Knowledge-Based Systems 231 (2021) 107400

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

EBOD: An ensemble-based outlier detection algorithm for noisy datasets

Boya Ouyang [a,b], Yu Song [a], Yuhai Li [a], Gaurav Sant [b,c,d,e], Mathieu Bauchy [a,c,*]

[a] Physics of AmoRphous and Inorganic Solids Laboratory (PARISlab), Department of Civil and Environmental Engineering, University of California , Los Angeles, CA, USA
[b] Laboratory for the Chemistry of Construction Materials (LC^2), Department of Civil and Environmental Engineering, University of California, Los Angeles, CA, USA
[c] Institute for Carbon Management (ICM), University of California, Los Angeles, CA, USA
[d] Department of Materials Science and Engineering, University of California, Los Angeles, CA, USA
[e] California Nanosystems Institute, University of California, Los Angeles, CA, USA

ARTICLE INFO

Article history:
Received 8 February 2021
Received in revised form 16 July 2021
Accepted 12 August 2021
Available online 16 August 2021

Keywords:
Outlier detection
Data cleansing
Machine learning
Concrete strength

ABSTRACT

Real-world datasets often comprise outliers (e.g., due to operational error, intrinsic variability of the measurements, recording mistakes, etc.) and, hence, require cleansing as a prerequisite to any meaningful machine learning analysis. However, data cleansing is often a laborious task that requires intuition or expert knowledge. In particular, selecting an outlier detection algorithm is challenging as this choice is dataset-specific and depends on the nature of the considered dataset. These difficulties have prevented the development of a "one-fits-all" approach for the cleansing of real-world, noisy datasets. Here, we present an unsupervised, ensemble-based outlier detection (EBOD) approach that considers the union of different outlier detection algorithms, wherein each of the selected detectors is only responsible for identifying a small number of outliers that are the most obvious from their respective standpoints. The use of an ensemble of weak detectors reduces the risk of bias during outlier detection as compared to using a single detector. The optimal combination of detectors is determined by forward–backward search. By taking the example of a noisy dataset of concrete strength measurements as well as a broad collection of benchmark datasets, we demonstrate that our EBOD method systematically outperforms all alternative detectors, when used individually or in combination. Based on this new outlier detection method, we explore how data cleansing affects the complexity, training, and accuracy of an artificial neural network.

1. Introduction

The recent growth of machine learning approaches is rapidly reshaping our understanding of the unknown [1–4]. As an alternative route to the long-established ways to develop our cognition based on the progressive accumulation of knowledge, machine learning algorithms approach a puzzle directly from an ensemble of existing data [5]. As such, machine learning has changed engineering practices and offers practical (and often surprisingly accurate) solutions to problems that, previously, required experience, intuition, or theoretical knowledge [6]. Since machine learning approaches solely rely on the analysis of data, their outcomes are unsurprisingly strongly affected by the size, distribution, and quality of the dataset [7]. In particular, many studies have stressed the importance of data quality for the success of a machine learning analysis [8–12].

In that regard, unreliable, inaccurate, or noisy data often hinders the learning efficiency of a model and, in extreme cases, can even mislead the learning process and result in biased predictions [6]. This is a serious issue as datasets for engineering applications are often based on experimental observations and, hence, can exhibit various types of imperfections, e.g., experimental errors, uncertainties regarding the tested system, variability resulting from the effect of missing features, data entry error, etc. Such datapoints are usually referred to as "outliers" [13]. Importantly, if numerous enough, the presence of outliers in a dataset, can negatively impact machine learning models, which, in turn, can reduce the trust of the public, industry, and governmental agencies in machine learning approaches. As such, proper data cleansing is often a prerequisite to any machine learning analysis. However, it should be noted that "extreme datapoints" that are simply far from the distribution of most of the observations are not always detrimental and can even be extremely informative—as they sometimes capture behaviors in regions of the features space that are poorly sampled.

[*] Correspondence to: 5731-E Boelter Hall, Los Angeles, CA, 90095-1593, USA.
E-mail address: bauchy@ucla.edu (M. Bauchy).

https://doi.org/10.1016/j.knosys.2021.107400

To reduce the impact of outliers on machine learning models, two solutions are commonly adopted: (1) expanding the dataset to minimize the weight of outliers and (2) identifying outliers and excluding them from the dataset [14]. The first option is often not practical for reasons of time, cost, or unavoidable uncertainties during data collection. Hence, it is of special importance to find efficient ways to detect outliers in datasets. To this end, various outlier detection methods are available. Simple approaches are based on identifying points that are far away from an expected pattern (e.g., Gaussian distribution) [15], further than n standard deviations away from the mean [16], or beyond the interquartile range, as defined with boxplots [17]. More advanced outlier detection algorithms have been developed in the field of data mining. Many studies have focused on data clustering [12,18–20], where isolated clusters are considered as outliers. Alternatively, outlier detection approaches can be based on the analysis of the distance [21], density [22], or angle [23] between datapoints.

As non-parametric methods, the detection methods mentioned above facilitate the data cleansing process as they can identify outliers without the need for any assumptions regarding the nature of the data distribution. However, these detection algorithms are sensitive to hyperparameter settings, e.g., the number of neighbors for cluster-based algorithms. In detail, outlier detection algorithms usually rely on the fairly arbitrary choice of a "threshold" value in discriminating outliers from non-outlier datapoints [24]. Selecting the optimal threshold is often a complex choice—as a loose threshold may not properly detect outliers, whereas, in turn, a strict threshold may result in the removal of valuable information from the dataset. Further, their performance can largely depend on the spatial distribution of the data points (i.e., depending on the sparsity and homogeneity of the dataset). Due to these difficulties, in practice, no single detection algorithm can universally apply to all datasets. For instance, cluster-based algorithms do not perform well in highly-dimensional spaces, since the sparse distribution of the datapoints makes it unlikely for locally-clustered datapoints to be found [25]. Since each outlier detection algorithm comes with strengths and weaknesses, selecting a detector often requires some level of intuition or knowledge on the nature of the dataset—since each outlier detection algorithm comes with its own definition regarding how outliers differ from normal datapoints. For all these reasons, data cleansing is often highly subjective.

As an alternative route to individual detectors (i.e., that rely on a single algorithm), combining a number of individual detectors (i.e., base learners) into an ensemble-based detector can efficiently address the limitations raised above. Over the past recent years, a number of ensemble-based outlier detection algorithms have been proposed to improve the detection accuracy and robustness of data cleansing pipelines—especially for the case of noisy datasets wherein single detectors tend to be less reliable [26–28]. This approach typically relies on (i) a collection of dissimilar "base learners" (i.e., outlier detection algorithms) and (ii) a "combiner" that selects a set of optimal base learns and fuses their outcomes. However, combining dissimilar base learners into an ensemble of detectors is a tricky task that requires intuition or expert knowledge—since each single detector can dramatically affect the performance of the ensemble [28]. Indeed, poorly-performing base detectors can substantially weaken the accuracy of ensemble-based detectors (e.g., by removing non-outlier datapoints that are actually informative) [29]. As a result, outlier detection algorithms usually need to be adjusted, replaced, or recombined dynamically from one dataset to the other.

Unfortunately, over the past years, far more attention has been placed on designing complex regression/classification machine learning algorithms than on developing robust, non-biased outlier detection methods—so that one may argue that outlier detection might be the actual bottleneck of many machine learning applications (rather than more complex machine learning algorithms or increased numbers of datapoints). Here, as a steppingstone toward this end, we propose an unsupervised, ensemble-based outlier detection (EBOD) approach that automatically determines the optimal the union of different outlier detection algorithms—wherein each outlier detector is used to solely detect the most extreme outliers (based on how each detector define outliers). Specifically, the EBOD approach considers a pool of individual outlier detection algorithms as base learners. A combiner relying on a forward–backward search then identifies the optimal set of base learners. The use of such an ensemble of loose detectors reduces the risk of bias during outlier detection as compared to data cleansing conducted based on a single detector.

To illustrate this approach, we apply EBOD on a series of regression tasks. As a measure of performance of data cleansing, we compare the test set accuracy of a base machine learning model on a large number of regression datasets before and after data cleaning—so as to assess the ability of the EBOD detector to generalize. First, we consider the example of a noisy dataset of production concrete strength measurements previously presented in Ref. [30]. This dataset comprises concrete mixing proportions (as inputs) and associated measured strength after 28 days (as output) for more than 10,000 concrete samples. Further, to demonstrate the generic nature of the EBOD method, we apply it to a selection of ten additional benchmark regression datasets and evaluate its cleaning performance based on the test set regression accuracy of the same learning algorithm. We demonstrate that our EBOD outlier detection method systematically outperforms all detectors in terms of test set regression accuracy after data cleaning, when used individually or in combination (based on a comparison with several other prevailing ensemble-based outlier detectors). Based on this new outlier detection method, we also explore how data cleansing affects the complexity, training, and accuracy of the machine learning model trained based on the concrete dataset.

2. Methodology

2.1. Datasets

2.1.1. Concrete dataset

To illustrate our EBOD outlier detection approach, we consider the concrete strength regression dataset described in Ref. [30,31]. This dataset comprises a total of 10,264 concrete strength measurements, which are sourced from real concrete production without any pre-cleaning. It should be noted that concrete is by far the most manufactured material in the world and, hence, accurately predicting its strength is critical to ensure the integrity of the built environment [32]. Concrete takes the form of a mixture of cement, water, sand (fine aggregates), stones (coarse aggregates), supplementary cementitious materials (e.g., fly ash), and chemical additives [33]. To the first order, the strength of a given concrete depends on the mixing proportions of these raw ingredients [34]. As such, the regression dataset considered herein presents the following inputs: (1) water-to-cementitious ratio, i.e., the ratio between the mass of water and that of the cementitious materials (here, cement and fly ash), (2) cement mass fraction, (3) fly ash mass fraction, (4) fine aggregate mass fraction, (5) coarse aggregate mass fraction, (6) dosage of air-entraining chemical admixture, and (7) dosage of water-reducing chemical admixture. The output is the concrete strength, measured 28 days after production following ASTM C39 [35]. The 28-day strength of concrete is indicative of its long-term strength and largely dictates its performance (and price).

This concrete dataset exemplifies many difficulties associated with real-world regression datasets. For example, strength measurements exhibit some intrinsic variability, which makes it hard

Table 1
Benchmark regression datasets considered in this study.

Data	Instances	Input features	Reference
Real estate	414	8	[38]
Delta Elevators	9,517	7	[39]
Qsar fish toxicity	908	7	[40]
California Housing	20,640	9	[41]
Red wine	1,599	12	[42]
Boston Housing	506	14	[35]
UCI concrete	1,030	9	[43]
Abalone	4,177	9	[44]
Ailerons	7,154	41	[45]
Airfoil noise	1,503	6	[46]

to discriminate outliers from legitimate measurement variabilities [36]. Outliers may also result from data entry typos or experimental errors, such as errors in mixing proportions (e.g., excess of water). Strength measurements can also be affected by external factors that are not captured by the present features (e.g., temperature, relative humidity, raw material quality, mixing protocols, etc.). As such, this dataset offers an ideal, archetypal, and challenging basis to illustrate our EBOD outlier detection approach.

2.1.2. Additional benchmark regression datasets

To demonstrate the generic nature of the proposed EBOD approach, we consider a series of ten additional benchmark regression datasets. The benchmark datasets are listed in Table 1. These datasets were selected to encompass a broad variety of size (number of data points), dimensionality (number of input features), and data distribution. The Red wine, Housing, UCI concrete, Qsar fish toxicity, Airfoil noise, and Abalone datasets are sourced from the UCI repository [37], while the California Housing, Ailerons, and Delta Elevators datasets are obtained from Luís Torgo's repository (https://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html).

2.2. Artificial neural network model

We train an artificial neural network (ANN) regression model aiming to predict concrete strength as a function of the mixture proportions, as well as outputs of the datasets listed in Table 1. Although ANN may not offer the highest accuracy for all these datasets [31], we select this regressor based on its sensitivity to outliers [47]. The ANN model is implemented and trained by using Scikit-learn [48]. We adopt resilient backpropagation to optimize the model parameters [45]. For simplicity, we restrict the ANN model to a single hidden layer and use a sigmoid as activation function. During training, the model is iteratively updated using a stochastic gradient descent optimizer until the relative change of the loss (here, the mean-square error) becomes minuscule ($< 10^{-4}$). Once trained, the performance of the ANN model is evaluated based on the root-mean-square error (RMSE) and coefficient of determination ($R2$). Here, the RMSE is the averaged Euclidian distance between predicted and measured data and R^2 quantifies their corresponding degree of scattering.

Prior to any training, all the datasets are randomly divided into a training set (80% of the datapoints), which is used to train the model, and a test set (remaining 20% of the datapoints), which is kept invisible to the model during its training and, eventually, is used to assess its ability to generalize to unknown samples. The model hyper-parameters are then optimized based on the concrete dataset. To this end, we implement five-fold cross-validation within the training set [49]. In this study, the only hyperparameter that is considered is the number of neurons in the single hidden layer—wherein a deficit of neurons results in a simple model that is prone to underfitting (high bias), whereas an excess of neurons leads to an unnecessarily complex model that exhibits overfitting (high variance) and poorly generalizes to new samples that are not included in the training set. The optimal number of hidden neurons (and the dependence thereof on the presence of outliers) is determined based on the average cross-validation RMSE (see below). To ensure a meaningful comparison among different benchmark datasets, we then keep the same ANN model (number of hidden layers and neurons, hyperparameters, etc.) for all the other datasets.

Table 2
Individual outlier detection algorithms considered as base learners for constructing the proposed EBOD ensemble-based outlier detector.

Detection algorithm	Description	Reference
KNN	K-nearest neighbors	[21]
LOF	Local outlier factor	[22]
COF	Connectivity-based outlier factor	[50]
OCSVM	One-class support vector machine	[51]
IFOREST	Isolation forest	[52]
ABOD	Angle-based outlier detection	[23]
SOS	Stochastic outlier selection	[53]

2.3. Collection of base outlier detection algorithms

In this study, we introduce an ensemble-based outlier detection (EBOD) method that is based on an optimized combination of several base detectors. Our approach combines these base detectors so as to yield an optimal detection of outliers in various datasets. To this end, we select seven common outlier detection algorithms as the base learners, which are listed in Table 2. This selection is based on the wide acceptance, simplicity of implementation, complementarity, and variety of these algorithms.

Among the individual base learners shown in Table 2, the LOF, KNN, and SOS detectors can be classified as belonging to the family of distance-based algorithms, but differ in their approach and mathematical basis for identifying outliers. LOF approaches the problem from the concept of local density (which is estimated by the distance over which a point can be reached by its neighbors) since outliers tend to reside in low-density regions. Outliers are defined as the points that exhibit a density of neighbors that is low enough. Likewise, KNN evaluates the average distance between a central data point and its k nearest neighbors and scores its probability of being an outlier based this distance. The detection offered by SOS is based on the concept of affinity. This algorithm first computes the distance matrix of feature vectors for a datapoint, and then transforms this distance matrix into an affinity matrix. As such, outliers are defined as points showing a low affinity with the other datapoints.

The other algorithms are rooted in alternative viewpoints regarding what differs outliers from normal datapoints. In that regard, ABOD detects the outliers based on the weighted variance of the angles between a datapoint and its neighbors—wherein outliers are defined as datapoints that are far from the majority of the other data points in the hyperspace, with a low variance of the angles. This algorithm is efficient for identifying outliers in high-dimensional space by alleviating the curse of dimensionality [23]. COF identifies outliers based on the degree of connection of a datapoint. IFOREST carries out the detection using a tree-based model, wherein outliers are more likely to be isolated near the root of the tree. Finally, OCSVM relies on a support vector machine to draw the boundary segregating true datapoints from anomalies.

2.4. Description of the EBOD combiner

As the core component of our proposed EBOD approach, we implement as combiner a forward–backward search approach aiming to pinpoint the optimal combination of the base detectors for flagging outliers across datasets. This method relies on the following steps. First, to avoid any bias regarding the choice of the threshold value to be used for each algorithm, the sensitivity of each outlier detector is tuned so as to systematically flag 10% of the datapoints as outliers. This aims to ensure that each detector identifies a small, constant fraction of the datapoints as being abnormal. Second, the performance of each single outlier detector (when used individually) is evaluated by comparing the test set R^2 of the base ANN model (see Section 2.2) before and after removing the detected outliers—wherein outlier detector featuring good generalizability is expected to notably increase the test set R^2. It should be noted that a single detector can either result in an increase in the test set R^2 (if it successfully removes abnormal datapoints) or, potentially, in a decrease in the test set R^2 (if it actually removes useful information, which harms the training of the model). The detectors are then ranked in terms of test set R^2 (i.e., from the best to the worse detector). Third, we conduct the forward–backward search to identify the optimal combination of these outlier detectors, as detailed below.

The general algorithm of the forward–backward search used herein as combiner is summarized in Fig. 1. After determining the cleaning effect of the individual detectors (based on the test set R^2 of the ANN model), we first conduct a forward search (Fig. 1a). This search comprises the following steps: (i) assess the model accuracy by removing the outliers identified by each of the detectors in the algorithm pool, P, in a one-by-one fashion, (ii) add the best detector to the detector ensemble, U; (iii) remove the union of the outliers identified by the selected detectors in U; (iv) calculate the test set model accuracy based on the cleaned dataset; and (v) repeat the above steps iteratively until the model accuracy does not improve any further. In parallel, we conduct a backward search, which basically mirrors the forward search, namely, starting with all the detectors being in the ensemble U, we remove one detector at a time by systematically selecting the action that yields the largest increase in the test set model accuracy. As such, we track the evolution of the model test set accuracy during both the forward and backward processes, and, based on this information, we select the optimal set of detectors as the one that maximizes the test set R^2 of the ANN model during the forward–backward search.

2.5. Alternative benchmark ensemble-based detectors

To illustrate the ability of the new EBOD data cleansing method (see Section 2.4) to robustly identify outliers, we compare it with alternative ensemble-based outlier detection methods listed in Table 3. The characteristics of each alternative ensemble-based outlier detector is briefly summarized as follows. The Averaging algorithm attributes an outlier score to each datapoint based on the average of the scores yielded by each individual detector [26]. In contrast, Maximization defines the final score as the maximum of the scores offered by the detectors [54]. Building on these two ideas, AOM further introduces a bootstrap process, wherein the base individual detectors are first randomly divided into predefined subgroups and the final score is calculated by averaging the maximum scores within each subgroup [54]. Similarly, MOA defines the final score as the maximum of the average scores within each subgroups [54]. Feature Bagging combines the outcome of several base outlier detection algorithms by fitting them on random subset of features [55]. LODA identifies outliers by modeling the probability of observed samples based on a collection of one-dimensional histograms. Each one-dimensional histogram is weak in detecting outlies, but the combination of these weak detectors eventually results in a strong anomaly detector [56]. LSCP is based on the idea that outliers located in distinct regions of the feature space are likely to be properly identified by different individual detectors. As such, this algorithm evaluates the competency of each individual base detector in identifying outliers within a given local region and subsequently combines the top-performing detectors for each region as the final output [57]. SUOD initially fits unsupervised base detectors on randomly projected feature space (like Feature bagging). It then evaluates the computational cost of each base model and replace the costly model with a faster supervised regression model, which can increase interpretability and reduce storage costs [58]. The last algorithm considered herein, a recent outlier ensemble method AKPV (named after the authors of the source paper), combines individual detectors by averaging the scores of three detectors that have best performance [29]. For consistency, the implementation of all the above ensemble-based detectors relies on the same pool of individual base detectors, as introduced in Section 2.3. To ensure a meaningful comparison, we tune the detection parameters used the ensemble-based detectors such that they yield a number of outliers that is identical to that offered by our new EBOD method. In addition to these unsupervised detectors, many supervised ensemble-based detectors that have been developed over the past years, e.g., Bagged Outlier Representation Ensemble (BORE) [59] or Extreme Gradient Boosting Outlier Detection (XGBOD) [60]. However, these supervised approaches are not considered herein since, in the case of the present dataset (as well as in many other engineering datasets), the nature of the outliers is not *a priori* known.

Table 3
Summary of the previously proposed ensemble-based outlier ensemble algorithms that are considered herein as benchmarks to quantify the performance of the proposed EBOD ensemble-based outlier detector.

Ensemble-based detection algorithm	Description	Reference
Averaging	Simple combination by averaging the scores	[26]
Maximization	Simple combination by taking the maximum scores	[54]
AOM	Average of Maximum	[54]
MOA	Maximum of Average	[54]
Feature bagging	Combine multiple outlier detection algorithms using different set of features.	[55]
LODA	Lightweight On-line Detector of Anomalies	[56]
LSCP	Locally Selective Combination of Parallel Outlier Ensembles	[57]
SUOD	Large-Scale Unsupervised Heterogeneous Outlier Detection	[58]
AKPV	Average the scores of top three outlier detectors	[29]

2.6. Non-parametric statistical tests

To ensure the statistical significance of the comparison between the performance of the present EBOD approach and that offered by nine alternative outlier detection methods (Table 3), we carry out two non-parametric tests—also referred as distribution-free tests, which do not assume that the data is normally distributed. To ensure the generality of our results,

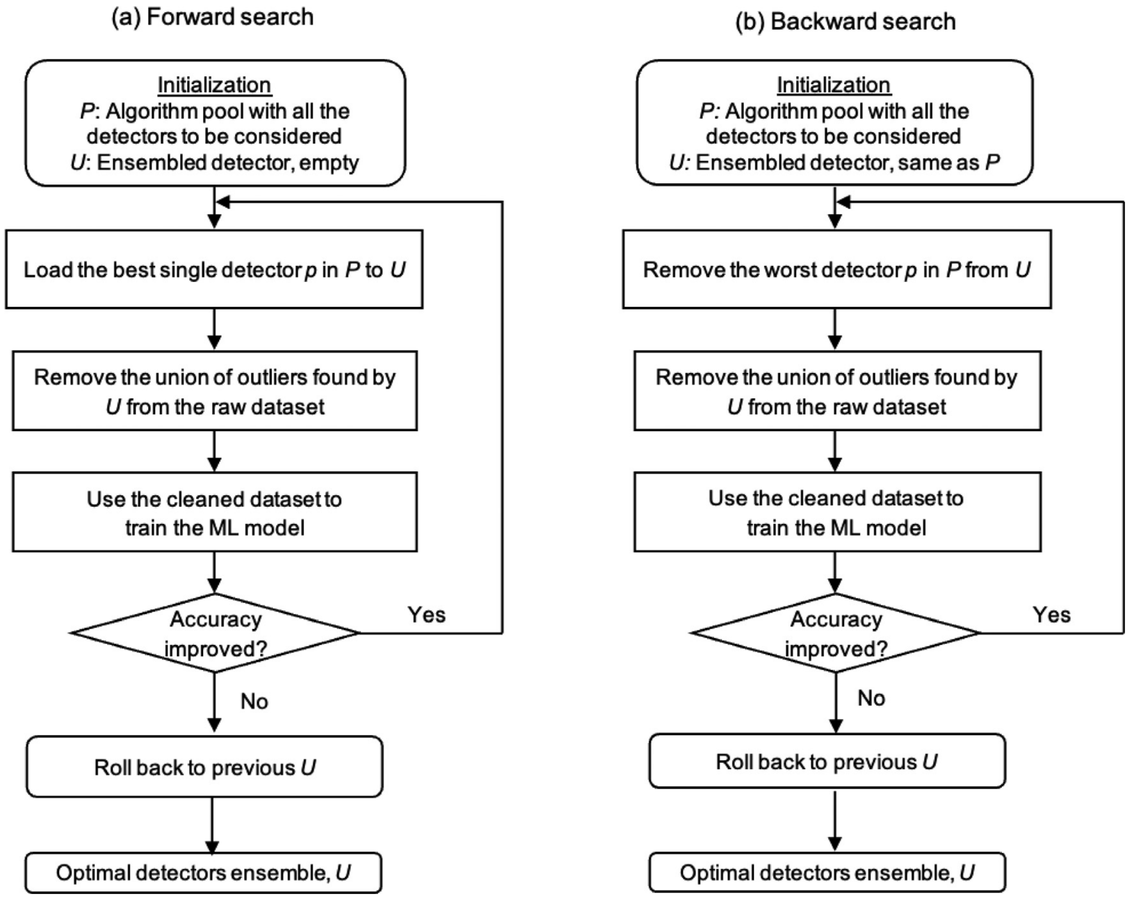
(a) Forward search
Initialization
P: Algorithm pool with all the detectors to be considered
U: Ensembled detector, empty
Load the best single detector p in P to U
Remove the union of outliers found by U from the raw dataset
Use the cleaned dataset to train the ML model
Accuracy improved?
Yes
No
Roll back to previous U
Optimal detectors ensemble, U
(b) Backward search
Initialization
P: Algorithm pool with all the detectors to be considered
U: Ensembled detector, same as P
Remove the worst detector p in P from U
Remove the union of outliers found by U from the raw dataset
Use the cleaned dataset to train the ML model
Accuracy improved?
Yes
No
Roll back to previous U
Optimal detectors ensemble, U

Fig. 1. Flowchart illustrating the (a) forward and (b) backward searching processes for determining the optimal combination U of the detectors in our ensemble-based outlier detection (EBOD) method.

we further extend the analysis to a series of ten additional benchmark regression datasets (see Table 1). We first implement the Friedman test [61]. Starting from ten groups of data (i.e., the ten regression datasets considered herein) and ten treatments (i.e., the ten ensemble-based detectors), the Friedman test first ranks the performance achieved for each dataset/treatment combination and then computes the summed ranking for each treatment. The Friedman test statistics are then used for calculating the p-value—wherein a p-value that is smaller than 0.05 indicates that at least one of the treatments is statistically different from the others. To quantify how the performance of EBOD compares to that of the other methods, a post hoc Dunn's test [62] is also performed based on the mean rank differences provided by the Friedman's test. The Dunn's test runs multiple pairwise comparisons using Z-test statistics, which can be used to obtain the p-value for each comparison (i.e., for each pair of detectors) [63]. A p-value that is smaller than a certain threshold (typically 0.05) indicates the existence of a statistically significant difference of performance between the compared pair of detectors.

3. Results and discussion

3.1. Performance of the individual outlier detectors

First, we evaluate the performance of individual (i.e., non-ensembled) outlier detectors based on the concrete dataset. For each outlier detector, the threshold value is adjusted so as to identify (and remove) 10% of the datapoints from the dataset (i.e., 1,027 observations). The ability of each detector to increase the predictive accuracy of the ANN model (as compared to that of the ANN model trained based on uncleaned data) is presented in Table 4. We find that the COF, SOS, LOF, and ABOD detectors tend to improve prediction accuracy, both for the training and test sets. In contrast, the OSCVM, IFOREST, and KNN detectors result in a decrease in the accuracy of the ANN model. This exemplifies the fact that removing datapoints from a dataset can be beneficial or detrimental—since removing outliers can either filter out the noise of the dataset or remove useful information. Among all these detectors, we find that the ABOD algorithm offers the largest increase in the test set R^2 (from 0.49 to 0.54, i.e., a 10% increase). The high performance of the ABOD algorithm for the present dataset may arise from the fact that, in high-dimension spaces, the concept of "angle" between datapoints is more meaningful than those of proximity or distance [23]. A detailed inspection of the results reveals that, even though the COF, SOS, LOF, and ABOD detectors all have a positive effect on the accuracy of the ANN model, the outliers that are detected by each algorithm are (unsurprisingly) not the same. This suggests that combining several detectors may further increase the accuracy of the ANN model—which is the basis of the EBOD method presented herein.

3.2. Determination of the optimal union of detectors

We now assess the effect of combining detectors, which is at the core of our ensemble-based EBOD approach. To this end, Fig. 2 shows the evolution of the accuracy of the ANN model during the forward–backward search in the case of the concrete dataset (see Section 2.4). Note that, for illustration purposes, we

Table 4
Coefficient of determination (R^2) of the artificial neural network considered herein before and after removing 10% of the datapoints from the dataset (1027 samples) based on the selection of outliers identified by a series of detectors. Results are presented for the concrete dataset (see Section 2.1).

R^2 accuracy	Original dataset	Outlier detector that is used to remove outliers						
		OSCVM	KNN	IFOREST	SOS	COF	LOF	ABOD
Training set	0.53	0.33	0.39	0.44	0.53	0.54	0.53	0.55
Test set	0.49	0.31	0.41	0.42	0.51	0.52	0.52	0.54

continue the forward and backward searches even after finding the optimal combination of detectors, that is, even after the accuracy reaches a maximum and starts to decrease (which is slightly different from the algorithm described in Fig. 1). This aims to illustrate how the optimal set of detectors yield maximum accuracy. Interestingly, we note that the series of detectors that are iteratively selected at each step of the search does not systematically follow their ranking, when used as single detector (see Table 1). This highlights the existence of some combined effects, wherein a given detector may not exhibit notable benefits when used alone, but can positively complement other detectors when used in pairs. Overall, we find that the optimal ensemble of detectors consists in using the union of ABOD, COF, SOS, and LOF. Importantly, both the forward and backward search yield the same optimal ensemble, which confirms the robustness of the present EBOD approach. The optimal ensemble of detectors results in a significant increase in the accuracy of the ANN model, which increases from 0.53 to 0.63 and from 0.49 to 0.60 for the training and test set R^2, respectively.

Fig. 3 illustrates the combined evolution of model accuracy and dataset size (i.e., based on the number of removed outliers) during the forward–backward search (in the case of the concrete dataset). As expected, each iteration of the forward and backward search reduces and increases the size of the dataset, respectively. Nevertheless, we note that the number of outliers that are removed at each iteration is not constant. This illustrates the fact that, during the forward search, the first detector already removes most of the outliers, while each subsequent detector adds its own contribution. The contribution of each detector tends to decrease over time as the dataset gradually runs out of "true" outliers, which manifests itself by a gradual decrease in the number of outliers that are removed at each iteration, as well as a gradual decrease in the associated increment in accuracy. At some point, the search approach leads to excessive removal of outliers and, hence, results in the disappearance of some useful information from the dataset—which, in turn, negatively affects the accuracy of the ANN model. Overall, we find that optimal performance is achieved after removing 2,645 data points (i.e., about 25%) from the dataset.

3.3. Influence of data cleansing on model complexity

Having established our EBOD data cleansing approach, we then discuss how the removal of outliers affects the optimal degree of complexity of the ANN model trained based on the concrete dataset. To this end, we conduct a comparative hyperparameter optimization, both before and after data cleansing. This is achieved by five-fold cross-validation, wherein we train based on the concrete dataset a series of ANN models with varying number of hidden layers (the sole hyperparameter considered herein, for simplicity). Fig. 4 shows the evolution of the training and validation set RMSE as a function of the number of hidden neurons. As expected, we note that, independently of whether data cleansing is conducted or not, increasing the number of hidden neurons systematically results in a decrease in the training set RMSE. This signals the fact that, as the model becomes more complex, it gradually manages to better interpolate all the details of the training set [64]. Similarly, the validation set RMSE initially decreases upon increasing number of hidden neurons. In this regime, the model is underfitted and exhibits high bias, which is evident from the fact that the RMSE of the training and validation sets are both high and equal to each other. However, in contrast to the RMSE of the training set, the validation set RMSE eventually does not decrease any further and exhibits a plateau. In this regime, the difference between the RMSEs of the training and validation set suggests that the model becomes overfitted. Based on this analysis, we select the optimal number of neurons (a measure of model complexity) as the minimum number of neurons that yields a validation set RMSE that is less than one standard deviation away from the minimum RMSE [65], wherein the standard deviation is calculated based on the various RMSE values obtained during cross-validation.

Based on this analysis, we assess how data cleansing affects the optimal complexity of the model trained based on the concrete dataset (see Fig. 4a and Fig. 4b). We note that the ANN model trained with the cleaned dataset systematically achieves lower RMSE values than its counterpart trained with the non-cleaned dataset—both for the training and validation sets. This indicates that, at an equivalent degree of complexity (i.e., constant number of neurons), the model trained based on cleaned data systematically outperforms the one that is trained on the raw dataset. In addition, we find that the standard deviation of the validation set RMSE (represented by the light blue shaded area in Fig. 4) is systematically larger when the model is trained based on the uncleaned dataset. This suggests that, based on the folds used for training, the presence of outlier greatly impact the training of the model and its ability to generalize well so as to reliably and consistently predict the validation set. Importantly, we observe that the plateau of the validation set RMSE occurs sooner in the case of the cleaned dataset. In fact, we find that the optimal number of hidden neurons prescribed by the present analysis is 15 and 9 before and after cleansing, respectively. This indicates that data cleansing reduces the optimal degree of complexity of the ANN model. This can be understood from the fact that, when trained from the raw dataset, the model does not properly capture the intrinsic relationship between inputs and output and tends to become more complex than necessary so as to capture some fluctuations in the training set induced by the presence of outliers (see below for more detail on this). In the following, we fix the number of neurons in the hidden layer to these optimal values.

3.4. Influence of data cleansing on learning efficiency

Next, we further investigate how the presence of outliers negatively affects the learning process of the ANN model in the case of the concrete dataset. To this end, we compute the learning curve of the model, both before and after data cleansing. This is achieved by iteratively training the ANN model based on increasing fractions of the training set and subsequently testing its prediction based on the same test set. To enhance the statistical significance of this analysis, the analysis is repeated five times (based on different random training-test splits). The resulting learning curves are shown in Fig. 5. As expected, the training set RMSE is initially low and then gradually increases with the number of training examples. This is a consequence of the fact that it becomes harder and harder for the model to perfectly interpolate the training set (with a fixed, limited number of adjustable parameters). In contrast, the test set RMSE gradually decreases with the number of training examples—since the model

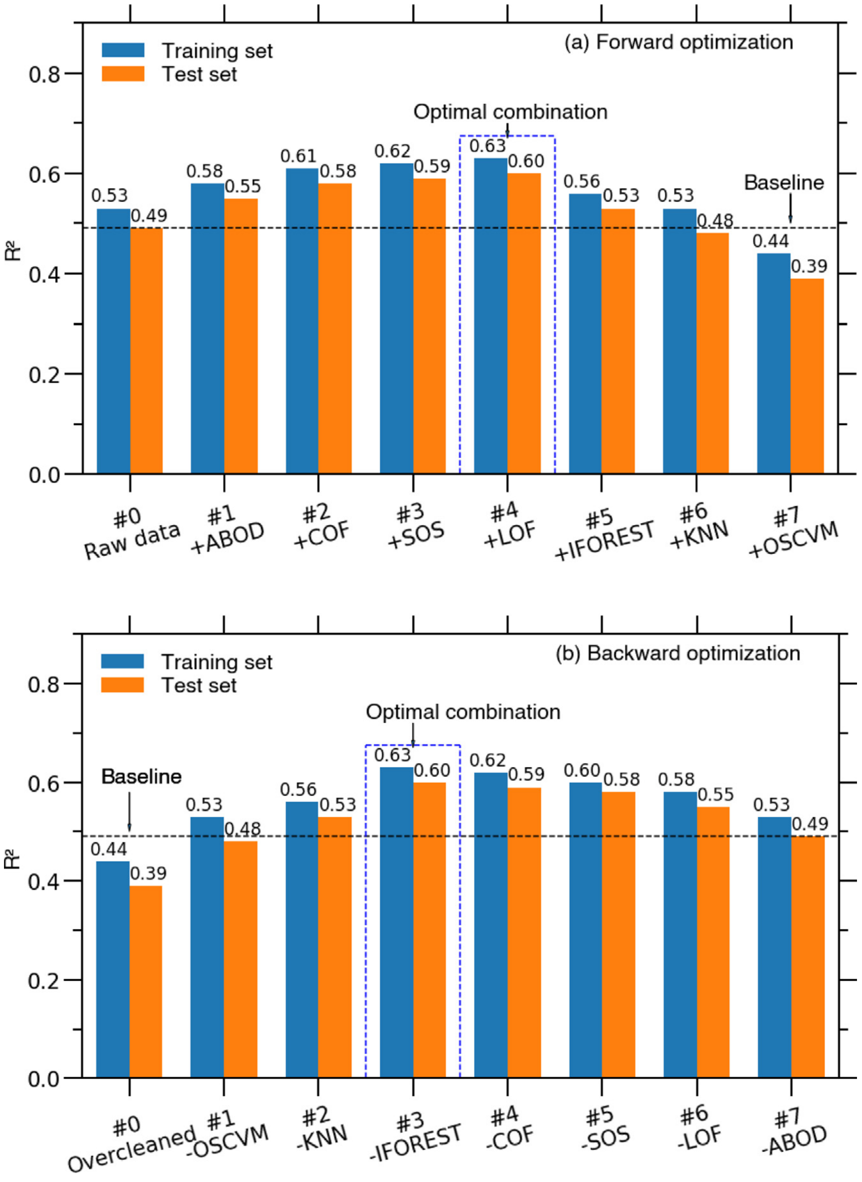

Fig. 2. Coefficient of determination (R^2) achieved by the artificial neural network considered herein for the training and test sets at each iteration of the (a) forward and (b) backward searching process. Results are presented for the concrete dataset (see Section 2.1). Note that, here, to illustrate how the optimal set of detectors yield maximum accuracy, we continue the forward and backward searches after finding the optimal combination of detectors, that is, even after the accuracy reaches a maximum and starts to decrease (which is slightly different from the algorithm described in Fig. 1).

gradually learns how to properly generalize to unknown observations. Irrespectively of whether data cleansing is conducted or not, we note that the training and test set RMSE eventually converge toward a fairly similar value, which confirms that these models do not exhibit any significant degree of overfitting. Note that the maximum size of the training set is smaller in the case of the cleaned data since a given fraction of the datapoints is flagged as outliers and removed.

By comparing Fig. 5a and b, we observe that, at fixed number of training examples, the training set RMSE is systematically lower after cleansing. This confirms that, despite remaining fully unsupervised, our EBOD outlier detection indeed removes points that are far away from the interpolated model. This suggests that the points that are removed indeed act as true outliers. Similarly, the test set RMSE is systematically lower after cleansing, which suggests that the removal of the outliers enhances the ability of the model to learn how to properly generalize to unknown observations. Finally, we find that the ANN model converges faster toward its optimal accuracy (i.e., after being exposed to a lower number of training examples) when trained based on the cleaned dataset. This demonstrates that proper data cleansing effectively reduces the number of datapoints that is needed to train the ANN model.

3.5. Influence of data cleansing on model accuracy

We now further discuss how data cleansing affects the final accuracy of the ANN model trained based on the concrete dataset (i.e., after hyperparameter optimization). Fig. 6 shows the strength values that are predicted by the ANN model for the test set (i.e., for unknown samples that are invisible to the model during its training) as a function of the actual strength values—wherein the $y = x$ line indicates ideal agreement between predicted and true strength values. In these figures, the color of each pixel indicates the number of overlapped datapoints locally. Overall, we find that the ANN model trained with the cleaned dataset exhibits a higher accuracy—the test set RMSE

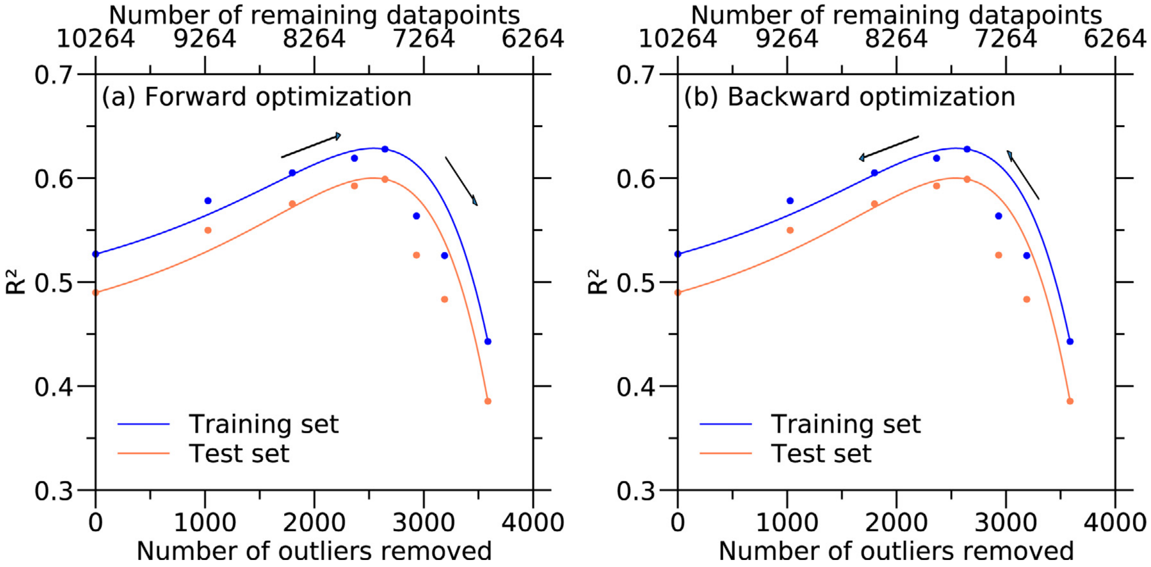

Fig. 3. Coefficient of determination (R^2) achieved by the artificial neural network considered herein for the training and test sets at each iteration of the (a) forward and (b) backward searching process as a function of the number of removed outliers (bottom axis) and remaining datapoints in the dataset (top axis), wherein the solid line is solely meant to guide the eye. Results are presented for the concrete dataset (see Section 2.1).

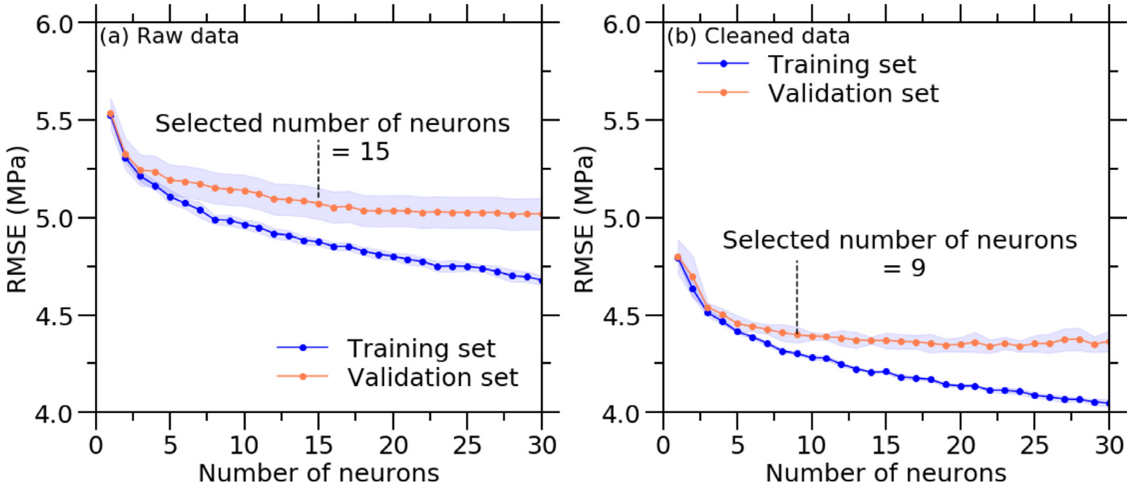

Fig. 4. Root-mean-square error (RMSE) achieved by the artificial neural network considered herein for the training and test sets as a function of the number of neurons in the single hidden layer when trained based on the (a) raw and (b) cleaned dataset. Results are presented for the concrete dataset (see Section 2.1).

decreasing from 5.07 to 4.40 MPa. A visual inspection of Fig. 6 also reveals that data cleansing results in a distribution of the datapoints that is more sharply centered around $y = x$. To further assess the overall performance of the ANN model on the concrete dataset, we compute the error distributions (i.e., distribution of the deviation between the prediction and true strength values), which are displayed in Fig. 7. We find that the ANN model trained on the uncleaned dataset is slightly biased as, overall, it tends to slightly underestimate concrete strength (which manifests itself by a negative mean error). In contrast, the mean error offered by the model trained based on the cleaned dataset is one order of magnitude smaller. Furthermore, we find that the error distribution becomes notably sharper after data cleansing. To quantify this change, we calculate the 90% and 95% confidence intervals based on a Gaussian fit. We find that, after data cleansing, the 90% and 95% confidence intervals decrease from ±8.3 to ±7.2 MPa and from ±9.9 to ± 8.6 MPa, respectively. This one more time illustrates that the outliers that are identified by our unsupervised EBOD approach are indeed far away from the interpolation model, which confirms their outlier nature.

We further explore whether the presence of outliers "deforms" the model trained based on the concrete dataset. This analysis aims to understand if the outliers that are present in the dataset simply increase the overall error of the model by lying far from the interpolated function or if the error of the model actually arises from the fact that the model itself is affected by the presence of outliers. To this end, since a direct data visualization is not possible in the entire feature space, we focus on the role of two select important features: (i) the water-to-cementitious (w/cm) ratio and (ii) the weight fraction of fly ash. These features are convenient since common concrete engineering knowledge suggests that concrete strength should monotonically decrease upon increasing w/cm and fly ash fraction [66]. For illustration purposes, Fig. 8 shows the evolution of the strength that is predicted by the ANN models (with and without data cleansing) as a function of these two features. Note that, in this case, the other features are fixed to their average values. In both cases, the predicted values are compared with actual datapoints. Note that these datapoints are not exactly comparable to the predicted strength values as their features are not perfectly equal to the

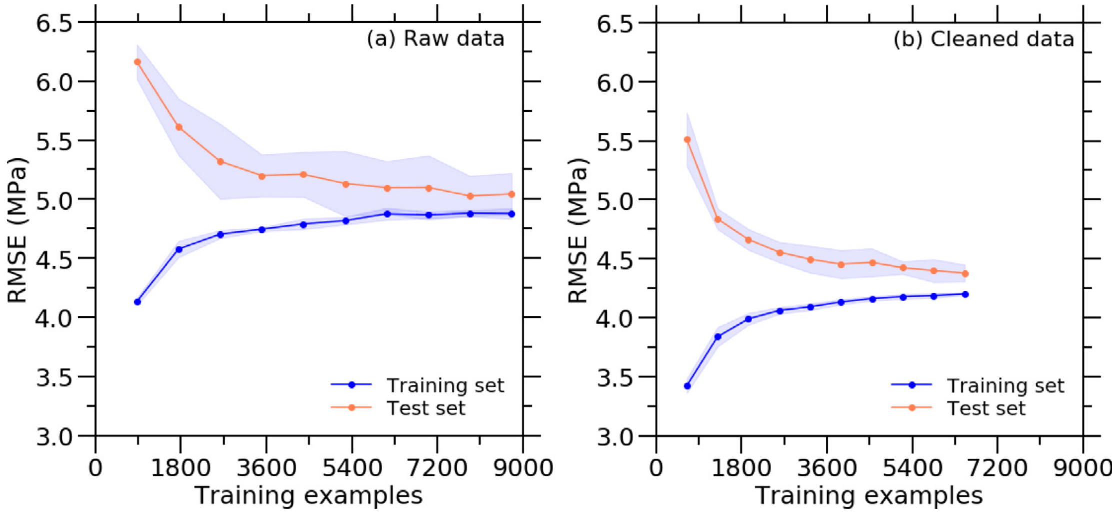

Fig. 5. Learning curves showing the root-mean-square error (RMSE) achieved by the artificial neural network considered herein for the training and test sets as a function of the number of training examples when trained based on the (a) raw and (b) cleaned dataset. Results are presented for the concrete dataset (see Section 2.1).

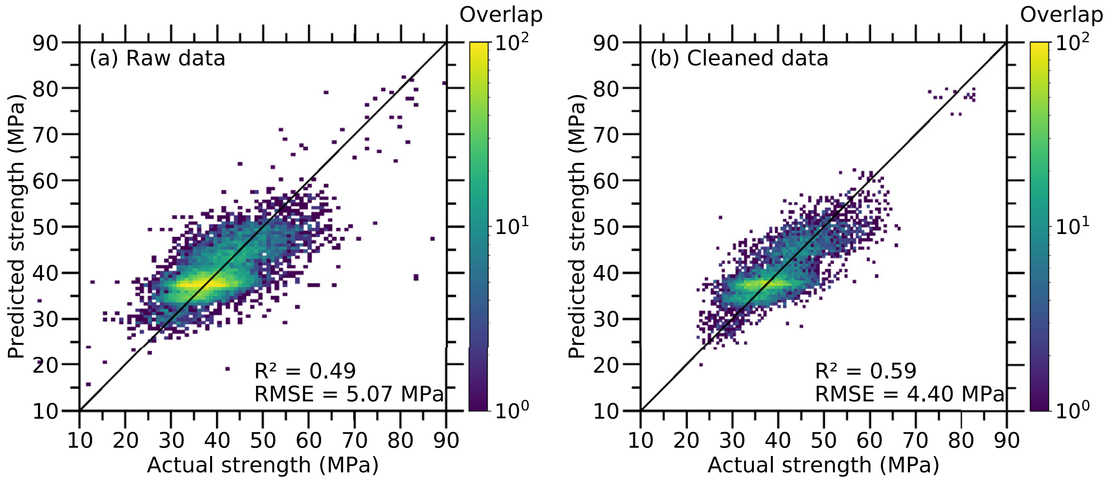

Fig. 6. Test set concrete strength values predicted by the artificial neural network model as a function of the measured strength, when the model is trained based on the (a) raw and (b) cleaned dataset (note: the color here indicates the number of overlapped data points at each pixel). Results are presented for the concrete dataset (see Section 2.1).

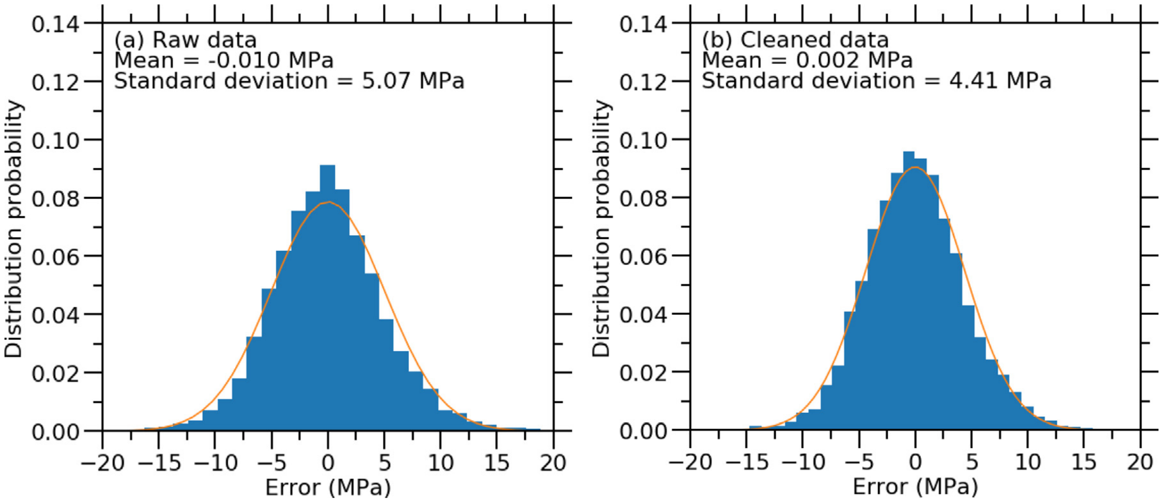

Fig. 7. Test set error distribution shown by the artificial neural network models trained based on the (a) raw and (b) cleaned dataset (note: the distributions are fitted by some Gaussian distributions). Results are presented for the concrete dataset (see Section 2.1).

Table 5
Summary of the test set performance of the artificial neural network models considered in this study, both before and after using the proposed EBOD outlier removal method. Results are presented for the concrete dataset (see Section 2.1).

Dataset	Complexity	Accuracy				
	Number of neurons	R^2	RMSE (MPa)	Bias (MPa)	Confidence interval (MPa)	
					90%	95%
Original	15	0.49	5.07	0.010	±8.3	±9.9
Cleaned	9	0.59	4.40	0.002	±7.2	±8.6

average values used to interrogate the ANN models. Nevertheless, these points are selected based on their vicinity within the feature space (with a relative variation of the features that is less than 10%). Among these datapoints, those that are flagged as outliers (and, hence, eventually removed) are highlighted in orange.

In both cases (see Fig. 8a and Fig. 8b), we observe that datapoints that are identified as outliers are indeed far away from the rest of the datapoints (in terms of the output value, but not in terms of the input features). Importantly, we find that the presence of outliers significantly deforms the ANN model. First, in both cases, we find that the outliers tend to shift the model toward lower strength values. This echoes the fact the, without any data cleansing, the model exhibits a negative mean error (see Fig. 7a). Further, we find that the presence of outliers tends to make the model less monotonic and more prone to fluctuations (see Fig. 8b). This indicates that the model is locally deformed so as to attempt to fit the outliers. This illustration exemplifies why the model that is trained based on the non-cleaned dataset is associated with a higher optimal degree of complexity—since this increased complexity is required to fit the variability of the training set. In both cases, the non-monotonic behavior exhibited by the model trained based on the raw data is not supported by common concrete engineering knowledge and, hence, is solely a spurious effect arising from the outliers.

The key effects of data cleansing on the complexity and accuracy of the ANN model are summarized in Table 5. Overall, our proposed EBOD method not only lowers the complexity of the ANN model by reducing the neurons needed for training, but also improves its accuracy in terms of increasing R^2 while reducing RMSE, bias, and the confidence interval.

3.6. Comparison with alternative ensemble-based outlier detection methods

To further illustrate the performance of the present EBOD data cleansing method, we compare it with several prevailing unsupervised ensemble-based outlier detection methods (see Section 2.5 for details) by first taking the example of the concrete dataset. To this end, we first clean the dataset by removing the outliers identified by each method. Note that, to enable a meaningful comparison, the threshold parameters of each detector are adjusted so as to yield the same number of outliers as those identified by the present EBOD method (i.e., 25% of the datapoints in the raw dataset are detected as outliers and removed, see Section 2.5). As such, at constant number of removed datapoints, this comparative analysis aims to assess the ability of each method to identify and remove "true" outliers. Following data cleansing, the ANN model presented in Section 2.2 is then independently retrained for each ensemble-based outlier detection method.

To assess the performance of these ensemble-based outlier detection methods, we calculate the coefficient of determination R^2 accuracy and the mean absolute error achieved by each ANN model on the test set, after applying each data cleansing method. Associated results are displayed in Table 6. We first note that, in contrast to the individual detector algorithms (see Table 1), no ensemble-based outlier detection method has a notably detrimental effect on the performance of the ANN model—as the test set accuracy remains fairly similar or larger than that achieved after training the model based on the raw, uncleaned dataset. This highlights the fact that, in general, ensemble-based outlier detection methods are effective in preventing inappropriate individual detector (e.g., OSCVM) from removing too many non-outlier datapoints that are actually useful to train the ANN model.

Overall, we find that the Averaging, Maximization, AOM, and MOA methods yield fairly similar performance (i.e., test set $R^2 =$ 0.48-to-0.49). This suggests that these methods, although they are not harmful to the model, do not succeed at identifying and removing the very outliers that limit the accuracy of the model. The lower performance of the Averaging and Maximization approaches likely arises from the fact that these approaches do not exclude the prediction from poorly performing individual base detectors. In fact, all these methods eventually yield a model accuracy that is inferior to that offered by the ABOD detector alone (see Table 4). This illustrates the fact that, although all these ensemble-based methods use the ABOD detector in their pool, their outcomes are contaminated by the inaccurate predictions of the other detectors in the pool.

In contrast, the Feature Bagging, LSCP and AKPV methods offer a notably improved accuracy as compared to the other alternative ensemble-based outlier detection approaches. This is likely due to the fact that, in many cases, outliers can only be identified in certain subsets of the feature space while it might be difficult to find outliers within the entire feature space (by simultaneously considering all the dimensions)—especially when some features are less influential than others. Feature bagging addresses this difficulty by combing the outlier scores offered by different detectors applied to different subsets of the features [55], while LSCP considers the location of the datapoints when combining the predictions from the different detectors [57]. Nevertheless, the accuracy offered by the Feature Bagging and LSCP methods ($R^2 = 0.54$) remains similar to that offered by the ABOD detector alone. Even though AKPV achieves higher accuracy than Feature Bagging and LSCP, its performance is limited by the fact that this algorithm only considers the top three detectors (while EBOD is more flexible and can consider a larger set of detectors—ABOD, COF, SOS, and LOF in this case). This indicates that, overall, these methods do not succeed at meaningfully combining the individual detectors so as to leverage and combine their respective strengths.

Compared to Feature Bagging, LODA and SUOD also rely on a random sampling of the feature space by a series of base detectors. However, both of them exhibit inferior performance in the case of the present dataset. This may arise from the fact that LODA relies on simple one-dimensional histograms as base detectors, which might be too simplistic for the present dataset. In turn, by replacing costly unsupervised algorithms with faster supervised regression algorithms [58], this surrogate model generated by SUOD can affect the accuracy of the detection. Although this approach has some merits for large datasets, computational burden is not a bottleneck in the case of the present small dataset.

Overall, as a key conclusion, we find that our new EBOD method systematically outperforms all these alternative ensemble-based methods by a large margin. This denotes that, in the case of the present dataset, our EBOD approach is the most effective at identifying the optimal union of detector algorithms, wherein the selected detectors tend to most positively complement each other. This is likely a positive consequence of the forward–backward search for the top outliers as implemented in the present EBOD method, which is efficient at pinpointing

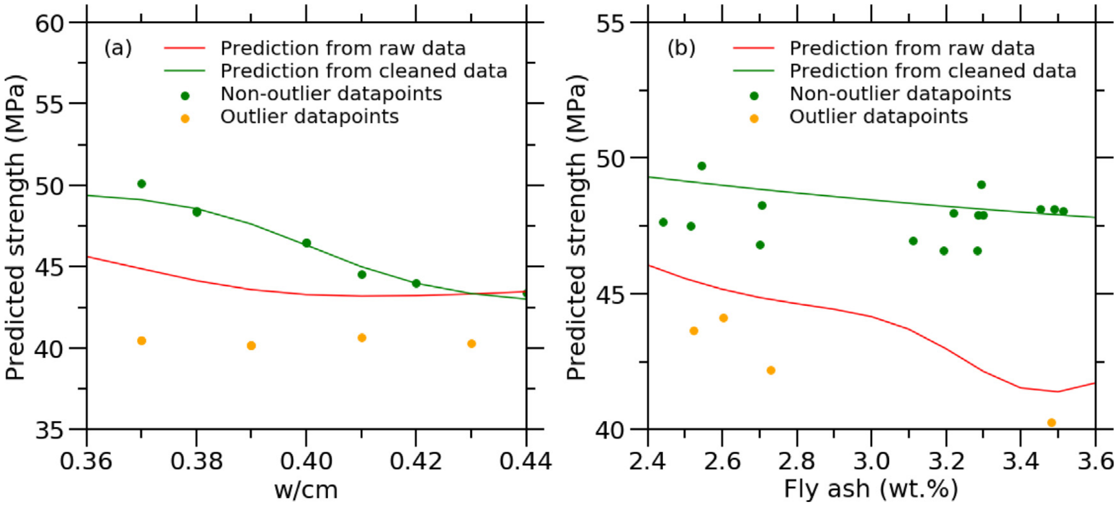

Fig. 8. Concrete strength predicted by the artificial neural network models trained based on the raw and cleaned dataset as a function of two select features, namely, (a) the water-to-cementitious ratio (w/cm) and (b) the weight fraction of fly ash. Other features are fixed to their average values. The predicted values are compared with actual datapoints that are located at the vicinity of the predicted datapoints in the feature space (see text). Results are presented for the concrete dataset (see Section 2.1).

Table 6
Performance on the test set of the artificial neural network model trained using the concrete dataset cleaned by select ensemble-based outlier detection method.

Ensemble-based detector	R^2	Mean absolute error of the model (MPa)	Mean absolute distance of outliers from model (MPa)
Raw uncleaned dataset	0.49	3.91	–
Averaging	0.49	4.39	2.66
Maximization	0.48	4.32	2.85
AOM	0.49	4.30	2.95
MOA	0.48	4.12	3.29
SUOD	0.43	3.25	6.53
LODA	0.51	3.95	4.63
Feature bagging	0.54	3.83	5.00
LSCP	0.54	3.47	5.49
AKPV	0.55	3.41	5.96
Present EBOD method	0.59	3.45	5.86

the most optimal combination of detectors and disregarding poorly-performing detectors.

As an alternative assessment of the ability of each ensemble-based outlier detection method to identify and remove true outliers (rather than non-outlier datapoints that are useful to train the ANN model), we compute the distance between the labels and the model predictions of the datapoints identified by each method. This analysis is based on the idea that, after training, the model filters out the noise of the dataset and offers an estimation of what the ground truth should be in the absence of any noise. As such, provided that the model is accurate, the distance of a given datapoint from the model prediction offers a *posteriori* validation of whether this datapoint was indeed an outlier or not—wherein true outliers are associated with large absolute distances from the model prediction. It is worth noting that this analysis offers a meaningful, independent check on each data cleansing method, since all the detectors considered herein are unsupervised and are never exposed to the predictions of the ANN model.

Fig. 9 shows the distributions of the distance from the model prediction of the outliers detected by each of the data cleansing approaches considered herein. In line with the coefficient of determination values presented in Table 3, we find that most of the outliers identified by the Averaging, Maximization, AOM, and MOA methods are located in the vicinity of the model surface (i.e., < 8 MPa absolute distance). In fact, the average absolute distance of the outliers identified by these methods is lower than the mean absolute error of the model (see Table 3). This indicates that these approaches tend to classify useful datapoints as outliers and, in turn, fail to detect the true outlier datapoints that are far from the model prediction. In contrast, SUOD, LODA, Feature Bagging, LSCP, AKPV as well as the present EBOD approach tend to classify as outliers datapoints that, on average, are further away from the model prediction than the mean absolute error of the model (see Table 3), which shows that all these methods successfully identify the outliers that are far from the model prediction. Although the outliers detected by SUOD show the largest distance from the model prediction, the ANN model trained based on remaining datapoints eventually exhibit the lowest coefficient of determination R^2. This implies that some of the outliers that are identified by SUOD are actually rather informative for the model training, so that removing them greatly reduces the accuracy of the trained model. In contrast, our new EBOD approach outperforms other methods in their ability to detect and remove the most extreme outliers that are the furthest away from the model prediction (e.g., > 15 MPa away from the model, see Fig. 9).

Finally, Fig. 10 shows the strength (i.e., output) distribution of the outliers identified by each ensemble-based data cleansing approach. We find that the Averaging, Maximization, AOM, and MOA methods tend to systematically classify as outliers some datapoints that are associated with intermediate strength values (i.e., between 30 and 45 MPa), which is also the most densely populated region of the original dataset. In turn, these approaches do not label as outliers most of the extreme datapoints (i.e., associated with very low or very high strength values). This behavior contrasts with the fact that concrete strength measurement outliers are essentially random and are not expected to be solely encountered for intermediate strength values [36,67,68]. Rather, the fact that these detector methods tend to classify as outliers some datapoints that are associated with intermediate strength values suggests that these methods perform better in the densely populated regions of the dataset (i.e., wherein each datapoint shows a large density of neighbors) but do not perform well in the low-density, sparser regions of the dataset. In contrast, SUOD, LODA, Feature Bagging, LSCP, AKPV as well as the present EBOD approach classify as outliers a notably increased fraction of extreme datapoints, that is, which are associated with strength values (that are lower than 30 MPa or larger than 45 MPa). Furthermore, the present EBOD method also yields a distribution

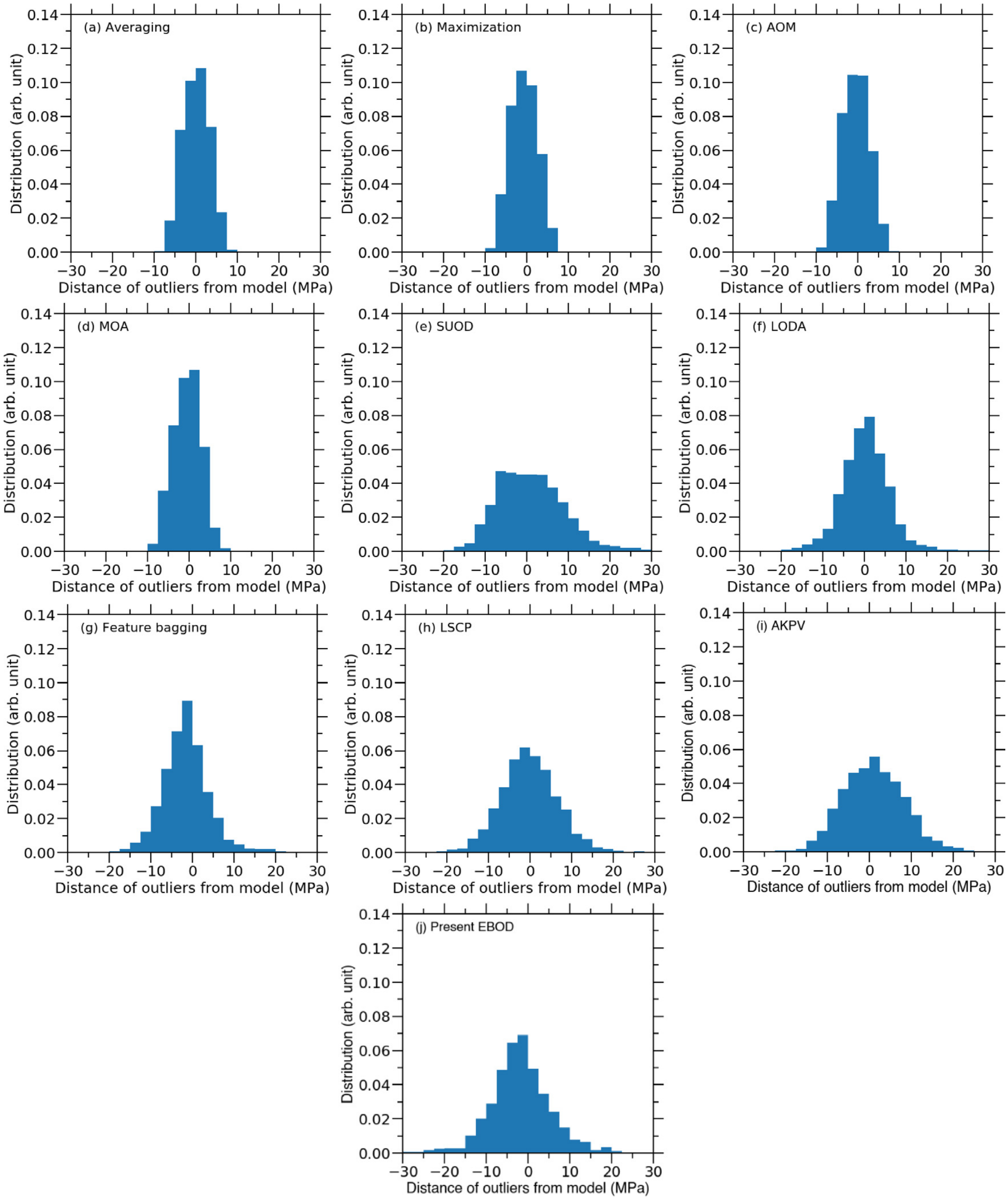

Fig. 9. Distributions of the distance from the artificial neural network (ANN) model prediction (i.e., in terms of output strength difference) of the outliers detected by each of the ensemble-based outlier detection methods considered herein. Note that, in each case, a distinct ANN model is trained after the outliers identified by each approach are removed. Results are presented for the concrete dataset (see Section 2.1).

of outlier strength values that exhibits the largest degree of similarity (in shape) as compared to the distribution of the initial strength values in the entire dataset. This echoes the fact that, in the case considered herein (i.e., concrete strength measurement), the probability of a datapoint to be an outlier is expected to be fairly independent of strength. Overall, all these observations confirm that the present EBOD method features an enhanced ability to meaningfully detect and remove true outlier datapoints as compared to other alternative ensemble-based methods.

3.7. Comparison with alternative ensemble-based detectors based on benchmark datasets

To ensure that the obtained results are generic and representative of various types of datasets, we extend the analysis to a series of ten additional benchmark regression datasets (see Table 1). To evaluate the robustness of the proposed EBOD method across datasets, we then carry out a series of statistical tests to compare its cleansing performance to that of nine established ensemble-based outlier detection methods (see Table 3). We first consider the test set accuracy of the ANN model considered herein (see Section 2.2) when trained based on the cleaned dataset as a measure of the performance of the detector. To this end, we

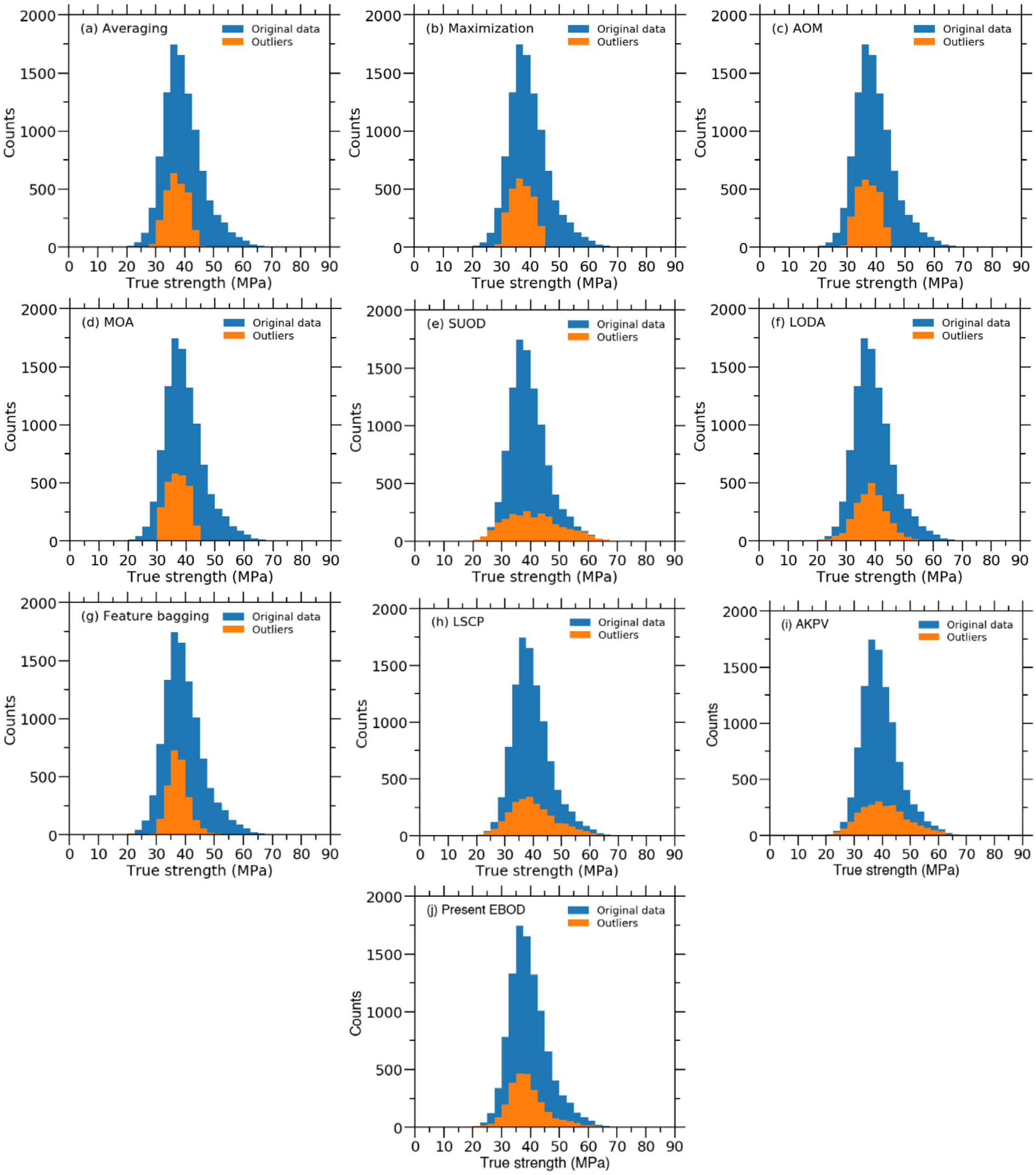

Fig. 10. Distribution of the true strength of the outliers detected by each of the ensemble-based outlier detection methods considered herein. The distributions are compared with the distribution of the strength values in the raw, uncleaned dataset. Results are presented for the concrete dataset (see Section 2.1).

tune the threshold of each of the ensembled methods to flag the same number of outliers as EBOD, for each dataset. After data cleaning, we train the ANN model (using the same fixed hyperparameters) to assess the performance of all the ensemble-based detectors. The R^2 results on the test set (20% of the cleaned dataset) achieved by each detector for each dataset are compared in Table 7. Note that these R^2 results are evaluated based on ten random repetitions of the train-test split. Based on this evaluation, we find that EBOD broadly outperforms its counterparts, since EBOD achieves the highest test R^2 for seven out of the ten datasets and consistently exhibits a performance that is comparable to that of the top detection algorithm in all the other three cases.

To further establish the statistical significance of the performance increase offered by EBOD, we then carry out some Friedman and post hoc Dunn's tests by following the procedures presented in Section 2.6. Based on the Friedman test, the p-value for EBOD is determined to be 1.11×10^{-16}. This extremely low value suggests that there exists a statistically significant performance difference between EBOD and the other methods (see Table 8). In addition, a post hoc Dunn's test is conducted for each pair of detectors—to individually compare the performance of EBOD with that offered by each of the other alternative outlier ensemble methods (see Table 9). In this case, the p-values represent the significance of the pairwise comparison. For most pair comparisons, the null hypothesis (H_0) that both methods exhibit comparable performance (i.e., p-values > 0.05) is rejected. Although the above statistical tests suggest that both AKPV and Feature bagging are not significantly different from EBOD (with their p-values > 0.1), the test R^2 results shown in Table 7 nevertheless still affirm the superior performance of EBOD—both in terms of increased accuracy of compared to alternative cleansing approaches and generalizability to various datasets.

Table 7
Comparison of the R^2 prediction accuracy on the test set (20%) offered by the artificial neural network model trained using the benchmark datasets—wherein each dataset is cleaned by each of the ensemble-based detectors. The test R^2 values shown here are averaged based on ten repetitions of random train-test split. The highest score is highlighted in bold.

Dataset	EBOD	Alternative detectors								
		Averaging	Maximization	AOM	MOA	SUOD	LODA	Feature bagging	LSCP	AKPV
Delta Elevators	**0.711**	0.661	0.663	0.662	0.663	0.664	0.642	0.707	0.697	0.706
Red wine	**0.362**	0.247	0.221	0.221	0.221	0.281	0.216	0316	0.325	0.320
Airfoil noise	**0.931**	0.884	0.905	0.899	0.901	0.896	0.854	0.909	0.908	0.928
Qsar fish toxicity	0.725	0.643	0.622	0.622	0.641	0.657	0.546	0.706	0.704	**0.740**
Boston Housing	**0.909**	0.762	0.744	0.769	0.769	0801	0.831	0.834	0.805	0.864
California Housing	0.842	0.739	0.731	0.735	0.731	0.803	0.703	**0.847**	0.821	0.843
Ailerons	**0.841**	0.775	0.782	0..775	0.834	0.798	0.824	0.834	0.829	0.836
Abalone	0.610	0.552	0.487	0.615	0.474	0.538	0.476	**0.611**	0.555	0.572
UCI concrete	**0.918**	0.886	0.899	0.894	0.917	0.895	0.891	0.905	0.899	0.901
Real estate	**0.752**	0.603	0.624	0.528	0.624	0.471	0.492	0.678	0.554	0.726

Table 8
Ranking of the ensemble-based outlier detection algorithms considered herein using the Friedman test.

Rank score	Algorithm
6.94	EBOD
6.27	AKPV
6.05	Feature bagging
4.72	LSCP
3.58	MOA
3.02	SUOD
2.91	Maximization
2.84	AOM
2.54	Averaging
1.9	LODA

Table 9
Post hoc Dunn's test, wherein the performance of the EBOD approach is compared with that of each alternative individual ensemble-based outlier detection algorithm.

Pairwise comparison	p-value	Result
EBOD vs. AKPV	0.50	$p > 0.1$; H_0 is accepted
EBOD vs. Feature bagging	0.37	$p > 0.1$; H_0 is accepted
EBOD vs. LSCP	0.0260	$p < 0.05$; H_0 is rejected
EBOD vs. MOA	0.00070	$p < 0.05$; H_0 is rejected
EBOD vs. SUOD	9.06×10^{-5}	$p < 0.05$; H_0 is rejected
EBOD vs. Maximization	5.69×10^{-5}	$p < 0.05$; H_0 is rejected
EBOD vs. AOM	4.15×10^{-5}	$p < 0.05$; H_0 is rejected
EBOD vs. Averaging	1.11×10^{-5}	$p < 0.05$; H_0 is rejected
EBOD vs. LODA	1.57×10^{-7}	$p < 0.05$; H_0 is rejected

3.8. Limitations

Despite its performance on the broad selection of datasets considered herein, the proposed EBOD method comes with a certain number of limitations. First, as an unsupervised outlier detection approach, the EBOD approach cannot benefit from the knowledge of labeled outliers. Although the unsupervised characteristic of the EBOD approach is primarily a strength (since it does not require any manual labeling of outliers), EBOD would likely be outperformed by supervised approaches if a collection of representative ground-truth outliers is *a priori* known. Second, the EBOD does not rely on any assumptions regarding the nature of the outliers. Although this makes the EBOD approach robust and general (especially for noisy datasets), the EBOD approach is likely to be outperformed by more specialized outlier detection approaches for specific datasets wherein the outliers exhibit a clear, distinctive fingerprint. Finally, the EBOD approach comes with a notable computational burden. To illustrate this, Table 10 shows the running time of each detector for cleaning the ten benchmark datasets considered herein. The results show that, although it offers an enhanced accuracy, EBOD consistently requires more computing time to complete the cleaning. This is mainly due to the fact that EBOD relies on a costly forward–backward search for the optimal set of detectors. In practice, this high computing cost can prevent the use of EBOD on very large datasets—wherein model training takes a significant time. Nevertheless, the computational cost of EBOD still remains reasonable for all the datasets considered herein (which present a maximum of 20,640 data points and 41 input features). More generally, this computational time is expected to remain small as compared to the time that needs to be invested to conduct a typical data cleansing based on trial-and-error, Edisonian approaches. This computational time is also considered reasonable since a robust data cleansing can result in a larger increase in model accuracy than a costly systematic optimization of model learning algorithm and hyperparameters.

4. Conclusions

Overall, we find that our proposed EBOD outlier detection method improves the learning efficiency of the ANN model considered herein by decreasing the number of required hidden neurons, as well as the number of datapoints that are needed for the model to learn how to map inputs to output. Importantly, we find that our EBOD outlier detection approach considerably improves the accuracy of the trained ANN model, which is systematically illustrated by the test set R^2, RMSE, bias, and confidence interval. Importantly, our new EBOD method systematically outperforms alternative outlier detector algorithm, when used either individually or in ensemble. The performance of the EBOD method consistently apply to a broad range of regression datasets. Altogether, these results suggest that considering an optimized ensemble of outlier detection algorithms (rather than a single detector or simply an average of several detectors) offers a more robust data cleansing and, consequently, notably increases the performance of the subsequent machine learning model. It is also worth mentioning that the EBOD approach does not require any intuition or knowledge regarding which type of detector (e.g., distance-based, angle-based, etc.) is best suited to tackle a given dataset. This approach also has the advantage of being fully unsupervised, that is, it does not require any expert-based examples of outliers or of any preexisting knowledge of what the typical signature of an outlier should be.

Table 10
Comparison of the running time (in seconds) of the EBOD method, as well as that of alternative ensemble-based outlier detection methods.

Dataset	EBOD	Averaging	Maximization	AOM	MOA	SUOD	LODA	Feature bagging	LSCP	AKPV
Delta elevators	359.9	57.7	59.1	55.3	52.1	22.1	8.9	16.4	92.3	89.5
Red wine	385.8	5.2	4.8	4.1	4.2	6.7	3.4	9.2	6.08	46.6
Airfoil noise	266.4	6.3	6.6	5.1	7.6	5.7	3.9	12.7	7.1	31.3
Qsar fish toxicity	197.1	3.0	3.1	2.7	2.9	4.9	1.9	7.6	4.8	22.2
Boston Housing	121.5	2.3	2.5	2.2	2.4	4.4	1.7	4.1	3.0	12.9
California Housing	2132.6	613.6	474.9	656.1	730.3	90.8	34.8	67.7	699.2	496.0
Ailerons	2126.6	75.4	60.9	302.7	54.7	35.7	24.0	80.1	67.3	289.1
Abalone	826.6	17.7	17.3	18.2	16.4	12.7	6.3	20.6	132.1	95.4
UCI concrete	211.6	5.1	4.2	4.1	4.8	5.9	2.0	6.4	3.5	25.4
Real estate	94.9	1.7	2.2	2.1	2.4	5.7	1.9	1.1	2.3	14.8

CRediT authorship contribution statement

Boya Ouyang: Methodology, Software, Validation, Formal analysis, Writing – Draft & Review & Editing, Writing – Original Draft. **Yu Song:** Methodology, Software, Writing – Draft & Review & Editing, Investigation. **Yuhai Li:** Methodology, Software, Validation. **Gaurav Sant:** Resources, Supervision, Funding acquisition. **Mathieu Bauchy:** Conceptualization, Supervision, Writing – Review & Editing, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors acknowledge some financial support for this research provided by the U.S. Department of Transportation through the Federal Highway Administration (Grant #: 693JJ31950021) and the U.S. National Science Foundation (DMREF: 1922167).

References

[1] K. Hazelwood, S. Bird, D. Brooks, S. Chintala, U. Diril, D. Dzhulgakov, M. Fawzy, B. Jia, Y. Jia, A. Kalro, J. Law, K. Lee, J. Lu, P. Noordhuis, M. Smelyanskiy, L. Xiong, X. Wang, Applied machine learning at facebook: A datacenter infrastructure perspective, in: 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA). Presented At the 2018 IEEE International Symposium on High Performance Computer Architecture, HPCA, 2018, pp. 620–629, http://dx.doi.org/10.1109/HPCA.2018.00059.

[2] S. Idowu, S. Saguna, C. Åhlund, O. Schelén, Applied machine learning: Forecasting heat load in district heating system, Energy Build. 133 (2016) 478–488.

[3] S.O. Uwagbole, W.J. Buchanan, L. Fan, Applied machine learning predictive analytics to SQL injection attack detection and prevention, in: 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). Presented At the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), IEEE, Lisbon, Portugal, 2017, pp. 1087–1090, http://dx.doi.org/10.23919/INM.2017.7987433.

[4] Y. Zhang, C. Ling, A strategy to apply machine learning to small datasets in materials science, Npj Comput. Mater. 4 (2018) 1–8.

[5] A. Smola, S.V.N. Vishwanathan, Introduction to machine learning, Camb. Univ. UK 32 (2008) 2008.

[6] H. Liu, Z. Fu, K. Yang, X. Xu, M. Bauchy, Machine learning for glass science and engineering: A review, J. Non-Cryst. Solids X 4 (2019) 100036, http://dx.doi.org/10.1016/j.nocx.2019.100036.

[7] C. Cortes, L.D. Jackel, W.-P. Chiang, Limits on learning machine accuracy imposed by data quality, in: Advances in Neural Information Processing Systems, 1995, pp. 239–246.

[8] H.J. Escalante, A comparison of outlier detection algorithms for machine learning, in: Proceedings of the International Conference on Communications in Computing, 2005, pp. 228–237.

[9] D. Hendrycks, M. Mazeika, T. Dietterich, Deep anomaly detection with outlier exposure, 2018, arXiv prepr. arXiv:181204606.

[10] J. Mourão-Miranda, D.R. Hardoon, T. Hahn, A.F. Marquand, S.C.R. Williams, J. Shawe-Taylor, M. Brammer, Patient classification as an outlier detection problem: An application of the One-Class Support Vector Machine, NeuroImage 58 (2011) 793–804, http://dx.doi.org/10.1016/j.neuroimage.2011.06.042.

[11] A. Vinueza, G. Grudic, Unsupervised outlier detection and semi-supervised learning, Tech. Rep. CU-CS-976-04, 2004.

[12] K. Zhang, M. Luo, Outlier-robust extreme learning machine for regression problems, Neurocomputing 151 (2015) 1519–1527, http://dx.doi.org/10.1016/j.neucom.2014.09.022.

[13] M. Last, A. Kandel, Automated detection of outliers in real-world data, in: Proceedings of the Second International Conference on Intelligent Technologies, 2001, pp. 292–301.

[14] S.-M. Udrescu, M. Tegmark, AI Feynman: A physics-inspired method for symbolic regression, Sci. Adv. 6 (2020) eaay2631.

[15] V. Barnett, The study of outliers: purpose and model, J. R. Stat. Soc. Ser. C. Appl. Stat. 27 (1978) 242–250.

[16] R.C. Geary, The ratio of the mean deviation to the standard deviation as a test of normality, Biometrika 27 (1935) 310–332.

[17] C.H. Sim, F.F. Gan, T.C. Chang, Outlier labeling with boxplot procedures, J. Am. Stat. Assoc. 100 (2005) 642–652.

[18] S. Guha, R. Rastogi, K. Shim, CURE: an efficient clustering algorithm for large databases, ACM SIGMOD Rec. 27 (1998) 73–84.

[19] R.T. Ng, J. Han, Efficient and Effective clustering methods for spatial data mining, in: Proceedings of VLDB, 1994, pp. 144–155.

[20] S. Sheikholeslami, S. Chatterjee, A. Zhang, A multi-resolution clustering approach for very large spatial databases, in: Proceedings of the 24th International Confere Shaw MJ Shaw; Chandrasekar Subramaniam a, Gek Woo Tan a, Michae International Conference on Formal Ontology in Information Systems, 2002, pp. 622–630.

[21] N. Roussopoulos, S. Kelley, F. Vincent, Nearest neighbor queries, in: Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, 1995, pp. 71–79.

[22] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, pp. 93–104.

[23] H.-P. Kriegel, M. Schubert, A. Zimek, Angle-based outlier detection in high-dimensional data, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 444–452.

[24] H.J. Motulsky, R.E. Brown, Detecting outliers when fitting data with nonlinear regression–a new method based on robust nonlinear regression and the false discovery rate, BMC Bioinformatics 7 (2006) 123.

[25] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, When is nearest neighbor meaningful? in: International Conference on Database Theory, Springer, 1999, pp. 217–235.

[26] C.C. Aggarwal, Outlier ensembles: position paper, ACM SIGKDD Explor. Newsl. 14 (2013) 49–58.

[27] C.C. Aggarwal, S. Sathe, Outlier Ensembles: An Introduction, Springer, 2017.

[28] S. Rayana, W. Zhong, L. Akoglu, Sequential ensemble learning for outlier detection: A bias–variance perspective, in: 2016 IEEE 16th International Conference on Data Mining, ICDM, IEEE, 2016, pp. 1167–1172.

[29] S.-A.N. Alexandropoulos, S.B. Kotsiantis, V.E. Piperigou, M.N. Vrahatis, A new ensemble method for outlier identification, in: 2020 10th International Conference on Cloud Computing, Data Science & Engineering, Confluence, IEEE, 2020, pp. 769–774.

[30] B.A. Young, A. Hall, L. Pilon, P. Gupta, G. Sant, Can the compressive strength of concrete be estimated from knowledge of the mixture proportions?: New insights from statistical analysis and machine learning methods, Cem. Concr. Res. 115 (2019) 379–388, http://dx.doi.org/10.1016/j.cemconres.2018.09.006.

[31] B. Ouyang, Y. Li, F. Wu, H. Yu, Y. Wang, G. Sant, M. Bauchy, Computational modeling – predicting concrete's strength by machine learning: Balance between accuracy and complexity of algorithms, ACI Mater. J. (2020).

[32] J.-I. Kim, D.K. Kim, M.Q. Feng, F. Yazdani, Application of neural networks for estimation of concrete strength, J. Mater. Civ. Eng. 16 (2004) 257–264.

[33] G.E. Troxell, H.E. Davis, J.W. Kelly, Composition and properties of concrete, 1968.

[34] P.L.J. Domone, M.N. Soutsos, Approach to the proportioning of high-strength concrete mixes, Concr. Int. 16 (1994) 26–31.

[35] A.S. for T. Materials, Standard test method for compressive strength of cylindrical concrete specimens, in: ASTM C39, 2003.

[36] D. Zhenchao, Discussion on problem of standard deviation of concrete strength, ACI Mater. J. 117 (2020).

[37] D. Dua, C. Graff, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2019, http://archive.ics.uci.edu/ml, zuletzt abgerufen am: 14.09. 2019. Google Sch.

[38] I.-C. Yeh, T.-K. Hsu, Building real estate valuation models with comparative approach through case-based reasoning, Appl. Soft Comput. 65 (2018) 260–271.

[39] P. Reiner, B.M. Wilamowski, Efficient incremental construction of RBF networks using quasi-gradient method, Neurocomputing 150 (2015) 349–356.

[40] M. Cassotti, D. Ballabio, R. Todeschini, V. Consonni, A similarity-based QSAR model for predicting acute toxicity towards the fathead minnow (Pimephales promelas), SAR QSAR Environ. Res. 26 (2015) 217–243.

[41] X. Qiu, L. Zhang, Y. Ren, P.N. Suganthan, G. Amaratunga, Ensemble deep learning for regression and time series forecasting, in: 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning, CIEL, IEEE, 2014, pp. 1–6.

[42] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis, Modeling wine preferences by data mining from physicochemical properties, Decis. Support Syst. 47 (2009) 547–553.

[43] I.-C. Yeh, Modeling of strength of high-performance concrete using artificial neural networks, Cem. Concr. Res. 28 (1998) 1797–1808.

[44] W.J. Nash, T.L. Sellers, S.R. Talbot, A.J. Cawthorn, W.B. Ford, The population biology of abalone (haliotis species) in Tasmania. I, in: Blacklip Abalone (H. Rubra) from the North Coast and Islands of Bass Strait, Tech. Rep. 48, Sea Fish. Div., 1994, p. 411.

[45] M. Riedmiller, H. Braun, Rprop-a fast adaptive learning algorithm, in: Proc. of ISCIS VII, Universitat. Citeseer, 1992.

[46] T.F. Brooks, D.S. Pope, M.A. Marcolini, Airfoil Self-Noise and Prediction, National Aeronautics and Space Administration, Office of Management, 1989.

[47] A. Khamis, Z. Ismail, K. Haron, A. Tarmizi Mohammed, The effects of outliers data on neural network performance, J. Asian Pac. Soc. Cardiol. 5 (2005) 1394–1398.

[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[49] M. Stone, Cross-validatory choice and assessment of statistical predictions, J. R. Stat. Soc. Ser. B Methodol. 36 (1974) 111–133, http://dx.doi.org/10.1111/j.2517-6161.1974.tb00994.x.

[50] J. Tang, Z. Chen, A.W. Fu, D. Cheung, A robust outlier detection scheme for large data sets, in: 6th Pacific-Asia Conf. on Knowledge Discovery and Data Mining, 2001, pp. 6–8.

[51] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (2001) 1443–1471.

[52] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, in: Eighth IEEE International Conference on Data Mining. Presented at the 2008 Eighth IEEE International Conference on Data Mining, ICDM, IEEE, Pisa, Italy, 2008, pp. 413–422, http://dx.doi.org/10.1109/ICDM.2008.17.

[53] J.H.M. Janssens, F. Huszár, E.O. Postma, H.J. van den Herik, Stochastic Outlier Selection, Tech Rep., 2012.

[54] C.C. Aggarwal, S. Sathe, Theoretical foundations and algorithms for outlier ensembles, ACM SIGKDD Explor. Newsl. 17 (2015) 24–47.

[55] A. Lazarevic, V. Kumar, Feature bagging for outlier detection, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, 2005, pp. 157–166.

[56] T. Pevný, Loda: Lightweight on-line detector of anomalies, Mach. Learn. 102 (2016) 275–304.

[57] Y. Zhao, Z. Nasrullah, M.K. Hryniewicki, Z. Li, LSCP: Locally selective combination in parallel outlier ensembles, in: Proceedings of the 2019 SIAM International Conference on Data Mining, SIAM, 2019, pp. 585–593.

[58] Y. Zhao, X. Ding, J. Yang, H. Bai, SUOD: Toward scalable unsupervised outlier detection, 2020, arXiv prepr. arXiv:200203222.

[59] B. Micenková, B. McWilliams, I. Assent, Learning representations for outlier detection on a budget, 2015, arXiv prepr. arXiv:150708104.

[60] Y. Zhao, M.K. Hryniewicki, XGBOD: improving supervised outlier detection with unsupervised representation learning, in: 2018 International Joint Conference on Neural Networks, IJCNN, IEEE, 2018, pp. 1–8.

[61] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Stat. Assoc. 32 (1937) 675–701.

[62] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[63] W.H. Kruskal, W.A. Wallis, Use of ranks in one-criterion variance analysis, J. Am. Stat. Assoc. 47 (1952) 583–621, http://dx.doi.org/10.1080/01621459.1952.10483441.

[64] N.A. Krishnan, S. Mangalathu, M.M. Smedskjaer, A. Tandia, H. Burton, M. Bauchy, Predicting the dissolution kinetics of silicate glasses using machine learning, J. Non-Cryst. Solids 487 (2018) 37–45.

[65] K. Yang, X. Xu, B. Yang, B. Cook, H. Ramos, N.M.A. Krishnan, M.M. Smedskjaer, C. Hoover, M. Bauchy, Predicting the young's modulus of silicate glasses using high-throughput molecular dynamics simulations and machine learning, Sci. Rep. 9 (2019) 1–11, http://dx.doi.org/10.1038/s41598-019-45344-3.

[66] S. Popovics, J. Ujhelyi, Contribution to the concrete strength versus water-cement ratio relationship, J. Mater. Civ. Eng. 20 (2008) 459–463.

[67] A.C.I. Committee, Standardization, I.O. for, Building Code Requirements for Structural Concrete (ACI 318-08) and Commentary, American Concrete Institute, 2008.

[68] R.E. Philleo, Increasing the Usefulness of ACI 214: Use of Standard Deviation and a Tech-nique for small sample sizes, Concr. Int. 3 (1981) 71–74.