Trustworthy Machine Learning for Health Care: Scalable Data Valuation with the Shapley Value

Konstantin D. Pandl
Karlsruhe Institute of Technology
konstantin.pandl@kit.edu

Fabian Feiland
Karlsruhe Institute of Technology
fabianfeiland@web.de

Scott Thiebes
Karlsruhe Institute of Technology
scott.thiebes@kit.edu

Ali Sunyaev
Karlsruhe Institute of Technology
sunyaev@kit.edu

ABSTRACT

Collecting data from many sources is an essential approach to generate large data sets required for the training of machine learning models. Trustworthy machine learning requires incentives, guarantees of data quality, and information privacy. Applying recent advancements in data valuation methods for machine learning can help to enable these. In this work, we analyze the suitability of three different data valuation methods for medical image classification tasks, specifically pleural effusion, on an extensive data set of chest X-ray scans. Our results reveal that a heuristic for calculating the Shapley valuation scheme based on a k-nearest neighbor classifier can successfully value large quantities of data instances. We also demonstrate possible applications for incentivizing data sharing, the efficient detection of mislabeled data, and summarizing data sets to exclude private information. Thereby, this work contributes to developing modern data infrastructures for trustworthy machine learning in health care.

CCS CONCEPTS

• Applied computing -> Health informatics • Computing methodologies -> Machine learning

KEYWORDS

Computer Vision, Data Valuation, Machine Learning, Medical Imaging, Shapley Value

ACM Reference format:

Konstantin D. Pandl, Fabian Feiland, Scott Thiebes, Ali Sunyaev. 2021. In *ACM Conference on Health, Inference, and Learning (ACM CHIL '21)*, April 8–10, 2021, Virtual Event, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3450439.3451861>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ACM CHIL '21, April 8–10, 2021, Virtual Event, USA

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8359-2/21/04.

<https://doi.org/10.1145/3450439.3451861>

1 Introduction

The application of recent advances in machine learning (ML) can enable data-driven, personalized, and resource-efficient health care systems. For example, in the field of medical imaging, ML-based methods have repeatedly shown to accurately detect diseases such as pneumonia [1] or more recently COVID-19 [2] on chest X-ray scans, sometimes more accurately than professional human radiologists. While such research results are promising, the actual deployment of ML-based information systems in health care settings is still at an early stage, at best. Health care requires trustworthy ML systems built on new types of trustworthy data infrastructures [3]. For example, ML-based health information systems should ensure the protection of patient's information privacy [4]. At the same time, large data sets are required for the training of ML-based systems, and these data sets should include data instances from a variety of sources, including minorities, to perform well on a wide range of tasks [4]. Toward this end, prior research proposed incentivizing data sharing through marketplaces, which may, however, be in tension with privacy requirements [3]. Furthermore, to ensure well-performing ML-based systems, ensuring data quality is of utmost importance. Even data sets labeled by professional radiologists can contain a significant portion of mislabeled instances, which manifests the need for methods to identify mislabeled instances [5].

Data valuation methods are an emerging field of ML research that may serve to address several of the challenges mentioned earlier. They assign a value to each data instance in a data set, whereby a value corresponds to the respective data instances' usefulness for the overall ML task. Applications of data valuation methods include, among others, the identification of valuable data instances in a data set to incentivize the sharing of this data instance [6, 7], the identification of mislabeled data instances, or the summarization of data sets [8], for example, to improve privacy by excluding certain, highly-sensitive data instances.

Extant research has proposed several methods for data valuation [8-10], which differ in their data valuation performance, computational cost, and guaranteed fairness properties. By evaluating the findings with professional radiologists, recent research indicates that data valuation methods can be beneficial for medical imaging tasks, for example, to detect mislabeled instances on a data set of ca. 2,000 chest X-ray scans, or to remove data instances [11].

With this research, we aim to empirically analyze the suitability of more scalable data valuation methods to value extensive amounts of medical imaging data (in the order of 100,000 data instances). Furthermore, we explore how they can be applied to deploy trustworthy ML in the medical domain with regard to incentivizing data sharing to establish large and diverse data sets, enabling data label correctness, and summarizing data sets, thereby enhancing individual's privacy. In doing so, we contribute to the extant body of knowledge on trustworthy ML in the medical field in three ways. First, we analyze the phenomena of different data instance values on an extensive medical imaging data set. Thereby, we show that an approximation of the Shapley value (SV) based on a k -nearest neighbors (KNN) classifier can successfully value large amounts of data instances within a reasonable amount of computation time. Second, we publish our source code [12], allowing other researchers to reproduce our results and evaluate our methods in different settings. Third, we demonstrate useful applications of the SV for incentivizing medical data sharing by valuing medical imaging data instances, identifying mislabeled instances, and summarizing data sets.

The remainder of this paper is organized as follows. Section two provides a brief background on machine learning and in-depth background on various data valuation methods. Afterward, we outline the conceptual foundations of our experiments in section three. In section four, we first describe the results of a runtime comparison of different data valuation methods before focusing on the scalable KNN Shapley value method, presenting the SV distribution, and analyzing two potential applications. We discuss these results in section five and conclude the paper with section six.

2 Background

2.1 Machine learning

Nowadays, many ML settings are supervised ML settings, consisting of three main components [8]. First, the training data set $D = \{z_i\}_{i=1}^N$ that defines the amount of experience an ML algorithm has during the training phase. Each training data instance z_i is a feature-label pair (x_i, y_i) . Apart from the training data, there is a disjoint set of data, the test data D_{val} , used to evaluate the ML algorithm. Second, supervised ML consists of a learning algorithm A that is used to learn a task. This learning algorithm can be imagined as a black box that

takes the training data set D as an input and returns a trained model. Third, supervised ML settings consist of a performance metric U that measures a trained model's overall performance. Again, the performance metric can be thought of as a black box that takes any predictor as an input and returns a score. The score $U(S, A, D_{val})$ is calculated on the test set and indicates the quality of a model trained on a subset S of the training data set D .

A common performance metric for binary classification, and in health care in general [4], is the area under the receiver operating characteristic curve (henceforth AUC score). For the AUC score, the predicted class is compared with the actual class after the prediction. The four possible outcomes of the comparison are true positive TP (i.e., the data instance is correctly classified as positive), false negative FN (i.e., the data instance is falsely classified as negative), true negative TN (i.e., the data instance is correctly classified as negative) and false positive FP (i.e., the data instance is falsely classified as positive). With this knowledge, the receiver operating characteristics (ROC) score can be calculated as follows. First, every classified instance is assigned to one of the four possible outcomes. Second, the true positive rate $TPR = \frac{TP}{TP+FN}$ and false positive rate $FPR = \frac{FP}{FP+TN}$ are calculated. Third, the ROC curve, a two-dimensional graph with TPR on the Y-axis and FPR on the X-axis, is plotted. An important interpretation of the ROC curve is that it reveals the trade-offs between benefits (i.e., rate of data instances correctly classified as positive, TPR) and costs (i.e., rate of data instances falsely classified as positive, FPR). Finally, the AUC score is measured as the geometrical area under the receiver operating characteristic curve. Notable AUC scores are 1.0, which corresponds to a perfect, faultless classification, and the AUC score of 0.5, which represents the strategy of randomly guessing a class.

2.2 Data valuation for machine learning

Not all training data instances contribute equally to an ML task, but some data instances contribute much more than others [13, 14]. Data valuation methods aim to assign a value to each training data instance z_i , with respect to the learning algorithm A and the performance metric U [9]. Each training data instance is allocated an 'equitable' share for its contribution toward achieving the performance. This data value is denoted by $\varphi(z_i, D, A, D_{val}, U)$, short $\varphi(z_i)$.

For an equitable valuation scheme, three conditions are desirable [8-10]. First, group rationality, meaning the entire data set's value is completely distributed among all training data instances, as denoted in the following equation: $U(D) = \sum_{z_i \in D} \varphi(z_i)$. Second, fairness, meaning the value of a data instance z_i should be zero if the data instance has zero marginal contribution to all subsets of the data set: i.e., $\varphi(z_i) = 0$ if $U(S \cup \{z_i\}) = 0$, for all $S \subseteq D \setminus \{z_i\}$. At the same time, two data instances z_i, z_j should have the same value if they both add the

same performance to a subset of the data set: i.e., $\varphi(z_i) = \varphi(z_j)$ if $U(S \cup \{z_i\}) = U(S \cup \{z_j\})$, for all $S \subseteq D \setminus \{z_i, z_j\}$. Third, additivity, meaning that if the overall performance score is the sum of separate scores U_1 and U_2 , the overall value of a data instance should be the sum of its value for each score U_1 and U_2 : $\varphi(z_i, U_1 + U_2) = \varphi(z_i, U_1) + \varphi(z_i, U_2)$, for all $z_i \in D$.

2.3 Principal methods for data valuation

2.3.1 Leave-one-out method. A simple way to allocate a value to a data instance is the so-called leave-one-out (LOO) method. With this method, the ML model is trained on the entire training data set, and the performance score U is calculated on the test data set. Afterward, the examined training instance is removed from the training data set (hence the name 'leave one out'), and the model is retrained on this subset. The model trained on this subset is then again evaluated on the test data. Finally, $\varphi_{loo}(z_i)$ is calculated for the examined training instance by comparing the two different performance scores: $\varphi_{loo}(z_i) = U(D) - U(D \setminus \{z_i\})$. This procedure can be repeated for all the other data instances within the training data set. Figure 1A illustrates the principal procedure of the LOO method.

Despite being easy to calculate, LOO values face two major limitations. On the one hand, LOO calculations do not meet the standards of an equitable value scheme as defined in section 2.2. On the other hand, calculating LOO values for N data instances requires retraining the model for N times. As such, LOO calculations result in high computational costs, making them potentially improper for large data sets and models.

2.3.2 Shapley value. Another method to compute the value of a data instance is the so-called SV. Originally, the SV is a game-theoretic valuation scheme that uniquely divides the reward for players' cooperation, such that each player would get an equitable share of the reward [8]. A supervised ML problem can also be thought of as a cooperative game, whereby each training data instance is a player, and the players work together through the learning algorithm A to achieve the prediction score $U(S, A, D_{val})$. Accordingly, the SV for every training data instance z_i as its average marginal contribution to all possible subsets S of D formed by other training data instances is defined as [9]:

$$\varphi_{shap}(z_i) = \frac{1}{N} \sum_{S \subseteq D \setminus \{z_i\}} \frac{1}{\binom{N-1}{|S|}} [U(S \cup \{z_i\}) - U(S)]$$

Following this definition, the SV uniquely satisfies the properties of group rationality, fairness, and additivity. Thereby, it is capable of identifying valuable instances in a training data set [8, 9]. Low SVs capture counterproductive outliers, whereas high SVs provide insight into which type of data instances help improve the ML predictor. As such, the SV can be utilized as a tool for tasks like the correction of noisy

labels, detection of watermarks, summarization of data sets, or active data acquisition. In comparison with LOO methods, empirical experiments have shown that SV-based methods are more effective at identifying valuable data instances [8, 9].

However, computing the SV of a data set's data instances is even more complex than calculating LOO values. Evaluating the exact SV requires computing the marginal contribution of each training point to all possible subsets. Hence, SV calculations grow exponentially in the number of training data instances with a complexity of $O(2^N)$ [9].

2.4 Approximations for the Shapley value

Since computing the exact SV is NP-hard and therefore only feasible for a small number of players (i.e., input features in case of ML) [15], extant research proposes several approximations for the SV.

Among these are heuristic methods that aim to calculate Shapley values more efficiently, such as the Truncated Monte Carlo (TMC) Shapley value [16], or methods that aim to approximate the ML model itself, such as the KNN-Shapley value [9]. In the following, we describe both approaches in detail, which we also later implement and evaluate.

2.4.1 Truncated Monte Carlo Shapley value. A state-of-the-art approach to estimate the SV is based on Monte Carlo simulations [16]. Thereby, the SV is considered as the expectation of a training instance's marginal contribution to a random set, and the sample mean is used for the approximation. A heuristic ignores the large random sets since a data sample's contribution to these will be small [8]. Figure 1B illustrates the calculation of SVs with the TMC-Shapley method.

First, a random permutation of the training data set is sampled. Second, the permutation is scanned from the first element to the last element, and the marginal contribution for every training data instance to the subset is calculated. Once the marginal contributions become small, the computations are truncated, and the marginal contributions of the following elements are approximated with zero. The same procedure is repeated over multiple Monte Carlo permutations. Finally, the SVs for the different training instances are computed as the average of the marginal contributions to the different permutations.

While this approach improves the SV's efficiency, it still requires retraining the model multiple times, which results in high, albeit reduced, computational cost [9]. Prior research, therefore, applied the TMC-Shapley method with a logistic regression learning algorithm on extracted features from a pre-trained convolutional neural network (CNN) and successfully valued medical image scans in the order of 2,000 data instances [11]. Thereby, the TMC-Shapley proved to be a more accurate measure of a data instance's importance than the LOO value.

Figure 1: Principal procedure of the LOO, TMC-Shapley, and KNN-Shapley method for data valuation.

2.4.2 k -nearest neighbor Shapley value. The core idea of the KNN-Shapley is to approximate the complex deep neural network model by a KNN classifier [9]. In general, a KNN classifier is trained by merely storing the training data instances. When a new query instance is encountered, similar related instances are recalled from the memory and used to classify the new query instance [17]. As such, a KNN classifier's training process is typically computationally inexpensive, and only similar training instances to the query instance affect a prediction, but not all of the training instances. More precisely, for a single query instance x_{val} with the label y_{val} , KNN finds the first K training data instances $(x_{\alpha_1}, \dots, x_{\alpha_K})$ that are most similar to x_{val} , where the similarity is defined by a certain distance metric. The corresponding labels of these neighbors $(y_{\alpha_1}, \dots, y_{\alpha_K})$ then provide information about the likely label of the query instance x_{val} .

The KNN classifier can either be trained with raw data or with a feature representation of the data set (i.e., embeddings or deep features). Deep features correspond to the outputs of the different feature extraction layers of a deep neural network. To obtain these deep features, the deep neural network is trained on the training data set. For every deep feature of the test set, a KNN classifier is trained. Thereby, k (i.e., the number of neighbors) is calibrated such that the resulting KNN classifier imitates the performance of the original deep neural network.

Afterward, the KNN-Shapley values can be computed for all training data instances. For this, the confidence of predicting the right labels can be used as the performance score: $U(S) = \frac{1}{K} \sum_{k=1}^{\min(K, |S|)} \mathbb{1}[y_{\alpha_k(S)} = y_{val}]$. The SVs for each training data instance can then be calculated recursively [18]. Thereby, the

SV for the k -th data instance is computed first using equation 1. Afterward, the other SVs can be computed using equation 2.

$$\varphi(z_{\alpha_N}) = \frac{\mathbb{1}[y_{\alpha_N} = y_{val}]}{N} \quad (1)$$

$$\varphi(z_{\alpha_i}) = \varphi(z_{\alpha_{i+1}}) + \frac{\mathbb{1}[y_{\alpha_i} = y_{val}] - \mathbb{1}[y_{\alpha_{i+1}} = y_{val}]}{K} \frac{\min(K, i)}{i} \quad (2)$$

The main advantage of KNN-Shapley is that the SVs are calculated with fewer and smaller subsets of the nearest neighbors of the valued data instance, instead of all subsets from the entire training data set $S \subseteq D \setminus \{z_i\}$ [9].

With a complexity of $O(Nd + N \log N)$ (i.e., N training data instances, d is the dimension of deep feature representation), the complexity scales quasi-linearly with the training data set size, linearly with the validation data set size, and independent of the model size [9]. Remarkably, the model does not need to be retrained for the calculations of the SVs. In conclusion, the KNN-Shapley seems to be well suited for large models as the complexity is independent of the model size, and the model does not need to be retrained. Furthermore, it still satisfies the above-outlined properties of group rationality, fairness, and additivity.

3 Experimental setup

To evaluate the previously described data valuation methods' suitability for large medical data sets, we conducted a set of four experiments. We describe these experiments' basic setup, including the underlying data set, data preprocessing steps, principal machine learning approach, and actual implementation in the following.

3.1 Data set

As a data set for our experiments, we employ the CheXpert data set [19]. It is an extensive data set of 224,316 chest X-ray scans, which has received significant attention in prior research on computer vision for medical imaging, such as explainable AI in radiology [20] or the comparison of neural network architectures for radiology [21]. The CheXpert data set was recently used to create COVID-19 disease detectors on chest X-rays [22, 23]. Within the data set, 85% are frontal scans, while the remaining 15% account for lateral scans. Each data instance of the data set consists of an X-ray scan and 19 labels. While five of the labels provide general information about the patient, like their patient ID (total: 65,240 patients), age, or sex, 14 labels represent clinical observations (i.e., no finding, support devices, and 12 medical conditions such as fractures or pleural effusion) of a given X-ray scan. Each of the 14 labels either contains a 1 (i.e., clinical observation is present for the scan), a 0 (i.e., clinical observation is confidently absent for the scan), a U (i.e., clinical observation is uncertainly present for the scan) or NaN (i.e., no information about clinical observation for the

scan). Overall, out of the 224,316 scans and corresponding labels of the CheXpert data set, the training data set (223,414 data instances) and the validation set (234 data instances) are provided publicly, whereas a test set with the other instances is kept private [19].

3.2 Data preprocessing

For our study, we focused on the medical condition of pleural effusion for multiple reasons. First, pleural effusion has high clinical importance as every year more than 1.5 million people are affected in the United States alone [24]. Second, it is analyzed with ML models in the original CheXpert study, allowing us to better compare our results [19]. Third, among this subset, the share of 1.0 labels for pleural effusion is the highest (38.59%). In contrast, the NaN labels' share is the lowest compared to the other observations (40.37%), making the pleural effusion subset an interesting subset for data valuation. As we aim to value the data instances of an extensive, real-world medical data set, we considered scans of all available labels and types: frontal and lateral scans, with explicit (i.e., 1, 0, or U), as well as implicit labels (i.e., NaN). For the classification task, we treated data instances with a NaN and a U-label as confidently absent (0), similar to prior work on the CheXpert data set [19]. As a result, 38.59% of the data instances had a pleural effusion present (1) and 61.41% absent (0). The ML task thus is a binary classification problem for one medical observation.

To create subsets (i.e., training data set, test data set), we first grouped the data instances by patient ID to ensure that a patient occurs only in one subset. We then applied a common 80:20-split to create a disjoint set of 178,731 training data instances and 44,683 test data instances. On top of the training and test sets, we used the publicly available validation set of 234 data instances to assess the model's performance during the training phase.

As a further preprocessing step, we resized the X-ray images to 224×224 pixels, enabling the use of the pretrained DenseNet-121 on the weights of the ImageNet data set [25], which we describe in the next section. By doing so, we achieved better results than by keeping the input image size of 320×320 and not using the pre-trained DenseNet-121. We achieved the best results with images that were neither normalized nor augmented.

3.3 Principal machine learning approach

3.3.1 Performance metric. For the data valuation, we used the prediction certainty as the performance score. Furthermore, we also evaluated the ML models with the AUC score. It has the advantage that it is not affected by class imbalances in the data set. It is also commonly used in medical decision making in general and binary classification tasks in ML specifically [26]. Furthermore, the original CheXpert study also uses the AUC

Table 1: DenseNet-121 architecture for binary classification task on CheXpert data set.

Layer name	Output Size	Layer details
<i>Input</i>	224×224	
Convolution	112×112	7×7 conv, stride 2
Pooling	56×56	3×3 max pool, stride 2
Dense Block (1)	56×56	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 6$
Transitional Layer (1)	56×56	1×1 conv
	28×28	2×2 average pool, stride 2
Dense Block (2)	28×28	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 12$
Transitional Layer (2)	28×28	1×1 conv
	14×14	2×2 average pool, stride 2
Dense Block (3)	14×14	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 24$
Transitional Layer (3)	14×14	1×1 conv
	7×7	2×2 average pool, stride 2
Dense Block (4)	7×7	$[1 \times 1 \text{ conv } 3 \times 3 \text{ conv}] \times 16$
Classification Layer	1×1	7×7 average pool
	1	1D fully connected, softmax
<i>Output</i>	1	

score as a performance measure, allowing better comparability of results [19].

3.3.2 Neural network architecture. We used the DenseNet-121 [25], which is widely used in computer vision for our neural network architecture. In the original CheXpert study, Irvin et al. [19] compared different ML architectures and showed that DenseNet-121 also performed best on the CheXpert data set. We also compared it with the ResNet-34 architecture [27], whereby the DenseNet-121 slightly outperformed it on the data set. DenseNets consist of a combination of dense blocks and transitional layers. We used a pre-build DenseNet-121 from the open-source ML library PyTorch and adjusted the classification layer for the binary CheXpert classification task to implement the architecture. Table 1 shows the modified DenseNet-121 architecture.

3.3.3 Hyperparameters. We used an Adam optimizer [28], with the default β -parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The optimizer ensures that all parameters get updated iteratively and that the weights for the network get optimized. We kept the learning rate constant at 1×10^{-4} . These hyperparameters have shown to work well for CNNs on the CheXpert data set [19, 21]. In our evaluation of different batch sizes (i.e., 8, 32, 64, 96), a batch size of 16 performed best.

3.3.4 Training. During the training process, we evaluated the CNN's performance after each training epoch on the validation set according to the early stopping approach [29]. Thereby, we

Figure 2: Training and validation error curves. **Figure 3: ROC curve of the trained ML model evaluated on the validation set.**

calculated the error on the training set and the error on the validation set after every epoch; the results are plotted in Figure 2. The figure reveals that the training error decreases continuously, whereas the validation loss rises after three epochs, which is an indicator of overfitting. Consequently, the early stopping approach stopped the training procedure after three epochs when the validation loss reached its minimum. On the validation set, the trained model performs well with an AUC score of 0.93 (see Figure 3), which is similar to the score reached in the original CheXpert study on the same validation set (i.e., a range from 0.91 to 0.97) [19].

3.4 Implementation

We implemented three data valuation methods, one based on the LOO-method, the TMC-Shapley, and the KNN-Shapley using Python 3.8 and PyTorch 1.6.0. We computed the KNN-Shapley on deep features that we obtained from the CNN. Following Table 1, suitable deep features can be extracted as the output from Dense Block (4), Dense Block (3), Dense Block (2), and Dense Block (1).

Table 2: Distribution of the KNN-Shapley values of certain labels or scan types in deciles of the sampled training data set. For each label or scan type, the two lowest values and highest values are highlighted. Furthermore, the standard deviations are shown.

KNN-Shapley value decile	Labels				Frontal/Lateral	
	NaN	1.0	0.0	U	Frontal	Lateral
Total number of data instances	41,313 (100%)	32,779 (100%)	20,494 (100%)	5,414 (100%)	79,152 (100%)	20,848 (100%)
0-10% (low)	9.86%	10.15%	9.93%	10.44%	9.99%	10.05%
10-20%	10.05%	10.01%	9.87%	10.08%	9.97%	10.10%
20-30%	10.08%	9.88%	9.84%	10.66%	10.03%	9.88%
30-40%	10.07%	9.84%	10.05%	10.20%	10.03%	9.89%
40-50%	10.16%	9.71%	10.14%	10.01%	9.98%	10.09%
50-60%	10.02%	9.88%	10.28%	9.53%	10.04%	9.85%
60-70%	9.99%	10.13%	9.95%	9.44%	9.96%	10.14%
70-80%	9.93%	10.04%	10.02%	10.18%	10.00%	10.02%
80-90%	9.91%	10.27%	9.79%	9.84%	10.03%	9.90%
90-100% (high)	9.92%	10.08%	10.12%	9.62%	9.98%	10.09%
Standard dev.	0.09%	0.16%	0.14%	0.37%	0.03%	0.1%

In a first experiment, we computed the runtime of the different data valuation methods depending on the number of data instances. Using the KNN-Shapley as a scalable data valuation method, we afterward analyzed the distribution of the KNN-Shapley values and conducted a data instance removal experiment, as well as a detection-of-noisy-labels experiment, which is in detail described in the next section.

We run our experiments on a workstation running on Red Hat Enterprise Linux 7.7 with 4 Nvidia Tesla V100 graphic cards (32 GB of VRAM each); the CUDA toolkit version was 10.2. Further details on our experiments are also available in the source code [12].

4 Experiments and results

4.1 Runtime comparison

In the first experiment, we compared the runtime of the different data valuation methods. The experiment aimed to calculate the TMC-, KNN-Shapley, and LOO values for many different numbers of training data instances (i.e., 10, 100, 200, 400, 800, 1,000, 5,000, 10,000, 20,000, and 50,000 data instances). Thereby, we set the number of test data instances according to the 80:20-split between train and test data instances. The training and test data instances were randomly sampled from the training and test set. Due to the experiment's large computational resource consumption, we stopped calculations if they ran for longer than 10^5 seconds (ca. 28 hours). Figure 4 shows the results of the runtime comparison experiment.

The LOO values and TMC-Shapley both fail to value instances of larger medical image data sets (>1,000 instances)

based on these results. On the contrary, the KNN-Shapley approximation can value 50,000 training data instances in approximately 24 hours. Therefore, only the KNN-Shapley estimates will be considered in the subsequent experiments.

Figure 4: Runtime comparison of different data valuation methods.

4.2 Distribution of KNN-Shapley values

A means to further improve the computational efficiency of the KNN-Shapley calculations is to limit the test data size to 5% of the training data set size [9]. As such, we were able to compute the KNN-Shapley values of 100,000 training data instances on 5,000 test data instances within a computing time of approximately 21 hours.

Table 2 shows how the different KNN-Shapley values (from low value to high value) of different labels and scan types are

(a) Patient 22182, study 19, view 1 (b) Patient 14308, study 7, view 1

Figure 5: X-ray scans with a corresponding KNN-Shapley value within the lowest decile (a) or highest decile (b) for patients with the clinical observation of pleural effusion.

distributed on a decile scale of all KNN-Shapley values. Thereby, Table 2 reveals that there are instances for all labels and scan types that add valuable information and instances that do not add valuable information to the learning task. As a result, the SVs are relatively evenly distributed across labels and scan types. The instances with the highest standard deviation of 0.37% have an uncertainty label (i.e., pleural effusion is uncertainly present for the scan): 10.66% of the instances are within the low range of 20-30% of the overall SVs, whereas only 9.44% are within the high range of 60-70%. Furthermore, instances with an uncertainty label are more likely to be in the lower half of the SV distribution than in the upper half. Instances with a 1 label (i.e., the pleural effusion is present for the scan) cumulate at two extremes: 10.27% of the instances are within the very high range of 80-90%, and 10.15% are within the lowest decile, at a standard deviation of 0.16%. Instances of a 0 label (i.e., pleural effusion is confidently absent for the scan) have a standard deviation of 0.14%, and instances with a NaN label (i.e., no information about the clinical observation for the scan) the lowest standard deviation of 0.09%. The KNN-Shapley values for frontal scans are more evenly distributed with a standard deviation of 0.03% than the KNN-Shapley values for the lateral scans with a standard deviation of 0.1%.

Figure 5 shows two scans of different patients with the clinical observation of pleural effusion (1 label). It is clearly visible that the scan of the lowest KNN-Shapley value decile is of lower quality than the scan of the highest value decile.

4.3 Removal of data application

Besides the plain valuation of data instances, another application, which can also evaluate the SV calculations' quality, is the removal of data instances [8, 9]. The main idea is to remove data instances of certain values from the training data set and evaluate the trained model's performance on a test set. By doing so, it can be assessed how data instances of certain values influence the model performance (i.e., decrease, increase, or no effect).

To do so, we first ranked the 100,000 training data instances according to their value from high value to low value. Second,

Figure 6: Removing training data instances based on their KNN-Shapley values.

we removed either the most valuable or the least valuable training data instances from the training set. Finally, we evaluated the trained model's performance on a set of 10,000 test data instances that is disjoint from the set used for the calculation of the KNN-Shapley values. We repeated this procedure for 5% to 45% of the training data instances. The results are plotted in Figure 6. On the one side, Figure 6 reveals that the model's performance decreases significantly when removing training data instances of high value. On the other side, the model performance changes little when removing training data instances of low value. It is noteworthy that removing training data instances randomly results in similar model performances as removing training data instances of low value.

4.4 Detection of noisy labels

Another application of data valuation methods is prioritizing data instances for label verification [8, 9]. Even professionally-labeled medical data sets, such as the examined CheXpert data set, are prone to label errors with estimates of ca. 7.4% potentially false labels of implicit instances. This may impede the performance and explainability, and, ultimately, the trustworthiness of ML models applied in health care. However, verifying the label of each instance is often too expensive, especially when experts are required, such as in radiology [5]. Since mislabeled data instances add incorrect information to the learning task, they are likely to contribute little to the model's performance and, thus, have a low SV. Therefore, we evaluate the effectiveness of the SV for such an experiment.

First, we randomly sampled 10% of the data instances from the training data set and flipped their labels. Afterward, we trained the model again on the entire training data set and obtained the deep features. It is noteworthy that the model trained on the original data reaches an AUC score of 86.46%, whereas the model trained on the flipped data reaches an AUC score of 85.95% when evaluated on the test set. Then we calculated the KNN-Shapley values for the flipped data instances. Like in the

Figure 7: Detecting noisy labels with the help of KNN-Shapley values. The training data instances with low KNN-Shapley values were inspected first.

removal of data experiment, we calculated the KNN-Shapley for 100,000 training data instances on 5,000 test data instances, within a computation time of 22 hours, and ordered the instances from a low to a high value. The results of the detecting-noisy-labels-experiment are plotted in Figure 7. It shows that data instances of low value are more likely than data instances of high value to have an erroneous label. By prioritizing instances of low value, the verification process can be conducted more efficiently: 80% of the incorrect labels are among the 30% of data instances with the lowest KNN-Shapley values.

5 Discussion

5.1 Principal findings

This paper empirically evaluated different data valuation methods for machine learning on an extensive medical imaging data set. Our findings suggest that the KNN-Shapley method is computationally capable of handling larger data sets in the order of 100,000 scans on contemporary hardware infrastructures. In contrast, TMC-Shapley and LOO values fail to calculate data values in a reasonable time for contemporary medical ML data sets. These findings are in line with theory, which suggests that the KNN-Shapley is a computationally efficient data valuation method and reinforces the usefulness of the KNN-Shapley as a data valuation method for large ML data sets. Furthermore, as an SV-based method, the KNN-Shapley satisfies the properties of group rationality, fairness, and additivity.

Data instances from all labels and scan types can be highly beneficial or highly redundant for the ML task, even if there is an uncertainty label. Still, the outcomes from Table 2 indicate a weak correlation between the value of a training data instance

and its label. Remarkably, training data instances with a NaN label or an uncertainty label on average have lower KNN-Shapley values relative to the training data instances with other labels. This finding supports prior research, suggesting that the interpretation of a NaN label as a 0 label might be questionable in some cases, especially cases that may contain undetected diseases [5].

Based on our findings, it appears that the KNN-Shapley values do not only depend on the label of the training instance. Apart from a noisy label, there might be other factors, such as the quality of the image (i.e., the X-ray scan), that influence the training points' contribution toward the performance score.

With our research, we demonstrate three possible applications of the Shapley value for trustworthy ML in health care. First, the valuation of data instances, which showed that the KNN-Shapley is useful to incentivize the collection of extensive and diverse data sets of health data. Second, the removal of data experiment showed that training data instances of different values have very different effects on a trained model's prediction score. A remarkable result to emerge from the experiment is that training data instances of low value have little influence on the prediction score of a model, whereas training data instances of high value influence the prediction score of a model strongly. Therefore, the KNN-Shapley values can help discriminate 'good' from 'bad' training data instances. Recent research confirms this ability of the SV through a study with professional radiologists [11]. Third, the detecting-noisy-labels experiment showed that KNN-Shapley values could help prioritize the verification process of such noisy labels. Since an ML classifier with flipped labels results in a lower prediction score, the KNN-Shapley can help identify the data instances that cause the decline in performance score. By ranking the training data instances by their value and inspecting the low-value data instances first, a set of mislabeled instances can be detected more efficiently than if the data instances were inspected randomly, and experts (i.e., radiologists) can save valuable time.

5.2 Implications for research and practice

In this work, we demonstrate that the KNN-Shapley, a Shapley-based valuation method, can quantify individual data instances' contribution to the training of ML models on extensive, real-world medical imaging data sets. This can be helpful for future research on the collection of high-quality data for ML, for example, by designing trustworthy data sharing infrastructures for health care, such as data marketplaces or token economies based on distributed ledger technology [7]. With these, individuals can get rewarded for sharing data depending on the usefulness of their data, which can help to collect large and diverse data sets for ML [3, 7]. Our findings indicate that the SV as a data valuation measure can be applied and evaluated in real health care settings. The application in practice may prompt new questions, for example, on how to

map a (positive or negative) KNN-Shapley value of a data instance, fulfilling the equitable valuation scheme, to a real-world reward (e.g., a financial reward).

Furthermore, our empirical results suggest that data valuation methods could be taken advantage of to improve the quality of data sets. It has been shown that data sets can be summarized by removing the data instances of low value without significantly losing prediction accuracy. This is especially interesting in the context of information privacy, which is a core pillar of trustworthy ML for health care [3, 4]. In general, there is a trade-off between information privacy and data-driven health care. An SV-based method may indicate which data instances would be beneficial to share, and which instances may provide little value to a data set, and individual privacy concerns may outweigh the benefit of sharing these instances.

Additionally, data valuation methods can be used to prioritize the verification process of noisy labels. Recent research has suggested that even medical research data sets as the CheXpert data set, labeled with information from professional radiologists, contain erroneous labels and asked for ways to identify mislabeled instances ex-post [5]. Based on our experiment, the KNN-Shapley value appears to be a promising method for this application.

5.3 Limitations and future research

One limitation of our experiments is that we focus only on one observation, pleural effusion. Especially the label distribution of instances with a pleural effusion observation appeared to be promising for interesting experiments. We expect that the KNN-Shapley value is also capable of valuing instances with other data instances, which future research can emphasize on. Another limitation is that the data valuation schemes fail to provide details on the reasoning of the value of data instances. Although the KNN-Shapley reveals which of the data instances contribute little to the model's performance, it fails to reason why these data instances do so. Consequently, it is hard to tell how to improve the value of data instances. At the same time, it also fails to reason why data instances are beneficial to the ML algorithm. Future research could extend this in two ways. First, future research should advance the reasoning of the valuation process by combining research areas of feature selection and data valuation. Second, it should analyze data instances of a good or bad value in depth. For trustworthy ML, for example, it is desirable that data instances from underrepresented groups are not assigned low values [4], but that the overall incentive structure of potential data marketplaces favors diverse data sets [3].

The data valuation methods we analyzed are designed for fixed data sets (e.g., hospitals already share data instances and agree to evaluate instances). In some cases, it may also be interesting to value data instances without sharing these prior to the valuation, for example, to assess whether data sharing is

reasonable without impeding privacy. Evaluating methods from recent research on data valuation in the context of a distributional framework [30, 31] or federated learning [32] may, therefore, be interesting avenues for future research, as well.

Further exciting avenues for future research are a temporal value of data instances (e.g., data shared at the beginning of a pandemic such as COVID-19 may be more valuable than later on), the valuation of data instances in the context of differential privacy techniques, or the integration of data valuation frameworks into token economies for health care [7].

6 Conclusion

We empirically evaluate the LOO values, the TMC-Shapley, and the KNN-Shapley methods for data valuation on an extensive medical imaging data set with this research. Our results reveal that only the KNN-Shapley value is capable of valuing a large number of data instances within a reasonable timeframe. We then analyzed the distribution of the data values, which indicates that the KNN-Shapley can be useful to reward participants in data sharing infrastructures. Our data instance removal experiment confirms the successful valuation and demonstrates that another use case of the KNN-Shapley can be the summarization of data sets. Furthermore, we showed that the SV can effectively prioritize the analysis and detection of noisy labels. With our research, we contribute to the extant body of knowledge in three ways. First, we show that the SV approximation with a KNN classifier can successfully value data instances on an extensive medical imaging data set. Second, we publish our source code [12], allowing other researchers to reproduce our results and evaluate our methods in other contexts. Third, we illustrate useful applications where the SV can embrace the deployment of trustworthy ML in health care. These applications include data markets, data set summarization, and the detection of mislabeled instances. To the best of our knowledge, we are the first to analyze how to improve a patient's privacy by compressing data sets.

ACKNOWLEDGMENTS

We acknowledge support by the state of Baden-Württemberg for computing on the high-performance computing cluster bwHPC.

REFERENCES

- [1] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C. and Shpanskaya, K. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225v3* (2017).
- [2] Jin, C., Chen, W., Cao, Y., Xu, Z., Tan, Z., Zhang, X., Deng, L., Zheng, C., Zhou, J. and Shi, H. Development and evaluation of an artificial intelligence system for COVID-19 diagnosis. *Nature communications*, 11, 1 (2020), 1-14.
- [3] Thiebes, S., Lins, S. and Sunyaev, A. Trustworthy artificial intelligence. *Electronic Markets* (2020), 1-18.

- [4] Wiens, J., Saria, S., Sendak, M., Ghassemi, M., Liu, V. X., Doshi-Velez, F., Jung, K., Heller, K., Kale, D. and Saeed, M. Do no harm: a roadmap for responsible machine learning for health care. *Nature medicine*, 25, 9 (2019), 1337-1340.
- [5] Rädtsch, T., Eckhardt, S., Leiser, F., Pandl, K. D., Thiebes, S. and Sunyaev, A. What Your Radiologist Might be Missing: Using Machine Learning to Identify Mislabeled Instances of X-ray Images. In *Proceedings of the Proceedings of the 54th Hawaii International Conference on System Sciences (HICSS)* (2021).
- [6] Hynes, N., Dao, D., Yan, D., Cheng, R. and Song, D. A demonstration of sterling: a privacy-preserving data marketplace. *Proceedings of the VLDB Endowment*, 11, 12 (2018), 2086-2089.
- [7] Pandl, K. D., Thiebes, S., Schmidt-Kraepelin, M. and Sunyaev, A. On the Convergence of Artificial Intelligence and Distributed Ledger Technology: A Scoping Review and Future Research Agenda. *IEEE Access*, 8 (2020), 57075-57095.
- [8] Ghorbani, A. and Zou, J. *Data Shapley: Equitable Valuation of Data for Machine Learning*. International Conference on Machine Learning (2019).
- [9] Jia, R., Sun, X., Xu, J., Zhang, C., Li, B. and Song, D. An Empirical and Comparative Analysis of Data Valuation with Scalable Algorithms. *arXiv preprint arXiv:1911.07128v1* (2019).
- [10] Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gurel, N. M., Li, B., Zhang, C., Song, D. and Spanos, C. *Towards Efficient Data Valuation Based on the Shapley Value*. The 22nd International Conference on Artificial Intelligence and Statistics (pp. 1167-1176) (2019).
- [11] Tang, S., Ghorbani, A., Yamashita, R., Rehman, S., Dunnmon, J. A., Zou, J. and Rubin, D. L. Data Valuation for Medical Imaging Using Shapley Value: Application on A Large-scale Chest X-ray Dataset. *arXiv preprint arXiv:2010.08006v1* (2020).
- [12] <https://github.com/kpandl/Scalable-Data-Valuation-Health-Care-Shapley-Value>
- [13] Toneva, M., Sordani, A., Combes, R. T. d., Trischler, A., Bengio, Y. and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. *International Conference on Learning Representations 2019, arXiv preprint arXiv:1812.05159v3* (2018).
- [14] Yoon, J., Arik, S. and Pfister, T. *Data valuation using reinforcement learning*. International Conference on Machine Learning (2020).
- [15] Matsui, Y. and Matsui, T. NP-Completeness for Calculating Power Indices of Weighted Majority Games. *Theoretical Computer Science*, 263, 1-2 (2001), 305-310.
- [16] Maleki, S., Tran-Thanh, L., Hines, G., Rahwan, T. and Rogers, A. Bounding the Estimation Error of Sampling-based Shapley Value Approximation. *arXiv preprint arXiv:1306.4265v2* (2013).
- [17] Mitchell, T. M. *Machine Learning*. McGraw-Hill, New York (1997).
- [18] Jia, R., Dao, D., Wang, B., Hubis, F. A., Gurel, N. M., Li, B., Zhang, C., Spanos, C. and Song, D. Efficient task-specific data valuation for nearest neighbor algorithms. *Proc. VLDB Endow.*, 12, 11 (2019), 1610-1623.
- [19] Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Illcus, S., Chute, C., Marklund, H., Haghighi, B., Ball, R. and Shpanskaya, K. *CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison*. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019).
- [20] Xie, Y., Chen, M., Kao, D., Gao, G. and Chen, X. A. *CheXplain: Enabling Physicians to Explore and Understand Data-Driven, AI-Enabled Medical Imaging Analysis*. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (2020).
- [21] Bressen, K. K., Adams, L., Erxleben, C., Hamm, B., Niehues, S. and Vahldiek, J. Comparing Different Deep Learning Architectures for Classification of Chest Radiographs. *arXiv preprint arXiv:2002.08991v1* (2020).
- [22] Minaee, S., Kafieh, R., Sonka, M., Yazdani, S. and Soufi, G. J. Deep-Covid: Predicting Covid-19 from Chest X-ray Images Using Deep Transfer Learning. *arXiv preprint arXiv:2004.09363v3* (2020).
- [23] Cohen, J. P., Dao, L., Morrison, P., Roth, K., Bengio, Y., Shen, B., Abbasi, A., Hoshmand-Kochi, M., Ghassemi, M. and Li, H. Predicting Covid-19 Pneumonia Severity on Chest X-ray with Deep Learning. *arXiv preprint arXiv:2005.11856v3* (2020).
- [24] Porcel, J. M. and Light, R. W. Diagnostic Approach to Pleural Effusion in Adults. *American Family Physician*, 73, 7 (2006), 1211-1220.
- [25] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K. Q. *Densely Connected Convolutional Networks*. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017).
- [26] Fawcett, T. An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27, 8 (2006), 861-874.
- [27] He, K., Zhang, X., Ren, S. and Sun, J. *Deep Residual Learning for Image Recognition*. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016).
- [28] Kingma, D. P. and Ba, J. Adam: A method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980v9* (2014).
- [29] Prechelt, L. *Early Stopping-but When?* Neural Networks: Tricks of the trade, Berlin (1998).
- [30] Ghorbani, A., Kim, M. P. and Zou, J. *A Distributional Framework for Data Valuation*. International Conference on Machine Learning (2020).
- [31] Kwon, Y., Rivas, M. A. and Zou, J. Efficient computation and analysis of distributional Shapley values. *arXiv preprint arXiv:2007.01357v3* (2020).
- [32] Wang, T., Rausch, J., Zhang, C., Jia, R. and Song, D. *A Principled Approach to Data Valuation for Federated Learning*. Federated Learning: Privacy and Incentive, Lecture Notes in Computer Science, Springer International Publishing (2020).