# An Evolutionary Approach to Data Valuation

Natalia Khuri, Sapan Bhandari, Esteban Murillo Burford, Nathan P. Whitener and Konghao Zhao
Wake Forest University
Winston-Salem, North Carolina, USA
natalia.khuri@wfu.edu

## ABSTRACT

Data valuation in machine learning comprises computational methods for the estimation of the importance of individual training instances. It has been used to remove noise, uncover biases, and improve the accuracy of trained models. Current data valuation techniques do not scale up for large datasets and do not work for regression tasks, where the objective is to predict a numerical outcome rather than a small number of nominal class labels. In this work, an evolutionary approach for qualitative and quantitative data valuation, is presented. The proposed approach is tested on regression and classification benchmarks, and on several bioinformatics and health informatics datasets. In addition, models trained with most valuable subsets of data are validated on independently acquired tests, demonstrating the generalizability as well as the practical utility of the proposed approach.

## CCS CONCEPTS

• **Computing methodologies → Genetic algorithms**; • **Applied computing → Bioinformatics**.

## KEYWORDS

data valuation, classification, genetic algorithms, machine learning, regression

**ACM Reference Format:**
Natalia Khuri, Sapan Bhandari, Esteban Murillo Burford, Nathan P. Whitener and Konghao Zhao. 2022. An Evolutionary Approach to Data Valuation. In *13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '22), August 7–10, 2022, Northbrook, IL, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3535508.3545522

## 1 INTRODUCTION

Supervised and unsupervised machine learning (ML) are routinely used in bioinformatics and health informatics. There are three required components for building a successful ML system, namely, a database of prior knowledge, a robust learning process, and an effective inference function [20]. While the vast majority of past research focused on improvements to the learning processes, issues related to data quality are often cited among the reasons for the lagging real-world performance of ML systems [13, 37].

Some of the quality issues may be due to the selection, capture, labeling or co-occurrence biases. Because, an automated detection of biases and their types is challenging, a common mitigation strategy is to collect larger datasets, engineer better features from existing data, or integrate data from multiple samples, experiments and measurements.

Unfortunately, an increase in data must be substantial due to the logarithmic dependency between the size of the training data and model's performance [37]. In addition, new data may contain the same biases as the original datasets, thus, bringing no additional value to the learning process. Sophisticated feature engineering relies on expert knowledge and manual labor, and can exacerbate the "curse of the dimensionality" problem, often encountered in bioinformatics and health informatics [5]. Data integration is a grand computational challenge on its own, in which additional data biases can arise due to batch effects [28]. Lastly, training with massive datasets incurs significant burdens on computational resources and on energy usage [35].

On the other hand, there is a growing evidence that training with smaller datasets may be as effective as training with full-sized datasets [6, 17, 22–24, 36]. The selection of the smaller data subsets that are valuable for training of an accurate model can be done automatically. In data valuation, for example, computational methods are used to estimate the importance of each training instance. Low-valued instances can be filtered out to find representative data, remove noise, reduce biases and improve real-word performance. These methods include, for example, leave-one-out validation (LOO) [33], influence functions [25] and Shapley data values [17].

The utility of data valuation in bioinformatics and health informatics remains limited due to several shortcomings of existing methods. First, computational costs are high, in particular when precise estimates are computed, such as in Shapley data valuation, for example. Second, existing techniques work for classification tasks only, where the outcomes are discrete. While classification tasks are important, many practical ML systems must be also able to predict numerical outcomes in regression tasks. Third, because data valuation produces a ranked list of the estimated values, the decision about the best threshold between high and low values must be made by the user. This can lead to inconsistencies and new biases. Finally, it has been noted that the valuation process is dependent on the choice of the learning algorithm and the choice of the performance metric. This means that the costly process of data valuation must be repeated for the different combinations of learning algorithms and metrics.

In this work, we address these shortcomings and introduce an evolutionary approach for data valuation, which is suitable for regression and classification tasks with grouped and ungrouped data. Starting with a random subset of the data, we use a Genetic

*BCB '22, August 7–10, 2022, Northbrook, IL, USA*

ACM ISBN 978-1-4503-9386-7/22/08…$15.00
https://doi.org/10.1145/3535508.3545522

Algorithm (GA) to iteratively improve the solution by applying genetic operators. The search for the best subset is guided by the fitness function that evaluates the goodness-of-fit of the trained model. Independently repeated GA runs produce an ensemble of solutions, which can be either analyzed individually or used to compute quantitative estimates of the values of the training instances. As the result, we made three main contributions.

First, we designed and implemented an evolutionary approach to automatically select the most valuable data subsets for regression or classification tasks. Our proposed approach can select subsets from as few as eight and as many as thousands of training instances. In addition to selecting most valuable instances, the proposed method can successfully select most valuable groups of data.

Second, we systematically tested our approach on standard regression and classification benchmarks, on real-world datasets with independent tests, and on synthetic and experimental large, high-dimensional transcriptomics data. In all use-cases, we demonstrated that subset-based training led to models that performed on par or better than models trained with the original datasets.

Third, we showed that by computing an ensemble of evolutionary subsets, data values of individual training instances can be derived and used to interpret the latent patterns in the training data or to explain model's decisions.

The remainder of this work is organized as follows. In Section 2, prior research is reviewed. We describe our evolutionary approach and datasets in Section 3. The results of computational experiments are presented and analyzed in Section 4. Finally, we summarize our findings and propose future extensions of this work in Section 5.

## 2 PRIOR AND RELEVANT WORK

Two groups of computational methods are relevant to our work, namely, coreset selection and data valuation.

In computational geometry, a coreset is defined as a minimal set of points that approximates the shape of a larger set of points. In ML, a coreset is a subset of the original data, which guarantees that models trained on them will be a good fit for the original dataset [32]. Many of the early coreset discovery methods used techniques for dimensionality reduction, where the goal is to project a high-dimensional dataset onto a low-dimensional space [15, 31].

Recognizing that models trained with a coreset should also generalize to unseen data, subsequent methods redefined coresets as the smallest subsets of the original datasets that are used to train predictive models [4, 8, 12]. In the work closest to ours, a multi-objective evolutionary optimization algorithm has been proposed, which finds a Pareto front of possible coresets [4]. These coresets represent the best trade-offs between the coreset sizes and the classification errors. On four well-known benchmarks ranging in size from 150 to 400 instances, the algorithm outperformed traditional subset selection methods. The evaluation was performed using a random split of the data into training (66%) and test (33%) sets. In all benchmarks, reported accuracy was around 90%, significantly higher than the accuracy of the alternative methods.

In addition to data summarization, coresets can provide qualitative insights about the values of the training instances. Namely, the discarded instances are of low value, while the selected instances are highly valuable to the learning process. One of the main limitations of existing coreset selection methods is that there is no ranking of the importance of the training instances.

This limitation is addressed by the methods for data valuation. The main idea of data valuation is to assign a numerical score to each training instance in the original dataset, based on the estimated contribution of that instance to the overall performance of the trained model. Once estimated, low-valued training instances may be removed without negative consequences to model's predictive performance. Conversely, when high-valued instances are removed, model's performance decreases. Methods for quantitative data valuation include LOO validation, influence functions, Shapley data values and reinforcement learning.

In LOO validation, for example, a classifier is repeatedly trained using all instances except for one [33]. The difference in the performance of a model trained with all data of size $N$, and a model trained with data of size $N - 1$, serves as a proxy for the value of the withheld sample.

Influence functions compute how the parameters of a model change when the weight of a single training instance is increased by a very small amount, thus, identifying training instances that are most valuable for the predictor [25]. While the original statistical influence functions do not scale up to real-world data, their computation can be somewhat improved using the second-order optimization techniques.

Data Shapley and its derivatives are inspired by the problem of the fair division of profit in collaborative game theory. By representing the learning process as a collaborative game and the accuracy of predictions as a profit to be divided fairly among all training instances, Shapley values can be computed. For a single training instance, its marginal contribution to all possible subsets of training data must be computed, thus, making this problem intractable. Therefore, data valuation methods use heuristics, such as the Monte Carlo simulation, gradient-based estimation and Locality Sensitive Hashing [9, 17, 19, 30]. Among the most recent techniques for computing data values is the reinforcement learning approach, which uses, as the reinforcement signal, the profit that is obtained on a small validation set [40]. Once low-valued training instances have been identified, they may be removed from the training data without a significant loss in model's performance. The decision about which threshold to use for filtering of the training data, must be made by the user, however.

Data valuation techniques have been successful in noise detection [17, 25], improved classification of X-ray images [36], automatic selection of subjects from the Alzheimer's disease database [6], aggregation of sensor data [22] and virtual drug screening [23]. However, they remain impractical for real-world applications due to their computational complexity, dependency on user-defined thresholds, and a lack of support for regression tasks.

Here, we propose a different approach that can automatically detect most valuable training instances and estimate the quantitative value of each instance in the original dataset. In what follows, we describe the proposed evolutionary approach to data valuation with a focus on applications in bioinformatics and health informatics. Moreover, the proposed approach is applicable to different domains and we rigorously test it with diverse and well-established ML benchmarks.

# 3 METHODS AND DATA

## 3.1 Proposed Data Valuation Approach

Given a dataset of size $N$, the objective is to find $v = v_1, \ldots, v_N$, such that each $v_i$ estimates the value of the corresponding datum $i$.

We implement a GA to solve this problem. Specifically, we begin by initializing $v$ with zeros. Next, we generate the initial population of data subsets and evolve a GA to find the best subset comprising $S$ most valuable instances of $N$. Once GA converges or reaches its predefined maximum number of iterations, we increment $v_i$ by 1, for each instance $i$ that is found in $S$. We repeat the GA $k$ times and divide each $v_i$ by $k$. Thus, if a training instance was not chosen in any of the $k$ runs, its value will be 0. On the other hand, if an instance participated in every final solution, its $v_i$ value will be 1. In this work, we set $k$ to be 100.

Next, we describe the GA and its operators.

## 3.2 Genetic Algorithm

GAs are inspired by the process of natural selection in evolution, where the fittest individuals produce offspring of the next generation [18]. They are used as heuristics for solving the optimization problems. The design of a GA involves choosing an encoding, a fitness function and genetic operators.

In this work, each chromosome is encoded as a binary vector of size $N$, the number of training samples. The elements of each vector are drawn from [0,1] with uniform probability. Thus, 1 in position $i$ means that the $i^{th}$ instance is included in the solution, and 0 in position $i$ means that the $i^{th}$ instance is excluded, respectively.

We begin by creating a population pool of $P$ chromosomes and evolve the GA until convergence or for a predefined number of iterations. At each iteration, we perform the selection process. First, we transfer 1% of the best solutions to the next generation. Next, $p$ chromosomes are randomly drawn from the population pool. Their fitness is computed and normalized, such that the resulting sum of all fitness values is equal to 1. Pairs of chromosomes are selected from $p$, for recombination. To pick pairs of chromosomes, we use a roulette-wheel selection, where the probability of selection is inversely proportional to the fitness of a chromosome. We repeatedly select pairs of chromosomes, until a new population of size $P$ is generated.

The purpose of the recombination is to explore the solution space of the problem. In this work, we perform the recombination by using a crossover operator, which swaps portions of the parent solutions to produce the offspring. In addition, we use a mutation operator to diversify the chromosomes from one generation to the next, by flipping a randomly selected position in a chromosome, with a probability of 0.01.

The fitness function depends on the predictive task, namely regression or classification. In both tasks, the fitness of a chromosome measures how well a model trained on $S$ instances fits these same instances (self-validation). In classification tasks, we use the multi-class Cohen's Kappa metric, and in regression tasks, we use the R-squared metric.

Cohen's Kappa ($\kappa$) measures the observed accuracy compared to the expected accuracy, which can be computed for $K$ classes as follows.

$$\kappa = \frac{c \times s - \sum_k^K p_k \times t_k}{s^2 - \sum_k^K p_k \times t_k} \tag{1}$$

Here, $c$ is the total number of correctly predicted instances, $s$ is the total number of instances, $p_k$ is the number of times that class $k$ was predicted, and $t_k$ is the number of times that class $k$ truly occurs. Kappa scores range from [-1, 1], and positive values indicate that the actual performance is greater than expected. A Kappa score equal to 0 means that model's performance is similar to a random classification, and a negative Kappa value represents performance worse than what would be expected by chance.

We use the ML definition of the R-squared metric ($R^2$), which is computed as follows.

$$R^2 = 1 - \frac{\sum |y - \hat{y}|^2}{\sum |y - \bar{y}|^2} \tag{2}$$

Here, $R^2 = 0$ implies that the regression model always predicts the expected value for the dependent variable, $\hat{y}$. Notably, $R^2$ can be negative, implying that model's performance is worse than random.

Because Kappa and $R^2$ cannot be computed for fewer than two samples, we automatically set the fitness score to $-1$ when the size of $S$ is 2. Also, GA minimizes the fitness function and, therefore, we multiply the computed fitness values by $-1$.

## 3.3 Datasets

We use multiple datasets to test our evolutionary approach, including standard ML benchmarks, datasets used in prior works, synthetic data and real-world datasets, retrieved from public repositories. The diversity and complexity of these datasets allows us to rigorously test our proposed approach in different domains, including bioinformatics and health informatics.

*Standard machine learning benchmarks.* Twenty three regression benchmarks and twenty four classification benchmarks were downloaded from the KEEL repository [3]. We downloaded all benchmarks with real-valued attributes and no missing values. All benchmarks are pre-partitioned into five-folds, and the outcome attribute is identified by the output field in the file header. The number of attributes in regression benchmarks ranges from 2 to 31, and the number of instances ranges from 43 to 40,768. Similar criteria were applied to download the classification benchmarks. The number of attributes in classification benchmarks ranges between 2 and 90, the number of output classes is between 2 and 15, and the number of instances is between 106 and 19,020.

*Drug datasets.* The second dataset comprises results of high-throughput screening data. Drug libraries were tested in *in vitro* inhibition assays against three human liver transporters. These datasets were selected because they had been used in prior work on data valuation [23]. Each dataset comprises two different screens against one of the transporters [2, 10, 14, 21]. OCT1 training data has 1,718 instances and 38 attributes, and test data has 188 instances and 38 attributes. Both, OATP1B1 and OATP1B3, have 224 training instances and 1,770 test instances, respectively, and the numbers of attributes in OATP1B1 and OATP1B3 datasets are 27 and 31, respectively. The regression task is to predict drug's activity, measured as percent inhibition.

*Cycling dataset.* The third dataset comprises 1,936 sensor measurements of fitness workouts. These data are grouped by athlete. The measurements have been collected by six different cyclists, four professional athletes [16], one amateur [22] and one of an unknown status [34]. Additionally, two independent test datasets, with 24 and 26 workouts respectively, were acquired by the amateur athlete, at two different time points. The regression task is to predict the average cycling power using 13 attributes.

*Parkinson's dataset.* The fourth dataset also comprises grouped data from 42 participants in a six-month trial of a telemonitoring device [38]. All participants were diagnosed with an early-stage Parkinson's disease, and the dataset contains multiple biomedical voice measurements, which were captured in patients' homes. We use linearly-interpolated clinician's Unified Parkinson's Disease Rating Scale (UPDRS) as the outcome attribute in the regression task. There are 16 attributes and 5,867 instances in this dataset.

*Single cell RNA sequencing datasets.* We included two types of single-cell RNA-sequencing (scRNA-seq) datasets to evaluate the utility of our approach by applying to very large and high-dimensional grouped data.

First, eight datasets were simulated using the Splatter package (version 1.18.1) [41]. To simulate benchmarks SIM1 to SIM4, we set the total number of cells to 50,000 and the number of genes to 720. We varied the number of samples from 5 to 25, keeping the number of cell types in each batch at 2. These datasets are representative of the scRNA-seq experiments, where samples are collected at different time-points, in different laboratories or using different sequencing platforms, for example. Datasets SIM5 to SIM7 comprise 5 samples each. The number of cell types varies from 4 to 16, and the total numbers of cells range from 16,000 to 64,000. To create SIM8 dataset, we used the experimental dataset EXP1 as the template and generated 19 different samples. The total number of cells in SIM8 is 50,220 and the number of cell types is 5.

To create the experimental dataset EXP1, we queried Gene Expression Omnibus (GEO) at the National Library of Medicine, for the recently reported studies of the COVID-19 disease, comprising multiple samples. EXP1 was constructed from samples of an immunophenotyping study of human donors, available as GSE149689 series [27]. Individual samples of each study were downloaded, preprocessed and integrated. Samples were labeled using the metadata that was provided by the depositors. We assigned to each sample, a distinct class label: healthy, influenza, asymptomatic, mild or severe COVID-19.

## 3.4 Data Preprocessing Workflows

KEEL benchmarks, Drug, Cycling and Parkinson's datasets are normalized and scaled to the range [-1, 1].

To preprocess the scRNA-seq data, we use the standard workflow and implement it with the help of the scanpy package (version 1.8.2) [39]. For each dataset, we aggregate samples into one matrix, using the intersection of the gene names. That means that we only retain the expression counts of gene transcripts that are common to all samples. Next, transcripts detected in fewer than 3 cells are filtered out, and expression counts are normalized. Transcript counts are normalized by dividing the raw measurements of each cell by the total gene expression and multiplying the result by a scale factor of 10,000. Next, normalized gene expressions are transformed using a logarithmic function. We reduce the dimensions of the matrix to highly variable genes only, and scale their log-normalized expressions, such that the mean expression across cells is 0 and the variance is 1.

Finally, the dimension of the aggregated normalized count matrices are further reduced to 10 harmonized principal components. These components are found by performing the principal component analysis, followed by the data harmonization using the Harmony package [26].

## 3.5 Regression and Classification Methods

To compute the fitness, a regression or a classification model must be fitted to the subset of the data encoded in a chromosome. We use multiple linear regression (MLR) in regression tasks, and logistic regression (LR) or feed forward neural network (FFNN) in classification tasks.

*Multiple linear regression.* We decided to use MLR because it is fast to train and test. In regression, the objective is to find a regression function, $f(x) = y$, which maps the attributes $x = x_1, \ldots, x_M$ to the numerical output $y$, where $M$ is the number of attributes. Assuming a linear relationship between $y$ and $x$, the regression function can be expressed as $f(x) = b_0 + \sum_{j=1}^{M} x_j b_j$, where $b_0$ is a bias term. Therefore, we want to find the coefficients $b$, given a large number of attributes $x$ and output values $y$, that minimize the prediction error over the given data set of size $N$.

$$\hat{b} = \underset{b}{argmin} \sum_{i=1}^{N} (y_i - f(x_i))^2 = \underset{b}{argmin} \sum_{i=1}^{N} \left( y_i - \sum_{j=1}^{M} x_{ij} b_j \right)^2 \quad (3)$$

*Logistic regression.* In classification experiments, we use logistic regression because it has been successful in prior works on Shapley data valuation [17], and because it is fast to train and test. The logistic regression uses a logistic function to output a nominal rather than a numeric value.

*Feed forward neural network.* For the classification of cell types in scRNA-seq datasets, we implement a simple, yet effective FFNN. We chose the FFNN classifier because it outperformed alternative classification techniques in prior works [29]. Our FFNN architecture comprises one hidden layer with 8 hidden nodes. The input layer has 10 nodes, which accept 10 harmonized principal components of cells. The output layer consists of $K$ nodes, one for each of the $K$ classes in a dataset. All nodes of the input layer connect to all hidden nodes, and all hidden nodes connect to all output nodes, forming a dense network. The information in the FFNN flows in the forward direction only.

In the hidden layer, we use the ReLU activation functions and L2 regularization penalty with a factor of 0.01. In the output layer, we the softmax activation function, and each output node emits predicted probabilities for each of the $K$ classes. Predictions are made using the argmax function. The network is trained using batches of 128 cells and optimized using RMSprop optimizer with a learning rate of 1E-4. Finally, categorical cross-entropy loss function is used to tune the parameters.

### 3.6 Implementation

All code is written in the Python programming language. We use Keras Application Programming Interface [11], sklearn [7] and Tensorflow [1] libraries. All experiments are performed on a High Performance Computing cluster.

## 4 RESULTS

*Result 1: Evolutionary subset selection performs well on standard ML regression and classification benchmarks.* We first show that a single GA run can detect a valuable subset $S$ from the original dataset of size $N$. To that end, we tested the proposed approach on classical regression and classification benchmarks from KEEL database. Each benchmark was pre-partitioned into five folds. We applied the evolutionary data valuation to one fold, selected the best subset from that fold, and tested it on the remaining four folds. The reasoning behind this is as follows. While it is common in ML to use four folds to tune a model, in real-world applications, training set is significantly smaller than the test set. We repeated the process for each of the five folds, and averaged the results.

In regression task, models trained with the subsets performed as well as the models trained with the original data. The Root Mean Square Error (RMSE) decreased or remained unchanged in 21 out of 23 benchmarks (Table 1). The Pearson's correlation between RMSEs was 0.99. In two regression benchmarks, baseball and california, the performance of the subsets' models degraded. The RMSE scores increased from 820.05 to 828.81 in the baseball benchmark when the fold size was reduced by about 5%, on average. In addition, the RMSE score changed from 69747.52 to 69855.35 in california benchmark, when about 42% of the data were discarded. Overall, the average size of regression benchmarks was reduced by between 2.32% (dee) to 44.28% (mv), with an average reduction of 20.38% across all benchmarks. There was a clear dependency between the average fold size and the number of discarded instances, and larger folds were reduced by a larger percentage.

The accuracy of classification, measured by Kappa, changed significantly when models were trained with subsets (paired t-test p-value=9.56154E-08). Interestingly, there was no linear dependency between the sizes of the original dataset and their subsets. The scores were moderately and positively correlated (Pearson's correlation of 0.55). On average, sizes of classification benchmarks were reduced by 51%, while Kappa scores increased from 0.58 to 0.92 (Table 2). Despite the noticeable increase in performance, subsets' models of banana and winequality-white remained unreliable. Their Kappa scores were below 0.6. Banana benchmark is one of the larger ones with 2 classes, whereas winequality-white benchmark has 11 classes.

*Result 2: Performance can be successfully replicated in independent tests with high-throughput screening data.* Having shown that the proposed method can reduce the size of the training data while maintaining and even improving models' performance, we asked if subset-based models were generalizable to unseen datasets. We applied our evolutionary approach to three paired data sets of drug screening experiments. Specifically, for each pair, we selected most valuable subsets of drugs in one of the datasets, such as OCT1-1, OATP1B1-1, and OATP1B3-1. Next, MLR models were trained using the original data or using the subsets, and models were tested on the independent datasets, OCT1-2, OATP1B1-2, OATP1B3-2, respectively.

**Table 1: Performance in Regression Tasks. Shown are KEEL benchmark names, average number of training instances and average RMSE scores of models trained with the original dataset or with the subset of data. Reduction in the size is reported as percent change.**

| | Original | | Subset | | |
|---|---|---|---|---|---|
| **Benchmark** | **Size** | **RMSE** | **Size** | **RMSE** | **Change (%)** |
| **ANACALT** | 810.40 | 0.42 | 547.89 | 0.42 | 32.39 |
| **abalone** | 835.40 | 2.23 | 560.49 | 2.24 | 32.91 |
| **autoMPG6** | 78.40 | 3.59 | 76.22 | 3.59 | 2.78 |
| **autoMPG8** | 78.40 | 3.50 | 76.48 | 3.51 | 2.45 |
| **baseball** | 67.40 | 820.05 | 63.84 | 828.81 | 5.28 |
| **california** | 4128.00 | 69747.52 | 2391.14 | 69855.35 | 42.08 |
| **concrete** | 206.00 | 10.70 | 171.36 | 10.78 | 16.82 |
| **dee** | 73.00 | 0.43 | 71.31 | 0.43 | 2.32 |
| **delta_ail** | 1425.80 | 0.00 | 907.22 | 0.00 | 36.37 |
| **delta_elv** | 1903.40 | 0.00 | 1175.69 | 0.00 | 38.23 |
| **diabetes** | 8.60 | 0.76 | 7.13 | 0.79 | 17.09 |
| **ele** | 99.00 | 661.78 | 93.26 | 662.54 | 5.80 |
| **ele-2** | 211.20 | 166.45 | 177.42 | 167.06 | 15.99 |
| **friedman** | 240.00 | 2.73 | 195.98 | 2.74 | 18.34 |
| **laser** | 198.60 | 23.99 | 166.32 | 24.16 | 16.25 |
| **machineCPU** | 41.80 | 82.31 | 40.52 | 82.72 | 3.06 |
| **mortgage** | 209.80 | 0.13 | 175.92 | 0.14 | 16.15 |
| **mv** | 8153.60 | 4.49 | 4543.03 | 4.50 | 44.28 |
| **plastic** | 330.00 | 1.54 | 255.00 | 1.54 | 22.73 |
| **puma32h** | 1638.40 | 0.03 | 1010.83 | 0.03 | 38.30 |
| **treasury** | 209.80 | 0.26 | 175.60 | 0.26 | 16.30 |
| **wankara** | 321.80 | 1.60 | 250.83 | 1.61 | 22.05 |
| **wizmir** | 292.20 | 1.29 | 231.31 | 1.30 | 20.84 |

Our results show that the original data may be reduced by 47% for OCT1-1, 32% for OAT1B1-1 and 25% for OATP1B3-1, without significant changes in the RMSEs of the independent tests (Table 3). In the case of OCT1-1, RMSE improved from 28.18 to 26.88 when low-valued drugs were removed from the training data. The RMSE of OATP1B1-1 dropped from 58.45 to 57.27, and the RMSE of OATP1B3-1 decreased from 39.07 to 37.71. Our results agree qualitatively with those of the previously reported data valuation on the same datasets. In prior work, approximately 30% of data were removed from OATP1B1-1 and OCT1-1, and 10% of the original data were removed from OATP1B3-1, without a decrease in performance [23].

Next, we repeated GA runs 100 times and examined an ensemble of evolutionary subsets detected in the Drug datasets. Each of the 100 subsets was tested on independent tests. We analyzed the distributions of subsets' sizes and their RMSE scores. On average, OCT1-1 could be reduced from 1,718 to 956.47, while keeping the average RMSE around 28.11, lower than the original score (Figure 1A and D). The average RMSE of OATP1B1-1 subsets was 58.72, on par with the model trained using all of the data, and the average size of the subsets was around 159.86 (Figure 1B and E). Finally, a

**Table 2: Performance in Classification Tasks. Shown are KEEL benchmark names, average number of training instances and average Kappa scores of models trained with the original dataset or with the subset of data. Reduction in the size is reported as percent change.**

| | | Original | | Subset | | |
|---|---|---|---|---|---|---|
| **Benchmark** | **Classes** | **Size** | $\kappa$ | **Size** | $\kappa$ | **Change (%)** |
| **appendicitis** | **2** | 21.20 | 0.48 | 10.40 | 1.00 | 50.94 |
| **balance** | **3** | 125.00 | 0.77 | 58.80 | 1.00 | 52.96 |
| **banana** | **2** | 1060.00 | 0.16 | 505.40 | 0.40 | 52.32 |
| **cleveland** | **5** | 59.40 | 0.24 | 29.20 | 1.00 | 50.84 |
| **ecoli** | **8** | 67.20 | 0.39 | 30.00 | 1.00 | 55.36 |
| **glass** | **7** | 42.80 | 0.39 | 16.00 | 1.00 | 62.62 |
| **iris** | **3** | 30.00 | 0.89 | 14.60 | 1.00 | 51.33 |
| **led7digit** | **10** | 100.00 | 0.68 | 42.00 | 1.00 | 58.00 |
| **magic** | **2** | 3804.00 | 0.52 | 1844.20 | 0.71 | 51.52 |
| **movement_libras** | **15** | 72.00 | 0.58 | 38.40 | 1.00 | 46.67 |
| **phoneme** | **2** | 1080.80 | 0.35 | 506.00 | 0.80 | 53.18 |
| **pima** | **2** | 153.40 | 0.44 | 93.80 | 1.00 | 38.85 |
| **ring** | **2** | 1480.00 | 0.51 | 714.40 | 0.77 | 51.73 |
| **segment** | **7** | 461.80 | 0.90 | 272.00 | 1.00 | 41.10 |
| **sonar** | **2** | 41.60 | 0.44 | 23.20 | 1.00 | 44.23 |
| **spambase** | **2** | 919.40 | 0.82 | 492.00 | 1.00 | 46.49 |
| **texture** | **11** | 1100.00 | 0.99 | 551.80 | 1.00 | 49.84 |
| **titanic** | **2** | 440.20 | 0.44 | 207.40 | 0.99 | 52.89 |
| **twonorm** | **2** | 1480.00 | 0.95 | 751.80 | 1.00 | 49.20 |
| **wdbc** | **2** | 113.80 | 0.93 | 55.40 | 1.00 | 51.32 |
| **wine** | **3** | 35.60 | 0.95 | 16.40 | 1.00 | 53.93 |
| **winequality-red** | **11** | 319.80 | 0.31 | 154.40 | 0.85 | 51.72 |
| **winequality-white** | **11** | 979.60 | 0.24 | 468.00 | 0.55 | 52.23 |
| **yeast** | **10** | 296.80 | 0.44 | 134.20 | 0.92 | 54.78 |

**Table 3: Performance in Virtual Drug Screening. For each paired dataset, shown are its name, original and reduced size, and the RMSEs of the models trained using the original dataset or models trained with subsets of data.**

| | Original | | Subset | |
|---|---|---|---|---|
| **Name** | **Size** | **RMSE** | **Size** | **RMSE** |
| **OCT1-1** | 1718 | 28.18 | 914 | 26.88 |
| **OATP1B1-1** | 224 | 58.45 | 152 | 57.27 |
| **OATP1B3-1** | 224 | 39.07 | 169 | 37.71 |

wider spread was observed in OATP1B3-1 solutions (Figure 1C and F). Although the average RMSE score was 40.95, about half of the subsets had RMSE scores lower than the original RMSE score, and the average subset size of OATP1B3-1 was 159.43.

*Result 3: Evolutionary subsets perform well irrespective of the training algorithm.* One of the limitations of prior works is that highly valuable data selected by one algorithm may not be useful for retraining by a different algorithm [4, 17]. Therefore, we examined the performance of evolutionary-based subsets in training with a different algorithm. Random forest (RF) regression models were fitted using each of the 100 subsets of OCT1-1, OATP1B1-1 and OATP1B3, respectively, and predictions were made for the corresponding independent tests.

Our results show that for all of the three paired datasets, subset-based training is similar or better than training with the original data, irrespective of the algorithm. Interestingly, RF models had better performance than the corresponding MLR models (Figure 1, G-I). Specifically, the average RMSEs on the independent tests were 24.10 for OCT1-2, 55.66 for OATP1B1-2, and 36.30 for OATP1B3-2, respectively. We also note that the RF models trained using the original data outperformed MLR models trained using the original data. The practical application of these results is that evolutionary subsets can be first selected using a fast algorithm, such as MLR, and then used to train a final model with a computationally slower technique.

*Result 4: External validation and subset-size penalty improve generalizability of the evolutionary subsets.* As an alternative to multi-objective optimization algorithm [4], we designed and tested two modifications to our fitness function and its evaluation. We tested these modifications on the Cycling dataset, which comprises grouped training data and two independent test datasets.

First, we studied the importance of the external validation in the evaluation of fitness. The motivation behind this is that subsets must be representative of the original data and be generalizable to unseen data. Our original fitness function, $F1$, computes performance in self-validation (Section 3). In self-validation, training and validation subsets are identical. We designed an alternative fitness function, $F2$ which computes the goodness-of-fit, such as Kappa or $R^2$, using an external validation set.

Second, to determine whether models trained with subsets are prone to overfitting, we designed a third fitness function, $F3$, comprising the original goodness-of-fit score and a penalty term. The second term of $F3$ penalizes smaller subsets and rewards larger ones, to avoid memorization and improve the generality of the trained model. We compute the second term as $\frac{(|N|-|S|)}{|N|}$, where $|S|$ is the size of the subset and $|N|$ is the size of the original training dataset. Therefore, the penalty is 0 when all training instances are selected, and 1 when none of the training instances had been selected.

Finally, we combined both external validation and the penalty term in the fourth fitness function, $F4$.

There are different methodologies for the construction of an external validation set. Here, we split the original data into train and validation folds as follows. We create a balanced training fold by selecting 100 instances from each athlete, randomly. All remaining instances are set aside as the external validation set, which is imbalanced in the number of the per-athlete instances.

We tested the subsets obtained by each fitness function in two independent tests. As expected, there was a significant difference in the subsets' sizes, when the penalty term was added to the fitness function. Specifically, the average subset size ranged from 269.06 for $F1$ to 418.67 for $F4$ (Figure 2A). There was also a slight increase when fitness was evaluated in external validation. The average subset size increased by about 20 instances between $F1$ and $F2$, and by 45 instances between $F3$ and $F4$, respectively.

The RMSE scores of the independent tests were 33.26 and 47.79, when training was done with the entire dataset of 1,936 instances. On average, similar RMSE scores were observed by training MLR with much smaller subsets of the data. Specifically, average RMSE scores on independent test 1 were 40.55 for $F1$, 34.39 for $F2$, 37.70

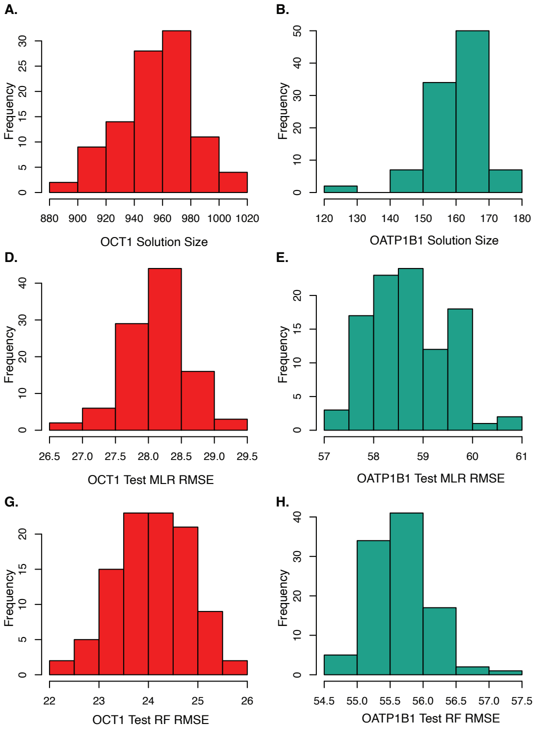

**Figure 1: Distribution of Subsets' Sizes and RMSEs in Drug Tests. Shown are barplots for three paired datasets, OCT1 (left column, red), OATP1B1 (middle column, green), and OATP1B3 (right column, yellow). A. to C. Top row shows size distributions of 100 GA runs. D. to F. Middle row shows RMSE scores of MLR models in independent tests. G. to I. Bottom row shows RMSE scores of RF models in independent tests.**

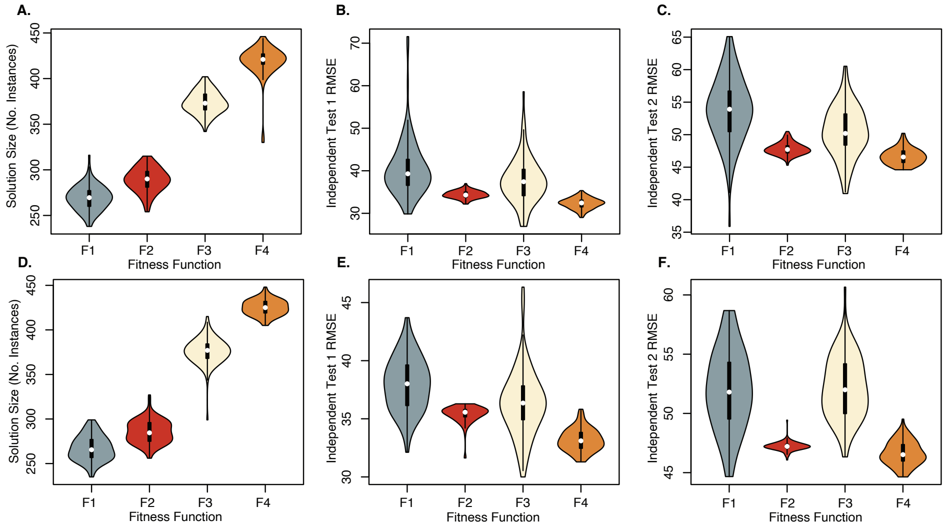

**Figure 2: Distributions of Subsets' Sizes and RMSEs in Cycling Tests. Shown are violin plots for four fitness functions. Top row shows 100 subsets computed by repeated data balancing and bottom row shows 100 subsets selected from the same balanced dataset. A. and D. Distribution of sizes. B. and E. Distribution of Test 1 RMSE scores of MLR models. Distribution of Test 2 RMSE scores of MLR models.**

for $F3$ and 32.32 for $F4$, respectively (Figure 2B). Independent test 2 was more challenging to predict, and the average RMSE scores were 53.78, 47.80, 50.46, and 46.66, for $F1$, $F2$, $F3$ and $F4$, respectively (Figure 2C).

Overall, we found that the external validation of fitness had a greater influence on performance than the penalty term. There was less variability in RMSE scores of $F2$ and $F4$, compared to $F1$ and $F3$. Also, we observed that the best performance on the independent tests 1 and 2 was achieved by training with as few as 287 ($F2$) and 294 ($F4$) instances out of the original 600.

To examine, how much variability is due to data balancing, we repeated the experiments while keeping the balanced dataset constant. Overall, we observed similar trends, where models generalized to test data better when fitness functions comprised two terms and were evaluated on external validation sets (Figure 2D-F). Notably, the distributions of subsets' sizes did not change, indicating that the balanced data of size 600 can be reduced to fewer than 300 training instances without a decrease in performance. Additionally, we confirmed that although the best performance was seen for subsets created with $F4$, an external validation of the fitness function had greater importance for the generalizability of the fitted model than the penalty term.

Next, we proceeded to quantitatively estimate data values of all 600 instances of the balanced dataset, as the frequency of their occurrences in the ensemble of 100 evolutionary subsets (Section 3). Once data values were computed, we ranked them in the increasing order and examined the per-athlete composition of the top 100 and the bottom 100 instances. Interestingly, data of the amateur athlete 5 were overrepresented among the least valuable instances and underrepresented among the most valuable instances in experiments without external validation (Table 4). On the other hand, contributions of different athletes' data to the top 100 instances were somewhat balanced for $F4$ function, yet they differed significantly among the least valuable instances for the same function. For example, only 6 out of 100 bottom instances belonged to athlete 3 compared with 26 instances of athlete 5. Remarkably, we note that for $F1$ and $F3$ functions, which perform self-validation, data of athlete 5 comprised almost 50% of the least valuable instances, indicating that this athlete's data may be very different from the data of the other athletes. These differences may be attributed to the selection bias because this athlete is an amateur, while 4 out of 5 remaining athletes are professionals and 1 out of 5 is of an unknown status. Alternatively, the differences may be due to the data capture biases arising from the types of sensors, which were used to collect the data.

*Result 5: Evolutionary data valuation provides insights about feature importance.* To understand whether the proposed data valuation approach can uncover putative biases in data or explain important instances by their features, we analyzed grouped data of approximately 5,875 voice recordings from 42 patients with Parkinson's disease. We used $F4$ fitness function to estimate data values of a balanced training set of 4,200 instances.

Once data values were computed from an ensemble of 100 GA subsets, we sorted all instances by their estimated values and compared the attributes of the top and bottom 10, 20, and 100 instances. For each attribute, including sex and age, a pairwise Student's t-test was applied to the distributions of high-valued and low-valued data and p-values less than 0.05 were deemed significant. The three attributes that had significant differences between the top and bottom 10 and top and bottom 20 instances, were harmonics-to-noise ratio, recurrence period density entropy, and detrended fluctuation analysis (Figure 3A-F). On the other hand, when the top and bottom 100 instances were compared, the only statistically different attributes were sex and age. The original dataset was relatively well balanced by age and was imbalanced in the sex attribute, with a ratio of 2.15 to 1 of male to female recordings. There is a higher prevalence of Parkinson's disease in males, which explains the imbalance. Interestingly, we found that the ratio of male to female patients among the top 100 instances was 1.17, and among the bottom 100 instances, it was 2.67, respectively (Figure 3H-I). In addition, the top 100 instances comprised data of younger patients, with the mean age of 59.60, compared with the mean age of 68.10 in the bottom 100 instances (Figure 3G).

**Table 4: Distribution of Per-Athlete Data Among High-Valued and Low-Valued Instances. For each fitness function, shown are the number of per-athlete instances among 100 most valuable and 100 least valuable instances.**

| Function | Top 100 | | | | | | Bottom 100 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **1** | **2** | **3** | **4** | **5** | **6** | **1** | **2** | **3** | **4** | **5** | **6** |
| **F1** | 11 | 26 | 17 | 19 | 6 | 21 | 17 | 14 | 5 | 6 | 54 | 4 |
| **F2** | 22 | 17 | 15 | 16 | 14 | 16 | 6 | 12 | 47 | 6 | 20 | 9 |
| **F3** | 13 | 24 | 10 | 20 | 9 | 24 | 14 | 13 | 12 | 7 | 51 | 3 |
| **F4** | 16 | 21 | 18 | 13 | 13 | 19 | 21 | 13 | 6 | 21 | 26 | 13 |

This result demonstrates the utility of our proposed approach. By automatically detecting the best subset in an ensemble of GA solutions, one can train a model more rapidly and sometimes, more accurately. On the other hand, by rank-based analysis of data values, one can explain which attribute values have high or low impact on the predictive performance.

*Result 6: Evolutionary approach detects most valuable samples in grouped data.* In the last experiment, we applied our approach in a slightly different setting. We aimed to identify most valuable samples of scRNA-seq studies. In these experiments, gene expression is measured in individual cells, comprising a biological sample. Because the individual cells are of different types and states, we wanted to include all cells of a specific sample into a common integrated dataset, and exclude samples that are redundant or are outside of the application domain, for example.

In this formulation, therefore, each chromosome encodes a subset of samples rather than a subset of individual cells. We tested this approach in the classification mode, using $F2$ fitness function and the FFNN. We report Kappa scores of self-validation on the original samples and on the subset of samples. On average, the size of the scRNA-seq datasets was reduced by approximately 40%, and datasets with the larger number of samples had bigger reduction in size (Table 5). There was no statistically significant difference between the Kappa scores of FFNNs that were trained with the subsets compared with the FFNNs that were trained with the original data. The only dataset, for which Kappa decreased when the

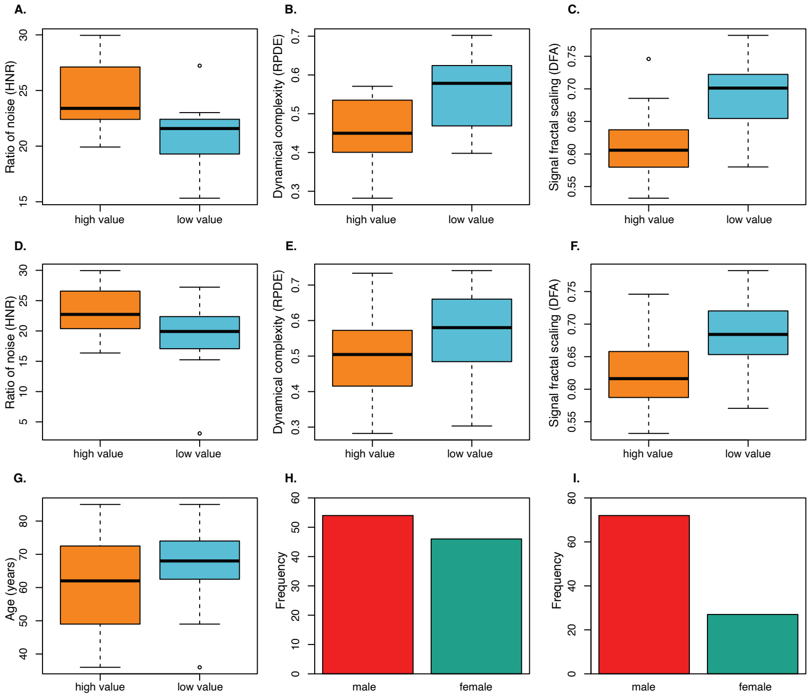

**Figure 3: Differences Between Attributes. Shown are differences in attributes of high-valued (orange) and low-valued (blue) training instances. A. Boxplot of HNR attribute in top 10 and bottom 10 instances. B. Boxplot of RPDE attribute in top 10 and bottom 10 instances. C. Boxplot of DFA attribute in top 10 and bottom 10 instances. D. to F. Boxplots of HNR, PDE and DFA attributes in top 20 and bottom 20 instances. G. Boxplot of age distribution in top 100 and bottom 100 instances. H. Barplot of sex attribute in top 100 (red: male, green: female). I. Barplot of sex attribute in bottom 100 instances (red: male, green: female).**

**Table 5: Performance in scRNA-seq Classification. For each dataset, shown are its name, total number of cells, and cell-type types. Number of samples and Kappa core in self-validation, are shown for the original dataset and for subset of samples. Percent reduction in the number of samples is shown in the last column.**

| | | | Original | | Subset | | |
|---|---|---|---|---|---|---|---|
| Dataset | Cells | Classes | Samples | Kappa | Samples | Kappa | Change (%) |
| **SIM1** | 50000 | 2 | 5 | 1.00 | 2.98 | 0.99 | 40.40 |
| **SIM2** | 50000 | 2 | 10 | 1.00 | 5.58 | 0.99 | 44.20 |
| **SIM3** | 50000 | 2 | 20 | 0.99 | 10.3 | 0.99 | 48.50 |
| **SIM4** | 50000 | 2 | 25 | 0.99 | 12.31 | 0.99 | 50.76 |
| **SIM5** | 16000 | 4 | 5 | 1.00 | 4.02 | 0.99 | 19.60 |
| **SIM6** | 32000 | 8 | 5 | 0.97 | 3.83 | 0.92 | 23.40 |
| **SIM7** | 64000 | 16 | 5 | 0.82 | 3.69 | 0.72 | 26.20 |
| **SIM8** | 50220 | 5 | 19 | 0.34 | 9.47 | 0.97 | 50.16 |
| **EXP1** | 50220 | 5 | 19 | 0.08 | 8.96 | 0.60 | 52.84 |

network was trained with fewer than 5 samples, was the synthetic dataset SIM7. The drop in performance to Kappa score of 0.72, may, however, be explained by the large number of cell-types in the dataset.

Most interesting results were observed for datasets SIM8 and EXP1. Recall, that SIM8 dataset was constructed using EXP1 as a template. Interestingly, starting Kappas for both of these datasets were very low, 0.34 for SIM8 and 0.08 for EXP1, pointing to the lack of confidence in the cell-type classification. Remarkably, when the number of samples in SIM8 was reduced from 19 to about 9.47, Kappa scores increased to an average of 0.97. Moreover, several solutions had a perfect Kappa score of 1.0.

We also observed a sharp increase in Kappa of EXP1, from 0.08 to 0.60, on the average. Lower performance in the experimental dataset compared with the synthetic one, however, indicates that synthetic data is easier to classify and may not be a good benchmark for bioinformatics tools. The most representative and discriminating subset of samples in EXP1 dataset contained 9 out of 19 samples. It comprised samples of 3 healthy donors, 1 of influenza patient, 1 of asymptomatic COVID-19 patient, 2 samples from mild and 2 from severe COVID-19 patients, respectively. Not surprisingly, the asymptomatic COVID-19 sample was selected in all 100 subsets, because this was the only sample of this type. Interestingly, the best subset encompassed all of the available healthy patients' samples. On the other hand, only 1 out of 3 influenza samples was selected into the best subset of EXP1. In contrast, influenza samples were overrepresented among the subsets of SIM8. Finally, we found that influenza sample 13 was the most valuable, in both datasets, SIM8 and EXP1.

## 5 CONCLUSION

Bioinformatics and health informatics rely on supervised and unsupervised ML to analyze massive and high-dimensional data. The growth in the number and size of new data is accompanied by an increased complexity of ML models and an increased burden on computational resources and energy. Rather than focusing on the learning process, data valuation methods estimate the importance of data that are used for training. Selection and quantification of the most valuable training data are active areas of academic research.

In this work, we proposed a practical approach to data valuation, inspired by natural evolution, and showed that it can be successfully used in the classification and regression tasks, with benchmarks as well as with real-world bioinformatics and health informatics data. Moreover, we demonstrated that evolutionary subsets are both, representative of the original data and generalizable to the unseen data. Finally, our approach can be used to estimate data values from an ensemble of evolutionary subsets, and can, in turn, help to explain predictions made by the models.

Future work will extend the proposed evolutionary approach to unsupervised learning, in particular, to cluster analysis.

## ACKNOWLEDGMENTS

This research was partially supported by the Pilot Award from the Wake Forest Center for Biomedical Informatics.

## REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, et al. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. USENIX Association, Savannah, GA, 265–283.

[2] Gustav Ahlin, Johan Karlsson, Jenny M Pedersen, Lena Gustavsson, Rolf Larsson, et al. 2008. Structural requirements for drug inhibition of the liver specific human organic cation transport protein 1. *Journal of medicinal chemistry* 51, 19 (2008), 5932–5942.

[3] Jesús Alcalá-Fdez, Alberto Fernández, Julián Luengo, Joaquín Derrac, Salvador García, Luciano Sánchez, and Francisco Herrera. 2011. KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing* 17 (2011).

[4] Pietro Barbiero, Giovanni Squillero, and Alberto Tonda. 2019. Evolutionary discovery of coresets for classification. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1747–1754.

[5] Richard E Bellman. 2015. *Adaptive control processes*. Princeton university press.

[6] Louise Bloch and Christoph M Friedrich. 2021. Data analysis with Shapley values for automatic subject selection in Alzheimer's disease data sets using interpretable machine learning. *Alzheimer's Research & Therapy* 13, 1 (2021), 1–30.

[7] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, et al. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 108–122.

[8] Trevor Campbell and Tamara Broderick. 2018. Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*. PMLR, 698–706.

[9] Javier Castro, Daniel Gomez, and Juan Tejada. 2009. Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research* 36 (May 2009), 1726–1730.

[10] Eugene C. Chen, Natalia Khuri, Xiaomin Liang, Adrian Stecula, Huan-Chieh Chien, et al. 2017. Discovery of Competitive and Noncompetitive Ligands of the Organic Cation Transporter 1 (OCT1; SLC22A1). *Journal of Medicinal Chemistry* 60, 7 (2017), 2685–2696.

[11] Francois Chollet et al. 2015. *Keras*. https://github.com/fchollet/keras

[12] Kenneth L Clarkson. 2010. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms (TALG)* 6, 4 (2010), 1–30.

[13] Roxana Daneshjou, Mary P Smith, Mary D Sun, Veronica Rotemberg, and James Zou. 2021. Lack of Transparency and Potential Bias in Artificial Intelligence Data Sets and Algorithms: A Scoping Review. *JAMA dermatology* 157, 11 (2021), 1362–1369.

[14] Tom De Bruyn, Gerard JP Van Westen, Adriaan P IJzerman, Bruno Stieger, de Witte, et al. 2013. Structure-based identification of OATP1B1/3 inhibitors. *Molecular pharmacology* 83, 6 (2013), 1257–1267.

[15] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. 2004. Least angle regression. *The Annals of statistics* 32, 2 (2004), 407–499.

[16] Iztok Fister Jr., Samo Rauter, Duš an Fister, and Iztok Fister. 2017. A collection of sport activity datasets with an emphasis on powermeter data.

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

[18] John J Grefenstette. 1993. Genetic algorithms and machine learning. In *Proceedings of the sixth annual conference on Computational learning theory*. 3–4.

[19] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, et al. 2019. Efficient Task-Specific Data Valuation for Nearest Neighbor Algorithms. *Proc. VLDB Endow.* 12, 11 (July 2019), 1610–1623.

[20] Michael I Jordan and Tom M Mitchell. 2015. Machine Learning: trends, perspectives, and prospects. *Science* 349, 6245 (2015), 255–260.

[21] Maria Karlgren, Anna Vildhede, Ulf Norinder, Jacek R. Wisniewski, Emi Kimoto, et al. 2012. Classification of inhibitors of hepatic organic anion transporting polypeptides (OATPs): influence of protein expression on drug–drug interactions. *Journal of medicinal chemistry* 55, 10 (2012), 4740–4763.

[22] Natalia Khuri, Esteban Murillo Burford, Sarah Parsons, and Chenqi Xu. 2021. A Game-Theoretical Approach for Data Acquisition From Fitness Tracking Devices. In *2021 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 1–6.

[23] Natalia Khuri and Sarah Parsons. 2021. A value-based approach for training of classifiers with high-throughput small molecule screening data. In *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. 1–10.

[24] Natalia Khuri and Anish Prasanna. 2021. Using Game Theory to Guide the Classification of Inhibitors of Human Iodide Transporters. In *Proceedings of ACM/SIGAPP Symposium On Applied Computing*. 916–923.

[25] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1885–1894.

[26] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, et al. 2019. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature methods* 16, 12 (2019), 1289–1296.

[27] Jeong Seok Lee, Seongwan Park, Hye Won Jeong, Jin Young Ahn, Seong Jin Choi, et al. 2020. Immunophenotyping of COVID-19 and influenza highlights the role of type I interferons in development of severe COVID-19. *Science Immunology* 5, 49 (2020), 1122–1127.

[28] Jeffrey T Leek, Robert B Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, et al. 2010. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics* 11, 10 (2010), 733–739.

[29] Feiyang Ma and Matteo Pellegrini. 2020. ACTINN: automated identification of cell types in single cell RNA sequencing. *Bioinformatics* 36, 2 (2020), 533–538.

[30] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. 2013. Bounding the estimation error of sampling-based shapley value approximation with/without stratifying. *CoRR, abs/1306.4265* 2 (2013), 1.

[31] Stéphane G Mallat and Zhifeng Zhang. 1993. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing* 41, 12 (1993), 3397–3415.

[32] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. 2020. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*. PMLR, 6950–6960.

[33] Annette M Molinaro, Richard Simon, and Ruth M Pfeiffer. 2005. Prediction error estimation: a comparison of resampling methods. *Bioinformatics* 21, 15 (2005), 3301–3307.

[34] Samo Rauter, Iztok Fister Jr., and Iztok Fister. 2015. *A collection of sport activity files for data analysis and data mining*. Technical Report. Uversity of Ljubljana and University of Maribor.

[35] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 3645–3650.

[36] Siyi Tang, Amirata Ghorbani, Rikiya Yamashita, Sameer Rehman, Jared A Dunnmon, et al. 2021. Data valuation for medical imaging using Shapley value and application to a large-scale chest X-ray dataset. *Scientific reports* 11, 1 (2021), 1–9.

[37] Antonio Torralba and Alexei A Efros. 2011. Unbiased look at dataset bias. In *CVPR 2011*. IEEE, 1521–1528.

[38] Athanasios Tsanas, Max Little, Patrick McSharry, and Lorraine Ramig. 2009. Accurate telemonitoring of Parkinson's disease progression by non-invasive speech tests. *Nature Precedings* (2009), 1.

[39] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. 2018. SCANPY: large-scale single-cell gene expression data analysis. *Genome biology* 19, 1 (2018), 1–5.

[40] Jinsung Yoon, Sercan Arik, and Tomas Pfister. 2020. Data Valuation using Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning*, Hal Daumé III and Aarti Singh (Eds.), Vol. 119. PMLR, 10842–10851.

[41] Luke Zappia, Belinda Phipson, and Alicia Oshlack. 2017. Splatter: simulation of single-cell RNA sequencing data. *Genome biology* 18, 1 (2017), 1–15.