Slices for ML Model Validation: A Shapley Value Approach. In *ICDE*. 1937–1942.

[37] Matthias Feurer, Jan N. van Rijn, Arlind Kadra, Pieter Gijsbers, Neeratyoy Mallik, Sahithya Ravi, Andreas Mueller, Joaquin Vanschoren, and Frank Hutter. 2020. OpenML-Python: an extensible Python API for OpenML. *arXiv* 1911.02490 (2020). https://arxiv.org/pdf/1911.02490.pdf

[38] Philip Hans Franses, Jiahui Zou, and Wendun Wang. 2024. Shapley-value-based forecast combination. *Journal of Forecasting* 43, 8 (2024), 3194–3202.

[39] Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. 2021. Shapley explainability on the data manifold. In *ICLR*.

[40] Daniel Fryer, Inga Strümke, and Hien Nguyen. 2021. Shapley values for feature selection: The good, the bad, and the axioms. *IEEE Access* 9 (2021), 144352–144360.

[41] Felipe Garrido Lucero, Benjamin Heymann, Maxime Vono, Patrick Loiseau, and Vianney Perchet. 2024. Du-shapley: A shapley value proxy for efficient dataset valuation. *Advances in Neural Information Processing Systems* 37 (2024), 1973–2000.

[42] Amirata Ghorbani, Michael Kim, and James Zou. 2020. A distributional framework for data valuation. In *ICML*. 3535–3544.

[43] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *ICML*. 2242–2251.

[44] Amirata Ghorbani, James Zou, and Andre Esteva. 2022. Data shapley valuation for efficient batch active learning. In *ACSSC 2022*. 1456–1462.

[45] Jiyang Guan, Zhuozhuo Tu, Ran He, and Dacheng Tao. 2022. Few-shot backdoor defense using shapley estimation. In *CVPR*. 13358–13367.

[46] Eberhard Hechler, Maryela Weihrauch, and Yan (Catherine) Wu. 2023. *Data Fabric Architecture Patterns*. 231–255.

[47] Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. 2022. Collective eXplainable AI: Explaining Cooperative Strategies and Agent Contribution in Multiagent Reinforcement Learning With Shapley Values. *IEEE Computational Intelligence Magazine* 17, 1 (2022), 59–71.

[48] Jiyue Huang, Chi Hong, Lydia Y Chen, and Stefanie Roos. 2021. Is Shapley value fair? Improving client selection for mavericks in federated learning. *arXiv:2106.10734* (2021).

[49] Jiyue Huang, Rania Talbi, Zilong Zhao, Sara Bouchenak, Lydia Yiyu Chen, and Stefanie Roos. 2020. An Exploratory Analysis on Users' Contributions in Federated Learning. *IEEE TPS 2020* (2020), 20–29.

[50] Xuanxiang Huang and Joao Marques-Silva. 2024. On the failings of Shapley values for explainability. *International Journal of Approximate Reasoning* (2024), 109112.

[51] Lukas Huber, Marc Alexander Kühn, Edoardo Mosca, and Georg Groh. 2022. Detecting word-level adversarial text attacks via SHapley additive exPlanations. In *RepL4NLP 2022*. 156–166.

[52] Wasnaa Kadhim Jawad and Abbas M. Al-Bakry. 2022. Big Data Analytics: A Survey. *IJCI* (2022).

[53] Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. 2021. FastSHAP: Real-time shapley value estimation. In *ICLR*.

[54] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. 2019. Towards efficient data valuation based on the shapley value. In *AISTATS 2019*. 1167–1176.

[55] Nailcan Kara, Yagiz Levent Gume, Umit Tigrak, Gokce Ezeroglu, Serdar Mola, Omer Burak Akgun, and Arzucan Özgür. 2022. A SHAP-based Active Learning Approach for Creating High-Quality Training Data. In *IEEE BigData 2022*. 4002–4008.

[56] Seo-Hee Kim, Sun Young Park, Hyungseok Seo, and Jiyoung Woo. 2024. Feature selection integrating Shapley values and mutual information in reinforcement learning: An application in the prediction of post-operative outcomes in patients with end-stage renal disease. *Computer Methods and Programs in Biomedicine* 257 (2024), 108416.

[57] Patrick Kolpaczki, Viktor Bengs, Maximilian Muschalik, and Eyke Hüllermeier. 2024. Approximating the shapley value without marginal contributions. In *Proc.*

AAAI Conf. Artif. Intell, Vol. 38. 13246–13255.
[58] Patrick Kolpaczki, Georg Haselbeck, and Eyke Hüllermeier. 2024. How Much Can Stratification Improve the Approximation of Shapley Values?. In *Explainable Artificial Intelligence*. Cham, 489–512.
[59] Indra Kumar, Carlos Scheidegger, Suresh Venkatasubramanian, and Sorelle Friedler. 2021. Shapley Residuals: Quantifying the limits of the Shapley value for explanations. *NeurIPS* 34 (2021), 26598–26608.
[60] Indra Elizabeth Kumar, Carlos Eduardo Scheidegger, Suresh Venkatasubramanian, and Sorelle A. Friedler. 2021. Shapley Residuals: Quantifying the limits of the Shapley value for explanations. In *NeurIPS*.
[61] I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. 2020. Problems with Shapley-value-based explanations as feature importance measures. In *ICML*. 5491–5500.
[62] Yongchan Kwon, Manuel A Rivas, and James Zou. 2021. Efficient computation and analysis of distributional shapley values. In *AISTATS*. 793–801.
[63] Yongchan Kwon and James Y. Zou. 2022. WeightedSHAP: analyzing and improving Shapley based feature attributions. *arXiv* abs/2209.13429 (2022).
[64] Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. 2021. Shapley counterfactual credits for multi-agent reinforcement learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 934–942.
[65] Meng Li, Hengyang Sun, Yanjun Huang, and Hong Chen. 2024. Shapley value: from cooperative game to explainable artificial intelligence. *Auton. Intell. Syst.* 4 (2024), 2.
[66] Hong Lin, Shixin Wan, Zhongle Xie, Ke Chen, Meihui Zhang, Lidan Shou, and Gang Chen. 2024. A Comprehensive Study of Shapley Value in Data Analytics. *arXiv preprint arXiv:2412.01460* (2024).
[67] Xiaoqiang Lin, Xinyi Xu, See-Kiong Ng, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. 2023. Fair yet asymptotically equal collaborative learning. In *ICML*. 21223–21259.
[68] Jinfei Liu, Jian Lou, Junxu Liu, Li Xiong, Jian Pei, and Jimeng Sun. 2021. Dealer: an end-to-end model marketplace with differential privacy. *VLDB* 14, 6 (2021).
[69] Yuan Liu, Zhengpeng Ai, Shuai Sun, Shuangfeng Zhang, Zelei Liu, and Han Yu. 2020. Fedcoin: A peer-to-peer payment system for federated learning. In *Federated learning: privacy and incentive*. 125–138.
[70] Zelei Liu, Yuanyuan Chen, Han Yu, Yang Liu, and Lizhen Cui. 2022. Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning. *TIST* 13, 4 (2022), 1–21.
[71] Zelei Liu, Yuanyuan Chen, Yansong Zhao, Han Yu, Yang Liu, Renyi Bao, Jinpeng Jiang, Zaiqing Nie, Qian Xu, and Qiang Yang. 2022. Contribution-Aware Federated Learning for Smart Healthcare. *Proc. AAAI Conf. Artif. Intell* 36, 11 (2022), 12396–12404.
[72] Zhihong Liu, Hoang Anh Just, Xiangyu Chang, Xi Chen, and Ruoxi Jia. 2023. 2D-shapley: a framework for fragmented data valuation. In *ICML*. 21730–21755.
[73] Ester Livshits, Leopoldo Bertossi, Benny Kimelfeld, and Moshe Sebag. 2021. Query games in databases. *SIGMOD* 50, 1 (2021), 78–85.
[74] Ester Livshits, Leopoldo Bertossi, Benny Kimelfeld, and Moshe Sebag. 2021. The Shapley value of tuples in query answering. *LMCS* 17 (2021).
[75] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell* 2, 1 (2020), 56–67.
[76] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *NeurIPS*. 4768–4777.
[77] Xinjian Luo, Yangfan Jiang, and X. Xiao. 2022. Feature Inference Attack on Shapley Values. *ACM CCS 2022* (2022).
[78] Xuan Luo and Jian Pei. 2024. Applications and Computation of the Shapley Value in Databases and Machine Learning. In *SIGMOD/PODS '24*. 630–635.
[79] Xuan Luo, Jian Pei, Cheng Xu, Wenjie Zhang, and Jianliang Xu. 2024. Fast Shapley Value Computation in Data Assemblage Tasks as Cooperative Simple Games. *PACMMOD* 2, 1 (2024), 1–28.
[80] Shuaicheng Ma, Yang Cao, and Li Xiong. 2021. Transparent Contribution Evaluation for Secure Federated Learning on Blockchain. In *ICDEW*. 88–91.
[81] Sisi Ma and Roshan Tourani. 2020. Predictive and Causal Implications of using Shapley Value for Model Interpretation. In *PMLR*, Vol. 127. 23–38.
[82] Srujana Maddula. 2024. An Introduction to Data Orchestration: Process and Benefits. https://www.datacamp.com/blog/introduction-to-data-orchestration-process-and-benefits
[83] Wilson E Marcílio and Danilo M Eler. 2020. From explanations to feature selection: assessing SHAP values as feature selection mechanism. In *SIBGRAPI*. 340–347.
[84] Kolby Nottingham Markelle Kelly, Rachel Longjohn. 2025. The UCI Machine Learning Repository. https://archive.ics.uci.edu
[85] H. B. McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2016. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*.
[86] Ayesh Meepaganithage, Suman Rath, Mircea Nicolescu, Monica Nicolescu, and Shamik Sengupta. 2024. Feature Selection Using the Advanced Shapley Value. In *CCWC*. 0207–0213.
[87] Luke Merrick and Ankur Taly. 2020. The Explanation Game: Explaining Machine Learning Models Using Shapley Values. In *MAKE*. 17–38.
[88] Rory Mitchell, Joshua Cooper, Eibe Frank, and Geoffrey Holmes. 2022. Sampling permutations for shapley value estimation. *JMLR* 23, 43 (2022), 1–46.
[89] Maximilian Muschalik, Fabian Fumagalli, Barbara Hammer, and Eyke Hüllermeier. 2024. Beyond TreeSHAP: Efficient Computation of Any-Order Shapley Interactions for Tree Ensembles. *Proc. AAAI Conf. Artif. Intell* 38, 13 (2024), 14388–14396.
[90] Lokesh Nagalapatti and Ramasuri Narayanam. 2021. Game of gradients: Mitigating irrelevant clients in federated learning. In *Proc. AAAI Conf. Artif. Intell*, Vol. 35. 9046–9054.
[91] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. 2022. Trade-off between Payoff and Model Rewards in Shapley-Fair Collaborative Machine Learning. In *NeurIPS*, Vol. 35. 30542–30553.
[92] Ramin Okhrati and Aldo Lipani. 2020. A Multilinear Sampling Algorithm to Estimate Shapley Values. *ICPR* (2020), 7992–7999.
[93] Lars H. B. Olsen, Ingrid K. Glad, Martin Jullum, and Kjersti Aas. 2022. Using Shapley Values and Variational Autoencoders to Explain Predictive Models with Dependent Mixed Features. *JMLR* 23, 213 (2022), 1–51.
[94] Lars Henry Berge Olsen, Ingrid Kristine Glad, Martin Jullum, and Kjersti Aas. 2024. A comparative study of methods for estimating model-agnostic Shapley value explanations. *Data Min. Knowl. Discov.* 38, 4 (2024), 1782–1829.
[95] Ontotext. 2023. What Is Data Fabric? https://www.ontotext.com/knowledgehub/fundamentals/what-is-data-fabric/
[96] OpenAI. 2023. GPT-4 Technical Report. https://cdn.openai.com/papers/gpt-4.pdf
[97] Khaoula Otmani, Rachid Elazouzi, and Vincent Labatut. 2024. FedSV: Byzantine-Robust Federated Learning via Shapley Value. In *IEEE ICC*.
[98] Manisha Padala, Lokesh Nagalapatti, Atharv Tyagi, Ramasuri Narayanam, and Shiv Kumar Saini. 2025. Tab-Shapley: Identifying Top-k Tabular Data Quality Insights. *arXiv preprint arXiv:2501.06685* (2025).
[99] Konstantin D Pandl, Fabian Feiland, Scott Thiebes, and Ali Sunyaev. 2021. Trustworthy machine learning for health care: scalable data valuation with the shapley value. In *ACM CHIL*. 47–57.
[100] Jian Pei. 2020. A survey on data pricing: from economics to data science. *IEEE TKDE* 34, 10 (2020), 4586–4608.
[101] Guilherme Dean Pelegrina, Miguel Couceiro, and Leonardo Tomazeli Duarte. 2024. A preprocessing Shapley value-based approach to detect relevant and disparity prone features in machine learning. *ACM FACCT* (2024).
[102] Guilherme Dean Pelegrina and Sajid Siraj. 2024. Shapley value-based approaches to explain the quality of predictions by classifiers. *IEEE Transactions on Artificial Intelligence* (2024).
[103] Guilherme Dean Pelegrina, Sajid Siraj, Leonardo Tomazeli Duarte, and Michel Grabisch. 2024. Explaining contributions of features towards unfairness in classifiers: A novel threshold-dependent Shapley value-based approach. *Engineering Applications of Artificial Intelligence* 138 (2024), 109427.
[104] Annabelle Redelmeier, Martin Jullum, and Kjersti Aas. 2020. Explaining Predictive Models with Mixed Features Using Shapley Values and Conditional Inference Trees. In *MAKE*. 117–137.
[105] Market Research Report. 2024. Big Data Technology Market Size, Share & Industry Analysis. (2024). https://www.fortunebusinessinsights.com/data-analytics-market-108882
[106] Maria Rigaki and Sebastian Garcia. 2023. A survey of privacy attacks in machine learning. *Comput. Surveys* 56, 4 (2023), 1–34.
[107] Benedek Rozemberczki and Rik Sarkar. 2021. The shapley value of classifiers in ensemble games. In *ACM CIKM*. 1558–1567.
[108] Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. 2022. The Shapley Value in Machine Learning. In *IJCAI-22*, Lud De Raedt (Ed.). 5572–5579. Survey Track.
[109] Franco Ruggeri, William Emanuelsson, Ahmad Terra, Rafia Inam, and Karl H. Johansson. 2024. Rollout-based Shapley Values for Explainable Cooperative Multi-Agent Reinforcement Learning. In *2024 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*. 227–233.
[110] Mohammadreza Salarbashishahri, Samuel D. Okegbile, and Jun Cai. 2022. A Shapley value-enhanced evaluation technique for effective aggregation in Federated Learning. In *FNWF*. 88–93.
[111] Stephanie Schoch, Haifeng Xu, and Yangfeng Ji. 2022. CS-Shapley: class-wise Shapley values for data valuation in classification. *NeurIPS* 35 (2022), 34574–34585.
[112] Carlos Sebastián and Carlos E González-Guillén. 2024. A feature selection method based on Shapley values robust for concept shift in regression. *Neural. Comput. Appl.* (2024), 1–23.
[113] Lloyd S Shapley. 1953. A value for n-person games. *Contribution to the Theory of Games* 2 (1953).
[114] Yong Shi. 2022. Advances in Big Data Analytics: Theory, Algorithms and Practices. *Advances in Big Data Analytics* (2022).

[115] Yiwei Shi, Qi Zhang, Kevin McAreavey, and Weiru Liu. 2024. Counterfactual shapley values for explaining reinforcement learning. *arXiv e-prints* (2024), arXiv–2408.

[116] Yiwei Shi, Qi Zhang, Kevin McAreavey, and Weiru Liu. 2024. Explaining Reinforcement Learning: A Counterfactual Shapley Values Approach. *arXiv preprint arXiv:2408.02529* (2024).

[117] Zhuan Shi, Lan Zhang, Zhenyu Yao, Lingjuan Lyu, Cen Chen, Li Wang, Junhao Wang, and Xiang-Yang Li. 2022. FedFAIM: A Model Performance-based Fair Incentive Mechanism for Federated Learning. *IEEE TBD* (2022), 1–13.

[118] Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. 2021. Online class-incremental continual learning with adversarial shapley value. In *Proc. AAAI Conf. Artif. Intell*, Vol. 35. 9630–9638.

[119] Seyedamir Shobeiri and Mojtaba Aajami. 2021. Shapley value in convolutional neural networks (CNNs): A Comparative Study. *AJMSE* 2, 3 (2021), 9–14.

[120] Seyedamir Shobeiri and Mojtaba Aajami. 2022. Shapley Value is an Equitable Metric for Data Valuation. *IJEEE* 18, 2 (2022).

[121] Michelle Si and Jian Pei. 2024. Counterfactual Explanation of Shapley Value in Data Coalitions. *pVLDB* 17, 11 (2024), 3332–3345.

[122] Rachael Hwee Ling Sim, Xinyi Xu, and Bryan Kian Hsiang Low. 2022. Data Valuation in Machine Learning:" Ingredients", Strategies, and Open Challenges. In *IJCAI*. 5607–5614.

[123] Rachael Hwee Ling Sim, Yehong Zhang, Mun Choon Chan, and Bryan Kian Hsiang Low. 2020. Collaborative Machine Learning with Incentive-Aware Model Rewards. In *ICML PMLR*, Vol. 119. 8927–8936.

[124] Pranava Singhal, Shashi Raj Pandey, and Petar Popovski. 2024. Greedy Shapley Client Selection for Communication-Efficient Federated Learning. *IEEE Networking Letters* 6, 2 (2024), 134–138.

[125] Qiheng Sun, Xiang Li, Jiayao Zhang, Li Xiong, Weiran Liu, Jinfei Liu, Zhan Qin, and Kui Ren. 2023. Shapleyfl: Robust federated learning based on shapley value. In *ACM SIGKDD*. 2096–2108.

[126] Qiheng Sun, Jiayao Zhang, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. 2024. Shapley Value Approximation Based on Complementary Contribution. *IEEE Transactions on Knowledge and Data Engineering* 36, 12 (2024), 9263–9281.

[127] Mukund Sundararajan and Amir Najmi. 2020. The many Shapley values for model explanation. In *ICML*. 9269–9278.

[128] Zuoqi Tang, Zheqi Lv, and Chao Wu. 2020. A Brief SURVEY OF DATA PRICING FOR MACHINE LEARNING. In *CS & IT Conference Proceedings*, Vol. 10.

[129] Zuoqi Tang, Feifei Shao, Long Chen, Yunan Ye, Chao Wu, and Jun Xiao. 2021. Optimizing Federated Learning on Non-IID Data Using Local Shapley Value. In *Artif. Intell.* 164–175.

[130] Nurbek Tastan, Samar Fares, Toluwani Aremu, Samuel Horvath, and Karthik Nandakumar. 2024. Redefining Contributions: Shapley-Driven Federated Learning. *arXiv:2406.00569* (2024).

[131] Yingjie Tian, Yurong Ding, Saiji Fu, and Dalian Liu. 2022. Data boundary and data pricing based on the shapley value. *IEEE Access* 10 (2022), 14288–14300.

[132] Zhihua Tian, Jian Liu, Jingyu Li, Xinle Cao, Ruoxi Jia, Jun Kong, Mengdi Liu, and Kui Ren. 2022. Private data valuation and fair payment in data marketplaces. *arXiv:2210.08723* (2022).

[133] Sandhya Tripathi, N Hemachandra, and Prashant Trivedi. 2020. Interpretable feature subset selection: A Shapley value based approach. In *IEEE BigData 2020*. 5463–5472.

[134] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. 2013. OpenML: networked science in machine learning. *SIGKDD Explorations* 15, 2 (2013), 49–60. https://doi.org/10.1145/2641190.2641198

[135] Fangdi Wang, Jiaqi Jin, Jingtao Hu, Suyuan Liu, Xihong Yang, Siwei Wang, Xinwang Liu, and En Zhu. 2024. Evaluate then Cooperate: Shapley-based View Cooperation Enhancement for Multi-view Clustering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=xoc4QOvbDs

[136] Guan Wang, Charlie Xiaoqian Dang, and Ziye Zhou. 2019. Measure contribution of participants in federated learning. In *IEEE BigData 2019*. 2597–2604.

[137] Junhao Wang, Lan Zhang, Anran Li, Xuanke You, and Haoran Cheng. 2022. Efficient Participant Contribution Evaluation for Horizontal and Vertical Federated Learning. In *ICDE*. 911–923.

[138] Jiachen T Wang, Tianji Yang, James Zou, Yongchan Kwon, and Ruoxi Jia. 2024. Rethinking data shapley for data selection tasks: Misleads and merits. *arXiv preprint arXiv:2405.03875* (2024).

[139] Jiachen T Wang, Yuqing Zhu, Yu-Xiang Wang, Ruoxi Jia, and Prateek Mittal. 2023. Threshold knn-shapley: A linear-time and privacy-friendly approach to data valuation. *arXiv:2308.15709* (2023).

[140] Rui Wang, Xiaoqian Wang, and David I. Inouye. 2021. Shapley Explanation Networks. In *ICLR*.

[141] Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. 2020. A principled approach to data valuation for federated learning. *Federated Learning: Privacy and Incentive* (2020), 153–167.

[142] Yong Wang, Kaiyu Li, Yuyu Luo, Guoliang Li, Yunyan Guo, and Zhuo Wang. 2024. Fast, Robust and Interpretable Participant Contribution Estimation for Federated Learning. In *ICDE*. 2298–2311.

[143] Ziming Wang, Changwu Huang, Yun Li, and Xin Yao. 2024. Multi-objective feature attribution explanation for explainable machine learning. *ACM Transactions on Evolutionary Learning and Optimization* 4, 1 (2024), 1–32.

[144] Zexin Wang, Biwei Yan, and Anming Dong. 2022. Blockchain Empowered Federated Learning for Data Sharing Incentive Mechanism. *Procedia Computer Science* 202 (2022), 348–353. International Conference on Identification, Information and Knowledge in the internet of Things, 2021.

[145] David Watson, Joshua O' Hara, Niek Tax, Richard Mudd, and Ido Guy. 2023. Explaining Predictive Uncertainty with Information Theoretic Shapley Values. In *NeurIPS*, Vol. 36. 7330–7350.

[146] Lauren Watson, Rayna Andreeva, Hao Yang, and Rik Sarkar. 2022. Differentially Private Shapley Values for Data Evaluation. *arXiv* abs/2206.00511 (2022).

[147] Shuyue Wei, Yongxin Tong, Zimu Zhou, and Tianshu Song. 2020. Efficient and fair data valuation for horizontal federated learning. *Federated Learning: Privacy and Incentive* (2020), 139–152.

[148] Wikipedia. 2025. Law of Large Numbers. https://en.wikipedia.org/wiki/Law_of_large_numbers

[149] Brian Williamson and Jean Feng. 2020. Efficient nonparametric statistical inference on population feature importance using Shapley values. In *ICML PMLR*, Vol. 119. 10282–10291.

[150] Chengqian Wu, Xuemei Fu, Xiangli Yang, Ruonan Zhao, Qidong Wu, and Tinghua Zhang. 2023. CP-Decomposition Based Federated Learning with Shapley Value Aggregation. In *ICPADS*. 571–577.

[151] Mengmeng Wu, Ruoxi Jia, Changle Lin, Wei Huang, and Xiangyu Chang. 2023. Variance reduced Shapley value estimation for trustworthy data valuation. *COR* 159 (2023), 106305.

[152] Binhan Xi, Shaofeng Li, Jiachun Li, Hui Liu, Hong Liu, and Haojin Zhu. 2021. BatFL: Backdoor Detection on Federated Learning in e-Health. In *IWQOS*. 1–10.

[153] Haocheng Xia, Xiang Li, Junyuan Pang, Jinfei Liu, Kui Ren, and Li Xiong. 2024. P-Shapley: Shapley Values on Probabilistic Classifiers. *VLDB* 17, 7 (2024), 1737–1750.

[154] Haocheng Xia, Jinfei Liu, Jian Lou, Zhan Qin, Kui Ren, Yang Cao, and Li Xiong. 2023. Equitable data valuation meets the right to be forgotten in model markets. *VLDB* 16, 11 (2023), 3349–3362.

[155] Haocheng Xia, Jiayao Zhang, Qiheng Sun, Jinfei Liu, Kui Ren, Li Xiong, and Jian Pei. 2025. Computing Shapley Values for Dynamic Data. *IEEE Transactions on Knowledge and Data Engineering* (2025).

[156] Lei Xu, Jiaqing Chen, Shan Chang, Cong Wang, and Bo Li. 2023. Toward Quality-aware Data Valuation in Learning Algorithms: Practices, Challenges, and Beyond. *IEEE Network* (2023), 1–1.

[157] Xinyi Xu, Thanh Lam, Chuan Sheng Foo, and Bryan Kian Hsiang Low. 2024. Model Shapley: Equitable Model Valuation with Black-box Access. *NeurIPS* 36 (2024).

[158] Xinyi Xu, Lingjuan Lyu, Xingjun Ma, Chenglin Miao, Chuan Sheng Foo, and Bryan Kian Hsiang Low. 2021. Gradient Driven Rewards to Guarantee Fairness in Collaborative Machine Learning. In *NeurIPS*, Vol. 34. 16104–16117.

[159] Chengyi Yang, Zhaoxiang Hou, Sheng Guo, Hui Chen, and Zengxiang Li. 2023. SWATM: Contribution-Aware Adaptive Federated Learning Framework Based on Augmented Shapley Values. In *ICME*. 672–677.

[160] Chengyi Yang, Jia Liu, Hao Sun, Tongzhi Li, and Zengxiang Li. 2022. WTDP-Shapley: Efficient and effective incentive mechanism in federated learning for intelligent safety inspection. *IEEE TBD* (2022).

[161] Jilei Yang. 2021. Fast TreeSHAP: Accelerating SHAP Value Computation for Trees. *arXiv* abs/2109.09847 (2021).

[162] Xun Yang, Weijie Tan, Changgen Peng, Shuwen Xiang, Kun Niu, et al. 2022. Federated learning incentive mechanism design via enhanced shapley value method. *Wireless Communications and Mobile Computing* 2022 (2022).

[163] Xun Yang, Shuwen Xiang, Changgen Peng, Weijie Tan, Zhuguo Li, Ningbo Wu, and Yan Zhou. 2023. Federated Learning Incentive Mechanism Design via Shapley Value and Pareto Optimality. *Axioms* 12 (2023), 636.

[164] Xun Yang, Shuwen Xiang, Changgen Peng, Weijie Tan, Yue Wang, Hai Liu, and Hongfa Ding. 2024. Federated Learning Incentive Mechanism with Supervised Fuzzy Shapley Value. *Axioms* 13, 4 (2024), 254.

[165] Dingze Yin, Dan Chen, Yunbo Tang, Heyou Dong, and Xiaoli Li. 2022. Adaptive feature selection with shapley and hypothetical testing: Case study of EEG feature engineering. *INS* 586 (2022), 374–390.

[166] Zhaoyang You, Xinya Wu, Kexuan Chen, Xinyi Liu, and Chao Wu. 2021. Evaluate the Contribution of Multiple Participants in Federated Learning. In *DEXA*. 189–194.

[167] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. 2020. A Fairness-aware Incentive Scheme for Federated Learning. In *AIES '20*. 393–399.

[168] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. 2020. A Sustainable Incentive Scheme for Federated Learning. *IEEE Intelligent Systems* 35, 4 (2020), 58–69.

[169] Peng Yu, Albert Bifet, Jesse Read, and Chao Xu. 2022. Linear tree shap. In *NeurIPS*.

[170] Dan Zhang, L.G. Pee, Shan L. Pan, and Lili Cui. 2022. Big data analytics, resource orchestration, and digital sustainability: A case study of smart city development. *Government Information Quarterly* 39, 1 (2022), 101626.

[171] Daniel Yue Zhang, Ziyi Kou, and Dong Wang. 2020. FairFL: A Fair Federated Learning Approach to Reducing Demographic Bias in Privacy-Sensitive Classification Models. In *IEEE BigData 2020*. 1051–1060.

[172] Jiayao Zhang, Qiheng Sun, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. 2023. Efficient sampling approaches to shapley value approximation. *PACMMOD* 1, 1 (2023), 1–24.

[173] Jiayao Zhang, Haocheng Xia, Qiheng Sun, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. 2023. Dynamic shapley value computation. In *ICDE*. 639–652.

[174] Ningsheng Zhao, Jia Yuan Yu, Krzysztof Dzieciolowski, and Trang Bui. 2024. Error Analysis of Shapley Value-Based Model Explanations: An Informative Perspective. In *International Symposium on AI Verification*. 29–48.

[175] Quan Zheng, Ziwei Wang, Jie Zhou, and Jiwen Lu. 2022. Shap-CAM: Visual explanations for convolutional neural networks based on Shapley value. In *ECCV*. 459–474.

[176] Shuyuan Zheng, Yang Cao, and Masatoshi Yoshikawa. 2022. Secure Shapley Value for Cross-Silo Federated Learning (Technical Report). *arXiv:2209.04856* (2022).

[177] Haolin Zhu, Ziye Li, Dingzhi Zhong, Cheng Li, and Yong Yuan. 2023. Shapley-value-based Contribution Evaluation in Federated Learning: A Survey. *2023 IEEE 3rd International Conference on Digital Twins and Parallel Intelligence (DTPI)* (2023), 1–5.

[178] Ye Zhu, Zhiqiang Liu, Peng Wang, and Chenglie Du. 2023. A dynamic incentive and reputation mechanism for energy-efficient federated learning in 6G. *Digital Communications and Networks* 9, 4 (2023), 817–826.