



# Active Learning for Multi-way Sensitivity Analysis with Application to Disease Screening Modeling

Mucahit Cevik<sup>1</sup>  · Sabrina Angco<sup>1</sup> · Elham Heydarigharai<sup>1</sup> · Hadi Jahanshahi<sup>1</sup> · Nicholas Prayogo<sup>1</sup>

Received: 18 December 2021 / Revised: 24 May 2022 / Accepted: 14 June 2022 /

Published online: 15 July 2022

© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2022

## Abstract

Sensitivity analysis is an important aspect of model development as it can be used to assess the level of confidence that is associated with the outcomes of a study. In many practical problems, sensitivity analysis involves evaluating a large number of parameter combinations which may require an extensive amount of time and resources. However, such a computational burden can be avoided by identifying smaller subsets of parameter combinations that can be later used to generate the desired outcomes for other parameter combinations. In this study, we investigate machine learning-based approaches for speeding up the sensitivity analysis. Furthermore, we apply feature selection methods to identify the relative importance of quantitative model parameters in terms of their predictive ability on the outcomes. Finally, we highlight the effectiveness of active learning strategies in improving the sensitivity analysis processes by reducing the total number of quantitative model runs required to construct a high-performance prediction model. Our experiments on two datasets obtained from the sensitivity analysis performed for two disease screening modeling studies indicate that ensemble methods such as Random Forests and XGBoost consistently outperform other machine learning algorithms in the prediction task of the associated sensitivity analysis. In addition, we note that active learning can lead to significant speed-ups in sensitivity analysis by enabling the selection of more useful parameter combinations (i.e., instances) to be used for prediction models.

**Keywords** Sensitivity analysis · Machine learning · Active learning · Disease screening · Regression

---

✉ Mucahit Cevik  
mcevik@ryerson.ca

<sup>1</sup> Toronto Metropolitan University, 44 Gerrard St E, Toronto M5B 1G3, Ontario, Canada

## 1 Introduction

Mathematical models are constructed in many fields of science and technology to simulate real-world phenomena. These models tend to be complicated and are usually implemented in a powerful computing environment. As such, it is often not clear how the model reacts to modifications on its inputs. Sensitivity analysis is a vital stage in the process of model construction that can be considered as a systematic way to study how the uncertainty in the model's input will affect the outputs. It provides crucial insights into the structure and robustness of the outcomes as model parameters change.

There is a wide range of analytical/simulation-based quantitative models that are used to study various practical problems. We refer to all such models as “quantitative models” for the sake of generalization. Sensitivity analysis strategies are usually tailored to different quantitative model specifications such as the existence of a correlation between inputs, multiple outputs and a lack of freedom in the choice of data [1]. These strategies are mainly divided into two categories: local (or deterministic) and global (or probabilistic).

In general, the easiest approach to challenge a quantitative model is to analyze its performance when only one part of the model varies, which is the basic strategy employed in the one-way sensitivity analysis. Such methods (i.e., one-factor-at-a-time) fall under the category of local sensitivity analysis. While the one-way strategy is easy to use, it does not fully explore the input space. Therefore, it often fails to identify the existence of interactions between different input variables [2].

It is possible to extend the theory of one-way sensitivity analysis to multi-way sensitivity analysis, which aims to explain the quantitative model output based on the changes in more than one of its parameters. In practice, the number of possible combinations of input parameters in multi-way sensitivity analysis might be excessively large. Therefore, this method is usually restricted to exploring the effects of changing fairly few parameters at the same time [3].

Global (or probabilistic) methods are a collection of mathematical methods to explore whether a quantitative model's output variation (i.e., uncertainty) can be linked to the variations in the input parameters. This strategy might be used to associate the output uncertainty to various aspects of quantitative model uncertainty. The basic hypothesis of a probabilistic sensitivity analysis is the existence of sufficient knowledge about the probability distribution and correlation among the quantitative model inputs [4].

Sophisticated quantitative models usually include complicated simulations. Accordingly, it is not always feasible to solely rely on the intuition for the recognition of the model output's reaction to variations in the model inputs. In such cases, a full sensitivity analysis may not always be achieved regardless of whether all the parameter combinations are taken into account or a sampling method is used to reduce the dimensionality of the problem. Specifically, in a problem with two distinct values for each quantitative model input, a complete sensitivity analysis for all parameter combinations requires performing a complete factorial design with two levels. Hence, the associated number of simulation runs for

a quantitative model with  $N$  inputs would be  $2^N$ , which may render such a calculation infeasible [5]. Such computational difficulties may force researchers to explore only a subset of available parameters and their interactions. For instance, while there can be thousands of parameters in a typical earth and environmental systems modeling, many studies consider 20 or fewer parameters/factors in their sensitivity analysis [6]. Gupta and Razavi [7] discuss the challenges of conducting sensitivity analysis in such settings.

By identifying smaller subsets of combinations that can be used to generate the required outputs for the others, the need to evaluate a large number of parameter combinations could be avoided. Accordingly, identifying such subsets might have critical importance for the success of sensitivity analysis. A prediction model (e.g., a machine learning model) that is built on a small number of evaluated input parameter combinations can be used to estimate the outcomes for the remaining (unlabeled) parameter combinations [8]. This way, the entire process of sensitivity analysis can be considerably accelerated.

A general concern with using a machine learning model for predicting the quantitative model outputs in the process of sensitivity analysis is the formation of the training and test sets. This can be alleviated by evaluating a random subset of all parameter combinations and using this subset to train/test the prediction model. However, the performance of the resulting prediction model can be arbitrarily poor due to random subset selection and the existence of a large number of unlabeled (unevaluated) parameter combinations. Active learning strategies can be employed to guide the formation of the training set, and improve the performance of the prediction model. Such active learning approaches are frequently used for similar learning tasks where the unlabeled instances are abundant and labeling these instances are costly [8].

In this study, we consider machine learning approaches for speeding up the sensitivity analysis. We specifically focus on multi-way sensitivity analysis processes where the impact of interactions between various input parameters on the quantitative model outcomes are examined. In these types of settings, usually a range of values for each input is determined and then the combinations of the parameters are evaluated through quantitative model simulations. Considering that the number of such parameter combinations can be excessively large, we propose employing supervised learning and active learning-based approaches to predict the outcomes associated with each parameter combination. Main contributions of our study can be summarized as follows:

- We show how machine learning methods can be employed to conduct sensitivity analysis more effectively. Our analysis include showing the performances of various machine learning models in predicting the quantitative model outcomes as well as illustrating how feature selection methods can be used to better understand the most important parameters to conduct sensitivity analysis for.
- We propose using active learning to speed up the sensitivity analysis processes. We note that the prediction problem involving the sensitivity analysis is particularly suited for active learning as the unlabeled instances (i.e., parameter combi-

nations) are abundant, and labeling those require costly quantitative model (i.e., oracle) runs.

- We conduct an extensive numerical study with various machine learning and active learning methods using two datasets obtained from previously performed sensitivity analysis. Our results show that active learning can be considerably more effective than random (passive) sampling in improving the prediction model performances, and, therefore, has potential to speed up the sensitivity analysis processes significantly.

The remainder of the paper is organized as follows. In Section 2, the related works are summarized. In Section 3, different machine learning methods for the sensitivity analysis problem, which is considered as a regression task, are discussed along with the proposed active learning algorithm. In Section 4, results on the performance of the proposed approaches under different scenarios are reported, and the impact of the training set size on prediction model performance is examined. Finally, in Section 5, threats to validity are explored, which is followed by discussion and conclusions.

## 2 Related Work

Due to the broad applicability of sensitivity analysis in various domains, there is an extensive literature on different aspects of sensitivity analysis methodologies. Therefore, we only summarize the most relevant studies and refer readers to a recent review paper by [9].

Considering its relative simplicity, there are not many methodological studies on the local sensitivity analysis. A closely related problem to the local sensitivity analysis is the identification of unknown quantitative model parameters through calibration. Pfingsten [10] argued that while Monte Carlo methods are commonly used for determining the model parameters in micro electro-mechanical device design, it would be infeasible to perform such analysis if the underlying quantitative models are computationally expensive. Accordingly, the author empirically illustrated how to reduce the number of associated simulation runs by using an active learning strategy for the sensitivity analysis derived directly from the expected loss of Bayesian quadrature. Using a similar strategy for simulation calibration, [8] developed an active learning algorithm that was able to increase the accuracy of the prediction model for their simulation calibration problem. They empirically demonstrated how active learning was able to speed up the calibration procedure of a previously validated breast cancer simulation model.

Several studies in the literature investigated the probabilistic sensitivity analysis. Oakley and O'Hagan [1] provided a Bayesian methodology that aims to provide a complete and in-depth assessment of a quantitative model's sensitivity to variations in its input features. Their model accounts for the uncertainties related to bias terms and allows the user to specify a variety of probability distributions for the bias parameters associated with the inputs. Chen et al. [11] argued that typical sensitivity analysis approaches could rarely be directly applied to sequential decision-making

problems in healthcare, considering that such problems would involve evaluation of all probable sequences of decisions that usually fall in the order of trillions. Accordingly, they provided a probabilistic univariate method for recognizing the most sensitive parameters in Markov decision processes along with a probabilistic multivariate strategy that considers common uncertainty in model parameters to assess general trust in the suggested optimal strategy.

In a recent study, [12] investigated probabilistic sensitivity analysis on Markov models with uncertain transition probabilities. Their study outlined two sampling methods. In the first method, each row of the transition probability matrix was independently selected from a uniform distribution, whereas in the second method, a random sampling from a multivariate normal distribution was employed.

### 3 Methodology

We employ various machine learning approaches for predicting the quantitative model outcomes. We note that the associated outcomes are typically continuous-valued and can be predicted via regression models. Accordingly, we first review the machine learning-based regression models, and then provide an active learning algorithm tailored for multi-way sensitivity analysis.

#### 3.1 Review of Machine Learning Models

We consider the most commonly used algorithms in the literature to train a regression-based prediction model on the given problem data. Specifically, we experiment with linear models such as linear regression (LR) and ridge regression, which can be preferable over other models due to their simplicity and interpretability. Distance-based learning methods such as  $k$ -nearest neighbor (KNN) can be considered as a suitable baseline for the sensitivity analysis problem because the similar parameter combinations (i.e., instances) can be considered to have similar outputs (i.e., labels). Decision trees (DT) and tree-based ensembles such as Random Forests (RF), XGBoost and Light Gradient Boosting Machines (LGBM) can be used to learn the nonlinear relations between input features. Alongside these methods, we also consider other popular machine learning models such as Support Vector Regressor, Multi-Layer Perceptron Regressor, Gradient Boosting Regressor and Extra Trees Regressor. However, we only provide results for a subset of these methods either because they do not perform well or their performance is very similar to that of another model.

#### 3.2 Active Learning

Active learning (AL) is an iterative supervised learning method that actively selects the most useful training data points to learn from. In theory, if the prediction model is able to strategically pick/query the data points, the prediction model could perform better with a smaller training set [13]. Accordingly, in an

AL scheme, model training usually starts with a small training set, and the prediction model is re-trained by including more training instances that are carefully selected based on a predetermined query strategy. Previous studies show that AL is an effective query-based method that performs well for problems with few available data points and is less prone to overfitting [13, 14].

In the literature, there are many applications of active learning that use different settings and query strategies. Burbidge et al. [14] used QBC for active learning in real-valued functions and found that the QBC approach works well when the learner's bias is small. In natural language processing problems, there are usually a large number of unlabeled samples, and labeling these samples can be time consuming and costly. Figueroa et al. [15] used active learning techniques to classify the clinical texts, and performed a comparative analysis with distance-based, diversity-based, and combination-based active learning algorithms. AL has also been used in software analytics, especially for software defect prediction. For example, [16] used active learning with uncertainty sampling to automate the development of models which improve the performance of defect prediction between successive product releases.

In the case of sensitivity analysis, evaluating the entire combinations of different parameters using the quantitative model could be infeasible because of the required total computation/run time. AL could thus be used to avoid having to evaluate the entire set of parameter combinations while giving comparable, if not better, results.

The most important aspect of an AL scheme is determination of the query strategy that is able to identify the most useful instances to be included in the training set. According to [17], three main factors can be considered in identifying the query instances: *informativeness* (based on different criteria such as entropy and expected model change), *representativeness* (e.g., ignoring the outliers) and *diversity*. Based on these factors, we employ a query-by-committee (QBC) approach as our query strategy, which is enhanced by a filtering function that leaves out less informative instances and a clustering approach which promotes diversity of the instances that are added to the training set.

QBC is one of the most popular AL approaches for regression and classification problems [14]. Earlier work in the field of active learning established the theory behind the effectiveness of QBC in identifying the most useful instances to query [18–20]. On the other hand, providing a rigorous proof on the effectiveness of QBC enhancements is a non-trivial task. Accordingly, many subsequent studies relied on empirical analysis to show the effectiveness of various active learning strategies. In a recent study, [21] conducted a detailed numerical study and showed that incorporating diversity in the queried instances improves QBC-only strategy through prevention of redundant instances in the queries. Similar empirical approach to verifying the effectiveness of the query strategies were adopted in other studies as well [8, 22].

In QBC, a committee of learners,  $\mathcal{Q} = \{\theta_1, \theta_2, \dots, \theta_n\}$ , where a committee member  $\theta_i$  can be taken as a machine learning model, are employed to predict the outcomes for the unlabeled instances ( $\mathcal{U}$ ). Each committee member,  $\theta_i \in \mathcal{Q}$ , is trained over the labeled instances ( $\mathcal{X}$ ) and predicts a label for each unlabeled instance. Let

the  $i$  th committee member's prediction for the unlabeled instance  $\ell \in \mathcal{U}$  be  $P_\ell^i$ . Then, we calculate the variance of the predictions over the committee  $\mathcal{Q}$  as

$$\text{Var}_\ell = (1/n) \sum_{i=1}^n (P_\ell^i - \bar{P}_\ell)^2, \quad \ell \in \mathcal{U},$$

where  $\bar{P}_\ell = (1/n) \sum_{i=1}^n P_\ell^i$ . Unlabeled instances with the highest variances are regarded as the most informative ones (i.e., the committee members have the highest disagreements over these instances), and constitute the set of candidate instances,  $\mathcal{S}$ , to be evaluated through the quantitative model and added to the training set  $\mathcal{X}$ .

We consider further refinements of the candidate set  $\mathcal{S}$ , to be able to add more informative instances to the training set. Specifically, we consider a *second filter*, which takes one committee member to be the *base learner* ( $f$ ), and compares the predictions of the base learner with the actual labels of the instances in  $\mathcal{S}$ . The instances with highest absolute differences between their actual labels and predicted labels are included in a filtered set of candidate instances  $\mathcal{S}^F$ , which can then be added to the training set  $\mathcal{X}$ .

Finally, we employ a clustering approach in order to promote the diversity and representativeness of the instances that are added to the training set in each iteration of the AL algorithm. That is, over the remaining set of unlabeled instances (i.e.,  $\mathcal{U} \setminus \mathcal{S}^F$ ), we perform clustering to group the unlabeled instances, and randomly pick one instance from each cluster to construct a set ( $\mathcal{R}$ ), which can then be added to the training set (after their actual labels are obtained). This way, we aim to ensure that relatively different instances are added to the training set.

Algorithm 1 summarizes our comprehensive AL approach. In the initialization part of the algorithm, we divide all available instances ( $\mathcal{D}$ ) into three distinct sets as the training set ( $\mathcal{X}$ ) and the test set ( $\mathcal{Z}$ ) with known labels and the set of unlabeled instances ( $\mathcal{U}$ ). The other AL algorithm parameters such as query batch size ( $b$ ), second filter ratio ( $\rho$ )—specifying ratio of  $b$  to be included in the training set— and number of clusters ( $k$ )—specifying number of randomly selected instances to be included in the training set—are also set in the initialization step of the algorithm.

Algorithm 1 iterates until a stopping criterion is satisfied. At each step, a set of instances  $\mathcal{S}$  is determined by `QueryByCommittee( $\mathcal{Q}, \mathcal{X}, \mathcal{U}, b$ )` (see Algorithm 2) which, trains the members of  $\mathcal{Q}$  over the training set  $\mathcal{X}$ , and returns  $b$  instances ( $\mathcal{U}[\mathcal{I}]$ ,  $\mathcal{I}$  specifying indices of such instances) with the highest variance in their predicted labels using a generic function named “Get Max Value Indices”.

`SecondFilter( $\mathcal{S}, f, \mathcal{X}, \rho$ )` algorithm (see Algorithm 3) filters the provided set of instances,  $\mathcal{S}$ , in order to identify those with the largest prediction errors. These instances are represented by  $\mathcal{S}^F$ . Specifically, for each instance in  $\mathcal{S}$ , the predicted label is obtained using a base learner ( $f$ ) and a generic function (`Predict( $\cdot, \cdot$ )`), and is represented by  $P$ . The predicted values are compared to the actual label, represented by  $A$ , which is obtained using a generic function named `EvaluateQuantitativeModel( $\cdot$ )`. Next, `GetMaxValueIndices( $\cdot, \cdot$ )` function returns the indices ( $\mathcal{I}$ ) of

$\rho \times |S|$  instances with highest prediction errors, which are calculated by taking the difference between the predicted and actual labels. Then, these values are used to obtain the associated instances (e.g., using  $\mathcal{S}[\mathcal{I}]$ ).

Remaining unlabeled instances are clustered using the generic function `ClusterInstances(·,·)` and one instance is picked from each of the resulting clusters (using the generic `RandomChoice(·)` function) to be added to the training set. We assume that the `RandomChoice(·)` function calls the `EvaluateQuantitativeModel(·)` function to retrieve the actual label for each randomly selected instance. After updating the training set and the set of unlabeled instances, the base learner is trained over the new training set and its performance over the test set is reported using another generic function named `ReportPerformance(·)`.

## 4 Numerical Study

We next discuss the experimental setup and our findings with various machine learning methods. We consider R-squared ( $R^2$ ), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as the performance metrics for assessing the performances of different approaches.  $R^2$ , also known as coefficient of determination, is a measure of goodness-of-fit for regression models, defined as the proportion of the variance in the dependent variable that is predictable by the independent variables [23]. The ideal value for  $R^2$  is 1.0; however, it can also have negative values when the prediction model is arbitrarily bad. RMSE is more sensitive to the outliers compared to MAE, and it can be the preferred approach in the case model robustness towards outliers is highly valued. While  $R^2$  is a generic metric (i.e., always  $\leq 1.0$ ), MAE and RMSE values depend on the scale of the predicted values.

### 4.1 Dataset

We consider two distinct datasets to illustrate the effectiveness of our approaches. The first dataset, referred to as “POMDP dataset”, is based on the sensitivity analysis conducted by [24]. Specifically, [24] propose Partially Observable Markov Decision Process (POMDP) models to investigate the impact of the supplemental screening tests such as ultrasound and Magnetic Resonance Imaging (MRI) for timely detection of breast cancer for women with different breast densities. As several inputs for their models are subject to variability, they conduct a multi-way sensitivity analysis to assess the robustness of their models. As it is the case in [24]’s study, we only consider the sensitivity analysis task performed over the patients with extreme breast densities. The second dataset, which we refer to as “DES dataset”, is adopted from [8]’s study, and shows how the discrete-event simulation (DES) model outcomes change



based on various model parameters. The DES model is originally developed for replicating historical observations for breast cancer, evaluating breast cancer screening and treatment strategies, and understanding the cancer outcomes [25]. Note that the DES model performance is measured by total deviations from historical observations (i.e., using a numerical “score” value), and each parameter combination (i.e., instance) has an associated (single) score value (i.e., label).

---

**Algorithm 1** Active Learning
 

---

**Input:**

Set of all parameter combinations (instances) as  $\mathcal{D}$ ;  
 A committee  $\mathcal{Q} = \{\theta_1, \dots, \theta_n\}$  of regression models;  
 A selected base learner  $f$ .

**Initialization:**

$\mathcal{Z}$ : test set (labeled instances)  
 $\mathcal{X}$ : training set (labeled instances)  
 $\mathcal{U} = \mathcal{D} \setminus (\mathcal{Z} \cup \mathcal{X})$  (unlabeled instances)  
 $b$ : query batch size,  $\rho$ : second filter ratio,  
 $k$ : # of clusters

**repeat**

$\mathcal{S} \leftarrow \text{QueryByCommittee}(\mathcal{Q}, \mathcal{X}, \mathcal{U}, b)$   
 $\mathcal{S}^F \leftarrow \text{SecondFilter}(\mathcal{S}, f, \mathcal{X}, \rho)$   
 $\{\mathcal{C}_j\}_{j=1}^k \leftarrow \text{ClusterInstances}(\mathcal{U} \setminus \mathcal{S}^F, k)$   
 $\mathcal{R} \leftarrow \cup_{j=1}^k \text{RandomChoice}(\mathcal{C}_j)$   
 $\mathcal{X} \leftarrow \mathcal{X} \cup (\mathcal{S}^F \cup \mathcal{R})$   
 $\mathcal{U} \leftarrow \mathcal{U} \setminus (\mathcal{S}^F \cup \mathcal{R})$   
 Train  $f$  on  $\mathcal{X}$   
**ReportPerformance**( $f, \mathcal{Z}$ )

**until** some stopping criterion

---



---

**Algorithm 2** QueryByCommittee( $\mathcal{Q}, \mathcal{X}, \mathcal{U}, b$ )
 

---

**Initialization:**

$V$ : empty list of size  $|\mathcal{U}|$   
 Train  $\theta_i$  on  $\mathcal{X}$ ,  $\forall \theta_i \in \mathcal{Q}$   
**for**  $\ell \in \mathcal{U}$  **do**  
 $V_\ell \leftarrow \text{variance}([\text{Predict}(\theta_i, \ell)]_{i=1}^n)$   
**end for**  
 $\mathcal{I} \leftarrow \text{GetMaxValueIndices}(V, b)$   
**return**  $\mathcal{U}[\mathcal{I}]$

---

**Algorithm 3** SecondFilter( $\mathcal{S}, f, \mathcal{X}, \rho$ )

---

**Initialization:**  
 $P$ : empty list of size  $|\mathcal{S}|$   
 $A$ : empty list of size  $|\mathcal{S}|$   
 $E$ : empty list of size  $|\mathcal{S}|$   
Train  $f$  on  $\mathcal{X}$   
**for**  $\ell \in \mathcal{S}$  **do**  
     $P_\ell \leftarrow \text{Predict}(f, \ell)$   
     $A_\ell \leftarrow \text{EvaluateQuantitativeModel}(\ell)$   
     $E_\ell \leftarrow |P_\ell - A_\ell|$   
**end for**  
 $\mathcal{I} \leftarrow \text{GetMaxValueIndices}(E, \rho \times |\mathcal{S}|)$   
**return**  $\mathcal{S}[\mathcal{I}]$

---

Table 1 summarizes the values for the input parameters that are considered in the sensitivity analysis associated with POMDP and DES datasets. “Baseline value” column includes the parameter values used in the main experiments, whereas “Sensitivity levels” column list the values used for the corresponding parameter during

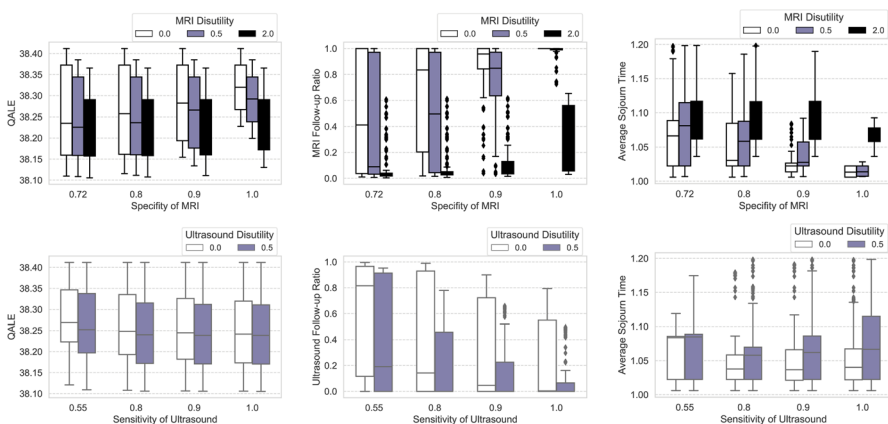
**Table 1** Input parameter values for the sensitivity analysis (baseline values refer to the parameter values that are originally used in the quantitative model, and sensitivity levels refer to the values a specific parameter can take during the sensitivity analysis)

Parameter	Baseline value	Sensitivity levels
(a) POMDP dataset		
Sensitivity (%), ultrasound	55	50, 70, 90, 100
Sensitivity (%), MRI	90	90, 95, 100
Specificity (%), ultrasound	94	90, 95, 100
Specificity (%), MRI	72	70, 80, 90, 100
Disutility (days), mammography	0.5	0, 0.5
Disutility (days), ultrasound	0.5	0, 0.5
Disutility (days), MRI	2	0, 0.5, 2
Disutility (days), positive test	14	0, 7, 14
(b) DES dataset		
Fraction of LMP tumors	0.42	0.00, 0.30, 0.60
LMP dwell time	0.5	1, 2, 3
Onset proportion	0.90	0.80, 0.90, 1.00
APC lag	3	1, 3, 7
Percentage of aggressive tumors	0.01	0.01, 0.10, 0.20
Percentage of highly aggressive tumors	0.02	0.006, 0.04, 0.10
Mean tumor growth	0.12	0.01, 0.02, 0.03, 0.04, 0.05
Variance tumor growth	0.012	0.00, 0.03, 0.05

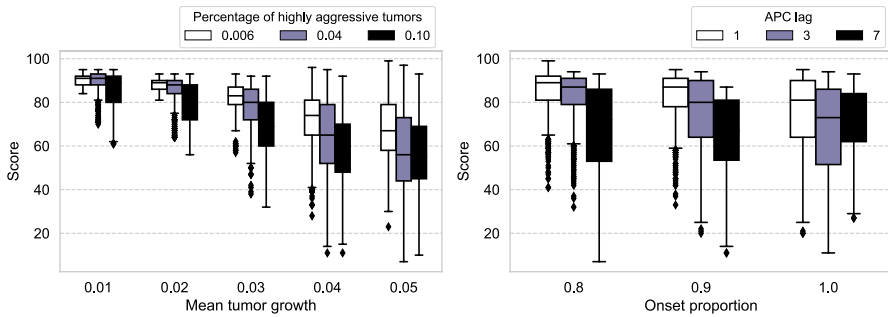
the sensitivity analysis. For instance, [24] took the ultrasound sensitivity as 55% in their original experiments; however, they used four distinct values ranging from 50% to 100% for the same parameter during the sensitivity analysis. Sandikci et al. [24] focus on five different model outcomes in the sensitivity analysis, namely, Quality-Adjusted Life year Estimate (QALE), the ratio of mammography screenings followed by MRI/ultrasound, average sojourn time, and detection rate. Note that we have eight different input parameters and each of them has two to four levels, which results in a total of 5,184 parameter combinations to be evaluated with two POMDP models proposed in the paper (i.e., a total number of quantitative model runs is 10,368). Assuming that each quantitative model run and the associated discrete-event simulation model, which is used for policy analysis, take 10 minutes, completing 10,368 runs would take 72 days if the simulation model runs are not done in parallel. Accordingly, it is necessary to employ a more clever approach to conduct such a sensitivity analysis. Table 1b demonstrates the input parameter values in DES dataset. The original dataset used by [8] has 378,000 parameter combinations; however, we downsampled some of the parameter combinations to make the dataset more inline with standard sensitivity analysis process as well as to have a more uniform experimental setup. The resulting DES dataset has 10,935 parameter combinations to be evaluated by the quantitative model.

## 4.2 Exploratory Data Analysis

Figures 1 and 2 summarize some of the input parameters and their interactions with the outcome measures in the POMDP and DES datasets, respectively. First of all, we observe that, while there is low variability in some outcomes (e.g., QALE and Average Sojourn Time in the POMDP dataset), there is high variability in the others (e.g., MRI and Ultrasound follow-up ratios in the POMDP dataset). Accordingly, we expect the prediction models to perform better for predicting the outcomes with low variability. Besides, we note that conducting a multi-way sensitivity analysis is



**Fig. 1** Distribution of data points for different class labels in POMDP dataset



**Fig. 2** Distribution of data points according to the class label in DES dataset

justified for both datasets as the interaction between input parameters might have a significant impact on the outcomes. For instance, in the POMDP dataset, MRI follow-up ratio approaches 100% only when the disutility of receiving MRI is not too high (e.g., 2 days). Lastly, complex interactions between different input parameters indicate the necessity to use nonlinear regression models (e.g., DTs, and ensemble models) for the prediction task.

### 4.3 Experimental Setup

We conduct our analysis using the machine learning models that are readily available in scikit-learn python library [26]. Our preliminary analysis did not show significant performance improvement attributed to various hyperparameters (e.g., those of machine learning models and active learning algorithm). Therefore, we do not perform extensive parameter tuning in our experiments with different machine learning methods.

In the active learning algorithm, there is a large number of possibilities to construct the committee,  $\mathcal{Q}$ , in the QBC step. We consider the committee members to be from a diverse family of machine learning models. In addition, through our preliminary analysis for committee selection, we identify the models that show a relatively large degree of disagreement in predictions. There are different choices for the stopping criterion in AL strategies such as getting a desired performance level by using the test set and running the algorithm for a specific number of iterations. In our implementations, since we are considering a fixed number of instances, and our aim is to compare AL with standard supervised learning approaches, AL algorithm is set to run until  $\mathcal{U} = \emptyset$ . In addition, we consider  $k$ -means clustering as the clustering approach. As different parameters (i.e., features) tend to be on different scales, we standardize the features before performing the  $k$ -means clustering. The remaining AL algorithm parameters and their values are presented in Table 2.

### 4.4 Comparison of Regression Models

We first experiment with various machine learning models to assess the effectiveness of these models for predicting the quantitative model outcomes, and their

**Table 2** AL algorithm parameter values

Parameter	Notation	Value
Set of all instances	$\mathcal{D}$	$ \mathcal{D}  = 5184$
Initial training set	$\mathcal{X}$	$ \mathcal{X}  = 1\% \times  \mathcal{D} $
Unlabeled instance set	$\mathcal{U}$	$ \mathcal{U}  = 79\% \times  \mathcal{D} $
Test set	$\mathcal{Z}$	$ \mathcal{Z}  = 20\% \times  \mathcal{D} $
QBC query batch size	$b$	$1\% \times  \mathcal{D} $
Second filter ratio	$\rho$	0.5
Number of clusters	$k$	$0.5\% \times  \mathcal{D} $

potential in facilitating sensitivity analysis processes. We employ 5-fold cross validation in these experiments, which divides the dataset to five folds, and in each step, one fold is used as the testing set and the remaining four folds are used as the training set. In order to eliminate the impact of randomization, this experiment is repeated ten times and the average of the outcomes is reported for each model. Table 3 shows the results of these experiments for various machine learning models over the POMDP and DES datasets based on three performance metrics: MAE, RMSE, and  $R2$ .

**Table 3** Performances of different prediction models based on MAE,  $R2$ , and RMSE values (the best performance in terms of  $R2$  value is bold-faced)

Output	Measure	LR	KNN	RF	DT	LGBM	XGBoost	
(a) POMDP dataset								
QALE	MAE	0.019	0.005	0.001	0.001	0.002	0.002	
	$R2$	0.925	0.993	<b>0.999</b>	<b>0.999</b>	<b>0.999</b>	<b>0.999</b>	
	RMSE	0.023	0.007	0.002	0.002	0.003	0.003	
MRI	MAE	0.192	0.045	0.018	0.017	0.034	0.031	
	Count	$R2$	0.662	0.953	<b>0.989</b>	<b>0.989</b>	0.978	0.982
Ultrasound	Count	RMSE	0.229	0.083	0.042	0.041	0.057	0.052
	MAE	0.190	0.063	0.014	0.012	0.039	0.035	
	Count	$R2$	0.562	0.907	0.987	<b>0.988</b>	0.965	0.973
Detection	Rate	RMSE	0.234	0.107	0.039	0.038	0.065	0.057
	MAE	0.001	0.001	0.001	0.001	0.001	0.001	
	$R2$	0.554	0.929	0.983	0.982	<b>0.986</b>	0.967	
Sojourn	Time	RMSE	0.002	0.001	0.001	0.001	0.001	0.001
	MAE	0.025	0.009	0.002	0.002	0.003	0.004	
	$R2$	0.514	0.917	<b>0.994</b>	<b>0.994</b>	0.986	0.986	
(b) DES dataset	Score	RMSE	0.032	0.013	0.003	0.003	0.005	0.005
	MAE	7.397	3.760	1.781	1.802	2.663	1.987	
	$R2$	0.635	0.881	0.960	0.934	0.942	<b>0.965</b>	
	RMSE	9.963	5.692	3.310	4.221	3.958	3.063	

We only present results with a representative machine learning models, namely, LR, KNN, RF, DT, LGBM, and XGBoost. Some other machine learning models with similar structures to the ones presented in Table 3 also provide good performance for predicting the quantitative model outcomes. For instance, GBR (a boosting ensemble similar to XGBoost and LGBM) and Extra Trees Regressor (a bagging ensemble similar to RF) show somewhat similar performance to their counterparts presented in this table. On the other hand, we note that machine learning models such as Naive Bayes, Support Vector Machines and Multi-layer Perceptrons, do not provide consistently good performance across different datasets and target labels, and therefore omitted from our analysis.

[24] developed linear regression meta-models on the (POMDP) dataset obtained through multi-way sensitivity analysis with the aim of guiding the reader to better evaluate the impact of input parameters. [8] considered a limited number of machine learning models, namely, LR, artificial neural networks (ANN), bagging ensembles with ANN (bagANN) for the simulation calibration dataset (which is a variant of our DES dataset), and concluded that bagANN is the best performing supervised learning approach. Different from these two studies, we observe that DT and the tree ensembles (e.g., RF, LGBM and XGBoost) provide the best performance in terms of predicting the quantitative model outcomes.

For the POMDP dataset, which contains five different target outputs (labels), we observe that DT and RF provide the best overall performance, as evidenced by all three performance metrics. Superior performances of these two models are more evident in predicting outcomes with higher variability such as MRI and ultrasound counts. On the other hand, the majority of the prediction models perform well for an outcome with low variability such as QALE, which only varies between 38.10 and 38.40 years over all the instances in the dataset (see Fig. 1). For the DES dataset, we note that XGBoost performs the best in terms of  $R^2$  and RMSE, whereas RF performs the best in terms of MAE, indicating that XGBoost is more robust towards outliers.

#### 4.5 Assessing Feature Importances

Assessing the relative importance of each input parameter on the prediction model outcomes through feature selection (FS) strategies can be useful, as it provides an extra tool for selecting the most important parameters to focus on during the sensitivity analysis, with respect to the output variable(s). Feature selection can also be used for dimensionality reduction and fine-tuning the performances of some machine learning models. We employ alternative FS strategies, namely, feature importances (ImpScore) from tree-based learners (e.g., see `feature_importances_` method in `scikit-learn`), F-statistic (F-stat), and Spearman correlation. DTs and DT-based ensembles (e.g., RF and XGB) provide a method for feature importance calculation. Specifically, in a DT, these values can be computed by going over all the splits that involve the target feature, and measuring how much variance (or Gini index) is reduced compared to the parent node due to the split on the target feature. On the other hand, F-statistic and Spearman correlation values provide a

more indirect method to calculate the feature importances, as they only consider the statistical relation between a given feature and the class label. Additionally, all three methods commonly do not take into account feature interactions while calculating the importance values, that is, each feature is considered in isolation. There exist other strategies to calculate the feature importances (e.g., mutual information and chi-squared statistic); however, they are omitted in our analysis for the sake of brevity.

Table 4 summarizes the feature selection results for each input parameter and prediction model outcome. We observe that, while there might be disagreements among feature selection methods in terms of the orderings of the input parameters for a given outcome, their agreements might be considered as a strong indicator for the impact of certain inputs on the outcome measures. For instance, for the POMDP dataset, all three FS strategies indicate that the disutility of MRI is the most dominant factor for the MRI and ultrasound ratios, which is reasonable considering that high disutility values for MRI prevents MRI recommendations, and only one of MRI and ultrasound is recommended to a patient at any given period. Similarly, for the DES dataset, we observe that Mean tumor growth parameter is significantly more important than others to predict the score (i.e., label) value. This observation can be taken as an indicator that more values should be sampled from this particular parameter to assess the robustness of the quantitative model through sensitivity analysis.

#### 4.6 Impact of Training Set Size

As discussed earlier, one of the main issues with using standard supervised learning approaches for sensitivity analysis is the fact that there is no systematic way of determining a sufficient training set size that potentially leads to a well performing prediction model. In order to assess the impact of the training set size on the predictive performance, we trained machine learning models using different training set sizes starting with employing 1% of the available data as the initial training set and 20% as the test set. We repeated the experiments 30 times by shuffling the data to eliminate the randomization effects. Figure 3 shows the performance of the machine learning models in terms of  $R^2$  values relative to the training set size. These results demonstrate that tree-based ensembles (RF, LGBM and XGBoost) converge faster than other machine learning models in all five regression tasks associated with the POMDP dataset as well as that of DES dataset. Overall, we observe that XGBoost and LGBM have a slight edge over RF in terms of convergence performance, which is especially evident for the DES dataset. On the other hand, LR performs the worst consistently in all cases. KNN and DT performances improve more slowly with increasing training set size.

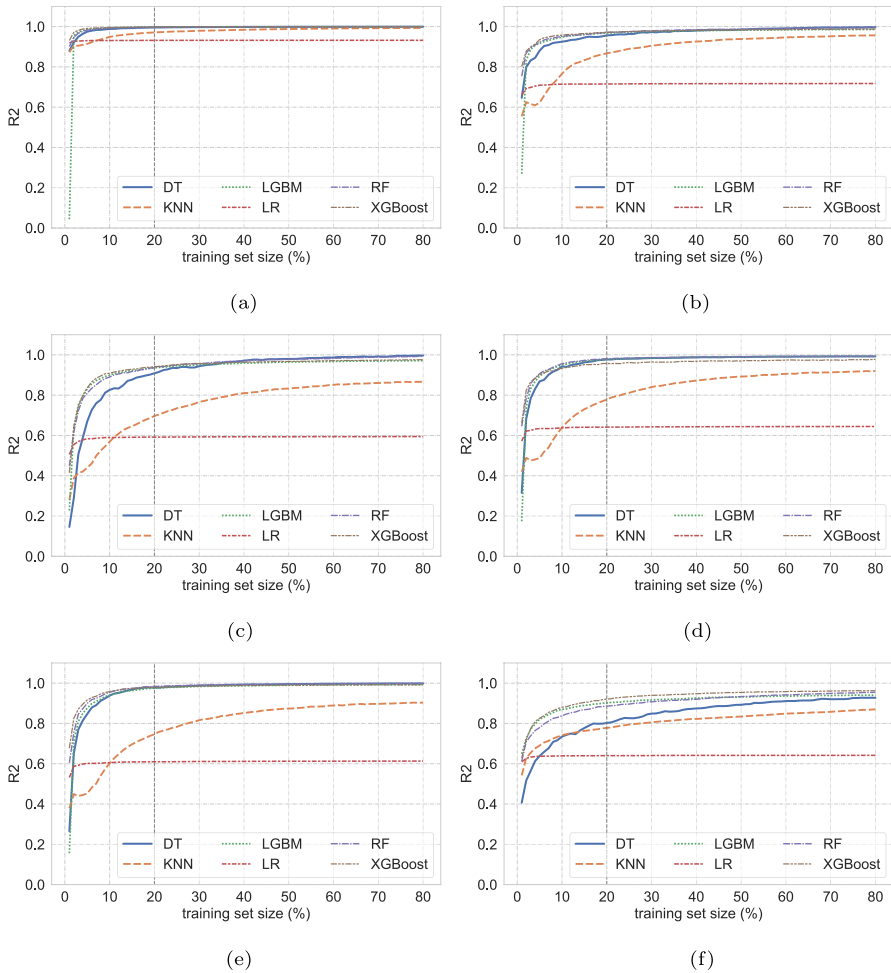
#### 4.7 Active Learning Results

We experiment with different active learning algorithm settings. Our preliminary analysis include examining different query strategies and batch size parameter (i.e., number of instances to be added to training set at each iteration). We observe that

**Table 4** The relative importance of each parameter based on Importance Score (ImpScore), F-statistic (F-stat) and Spearman Correlation (Spearman)

(a) POMDP dataset		Sensitivity (%)		Specificity (%)		Sensitivity (%)		Specificity (%)		Disutility (days)		Disutility (days)	
Output	Method	MRI	MRI	MRI	MRI	ultrasound	ultrasound	ultrasound	ultrasound	mammography	ultrasound	MRI	positive test
QALE	ImpScore	0.002	0.056	0.012	0.008	0.036	0.005	0.089	0.792				
	F-stat	5.5	165.8	32.5	17.2	197.7	11.7	350.2	19390.2				
MRI	Spearman	-0.043	0.182	-0.072	0.060	-0.190	-0.047	-0.243	-0.889				
	ImpScore	0.007	0.253	0.041	0.026	0.001	0.023	0.386	0.264				
Count	F-stat	10.9	1103.2	92.8	33.4	0.1	39.3	3020.8	869.1				
	Spearman	-0.071	0.445	0.158	-0.097	0.002	0.099	-0.586	-0.429				
Ultrasound	ImpScore	0.007	0.104	0.184	0.165	0.001	0.095	0.267	0.178				
	F-stat	7.3	388.2	683.0	563.6	0.1	316.6	1678.1	52.9				
Detection	Spearman	0.034	-0.319	-0.264	0.304	0.001	-0.273	0.536	0.120				
	ImpScore	0.004	0.275	0.058	0.125	0.001	0.042	0.221	0.276				
Rate	F-stat	2.3	959.4	0.1	204.2	0.1	74.6	1244.4	1721.1				
	Spearman	-0.099	-0.394	-0.041	-0.111	-0.002	0.077	0.490	0.493				
Sojourn	ImpScore	0.008	0.252	0.065	0.158	0.001	0.049	0.209	0.258				
	F-stat	6.5	899.2	0.4	252.6	0.1	86.9	1063.9	1626.3				
Time	Spearman	-0.110	-0.390	-0.058	-0.120	-0.003	0.083	0.487	0.485				
	Method	Onset	Mean	Variance	LMP	% of	% of highly	LMP	LMP				
Output		proportion	tumor	tumor	tumors	aggressive	aggressive	lag	APC				
		growth	growth	growth	tumors	tumors	tumors	time	APC				
		0.110	0.430	0.047	0.061	0.050	0.106	0.176	APC				
Score	ImpScore	415.1	7257.9	69.4	1.8	176.9	800.7	1341.3	0.5				
	F-stat	-0.202	-0.666	0.065	0.045	-0.143	-0.259	-0.367	0.009				

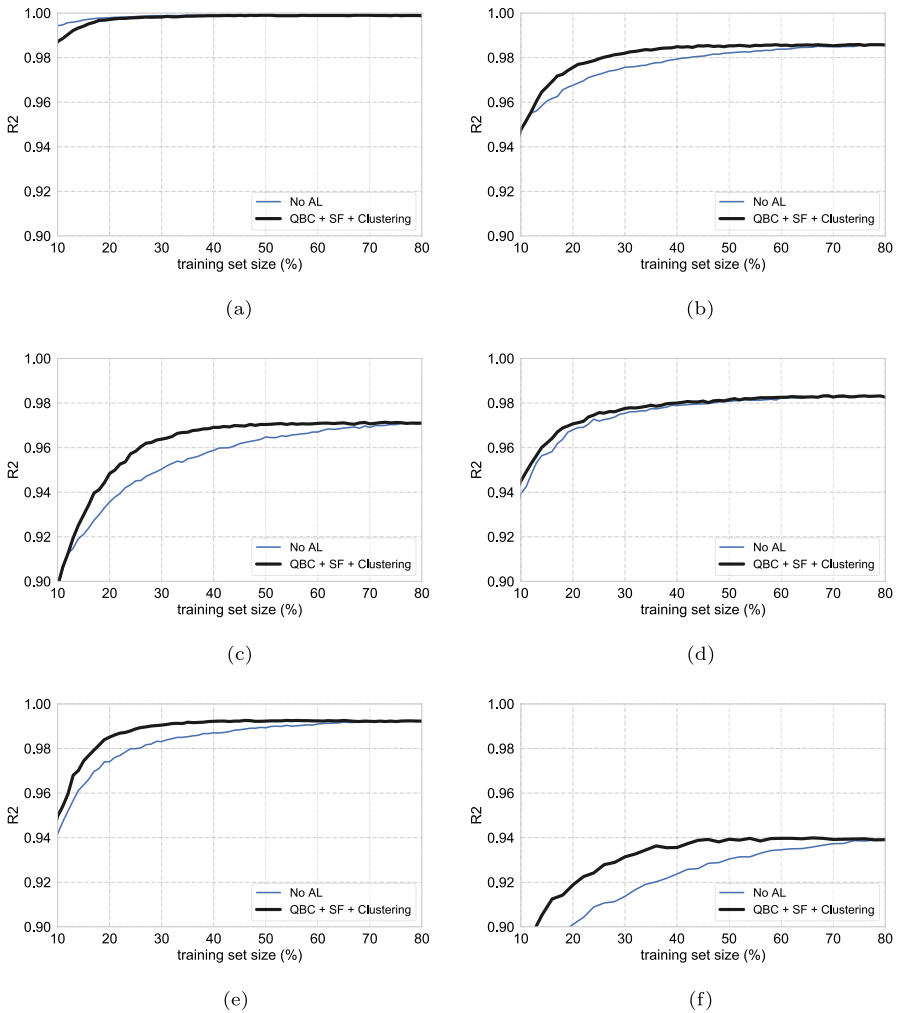




**Fig. 3** Training size effect on the performance of different regressors using standard supervised learning approaches. (a) POMDP — QALE. (b) POMDP — MRI follow-up ratio. (c) POMDP — ultrasound follow-up ratio. (d) POMDP — detection rate. (e) POMDP — sojourn time. (f) DES — score

these parameters can contribute to minor differences in the active learning algorithm performance. Due to its relatively fast convergence (i.e., as observed in training set size experiments), we consider LGBM as the base learner. Through our preliminary experiments, we identify the committee members in QBC approach as LGBM, DT, RF and XGBoost.

Figure 4 demonstrates the change in the  $R^2$  values using AL (QBC + SF + Clustering) and supervised learning (No AL) approaches with respect to increasing training set size. Note that, in No AL case (which is performed using LGBM), instances that are added to the training set are randomly sampled, whereas in AL case, instances are queried based on AL strategies. We repeat the experiments 30 times



**Fig. 4** Impact of active learning (base learner: LGBM, QBC committee: {DT, RF, XGB, LGBM}). (a) POMDP — QALE. (b) POMDP — MRI follow-up ratio. (c) POMDP — ultrasound follow-up ratio. (d) POMDP — detection rate. (e) POMDP — sojourn time. (f) DES — score

by shuffling the dataset to prevent bias in the outcomes. These results show that AL is performing equivalently or better than the supervised learning for all the learning tasks. In particular, for the POMDP dataset, in predicting the MRI and ultrasound follow-up ratios, AL performs significantly better, and prediction model performance reaches a steady state by using 30% of the instances. Similarly, for the DES dataset, AL consistently outperforms random sampling. We note that, best achievable prediction performance, as indicated by  $R^2$  values, is lower for the DES dataset, which shows that predicting the quantitative model outcomes is a more challenging task in this case. It is also important to note that, in certain cases (e.g., see Fig. 4b

and **c**, AL trails random sampling for small training set sizes (e.g., when training set size is smaller than 10% of the dataset), and then achieves better performance once the training set size reaches a certain level.

Overall, these results indicate a significant improvement over the standard supervised learning approach (i.e., random sampling) in terms of the number of quantitative model runs required, and highlights the potential of AL in improving sensitivity analysis processes. Also note that, while we do not provide results on the other performance metrics (e.g., RMSE and MAE) for the sake of brevity of the discussion, our analyses point to similar conclusions for such metrics as well. We provide a detailed empirical analysis with various AL settings (e.g., with different base learners and QBC committee compositions), and compare the results from different AL query strategies in the appendix (see Appendix ??).

## 5 Discussion and Conclusions

Any quantitative model's input values and assumptions might be subject to changes. The sensitivity analysis examines such changes and investigates their impact on the model conclusions. In many practical cases, complex quantitative models are used to analyze the system behavior, and a detailed sensitivity analysis based on repeated quantitative model simulations might not always be feasible. In this study, we aim to improve the sensitivity analysis processes by reducing the number of parameter combinations required to be evaluated during sensitivity analysis. We train different supervised regression models, such as LR, KNN, RF, and LGBM, for two distinct datasets: POMDP dataset [24] and DES dataset [8]. Both datasets contain eight input parameters; however, the POMDP dataset has five main model outcomes and DES dataset has a single model outcome. We examine the relations of the input parameters with the model outcomes through various feature selection strategies. Our numerical analysis indicates that it is possible to obtain highly accurate prediction models for the underlying regression task for both datasets.

Our detailed numerical study shows that AL strategies can be employed to further improve the sensitivity analysis processes. We find that AL leads to a faster prediction model performance convergence compared to random sampling, indicating that fewer instances are required to train the prediction models if AL-based query strategies are employed. In most cases, less than 20% of the all parameter combinations are sufficient to train a highly accurate prediction model. Accordingly, the sensitivity analysis procedure that would require 72 days of run times (on a standalone computer without parallelization) in [24]'s study can be completed in less than 18 days. We can expect run time savings to be higher for more detailed sensitivity analysis procedures that would involve significantly higher number of parameter combinations than the ones considered in our study.

While our results show that machine learning approaches can be effectively used for predicting quantitative model outcomes, and help easing computational burden of multi-way sensitivity analysis, there are certain challenges for direct employment and generalization of machine learning methods for this task. For instance, most feature selection strategies are designed for univariate feature selection, and do not

account for feature couplings. Therefore, examining the individual features and their impacts on the quantitative model outcomes might not be straightforward. Furthermore, the best performing machine learning model might change from one quantitative model to another. For example, our 5-fold cross validation results show that while RF and DT are the best performing models for the POMDP dataset, XGBoost is the best performing model for the DES dataset. Both datasets used in our analysis contain approximately 10,000 instances with eight features (i.e., input parameters for the quantitative model). As such, our findings with regards to the best performing machine learning and active learning strategies might not apply for the problems with significantly larger number of input parameter combinations (i.e., instances) [6]. For active learning, much larger batch sizes might be needed in this case to avoid excessive algorithm run times due to repeated prediction model training (e.g., for the base learner and for those in the committee).

We note that some quantitative models have multiple outputs (as in the case of POMDP dataset). Therefore, different machine learning models might be needed to be trained for predicting different outputs, which can be computationally burdensome. In active learning, the algorithm run time is significantly impacted by the selection of the base learner and the QBC members. We note that certain machine learning models (e.g., LGBM and Extra Trees Regressor) can be favored over others in the active learning algorithm based on their faster training times. More importantly, a suitable training set size is difficult to determine, and while having too few of quantitative model runs would lead to a small training set and a low performing prediction model, too many quantitative model runs would cause wasted computational resources. AL can be used to efficiently identify the set of training instances that would lead to a highly accurate prediction model. Note that most quantitative model outcomes are numeric valued, which require regression models for prediction. While active learning is well studied for classification tasks, there are relatively few well-established strategies for the regression tasks, which makes identifying the best performing active learning strategy challenging. In our analysis, we mainly relied on QBC, and conducted extensive preliminary analysis to identify the best performing committee for the quantitative model outcome prediction task.

Although we experimented with various base learner and committee member configurations, we did not test every possible combination of committee members for active learning. We demonstrated the results for KNN, DT, RF and LGBM as QBC committee members, and LGBM as the base learner in our baseline results with the AL algorithm. A different combination of machine learning models for the committee members and base learners may have given better results, but we chose these models based on the QBC method logic that the committee should be a diverse set of machine learning models which have large disagreements in predictions [14]. We note that new machine learning models are developed continuously, and these models could have given better results than the ones obtained in this study. In future research, we aim to investigate more complex machine learning models (e.g., different deep neural network configurations or custom ensemble methods) in the regression analysis in order to determine if they should be used as a committee member or base learner. Furthermore, our computational experiments are limited to two datasets, and depending on the underlying research problem and the structure of the

model outcomes, performances of the proposed approaches might vary significantly. That is, it might not always be possible to obtain a prediction model with a high  $R^2$  value (i.e., close to one). Besides, because the typical quantitative model output is numeric, we focused on regression models in our analysis. In case the quantitative model output dictates a classification task, classification models can be used as prediction models, both for standard supervised learning and active learning. In this regard, a comparative analysis with different datasets obtained from different sensitivity analysis processes can be used to understand the factors affecting the success of using machine learning for sensitivity analysis. In addition, investigating the relative performance of a detailed deterministic sensitivity analysis enabled through our proposed methodology with respect to the probabilistic sensitivity analysis is a subject of future research.

## Appendix : A. Active Learning Query Strategies

The performance of the AL algorithm is highly dependent on the employed query strategy to identify parameter combinations (i.e. instances) to be evaluated by the quantitative model. Accordingly, we first review various AL query strategies [27] and compare their results with our proposed AL approach.

### A.1. Expected Model Change Maximization

Expected model change maximization (EMCM) aims to add samples that create the maximum change in the current model. In this method, the model change is defined as the gradient of the loss for an unlabeled sample. Different EMCM approaches are proposed for linear and nonlinear regression tasks. For the case of linear regression, EMCM constructs  $n$  linear regression models by using bootstrap. Let the  $i$  th model's prediction for the  $j$  th unlabeled instance ( $\ell_j \in \mathcal{U}$ ) be  $P_j^i$ . For each unlabeled instance  $\ell_j \in \mathcal{U}$ , EMCM calculates

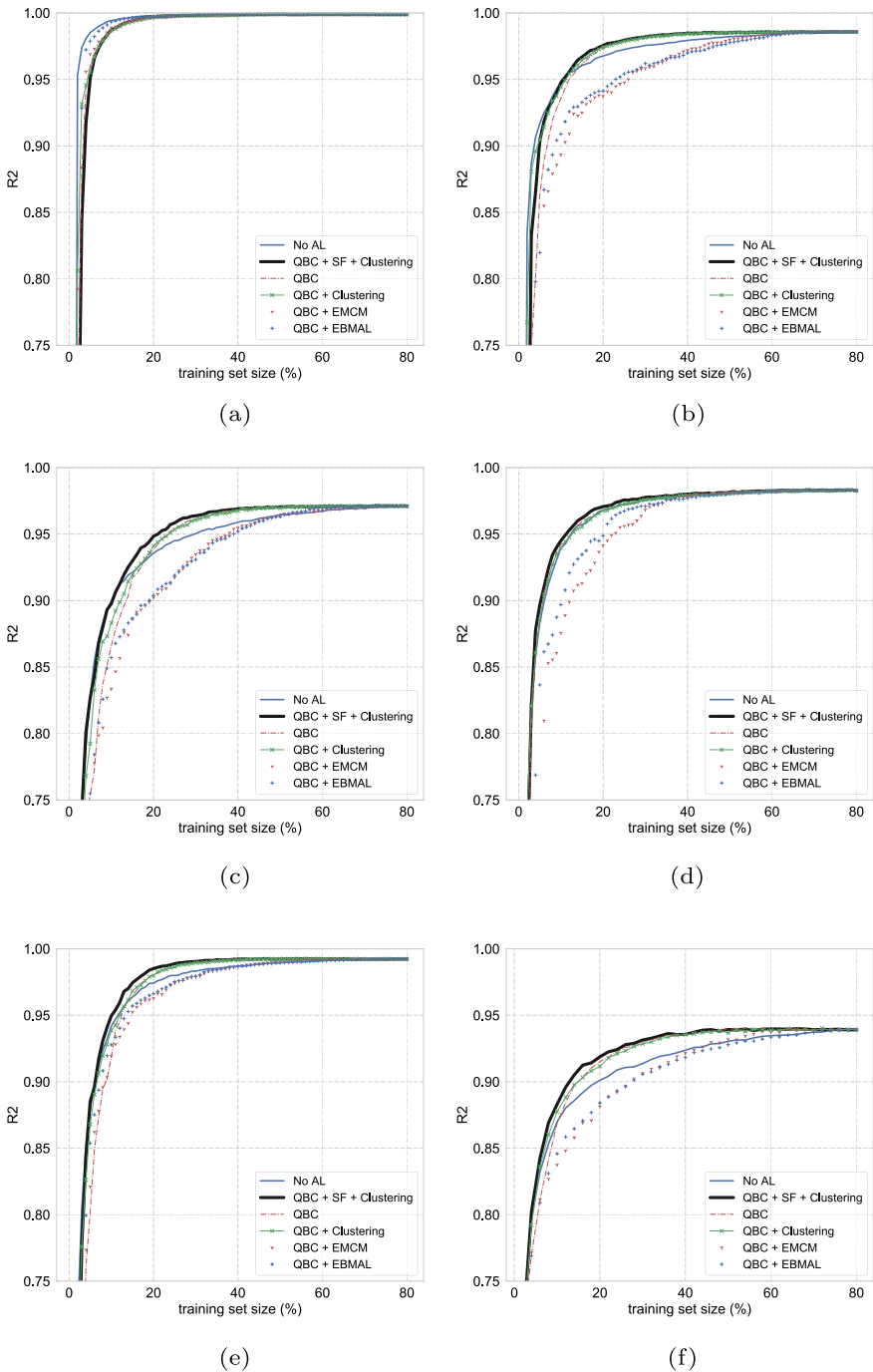
$$g(\ell_j) = \frac{1}{n} \sum_{i=1}^n \left\| (P_j^i - \hat{P}_j) \ell_j \right\|,$$

where  $\hat{P}_j = \frac{1}{n} \sum_{i=1}^n P_j^i$ . The instance with the maximum  $g(\ell_j)$  value is then selected to be labeled.

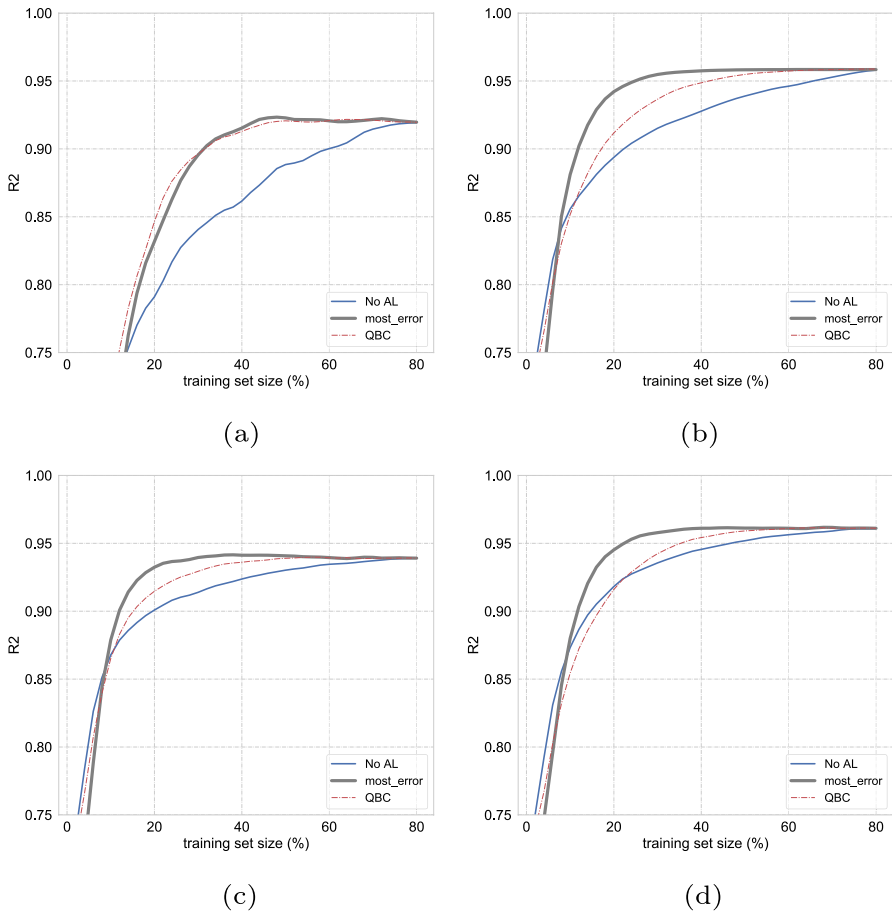
In the case of nonlinear regression, Gradient Boosting Decision Tree regression model is approximated as a linear regression model using feature mapping and an AL algorithm is derived from the resulting linear regression model. In our analysis, we implemented the EMCM that is based on the nonlinear regression.

### A.2 Enhanced Batch-Mode Active Learning

Enhanced Batch-Mode Active Learning (EBMAL) method focuses on representativeness and diversity. As such, it first applies  $k$ -means clustering to select the



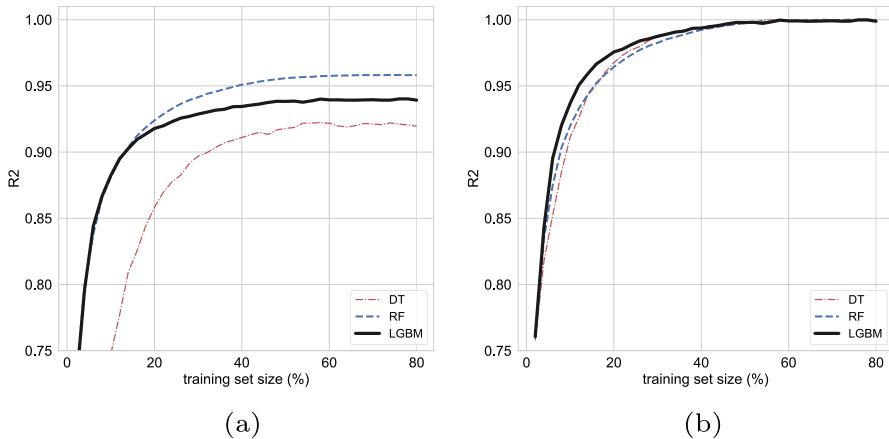
**Fig. 5** Comparison of different AL query strategies (base learner: LGBM, QBC committee: {DT, RF, XGB, LGBM}). (a) POMDP — QALE. (b) POMDP — MRI follow-up ratio. (c) POMDP — ultrasound follow-up ratio. (d) POMDP — detection rate. (e) POMDP — sojourn time. (f) DES — score



**Fig. 6** Performance of the most\_error query strategy (i.e., querying instances with most prediction error) in comparison to random sampling and QBC (Dataset: DES, QBC committee: {DT, RF, XGB, LGBM}). (a) DT. (b) RF. (c) LGBM. (d) XGB

instances to construct an initial training set. Next, by considering a baseline AL regression approach, such as QBC or EMCM, it chooses new instances for labeling. Accordingly, EBMAL approach can be considered as a preprocessing algorithm, where the focus is on the identification of the instances that are likely to be outliers, and hence would not significantly contribute to the learning task.

We also considered other query strategies such as Greedy Sampling (or Passive Sampling) which focuses on geometric characteristics of the instances and selects the instances that are far away from the previously selected and labeled samples. However, these approaches did not yield better results, which thus are disregarded in our comparative analysis for the sake of clarity and brevity.



**Fig. 7** Impact of base learner on active learning (Dataset: DES, QBC committee: {DT, KNN, RF, LGBM}, query strategy: QBC + SF + Clustering). (a) AL performance. (b) Scaled AL performance

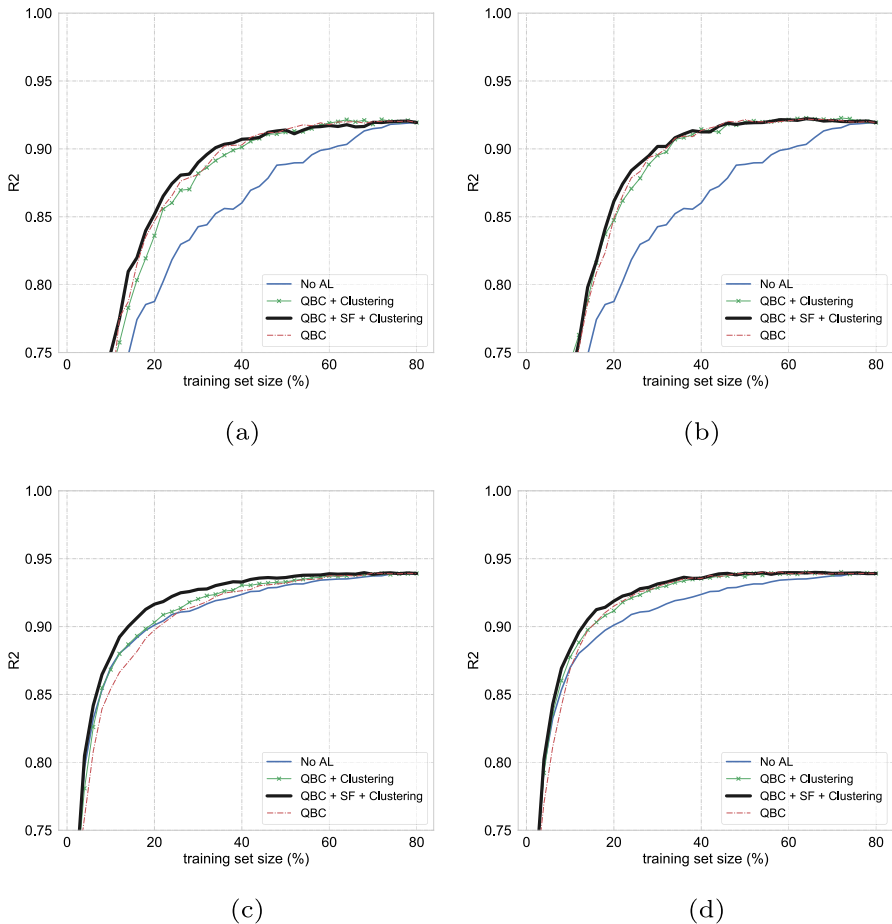
## Appendix : B. Comparison of Different Active Learning Approaches

Figure 5 shows the comparison of supervised learning (No AL) and various AL approaches namely, QBC, QBC with clustering, QBC with Second Filter and clustering, EMCM and EBMAL. We note that QBC alone typically does not perform well and the query strategies integrated into QBC improve the performance considerably. Among these query strategies, we observe that our proposed approach (QBC + SF + Clustering) performs consistently well, and the performance gains over other strategies are more apparent in predicting the outcomes with high variability (MRI and ultrasound follow-up ratios). The performance gains for active learning is more pronounced for DES dataset, and several QBC and QBC enhancements contribute to substantial improvements in convergence.

We next investigate the best achievable performance through careful selection of the instances to be queried. Specifically, we compare random sampling (No AL) and QBC with the case where the instances that lead to highest amount of prediction error are queried (most\_error) at each iteration. Figure 6 shows that, except for DT, prediction models can converge significantly faster if the instances that the model makes the most error in predicting can be successfully identified and queried. However, it is important to note that most\_error query strategy is purely hypothetical and, in practice, it is not possible to perfectly identify the instances with the most prediction error (because this would imply knowing the labels of the instances without actually querying them). These results also show that QBC can be used to approximate the most\_error query strategy to some extent (especially for DT); however, QBC enhancements are needed for querying the instances more efficiently.

We also perform additional experiments to show how active learning algorithm is impacted by various parameter choices including base learner and QBC committee. Figure 7 shows how different machine learning models benefit from active learning. Figure 7b is generated from Fig. 7a by scaling the performance values to the interval





**Fig. 8** Impact of QBC committee on active learning (Dataset: DES, QBC committee 1: {DT, LR, KNN, LGBM}, QBC committee 2: {DT, RF, XGB, LGBM}). (a) DT — committee 1. (b) DT — committee 2. (c) LGBM — committee 1. (d) LGBM — committee 2

[0.75, 1.00]. As demonstrated by the scaled performance ( $R2$ ) values, LGBM model requires the fewest amount of training instances to converge among the tested models that also include DT and RF. These results show that different algorithms benefit differently from active learning.

Figure 8 shows how active learning performance is impacted by the QBC committee. We perform this analysis for two different base learners (DT and LGBM), and two different committee settings (committee 1: {DT, LR, KNN, LGBM}, committee 2: {DT, RF, XGB, LGBM}). Note that two models in committee 1 (LR and KNN) has relatively poor performance than others (see Fig. 3). First, we observe that active learning is more useful for DT as “No AL” leads to a slower convergence in this case. We also note that a better committee (i.e., committee 1)

leads to a better active learning performance which can be seen by the differences between “No AL” and “QBC SF + Clustering” in Fig. 8d. Lastly, comparison between Fig. 8c and d show that a better committee implies a less pronounced effect of QBC enhancements such as SF and Clustering.

**Funding** This research is funded in part by NSERC Discovery Grant.

## Declarations

**Conflict of Interest** The authors declare no competing interests.

## References

- Oakley JE, O’Hagan A (2004) Probabilistic sensitivity analysis of complex models: a Bayesian approach. *J R Stat Soc Series B (Stat Methodol)* 66 (3):751–769
- Czitrom V (1999) One-factor-at-a-time versus designed experiments. *The American Statistician* 53(2):126–131
- Claxton K, Sculpher M, McCabe C, Briggs A, Akehurst R, Buxton M, Brazier J, O’Hagan T (2005) Probabilistic sensitivity analysis for NICE technology assessment: not an optional extra. *Health Econ* 14(4):339–347
- Saltelli A, Tarantola S (2002) On the relative importance of input factors in mathematical models: safety assessment for nuclear waste disposal. *J Am Stat Assoc* 97(459):702–709
- Borgonovo E (2010) Sensitivity analysis with finite changes: An application to modified EOQ models. *Eur J Oper Res* 200(1):127–138
- Razavi S, Jakeman A, Saltelli A, Prieur C, Iooss B, Borgonovo E, Plischke E, Piano SL, Iwanaga T, Becker W et al (2021) The Future of Sensitivity Analysis: An essential discipline for systems modeling and policy support. *Environmental Modelling & Software* 137:104954
- Gupta H, Razavi S (2017) Challenges and future outlook of sensitivity analysis. *Sensitivity Analysis in Earth Observation Modelling* 397–415
- Cevik M, Ergun MA, Stout NK, Trentham-Dietz A, Craven M, Alagoz O (2016) Using active learning for speeding up calibration in simulation models. *Med Dec Making* 36(5):581–593
- Borgonovo E, Plischke E (2016) Sensitivity analysis: A review of recent advances. *Eur J Oper Res* 248(3):869–887
- Pfingsten T (2006) Bayesian active learning for sensitivity analysis. In: *European conference on machine learning*. Springer, 353–364
- Chen Q, Ayer T, Chhatwal J (2017) Sensitivity analysis in sequential decision models: a probabilistic approach. *Med Dec Making* 37(2):243–252
- Zhang Y, Wu H, Denton BT, Wilson JR, Lobo JM (2019) Probabilistic sensitivity analysis on Markov models with uncertain transition probabilities: An application in evaluating treatment decisions for type 2 diabetes. *Health Care Management Science* 22(1):34–52
- Settles B (2009a) *Active Learning Literature Survey: Computer sciences technical report 1648* university of Wisconsin–Madison
- Burbidge R, Rowland JJ, King RD (2007) Active learning for regression based on query by committee. In: Yin H., Tino P., Corchado E., Byrne W., Yao X. (eds) *Intelligent data engineering and automated learning - IDEAL 2007*. ISBN 978-3-540-77226-2. Springer, Berlin, pp 209–218
- Figueroa RL, Zeng-Treitler Q, Ngo LH, Goryachev S, Wiechmann EP (2012) Active learning for clinical text classification: is it better than random sampling?. *J Am Med Inform Assoc* 19(5):809–816
- Lu H, Kocaguneli E, Cukic B (2014) Defect prediction between software versions with active learning and dimensionality reduction. In: *2014 IEEE 25Th international symposium on software reliability engineering*. IEEE, 312–322

17. Settles B (2009b) Active learning literature survey, Tech. Rep. University of Wisconsin-Madison Department of Computer Sciences
18. Seung HS, Opper M, Sompolinsky H (1992) Query by committee. In: Proceedings of the fifth annual workshop on computational learning theory, COLT '92. ISBN 0-89791-497-X. <https://doi.org/10.1145/130385.130417>. ACM, New York, pp 287–294
19. Freund Y, Seung HS, Shamir E, Tishby N (1997) Selective sampling using the query by committee algorithm. *Mach Learn* 28(2):133–168
20. Settles B (2012) Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6(1):1–114
21. Kee S, del Castillo E, Runger G (2018) Query-by-committee improvement with diversity and density in batch active learning. *Inf Sci* 454:401–418
22. Wang M, Min F, Zhang Z-H, Wu Y-X (2017) Active learning through density clustering. *Expert Syst Appl* 85:305–317
23. C Cameron A, Windmeijer FAG (1996) R-squared measures for count data regression models with applications to health-care utilization. *Journal of Business & Economic Statistics* 14(2):209–220. ISSN 07350015. <http://www.jstor.org/stable/1392433>
24. Sandikci B., Cevik M., Schacht D. (2020) Screening for Breast Cancer: The Role of Supplemental Tests and Breast Density Information, Chicago Booth Research Paper (18-03)
25. Fryback DG, Stout NK, Rosenberg MA, Trentham-Dietz A, Kuruchittham V, Remington PL (2006) Chapter 7: The Wisconsin breast cancer epidemiology simulation model. *JNCI Monographs* 2006(36):37–47
26. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V et al (2011) Scikit-learn: Machine learning in Python. *J Mach Learn Res* 12(Oct):2825–2830
27. Wu D (2018) Pool-based sequential active learning for regression. *IEEE Transactions on Neural Networks and Learning Systems* 30 (5):1348–1359

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.