FACTA UNIVERSITATIS
Series: **Mechanical Engineering** Vol. 20, Nº 3, 2022, pp. 479 - 501
https://doi.org/10.22190/FUME220307022M

Original scientific paper

ASSESSMENT AND PERFORMANCE ANALYSIS OF MACHINE LEARNING TECHNIQUES FOR GAS SENSING E-NOSE SYSTEMS

Lubna Mahmood[1], Zied Bahroun[2], Mehdi Ghommem[3], Hussam Alshraideh[2]

[1]Engineering Systems Management Graduate Program, American University of Sharjah, Sharjah, United Arab Emirates
[2]Department of Industrial Engineering, American University of Sharjah, Sharjah, United Arab Emirates
[3]Department of Mechanical Engineering, American University of Sharjah, Sharjah, United Arab Emirates

Abstract. *E-noses that combine machine learning and gas sensor arrays (GSAs) are widely used for the detection and identification of various gases. GSAs produce signals that provide vital information about the exposed gases for the machine learning algorithms, rendering them indispensable within the smart-gas sensing arena. In this work, we present a detailed assessment of several machine learning techniques employed for the detection of gases and estimation of their concentrations. The modeling and predictive analysis conducted in this paper are based on kNN, ANN, Decision Trees, Random Forests, SVM and other ensembling-based techniques. Predictive models are implemented and tested on three different MoX gas sensor-based experimental datasets as reported in the literature. The assessment includes a delineated analysis of the different models' performance followed by a detailed comparison against results found in the literature. It highlights factors that play a pivotal role in machine learning for gas sensing and sheds light on the predictive capability of different machine learning approaches applied on experimental GSA datasets.*

Key words: *Gas sensor arrays, E-nose, Machine learning, Feature extraction, Feature selection, Classification, Regression*

Received: March 07, 2022 / Accepted May 10, 2022

Corresponding author: Mehdi Ghommem
Department of Mechanical Engineering, American University of Sharjah, P.O. Box 26666, Sharjah, United Arab Emirates.
E-mail: mghommem@aus.edu

1. INTRODUCTION

Gas sensing technology which now encompasses the E-nose systems has witnessed tremendous growth in the past few decades. Since the advent of E-noses in the late 20th century, gas sensing technology has experienced an accelerating trend in the development of different types of GSAs that can be employed in E-noses [1-3]. These developments come in light of different technologies that have been deployed for manufacturing gas sensors. Sensors based on metal-oxide (MoX) semiconductors [4,5], carbon-nano materials [6,7] and polymer composites [8,9] operate on electrochemical principles, whereas, acoustic [10] and infrared [11] sensors focus on non-electrochemical operating principles. As a consequence, E-noses are currently used in a plethora of applications that require detection of the gas type and in many cases, the estimation of the gas concentrations. Given the correlation of radon with the occurrence of lung cancer, Blanco-Novoa et al. proposed an IoT system for estimating radon concentrations that could further activate mitigation devices to reduce its concentration in indoor environments [12]. Besides air quality control, E-noses have been employed for detecting the freshness levels of different kinds of meat based on their odor level, thereby, showing promising results for food quality control [13]. Wireless sensor networks are also used to detect and keep ammonia gas levels in check to minimize the environmental hazard they might pose within the livestock environment [14]. Furthermore, the ability of the E-noses to offer a relatively inexpensive and fast mechanism for the detection of various chemical compounds has made them increasingly viable for medical diagnosis, for example, monitoring diabetes based on exhaled acetone levels [15].

Machine learning within the E-nose forms a vital component of the smart gas-detection platform. The sensing materials of the GSAs develop a sensory response, based on the type of sensor, to the exposed chemical compounds which is then exploited by machine learning algorithms for their identification [16]. Despite their supreme functionality, GSAs are often riddled with challenges that impact their selectivity, sensitivity, stability and their ability to render a viable sensor response [17]. While many of these challenges are influenced by the sensors' external environment and have been tackled by implementing different sensor fabrication technologies, some challenges such as sensor drift [18,19] remain widely adverse. The two primary types of drift include the 'first-order' drift which occurs due to the chemical interaction between the sensor surface and the exposed gas. This could result in aging and an irreversible poisoning of the sensor surface due to prolonged gas exposure. 'Second-order' drift, on the other hand, is attributed to external factors such as fluctuations in the environment or aberrant changes in the experimental conditions. To overcome the challenges of a single gas sensor, E-noses combine GSAs with machine learning to mimic the human olfactory system in detecting/identifying various chemical compounds.

Machine learning for E-nose systems generally addresses two primary tasks, classification in which the type of chemical compound is identified and regression in which chemical compound concentration is estimated [20]. Based on the sensory response to a particular chemical compound, which represents the sensor signal, machine learning techniques attempt to extract relevant characteristics from the sensor signals using feature extraction and feature selection [21]. This approach of feature extraction was implemented by Vergara et al. to classify six different gases, i.e., ethylene, ethanol, ammonia, acetaldehyde, acetone and toluene, keeping in mind the challenge of sensor drift [19]. An extensive dataset that rendered the gas type and concentration obtained from 16 MoX sensors, was used to implement Support Vector Machine (SVM) ensemble classifiers to improve gas detection and counteract sensor drift simultaneously. Akin to the work presented in [19], Adhikari and Saha proposed the use of Artificial Neural Network (ANN) as a deep learning network [22,23] and k-Nearest Neighbors (kNN) ensembles to correctly identify the six gases in the face of drift [24]. The dataset consisting of six gas classes, posed a multi-class classification task that was tackled using ensembles combining several machine learning algorithms to render the final classification result. In addition to being used for classification, the dataset was employed to perform regression to predict the gas concentrations. Rehman and Bermak proposed a machine learning algorithm that incorporated feature selection within the regression model [25]. Through feature selection, the predictive model was able to identify relevant characteristics from the sensor responses to help estimate the concentrations for each of the six gases. Using predictive models, feature extraction, and feature selection for classification and regression using different GSA datasets, previous studies have worked towards improving the performance of the E-nose system. Consequently, significant advancements in different machine learning techniques continue to help tackle the challenges faced by GSAs while ameliorating their performance for different applications.

This paper first presents the general computational framework used for smart gas sensing and discusses several predictive techniques used in an E-nose system. The framework is applied to assess the performance of different machine learning techniques using three different experimental datasets that were previously obtained from MoX gas sensors. The assessment entails a detailed analysis of their implementation on the selected datasets and a comparison of their results against those obtained from previously-published studies. Finally, we summarize the main findings of the present work and conclude with some remarks.

2. Machine Learning-Based Approach for Gas Sensing

Smart gas sensing using E-noses for classification and regression involves three vital steps as shown in Fig. 1. Multiple sensors in the GSA generate a unique sensory response in the form of a time-series signal for each gas they are exposed to, depending on the type of sensor being used and its operating principle. In several cases, the sensors undergo a physical or chemical change that results in a measurable sensor output. For instance, the sensor output could be measured as the change in electrical resistance or conductance of a sensitive material when exposed to a particular gas. As a result, the sensor response is typically referred to as a unique fingerprint that is specific to a certain gas and a particular concentration. Once the signal is acquired, the next pivotal step of signal pre-processing helps in the useful and meaningful transformation of the original raw data into representative features through feature extraction. This results in the conversion of the sensor response into a set of n features that capture vital information from the sensor response. In many cases, the transformation can render a high-dimensional dataset due to the large number of features that can be extracted. This might cause complexities while implementing machine learning models, thereby, resulting in the poor performance of the overall E-nose system [26,27]. This can be encountered using feature selection techniques that help retain important features subset from the original feature set to improve model performance [20,21,28,29]. Consequently, the generated dataset can be divided into training and testing datasets. The training dataset is used for model building, whereas, the testing dataset is used to test for the model performance. Classification-based models attempt to identify and segregate the different gases using class labels, for example, segregating the quality of wine using wine odors as conducted in [30], whereas, regression-based models aim to establish a relationship between the features and the gas concentration as a numeric variable [31].

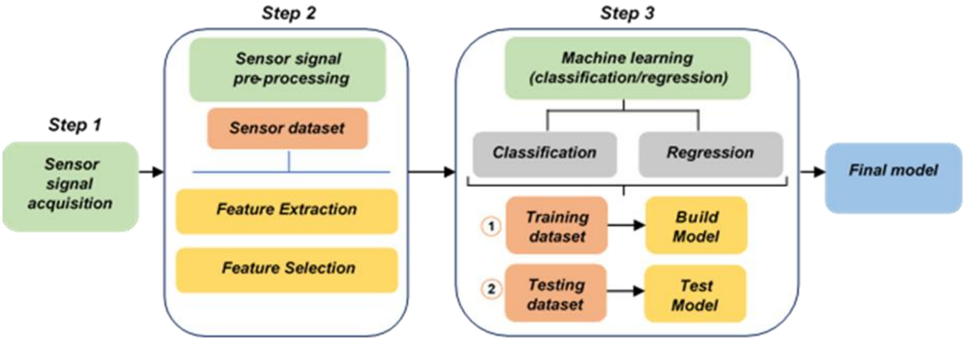

Fig. 1 Block diagram representation of the steps involved in the E-nose operation

Given the primary focus of this work on sensor data obtained from MoX sensors, Fig. 2 shows a typical MoX sensor response on exposure to a gas. The response depicts the change in conductance over time during the gas exposure with three different phases; Phase I (initial phase), Phase II (exposure phase), and Phase III (regeneration phase) [32]. It is quite evident that the raw time-series data from the sensor contains numerous data points. As a consequence, feature extraction and feature selection that form a subset of dimensionality reduction techniques, focus on extracting/selecting representative features from the sensor data [33]. The extracted features could represent information such as the slope of the signal in each phase, the rise time of the signal, the peak value of the signal, the trough of the signal and the overall average value of the signal. As a result, dimensionality reduction techniques are utilized to provide characteristic features that summarize a major portion of the information contained in the original raw data.

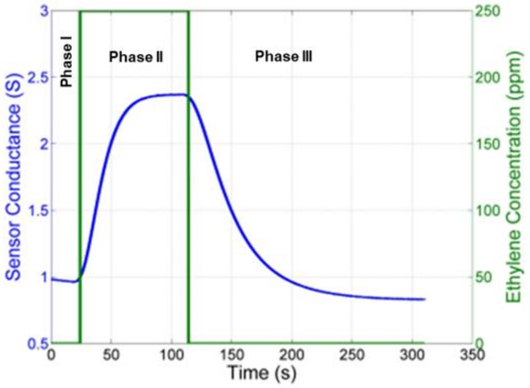

Fig. 2 Typical response of MoX gas sensor on exposure to ethylene. Adapted from [32]

3. Assessment of Machine Learning Techniques

The adopted E-nose machine learning-based framework is applied to three MoX gas sensor datasets to assess various machine learning models' performance. Whenever possible, a comparison of the adopted framework results with previously-published work is presented. Each dataset was obtained through different experimental protocols for detecting different gases and their concentrations [18,19,25,34,37]. The experimental protocol, assessment and the results for each dataset are discussed in the following sections.

3.1. Dataset 1- Experimental dataset of six gases with varying concentrations

This dataset contains measurements collected over three years for the detection of six analytes, namely, ammonia, acetaldehyde, acetone, toluene, ethanol and ethylene using 16 commercial MoX sensors in a GSA [19,25,34]. The complete experimental setup, fully controlled to provide high reliability and measurement reproducibility, consisted of four different types of MoX sensors (four of each type), thereby, resulting in 16 MoX sensors. Each gas and its respective concentration was exposed to the GSAs in no particular order to yield a complete dataset of *13,910* sensor signals, split into ten different batches. Further details regarding the experimental procedure, various gas concentrations tested and different batches can be found in the work presented by Vergara et al. [19].

3.1.1. Methodology

Following the signal acquisition, Fonollosa et al. resorted to extracting two types of features from the sensor response [19,34]. For r[k], that represents the sensor resistance at time step k, the steady-state feature ΔR represents the difference of the maximum resistance change and the baseline resistance value, and its normalized value ‖ΔR‖ is depicted as the ratio of this difference with respect to the baseline resistance value. The transient features represent the sensors' dynamic response by accounting for the rising/decaying potion of the signal using the exponential moving average (ema) for different smoothing parameter values. The complete feature dataset consists of 128 features (8 main features x 16 sensors) and can be found in [25]. Prior to the implementation of machine learning models, the features for each sensor were labeled and numbered accordingly.

3.1.2. Part 1: Machine learning – Regression

The main outline of the regression performed on this dataset followed the approach undertaken by Rehman and Bermak [25] for a fair comparison of the implemented models with the models presented in [25] and is shown in Fig. 3. For this analysis, we used data from *batch 1* to train the machine learning algorithm and develop the model, whereas, *batches 2-5* were used to test the model for the final prediction results. The final results were quantified by the mean absolute percentage error (MAPE) metric for numerical predictions. This process was repeated for each of the six gases to estimate their concentrations separately. During the cross-validation phase, the model hyperparameters were tuned through *parameter optimization* to avoid the occurrence of any model overfitting or underfitting.

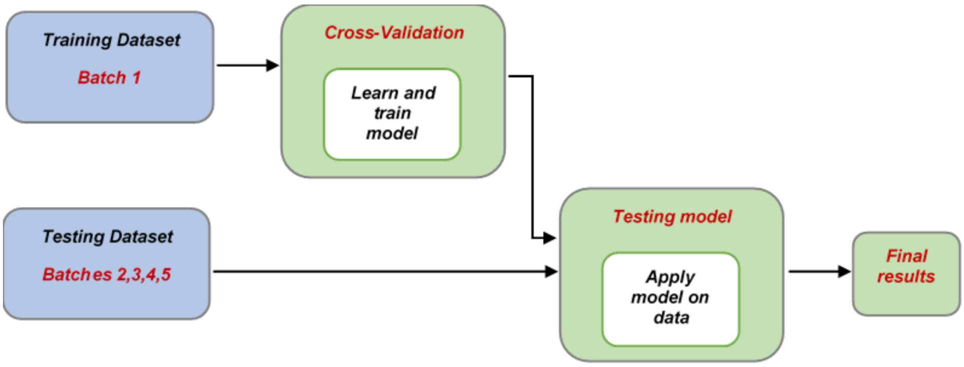

Fig. 3 Methodology outline for regression - part 1

The complete list of algorithms applied in the previous study [25] and those applied in this work are shown in Table 1. The kNN algorithm was initially applied to the training dataset using *k* as 5 and the distance metric as Euclidean distance. To optimize the model performance, these hyperparameters were tuned during the training phase. Prior to training the kNN algorithm, the training dataset was normalized to minimize the impact of the scale ranges for different features. For both the Decision Trees (DT) and the Random Forests (RF) models, regression was performed using the *least squares regression* splitting criterion. Regression results of the techniques used in this work for all the six analytes is shown in Table 2.

Table 1 List of machine learning models used in previous work and present work

Machine learning techniques	
Previous work [25]	Present work
Levenberg Marquardt (LMNN)	Bagging using Random Forests: Bagging (RF)
Scaled Conjugate Gradient (SCGNN)	kNN
Bayesian regularization (BRNN)	Vote using Gradient Boosted Trees and RF: Vote (GBT,RF)
Average Ensemble (AvgEnsemble)	RF(r)*
Support Vector Machines (SVM)	GBT
Gaussian Processes (GP)	Decision Trees (DT)
Random Forests (RF)	Vote using kNN, RF and DT: Vote (kNN,RF,DT)
Heuristic Random Forests (HRF)	

*Since RF is implemented in the previous and present work, the RF of this research is referred to as RF(r).

As shown in Table 2, in the case of e*thanol*, when testing batch 2, kNN (MAPE of *35.35%*) achieved the best performance in comparison to all the models reported in the previous study [25]. Despite the lack of high concentrations present in the training dataset, the present implementation of the models enabled to obtain relatively lower errors. This can be supported by the performance of DT (MAPE of *45.42%)*, Vote using kNN,RF,DT (MAPE of *51.05%*) and RF(r) (MAPE of *55.02%*), that had MAPEs lower than the HRF model [25]. For testing batches 2-5, kNN provided the least MAPE of *39.23%*, that was comparable with the MAPE of HRF. Other models that delivered a decent performance include the Vote using kNN,RF,DT (MAPE of *52.26%*) and RF(r) (MAPE of *59.18%*). For batch 2 of e*thylene*, kNN (MAPE of *15.66%*) demonstrated the best performance amongst the implemented models and those presented in the previous study [25]. In addition to kNN, Bagging using RF (MAPE of *21.82%*) and RF(r) (MAPE of *23.47%*) also delivered good performances. In case of batches 2-5, the lowest achievable MAPE was *58.22%* from Bagging using RF, wherein, the high MAPEs can be attributed to the sensor drift experienced over the course of the experiment. Nevertheless, Bagging using RF (MAPE of *58.22%*), kNN (MAPE of *64.97%*), DT (MAPE of *71.44%*) and RF(r) (MAPE of *63.03%*) performed better than SCGNN and SVM [25].

Table 2 MAPEs for six gases using different machine learning models - present work

Gas	Test batch	Machine learning model/MAPE(%)						
		Bagging (RF)	kNN	Vote (GBT,RF)	RF(r)	GBT	DT	Vote (kNN,RF,DT)
1	Batch 2	61.94	35.35	87.39	55.02	124.39	45.42	51.05
	Batches 2-5	60.35	39.23	72.08	59.18	89.74	72.35	52.26
2	Batch 2	21.82	15.66	61.29	23.47	102.55	33.16	32.98
	Batches 2-5	58.22	64.97	73.83	63.03	88.34	71.44	89.55
3	Batch 2	35.06	11.46	59.44	37.93	81.47	18.79	19.93
	Batches 2-5	54.23	38.22	81.15	56.15	106.31	39.79	41.69
4	Batch 2	22.71	55.05	25.99	20.47	39.74	38.73	37.65
	Batches 2-5	22.14	59.77	26.55	19.8	41.49	49.07	41.81
5	Batch 2	30.74	25.39	40.26	26.12	57.46	85.76	59.64
	Batches 2-5	33.17	32.54	53.75	32.09	77.3	86.25	58.54
6	Batch 2	38.2	28.28	41.74	40.02	43.46	52.22	59.00
	Batches 2-6	38.6	64.02	50.66	41.63	59.68	84.83	65.98

Gas 1 – ethanol, Gas 2 – ethylene, Gas 3 – ammonia, Gas 4 – acetone, Gas 5 – acetaldehyde, Gas 6 – toluene

When it comes to *ammonia*, the lowest MAPE of *11.46%* was achieved by kNN for batch 2 as seen in Table 2, which is lower than the error achieved by all the models presented in the previous study [25]. This was closely followed by DT and Vote using kNN,RF,DT with MAPEs of *18.79%* and *19.93%* respectively. As a consequence, kNN, DT and Vote using kNN,RF,DT collectively outperformed LMNN, SCGNN, SVM, GP and RF [25]. On the other hand, for testing batches 2-5, the lowest MAPEs were achieved by kNN (MAPE of *38.22%*) and DT (MAPE of *39.79%*). Hyperparameter tuning of kNN (setting $k=7$), helped kNN achieve the lowest MAPE in comparison to other models tested in this work, and those in the previous study [25]. For *acetone*, RF(r) provided the lowest MAPE of *20.47%* amongst all the implemented models. Although the best performance was attributed to SVM (MAPE of *10.16%*) from the previous study [25], the performance of RF(r) was comparable to that of HRF (MAPE of *20.44%*) that had the second-best performance. The training dataset had low frequencies of high concentrations, whereas, the testing dataset had much higher frequencies of high concentrations. Despite this discrepancy, RF(r) rendered the lowest MAPE of *19.8%* for testing batches 2-5 in this work. Besides RF(r), Bagging using RF (MAPE of *22.14%*) and Vote using GBT,RF (MAPE of *26.55%*) also delivered a good performance.

For testing batch 2 of *acetaldehyde*, kNN demonstrated the best performance with a staggering low MAPE of *25.39%*, followed by RF(r) (MAPE of *26.12%*) and Bagging using RF (MAPE of *30.74%*) (see Table 2). On the other hand, for testing batches 2-5, the best performance was observed for RF(r) with the lowest MAPE of *32.09%*, closely followed by kNN (MAPE of *32.54%*) and Bagging using RF (MAPE of *33.17%*). For *toluene,* batches 2-6 were tested as opposed to batches 2-5 since no measurements were recorded for toluene in batches 3, 4 and 5. Based on the regression results, the best performing model for testing batch 2 was kNN (MAPE of *28.28%*) with a performance comparable to HRF (MAPE of *21.48%*) and GP (MAPE of *26.58%*) from the previous study [25]. Other models that performed equivalently well included Bagging using RF (MAPE of *38.2%*) and RF(r) (MAPE of *40.02%*) which had MAPEs lower than the models used in the previous study [25]. For batches 2-6, the best results were attained using Bagging using RF (MAPE of *38.6%*), Vote using GBT,RF (MAPE of *50.66*%) and RF(r) (MAPE of *41.63%*), which collectively had errors lower than all the models except HRF from the previous study [25].

Going a step further, feature selection techniques of Sequential Forward Selection (SFS) and Sequential Backward Elimination (SBE), and the dimensionality reduction technique of Principle Component Analysis (PCA) were applied to investigate any potential change in the regression performance. PCA for the given dataset resulted in 128 PCs as the dataset contained 128 features. To minimize dimensionality while retaining maximum data variance, the limit for data variance was set to 95%, where the number of independent PCs that yielded *94.3%* variance in the data was found to be 4. A summary of the best performing models for each gas and the associated MAPEs is shown in Table 3. Feature selection or PCA helped to minimize the redundancy in the feature datasets and thereby, improve the performance of the implemented models. Moreover, models that did not perform well earlier were able to gain an advantage to their benefit.

In case of SFS, *82.61%* of the features selected represented the transient response of the sensors, whereas, quite a few pertained to the steady-state response of the sensors. Similarly, in case of SBE, the transient response of the sensors formed *74.76%* of the remaining features. Though more importance is attributed to the transient features, our work implies the importance of both steady-state and transient features for estimating the gas concentrations.

Table 3 Summary of best performing models and associated MAPEs - present work

Gas	Test batch	Machine learning model	Technique	MAPE (%)
1	Batch 2	RF(r)	SFS	29.75
	Batches 2-5	DT	PCA	36.59
2	Batch 2	Vote (kNN,RF,DT)	SBE	15.31
	Batches 2-5	RF(r)	SFS	47.69
3	Batch 2	kNN	SBE	8.19
	Batches 2-5	Bagging (RF)	SFS	11.38
4	Batch 2	Vote (GBT,RF)	PCA	13.45
	Batches 2-5	Bagging (RF)	SBE	29.59
5	Batch 2	RF(r)	SBE	12.94
	Batches 2-5	kNN	PCA	10.39
6	Batch 2	Vote (kNN,RF,DT)	PCA	39.9
	Batches 2-6	Bagging (RF)	SFS	38.48

Gas 1 – ethanol, Gas 2 – ethylene, Gas 3 – ammonia, Gas 4 – acetone, Gas 5 – acetaldehyde, Gas 6 – toluene

3.1.3. Part 2: Machine learning – Regression

So far, we focused on estimating concentrations of the six gases for batch 2, batches 2-5 (ethanol, ethylene, ammonia, acetone, acetaldehyde) and batches 2-6 (toluene). For the other batches of data, a different approach was followed, where, the machine learning algorithms were trained on *batches 1 to (m-1)* and tested on the *m*th *batch*. To clarify further, for ethanol, ethylene, ammonia, acetone and acetaldehyde, *m* ∈ *{6,7,8,9,10}*; whereas, for toluene *m* ∈ *{7,8,9,10}*. Since batch 6 was already tested for toluene in the previous analysis, batch 6 for toluene was not included in this section.

One of the main reasons for following this approach for batches 6-10 was sensor drift. The sensor data was collected over the course of 36 months, with batch 6 measurements beginning in month 17. Training on batch 1 where measurements were taken in months 1 and 2, and testing on batches with measurements taken much later in time could result in inaccurate prediction results due to changes in the sensor response over time. As a consequence, all the previous batches prior to the testing batch were included in the model training to estimate the gas concentrations. Results of the best performing regression models in the present work are shown in Table 4. Akin to the previous analysis, the kNN

Table 4 MAPEs for gases on testing batches 6-10 - present work

Gas	Training batches	Test batch	Machine learning model	MAPE (%)
1	Batches 1-5	Batch 6	kNN	22.17
	Batches 1-6	Batch 7	RF	22.22
	Batches 1-7	Batch 8	RF	23.73
	Batches 1-8	Batch 9	DT	38.14
	Batches 1-9	Batch 10	DT	54.02
2	Batches 1-5	Batch 6	DT	20.1
	Batches 1-6	Batch 7	RF	54.01
	Batches 1-7	Batch 8	RF	29.57
	Batches 1-8	Batch 9	RF	29.46
	Batches 1-9	Batch 10	RF	85.45
3	Batches 1-5	Batch 6	DT	14.58
	Batches 1-6	Batch 7	kNN	5.98
	Batches 1-7	Batch 8	kNN	58.9
	Batches 1-8	Batch 9	Bagging (RF)	8.89
	Batches 1-9	Batch 10	Bagging (RF)	52.21
4	Batches 1-5	Batch 6	RF	4.01
	Batches 1-6	Batch 7	RF	37.26
	Batches 1-7	Batch 8	RF	7.59
	Batches 1-8	Batch 9	Bagging (RF)	17.96
	Batches 1-9	Batch 10	DT	44.87
5	Batches 1-5	Batch 6	DT	19.87
	Batches 1-6	Batch 7	RF	18.55
	Batches 1-7	Batch 8	DT	30.82
	Batches 1-8	Batch 9	RF	6.35
	Batches 1-9	Batch 10	DT	59.61
6	Batches 1-6	Batch 7	Bagging (RF)	46.72
	Batches 1-7	Batch 8	Bagging (RF)	45.62
	Batches 1-8	Batch 9	DT	40.9
	Batches 1-9	Batch 10	Bagging (RF)	24.6

Gas 1 – ethanol, Gas 2 – ethylene, Gas 3 – ammonia, Gas 4 – acetone, Gas 5 – acetaldehyde, Gas 6 – toluene model was initially used with *k* as 5 using the Euclidean distance metric. The MAPEs are a result of this selection whereas, in some cases, the hyperparameters were tuned accordingly. As seen in Table 4, some models rendered remarkably low MAPEs such as kNN (MAPE of *22.17%*) for testing batch 6 of *ethanol*, DT (MAPE of *20.1%*) for testing batch 6 of *ethylene*, kNN (MAPE of *5.98%*) for testing batch 7 of *ammonia*, Bagging using RF (MAPE of *8.89%*) for testing batch 9 of *ammonia*, RF (MAPE of *4.01%*) for testing batch 6 of *acetaldehyde*, RF (MAPE of *6.35%*) for testing batch 9 of *acetone* and Bagging using RF (MAPE of *24.6%*) for testing batch 10 of *toluene*. The low MAPEs in majority of the cases except for when testing batch 10, indicated that using several batches of data for model training helped in addressing sensor drift. Using more recent data samples resulted in good predictions, with RF repeatedly delivering the best MAPEs, thereby, fortifying the ability of ensemble methods to deliver good prediction results.

Furthermore, SFS, SBE and PCA using 4 PCs were applied along with the machine learning models to improve their performance. For some models, the MAPEs remained the same despite the use of feature selection or dimensionality reduction. This was seen from testing batch 10 of *ethanol* using DT and SFS (MAPE of *54.02%*), batch 7 of *ammonia* using kNN and SBE (MAPE of *5.98%*), batch 10 of *acetone* using DT and SBE (MAPE of *59.61%*). Alternatively, in cases such as testing batch 6 of *ethanol* using kNN, SFS reduced the MAPE drastically from *22.17%* to *8.15%* and for testing batch 10 of *ethylene* using RF, SBE reduced the MAPE from *85.45%* to *80.03%*. Testing batch 8 of *acetaldehyde* with RF and PCA reduced the MAPE from *7.59%* to *6.24%*. Consequently, the use of either wrapper-based feature selection techniques or PCA helped to ameliorate the performance of the implemented models.

3.1.4. Machine learning – Classification

In addition to regression, classification was performed to predict the gas type using two different approaches:

- *Approach 1*: Training the algorithm on *batch m* and testing the model on the remaining consecutive batches, where $m \in \{1,2,3,4,5,6,7,8,9\}$.
- *Approach 2*: Training the algorithm on *batch (m-1)* and testing the model on the mth consecutive batch, where m $\in$ {2,3,4,5,6,7,8,9,10}.

The two approaches implemented are similar to those implemented in their respective previously published counterparts in [19,24]. This was done to enable a fair comparison of the results obtained. We implemented machine learning models such as RF, ANN, kNN, SVM, DT, Bagging using RF and Vote using GBT,RF for both aforementioned approaches. For ANN, a back propagation multi-layer perceptron (BP-MLP) model with 2 hidden layers and 15 nodes was used. Moreover, to get appropriate results, the training datasets were normalized to lie between -1 and 1 for SVM, ANN and kNN.

Using 10-fold, leave-one-out cross-validation, the classification accuracies for the machine learning models using Approach 1 are shown in Table 5. From Table 5, it is observed that training on batch 1 and testing on all consecutive batches results in the highest average classification accuracy of *60.75%* using SVM. A similar observation can be made for batch 2 (accuracy of *60.31%*) and batch 3 (accuracy of *63.59%*), which rendered accuracies in the same range but using kNN. For kNN, the value of *k* was initially set to 5, however, on tuning the model hyperparameters, setting the value of *k* equal to 7 yielded the best results for classification. Irrespective of the technique that rendered the highest accuracy, it was quite evident that the classification accuracy decreased on going from training batch 1 to batch 9 due to sensor drift. Despite the occurrence of drift, the classification accuracy of *60.75%* was higher than the accuracies rendered by ANN (accuracy of *42.43%*) and kNN (accuracy of *45.77%*) as reported in [24].

Our results were also compared to those presented by Vergara et al. [19], who used an SVM-based machine learning model for classification. On comparing accuracies, it was observed that the implemented SVM classifier (accuracy of *60.75%*) performed better than the SVM classifier in [19]. Besides SVM, kNN also provided a higher accuracy *60.31%* when trained on batch 2 and tested on batches 3-10 in comparison to SVM (accuracy of *58.35%*) in [19]. A similar claim can be made for training batch 4 (accuracy of *44.46%* with Bagging using RF), batch 5 (*47.23%* with SVM) and batch 8 (*56.44%* with RF). As a result, the classifiers in our research were successful in outperforming the SVM classifier in [19] in 50% of the cases.

Using the second approach, i.e., training classifiers using only the previous batch and testing on the next immediate consecutive batch, we set the classifiers to their initial parameters to perform classification. Results of Approach 2 for different models are shown in Table 6. It is quite clear that the accuracy for each test batch reaches its highest value when the model is trained with a previous batch. Moreover, classifiers such as Vote (GBT,RF), RF, SVM, Bagging (RF) and RF rendered quite high accuracies for 6 out of the

Table 5 Average classification accuracies for training with different batches – approach 1

Training batch	Test batches	Average classification accuracy (%)						
		ANN	kNN	SVM	RF	DT	Bagging (RF)	Vote (GBT,RF)
Batch 1	Batches 2-10	53.19	58.72	60.75	49.25	50.8	44.11	45.32
Batch 2	Batches 3-10	56.47	60.31	50.97	54.07	49.99	56.36	49.46
Batch 3	Batches 4-10	54.78	63.59	60.74	58.98	45.5	57.87	51.6
Batch 4	Batches 5-10	35.03	36.98	40.77	43.88	37.25	44.46	43.65
Batch 5	Batches 6-10	32.19	32.07	47.23	27.06	19.28	25.1	25.76
Batch 6	Batches 7-10	48.82	57.53	54.13	54.13	48.7	53.49	53.61
Batch 7	Batches 8-10	47.82	52.66	64.54	58.77	56.48	57.93	62.4
Batch 8	Batches 9-10	39.06	36.42	47.76	56.44	47.84	52.44	50.79
Batch 9	Batch 10	17.22	15.36	15.25	13.83	16.83	14.19	11.83

Table 6 Classification accuracies when trained on a previous batch and tested only on the subsequent batch – approach 2

Training batch	Test batch	Classification accuracy (%)						
		ANN	kNN	SVM	RF	Bagging (RF)	Vote (GBT,RF)	DT
Batch 1	Batch 2	45.42	70.58	53.62	77.25	74.36	77.33	76.29
Batch 2	Batch 3	85.75	91.55	79.07	91.42	94.96	88.91	77.8
Batch 3	Batch 4	61.49	70.19	67.7	83.85	81.99	78.88	80.12
Batch 4	Batch 5	58.88	50.25	54.82	89.34	89.85	93.4	88.32
Batch 5	Batch 6	57.3	37.48	67.96	39.74	38.57	37.96	37
Batch 6	Batch 7	69.28	74.4	78.19	74.31	74.62	71.52	65.15
Batch 7	Batch 8	89.12	70.07	92.86	86.73	83.33	80.27	66.67
Batch 8	Batch 9	54.04	48.51	72.98	92.98	91.91	86.38	73.62
Batch 9	Batch 10	17.22	15.36	15.25	13.83	14.19	11.83	16.83

9 cases, which reflects their superior performance when compared to the SVM classifier, SVM-based ensembles and the PCA models implemented in [19]. Thus, Approach 2 indicates that the most recent data samples provide a better prediction for the subsequent data samples, thereby, helping in reducing the impact of drift. As the test batch number increases, impact of drift also increases which further reduces the classification accuracy.

Using feature selection or PCA using 4 PCs, the average classification accuracy obtained for the machine learning models, either improved slightly or remained the same. Many classification accuracies especially when testing batches 9-10 and batch 10 remained the same, indicating that their optimal performance had already been achieved prior to feature selection or PCA. Just like for regression, more transient features were selected in SFS (*70% transient features*), and a higher proportion of transient features remained after implementing SBE (*76% transient features*). A similar observation was made by using filter-based feature selection to determine feature importance using weights. Using gini index, the weights of the top 30 features are shown in Fig. 4. The weights represented in Fig. 4 depict the decreasing order of importance for the features, where *27%* of the features represent steady-state features and *73%* represent transient features. With more importance given to the transient features, these features were capable of capturing more information from the sensor signals to yield good prediction results.

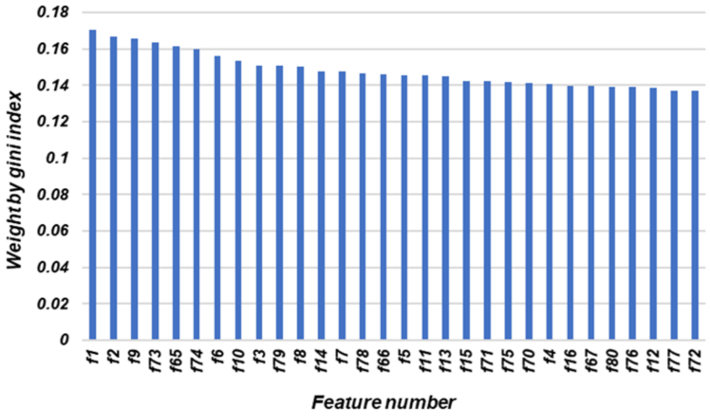

Fig. 4 Feature weights for top 30 features in the dataset 1

The impact of different number of features on the classification accuracy is shown in Fig. 5, which shows the average classification accuracies for the best performing models for training batch 1 and testing batches 2-10. RF and Bagging using RF reach their peak classification accuracies at the top 40 features, whereas, SVM uses the top 35 features to attain the highest classification accuracy. On the other hand. kNN, uses the top 50 features to reach its highest accuracy. Despite the different number of features used, it was evident that once the peak classification accuracy is attained by each model, the accuracy begins to stabilize and maintain a nearly steady value, with a minimum threshold of 30 features. With at least top 30 highest weighted features, the classification accuracy increased, following which it slowly waned off to provide a steadier value. Although, this filter-based feature selection technique used a different approach to identify the most relevant features, the results remained consistent with those obtained earlier using SFS and SBE.

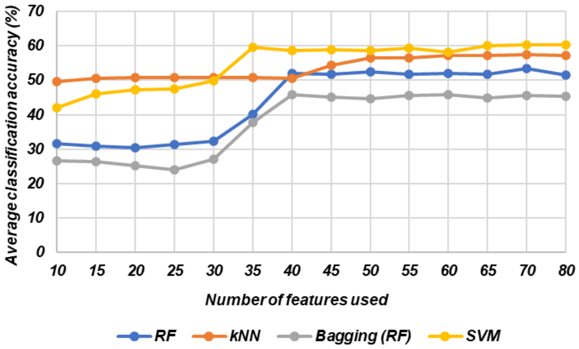

Fig. 5 Average classification accuracy using different number of features – dataset 1

3.2. Dataset 2 – Experimental dataset of four gases with varying concentrations

The following dataset was used for the detection of four gases: ethylene (Ey), ethanol (Ea), carbon monoxide (CO) and methane (Me). The experiment involved testing for 10 different concentrations of each analyte over a course of 22 days, resulting in a total of 40 different concentrations and *640* measurement samples. The experiment used eight MoX sensors in the GSAs referred to as chemical detection platforms (boards). Varying responses for each gas and its concentration were obtained using five identical boards, with any variability eliminated by maintaining the same configuration in all five boards. Further details pertaining to the experimental procedure can be found in the work of Fonollosa et al. [18].

The signals acquired from the eight sensors in a given board across different days, as reported in [18], are displayed in Fig. 6. All the sensors follow a similar trend during the *initial phase (I)* for *50s*, *exposure phase (E)* for *100s* and the *recovery phase (R)* for *450s*. The *exposure phase* can be easily identified by the dip in the sensor response when being exposed to the target gas and it is quite evident that the sensors render slightly varied responses on different days. This can be seen in Fig. 6, wherein, the stable response of each

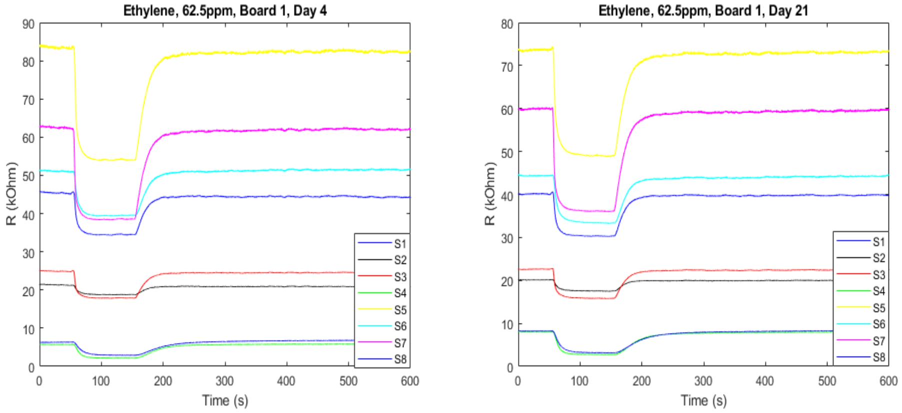

Fig. 6 Sensor signals acquired for detecting 62.5 ppm of ethylene on board 1, on day 4 and day 21

sensor on Board 1 varies from Day 4 to Day 21 for ethylene. Despite testing the same board over different days, the variability in the sensor responses can be attributed to drift, which could be a consequence of external factors.

3.2.1. Methodology

As an initial step, signal pre-processing using weighted moving average was performed to eliminate noise levels from the signals. Weighted moving average is the most common filter used for signal noise processing [35,36], which despite its simplicity, renders a high-attenuation ability, by reducing noise while retaining the shape of the signal. Given the sensor responses from the five boards, two main window lengths of 5 and 10 were used for noise removal, where Fig. 7 shows the impact of using weighted moving average with a window length of 5 on sensors 1 and 3 from board 1, for the detection of 12.5ppm of ethanol.

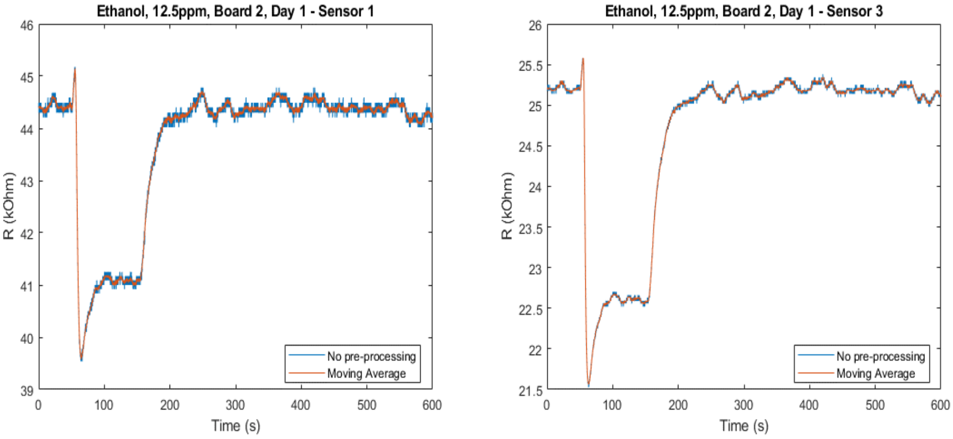

Fig. 7 Implementation of weighted moving average on sensor signals from sensor 1 and 3 on board 2 and day 1 for 12.5 ppm of ethanol

Following the pre-processing, relevant signal characteristics were extracted from $r[k + \Delta x]$ that represents the sensor resistance, with k as the discrete time-step, where $k \in [0{,}600]$ and Δx is used to indicate any incremental change in the time-step. The signal characteristics extracted from $r[k + \Delta x]$ in this work are discussed below:

Drop value as the signal transitions from the initial to the exposure phase is given as,

$$\boldsymbol{drop} = \underset{k=0,1,\ldots 50}{avg} \; r[k+\Delta x] \; - \underset{k=51,52,\ldots 150}{avg} \; r[k+\Delta x] \tag{1}$$

whereas, the transition rate is given by the slope,

$$\boldsymbol{slope} \; = \frac{r[k_1+\Delta x_1]-r[k_2+\Delta x_2]}{\Delta x_2-\Delta x_1}, \qquad k_1 = 50, k_2 = 51 \tag{2}$$

Average exposure value attained when the target gas is exposed to the sensor is expressed as,

$$\boldsymbol{average\ exposure} \; = \underset{k=51,52,\ldots 150}{avg} \; r[k+\Delta x] \tag{3}$$

Fall time (denoted by ***ft***) from 10% to 90% of the signal while transitioning from the initial to exposure phase is given as,

$$r[k+\Delta x]_{10\%} = \underset{k=0,1,\ldots 50}{avg} r[k+\Delta x] - \\ -\left[0.1 \times \left(\underset{k=0,1,\ldots 50}{avg} r[k+\Delta x] - \underset{k=51,52,\ldots 150}{avg} r[k+\Delta x]\right)\right] \tag{4}$$

$$r[k+\Delta x]_{90\%} = \underset{k=0,1,\ldots 50}{avg} r[k+\Delta x] - \\ -\left[0.9 \times \left(\underset{k=0,1,\ldots 50}{avg} r[k+\Delta x] - \underset{k=51,52,\ldots 150}{avg} r[k+\Delta x]\right)\right] \tag{5}$$

$$\boldsymbol{ft} = time|(r[k+\Delta x] = r[k+\Delta x]_{90\%}) - time|(r[k+\Delta x] = r[k+\Delta x]_{10\%}) \tag{6}$$

Rise time (denoted by ***rt***) from 10% to 90% of the signal while transitioning from the exposure to recovery phase is expressed as,

$$r[k+\Delta x]_{10\%} = \underset{k=151,152,\ldots 600}{avg} r[k+\Delta x] - \\ \left[0.1 \times \left(\underset{k=151,152,\ldots 600}{avg} r[k+\Delta x] - \underset{k=51,52,\ldots 150}{avg} r[k+\Delta x]\right)\right] \tag{7}$$

$$r[k+\Delta x]_{90\%} = \underset{k=151,152,\ldots 600}{avg} r[k+\Delta x] - \\ \left[0.9 \times \left(\underset{k=151,152,\ldots 600}{avg} r[k+\Delta x] - \underset{k=51,52,\ldots 150}{avg} r[k+\Delta x]\right)\right] \tag{8}$$

$$\boldsymbol{rt} = time|(r[k+\Delta x] = r[k+\Delta x]_{10\%}) - time|(r[k+\Delta x] = r[k+\Delta x]_{90\%}) \tag{9}$$

Fig. 8 illustrates the five different features in Eqs. (1)-(3), (6) and (9) extracted from sensor 4 of Board 1 for the detection of 12.5ppm of ethanol. The five features ensure that

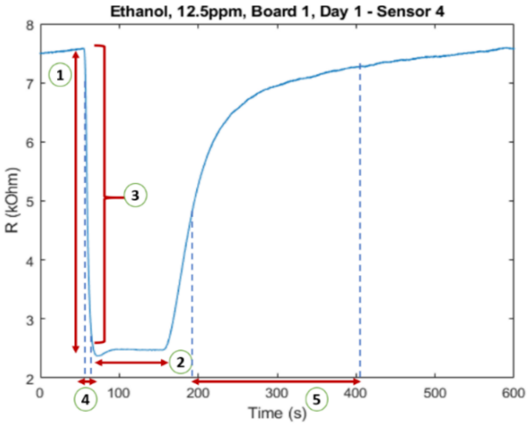

Fig. 8 Visual depiction of the five features extracted from sensor signal of sensor 4 on board 1 and day 1 for 12.5 ppm of ethanol

they cover all the three phases of the signal and are able to extract relevant information from the raw signal. The pre-processing and feature extraction was achieved for eight sensors on all five boards, for all the four gases, thereby, resulting in a total of 40 features (5 main features x 8 sensors) to be used for classification and regression. The summary of the complete feature dataset is shown in Table 7.

Table 7 Summary of the complete feature set - dataset 2

Sensor no.	Feature name/Feature number				
	Drop value	Average exposure value	Slope value in the IE	Fall time from 10% to 90% in the IE	Rise time from 10% to 90% in the ER
1	f1	f2	f3	f4	f5
2	f6	f7	f8	f9	f10
3	f11	f12	f13	f14	f15
4	f16	f17	f18	f19	f20
5	f21	f22	f23	f24	f25
6	f26	f27	f28	f29	f30
7	f31	f32	f33	f34	f35
8	f36	f37	f38	f39	f40

3.2.2. Machine Learning - Classification

The complete feature dataset of 40 features was subjected to machine learning for classification. 480 measurements (75% of the dataset) from boards 1, 2 and 3 were used to train the machine learning models, whereas, 160 measurements (25% of the dataset) from boards 4 and 5 were used for model testing. The training data was subjected to 10-fold, leave-one-out cross-validation to minimize the possibility of any model underfitting or overfitting. Classification models such as the kNN, SVM, DT and RF were implemented and the corresponding results are shown in Fig. 9. Using *k* as 5 with the Manhattan distance metric, the classification accuracy was found to be *96.25%*. It was observed that changing the value of *k* beyond 5 and lower than 3 reduced the classification accuracy, as a result, the tested *k* values were limited to 5,4, and 3. For both kNN and SVM, the features were normalized to attain the same scale range following which SVM rendered an accuracy of *99.38%*. Besides SVM, the RF model using 100 decision trees, a tree depth of 10 and the information gain splitting criterion rendered a good performance with *99.38%* accuracy.

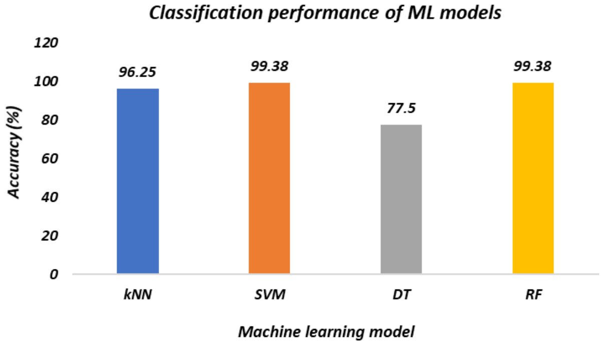

Fig. 9 Classification accuracies of machine learning models – dataset 2

For kNN, SBE improved the performance of the model from *96.25%* to *98.75%* using 38 features. On the other hand, using PCA with 15 PCs along with DT enhanced the performance of the model from *77.5%* to *90%*. Amongst the top 20 features shown in Fig. 10, drop value features accounted for *35%* and slope value features accounted for *30%*. This clearly shows that although drop value features accounted for a higher percentage, ample importance was given to the features from the other phases of the signal response as well.

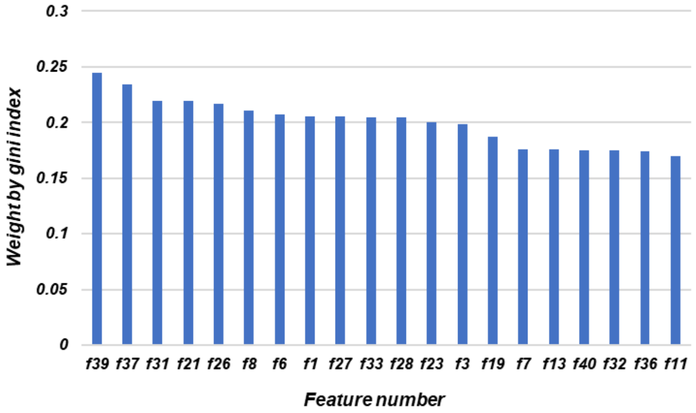

Fig. 10 Feature weights for top 20 features in the dataset - dataset 2

On implementing the machine learning models with different number of features based on their gini index weights, for all the four models, the accuracy steadily increased until the top 20 highest weighted features were used, following which it maintained a steady value (see Fig. 11). This shows that the top 20 features are capable of providing a steadily increasing accuracy and attaining a more stable value requires the use of at least 20 features.

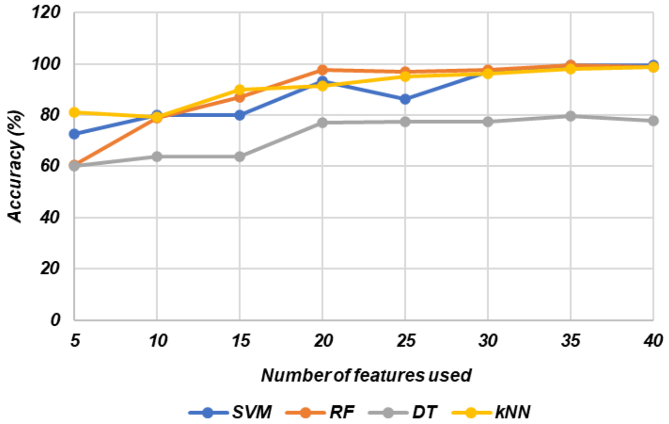

Fig. 11 Classification accuracy using different number of features – dataset 2

3.3. Dataset 3 – Experimental dataset of CO with varying concentrations

This dataset was collected through the detection of different carbon monoxide (CO) levels using fourteen MoX sensors under fluctuating humidity conditions [37]. The experiment employed seven different MoX sensors (2 sensors of each type), resulting in a total of 14 MoX sensors for detecting 10 different concentrations. Each concentration was tested 10 times, yielding a complete dataset of *1,300* measurement samples. More details regarding the experimental procedure can be found in the work presented by Burgués et al. [37]. Each sample of CO was measured for *900s*, during which the sensor heater voltage levels were fluctuated to generate high and low sensor temperatures in cycles of *25s*. As a result, the duration of each measurement being *900s*, with one cycle of *25s*, resulted in approximately 36 complete cycles during each measurement taken by Burgués et al. [37], as shown in Fig. 12.

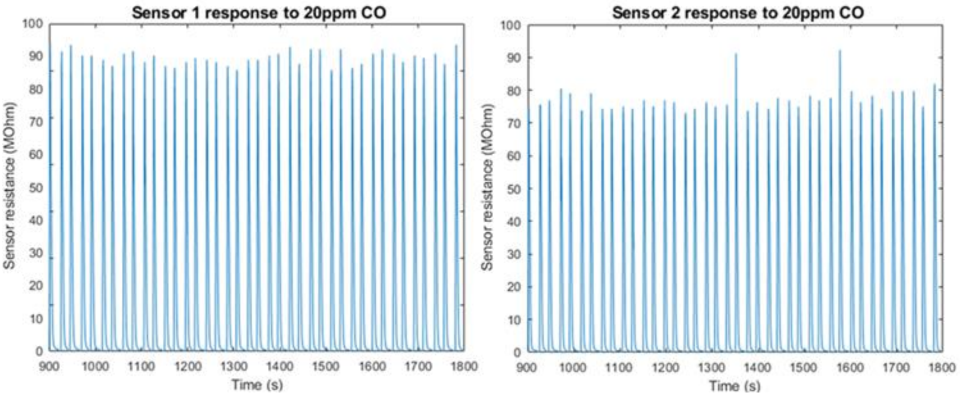

Fig. 12 Sensor response from sensor 1 and 2 to 20ppm of CO

3.3.1. Methodology

Based on the sensor responses, the three main features were extracted from $r[k + \Delta x]$ that represents the sensor resistance, with k as the discrete time-step, where $k \in [0,900]$ and Δx is used to indicate any incremental change in the time-step as shown below:

The maximum value of the signal amongst all the 36 cycles for a given gas is given as,

$$\text{maximum peak} = \underset{k=0,1,\ldots 900}{\text{abs.}\max}\ (\mathrm{r}[\mathrm{k} + \Delta \mathrm{x}]) \tag{10}$$

and the time-step at which the maximum peak occurs is defined as,

$$\text{time}_{\text{max. peak}} = \mathrm{k} + \Delta\mathrm{x} \tag{11}$$

Drop value as the signal transitions from the peak value to the lowest value in that cycle is given as,

$$\text{drop } = \text{maximum peak-} \underset{k=0,1,\ldots 900}{\text{local}\min}\ (\mathrm{r}[\mathrm{k} + \Delta \mathrm{x}]) \tag{12}$$

here, $k < time_{max.\ peak}$. On the other hand, the transition rate is determined through the slope as follows:

$$\text{minimum value} = \underset{k=0,1,\dots 900}{\text{local min}} \left(r[k + \Delta x]\right) \tag{13}$$

here, $k > time_{max.\ peak}$ and the time-step at which the minimum value occurs is defined as,

$$\text{time}_{\text{min. value}} = k + \Delta x \tag{14}$$

Consequently,

$$\boldsymbol{slope} = \frac{maximum\ peak - minimum\ value}{time_{max.peak} - time_{min.value}} \tag{15}$$

Fig. 13 illustrates the three different features extracted as shown in Eqs. (10), (12) and (15) from sensor 1 for the detection of 20ppm of CO. This was done for the responses from all 14 sensors for each of the CO concentrations, thereby, resulting in 42 features (3 main features x 14 sensors) to be used for regression. The feature and the feature number associated with each sensor is shown in Table 8.

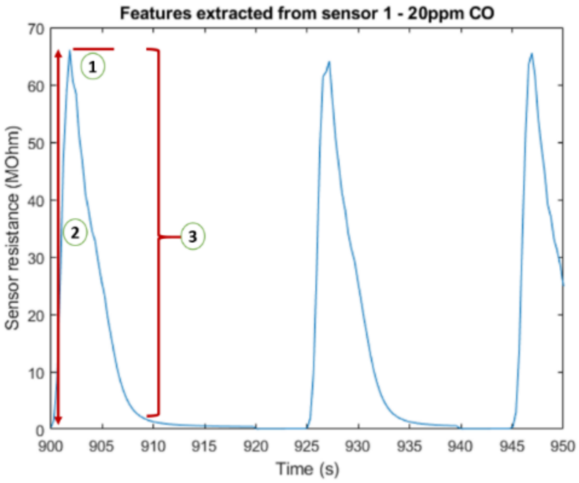

Fig. 13 Visual depiction of the three features extracted from sensor signal of sensor 1 for 20ppm of CO

Table 8 Summary of the complete feature set - dataset 3

Sensor no.	Feature name/Feature number			Sensor no.	Feature name/Feature number		
	Maximum peak value	Drop value	Slope value		Maximum peak value	Drop value	Slope value
1	f1	f2	f3	8	f22	f23	f24
2	f4	f5	f6	9	f25	f26	f27
3	f7	f8	f9	10	f28	f29	f30
4	f10	f11	f12	11	f31	f32	f33
5	f13	f14	f15	12	f34	f35	f36
6	f16	f17	f18	13	f37	f38	f39
7	f19	f20	f21	14	f40	f41	f42

3.3.2. Machine learning – Regression

The complete dataset of 42 features with *1,300* measurement samples was used for regression. The dataset was split using the 70:30 ratio, where 70% of the original dataset was reserved for training and the remaining 30% was used for testing the machine learning models. Both, classical models such as DT, kNN, SVM, ANN and ensemble machine learning models such as RF, Voting and Bagging were used to perform the regression, wherein, the performance of the models was compared using the symmetric mean absolute percentage error (sMAPE) to account for 0ppm concentrations present in the dataset. Results of regression are shown in Fig. 14, where the sMAPEs for all the models hovered around 60%. However, the lowest sMAPE was attained by RF (sMAPE of *58.82%*), followed by Bagging using RF (sMAPE of *60.28%*). Hyperparameter tuning for RF resulted in an optimal value of 48 trees, whereas, the hyperparameters for ANN were tuned to 2 hidden layers with 2 neurons each. For kNN, the optimal value of *k* was found to be 7 with Manhattan distance. The dataset contains 10 repetitions for each concentration of CO with varying levels of humidity for each repetition. As a result, the high sMAPEs can be attributed to the lack of repeatability in the sensor responses due to fluctuating humidity levels.

When determining the weights of the features using correlation, each type of feature, i.e., the drop value, the slope value and the peak value accounted for an equal portion of the top 21 features (*33.3% each*), which highlights their equal importance for regression. Furthermore, to alleviate the impact of high sMAPEs, SFS and SBE were implemented to reflect on any performance improvement that could be achieved. The results of using SFS and SBE indicated that feature selection proved to be successful in reducing the sMAPE for all the models. For some models such as Bagging using RF, the sMAPE reduced significantly from *60.28%* to *55.73%* and RF, where the sMAPE reduced to *52.71%* from *58.82%*. However, for some models such as SVM, the sMAPE remained almost the same irrespective of feature selection. In any case, the best performing models were found to be the RF, Bagging using RF, kNN and ANN models. Using SFS, the three types of features in the dataset were equally present in most of the cases, whereas, for SBE, an equal number of all the three features were dropped while training the models.

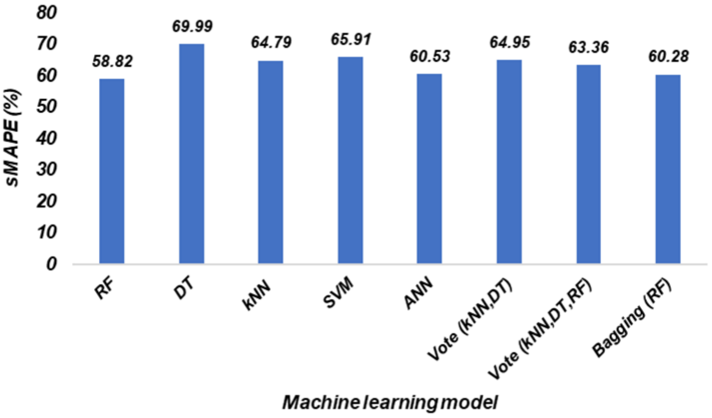

Fig. 14 Regression performance of machine learning models – dataset 3

Using PCA, 19 PCs were used along with the models to achieve a *95%* variance, where models of RF (sMAPE of *51.82%*), kNN (sMAPE of *56.81%*) and Bagging using RF (sMAPE of *54.12%*) provided the lowest sMAPEs. In other cases, the error remained the same or slightly decreased. This shows that for RF, kNN and Bagging using RF, a linear combination of the features through PCA could be used to obtain a better performance, whereas, the remaining models relied on the original feature dataset to estimate the concentrations of CO. On further observation, our results indicate the ability of the implemented models to be applied to regression tasks. They also highlight the importance of feature selection and PCA in reducing computational time to improve the model performance. Consequently, other factors that might have played a crucial role could be the sensor data itself. The inability of sensors to reproduce the same sensor response owing to external factors such as humidity can easily compromise the performance of the machine learning model for either classification, regression or both.

4. Conclusion

E-noses are used profusely for smart gas sensing applications such as medical diagnosis, environmental monitoring and food quality control. Their ability to identify different gases and estimate their concentrations makes them extremely viable for gas sensing. In this work, we provided a detailed assessment of different machine learning techniques used in E-noses for the purpose of classification and regression using three different experimental GSA datasets. The present analysis discusses concepts of signal pre-processing, feature extraction, feature selection followed by classification and regression using machine-learning techniques. Feature selection using wrapper-based techniques of SFS and SBS, and filter-based techniques accounting for gini index and correlation to determine feature importance is studied in detail to investigate the predictive capability of the developed models. In addition, the machine learning techniques employed were not limited to classical models such as DT, kNN and SVM but also incorporated ensemble techniques of Voting, Bagging and RF to offer a more comprehensive discussion.

The performance of the models was assessed using classification accuracies and regression errors, followed by a comparison of the results obtained from models presented in previously published studies. In most cases, classical models such as kNN, DT and SVM showed great capability for classification problems as demonstrated by the obtained high level of accuracies. Ensemble techniques such as Bagging (RF) and RF outperformed their classical counterparts presented in this work and published in previous works. The pivotal role of feature extraction was demonstrated by the good performance of models. Moreover, the models implemented in this work were able to offer more accurate prediction than most of the reported results obtained from the same datasets.

In terms of dimensionality reduction, PCA demonstrated improvement in the performance of the machine learning algorithms. However, Linear Discriminant Analysis, a technique widely used for feature extraction and classification, also helps minimize data dimensionality and can be used with GSA datasets. In addition, regression results revealed the impact of several factors that come into play when applying machine-learning techniques. These factors include the size of the training datasets, where it is crucial to pay heed to the size of the dataset, so as to avoid model underfitting or overfitting to reduce their impact on the models' performance. Moreover, a large number of features, non-representative features and poor quality of data for training the models can be a massive challenge for both classification and regression.

Despite the lack of publicly available GSA datasets, this work was able to identify three MoX sensor-based GSA datasets to provide a delineated assessment of different machine learning models for classification and regression. Diverse machine learning algorithms and techniques with different working principles were selected to perform classification and regression to provide a holistic discussion of machine learning in gas sensing. In all the three case studies, feature selection and dimensionality reduction were able to improve the predictive capability of machine learning models. In many cases, the approach undertaken was also capable of producing good results in the face of sensor drift. As a result, models in this work provided promising results with MoX sensor datasets and showed great predictive capability to be applied to datasets from other gas sensors such as carbon nanotube-based, polymer-based or acoustic sensors.

REFERENCES

1. Miller, D.R., Akbar, S.A., Morris, P.A., 2014, *Nanoscale metal oxide-based heterojunctions for gas sensing: A review*, Sensors and Actuators B: Chemical, 204, pp. 250-272.
2. Chen, X., Wong, C.K.Y., Yuan, C.A., Zhang, G., 2013, *Nanowire-based gas sensors*, Sensors and Actuators B: Chemical, 177, pp. 178–195.
3. Park, H.J., Kim, W., Lee, H., Lee, D., Shin, J., Jun, Y., Yun, Y., 2018, *Highly flexible, mechanically stable, and sensitive NO2 gas sensors based on reduced graphene oxide nanofibrous mesh fabric for flexible electronics*, Sensors and Actuators B: Chemical, 257, pp. 846-852.
4. Fois, M., Cox, T., Ratcliffe, N., Costello, B., 2021, *Rare earth doped metal oxide sensor for the multimodal detection of volatile organic compounds (VOCs)*, Sensors and Actuators B: Chemical, 330, 129264.
5. Kazemi, E., Zadeh, D.S., Moshiri, B., 2021, *Metal-oxide-semiconductor Sensors Modeling Using Ordered Weighted Averaging (OWA) Operators in Electronic Nose*, Measurement, 184, 109932.
6. El-Shamy, A.G., 2021, *New nano-composite based on carbon dots (CDots) decorated magnesium oxide (MgO) nano-particles (CDots@MgO) sensor for high H2S gas sensitivity performance*, Sensors and Actuators B: Chemical, 329, 129154.
7. Yoo, R., Kim, J., Song, M.J., Lee, W., Noh, J.S., 2015, *Nano-composite sensors composed of single- walled carbon nanotubes and polyaniline for the detection of a nerve agent simulant gas*, Sensors and Actuators B: Chemical, 209, pp. 444-448.
8. Matindoust, S., Farzi, G., Nejad, M.B., Shahrokhabadi, M.H., 2021, *Polymer-based gas sensors to detect meat spoilage: A review*, Reactive and Functional Polymers, 165, 104962.
9. Zhou, Z., Xu, Y., Qiao, C., Liu, L., Jia, Y., 2021, *A novel low-cost gas sensor for CO2 detection using polymer-coated fiber Bragg grating*, Sensors and Actuators B: Chemical, 332, 129482.
10. Jakubik, W.P., 2011, *Surface acoustic wave-based gas sensors*, Thin Solid Films, 520, pp. 986-993.
11. Nitzsche, L., Goldschmidt, J., Lambrecht, A., Wöllenstein, J., 2021, *Two-component gas sensing with MIR dual comb spectroscopy*, tm - Technisches Messen, 89, pp. 50-59.
12. Blanco-Novoa, O., Fernández-Caramés, T.M., Fraga-Lamas, P., Castedo, L., 2018, *A cost-effective IoT system for monitoring indoor radon gas concentration*, Sensors (Switzerland), 18, 2198.
13. Chen, J., Gu, J., Zhang, R., Mao, Y., Tian, S., 2019, *Freshness evaluation of three kinds of meats based on the electronic nose*, Sensors (Switzerland), 19, 605.
14. Ashari, I.A., Widodo, A.P., Suryono, S., 2019, *The Monitoring System for Ammonia Gas (NH3) Hazard Detection in the Livestock Environment uses Inverse Distance Weight Method*, 2019 Fourth International Conference on Informatics and Computing (ICIC), pp. 1-6.
15. Kao, K.A., Cheng, C., Gwo, S., Yeh, J.A., 2015, *A Semiconductor Gas System of Healthcare for Liver Disease Detection Using Ultrathin InN-Based Sensor*, ECS Transactions, 66, pp. 151-157.
16. Chen, Z., Chen, Z., Song, Z., Ye, W., Fan, Z., 2019, *Smart gas sensor arrays powered by artificial intelligence*, Journal of Semiconductors, 40, 111601.
17. Hunter, G.W., Akbar, S., Bhansali, S., Daniele, M., Erb, P.D., Johnson, K., Liu, C., Miller, D., Oralkan, O., Hesketh, P.J., 2020, *Editors' Choice—Critical Review—A Critical Review of Solid State Gas Sensors*, Journal of The Electrochemical Society, 167, 037570.

18. Fonollosa, J., Fernández, L., Gutiérrez-Gálvez, A., Huerta, R., Marco, S., 2016, *Calibration transfer and drift counteraction in chemical sensor arrays using Direct Standardization*, Sensors and Actuators B: Chemical, 236, pp. 1044-1053.
19. Vergara, A., Vembu, S., Ayhan, T., Ryan, M.A., Homer, M.L., Huerta, R., 2012, *Chemical gas sensor drift compensation using classifier ensembles*, Sensors and Actuators B: Chemical, 166-167, pp. 320-329.
20. Deng, C., Lv, K., Shi, D., Yang, B., Yu, S., He, Z., Yan, J., 2018, *Enhancing the discrimination ability of a gas sensor array based on a novel feature selection and fusion framework*, Sensors (Switzerland), 18, 1909.
21. Hira, Z.M., Gillies, D.F., 2015, *A review of feature selection and feature extraction methods applied on microarray data*, Advances in Bioinformatics, 2015, 198363.
22. Geng, A., Moghiseh, A., Redenbach, C., Schladitz, K., 2021, *Comparing optimization methods for deep learning in image processing applications*, tm - Technisches Messen, 88, pp. 443-453.
23. Hoffmann, L., Fortmeier, I., Elster, C., 2021, *Deep learning for tilted-wave interferometry,* tm - Technisches Messen, 89, pp. 33-42.
24. Adhikari, S., Saha, S., 2014, *Multiple classifier combination technique for sensor drift compensation using ANN & KNN*, 2014 IEEE International Advance Computing Conference (IACC), pp. 1184-1189.
25. Rehman, A.U., Bermak, A., 2019, *Heuristic random forests (HRF) for drift compensation in electronic nose applications*, IEEE Sensors Journal, 19, pp. 1443–1453.
26. Ma, D., Gao, J., Zhang, Z., Zhao, H., 2021, *Gas recognition method based on the deep learning model of sensor array response map*, Sensors and Actuators B: Chemical, 330, 129349.
27. Fu, X., Wang, L., 2003, *Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance*, IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics, 33, pp. 399–409.
28. Xue, X., Zhang, M., Browne, W.N., Yao, X., 2016, *A Survey on Evolutionary Computation Approaches to Feature Selection*, IEEE Transactions on Evolutionary Computation, 20, pp. 606–626.
29. Sánchez-Maroño, N., Alonso-Betanzos, A., Tombilla-Sanrománm M., 2007, *Filter methods for feature selection - A comparative study*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4881, pp. 178–187.
30. Borowik, P., Adamowicz, L., Tarakowski, R., Siwek, K., Grzywacz, T., 2020, *Odor detection using an e-nose with a reduced sensor array*, Sensors (Switzerland), 20, 3542.
31. Vito, S.D., Massera, E., Piga, M., Martinotto, L., Francia, G.D., 2008, *On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario,* Sensors and Actuators B: Chemical, 129, pp. 750-757.
32. Pashami, S., Lilienthal, A.J., Schaffernicht, E., Trincavelli, M., 2013, *TREFEX: Trend estimation and change detection in the response of MOX gas sensors*, Sensors (Switzerland), 13, pp. 7323-7344.
33. Zhang, S., Xie, C., Hu, M., Li, H., Bai, Z., Zeng, D., 2008, *An entire feature extraction method of metal oxide gas sensors*, Sensors and Actuators B: Chemical, 132, pp. 81–89.
34. Fonollosa, J., Rodríguez-Luján, I., Huerta, R., 2015, *Chemical gas sensor array dataset,* Data in Brief, 3, pp. 85–89.
35. Destro, R., Matakas, L., Komatsu, W., Ama, N.R.N., 2013, *Implementation aspects of adaptive window moving average filter applied to PLLs - Comparative study,* 2013 Brazilian Power Electronics Conference, COBEP 2013 – Proceedings, pp. 730–736.
36. Zhao, Y., He, X., Pecht, M.G., Zhang, J., Zhou, D., 2020, *Detection and detectability of intermittent faults based on moving average T2 control charts with multiple window lengths*, Journal of Process Control, 92, pp. 296–309.
37. Burgués, J., Jiménez-Soto, J.M., Marco, S., 2018, *Estimation of the limit of detection in semiconductor gas sensors through linearized calibration models*, Analytica Chimica Acta, 1013, pp. 13–25.