



HAL
open science

A review of algorithms for SAW sensors e-nose based volatile compound identification

Olivier Hotel, Jean-Philippe Poli, Christine Mer-Calfati, Emmanuel Scorsone,
Samuel Saada

► **To cite this version:**

Olivier Hotel, Jean-Philippe Poli, Christine Mer-Calfati, Emmanuel Scorsone, Samuel Saada. A review of algorithms for SAW sensors e-nose based volatile compound identification. *Sensors and Actuators B: Chemical*, 2018, 255, Part 3, pp.2472 - 2482. 10.1016/j.snb.2017.09.040 . hal-01879582

HAL Id: hal-01879582

<https://hal.science/hal-01879582>

Submitted on 5 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A review of algorithms for SAW sensors e-nose based volatile compound identification

Olivier Hotel, Jean-Philippe Poli, Christine Mer-Calfati,
Emmanuel Scorsone, Samuel Saada
CEA, LIST, 91191 Gif-sur-Yvette, France.
{firstname.lastname}@cea.fr

April 5, 2022

Abstract

Recent advances in odour sensors have led to the development of new applications; among them, electronic noses have gained major interest and found successful applications in many fields. An electronic nose is a device composed of an array of odour sensors with sensitivity to a wide range of chemical compounds. Reliable electronic nose systems rely on advanced data processing techniques. Among them, machine learning has become a core technique for electronic nose design. In this document, we describe several machine learning algorithms and compare their performances on different features used in state of the art electronic nose systems.

Keywords: odour recognition, electronic nose, SAW sensors, data processing, machine learning.

1 Introduction

For many years, volatile compounds detection was achieved via classical olfactometry: the identification was based on a sensory panel composed of selected peoples or based on animal olfaction. However, physiological differences and different smelling capabilities can lead to inaccurate and non-reproducible results. To overcome these issues, alternative methods were developed: gas chromatography, mass spectrometry, optical spectroscopy and chemical sensors [49]. In this article, we focus on chemical sensors only. Basically, these sensors have the same functioning principle. It is based on the physico-chemical interaction between a volatile compound and a sensitive coating material covering a transducer. The molecules interact with the sensitive surface resulting in a change of its physical properties. These changes are converted to a measurable signal by the transducer. Many volatile compounds sensors technologies have been developed: metal oxide semiconductor sensors, metal oxide semiconductor field-effect transistor sensors, conducting polymer sensors, biosensors and acoustic sensors [28].

In this document, we focus on a category of acoustic sensors called Surface Acoustic Waves (SAW) sensors. These sensors are based on the propagation of mechanical waves (100 MHz – 1000 MHz) produced by piezoelectric materials along a layer composed of a substrate covered by chemically interactive materials. Volatile compounds are absorbed onto the surface of the sensitive material, changing its properties and yielding to a measurable frequency shift of the mechanical waves [23]. SAW sensors have many advantages over the previously mentioned technologies: they have a high sensitivity, short response time and low power consumption. However, SAW sensors are very sensitive to the temperature and humidity and have a poor reproducibility in the deposition of the chemically interactive materials [28].

Odour sensors based systems have obtained successful applications in many fields, for instance, for quality control applications in the food [52, 5], for detection of disease specific odours [72], for the detection of pollutants and toxic gases [75]. In many applications, gas sensor arrays composed of several sensors belonging to different technology families or with different chemically interactive materials are used instead of a single gas sensor to increase the selectivity of the entire system and its ability to identify volatile compounds. Such arrays are used in conjunction with a multivariate data processing system capable of recognizing odours to form what *Gardner* defines as an electronic nose [29]. Nowadays, reliable electronic nose systems rely on advanced data processing techniques. Machine learning has become a core technique for electronic nose design [77].

Some authors have already reviewed the main pattern analysis algorithms for machine olfaction [36, 64]. In this document, we extend their works and we focus on the application of machine learning for processing the data generated by an electronic nose composed of SAW sensors. First, we give an overview of the existing signal pretreatments and feature extraction methods for SAW sensors based electronic nose. Then we describe the most commonly used dimensionality reduction and pattern recognition algorithms. Finally, we compare the performances of these algorithms on a data set composed of seven gases and on a real world problem consisting in identifying counterfeit coffee capsules.

2 Features extraction and normalization

As mentioned in the introduction, the response of a SAW sensor to a volatile compound is a frequency shift. This shift can be measured yielding to a one dimensional time series $\Delta f(t) = f(t) - f_0$ where $f(t)$ is the oscillation frequency and f_0 is the nominal oscillation frequency of the sensor in absence of volatile compounds. Subtracting the value of the baseline $\Delta f_2(t) = \Delta f(t) - \Delta f(t_i)$ prior to sample delivery is a common data pre-processing technique for drift compensation.

2.1 Data extraction

Each sensor produces a time series in response to a chemical compound. The response generated by an electronic nose composed of c sensors is a set of c time series. It can be regarded as a real c -by- n matrix, where n is the number of samples in the time series. Typically, the response of a SAW sensor can be divided into two parts: first the quantity Δf is varying (transient regime) and then the signal becomes almost constant (steady state regime). It is common to use the steady state values Δf as the input data of the pattern recognition system. The authors of [51] suggest that the transient response may contain relevant information. They propose to extract from each signal two features: the amplitude during the steady state and the rise time. The rise time is measured from 20% to 60% of the steady state value. The authors of [17] introduced the analysis of the sensor's phase space and proposed to use the area of the response

$$A = \int f(t)dt \text{ and the area of the phase space } A = \int \frac{df}{dt} df$$

as descriptors. Figure 1 shows a typical response of an electronic nose composed of 4 sensors with the rise times T_i and the steady state values A_i .

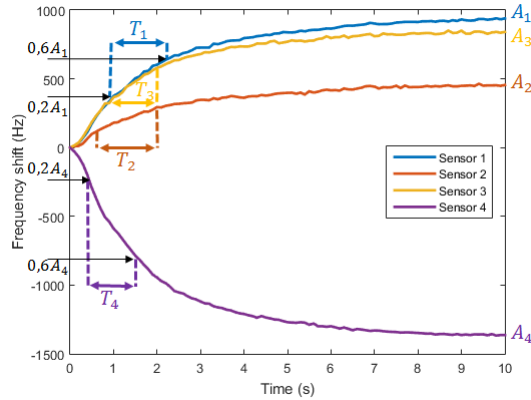


Figure 1: Typical response of an electronic nose based on 4 sensors

The authors of [54] propose three different parametric models R of sensor's response:

1. the exponential model : $R(t) = \beta\tau(1 - e^{-\frac{t}{\tau}})$;
2. the Lorentzian model : $R(t) = \beta\tau \tan^{-1}\left(\frac{t}{\tau}\right)$; and
3. the double sigmoid model :

$$R(t) = \frac{\alpha}{\pi} \left(1 - e^{-\left(\frac{t-\beta}{\gamma} + \epsilon\right)^\delta} \right)^\eta \left(\frac{\pi}{2} - \tan^{-1}\left(\frac{t-\mu}{\nu}\right) \right)^\lambda.$$

They estimate their parameters using the Levenberg Marquardt algorithm and use them as the input data of the pattern recognition system. Some authors suggest to use the set of c time series or the c -by- n matrix as the input data of the pattern recognition process. These features are interesting because on the one hand, they carry information about the transient response [55] and on the other hand, because robust features can be extracted through digital signal processing methods. Naidoo et al. [56] propose to use the coefficient of a discrete cosine transform (DCT) with maximum variance of the signal as the input data of the pattern recognition system. The DCT is a method which allows to express a signal as a sum cosine functions oscillating at different frequencies. It is a slightly modified version of the Fourier transform using only real numbers. More formally, the DCT of a finite length series x is given by

$$X_k = \sum_{n=0}^N x_n \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right).$$

Singh et al. [67] suggest to use the approximation coefficient of the discrete wavelet transform (DWT). The DWT use a scalable modulated window called a wavelet to solve the signal cutting problem of the short time FFT. The wavelet is shifted along the signal and the spectrum is computed. Then, this process is repeated several times with a shorter window. At the end of the process we obtain a set of time-frequency representation of the signal. More formally, the DWT is expressed by the following formula :

$$F(a, b) = \int_{-\infty}^{+\infty} f(x) \Psi(a, b)^* dx.$$

The child wavelets $\Psi(a, b)$ are generated from a raw wavelet Ψ through dilatation (coefficient a) and translation (coefficient b)

$$\Psi_{a,b} = \frac{1}{\sqrt{a}} \Psi\left(\frac{x-b}{a}\right).$$

Classical examples for Ψ are: the Haar wavelets and the Daubechies wavelets. The interested readers can refer to the paper of Grasp [35] for more information about wavelets.

2.2 Data normalization

Data normalization is the final step for data pre-processing. The primary idea that motivates data normalization is to increase the performances of the learning process and so the performances of the entire machine learning system by making the variables comparable [6]. Two steps are required to achieve data normalization: moving the entire data set so that it is centred around the origin and rescaling each feature so that each one of them has a variance equal to 1 or each feature has maximum absolute value equal to 1. The aim of centring is to verify that any features are arbitrarily large and the aim of scaling is to make

sure that all features have approximately the same scale. There exist different methods to perform data normalization:

- centring $X_{n,d} \leftarrow X_{n,d} - \mu(X_{:,d})$;
- variance scaling $X_{n,d} \leftarrow \frac{X_{n,d}}{\sigma(X_{:,d})}$;
- z-score $X_{n,d} \leftarrow \frac{X_{n,d} - \mu(X_{:,d})}{\sigma(X_{:,d})}$;
- absolute scaling $X_{n,d} \leftarrow \frac{X_{n,d}}{\max(|X_{:,d}|)}$; and
- 0-1 scaling $X_{n,d} \leftarrow \frac{X_{n,d} - \min(X_{:,d})}{\max(X_{:,d}) - \min(X_{:,d})}$

where $X_{n,d}$ refers to the d^{th} feature of the n^{th} sample, $\mu(X_{:,d})$ the mean of the d^{th} feature and $\sigma(X_{:,d})$ the variance of the d^{th} feature.

3 Change of representation and dimension reduction

Many machine learning algorithms, in particular the non-parametric and the kernel-based algorithms, tend to be dependent on their data representation. In this section, we describe two methods allowing to extract uncorrelated (PCA) or independent (ICA) factors.

3.1 Principal Component Analysis

The Principal Component Analysis (PCA) technique [46], also known as the Karhunen-Loève transform, is able to remove co-linearity in variables. PCA is a linear transformation that maps the original data to a new coordinate system where the greatest variance of the variables lies on the first coordinate, the second on the second coordinate and so on. PCA is one of the most used techniques for feature extraction and dimension reduction [70, 45, 7, 19, 65].

3.2 Independent Component Analysis

Independent Component Analysis (ICA) is a statistical method used to identify hidden factors of random variables. It is a linear generative model which assumes the observed variables are a linear mixture of unknown non Gaussian and mutually independent variables. The aim of ICA is to find those variables without making any assumptions about the mixing system. More formally, if the data are represented by the vector $x = (x_1, \dots, x_n)$ and the independent component by the vector $s = (s_1, \dots, s_n)$, the aim of ICA is to find a linear transformation \mathbf{W} verifying $s = \mathbf{W}x$ and minimizing a function F measuring

the statistical independence. More formally, it consists in solving the optimization problem

$$\begin{aligned} & \underset{s}{\text{minimize}} && F(s) \\ & \text{subject to} && s = \mathbf{W}x. \end{aligned}$$

A common example for F is the mutual information [42]. ICA is a very active field of research with many applications in different fields. Hyvarinen et al. [43] cover the entire topic, in particular, they deal with the formal definition of ICA and propose a survey of the algorithms used to determine the matrix \mathbf{W} . ICA has proved its utility for electronic nose systems design [76, 1, 4, 57].

3.3 Application to dimension reduction

PCA is one of the main dimensionality reduction technique. Such task is traditionally performed by keeping the first n components explaining $k\%$ of the variance. ICA was not designed to perform dimension reduction. In order to find n independent components, one should first reduce the dimension to n components using PCA and then apply ICA on the result of PCA as explained in [26].

4 Pattern recognition algorithms

Machine learning is a sub-field of computer science whose goal consists in automatically building a model of data without being explicitly programmed for. Among the problems addressed by machine learning, supervised learning is of major interest for the e-nose community. The objective is to find a function f reproducing a variable Y having observed a variable X :

$$Y = f(X) + \epsilon$$

where ϵ represents noise or measurement error [10, 2]. In this document, we will focus on supervised learning for which we have a training set consisting of n labelled observations also called labelled samples or labelled examples

$$\{(x_1, y_1) \dots (x_i, y_i) \dots (x_n, y_n)\}$$

with $x_i \in X$ and $y_i \in Y$. The objective is to build, from these learning samples, a model which allows to predict the output y associated with a new sample x . Most of the time, the model is built by choosing a parametrized function and by determining the parameters minimizing an error criterion on the training set. When $Y = \{-1, 1\}$, the supervised learning problem is commonly called binary discrimination. In the following sections, we describe several models that lead to different form of the function f .

4.1 Linear classifiers

4.1.1 Linear Discrimination Analysis

Linear Discrimination Analysis (LDA) is a binary linear classifier whose goal consists in finding the best separating hyperplane between two groups of samples. It can be expressed as: $f(x) = w^T x + w_0$. One approach to find the parameters was proposed by *Fisher* [25] who showed that they can be determined by solving the optimization problem:

$$w = \operatorname{argmin} \frac{w^t C_b w}{w^t C_w w},$$

where C_b is the covariance matrix between the two sample groups whereas C_w is the covariance matrix within the two sample groups. Although LDA was designed to tackle binary discrimination problem, the method was extended to multi-class problems in [62]. LDA was successfully applied to identify Italian wines [15].

4.1.2 Partial Least Squares Discriminant Analysis

As LDA, Partial Least Squares Discriminant Analysis (PLS-DA) is a linear binary classifier which aims to find a separating hyperplane. It is a variant of the PLS regression which builds a model between the descriptors X and the labels Y . The PLS-DA equations are

$$\begin{cases} X &= Tp + e \\ Y &= Tq + f \end{cases},$$

where T , p and q are parameters, e and f can be considered as residuals. The parameters are estimated using the widely used PLS1 algorithm [13] which has also been adapted to multi-class problems [41].

4.2 Neural networks

4.2.1 Feedforward neural networks

Since their introduction in the mid 60's, a huge amount of research has been conducted, yielding to several models. Basically, neural networks are made up of many artificial neurons. An artificial neuron is a mathematical function that models the behaviour of a biological neuron [9].

For a given neuron with n inputs x_i , the output of the neuron is given by

$$o_i = \phi \left(\sum_{i=1}^n w_i x_i + b \right)$$

where w_i are the weights of the inputs, b is a bias and ϕ is the transfer function. Classical examples for ϕ are :

- a linear function : $L(x) = x$;
- a sigmoid function : $S(x) = \frac{1}{1+e^{-x}}$; and
- a Gaussian radial function : $G(x) = e^{-\frac{\|c-x\|^2}{2\sigma^2}}$.

Artificial neurons are organized into bigger structures to perform pattern recognition. The most common way of linking the artificial neurons is by organizing them into a structure called a feedforward network : each neuron in each layer feeds its output forward to the input of the next layer. Many models of feedforward neural network have been developed [9]; among them, Multi-Layers Perceptron (MLP), Radial Basis Function network (RBF) and Probabilistic Neural Network (PNN) to name a few. Neural networks have found many applications in many different fields. In particular they have been successfully applied in food industry [68, 44, 58, 71, 31, 50] and in chemistry [16, 22].

4.2.2 Learning Vector Quantization

Learning Vector Quantization (LVQ) [48] is a supervised learning algorithm based on a Kohonen self-organizing map (SOM) [47]. SOM is a neural network formed by arranging a set of neurons in a two-dimensional grid, discretizing the representation of the input space of the training samples. A weight vector of the same dimension as the input data vectors is associated with each neuron. The learning process is based on a competitive algorithm: locating the neuron whose weight vector is the closest to the output and then updating the weight of this neuron and of its neighbourhood. LVQ is a slightly modified version of SOM. All the variants of LVQ algorithm share the same scheme. It consists of a competitive layer containing a number of competitive neurons. Each neuron is associated with a class and with a weight vector representing the centroid of the class. The learning process is slightly modified: the weight vector of the winning neuron is updated only if it is labelled with the same label as the training example. A new example is assigned to the class of its closest neuron. The number of neurons is determined experimentally starting with a small structure and then adding extra neurons iteratively or starting with a huge structure and then removing neurons. The learning process is stopped when one of the criteria described in the previous section is verified. LVQ has found many applications, especially in the food and agricultural industry to detect diseases in tomato crops [31] and for qualitative classification of fruit juices [60].

4.2.3 Spiking Neural Networks

Spiking Neural Networks (SNN) is a recently proposed, biologically inspired, novel model of neurones incorporating the notion of time and increasing the level of realism. Such networks have become a prolific field of research and many models have been developed. In particular, Dielh *et al.* [24] propose a classification algorithm using this new model of neurones whereas [73] provides an overview of the current trends concerning SNN.

4.3 Neighbourhood approaches

4.3.1 K Nearest Neighbours and Large Margin Nearest Neighbours

The k -Nearest Neighbours (KNN) is a non-parametric method for classification. In order to classify a new sample, the k nearest neighbours of the new sample are found, then the new sample is assigned to the most common class among its k nearest neighbours. More formally, for a given metric $d(-, -)$ and for a sample x the distances $d(x, X_i)$ are computed. Then, if the order statistics are denoted by

$$0 \leq d_{(1)}(x) \leq d_{(2)}(x) \leq \dots \leq d_{(k)}(x) \leq \dots \leq d_{(n)}(x)$$

the k nearest neighbours set is

$$A_k(x) = \{X_i : d(x, X_i) < d_{(k)}(x)\}$$

and the sample x is assigned to the majority class among the k nearest neighbours set. Fig. 2 shows the k nearest neighbours set for two different values of k .

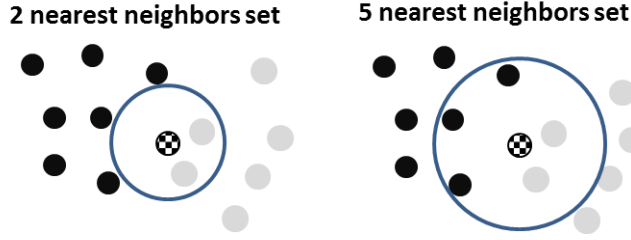


Figure 2: K nearest neighbours

The metric used to find the nearest neighbours is a critical parameter. To overcome this issue, the authors of [74] propose to compute the Euclidean distances after performing a linear transformation:

$$D(x, y) = \|\mathbf{L}(x - y)\|_2.$$

As shown in Fig. 3, the isovalue lines of the proposed metric are ellipses. They propose to determine \mathbf{L} by solving the following convex optimisation problem using specially designed gradient descent algorithm.

$$\begin{aligned} \mathbf{L} = \operatorname{argmin} & (1 - \mu) \sum_{i \rightarrow j} \|\mathbf{L}(x_i - x_j)\|_2^2 \\ & + \mu \sum_{i, j \rightarrow i} \sum_l (1 - y_{i,l})(1 + \|\mathbf{L}(x_i - x_j)\|_2^2 + \|\mathbf{L}(x_i - x_l)\|_2^2)_+ \end{aligned}$$

where, $y_{i,l} = 1$ iff $y_i = y_l$, and $y_{i,l} = 0$ otherwise, $(x)_+ = \max(0, x)$. The notation $i \rightarrow j$ indicates that x_i is a target neighbour of x_j . The target neighbours of x_i are those that we desire to be closer to x_i . This objective function is composed of two terms:

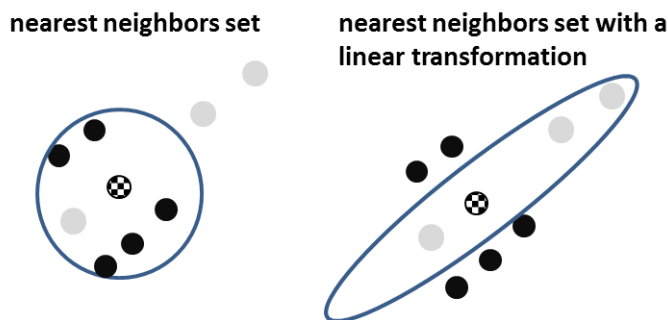


Figure 3: Large margin nearest neighbours

- the first term penalizes large distances between each sample of the same label; and
- the second term penalizes small distances between differently labelled samples.

The coefficient $\mu \in [0; 1]$ is a parameter that balances the two terms of the objective function; when $\mu = 0$, only the large distances between two samples of the same label whereas are penalized, when $\mu = 1$ only the small distances between differently labelled samples are penalized. The choice of k depends on the data. A larger value of k may reduce the effect of noise and outliers, while a small one may yield to more complex decision boundaries. The determination of k can be done using a grid search method or various heuristics. Another variant of k -nearest neighbours is weighted k nearest neighbours [39]. The main idea behind this variant is to take into account the distance from the query sample and its nearest neighbours. The class label of the query is predicted by summing the value of a decreasing function applied to the distance for each class, then the query sample is assigned to the class for which the sum is maximum. Since KNN requires to access the entire database, its computational cost may be important. When the dimension of the samples is small (≤ 15) it can be computationally effective to use the Bentley's k-d tree structure [8] to quickly retrieve the k nearest neighbours. If the class distribution is skewed, samples from the most frequent class tend to dominate the prediction of a new sample since they are more common in the nearest neighbour set. It is possible to overcome this issue by using the weighted variant of KNN. In spite of this defect, the capability of KNN to learn very complex decision boundaries makes it a very powerful tool. KNN has obtained successful results in a lot of fields: in the food and agricultural industry to identify fruits [70], in the medical field to monitor the bacterial growth [66], and in chemistry to determine the concentration of compounds [34].

4.3.2 Similarity measure

Instead of using distance function to identify the neighbourhood of a given sample, one can use a similarity measure. In this variant, the neighbourhood is the set of the k most similar samples. A simple example of similarity measure is the dot product. Indeed, the dot product of two unit length vectors x and y is given by $x^T \cdot y = 1 - \epsilon$, $0 \leq \epsilon \leq 1$. When these vectors are increasingly similar ϵ tends to 0. Thus the dot product matching algorithm consists in assigning a new example to the class of the training sample maximizing their dot product.

4.3.3 RMSE neighbourhood

Another variant, of the neighbourhood based algorithm consists in building a model for each elements in the database and associating an unknown sample to the model yielding to the lowest mean squared error. Many recursive models have been proposed in the literature [18]:

- The auto-regressive (AR) model : a p -order AR model can be expressed as $X_t = \phi_0 + \sum_{i=1}^p \phi_i X_{t-i} + \epsilon_t$ where ϕ_0, \dots, ϕ_p are parameters and ϵ_t is white noise.
- The moving-average (MA) model : a q -order MA model can be expressed as $X_t = \mu + \epsilon_t + \sum_{i=1}^q \theta_i \epsilon_{t-i}$ where $\theta_0, \dots, \theta_q$ are parameters, μ is the expectation of X_t and ϵ_t is with noise error terms.
- The auto-regressive-moving-average (ARMA) model is a combination of the AR and MA models : $X_t = \phi_0 + \sum_{i=1}^p \phi_i X_{t-i} + \epsilon_t + \sum_{i=1}^q \theta_i \epsilon_{t-i}$.

The parameters of those models are typically tuned using a least square approach or by likelihood maximization [11, 37]. The selection of the hyper-parameters p and q is a major issue. Many approaches have been investigated:

- Statistical test: this approach consists in starting with over-estimated values of p and q and performing an hypothesis test on the final lag parameters [53]. For instance, one might start by estimating $AR(N)$ and test whether the coefficient ϕ_N is significant or not and then reduce the value of N . F-statistic and t-statistic are often used to assess the level of significance of a coefficient.
- Information criteria based approach: another way is to select the parameters minimizing an information criterion [69]. Typical information criterion are:
 - the Bayes information criterion (BIC):

$$BIC_{p,q} = -2\log(\mathcal{L}) + k\log(n);$$

– and the Akaike information criterion (AIC) :

$$AIC_{p,q} = -2\log(\mathcal{L}) + 2k$$

where \mathcal{L} is the likelihood of the model, n is the number of observations and k is the total number of parameters. It can be noticed that since $\log(n) > 2$ for large n , the BIC tends to select more parsimonious models than AIC.

- Grid search may also be considered as suggested in [18].

Another way to model a signal consists in using a priori knowledge to model the signal in the temporal domain. For instance, Mayoue et al. [55] propose to model the sensors response with a first order system plus a linear drift

$$f(t) = K(1 - e^{-\frac{t}{\tau}}) + \alpha t + \beta.$$

Then the parameters are fitted using a recursive least squares algorithm and a new signal is assigned to the class which minimizes the gap between the model and the real data. This method was successfully used by Mayoue et al. [55] to detect and recognize explosive compounds (TNT and EGDN) in presence of interferences.

4.4 Support Vector Machine

In a binary classification task, the aim of the SVM [38] is to find a hyperplane separating the labelled data. The support vectors are the point, among the entire data set, which are the closest to the separating hyperplane. In many situations, when the data are linearly separable, there exist an infinity of separating hyperplanes. The principle of SVM consists in choosing the hyperplane maximizing the distance between the support vectors and the hyperplane, this distance is commonly called the margin (see Figure 4). Let us assume that the data are linearly separable. In this situation, there exists a hyperplane H of equation $\omega x + \omega_0 = 0$ separating binary labelled data x_i with label $y_i \in \{-1, 1\}$ and verifying

$$\begin{cases} \omega x_i + \omega_0 \geq 1 & \text{if } y_i = 1 \\ \omega x_i + \omega_0 \leq -1 & \text{if } y_i = -1 \end{cases}$$

or, by combining these two equations, verifying $y_i(\omega x_i + \omega_0) \geq 1$. The distance between this hyperplane and the hyperplane H_1 of equation

$$H_1 : \omega x + \omega_0 = 1 \text{ is } \frac{|1 - \omega_0|}{\|\omega\|}$$

and the distance with the hyperplane H_2 of equation

$$H_2 : \omega x + \omega_0 = -1 \text{ is } \frac{|1 + \omega_0|}{\|\omega\|}$$

The margin is the distance between the hyperplanes H_1 and H_2 , and is equal to $\frac{2}{\|\omega\|}$. Maximizing this quantity subject to the constraint $y_i(\omega x_i + \omega_0) \geq 1$ is equivalent to solve the following optimization problem

$$\begin{aligned} & \underset{\omega}{\text{minimize}} && \frac{1}{2} \|\omega\| \\ & \text{subject to} && y_i(\omega x_i + \omega_0) \geq 1. \end{aligned}$$

Typically, this is done by using the method of Lagrange's multipliers.

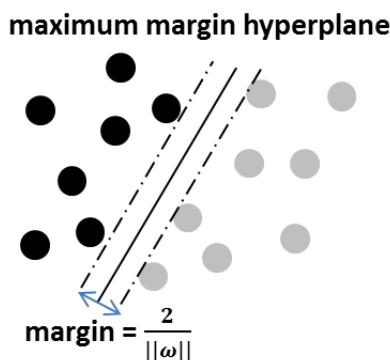


Figure 4: Support vector machine

However, most of the time, the data are not linearly separable. In this case the separating hyperplane does not exist. In this situation, new variables are artificially introduced in the optimization problem :

$$\begin{aligned} & \underset{\omega}{\text{minimize}} && \frac{1}{2} \|\omega\| + C \sum \epsilon_i \\ & \text{subject to} && y_i(\omega x_i + \omega_0) \geq 1 - \epsilon_i \end{aligned}$$

where $C \neq 0$ is a positive constant. Moreover, the data are projected in a higher dimension space thanks to a transformation ϕ . The main idea that motivates this transformation is that in a higher dimension the data will be more linearly separable and thus the error ϵ will be smaller. The objective is to determine the optimal separating hyperplane in the higher dimension space, in which the separating hyperplane can be expressed with the terms $\phi(x_i)^T \phi(x_j)$. Those ones can be replaced by a function $K(x_i, x_j)$ called a kernel. Classical kernel functions are:

- the linear kernel $K(x, y) = x \cdot y$;
- the polynomial kernel $K(x, y) = (x \cdot y)^d$ or $K(x, y) = (x \cdot y + 1)^d$; and
- the Gaussian kernel $K(x, y) = e^{-\gamma \|x-y\|^2}$, $\gamma > 0$.

The choice of the kernel function is a critical parameter that impacts the performance of a SVM classifier. The most common way to find the most appropriate kernel function consists in using a validation set to evaluate the performance of each choice.

Originally, SVM were designed to tackle binary classification tasks but they can be extended to multi-classes problems. There are two classical approaches [40]

- *One versus all*: it is the most intuitive methods to deal with multi-class problems. It consists in building a SVM per class, with the samples of that class labelled as positive samples and the samples of the other classes labelled as negative samples. This strategy requires the SVM to produce confidence score for its decision. A new sample is assigned to the class of the SVM outputting the highest confidence score.
- *One vs one*: this method consists in training $\frac{N(N-1)}{2}$ binary SVM where N is the number of classes; each receiving the samples of a pair of classes. A new sample is assigned to the most represented class among the ones outputted by the SVM. The main drawback of this method is its complexity since they need a quadratic growing number of classifiers.

SVM have successfully been applied in many fields, they were applied in the food industry to classify coffee blends and milk [59, 14] and in chemistry to discriminate different gases [30].

4.5 Decision trees and bagged trees

Decision Tree (DT) is a non-parametric supervised learning method used for classification. The aim is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data. DT have two kinds of nodes. A decision node is a point where a choice must be made. The branches extending from a decision node are decision branches, each branch representing one of the possible exclusive alternatives at that point. A terminal node represents the final result of a combination of decisions.

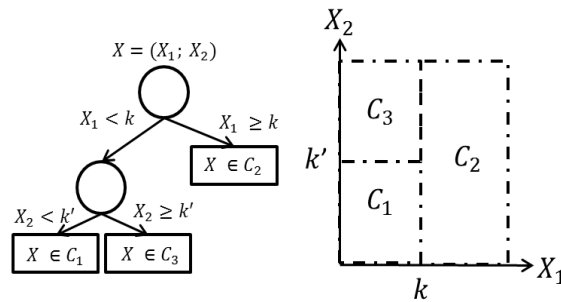


Figure 5: Decision tree and its decision boundaries

The problem of learning an optimal decision tree with a minimum of decision node is known to be NP-complete. Consequently, decision-tree learning algorithms are based on heuristic algorithms that have no guarantee to return the optimal decision tree. A common algorithm for building DT is ID3 [61] which employs a top-down greedy search strategy through all the possible branches. It divides the data set into smaller subsets while the tree is incrementally developed. The heuristic used to divide the data set proposed by the author is to choose the feature maximizing the information gain. First, entropy of every attribute is computed

$$H = - \sum p_i \log(p_i) , p_i = \frac{\text{number of elements in set } C_i}{\text{number of elements}}$$

then the data set is split into subsets using the attribute for which entropy is minimum. DT is a very popular machine learning algorithm because it is invariant under many transformations of the feature values and in particular it is invariant under scaling of the features. Moreover, it is robust to irrelevant features and, unlike the previously described algorithms, the output of a decision tree classifier can be interpreted since it consists of a set of rules. However, since decision trees divide the feature space into axis-parallel rectangles and label each rectangle with one of the classes (see Figure 5), they often are weaker than other algorithms. In addition, deep decision trees tend to learn highly irregular decision surface yielding to overfitting. To overcome this issue, Breiman *et al.* proposed the bagged tree algorithm (BT) [12]. The training set is divided into k subsets of n samples by selecting random training examples with replacement, then a classical decision tree is trained on each of the k subsets. After training, new samples are associated to the result of a majority vote of each DT. k and n are hyper-parameters which are commonly tuned using a 2-dimension grid search. Decision trees were successfully applied by Cho *et al.* [21] to categorize mushrooms and by Fujioka *et al.* [27] to identify chemical compounds.

5 Experimental results

In this section, we compare the performances of the machine learning algorithms described previously on two different datasets.

5.1 Outline of the electronic nose

The electronic nose system used is based on an array of six functionalized nano-diamond coated SAW sensors each with a fundamental frequency of 433.9 MHz [33, 32, 20]. The performances of the different feature sets and of the supervised learning algorithms were assessed on two data sets. The first data set is composed of seven different gases: ammonia (NH_3), sulphur dioxide (SO_2), hydrogen sulphide (H_2S), methanol (CH_3OH), toluene (C_7H_8), hydrogen cyanide (HCN) and dimethyl methylphosphonate ($C_3H_9O_3P$) at a concentration of 10 ppm, 8 ppm, 6 ppm, 4 ppm and 2 ppm (only 2 ppm for HCN). The gases

were generated from dilution from a calibrated gas or using a permeation tube ($C_3H_9O_3P$). Nitrogen was used as the reference and carrier gas to transport the volatile chemical compounds through the gas cell containing the sensors. The second data set concerns a real world application consisting of identifying counterfeit coffee capsules. The sensors were exposed to 21 types of genuine commercially available coffee capsules and 7 types of counterfeit capsules. The content of the capsules were emptied into a sealed beaker, the volatiles of a sample were transported through the gas cell containing the sensors using a pump. In both cases, the temperature of the sensors (22° C) and the flow rate (200 ml/min) above them were kept constant. Data acquisition was carried out at 10 Hz using the SAGAS instrument [63]. The sensors were exposed to the vapours for 15 seconds then the gas cell containing the sensors was purged for 30 seconds.

5.2 Experimental results

The previously described algorithms were implemented using Matlab with both the Statistics and Machine Learning and the Neural Network toolboxes. LMNN was implemented using the code available on the authors of [74] website and SNN was implemented by adapting the code available on the authors of [24] website. The results described in this section were obtained by selecting the most appropriate hyper parameters of each machine learning algorithm. The feedforward neural networks were trained using the Matlab’s automated regularization implementation of backpropagation algorithm. The topologies of the feedforward neural networks as well as the topology of the learning vector quantization network were obtained through an iterative process starting from a small structure and iteratively adding neurons. A training set and a validation set are used for the learning process; learning is stopped when the error on the validation step is increasing. The value of k for the k-NN, the kernel of the SVM and the parameters p and q of the ARMA model were obtained after performing a grid search process and selecting the ones yielding to the best performances. Samples were normalized using 0-1 scaling and dimension was reduced by keeping only the first k components explaining 90% of the variance. The results given in the following tables (Tab. 1 and Tab. 2) are the mean of the relative classification performances (ratio of the number of correctly classified samples over the number of samples) obtained during a 5-fold cross-validation process [3]. The ARMA models RMSE neighbourhood approach yields to a classification rate of respectively 0.863 and 0.485 on the gases data set and on the counterfeit coffee capsules data set. In k -fold cross-validation, the original database is split into k equal sized subsets. From the k subsets, a single subset is retained as the validation data for testing the model, and the remaining $k - 1$ subsets are used as training data. The cross-validation process is then repeated k times, with each of the k subsets used exactly once as the validation set. The primary idea that motivates this process is to try to estimate the accuracy of a classifier when applied to a new data set. Repeating the process allows to get more robust results since a classifier may have good performances on a data set

while having bad ones on another data set.

Table 1: Accuracy of the classifiers on the data set composed of the 7 gases.

	LDA	PLS-DA	DT	BT	KNN	LMNN	SVM	MLP	RBF	PNN	SNN	LVQ
Amplitude	Raw	0.883	0.882	0.879	0.886	0.882	0.888	0.879	0.882	0.872	0.853	0.87
	PCA	0.896	0.915	0.911	0.891	0.906	0.912	0.891	0.884	0.893	0.858	0.893
	ICA	0.874	0.914	0.888	0.883	0.888	0.889	0.914	0.892	0.873	0.86	0.874
Amplitude & rise time	Raw	0.868	0.862	0.888	0.868	0.87	0.876	0.882	0.878	0.872	0.862	0.858
	PCA	0.876	0.893	0.915	0.885	0.883	0.906	0.899	0.87	0.886	0.897	0.885
	ICA	0.886	0.902	0.883	0.874	0.872	0.892	0.897	0.898	0.865	0.869	0.865
Area	Raw	0.823	0.853	0.844	0.879	0.863	0.87	0.883	0.876	0.86	0.869	0.853
	PCA	0.838	0.884	0.871	0.905	0.89	0.885	0.919	0.899	0.882	0.893	0.877
	ICA	0.84	0.875	0.85	0.903	0.875	0.889	0.901	0.9	0.863	0.876	0.879
Phase Area	Raw	0.781	0.794	0.816	0.838	0.858	0.816	0.854	0.863	0.823	0.813	0.781
	PCA	0.787	0.812	0.839	0.85	0.885	0.834	0.881	0.881	0.84	0.822	0.788
	ICA	0.799	0.839	0.83	0.871	0.867	0.827	0.874	0.879	0.824	0.836	0.771
Exponential	Raw	0.863	0.87	0.873	0.886	0.89	0.883	0.881	0.883	0.874	0.87	0.852
	PCA	0.865	0.901	0.889	0.892	0.912	0.92	0.879	0.895	0.901	0.895	0.871
	ICA	0.868	0.905	0.865	0.894	0.907	0.885	0.892	0.902	0.882	0.87	0.863
Lorentzian	Raw	0.875	0.884	0.881	0.892	0.886	0.889	0.884	0.881	0.878	0.863	0.879
	PCA	0.898	0.913	0.9	0.935	0.889	0.913	0.918	0.902	0.903	0.892	0.904
	ICA	0.88	0.909	0.874	0.904	0.879	0.901	0.887	0.916	0.874	0.869	0.903
Double sigmoide	Raw	0.812	0.843	0.812	0.866	0.832	0.829	0.869	0.77	0.812	0.802	0.723
	PCA	0.813	0.859	0.833	0.883	0.861	0.858	0.896	0.793	0.827	0.833	0.743
	ICA	0.828	0.87	0.807	0.889	0.841	0.837	0.876	0.795	0.808	0.806	0.721
Cosine transform	Raw	0.804	0.792	0.793	0.802	0.776	0.822	0.819	0.829	0.813	0.781	0.772
	PCA	0.818	0.805	0.812	0.82	0.794	0.836	0.844	0.843	0.849	0.805	0.779
	ICA	0.818	0.824	0.797	0.813	0.777	0.798	0.83	0.847	0.819	0.791	0.791
Wavelet transform	Raw	0.823	0.813	0.848	0.856	0.844	0.863	0.87	0.866	0.858	0.852	0.848
	PCA	0.857	0.825	0.851	0.89	0.877	0.902	0.894	0.886	0.865	0.877	0.874
	ICA	0.841	0.855	0.86	0.874	0.829	0.868	0.9	0.897	0.873	0.869	0.839

Table 2: Accuracy of the classifiers on the counterfeit coffee capsules identification problem

	LDA	PLS-DA	DT	BT	KNN	LMNN	SVM	MLP	RBF	PNN	SNN	LVQ
Amplitude	Raw Data	0.536	0.542	0.521	0.536	0.615	0.6	0.612	0.593	0.593	0.542	0.554
	PCA	0.557	0.548	0.538	0.558	0.6	0.619	0.628	0.61	0.605	0.545	0.57
	ICA	0.536	0.547	0.529	0.535	0.587	0.602	0.625	0.578	0.596	0.551	0.564
Amplitude & rise time	Raw Data	0.553	0.559	0.522	0.563	0.623	0.615	0.628	0.615	0.599	0.578	0.562
	PCA	0.567	0.578	0.536	0.583	0.61	0.635	0.637	0.635	0.611	0.597	0.574
	ICA	0.554	0.568	0.526	0.568	0.641	0.612	0.637	0.586	0.604	0.576	0.575
Area	Raw Data	0.532	0.525	0.536	0.542	0.552	0.563	0.559	0.567	0.564	0.521	0.531
	PCA	0.538	0.534	0.54	0.564	0.564	0.578	0.578	0.577	0.571	0.536	0.539
	ICA	0.529	0.524	0.54	0.54	0.559	0.564	0.574	0.566	0.578	0.541	0.537
Phase Area	Raw Data	0.492	0.506	0.509	0.513	0.518	0.521	0.522	0.534	0.526	0.502	0.518
	PCA	0.5	0.513	0.517	0.527	0.537	0.54	0.537	0.552	0.538	0.508	0.537
	ICA	0.483	0.503	0.514	0.512	0.527	0.526	0.532	0.525	0.537	0.51	0.531
Exponential model	Raw Data	0.554	0.571	0.534	0.562	0.61	0.595	0.581	0.598	0.597	0.567	0.583
	PCA	0.576	0.583	0.552	0.574	0.618	0.606	0.602	0.613	0.61	0.583	0.59
	ICA	0.55	0.564	0.555	0.574	0.569	0.608	0.588	0.593	0.6	0.576	0.605
Lorentzian model	Raw Data	0.563	0.554	0.543	0.567	0.596	0.601	0.589	0.603	0.587	0.532	0.564
	PCA	0.579	0.576	0.549	0.588	0.546	0.613	0.613	0.63	0.595	0.541	0.567
	ICA	0.561	0.553	0.549	0.567	0.534	0.604	0.607	0.598	0.604	0.527	0.572
Double sigmoide model	Raw Data	0.532	0.521	0.546	0.576	0.532	0.556	0.541	0.556	0.536	0.512	0.523
	PCA	0.544	0.528	0.557	0.589	0.534	0.575	0.55	0.563	0.542	0.516	0.538
	ICA	0.524	0.523	0.547	0.586	0.515	0.532	0.553	0.544	0.552	0.521	0.532
Discrete cosine transform	Raw Data	0.526	0.529	0.548	0.551	0.536	0.561	0.564	0.561	0.549	0.538	0.554
	PCA	0.541	0.546	0.557	0.557	0.541	0.581	0.576	0.57	0.562	0.536	0.56
	ICA	0.525	0.529	0.555	0.549	0.536	0.58	0.584	0.552	0.566	0.554	0.56
Wavelet transform	Raw Data	0.522	0.519	0.518	0.523	0.531	0.539	0.551	0.553	0.548	0.512	0.537
	PCA	0.531	0.532	0.527	0.533	0.548	0.54	0.571	0.572	0.558	0.519	0.549
	ICA	0.52	0.512	0.525	0.522	0.525	0.539	0.554	0.543	0.554	0.514	0.542

5.3 Discussion

Firstly, we remark that, conversely to the linear classifiers which yield to the poorest results, neural networks, large margin nearest neighbours and support vector machine yield to the best classification rates. These results can be explained by their capabilities to learn complex decision boundaries. The incapacity of decision trees to deal with non axis-parallel decision boundaries explain why this algorithm yielded to the most inaccurate results. Those drawbacks are overcome by the bagged tree algorithm which do not suffer from these issues and hence yields to better performances. Other algorithms produced very disparate results and their lack of interpretability makes any further investigation strenuous. Secondly, it seems that preprocessing the data and reducing their dimension with the principal component analysis (PCA) increases the classification performances. ICA globally has less impact than PCA on the performances of each algorithm. One might explain this phenomenon by the fact that ICA aims at removing the linearity which are non-existent in this case since the used electronic nose is composed of six different sensors. Finally, this work confirms that the transient response contains relevant information when all the gases are transported through the gas cell containing the sensors in the same way. However, one has to be very careful when using the transient response since the performances obtained using transient response's information often yield to poor results in the data set composed of the coffee capsules. This can be explained by the fact that the experimental set up and the use of a pump generate very disparate transient response profiles. This has also to be considered when dealing with the DCT and the DWT because these features intrinsically have information about the transient response.

6 Conclusion

This paper has reviewed the reported literature on machine learning algorithms for e-nose based volatile compounds recognition. A benchmark of the performances obtained for different features, for different preprocessing techniques and for different machine learning algorithms was performed yielding to a quantitative comparison of those methods. The different machine learning algorithms and the data pretreatment methods produce very disparate results. However, we can remark a few general trends. The neural network and the large margin nearest neighbours (LMNN) always produce the best results followed by bagged trees and the support vector machine (SVM). Moreover, this benchmark confirms that reducing the data dimension with (PCA) increases the classification performance. It also confirms that the transient response contains relevant information but should be used carefully.

7 Acknowledgments

This research was supported by the French trans-governmental CBRN-E R&D program.

References

References

- [1] Teodoro Aguilera, Jesús Lozano, José A Paredes, Fernando J Alvarez, and José I Suárez. Electronic nose based on independent component analysis combined with partial least squares and artificial neural networks for wine prediction. *Sensors*, 12(6):8055–8072, 2012.
- [2] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.
- [3] Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4(1):40–79, 2010.
- [4] S. Balasubramanian, S. Panigrahi, C.M. Logue, C. Doetkott, M. Marchello, and J.S. Sherwood. Independent component analysis-processed electronic nose data for predicting salmonella typhimurium populations in contaminated beef. *Food Control*, 19(3):236 – 246, 2008.
- [5] Elizabeth A. Baldwin, Jinhe Bai, Anne Plotto, and Sharon Dea. Electronic noses and tongues: Applications for the food and pharmaceutical industries. *Sensors*, 11(5):4744–4766, 2011.
- [6] I.A Basheer and M Hajmeer. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43(1):3 – 31, 2000.
- [7] Simona Benedetti, Saverio Mannina, Anna Gloria Sabatini, and Gian Luigi Marcazzan. Electronic nose and neural network use for the classification of honey. *Apidologie*, 35(4):1–6, 2004.
- [8] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.
- [9] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [10] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.
- [11] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

- [12] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [13] Richard G Brereton and Gavin R Lloyd. Partial least squares discriminant analysis: taking the magic away. *Journal of Chemometrics*, 28(4):213–225, 2014.
- [14] K Brudzewski, S Osowski, and T Markiewicz. Classification of milk by means of an electronic nose and svm neural network. *Sensors and Actuators B: Chemical*, 98(2):291–298, 2004.
- [15] S Buratti, S Benedetti, M Scampicchio, and EC Pangerod. Characterization and classification of italian barbera wines by using an electronic nose and an amperometric electronic tongue. *Analytica Chimica Acta*, 525(1):133–139, 2004.
- [16] Hyung-Gi Byun, Nam-Yong Kim, Jeung-Soo Huh Krishna C. Persaud and, and Duk-Dong Lee. Application of adaptive rbf networks to odour classification using conducting polymer sensor array. In *International Symposium on Olfaction and Electronic Noses (ISOEN)*, pages 121–126, 2000.
- [17] L. Carmel, S. Levy, D. Lancet, and D. Harel. Feature extraction of chemical sensors in phase space. *Sensors and Actuators B: Chemical*, 95(1–3):132 – 139, 2003.
- [18] R. A. Carmona. Time series models: Ar, ma, arma, and all that. In *Statistical Analysis of Financial Data in S-Plus*, Springer Texts in Statistics, pages 239–309. Springer New York, 2004.
- [19] C. Cevoli, L. Cerretani, A. Gori, M. F. Caboni, T. Gallina Toschi, and A. Fabbri. Classification of pecorino cheeses using electronic nose combined with artificial neural network and comparison with gc–ms analysis of volatile compounds. *Food Chemistry*, 129(3):1315–1319, 2011.
- [20] E Chevallier, E Scorsone, and P Bergonzo. New sensitive coating based on modified diamond nanoparticles for chemical saw sensors. *Sensors and Actuators B: Chemical*, 154(2):238–244, 2011.
- [21] Jung Hwan Cho and Pradeep U Kurup. Decision tree approach for classification and dimensionality reduction of electronic nose data. *Sensors and Actuators B: Chemical*, 160(1):542–548, 2011.
- [22] Chen Chuanzhi, Ma Jinyi, Zuo Boli, and Jiang Hongmin. A novel toxic gases detection system based on saw resonator array and probabilistic neural network. In *Electronic Measurement and Instruments*, pages 499–503, 2007.
- [23] A. D’Amico and E. Verona. Saw sensors. *Sensors and Actuators*, 17(1):55 – 66, 1989.

- [24] Peter U Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2015.
- [25] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.
- [26] Imola K Fodor. A survey of dimension reduction techniques. Technical report, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 2002.
- [27] Kouki Fujioka, Nobuo Shimizu, Yoshinobu Manome, Keiichi Ikeda, Kenji Yamamoto, and Yasuko Tomizawa. Discrimination method of the volatiles from fresh mushrooms by an electronic nose using a trapping system and statistical standardization to reduce sensor value variation. *Sensors*, 13(11):15532–15548, 2013.
- [28] Diego L García-González and Ramón Aparicio. Sensors: From biosensors to the electronic nose. *Grasas y Aceites*, 53(1):96–114, 2002.
- [29] Julian W. Gardner and Philip N. Bartlett. A brief history of electronic noses. *Sensors and Actuators B: Chemical*, 18(1–3):210 – 211, 1994.
- [30] M. Gaudioso, W. Khalaf, and C. Pace. On the use of the svm approach in analyzing an electronic nose. In *International Conference on Hybrid Intelligent Systems (HIS)*, pages 42–46, 2007.
- [31] R. Ghaffari, Fu Zhang, D. Iliescu, E. Hines, M. Leeson, R. Napier, and J. Clarkson. Early detection of diseases in tomato crops: An electronic nose and intelligent systems approach. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, July 2010.
- [32] HA Girard, JC Arnault, S Perruchas, S Saada, T Gacoin, J-P Boilot, and P Bergonzo. Hydrogenation of nanodiamonds using mpcvd: A new route toward organic functionalization. *Diamond and Related Materials*, 19(7):1117–1123, 2010.
- [33] Hugues A Girard, Sandrine Perruchas, Celine Gesset, Marc Chaigneau, Laetitia Vieille, Jean-Charles Arnault, Philippe Bergonzo, Jean-Pierre Boilot, and Thierry Gacoin. Electrostatic grafting of diamond nanoparticles: a versatile route to nanocrystalline diamond thin films. *ACS applied materials & interfaces*, 1(12):2738–2746, 2009.
- [34] Selda Güney and Ayten Atasoy. Multiclass classification of n-butanol concentrations with k-nearest neighbor algorithm and support vector machine in an electronic nose. *Sensors and Actuators B: Chemical*, 166(1):721 – 725, 2012.

- [35] Amara Graps. An introduction to wavelets. *Computational Science & Engineering, IEEE*, 2(2):50–61, 1995.
- [36] Ricardo Gutierrez-Osuna. Pattern analysis for machine olfaction: a review. *Sensors Journal, IEEE*, 2(3):189–202, 2002.
- [37] James Douglas Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, 1994.
- [38] Marti A. Hearst, Susan T Dumais, Edgar Osman, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.
- [39] Klaus Hechenbichler and Klaus Schliep. Weighted k-nearest-neighbor techniques and ordinal classification. Technical report, 2004.
- [40] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *in: IEEE Transactions on Neural Networks*, pages 415–425, 2002.
- [41] Chi-Cheng Huang, Shih-Hsin Tu, Ching-Shui Huang, Heng-Hui Lien, Liang-Chuan Lai, and Eric Y Chuang. Multiclass prediction with partial least square regression for gene expression data: applications in breast cancer intrinsic taxonomy. *BioMed research international*, 2013, 2013.
- [42] Aapo Hyvarinen. Independent component analysis by minimization of mutual information. Technical report, Laboratory of Computer and Information Science, Helsinki University of Technology, 1997.
- [43] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000.
- [44] D. James, S. M. Scott, W. T. O’Hare, Z. Ali, and F. J. Rowell. Classification of fresh edible oils using a coated piezoelectric sensor array based electronic nose with soft computing approach for pattern recognition. In *Transaction of the Institute of Measurement and Control*, volume 26, pages 3–18, 2004.
- [45] Sunil K. Jha and R. D. S. Yadava. Preprocessing of saw sensor array data and pattern recognition. *IEEE Sensors Journal*, 9(10):1202–1208, 2009.
- [46] Ian Jolliffe. *Principal component analysis*. John Wiley & Sons, Ltd, 2014.
- [47] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, Sep 1990.
- [48] Teuvo Kohonen. *Learning vector quantization*. Springer, 2001.
- [49] Sichu Li. Overview of odor detection instrumentation and the potential for human odor detection in air matrices. Technical report, MITRE Nanosystems Group, 2009.

- [50] Wu Lili, Yuan Chao, Lin Aiyang, and Zheng Baozhou. Identification of early moldy rice samples by pca and pnn. *Communications and Information Processing*, 288(1):506–514, 2012.
- [51] Eduard Llobet, Jesús Brezmes, Xavier Vilanova, Jesús E. Sueiras, and Xavier Correig. Qualitative and quantitative analysis of volatile organic compounds using transient and steady-state responses of a thick-film tin oxide gas sensor array. *Sensors and Actuators B: Chemical*, 41(1–3):13 – 21, 1997.
- [52] Amy Loutfi, Silvia Coradeschi, Ganesh Kumar Mani, Prabakaran Shankar, and John Bosco Balaguru Rayappan. Electronic noses for food quality: A review. *Journal of Food Engineering*, 144(1):103 – 111, 2015.
- [53] Gangadharrao S Maddala and Kajal Lahiri. *Introduction to econometrics*. Wiley, 2009.
- [54] Eugenio Martinelli, Christian Falconi, Arnaldo D’Amico, and Corrado Di Natale. A feature extraction method for chemical sensors in electronic noses. *Sensors and Actuators B: Chemical*, 93(1–3):67 – 76, 2003.
- [55] A. Mayoue, A. Martin, G. Lebrun, and A. Larue. Recursive least squares algorithm dedicated to early recognition of explosive compounds thanks to multi-technology sensors. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8761–8765, May 2013.
- [56] B. Naidoo and A. D. Broadhurst. Sensor array data processing using a 2-d discrete cosine transform. In *International Symposium on Olfaction and Electronic Noses (ISOEN)*, pages 153–158, 2000.
- [57] Corrado Di Natale, Eugenio Martinelli, and Arnaldo D’Amico. Counteraction of environmental disturbances of electronic nose data by independent component analysis. *Sensors and Actuators B: Chemical*, 82(2–3):158 – 165, 2002.
- [58] V. O. S. Olunloyo, T. A. Ibidapo, and R. R. Dinrifo. Neural network based electronic nose for cocoa beans quality assessment. *Agricultural Engineering Journal*, 13(4):1521–1533, 2011.
- [59] Matteo Pardo and Giorgio Sberveglieri. Classification of electronic nose data with support vector machines. *Sensors and Actuators B: Chemical*, 107(2):730 – 737, 2005.
- [60] Shanshan Qiu, Liping Gao, and Jun Wang. Classification and regression of elm, lvq and svm for e-nose data of strawberry juice. *Journal of Food Engineering*, 144(1):77 – 85, 2015.
- [61] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, March 1986.

- [62] C Radhakrishna Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948.
- [63] M Rapp, J Reibel, A Voigt, M Balzer, and O Bülow. New miniaturized saw-sensor array for organic gas detection driven by multiplexed oscillators. *Sensors and Actuators B: Chemical*, 65(1–3):169 – 172, 2000.
- [64] Frank Rock, Nicolae Barsan, and Udo Weimar. Electronic nose: current status and future trends. *Chemical reviews*, 108(2):705–725, 2008.
- [65] Thaddeus A. Roppel and Denise Wilson. Improved chemical identification from sensor arrays using intelligent algorithms. *Advanced Environmental and Chemical Sensing Technology*, 4205(1):260–266, 2001.
- [66] SS Schiffman, DW Wyrick, R Gutierrez-Osuna, and HT Nagle. Effectiveness of an electronic nose for monitoring bacterial and fungal growth. In *International Symposium on Olfaction and Electronic Noses (ISOEN)*, pages 173–180, 2000.
- [67] Prashant Singh and R. D. S. Yadava. Wavelet based fuzzy inference system for simultaneous identification and quantitation of volatile organic compounds using saw sensor transients. In *International Conference on Swarm, Evolutionary, and Memetic Computing (SEMCCO)*, pages 319–327, 2011.
- [68] A. Che Soh, K. K. Chow, U. K. Mohammad Yusuf, A. J. Ishak, M.K. Hassan, and S. Khamis. Development of neural network based electronic nose for herbs recognition. *International Journal on Smart Sensing and Intelligent Systems*, 7(2):584–609, 2014.
- [69] Petre Stoica and Yngve Selen. Model-order selection: a review of information criterion rules. *IEEE Signal Processing Magazine*, 21(4):36–47, 2004.
- [70] Kea-Tiong Tang, Shih-Wen Chiu, Chih-Heng Pan, Hung-Yi Hsieh, Yao-Sheng Liang, and Ssu-Chieh Liu. Development of a portable electronic nose system for the detection and classification of fruity odors. *Sensors*, 10(10):9179–9193, 2010.
- [71] Bipan Tudu, Arun Jana, Animesh Melta, Devdulal Ghosh, Nabarun Bhattacharyya, and Rajib Bandyopadhyay. Electronic nose for black tea quality evaluation by an incremental rbf network. *Sensors and Actuators B: Chemical*, 138(1):90–95, 2009.
- [72] Anthony P.F. Turner and Naresh Magan. Electronic noses and disease diagnostics. *Nature reviews in Microbiology*, 2(2):161–166, 2004.
- [73] Jilles Vreeken et al. Spiking neural networks, an introduction. *Institute for Information and Computing Sciences, Utrecht University Technical Report UU-CS-2003-008*, 2002.

- [74] Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. In Y. Weiss, B. Schölkopf, and J.C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1473–1480. MIT Press, 2006.
- [75] Alphus Wilson. Review of electronic-nose technologies and algorithms to detect hazardous chemicals in the environment. *Procedia Technology*, 1(1):453–463, 2012.
- [76] R. D. S. Yadava and Ruchi Chaudhary. Solvation, transduction and independent component analysis for pattern recognition in saw electronic nose. *Sensors and Actuators B: Chemical*, 113(1):1–21, 2006.
- [77] Weixiang Zhao, Abhinav Bhushan, Anthony D Santamaria, Melinda G Simon, and Cristina E Davis. Machine learning: A crucial tool for sensor design. *Algorithms*, 1(2):130–152, 2008.