2018

Non-linear Machine Learning with Active Sampling for MOX Drift Compensation

Tamara Matthews
Technological University Dublin, tamara.matthews@tudublin.ie

Muhammad Iqbal
National College of Ireland, muhammad.iqbal@ncirl.ie

Horacio Gonzalez-Velez
National College of Ireland, horaciao@ncirl.ie

Recommended Citation

Matthews, T., Iqbal, M. & Gonzalez-Velez, H. (2018) Non-linear machine learning with active sampling for MOX drift compensation, *IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT)* DOI:10.1109/BDCAT.2018.00016

2018 IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT)

Non-linear machine learning with active sampling for MOX drift compensation

Tamara Matthews
School of Computing
CeADAR, Dublin Institute of Technology
Dublin, Ireland
matthewstmr@gmail.com

Muhammad Iqbal
School of Computing
National College of Ireland
Dublin, Ireland
muhammad.iqbal@ncirl.ie

Horacio González–Vélez
Cloud Competency Centre
National College of Ireland
Dublin, Ireland
horacio@ncirl.ie

***Abstract*—Metal oxide (MOX) gas detectors based on SnO_2 provide low-cost solutions for real-time sensing of complex gas mixtures for indoor ambient monitoring. With high sensitivity under ideal conditions, MOX detectors may have poor long-term response accuracy due to environmental factors (humidity and temperature) along with sensor aging, leading to calibration drifts. Finding a simple and efficient solution to correct such calibration drifts has been the subject of numerous studies but remains an open problem.**

In this work, we present an efficient approach to MOX calibration using active and transfer sampling techniques coupled with non-linear machine learning algorithms, namely neural networks, extreme gradient boosting (XGBoost) and radial kernel support vector machines (SVM). Applied on the UCI's *HT detectors* dataset, the study evaluates methods for active sampling, makes an assessment of suitable neural networks architectures and compares the performance of neural networks, XGBoost and radial kernel SVM to classify gas mixtures (banana and wine odours, clean air) in the presence of humidity and temperature changes. The results show high classification accuracy levels (above 90%) and confirm that active sampling can provide a suitable solution.

***Index Terms*—Neural Networks, Extreme Gradient Boosting, XGBoost, Support Vector Machines, Non-Linear Learning Methods, Machine Learning**

I. Introduction

In-situ real-time gas monitoring has been increasing in popularity, as affordable metal oxide (MOX) gas sensors enable applications aimed at gas detection for personal/wearable use, automated mobile inspection of a site, home surveillance, and public and industrial sites monitoring. Built as light detector arrays or as Electronic Nose (EN) systems, such devices bypass the need for a physical interface using wi-fi connections through mobile phone or dedicated computer applications.

Nevertheless, MOX sensors deliver real-time, dynamic, multivariate signals which require complex analysis to derive accurate readings for real-word applications. MOX signals are typically convoluted and unclean through different factors such as sensitivity changes due to humidity and temperature, baseline signal instability and sensor aging, all leading to sensor calibration drifts.

While electronics and material sciences continue to improve the response quality by improving device-physics and technical output [1], analysing sensor responses to distinguish among various gas components is a challenging task where advanced data analytics techniques have to be deployed.

Initial approaches to improve calibration stability used extensive feature engineering and model learning of complex multivariate time-series, which lead to models unstable over time as calibration patterns no-longer match. This research evolved into advanced bio-inspired odour sensing algorithms but also towards techniques for dimensionality reduction, simplified input features and active sampling. As discussed in Section II, the latter approaches show numerous advantages in computational costs and classification performance.

Inline with these recent approaches, this paper explores the use of simple features and reduced dimensionality together with deep learning techniques to improve odour classification using time-dependent MOX sensors in the presence of humidity and temperature changes.

II. Related Work

Finding solutions to monitoring air quality in human habitats [2] is crucial in closed environments where air is recycled such as in modern energy-saving homes, industrial sites, aircraft, or greenhouse farms, as well as inside space travel cabins or the International Space Station. Gas monitoring allows prevention and early warning of dangerous gas accumulation in homes and can also serve to monitor home activities [3].

The development of electronic sensor arrays in the mid 1990s based on Conducting Polymers [4] or Metal-Oxide Sensors [5] has opened the area of electronic nose (EN) devices with advantages in low-cost, availability and connectivity. EN devices incorporate classes of sensors designed for the detection of one specific pollutant gas (such as: CO, CO_2, O_3, NO_x, NH_3, SO_x, H_2S and VOC′s[1]) but soon evolved into multi-gas sensors for applications such as detection of insecticides, nerve agents or refrigerant gases [6].

A. Electronic Gas Sensors

The MOX gas sensors based on n-type oxide semiconductors such as SnO_2 detect gases from a change in the electrical resistance of a porous sensing body. The device consists of

[1]Volatile Organic Compounds: ethanol, propanol, butanol, acetone, toluene, benzene, xylene, n-octane, methane, cyclohexane, trichloromethane, tetrachloromethane, tetrachloroethylene

978-1-5386-5502-3/18/$31.00 ©2018 IEEE
DOI 10.1109/BDCAT.2018.00016

two metallic contacts deposited on a poly-crystalline oxide pallet mounted on a micro-heater which maintains the required operating temperature [7], [8].

Reviews from [9] and later by [10] provide details on physico-chemical phenomena that characterise MOX sensors and explain the influence of temperature and humidity on MOX response as strongly related to surface reactions (adsorption) and humidity induced effects can be counteracted by heating the device to temperatures above 400°C.

Sensors responses are therefore dependent on their detection principle and extracting reliable data from these signals has to eliminate device-own noise, false signal (detected from other sources, interference) and device response to environmental factors (temperature, humidity, light, dust). Moreover, such time-series of detector response can have unexpected or variable window widths–bringing increasing complexity into the analysis.

While MOX gas sensitivities are excellent (at 1 ppm[2], for certain models going down to 1 ppb[3]) with high selectivity, durability and ruggedness at low cost, the major limitation of currently available detectors is their sensitivity to changes in humidity, temperature and gas flow rate [6], [8]. These effects are observed as drifts in response baseline, which is the sensor's initial resistance in dry clean air [11] .

B. Drift compensation methods

To compensate for environmental influences or detector aging which can change the baseline signal, re-calibrations are required to restore models accuracy at weeks intervals, and procedures are time consuming and expensive [12]. Finding efficient methods to achieve reliable drift compensation (corrections) can improve calibration life and have spun significant research.

Among drift compensation methods as signal baseline subtraction, signal corrections: univariate (calibration-derived multiplication factors) or multivariate (linear discriminant analysis or principal component analysis) the univariate methods based on calibration achieve much better classification rates [13]. To ensure calibration stability, a 3-month calibration regimen is advised for best results.

Changes in calibration parameters are proven to be due to baseline drifts which cannot be corrected by multiplicative or differential methods [8]. Observing that detectors response R is a function of time-dependent humidity H(t) and temperature T(t), they expressed the measured gas concentration C as:

$$C = g(R, H, T) + E$$

with g a time-dependent function and E the error level. To determine the function g, or in fact, to predict C, they used a tree layer feed-forward artificial neural network (ANN) with 4 input neurons, 3 in the hidden layer and 1 in the output. The predicted C is within 3% error level and therefore drift-corrected when employing ANN, while the created model renders ambient-independent sensor readings.

[2]parts per million
[3]parts per billion

Other approaches for drift compensation use multivariate methods like PCA and CPCA (joint diagonalisation PCA) find components of the drift variance that are common to several gases in the feature space [14]. This ensures a correct classification, less dependent of the gas type and drift-induced errors.

Advanced multivariate methods such as Self-Organising Maps (SOM) [15] or adaptive SOM (based on SOM and component removal) [16] have obtained improved error rates of about 20%. It has been noted that unsupervised SOM methods are unreliable for overlapping classes as the reference pattern may follow a different class [12]. They propose methods based on sliding window wavelet decomposition de-compose data into fine and coarse time-scales as noise appears in the finer scales and drift is captured by the coarser scales. Using Orthogonal Signal Correction (OSC) and Component Correction PCA (CC-PCA) - they tested 17 conductive polymer gas sensors over a period of ten months. They show that OSC - a technique used in spectroscopy to remove baseline trends - is suitable for drift compensation.

Both methods perform well on a reduced training size, with only 10 training samples from the reference class. For effective validation they observe that test data has to be sampled at time intervals situated after the time of train intervals.

Ref. [17] proposes λ-SVM, a Byes consistent algorithm for multi-class classification derived from the Inhibitory SVM formalism (equivalent to λ=1) designed for optimizing non-linear problems and define a range of values for λ that ensures low training times. They reached a classification accuracy of 82.6%. To enable classification of continuous incoming olfactory data, [18] has also developed a spiking neural network with bio-inspired architecture (insect olfactory system).

C. Novel drift compensation methods

More recent methods for controlling drift calibration approach data dimensionality reduction through reducing the number of experiments and active sampling.

As opposed to passive sampling which is uniform (non-adaptive) considering identically distributed and independent observations, the active (adaptive or controlled) data sampling can maximise learning while reducing the number of samples. The active sampling methods (query learning, instance selection or sequential sampling [19]) were proven superior in terms of generalisation error and reducing data dimensionality but they are not always suitable for real-world pattern recognition problems involving noisy data.

One of such approaches is optimising the choice of sample concentration so that it minimises the cross-validation results for a given classifier (the multi-class Inhibitory SVM, ISVM) [20]. Comparing accuracy for passive and active sampling, the active sampling can improve results when little information is available as training samples will have a higher contribution.

Ref. [21] have designed a method for *active sensing* allowing to discriminate multiple odours with one detector by adaptive and *inverse* temperature modulation (dependent on

a closed-loop feedback where detectors response controls the temperature modulation). This method leads to a reproducible response pattern for each odour and improvements in classification (by SVM, using the `libsvm` toolkit). Training was done on 3000 random vectors (5% of the 60,400 concentrations) reaching 92% accuracy.

Ref. [22] proposes the *transfer sample-based coupled task learning* (TCTL) based on *transfer learning* and multi-task learning (MTL) - i.e. learning multiple models simultaneously and share information across models to improve accuracy. Given labelled samples without drift (source domain) and a small set of transfer samples as inputs, TCTL simultaneously learns a prediction model for data in the source domain and one for data in the target domain (from the device with drift). The transfer samples are incorporated into a regularisation term of the objective function.

The *Direct Standardisation* (DS) is proposed by [23] in order to extend calibration models by mapping signals from the reference unit (without drift) to other detection units (or same detection unit later in time) using a reduced number of transfer samples. The transformation relationship is :

$$S_{\text{master}} = S_{\text{slave}} \cdot F$$

where S are the response matrices and F is the transfer matrix which can be derived from measured transfer matrices. This method assumes a linear relationship between signals and was applied previously in near-infrared spectroscopy.

When the master calibration model is transferred several times the prediction error remains constant. Also, slave units can be trained with a smaller set of transfer samples (60% less samples) coupled with DS resulting in same prediction error as if calibrated with the entire set of calibration samples.

Using a wrapper approach to investigate optimal feature selection for MOX detectors based on SVM all-against-all classifier and a subset of data points (responses at 6 time points for 12 sensors) [24] find that using all available data points for each sensor does not perform better than sub-sampled sets of simple features. Also, clustering properties of the data and correlation of detector responses can influence classification performance (but not in an obvious way). Choosing simple features lead to better results than derived features (response maxima, area, moving averages, fast Fourier transforms or discrete wavelet transform) and improved performance is obtained when using a wrapper approach, although a filtering approach can be useful.

D. Contribution

The methods for drift compensation have reached a turnaround in complexity, starting from basic electrical and thermal signal compensation, to complex feature engineering, signal de-convolution through univariate and multivariate methods (removing the noise and drift components of signal), culminate with neural networks and complex SVM with bio-inspired architectures, and finally achieve true optimal solutions through dimensionality reductions like active sampling, sub-sampling and transfer learning.

Summarised in Table I), these findings encourage the idea that active sampling with simple features selection with various degrees of correlation, coupled to deep learning models can provide drift compensation methods that are cost-effective, faster and with improved accuracy, hypotheses that are tested in this work.

The proposed active sampling generates a new class-balanced dataset, firstly selecting data within an exposure interval (considering the time sequence) then generating the balanced classes by random selection. While the new dataset is no longer a time series, the advantage is that new and old data are learned together as in transfer learning.

These approaches are observing the latest developments in this field aiming at a fast and reliable model generation– requiring simple implementation and reduced data size for *calibration*–as model learning.

TABLE I: Performance comparison for passive and active sampling

Method	Metrics	Result	Refs.
Passive Sampling			
Univariate (calibration factors)	better than	multivariate (LDA, PCA)	[13]
SOM, adaptive SOM	error	20% lower than PCA	[16]
ANN (4:3:1)	error on C	<3%	[8]
OSC, CC-PCA	accuracy	max. 97%	[12]
λ-SVM	accuracy	82.6%	[17]
Active Sampling			
selective sampling	error	exp. decrease	[19]
Inhibitory SVM	error on C	2.26%-26.13%	[20]
random selection	accuracy	92%	[21]
transfer learning	accuracy	90%-99%	[22]
DS	error	4% (30% better)	[23]
sub-sampling with SVM	better than	all data sampling	[24]

III. Methodology

The proposed research investigates the use of active sampling in connection with artificial neural networks (ANN), XGBoost and radial-SVM classifiers aiming to improve classification performance. Results are supported by cross-validation. The importance of co-variates in the dataset has been assessed using multi-linear regression and PCA to understand their contribution to the model.

The applied methodology includes the following steps:

- Data set presentation, exploration and pre-processing
- Data structure analysis, active sub-sampling and normalisation;
- Designing the ANN network by testing a set of hidden layer designs and sample sizes with cross-validation to find an architecture that allows for increased learning without over-fitting;
- Generating and optimising models: ANN, XGBoost, SVM;
- Evaluation and discussion.

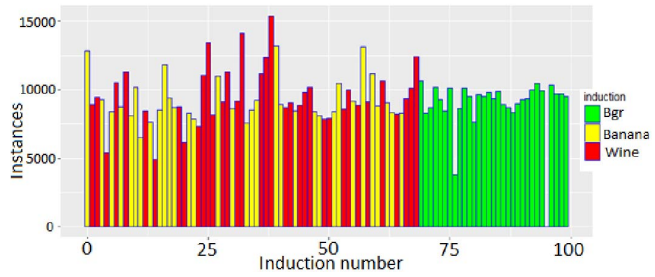


Fig. 1: Initial dataset: distribution of number of entries for each induction.

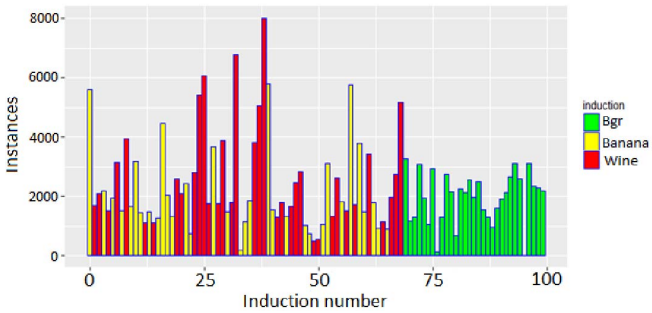


Fig. 2: Sub-sampled dataset: distribution of number of entries for each induction.

A. *Data set presentation, exploration and pre-processing*

The data used in this work has been generated in a series of experiments by [25] to demonstrate that the de-correlation of temperature and humidity from detectors' response can improve model prediction performance. Their original dataset (without de-correlation corrections) available to download from the UCI Machine Learning Repository. The "Gas sensors for home activity monitoring Data Set", presented as a multi-label, multivariate time-series (919438 instances) has been used in the present work.

The first data set consists of MOX sensor response collected from 8 MOX sensors exposed to two stimuli (banana and wine odours) for time intervals between 10 minutes and 1 hour along with corresponding baseline signals (no stimuli or background signal) creating a set of 34, 36 and 30 exposures for banana, wine and baseline samples, respectively. The signals were recorded with a sampling rate of 1/3600 $[s^{-1}]$ as time line. A second dataset contains the exposures (numbered from 0 to 99) showing for each exposure (induction) the exposure type, the starting time and the duration of exposure.

While the data is presented as a time series, the classification task does not require a temporal sequence for prediction. In this work each line of the dataset is considered as an independent vector of features (detector responses).

In pre-processing, the recording time (column "time" in the first dataset) has been converted to minutes. The induction duration (as hours in the second dataset) has been converted into minutes. The two datasets have been joined based on the induction number (id), attaching the labels column and the starting time and duration columns. For the analysis data has been normalised and the label column has been transformed by "one-hot-encoding"[4].

For each induction the data is recorded from before the start of the exposure (negative time) and continues after the exposure stopped (positive time beyond the exposure duration). Data sub-sampling was performed by selecting vectors recorded at times 2 minutes after the start and 2 minutes before the end of each induction. This choice is expected to eliminate transition signals at the beginning or end of exposure due to either sensor instabilities at the beginning of measurement or gas flow (uneven concentrations).

[4]For three levels of the labels (1, 2, 3) the *one-hot-encoding* changes label "1" to 1, 0, 0; label "2" to 0, 1, 0 and "3" to 0, 0, 1 generating three label vectors.

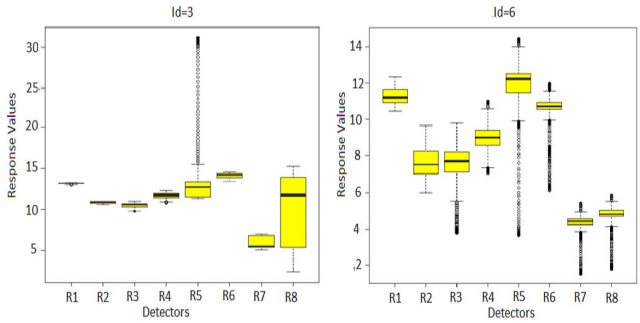


Fig. 3: Boxplot of detectors' responses for inductions 3 and 6.

In Fig. 1 and Fig. 2, the distribution of number of entries (response vectors) for each induction are shown for the initial data and for the sub-sampled data, respectively. Induction types are indicated in colours (background: green: banana: yellow, wine: red). It can be noticed that induction 95 is missing, leading to a total of 99 inductions.

No corrections were performed to de-correlate the temperature and humidity influences (as proposed by [25]). The resulting dataset has 190000 rows and was stored in a csv file.

Examining boxplots of response signal for each detector across various inductions shows that the signal from each detector has no specific pattern or repeatable response for similar inductions. Therefore, using the median of responses will not be a good choice. Also, removal of outliers will significantly affect the data as the outliers are numerous and their contribution to the overall response may have significance. Examples for inductions 3 and 6 are shown in Fig. 3 where induction 3 is for "wine" and 6 is for "banana" odours. The boxplots correspond to data from the initial (full-size) dataset. Regarding the time intervals for each induction, these have an irregular distribution shown as the height of bars ("instances") in Fig. 1 and Fig. 2.

More information on responses across all inductions are shown in Fig. 4 and sustains the general observation of

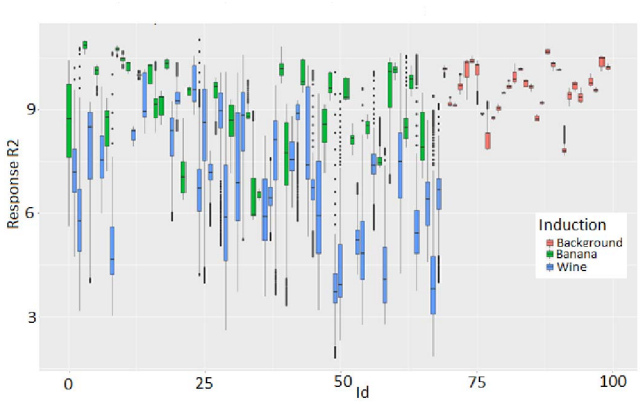


Fig. 4: Boxplots of responses for detector R2 across all inductions.

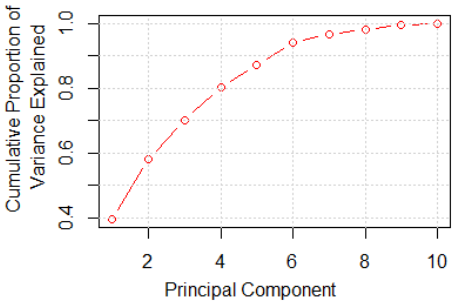


Fig. 5: Cumulative proportion of variance explained with PCA.

noisy and unstable detector responses. This chart shows the boxplots of response R2 (detector 2) across all 99 inductions, illustrating how different the signal for the same detector can be (even if the induction type is the same).

While it is expected that each detector has a specific distribution of response for each type of odour, these are overlapping or have very large variances, due to drifts induced by detector aging or variations in temperature and humidity.

B. Active sub-sampling

While removing the first and last minutes of exposure is a type of active sub-sampling applied for reasons of sensors' operation, other methods of active sub-sampling can be tested by selecting most or least correlated data. Performing Pearson correlation on the original dataset, two groups of strongly correlated responses (coefficients above 80%) are found: R1, R2, R3 and R4 and R7 and R8. The two separate correlation groups are due to the type of temperature regulation implemented for each (fixed temperature for the first group and reactive temperature control on the second group) leading to different response patterns.

It may therefore be possible to sub-sample the dataset by using only selected detectors from the groups R1-R4 and R7-R8. Interestingly, the influence of humidity and temperature do not appear as strongly correlated with responses.

Another possibility to extract the most meaningful data is to choose the variable (feature) having the highest coefficient in a regression analysis indicating its high explanatory power. A multi-linear regression analysis (package `lm` in R) finds highest coefficients for responses R1, R2, R3 and R5 while all 10 features are highly significant. Nevertheless, the obtained Multiple R-squared of 0.5421 shows a poor performance of linear regression as the model involves a more complex, non-linear relationship.

The PCA analysis searches for features with higher explanatory power in an orthogonal space where new variables are generated as linear combinations of initial features. Here, the first two components PC1 and PC2 can explain 39.5% and 18.6% of the variance, respectively (Fig. 5, Appendix). Their directions in the new feature space indicate PC1 having as main components R1, R2, R3, R4 while PC2 has as main components R5, R7, R8. While PC1 and PC2 can explain over 58% of the variance this is not enough for reaching high prediction accuracy. It also shows that the majority of variables (responses) have to be involved in generating the model. No separation of classes has been achieved indicating that PCA alone is not suitable for prediction.

As a conclusion, the active sampling by reducing the number of features (responses) is not fully justified here as this may affect the completeness of the model.

Other types of active sampling have been applied in this work by selecting data recorded within the time interval of stable detector operation (2 minutes after start, 2 minutes before end of each induction), then by sampling equal numbers of vectors from each class (improved version of stratified sampling). As the data is not used as a time series this allows for random sampling across all data, creating a mix (or superposition within the same model) of old patterns and new patterns (similar to the methods of transfer sampling proposed by [22]) enabling learning new behaviour in the context of (related to) previous patterns.

IV. Implementation

The implementation includes the following steps: data preparation and active sampling, testing for various types of ANN architectures (number of nodes and sample sizes) using a range of k-fold (10 - 50 folds) cross-validation, choosing a suitable network, implementing weights updating methods (two), training the model for various hyper-parameters with each method, choosing the optimal settings and validating and testing the model using 5-fold cross-validation. Re-sampling (10 times) with 3-fold cross-validation, tuning, validation and testing were also performed for XGBoost and radial-SVM.

A. Data preparation

The data set is biased as the exposure time (window) is different for each induction. This favours a certain behaviour related to the exposure with the longest time, leading to unbalanced data. In this work, balanced sets have been chosen by random sampling without repetition the same number of vectors for each target class. The final (balanced) dataset

TABLE II: Dataset structure details

	total size	size per class	proportion
training	135000	45000	0.75
validation	27000	9000	0.15
test	18000	6000	0.1
data set	180000		1

(180000 rows) has been split into subsets for training (75%), validation (15%) and test (10%) as shown in (Table II). The datasets are sampled in succession and the already used vectors are removed from the initial data so that they are not available for sampling in the next set. This ensures the split data has no cross-contamination (common vectors). As the length of the initial dataset with all inductions contains 231000 rows, the balanced dataset extracted contains 75% of the initial data.

The data has been normalised (interval [0,1]) and the classes column has been used as factor for SVM and XGBoost. For the ANN the classes column has been converted using "one-hot-encoding". The dataset has been split into training, validation and testing sets. For the numerical experiments (training) the training data set has been used, while the testing and validation datasets are kept for testing.

B. Machine learning approaches

The performance of several machine learning algorithms (MLA) that can generalize complex non-linear patterns–ANN neural networks, XGboost–has been tested in connection to the newly generated dataset obtained by active sampling.

Non-linear MLA are expected to augment the classification power as linear methods cannot explain all variance implying that a more complex, non-linear model is required (e.g. linear regression gives a low multiple R-squared of 0.5421, as in Section III-B).

For its flexibility and ease of programmable approach the `neuralnet` library in R has been used to implement several configurations of neural networks. While there are numerous variants of network architectures, a suitable network (with one hidden layer) has been chosen by observing the network performance function of the sample size and number of nodes in the hidden layer using multiple k-folds cross-validation (described here in Section V-A).

The `XGboost` algorithm in R has been preferred for its high performance in terms of solving non-linear multivariate problems and for its speed (based on multi-threading parallelization). The `SVM` radial kernel algorithm has been used for its versatility in solving non-linear problems and for its many strengths in dealing with multi-class classification.

These algorithms are based on significantly different approaches and comparing their performances (as the sum of square errors and or accuracy) can provide insight into whether the active sampling has enabled reaching a stable and more accurate solution, independent of external influences and the inherent detector aging and measurement noise and also independent of the type of classification algorithm used.

C. Neural networks

Neural networks generate classification models using a process that imitates neuron connections and decision making in the human brain. Neural networks learn complex relationships between features by constructing and solving a linear system of equations in several steps (neuron layers), starting from all features (input variables) and solving to fewer variables (thus encoding information into more relevant outputs - which are taken as the input for the next layer) and finally solving for the desired number of response variables.

To speed up reaching convergence, an activation function (usually a step function with limits between 0 and 1 like *sigmoid*, *hyperbolic tangent*, *relu*) is applied in order to transform the output of each layer to either 0 or 1; thus generating inputs for the next layer. In *backpropagation neural networks* the error at the output layer can be improved by re-adjusting the weights (coefficients in the systems of equations) in the direction and amount required to improve the error, in many successive runs, until convergence is reached (overall error is smaller than a given threshold).

In this work the `neuralnet` library in R by [26] has been used to implement the ANN classification model. The *neuralnet* function includes several backpropagation methods, out of which the resilient backpropagation (`rdprop+`) has been used in this work ([27]).

The neuralnet function allows for custom setting of several hyper-parameters : the hidden layer structure (number of hidden layers and nodes in each layer), error type: sum of squared errors (SSE) or cross-entropy (CE) and activation functions : logistic (sigmoid) and hyperbolic tangent. The start weights can be randomly generated or assigned from previous steps. The weights update methods implemented and the hyper-parameters choice performed in this study will be discussed for each numerical experiment.

D. XGBoost

Developed by [28], XGBoost is an advanced gradient tree boosting algorithm that gained notoriety for its excellent performance in standard benchmarking as well as in many high complexity classification and ranking problems. Its performance is enhanced by a high scalability using parallel and distributing computations and out-of-core memory. XGBoost includes novel tree-learning algorithm optimisations in finding the best split as percentiles of features distribution and solving this globally across the entire tree, for all leafs simultaneously, while data is organised in a block (column) structure. Finding the optimal split from statistics within a block data structure allows for distributed computing and parallelization. XGBoost uses a regularised model which improves error levels and prevents over-fitting.

E. Support Vector Machines

Support vector machines (SVM) can generate non-linear learning models for data classification and regression analysis by mapping data vectors as points in a higher dimensional space where a suitable separation hyperplane can be found.

The separation margin on each side of the hyperplane is controlled by the Cost (C) parameter which decides the trade-off in the optimisation problem between maximising the margin and accurately classifying data points. A large C can lead to high classification error while a low C improves the error levels but can lead to a difficult to solve optimisation problem and also to over-fitting.

The projection to a higher dimensional space, known as the Kernel trick (proposed by [29]) performs a mapping assigning to the scalar product between vectors (i.e. $\mathbf{x}$ and $\mathbf{y}$) a specific function (which can be a polynomial, a hyperbolic tangent or other). When using a radial function described by: $K(\mathbf{x}, \mathbf{y}) = e^{-\gamma||\mathbf{x}-\mathbf{y}||^2}$; each point is characterized by a Gaussian distribution (described by the radial function).

The parameter gamma defines the radius of influence of each vector (i.e. a large gamma leads to a narrow influence region and ultimately to over-fitting) while a too small gamma can loose the specificity in the classification task leading to an imprecise boundary. The gamma parameter controls the trade-off between bias and variance in the model. Due to its flexibility and non-liner projection, the radial function SVM has been used in this work.

V. Evaluation

A. Choosing the ANN architecture

As the *neuralnet* function is known to work well with small inputs (several hundred rows) [30] batch sampling is performed to learn large input data.

There are numerous choices of hyper-parameters and layers & nodes architecture which have to be tested. As complex architectures are computationally expensive, a series of architectures with one hidden layer are tested as a function of the number of nodes in the hidden layer, for various sizes of the data (100–500 rows). To assess the results, a k-fold cross-validation (CV) experiment is performed with k taking successive values 10, 20, 30, 40 and 50 by recording the evolution of model and test accuracy.

The *neuralnet* parameters used here are: activation function: "sigmoid", error type:(SSE and these parameters are be kept for all experiments. The weights are randomly selected (by default) and 5 repetitions of the algorithm have been run, with the best output recorded. The k-fold CV is run in a double loop (for each k-fold CV and number of nodes in the hidden layer). The samples with sizes from 100 to 500 rows are randomly selected from the initial training dataset and confusion matrices are generated for both the train/test sets created by cross-validation from which accuracy is calculated.

The obtained results are shown in Fig. 6, where test accuracy is about 10% lower than model accuracy for all experiments and accuracy depends noticeably on the number of nodes in the hidden layer and on the sample size. The algorithm is not convergent for some of the tested configurations (which are not shown).

An interesting observation is that accuracy is high for low number of nodes at low sample sizes (e.g. for h=2 and S=100) and for high number of nodes at high sample sizes (h=9 and S=500). The configuration with h=7 appears to be over-fitted (with model accuracy above 95% and much lower test accuracy, 70%) and non-convergent for the larger sample sizes.

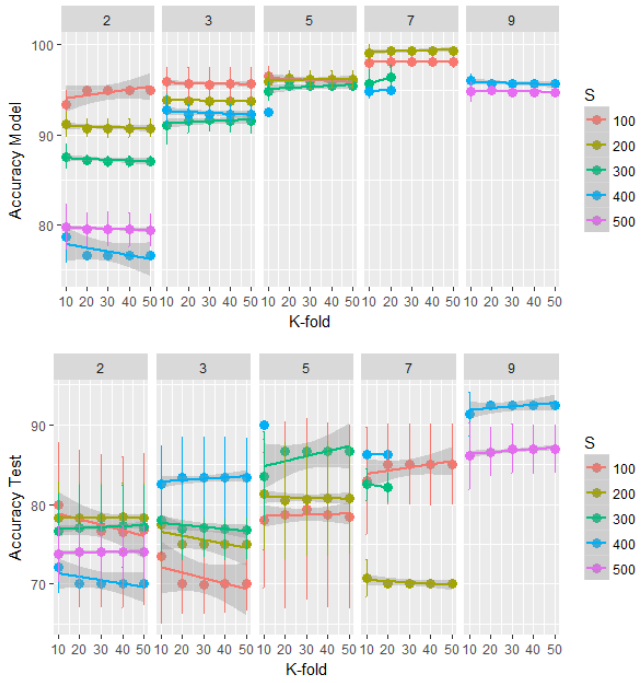


Fig. 6: Results on model accuracy and test accuracy with k-fold cross-validation. The number of nodes are shown on top panels, the data size (S) in legend. Error bars are the standard deviation over 5 repetitions.

The accuracy from k fold CV shows low sensitivity with increase in k but high sensitivity with the sample size and number of nodes in the hidden layer. This can be explained by observing that a more complex architecture can accommodate larger data size (a more complex model) while a reduced architecture maps well (generates a better model) for small data size. The best choices appear to be h=9 and large sample sizes (400-500 rows), achieving model accuracy above 95% and test accuracy of 92% and 87% respectively .

The following experiments will use this suitable architecture (one hidden layer, 9 nodes) to implement various batch data sizes for training the ANN.

B. Experiments with neural networks

The experiments were performed on a network with 10 inputs, 1 hidden layer with 9 nodes and 3 outputs (as one-hot-encoded class type). Two methods (M) for weights updates have been tested:

1) weights updates from previous batch, Method 1;
2) weights updates from previous batch (within one epoch) and averaged between epochs, Method 2.

The type of experiments performed to determine an optimal model are shown in Table III, where "LF" is the learning rate factor and "M" is the weights updating methods used. Three

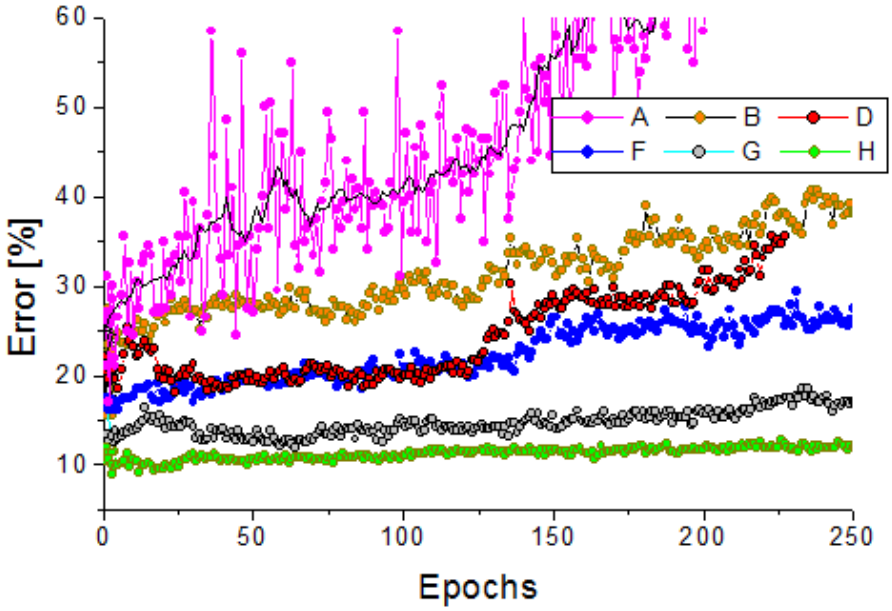


Fig. 7: Evolution of SSE error with number of epochs. The labels correspond to the ID of each experiment in Table III.

TABLE III: Choices of hyper-parameters and batch sizes for the `neuralnet` model.

batch size	batches per epoch	LF limits	Min. Error	Mean Error	Stdev	M	ID
1000	135	0.5; 1.2	22.0	44.7	13.5	1	**A**
1000	135	0.5; 1.2	22.3	37.9	5.7	2	**B**
450	300	0.5; 1.2	22.3	27.4	5.6	1	**C**
450	300	0.95; 1.2	18.2	24.1	4.7	1	**D**
450	300	0.7; 1.2	22.2	25.2	1.1	2	**E**
450	300	0.7; 1.4	16	28.0	5.1	2	**F**
150	900	09; 1.2	12.0	16.1	1.3	2	**G**
150	900	05; 1.2	9.0	11.3	0.7	2	**H**
150	900	0.7; 1.4	n.a	37.3	6.0	2	**I**

batch sizes (1000, 450 and 150 vectors each) where used; several changes in the minimum and maximum limits of the learning rate factor (LF) which has as default limits (0.5; 1.2). Experiments with various learning rate (LR) values have not shown any change in the algorithm convergence or error levels (not an active parameter).

Although a shallow convergence is obtained for all cases, two of the models show promising results: model D with a region of error decrease and model G (with low error levels) in Fig. 7 (red trend and gray trend, respectively). The methods D and G are showing low error levels and are expected to provide the best classification accuracy. The other methods (A-C, E-F) have increasing error with the number of epochs showing a lack of convergence of the ANN.

Considering the evolution of `SSE` for train an validation sets with number of epochs (Fig. 8) the optimal models were chosen as the ones generated after the first 25 epochs for model D and after only 5 epochs for model G. Their accuracy for

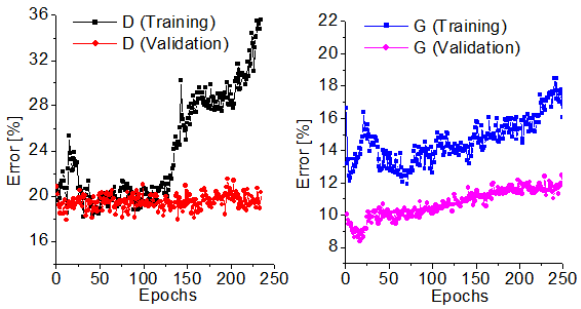
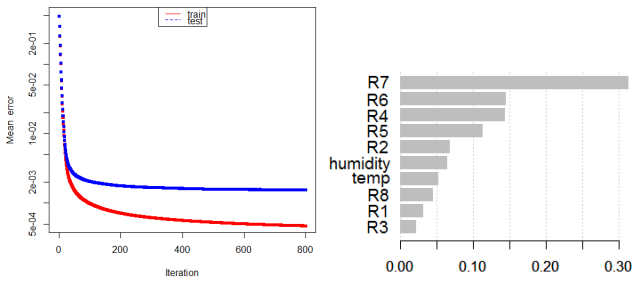


Fig. 8: Evolution of SSE error with number of epochs for the train and validation sets for models D and G (from Table III).

training and testing (obtained by 5-fold cross-validation) are presented in Table IV. The cross-validation results confirm that the model is not over-fitted as the test accuracy for both models are close to training accuracy.

C. Experiments using XGBoost

The dataset obtained using the same type of active sampling described in Section IV-A has been used with XGBoost to assess classification performance. The main parameters set-up for XGboost are: number of classes: 3, maximum tree depth ("max_depth"): 16, proportion of data instances to grow tree ("subsample"): 0.7, step size shrinkage ("eta"): 0.3, minimum sum of instance weight needed in a child (min_child_weight):12. For multi-class classification a "multi-softmax" algorithm is used and as evaluation metric the "mlogloss".



(a) Error during tuning for the train and test data.

(b) Feature importance.

Fig. 9: `XGBoost`

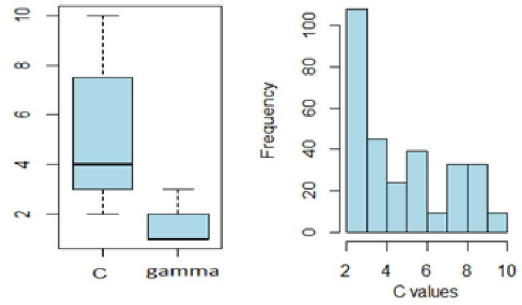


Fig. 10: Statistics on grid search parameters C and gamma from SVM tuning across the dataset.

While a 3-fold cross validation is performed by the XGBoost algorithm, in order to provide for data variability another 10 fold cross-validation loop has been designed to sample in a different manner the vectors from the initial dataset (in fact by re-applying active sampling 10 times). The train and test datasets initially created are joined after each selection, to allow for the XGboost's internal CV to be applied. The train-test split used by XGBoost cross-validation is 0.7:0.3. The tuning shown in Fig.9a shows a very fast convergence allowing for early stopping after the first 20 rounds. The performance results from XGBoost from 10 times 3-fold cross-validation (10 re-sampled datasets) are shown in Table IV.

The results obtained with XGBoost outperform those from neuralnet but show that the classification problem is relatively easy to solve when using the proposed active sampling. The low standard deviation between the 10 trials indicates that each new re-sampled dataset has a similar data distribution. The importance of features (responses R1 - R8, humidity and temperature) as calculated by XGboost algorithm provides alternatives for feature selection in active sampling (Fig. 9b).

D. Experiments with Support Vector Machines

In this work, the `e1071` SVM library in R has been used to assess the classification performance ofa radial basis SVM. The data is prepared as described in Section IV-A. The parameters C and gamma have been tuned performing a grid search against a range of C values and a range of gamma

TABLE IV: Comparative results on accuracy using neuralnet, XGBoost and SVM.

	Train		Test	
	Accuracy[%]	stdev	Accuracy[%]	stdev
neuralnet model D	91.17	1.42	91.26	1.24
neuralnet model G	89.51	1.18	89.10	1.33
XGBoost	99.96	0.14	99.43	0.36
SVM	97.71	0.53	96.62	0.85

values across 100 batches (each of size 1500) sampled from the dataset. As tuning is performed across tuning parameters and batches from the dataset, the tuning cannot be represented as a grid search, a boxplot and histogram where median values for C and gamma can be seen is preferred (Fig. 10), from which the median values of C=4 and gamma=1 are chosen for the tuned model. These best parameters have been chosen just outside the best tuning region (C=2, gamma=1) to avoid over-fitting. The results for the average accuracy in a 3-fold cross-validation (per batch) and for 10 re-sampled datasets are shown in Table IV.

E. Discussion

The evolution of the SSE error with the number of epochs shown in Fig. 7 shows a very slow convergence and shallow minimum regions. This slow learning of the network can have several causes: (i) when the neuron's output is close to 1 the learning rate becomes very small as the derivatives of the sigmoid function are very small; (ii) insufficient network complexity; (iii) data requires standardisation; (iv) data is too noisy and random (an ill-posed problem, with multiple solutions or no solution).

Observing the training and validation error over 250 epochs (Fig. 8) the validation errors lay in a constant range, at same level as the lowest level of the training error (or even below, for experiment G). This behaviour is unexpected as usually the validation error is higher than the training error. This can be explained as large datasets with balanced sampling have a good representation of training data pattern in the validation set, leading to similar error levels for validation. This is not consistent with over-fitting where usually the validation error is much higher than the training error.

This is due to the type of active sampling that restricts the size of the training set (to reliable interval) and selects data regardless of recording time (transfer learning) leading to higher accuracy as data is easier to fit into the model. While more complex ANN (2 hidden layers, e.g. with 5:3 nodes in each layer) have been tested, these configurations were computationally expensive and not even convergent within the maximum step and threshold settings of `neuralnet`.

The solutions found using the proposed active sampling with the neuralnet model satisfy conditions of simple and less expensive computational model and provides good accuracy levels. The particularities of low "learning rates" and flat test and validation error levels are a consequence of the type of problem (noisy input data) relaxed solver (network with 9 nodes, close to 10 inputs number) and also the inclusion

of correlated features which bring an excess of information leading to a flat cost functional (without obvious minima).

The classification accuracy from XGBoost and SVM models are also high and with low variances across 10-fold cross-validation and multiple re-sampling, as shown in the comparative table (Table IV). This indicates that results are consistent across various types of solvers.

VI. Conclusions

The proposed work has assessed one of the recent approaches for modelling MOX detectors calibration drifts using active sampling. The proposed active sampling is performed by choosing a class-balanced dataset where recording times are mixed, thus including in the model new data along with old data. This type of active sampling is consistent with the methods of *transfer sampling* proposed by [22] and data sub-sampling proposed by [24].

The results using ANN (neuralnet), XGBoost and SVM (radial function) algorithms show that classification accuracy is significantly improved when using a dataset that has been actively sampled as proposed here. The classification accuracy levels are high (above 90 %) and with small variances (lower that 2 %) across 10-fold cross validation and 5 to 10 times re-sampling. This is in agreement with above cited work and other results from active sampling (Table I).

The methodology used in this project has reached to powerful, non-linear machine learning algorithms but only as basic methods that allow direct parameter control in tuning and validation and provide an insight into partial results which have lead into making choices on sample and batch sizes or hyper-parameters. Due to time constraints, only one type of active sampling has been tested. However, we intend to eventually explore a more comprehensive parameter space using scalable data analytics techniques [31].

While advanced deep learning can be used (convolutional or recurrent networks, LSTM) here a basic use of neural networks allows a study of batch size versus network architecture, providing a means for network optimisation.

Further work will investigate more examples of active sampling considering choices of correlated features, examining limits of validity for sub-sampling and dimensionality reduction.

References

[1] G. F. Fine *et al.*, "Metal oxide semi-conductor gas sensors in environmental monitoring," *Sensors*, vol. 10, no. 6, pp. 5469–5502, 2010.

[2] J. Fonollosa *et al.*, "Human activity monitoring using gas sensor arrays," *Sens. Actuator B-Chem.*, vol. 199, pp. 398–402, 2014.

[3] M. Ogawa and T. Togawa, "Monitoring daily activities and behaviors at home by using brief sensors," in *1st IEEE-EMBS*. Lyon: IEEE, Oct. 2000, pp. 611–614.

[4] A. Hierlemann *et al.*, "Polymer-based sensor arrays and multicomponent analysis for the detection of hazardous orgagnic vapours in the environment," *Sens. Actuator B-Chem.*, vol. 26, no. 1-3, pp. 126–134, 1995.

[5] W. Gopel and K. D. Schierbaum, "SnO2 sensors: current status and future prospects," *Sens. Actuator B-Chem.*, vol. 26, no. 1, pp. 1–12, 1995.

[6] W. Bourgeois, A.-C. Romain, and R. M. Stuetz, "The use of sensor arrays for environmental monitoring : interests and limitations," *J. Environ. Monit.*, vol. 5, pp. 852–860, 2003.

[7] M. Tiemann, "Porous metal oxides as gas sensors," *Chem. Eur. J.*, vol. 13, pp. 8376–8388, 2007.

[8] F. Hossein-Babaei and V. Ghafarinia, "Chemical compensation for the drift-like terms caused, by environmental fluctuations in the responses of chemoresistive gas sensors," *Sens. Actuator B-Chem.*, vol. 143, pp. 641–648, 2010.

[9] N. Barsan and U. Weimar, "Understanding the fundamental principles of metal oxide based gas sensors; the example of CO sensing with SnO2 sensors in the presence of humidity," *J. Phys. Condens. Matter*, vol. 15, no. 20, pp. R813–R839, 2003.

[10] C. Wang *et al.*, "Metal oxide gas sensors: Sensitivity and influencing factors," *Sensors*, vol. 10, pp. 2088–2106, 2010.

[11] H. A. and R. Gutierrez-Osuna, "Higher-order chemical sensing," *Chem. Rev.*, vol. 108, pp. 563–613, 2008.

[12] M. Padilla *et al.*, "Chemometrics and intelligent laboratory systems drift compensation of gas sensor array data by orthogonal signal correction," *Chemom. Intell. Lab. Syst.*, vol. 100, no. 1, pp. 28–35, 2010.

[13] A. C. Romain and J. Nicolas, "Chemical long term stability of metal oxide-based gas sensors for e-nose environmental applications : An overview," *Sens. Actuator B-Chem.*, vol. 146, no. 2, pp. 502–506, 2010.

[14] A. Ziyatdinov *et al.*, "Drift compensation of gas sensor array data by common principal component analysis," *Sens. Actuator B-Chem.*, vol. 146, no. 2, pp. 460–465, 2010.

[15] M. Zuppa *et al.*, "Drift counteraction with multiple self-organising maps for an electronic nose," *Sens. Actuator B-Chem.*, vol. 98, no. 2, pp. 305 – 317, 2004.

[16] S. Marco, A. Ortega, A. Pardo, and J. Samitier, "Gas identification with tin oxide sensor array and self-organizing maps : Adaptive correction of sensor drifts," *IEEE Trans Instrum Meas.*, vol. 47, no. 1, pp. 316–321, 1998.

[17] I. Rodriguez-Lujan and R. Huerta, "A Fisher consistent multiclass loss function with variable margin on positive examples," *Electron. J. Statist.*, vol. 9, no. 2, pp. 2255–2292, 2015.

[18] A. Diamond *et al.*, "Classifying continuous, real-time e-nose sensor data using a bio-inspired spiking network modelled on the insect olfactory system," *Bioinspir Biomim.*, vol. 11, no. 2, p. 0260002, 2016.

[19] D. Cohn, "Improving generalization with active learning," *Machine Learning*, vol. 15, pp. 201–221, 1994.

[20] I. Rodriguez-Lujan *et al.*, "Chemometrics and intelligent laboratory systems on the calibration of sensor arrays for pattern recognition using the minimal number of experiments," *Chemom. Intell. Lab. Syst.*, vol. 130, pp. 123–134, 2014.

[21] P. Herrero-Carrón *et al.*, "An active, inverse temperature modulation strategy for single sensor odorant classification," *Sens. Actuator B-Chem.*, vol. 206, pp. 555–563, 2015.

[22] K. Yan and D. Zhang, "Calibration transfer and drift compensation of e-noses via coupled task learning," *Sens. Actuator B-Chem.*, vol. 225, pp. 288–297, 2016.

[23] J. Fonollosa *et al.*, "Calibration transfer and drift counteraction in, chemical sensor arrays using direct standardization," *Sens. Actuator B-Chem.*, vol. 236, pp. 1044–1053, 2016.

[24] T. Nowotny *et al.*, "Optimal feature selection for classifying a large set of chemicals using metal oxide sensors," *Sens. Actuator B-Chem.*, vol. 187, pp. 471 – 480, 2013.

[25] R. Huerta *et al.*, "Online decorrelation of humidity and temperature in chemical sensors for continuous monitoring," *Chemom. Intell. Lab. Syst.*, vol. 157, pp. 169 – 176, 2016.

[26] S. Fritsch, F. Guenther, and M. F. Guenther, "Package neuralnet," in *The Comprehensive R Archive Network.*, 2016.

[27] F. Gunther and S. Fritsch, "neuralnet: Training of neural networks," *The R Journal*, vol. 2, no. 1, pp. 30–38, 2010.

[28] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *KDD '16*. San Francisco: ACM, Aug. 2016, pp. 785–794.

[29] V. Vapnik, "The support vector method of function estimation," in *In Nonlinear Modeling: advanced black-box techniques*. Boston: Kluwer Academic Publishers, 1998, pp. 55–85.

[30] N. S. Keskar *et al.*, "On large-batch training for deep learning: Generalization gap and sharp minima," *CoRR*, vol. abs/1609.04836, 2016. [Online]. Available: http://arxiv.org/abs/1609.04836

[31] B. Veloso *et al.*, "Scalable data analytics using crowdsourced repositories and streams," *J Parallel Distrib Comput.*, vol. 122, pp. 1–10, 2018.