REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

CLASSIFICATION OF VOC VAPORS USING MACHINE LEARNING ALGORITHMS

Serra AKSOY

MASTER OF SCIENCE THESIS

Department of Mathematics

Mathematics Program

Supervisor

Assoc. Prof. Dr. Muttalip ÖZAVŞAR

Co-Supervisor

Prof. Dr. Ahmet ALTINDAL

July, 2022

REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

CLASSIFICATION OF VOC VAPORS USING MACHINE LEARNING ALGORITHMS

A thesis submitted by Serra AKSOY in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 18.07.2022 in Department of Mathematics, Mathematics Program.

Assoc. Prof. Dr. Muttalip ÖZAVŞAR Yıldız Technical University Supervisor	Prof. Dr. Ahmet ALTINDAL Yıldız Technical University Co-supervisor

Approved By the Examining Committee

Assoc. Prof. Dr. Muttalip ÖZAVŞAR, Supervisor
Yildiz Technical University _____

Assist. Prof. Dr. Sebahat Ebru DAŞ, Member
Yildiz Technical University _____

Prof. Dr. Nazım AGHAYEV, Member
Fatih Sultan Mehmet Foundation University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Classification of VOC Vapors Using Machine Learning Algorithms supervised by my supervisor, Assoc. Prof. Dr. Muttalip ÖZAVŞAR. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Serra AKSOY

Signature

Dedicated to my family

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor, Assoc. Prof. Dr. Muttalip ÖZAVŞAR, for his guidance and support throughout this research. He motivated me not only in this thesis but also in other academic issues, and his support made possible the completion of this thesis.

Also, I am indebted to my co-supervisor, Prof. Dr. Ahmet ALTINDAL, for his guidance to determine the topic of my thesis and valuable support in the experiment part.

Last but not least, a special thanks to my dear family, who has always been there to support me along the way. I am grateful for their belief in me.

Serra AKSOY

TABLE OF CONTENTS

LIST OF SYMBOLS

I_0	Baseline Current of CoPc Thin Film
ΔI	Change in Sensor Current
C_v	Concentration of the VOC Vapor
d(x,y)	Distance between x and y
$\vec{a} \cdot \vec{b}$	Inner Product of $\vec{a}$ and $\vec{b}$
S	Sensitivity

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
CoPc	Cobalt Phthalocyanine
CV	Cross Validation
KNN	K-Nearest Neighbor
RBFN	Radial Basis Function Network
SVM	Support Vector Machine
VOC	Volatile Organic Compound

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Classification of VOC Vapors Using Machine Learning Algorithms

Serra AKSOY

Department of Mathematics

Master of Science Thesis

Supervisor: Assoc. Prof. Dr. Muttalip ÖZAVŞAR

Co-supervisor: Prof. Dr. Ahmet ALTINDAL

In this thesis, the sensing properties of a cobalt phthalocyanine (CoPc) thin film at six different volatile organic compound (VOC) vapors (methanol, ethanol, butanol, isopropyl alcohol, acetone, and ammonia) concentrations from 50 to 450 ppm are investigated. It is observed that the interaction between the VOC vapors and the CoPc surface is not selective. In this sense, it is aimed that using machine learning algorithms the present sensor, which is poorly selective, can be transformed into a more efficient one with better detection ability. As a feature, 10 seconds of responses taken from the steady state region are used without any additional processing technique. Among classification algorithms, K-Nearest Neighbor (KNN) and Bagged Tree reach the highest accuracy of more than 95%. The performance of this feature is also compared with the traditional gas sensor response features such as maximum value,integral and derivative. Results indicate that the feature based on 10 seconds of responses taken from the steady state

region is much better than those based on the traditional features. The best accuracy at the end of this study is obtained as 96.7% with KNN method.

Keywords: Volatile organic compound (VOC), machine learning, classification, k-nearest neighbor (KNN), bagged tree.

ÖZET

Uçucu Organik Bileşen Buharlarının Makine Öğrenmesi Algoritmaları ile Sınıflandırılması

Serra AKSOY

Matematik Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. Muttalip ÖZAVŞAR

Eş-Danışman: Prof. Dr. Ahmet ALTINDAL

Bu tezde, kobalt ftalosiyanin (CoPc) ince filminin, konsantrasyonları 50 ile 450 ppm arasında değişen altı farklı uçucu organik bileşen (UOB) buharlarını (metanol, etanol, bütanol, izopropil alkol, aseton ve amonyak) algılama özellikleri incelenmiştir. UOB buharları ile CoPc yüzeyi arasındaki etkileşimin seçici olmadığı gözlemlenmiştir. Bu bağlamda seçiciliği zayıf olan mevcut sensörün makine öğrenmesi algoritmaları kullanılarak daha iyi algılama kabiliyetine sahip daha verimli bir sensöre dönüştürülmesi amaçlanmıştır. Öznitelik olarak, kararlı durum bölgesinden alınan 10 saniyelik yanıtlar herhangi bir ek işleme tekniği olmadan kullanılmıştır. Sınıflandırma algoritmaları arasında K-En Yakın Komşu (KNN) ve Torbalı Ağaç algoritmaları %95'in üzerinde en yüksek doğruluğa ulaşmıştır. Bu özniteliğin performansı aynı zamanda maksimum değer, integral ve türev gibi geleneksel gaz sensörü tepki öznitelikleriyle de karşılaştırılmıştır. Sonuçlar kararlı durum bölgesinden alınan 10 saniyelik yanıtlara dayalı özelliğin, geleneksel

özelliklere dayalı olanlardan çok daha iyi olduğunu göstermektedir. Bu çalışma sonucunda en iyi doğruluk KNN yöntemi ile %96,7 olarak elde edilmiştir.

Anahtar Kelimeler: Uçucu organik bileşen, makine öğrenmesi, sınıflandırma, k-en yakın komşu, torbalı ağaç.

1
INTRODUCTION

1.1 Literature Review

Volatile organic compounds (VOCs) are organic chemical compounds whose composition makes it possible for them to evaporate under normal indoor atmospheric conditions of temperature and pressure [1]. As VOCs are well-known air pollutants emitted from cooking, construction materials, office equipment and consumer products, etc., they have short- and long-term adverse health effects such as eye, nose, and ear irritation; headaches; nausea and cancer. According to World Health Organization (WHO) data, 4.2 million people die every year because of ambient air pollution [2]. Thus, monitoring VOC levels has a great importance for indoor air quality control to keep them within safe levels.

On the other hand, gases produced by the organs and exhaled through the breath differ in case of illness. Studies in recent years show that some diseases can be diagnosed at an early stage by comparing the breath samples of patients and healthy individuals. The selective detection of gas differences between breath samples can be used in the diagnosis of many cases such as in the early detection of some types of cancer [3,4], in the analysis of breath odor before and after hemodialysis in nephrism [5], in the diagnosis of breath odor caused by bacteria living in the oral cavity [6], and in the analysis of human sweat [7]. Thus, VOCs can also be used as a diagnostic biomarker in some diseases, and the detection of them has a vital importance. Although methods such as laser spectroscopy are used in breath analysis, these methods require very expensive and large-volume equipment and manpower trained in this field. Therefore, the applicability of these methods is very limited. Nano-sized chemical sensors are devices that have the potential to fill the gap in this field due to reasons such as being easily applicable, environmentally friendly, cost and energy efficient, and not requiring trained manpower.

As a result, it can be said that the selective identification of VOC vapors has a great importance in wide range of areas from indoor air quality control to early diagnosis of certain diseases [8-11]. In this regard, gas sensors offer many advantages including low cost, high response and recovery times, and low power consumption [12]. Among the various types of gas sensors based on different operating principles, such as acoustic wave-based, calorimetric, capacitive, optical, etc. resistive type gas sensors are of special importance. In its simplest form, a resistive gas sensor consists of a sensing element and a transducer which is capable of converting energy of one kind into energy of another kind. High sensitivity exhibited by phthalocyanines (Pcs) and their derivatives in the form of measurable changes in their conductivity when exposed to reducing or oxidizing gases make them natural choices for VOC vapor sensing. Pcs as sensing element in resistive type gas sensors have been studied intensively because of their open coordination sites for axial ligation [13]. As a result of their conductivity, Pcs exhibit excellent sensing characteristics of various gases, such as NO_2 [14], CO_2 [15], SO_2 [16], VOCs [17], etc.

The main drawback of Pc based sensing devices is indeed their lack of selectivity towards VOC vapors which strongly limits their use in sensing applications. Selectivity describes to characteristics that determine whether a sensor can respond selectively to a group of analytes or even specifically to a single analyte [18]. This is a problem that must be solved, especially if it is aimed to manufacture high-tech devices such as an electronic nose. One essential part of an electronic nose is a system that finds a relationship between the sensor response and the gas type in order to detect gases selectively. This can be achieved by utilizing machine learning approach.

When the gas sensor data set is considered, pattern classes are the types of gases and patterns are the responses of the sensor to each gas. The goal is to find a relationship between the sensor responses and the gas types. To achieve this, feature extraction and classification are two fundamental steps that must be considered. However, it is difficult to establish a general pattern recognition technique for gas classification under different conditions. For this reason, the techniques to be used vary from application to application. A wide variety of

feature extraction methods are used in chemical sensor applications (for a review see [19]). Finding the best suitable features that have adequate information about sensor response plays a key role in the performance of classification. When this is made, finding a successful model for classification is rather easy. Therefore, the features which represent the original data set must be chosen as informative as possible. According to the research on gas sensor data classification, generally there are three main feature extraction methods. The first one is to extract piecemeal signal features from the sensor response curve, such as maximum response [20], integral [21], derivative [22] etc. The second one is based on some transforms, such as Fast Fourier Transform [23] and Discrete Wavelet Transform [24]. The third one is based on curve fitting [25]. Besides the mentioned studies, there are other methods which used very often recent years such as Windowed Time Slicing [26], Phase Space [27], Parallel Factor Analysis [28] etc. On the other hand, for the successful classification of gases, the chosen model for classification also plays an important role. KNN [29], SVM [30], ANN [31] and RBFN [32] are some of the most commonly used methods that give high classification accuracy rate in many cases.

1.2 Objective of the Thesis

In this study, for detection of six different VOC vapors (methanol, ethanol, butanol, isopropyl alcohol, acetone, and ammonia) of nine different concentrations from 50 ppm to 450 ppm, the experiment was carried out using a resistive type gas sensor based on CoPc (cobalt phthalocyanine) surface. It is observed that the interaction between the vapors and the surface is not selective enough. This thesis aims to improve the selectivity of this resistive type gas sensor with machine learning approach for detection of considered VOC vapors. In order to solve this problem, it is aimed that the system can distinguish vapors with the use of machine learning algorithms.

1.3 Hypothesis

We were particularly interested in extracting piecemeal signal features from the sensor response curve, such as maximum values, integrals, derivatives. Since the maximum value represents the final steady-state feature of the entire dynamic response process in the final balance, which reflects the maximum reaction degree change of sensors responding to vapors, it is usually used as the most widely used and simple electronic nose feature [33]. Therefore, many previous works have made contributions based on the steady state response feature in various gas sensor applications [34-37]. As many studies did to extract robust information from the sensor response curve, we first used the steady state response. However, results show that the classification accuracy was very low when only the steady state response was used. Hence, it is thought that not only one response but also a few seconds of responses from the steady state region could be used in order to utilize more information from the response curve. Without any additional feature processing technique, the 10 s data extracted from the response curve after the sensor reaches 90% of its maximum value were directly used without taking into account the other information in the whole response curve. The performance of this feature was tested by various machine learning classification algorithms such as Decision Tree, Support Vector Machine (SVM), K-nearest Neighbor (KNN), and Ensemble Method. The results were compared with the other traditional feature extraction methods such as derivative and integral as well.

2
THEORETICAL BACKGROUND

This section aims to explain the basic issues related to the machine learning approach used in this thesis. In this regard, first the main idea of utilizing machine learning in real world applications is given. Then, the basic steps of machine learning and its types are explained. After, the criterias used to measure the performance of the models are given. Finally, the machine learning algorithms used in this thesis are introduced. Figure 2.1 shows a simple road-map of a machine learning task.

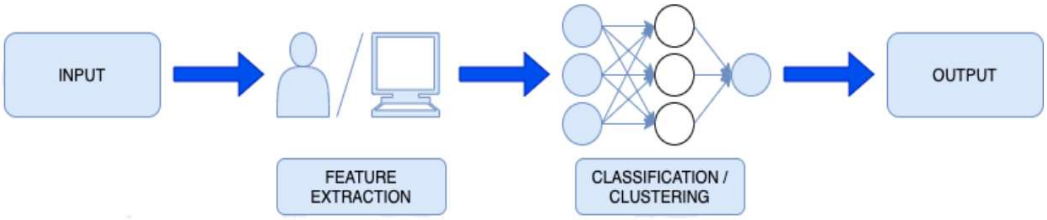

Figure 2.1 Sample machine learning road map [38]

2.1 The Aim of Using Machine Learning

Machine learning is a branch of artificial intelligence that allows for modeling by learning from data and has been the subject of numerous studies in recent years. It has a great potential in today's technology, and its importance is increasing very rapidly day by day. The majority of smart systems nowadays are built using a machine learning technique. Even if we aren't often aware of its use, it is utilized in various areas such as image recognition, speech recognition, traffic prediction, product recommendations, email spam and malware filtering, medical diagnosis, automatic language translation, etc.

The aim of machine learning is to develop and improve algorithms to reveal valuable information in the data we have. Machine learning is needed for a variety of reasons. Here are a few examples:

• Some works are difficult to express without using examples; for example, we may be able to provide input-output pairings but not a clear relationship between the inputs and the related outputs. The important thing is that machines can adjust their internal structure to produce true outputs for a large number of sample inputs and thus suitably constrain their input/output function to approximate the relationship implicit in the examples.

• It is possible that hidden among the large amounts of data are important relationships and correlations. These correlations are frequently extracted using machine learning approaches (data mining).

• Designers often construct machines that do not work as well as they should in their intended contexts. Because, several aspects of the working environment may not be fully understood at the time of design. Machine learning technologies can be used to improve existing machine designs on the job.

• The amount of information available regarding certain tasks may be too large for explicit encoding by humans. Machines that learn this information gradually may be able to catch more of it than humans would want to write down.

• Environments change over time. Machines that can adapt to a changing environment would reduce the need for constant redesign.

•Humans are continually discovering new information about tasks. In the world, there is a never-ending stream of new happenings. It's impractical to keep redesigning AI systems to accommodate to new knowledge, but machine learning approaches might be able to track a lot of it [39].

2.2 Machine Learning Steps

The five basic steps of machine learning can be explained as follows [Lantz, 2015].

2.2.1 Collecting Data

It is the process of gathering raw information from various sources. Since the quality and quantity of gathered data directly affect the accuracy of the desired

system, the better the diversity, density and volume of the data, the better the learning for machine learning will be.

2.2.2 Preparing the Data

Real world data is frequently disorganized, missing, or noisy. Hence, before the data is given to the system to create the training model, it has to be organized and cleaned up in the preprocessing step. In order to construct a successful model, this step is critical and usually takes time.

2.2.3 Training a Model

This step includes the selection of the appropriate algorithm to create a model. Here, the data connects to the selected algorithm that uses mathematical models to learn and develop predictions. If the considered type of machine learning is supervised learning, the data is divided into two parts: train and test sets. The system is run with the training set and a suitable model for the data is created to be used in the testing phase.

2.2.4 Evaluating the Model

In this step, in order to determine the performance of the system, the model constructed in the previous stage is applied to the test set, which was not seen previously by the model. The used model is analyzed by performance evaluation metrics. If the results are not good enough, the model should be improved and retrained in the next step.

2.2.5 Improving the Performance

If the results obtained in the previous step are not good enough, this step may require the choice of a completely different model or the model should be improved and retrained in this step.

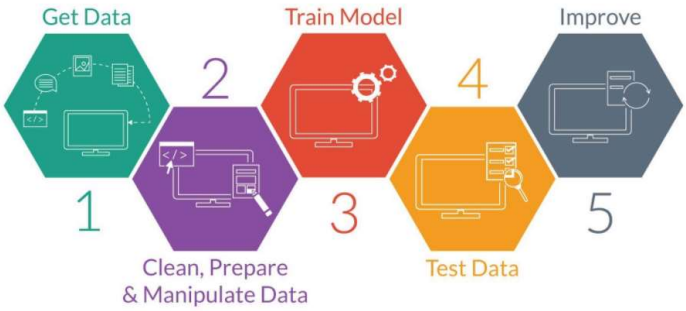

Figure 2.2 Machine Learning Workflow [40]

2.3 Types of Machine Learning

As shown in Figure 2.3, machine learning is basically divided into three categories dealing with different types of learning tasks.

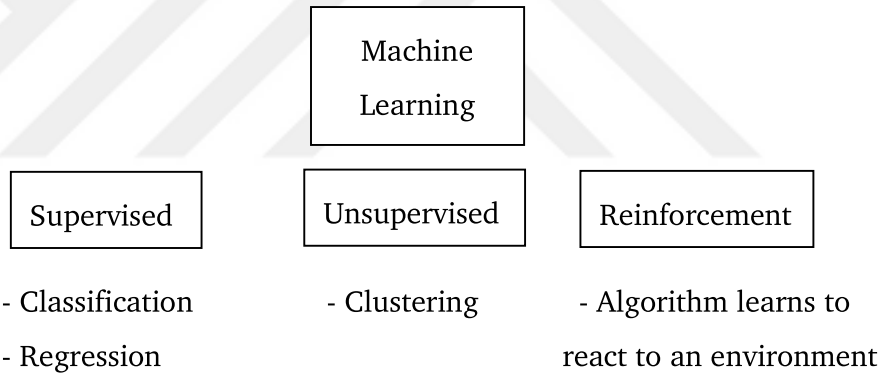

Figure 2.3 Types of Machine Learning

2.3.1 Supervised Learning

Supervised learning is developed based on samples consisting of both input and output parameters, in other words, models are trained using labeled data. The aim of this type of learning is to enable the machine to experience the problem situation with a certain result and learn the way that leads to the known result. It includes two types of problems. One is classification and the other one is regression. In both regression and classification problems; there is an input, an output, and the task is to learn the mapping from the input to the output.

In a classification problem, we are given a pattern to classify into one of c classes. It is assumed that the c number of classes is known a priori. Each pattern is

represented by a set of feature values, $x(i)$, $i = 1, 2, \ldots, l$, which make up the $l-$ *dimensional feature vector*[1] $x = [x(1), x(2), \ldots, x(l)]^T \in R^l$. We assume that each pattern is represented uniquely by a single feature vector and that it can belong to only one class [41]. If the desired output variable has a real or continuous value, the issue is known as regression. The data is fit to the best hyperplane through the points [42].

This thesis includes a data set where each data point has a label for the classification task. Hence, the all machine learning algorithms used in this work are supervised learning models.

2.3.2 Unsupervised Learning

In contrast to supervised learning, unsupervised learning consists solely of input data without any associated target value. The machine itself determines the output because there is no system that monitors and controls the learning process according to the presented input data. A classic example of such a task is splitting a data set into subgroups of comparable objects. This process is called as clustering. Basically the aim of such unsupervised learning tasks are clustering, density estimation and project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization [43].

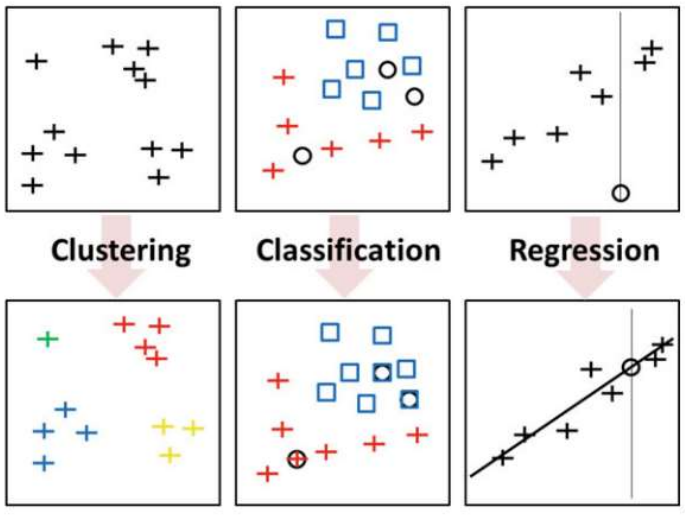

Figure 2.4 (a) Clustering, (b) Classification and (c) Regression [44]

As a summary, in Figure 2.4 the comparing scheme is given. As it can be seen, in classification task distinct groups of data exist and new data points should be classified into these existing groups. In clustering no groups of data exist initially but one creates groups from the data close to each other. Finding a line with a particular slope that can accurately represent the data is the aim of regression analysis techniques.

2.3.3 Reinforcement Learning

In reinforcement learning, there are reward values associated with the various steps. The purpose of the model is to collect as many rewards as possible in order to reach a final destination.

In some applications, the output of the system is a series of activities. In such a scenario, a single action is unimportant; what matters is the policy, which is the series of correct activities that leads to the desired outcome. There is no such thing as the best action in any intermediate state; an action is good if it is part of a good policy. In this situation, the machine learning algorithm should be able to evaluate policy goodness and learn from previous good action sequences in order to build a policy. Such learning methods are called reinforcement learning algorithms [45].

2.4 Evaluating Model Performance

A successful machine learning model should be able to properly generalize the training data to all the data in problem. This enables us to make future predictions based on data that the model has never seen before. Here, the two most common problems that affect the performance of model in machine learning are overfitting and underfitting. In order to avoid these problems cross valdaition method is used. Various performance metrics are utilized for evaluating the performance of the model.

2.4.1 Overfitting and Underfitting

When a machine learning model learns or memorizes patterns that only exist in a tiny percentage of the data, this is known as overfitting. In this way, a model learns

so much detail and noise in the training dataset. As a result, it has a negative effect on the performance on a dataset that the model has not seen in training process. To avoid overfitting, cross validation, data augmentation, coefficient regularization, early stopping, and simpler models can be used. In contrast, underfitting occurs when a model fails to capture important patterns in the training data. Hence, the model performance is low in underfit models. Underfitting can be avoided by using more training data, larger and more complicated models, and longer training times.

Models that are underfit or overfit can not be effectively generalized, and so are unable to make accurate predictions for cases outside of the training dataset [46]. The illustrations of underfitting, good compromise and overfitting are given in Figure 2.5.

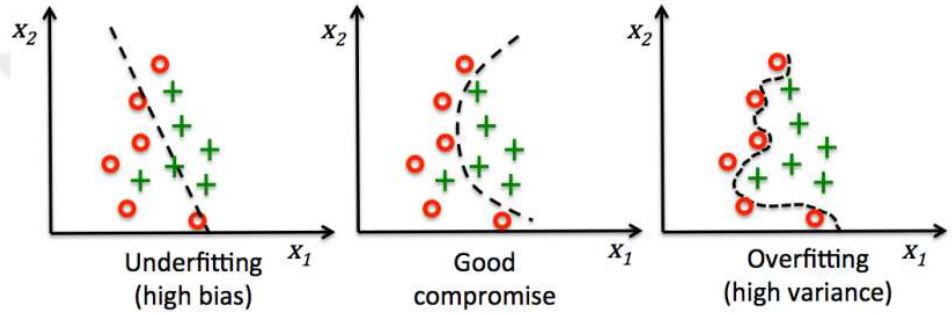

Figure 2.5 Underfitting, good compromise, and overfitting [47]

2.4.2 Cross Validation

Cross-Validation (CV) is a statistical method for testing the effectiveness of a machine learning classification model. To evaluate the effectiveness of any model, firstly we should test it on some unseen data. So this method is based on the idea of partition of the original data into smaller subsets. Firstly the data set is divided into k subsets. Each iteration, one of the k subsets is used as the test data set and the remaining k-1 subsets are used as a training data set. When the training set is used to fit model, the test set is used as a validation set to provide an unbiased evaluation of the chosen model. In other words, the test set is used to evaluate the performance of the model used on the training set. At the end of this process, every k-fold is used as the test data set. The accuracy of detection are calculated

for each test and average error is obtained. As can be seen in Figure 2.6, E_i represents the cross-validation error of the ith group. The average error value E of the k groups is calculated as an estimate of the model accuracy and is used as a performance indicator for the k-fold cross-validation model.

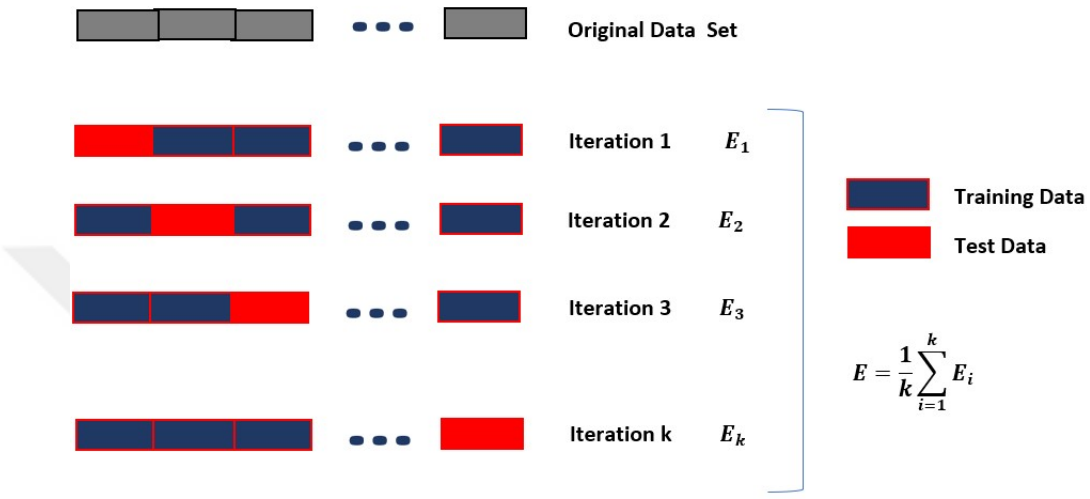

Figure 2.6 k-fold cross validation method

2.4.3 Performance Evaluation Metrics

The performance of each architecture is evaluated based on certain criteria. The results show whether the constructed system will be successful or not. The main criteria most frequently used in the field of machine learning consist of values such as confusion matrix, accuracy, precision, recall, and F1 value. These measures are detailed below.

The confusion matrix is utilized to perfectly assess a classifier's potential. Since by using confusion matrix, the relations between the classifier outputs and the true ones can be easily seen, it is one of the most widely used measures in classification problems. In Table 2.1, an example of a confusion matrix containing the predicted and actual values produced by the model for a binary classification problem is given.

Table 2.1 Confusion matrix for binary classification

		Predicted Value	
Actual Value		Yes	No
	Yes	TN	FP
	No	FN	TP

The four possible results in the confusion matrix are defined as the following:

- TP (True Positive): TP represents the number of positive examples classified accuractely.
- TN (True Negative): TN represents the number of negative examples accurately.
- FP (False Positive): FP represents the number of actual negative examples classified as positive.
- FN (False Negative): FN represents the number of actual positive examples classified as negative.

Once the confusion matris is constituted, the performance of classifiers can be compared by utilizing various parameters. Accuracy, precision, recall, and F1 score are other performance metrics that are calculated using the above mentioned TP, TN, FP, and FN. These parameters are determined as follows:

Accuracy is one of the most commonly utilized criteria while performing classification task. The accuracy score of an algorithm measures the ratio to the correctly classified examples of the total number of examples. Using the confusion matrix, the accuracy of a model is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

The other performance metris that can be determined directly from the confusion matrix is precision score. Precision score of a model is defined as the ratio of correctly classified positive examples to the total positive predicted examples.

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

Recall score is defined as the number of correctly classified positive examples divided by the number of positive examples.

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

Individual precision and recall are hardly helpful; these are imperfect views of the success of the classifier. Both precision and recall cannot differentiate classifiers that perform well from those forms of classifiers that perform poorly. A simplistic classifier may easily achieve a great recall score by predicting positive for each instance [48].

F1 score is defined as the harmonic mean of the precision score and the recall score. Higher F1 score means a better model.

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \tag{2.4}$$

$$F1 = \frac{2 * (precision * recall)}{precision + recall} \tag{2.5}$$

2.5 Classification Algorithms

Classification problem deals with predicting the class of the samples. In this work, we used the following machine learning methods for the classification task of VOC vapors.

2.5.1 K-Nearest Neighbor (KNN)

K-nearest neighbor (KNN) is one of the simplest and most widely used non-linear algorithms for classification developed by Fix and Hodges (1951) and later extended by Thomas Cover for the use of regression (Fix & Hodges, 1951; Altman, 1992). As it can be understood by its name, it uses k nearest labeled data points to predict the class of the unlabeled data point. Basically, it calculates the distance between a new data point and the other data points in training set as shown in Figure 2.7 with a simple example. Here, there are two important parameters that affect the performance: the value of k and the distance metric used. If the value of k is too large, classes with a great number of classified samples may overwhelm

small ones, and the results will be biased. On the other hand, if the value of k is choosen too small, the advantage of using a large number of samples in the training set may be lost. In many applications, k is usually set to 5 or 10. The other important parameter is the choice of the distance function. To measure the distance between two data points, various distance functions are used. Among the distance functions, the Euclidean distance function is the most widely used one. In addition, dependent on the distribution of the data set, Mahalanobis, Minkowski, Chebychev, Cosine, Hamming, and Spearman are other commonly used distance metrics. The formulas for these distance functions are given in Table 2.2. In Figure 2.7, an example of KNN classification with k neighbors k=3 and k=5 is shown. For k=3, the test data (green circle) is assigned to the rectangular class. But for k=5, it is assigned to the square class.

Table 2.2 Distance functions

Distance Metric	**Definition**
Euclidean Distance	$d(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$
Minkowski Distance	$d(x,y)=\left(\sum_{i=1}^{n}\lvert x_i-y_i\rvert^p\right)^{1/p}$
Manhattan Distance	$d(x,y)=\sum_{i=1}^{n}\lvert x_i-y_i\rvert$
Cosine Distance	$\cos\theta=\frac{\vec{a}\cdot\vec{b}}{\lvert\lvert\vec{a}\rvert\rvert.\lvert\lvert\vec{b}\rvert\rvert}$

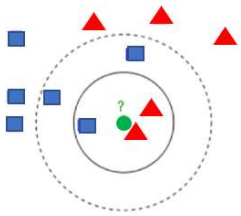

Figure 2.7 An example of KNN classification with k neighbors k=3 (solid circle) and k=5 (dashed line circle)

2.5.2 Support Vector Machine (SVM)

Support vector machines, which can be utilized for both classification and regression problems, were developed by Vladimir Vapnik and Corrina Cortes [49] in 1995. Support vector machines are utilized in various applications, ranging from hand-written digit recognition ([Cort 95]), to object recognition ([Blan 96]), person identification ([Ben 99]), spam categorization ([Druc 99]), and channel equalization ([Seba 001). The results from these applications demonstrate that SVM classifiers exhibit enhanced generalization performance, which seems to be the power of this classifier [50].

The main idea behind the algorithm is to find the best line or decision boundary for categorizing n-dimensional space into classes so that a new data point can be put in the appropriate category. The optimal boundary is called a hyperplane and the data points nearest to the hyperplane are called support vectors. The algorithm basically tries to find the optimum position of the hyperplane according to the support vectors. The maximum distance between data points is called margin and the aim is to find a plane that has the maximum margin. As a result, the model selects the hyperplane that maximizes the margin. As shown in Figure 2.8 and 2.9, depending on the number of classes, the hyperplane type changes as linear or nonlinear.

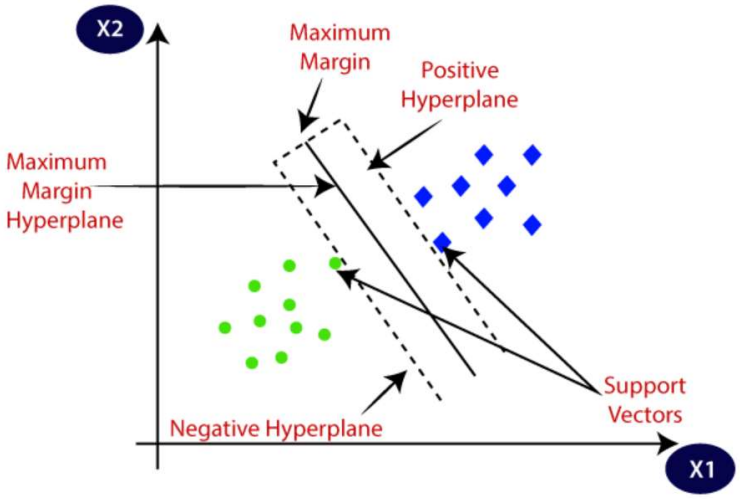

Figure 2.8 Classification with Linear SVM [51]

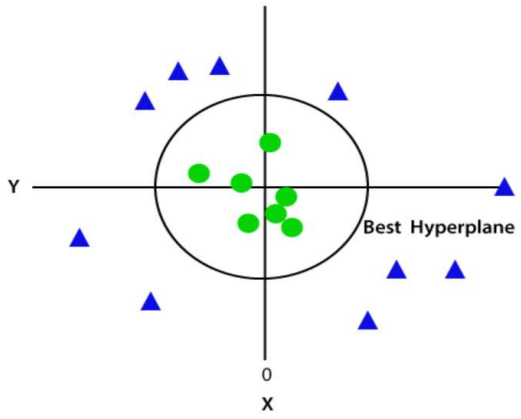

Figure 2.9 Classification with Non-linear SVM [51]

The advantages of this algorithm as follows: [Akarsu, 2016]

- This classifier is able to model complex decision boundaries.
- It can work with a large number of independent variables.
- It can be applied to linearly separated data as well as to data that cannot be linearly separated.
- The problem of overfitting is less than many other methods.

2.5.3 Decision Tree

Decision trees are structures that are commonly used for classification. Decision tree algorithm displays the classifier's information set in a more understandable manner, and displays the class alternatives and probability circumstances in a tree structure by organizing them in a specific order. Because of its simple analysis and

its precision on multiple data forms, decision tree algorithm has utilized in many implementation fields [52].

As it is obvious from its name, decision tree model is a tree-structured classifier consists of basically three parts: node, branch and leaf. Figure 2.10 illustrates a structure of a decision tree model. The upper part is called the root node and the lower part is called the leaf node. The division between root and leaf nodes is referred to as branches. The decision tree consists of paths starting from the root node and continuing according to the yes-no answers applied to the experimental units. (Han & Kamber, 2000; Lewis, 2000; Pehlivan, 2006)

In decision trees, the depth in the mountain structure is as important as the nodes, branches and leaves. Each branch in the tree structure is a function that creates a rule. If the classification is not completed during the new branch creation, a new decision node is created. The number of decision nodes formed as a result of branching is expressed in depth. The number of depths depends on the size and homogeneity of the data set (Carvalho & Freitas, 2004).

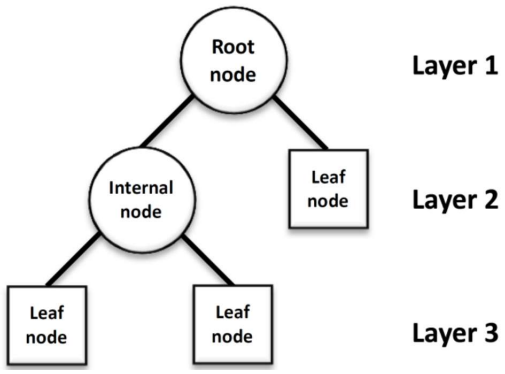

Figure 2.10 Decision Tree Structure [53]

2.5.4 Ensemble Methods

Ensemble is a word that refers to a community or a group. In machine learning terminology, ensemble learning is similar to that definition. To get a better prediction result for a classification problem, ensemble models use different learning methods. Ensemble methods combine the decisions from several base machine learning models to increase the benefit and the efficiency of a machine learning system used alone. Since ensemble learning involves multiple learning

algorithms model prediction and validation takes longer than a single learning algorithm.

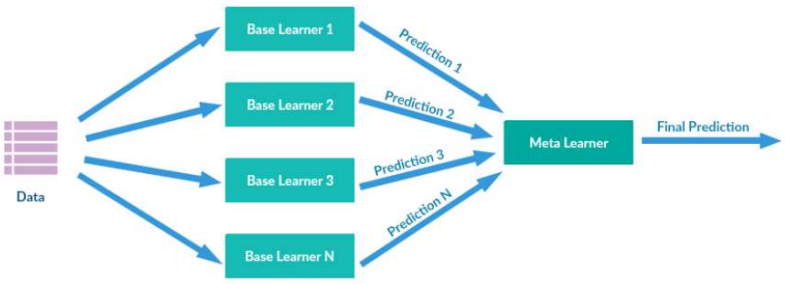

Figure 2.11 Ensemble Learning [54]

As it can be seen from Figure 2.11, many different learners are trained on random parts of the data and all learners are capable of making predictions. They communicate their predictions to a meta-learner, which creates the final prediction based on the base learners' predictions. [54].

Bagging and boosting are two most common types of ensemble learning. Bagging (Breiman, 1996), also known as bootstrap aggregating, developes a base model for each by dividing the initial dataset into subsets of the same size, and the models operate independently. The final prediction is obtained by integrating the results of all models that have been created. Another strategy is Boosting (Duffy and Helmbold, 2002), in which each subsequent model seeks to fix the mistakes of the prior model through producing subsets of the dataset. The final model is the average of all the previous models [55]. The algorithms using boosting techniques are GradientBoost, AdaBoost, and XGBoost.

3
EXPERIMENTAL PROCEDURE

3.1 Experiment Setup

Within the scope of the thesis, studies on the production of sensors and determination of gas detection performance were carried out in the laboratory of the Physics Department of the Faculty of Arts and Sciences of Y.T.U. The list of the equipment is given below.

- Thermal evaporator (to be used in the production of interdigital electrodes)
- Spin Coating Device (to be used for coating the sensor unit)
- Electrometer (to be used to measure sensor currents)
- Gas flow meters (4 pieces) (to be used to provide controlled gas flow from the sensor surface.)
- Temperature controller (to be used to control the temperature of the sensors)
- Impedance Analyzer (to be used in the characterization of sensor units)

3.2 VOC Sensing Measurement

Spin coated thin film of 5'-6'-Bis(17',25',32'-trinitro-phthalocyaninyl) (1,4,7,10-tetrathia-12-crown)dicobalt(II) on interdigital array (IDAT) of Au electrodes was used as sensing element for our VOC sensing measurement task. The synthesis details of the sensing layer, illustrated in Figure 3.1, were described in [56]. After the spin coating of the sensory layer, coated IDA was installed in a home-made detection cell with a capacity of 5×10^{-4} liters. During the VOC sensing experiments, the carrier gas was dry nitrogen with a purity of 99.8% and the desired level of relative humidity was obtained by bubbling the carrier gas through liquid VOC. The experimental setup used during VOC vapor sensing studies is shown in Figure 3.2. For studying the sensing behaviour of organic vapours with

5'-6'-Bis(17',25',32'-trinitro-phthalocyaninyl)(1,4,7,10-tetrathia-12-crown)dicobalt(II) (CoPc) thin film, it was exposed to six different VOC vapours (methanol, ethanol, butanol, isopropyl alcohol, acetone, and ammonia) and the variations in the sensor current with time were recorded under the applied constant voltage of 0.5 V.

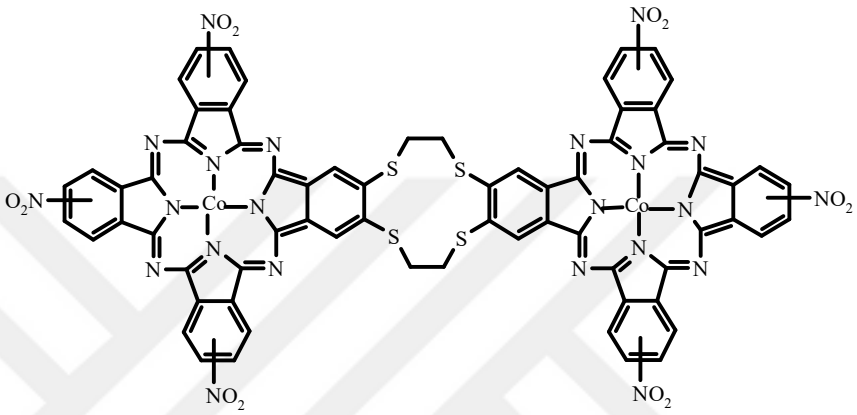

Figure 3.1 Structural formula for 5'-6'-Bis(17',25',32'-trinitro-phthalocyaninyl) (1,4,7,10-tetrathia-12-crown)dicobalt(II)

Well-defined concentrations of VOC vapors were prepared by mixing the carrier gas with the target vapors. The concentration of the target vapors was varied from 50 to 450 ppm by using mass flow controllers (Alicat Scientific Inc.). In a typical sensing experiment, the sensor surface was exposed to VOC vapor for 20 min. and then purged with carrier gas for another 20 min. to reset the baseline (Figure 3.2). Total flow rate of the carrier gas was adjusted as 100 standard cubic centimetre (sccm) during the purging experiments. All sensing experiments were performed at a cell temperature of ~ 28 °C.

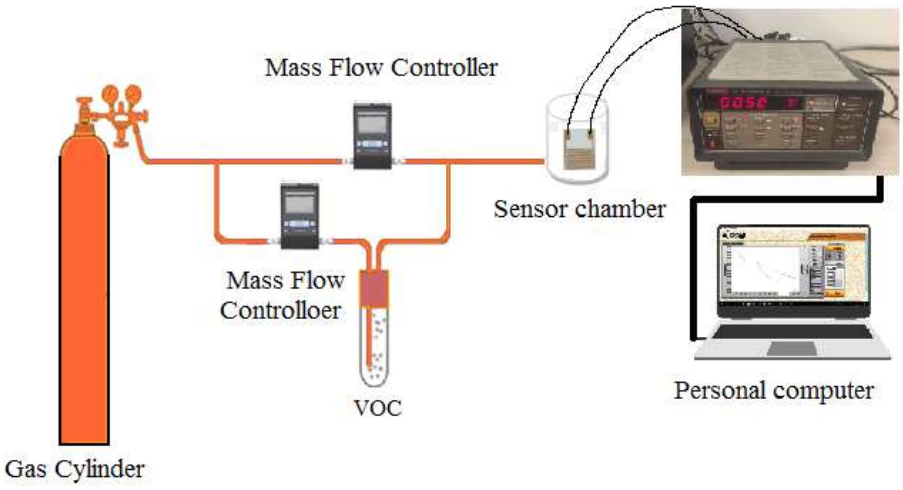

Figure 3.2 Schematic of the VOC vapor sensing configuration

3.3 Sensing Results

The response recovery characteristic of CoPc thin film with exposure of six organic vapors methanol, ethanol, isopropyl alcohol, butanol, acetone, and ammonia is illustrated in Figure 3.3 at different concentrations as labeled in this figure. We observed that CoPc film exhibits an increase in current after the exposure of all the VOC vapors even during ammonia exposure. We also observed during sensing studies of these six organic vapors that after nine cycles of exposure-purging stages, the sensor response recovers to 90% of its initial value. This finding indicates that the interaction between CoPc and VOC vapors is reversible. The maximum increase in sensor current has been observed for methanol vapours, followed by acetone for all concentrations of investigated VOC vapors.

The sensitivities (S) of the CoPc thin film towards VOC vapors have been calculated from the measured dynamic characteristics of the sensor using the following equation

$$S = \frac{1}{C_V}\frac{\Delta I}{I_0} \tag{3.1}$$

where C_v is the concentration of the VOC vapor under investigation, ΔI is the change in sensor current and I_0 is the baseline current of CoPc thin film before the

exposure of the VOC vapours. As is clear from Figure 3.3, for the same concentrations of VOC vapors, maximum sensitivity (0.0232 ppm^{-1}) has been observed for methanol vapor, while minimum sensitivity (0.0024 ppm^{-1}) is observed for isopropylalcohol vapor. It should be mentioned here that the sensitivity values towards methanol is nearly the same for acetone vapor. The similar trend of sensitivity has been observed for ethanol (0.0083 ppm^{-1}) and butanol (0.0077 ppm^{-1}), and for isopropylalcohol (0.0024 ppm^{-1}) and ammonia (0.003 ppm^{-1}) vapors, respectively. This reveals that the interaction between the VOC vapors and the CoPc surface is not selective.

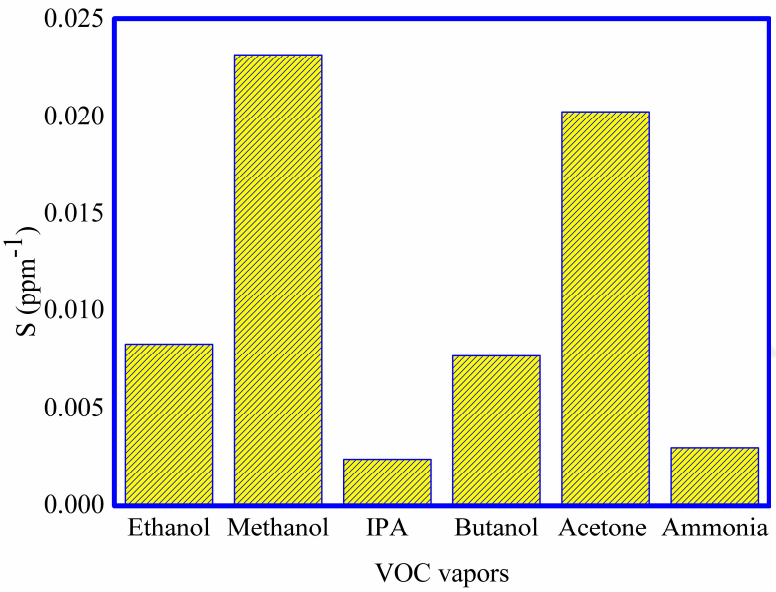

Figure 3.3 Sensitivities of CoPc thin film towards VOC vapors investigated

As it can be understood by Figure 3.3, the present sensor is poorly selective and it is not appropriate to use it in this form. Since it is not possible to provide a real identification among considered vapors with traditional methods, we utilized machine learning approach in order to give the system a way to identify the vapors by itself, as explained in the next section.

4
MATERIALS AND METHODS

This section covers the details of how algorithms in the field of machine learning are applied to the gas sensor data obtained from the experiment mentioned in the previous section. The aim of the machine learning approach used in this gas sensor application is to find a relationship between the sensor responses and the type of gas. In order to utilize the possibility of determining the gas type using machine learning algorithms, the collected data have been treated in a systematic way in this section. As illustrated in Figure 4.1, the process can simply be divided into four phases: gas sensor data acquisition, data preprocessing, feature extraction, and finally classification. Within the scope of this system, it is explained how the chosen models were constructed, the parameters that were used, and the training and testing procedures in this section. The outcomes of the related training and testing operations, and comparisons are also shown. Finally, the conclusion of our work is given.

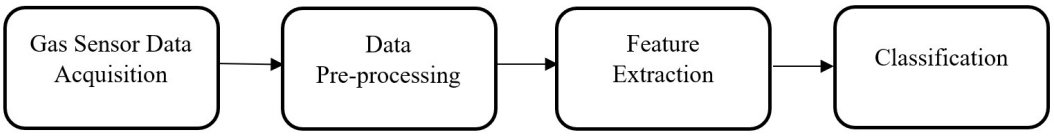

Figure 4.1 Data analysis and classification system of the gas sensor data

4.1 Dataset

According to the experiment mentioned in the previous chapter, each gas was injected into the sensor during 2400 seconds period, while 1200 seconds were allocated to a cleaning stage with dry air. Data was collected at a sampling period of 2s. This process was repeated for nine different concentration cycles for each gas separately. Thus, the high dimensional gas sensor response dataset contains 13910 data collected by one chemical gas sensor for nine different concentrations

of six different VOC vapors (including methanol, ethanol, acetone, ammonia, butanol, and isopropyl alcohol). Note that, in this work, the data obtained during the recovery time was ignored. In other words, we were particularly interested in the increasing part of the response curve. Figure 4.2 shows the typical response curve of the sensor when it is exposed to a gas.

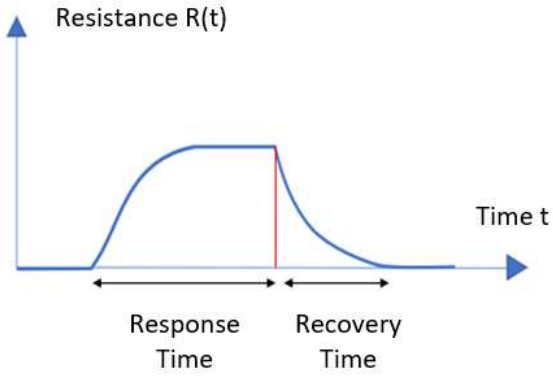

Figure 4.2 A typical response of a chemical gas sensor

In Figure 4.3, we show the responses of the sensor during the acquisition procedure for six VOCs. As there are nine concentrations ranging from 50 to 450 ppm, nine concentration cycles were observed for each gas.

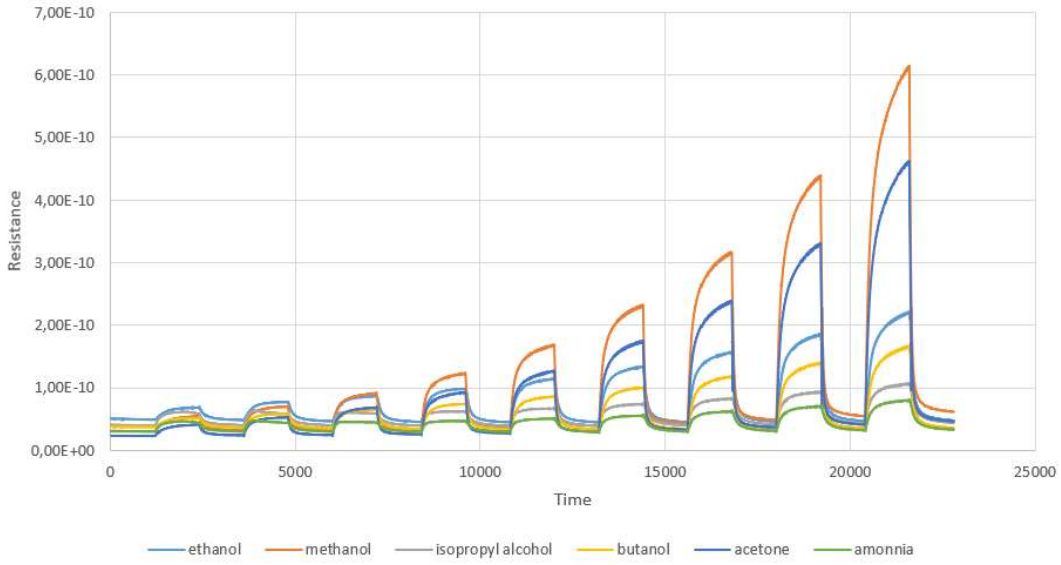

Figure 4.3 Graphical representation of gas sensor response data for six VOC vapors used in the experiment

4.2 Data Preprocessing

The collected data should be preprocessed before the feature extraction step. Since data processing in a machine learning system directly determines the input, it is an indispensable step before training a model. Various preprocessing techniques applied to gas sensor data have been proposed in the literature [57]. There are no general guidelines to determine the appropriate data preprocessing technique, so the technique to be used varies from application to application. Among different techniques, in this research, in order to reduce the effect of the baseline on the data and ensure the reliability of the data, the baseline subtraction method was applied to the raw data first. In order to eliminate the impact of the baseline on the data, baseline values were subtracted from sensor responses. This method, which is commonly used in temporal data collecting, eliminates a sensor's base reading [58].

The method is as shown in Equation (4.1), where $R(t)$ value is the dynamic response value, $R(0)$ is the baseline value which is the minimum sensor response value when exposed to a reference gas. That is, the preprocessed data is equal to the difference between the response value and the baseline value.

$$Y_S = R(t) - R(0) \tag{4.1}$$

4.3 Feature Extraction

In our case, the gas sensor data we collect is in large amounts, like in many other real life applications. So, we need a process to understand this data and transform the raw data to best meet the expectations of the algorithms. Since it is not possible to process them manually, this is where the idea of feature extraction comes into play.

In order to find characteristics of the obtained raw data, first some features have to be extracted from the original gas sensor response dataset, and then the relationship between the selected features and the gas type has to be examined by a learning procedure. In other words, the key point of gas identification is the feature extraction method, by which the robust information from the sensor

response curves is extracted with less redundancy, for the success of the subsequent classification algorithm. Since the selected features directly affect the performance of the classifier, this step has great importance.

In this study, among various feature extraction methods used in gas sensor applications, extracting piecemeal response features obtained from the original curve of the sensor has been discussed. We propose here the study of several features obtained from both steady-state sensor responses: maximum value and responses of special time interval of steady state region, and the transient part of the sensor response: derivative and integral. The features and their descriptions are listed in Table 4.1.

As the objective of this research is to detect target gases based on a selected portion of the response curve, different data sets were generated, including different features for the classification task. Each dataset contains extracted features of all vapors according to the selected feature. The graphical representations of the four features are illustrated in Figure 4.4.

Table 4.1 Features extracted from gas response curves

Number of Dataset	**Feature**	**Description**
1	Max Response	Max(sensor value): R_{max}
2	10 s responses	10 s response data from the response curve after the sensor reaches 90% of its maximum value: $R^1 \ldots\ldots R^{10}$
3	Derivative	Rising slope: Difference between two samples during measurement $D(t) = \frac{dR(t)}{dt} = \frac{R_{max} - R_o}{t_{max} - t_o}$ R_{max}: maximum sensor response R_o : baseline value t_o : is the time from the beginning of the adsorption stage to peak value t_{max}: time when sensor reaches its maximum

Table 4.1 Features extracted from gas response curves (continued)

4	Integral	Area Under The Increasing Curve: Area value of sensor response curve and time axis surrounded; the area under the increasing curve $I(t) = \int_{t_1}^{t_2} R(s)ds$ s : time from t_1 to t_2 t_1 : the begining of the transient phase t_2 : the end of the transient phase

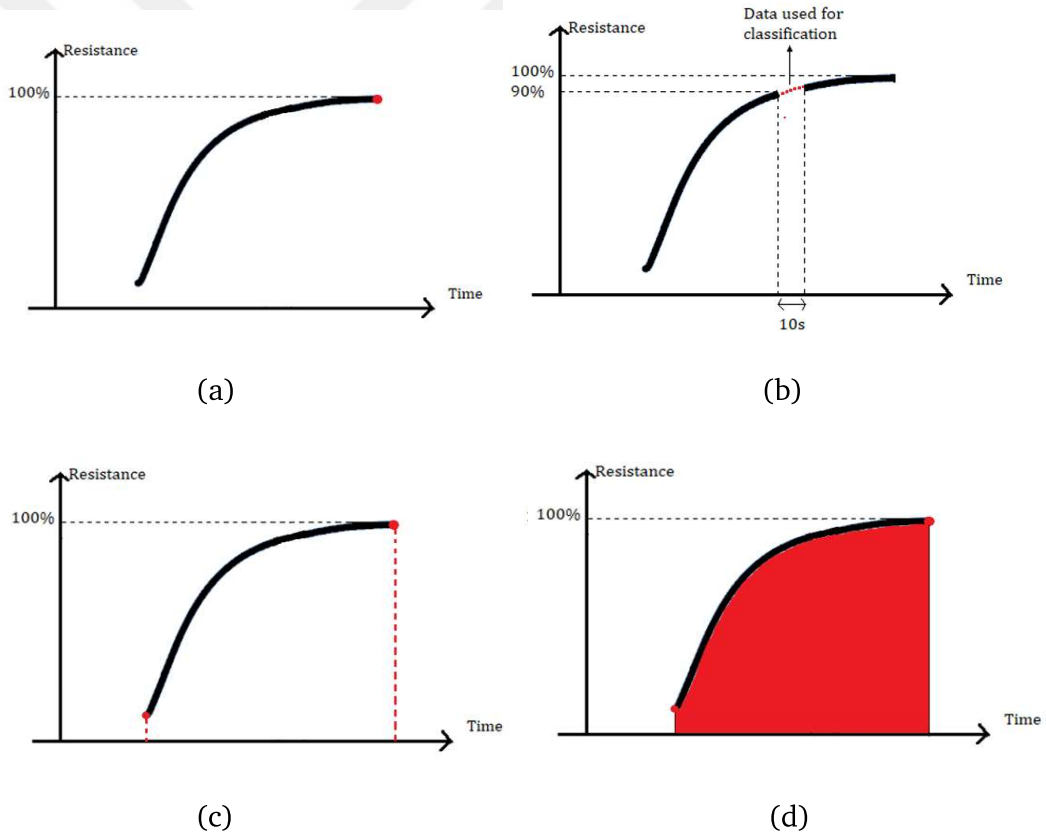

Figure 4.4 The graphical representations of the extracted features: (a) Max response (b) 10 s data (c) Derivative (d) Integral

Extracted features give us information about different aspects from the original reactions between the sensor and the injected gas. As illustrated in Figure 4.4, from each curve, four different features are extracted.

The maximum response value is the first feature to be used for creating a sub-data set from the original response curve. It is frequently chosen as the most common and basic E-nose feature because it represents the ultimate steady-state feature of the complete dynamic response process in the final balance, which indicates the maximum reaction degree change of sensors responding to vapors [33].

Besides, in many applications for chemical sensors, information can be gained not only from the steady state value of the sensor signal, but also from the kinetics response. Using steady state sensor values to categorize different mixed gases, on the other hand, results in the sensor signal losing a lot of information [20]. Hence, as a second feature, it is thought that not only one response but also a few seconds of responses from steady state region could be used in order to utilize more information from the response curve. The 10 s data extracted from the response curve after the sensor reaches 90% of its maximum value was directly used without taking into account the other information in the whole response curve. So, as a second feature we used the data obtained at 10 seconds intervals for creating second sub-data set.

The integrals and derivatives of signals also have specific physical meanings. They both reflect different information of the reaction kinetics at different aspects. Integrals reflect the total degree change of the reaction, and derivatives can represent the rate of the reaction [59]. Many studies have revealed that integral and derivative approaches have greater identification rates than maximum values, and that the desorption stage of the original gas sensor response curves is more informative than the adsorption stage [60].

Note that, in this study, the main feature is the few seconds of responses from steady state region during 10 seconds and the others given in Table 4.1 are also employed to compare different feature data sets in detection performance.

In the next section, the performance of each created subdata set will be tested separately by the classification algorithms. The process of finding the most proper feature is given in Figure 4.5.

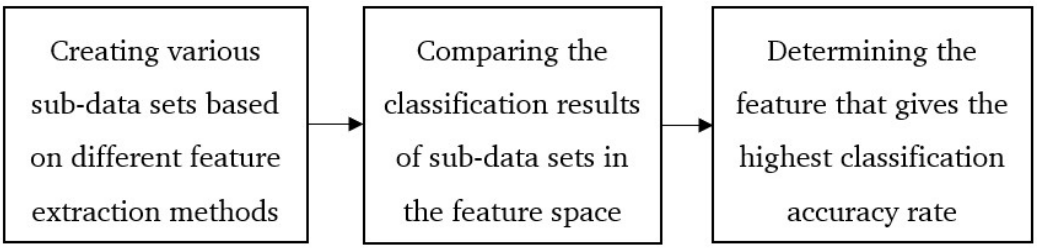

Figure 4.5 The process of finding the most proper feature

5
RESULTS AND DISCUSSION

5.1 Performance of the Classifiers on Each Feature Set

Once we have extracted the features, we need to choose a proper method for the classification task. Classification step addresses the problem of finding a relation between the sensor responses and the gas types. In this study, five different machine learning algorithms were used while predicting the gas type. These are Decision Tree, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Ensemble Method.

Each dataset, formed according to the mentioned feature in the previous part, is randomly split into 60% training and 40% test sets. For example, as for the feature number 2, the training set contains 192 samples for six gas types and the test set contains the remaining 133 samples which were not used during the training process. In this study, classifiers were trained by using the same training data, and then classifiers were tested by using the same test data.

Due to the small amount of data, k-fold cross validation method was used to avoid overfitting. According to the method, the classifier is trained with $k-1$ splits and validated on the missing split. The method is repeated k times until all of the data are used for validation. In our work, we have used 5-fold cross validation. In order to compare the performance of classification algorithms, classification accuracy was utilized as a performance metric. It is obtained by dividing the number of correctly recognized samples into the total number of samples. Furthermore, since the training set and test set were randomly selected from the original gas sensor dataset, we performed train-test procedure 10 times to avoid bias in the classification process. Then, the final classification accuracy of each classifier was calculated by averaging of ten iterations. Each classification algorithm was learned from the same training set and tested its classification accuracy on the same test data. Hence, the best classification method is clearly the one with the highest accuracy. The classifiers used in this work are given in Table 5.1.

Table 5.1 Classification algorithms used in this study

Decision Tree	K Nearest Neighbor	Support Vector Machine	Ensemble Learning
-Fine Tree -Medium Tree -Coarse Tree	-Fine KNN -Medium KNN -Coarse KNN -Cosine KNN -Cubic KNN -Weighted KNN	-Linear SVM -Quadratic SVM -Cubic SVM -Fine Gaussian SVM -Medium Gaussian SVM -Coarse Gasussian SVM	-Boosted Trees -Bagged Trees -RUSBoosted Trees -Subspace Discriminant -Subspace KNN

In this work, we have used classification learner in Matlab program to create and train the models of prediction options of Decision Tree, K Nearest Neighbor, Support Vector Machine, and Ensemble Learning. Classification accuracy was considered as a performance criterion. Classification results of each model are detailed below.

5.1.1 Classification Results of Decision Tree

In this work, three different types of decision tree classifiers are used as classifiers. The utilized decision tree classifiers are explained in Table 5.2, and the classification accuracy results are given in Table 5.3. The number of splits are 100, 20 and 4 respectively for fine tree, medium tree and coarse tree. Each of the algorithms is trained using Gini's diversity index as a split criterion. As it can be seen from Table 5.3, classification accuracy changes when the number of splits is changed. It was observed that all decision tree classifiers had poor performance. The model with the highest accuracy is the Fine Tree model with 100 splits for the 10 s response values from steady state region feature.

Table 5.2 Hyperparameters of Decision Tree Models

Decision Tree Model	Hyperparameters
Fine Tree	**Maximum number of splits:** 100 **Surrogate decision splits:** Off **Split Criterion:** Gini's diversity index
Medium Tree	**Maximum number of splits:** 20 **Surrogate decision splits:** Off **Split Criterion:** Gini's diversity index
Coarse Tree	**Maximum number of splits:** 4 **Surrogate decision splits:** Off **Split Criterion:** Gini's diversity index

Table 5.3 Classification accuracy (%) results of Tree Models

Model / Feature	Fine Tree	Medium Tree	Coarse Tree
Max response value	21.24	21.24	25.41
10 s responses	61.58	50.08	38.18
Integral	17.79	17.79	21.12
Derivative	36.66	36.66	37.76

5.1.2 Classification Results of KNN

All the KNN techniques available in the MATLAB software were used to classify our data sets. The main differences between the techniques are the used metrics and the chosen number of k values. Table 5.4 shows the hyperparameters of the classifiers, and Table 5.5 shows the classification accuracy of each model on each

of the feature sets. The results with maximum accuracy in the classification are obtained when considering the Weighted KNN and the Fine KNN. They give the best result with an accuracy rate of 96 percent. The obtained classification results point out that the 10 s response values as a feature gives the best accuracy rate.

Table 5.4 Hyperparameters of KNN Models

KNN Model	Hyperparameters
Fine Knn	**Number of neighbors:**1 **Distance metric:** Euclidean **Distance weight:** Equal **Standardize data:** true
Medium Knn	**Number of neighbors:** 10 **Distance metric:** Euclidean **Distance weight:** Equal **Standardize data:** true
Coarse Knn	**Number of neighbors:** 100 **Distance metric:** Euclidean **Distance weight:** Equal **Standardize data:** true
Cosine Knn	**Number of neighbors:** 10 **Distance metric:** Cosine **Distance weight:** Equal **Standardize data:** true
Cubic Knn	**Number of neighbors:** 10 **Distance metric:** Minkowski (cubic) **Distance weight:** Equal **Standardize data:** true
Weighted Knn	**Number of neighbors:** 10 **Distance metric:** Euclidean **Distance weight:** Squared inverse **Standardize data:** true

Table 5.5 Classification accuracy (%) results of KNN Models

Model / Feature	Fine Knn	Medium Knn	Coarse Knn	Cosine Knn	Cubic Knn	Weighted Knn
Max response value	8.33	22.91	16.7	20.42	22.91	9.17
10 s responses	96.75	34.1	31.75	16.7	34.1	96.28
Integral	9.48	17.22	33.3	26.66	17.22	8.91
Derivative	19.45	33.32	16.7	20.01	35.54	18.89

The performance of Fine KNN model with confusion matrices for train and test data is illustrated in Figure 5.1 and Figure 5.2. For Fine k-NN classifier, 178 cases have been correctly classified out of 192 cases in train procedure. In test procedure, 129 cases have been correctly classified out of 133 cases.

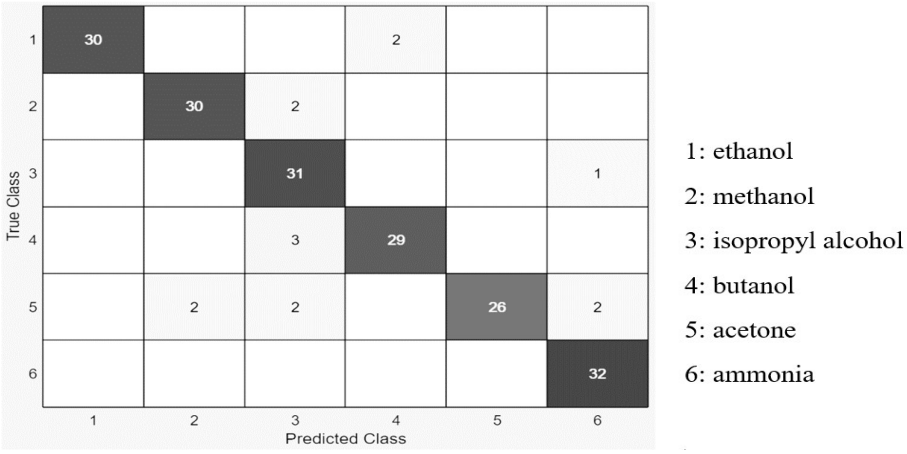

Figure 5.1 Confusion matrix using Fine KNN for train data when the 10 s response values are considered as a feature

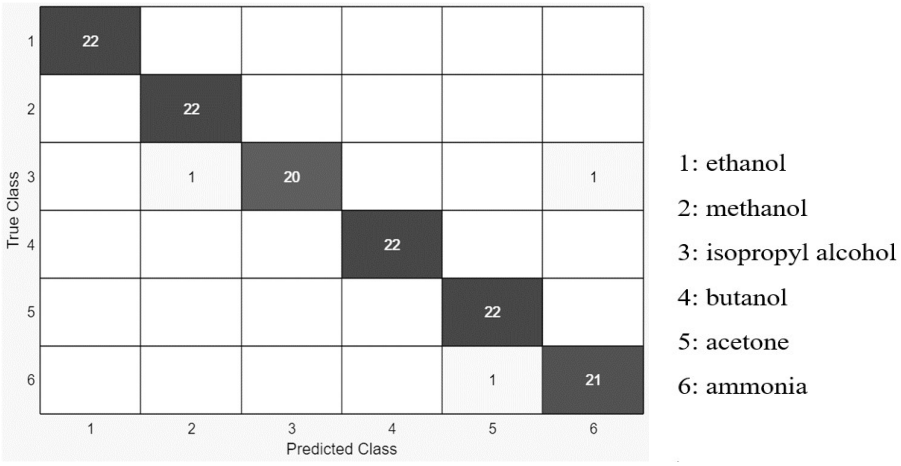

Figure 5.2 Confusion matrix using Fine KNN for test data when the 10 s response values are considered as a feature

Besides, in order to obtain the best classification accuracy with KNN algorithm, the optimum k value should be found. For this purpose, the above mentioned KNN-1 and KNN-2 algorithms were run with different k values from 1 to 10. Since KNN-1 model is based on weighting, the value of k does not affect the accuracy of the algorithm. Classification accuracy results for different k values for KNN-2 model, which is based on euclidean metric without weightening, are illustrated in Figure 5.3. As the number of k values increases, the accuracy value decreases.

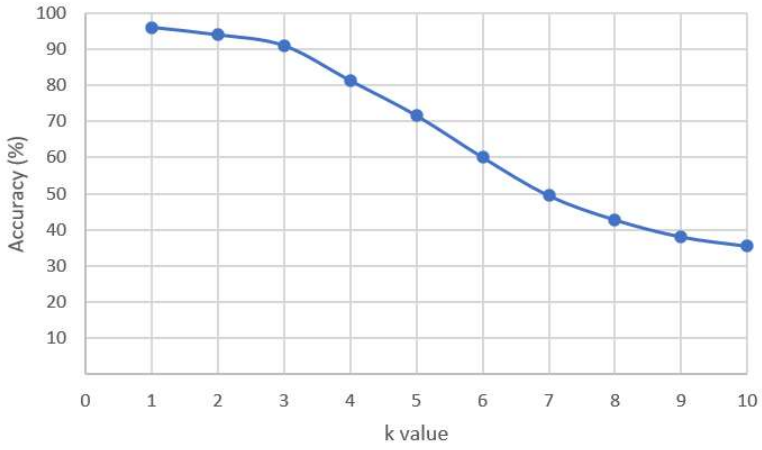

Figure 5.3 The accuracy of KNN model for different k

5.1.3 Classification Results of SVM

As it can be seen from Table 5.6, the SVM models are trained for all the six techniques available in MATLAB for implementing the gas classification. The

performance of these algorithms is evaluated through their accuracy, reported in Table 5.7. We see here that no single feature is enough to classify all the classes with SVM classifiers.

Table 5.6 Hyperparameters of SVM Models

SVM Model	Hyperparameters
Linear SVM	**Kernel Function:** Linear **Kernel Scale:** Automatic **Box Constraint Level:** 1 **Multiclass Method:** One-vs-One **Standardize data:** true
Quadratic SVM	**Kernel Function:** Quadratic **Kernel Scale:** Automatic **Box Constraint Level:** 1 **Multiclass Method:** One-vs-One **Standardize data:** true
Cubic SVM	**Kernel Function:** Cubic **Kernel Scale:** Automatic **Box Constraint Level:** 1 **Multiclass Method:** One-vs-One **Standardize data:** true
Fine Gaussian SVM	**Kernel Function:** Gaussian **Kernel Scale:** 0.25 **Box Constraint Level:** 1 **Multiclass Method:** One-vs-One **Standardize data:** true
Medium Gaussian SVM	**Kernel Function:** Gaussian **Kernel Scale:** 1 **Box Constraint Level:** 1 **Multiclass Method:** One-vs-One **Standardize data:** true

Table 5.6 Hyperparameters of SVM Models (continued)

Coarse Gasussian SVM	**Kernel Function:** Gaussian **Kernel Scale:** 4 **Box Constraint Level:** 1 **Multiclass Method:** One-vs-One **Standardize data:** true

Table 5.7 Classification accuracy (%) results of SVM Models

Model / **Feature**	**Linear SVM**	**Quadratic SVM**	**Cubic SVM**	**Fine Gaussian SVM**	**Medium Gaussian SVM**	**Coarse Gasussian SVM**
Max response value	24.16	22.08	24.17	17.08	20.82	22.91
10 s responses	28.93	30.67	27.65	44.48	32.88	32.49
Integral	32.76	24.45	17.8	22.21	29.43	33.3
Derivative	39.98	38.87	35.54	26.67	41.1	36.65

5.1.4 Classification Results of Ensemble Learning

By using the ensemble models given in Table 5.8, the classification results for all data sets are shown in Table 5.9. When the classification accuracy values are examined, it is seen that the proposed 10 s response values from the steady state region feature yields the most successful results with bagged trees and subspace knn method. These classifiers give us high accuracy of more than 95 percent.

Table 5.8 Hyperparameters of Ensemble Models

Ensemble Model	**Hyperparameters**
Boosted Trees	**Ensemble Method:** AdaBoost **Learner Type:** Decision Tree

Table 5.8 Hyperparameters of Ensemble Models (continued)

	Maximum number of splits: 20 **Number of learners:** 30 **Learning Rate:** 0.1
Bagged Trees	**Ensemble Method:** Bag **Learner Type:** Decision Tree **Maximum number of splits:** 35 **Number of learners:** 30
RUSBoosted Trees	**Ensemble Method:** RUSBoost **Learner Type:** Decision Tree **Maximum number of splits:** 20 **Number of learners:** 30 **Learning Rate:** 0.1
Subspace Discriminant	**Ensemble Method:** Subspace **Learner Type:** Discriminant **Number of learners:** 30 **Subspace Dimension:**1
Subspace KNN	**Ensemble Method:** Subspace **Learner Type:** KNN **Number of learners:** 30 **Subspace Dimension:**1

Table 5.9 Classification accuracy (%) results of Ensemble Models

Model / **Feature**	**Boosted Trees**	**Bagged Trees**	**RUSBoosted Trees**	**Subspace Discriminant**	**Subspace KNN**
Max response value	8.33	8.33	14.18	26.25	8.33

Table 5.9 Classification accuracy (%) results of Ensemble Methods (continued)

10 s responses	73.64	95.61	52.87	31.21	96.75
Integral	9.48	9.48	12.78	29.45	9.48
Derivative	19.45	20.01	23.33	29.44	19.45

5.2 Conclusion

In this work, the sensing performance of a resistive type gas sensor based on CoPc thin film was investigated with respect to six different VOC vapors (methanol, ethanol, butanol, isopropyl alcohol, acetone, and ammonia) concentrations from 50 to 450 ppm. As the present sensor can not achieve a sufficient level of selectivity in that form, in order to solve the issue of selectivity while using a single gas sensor, machine learning algorithms were applied to the gas sensor data set. It is shown that a desired selectivity level can be obtained by combining some part of the sensor response data with some classification algorithms based on a feature extracted from the original gas sensor response curve. By using 10 s response values from the steady state region as a feature, 95.6% classification accuracy with the bagged tree model and 96.7% classification accuracy with the knn model were reached for detection of six VOC vapors. The results point out that without any additional feature engineering, high accuracy rate can be obtained. Among various classification algorithms; KNN, which is one of the fast and easy to implement algorithms, achieved the highest accuracy on this classification problem, which is a notable achievement with a single resistive gas sensor. Besides, the performance of the feature proposed in this study was compared with the traditional features such as maximum response, integral and derivative, and it is shown that the proposed feature provides much better classification accuracy. Therefore, it is found that the most necessary information that helps to distinguish between the different types of VOC vapors can be gained using a few seconds of responses taken from steady state region. As KNN is easy to implement and works very fast, it is one advantage of the proposed method in this study. Besides, whereas in many

applications, a single feature can not fully reflect the characteristics of sensor responses, in this work using only one feature based on a single sensor, high classification was obtained. In this sense, it is shown that the desired selectivity of the sensor can be achieved through the application of machine learning to the responses from the gas sensor. In addition to this, though there are many researches based on acoustic wave sensors for the detection of the vapors investigated in the literature, the number of resistive based sensor studies we have introduced in this study is quite limited. As a result, our model could have great potential in practical applications for solving the issue of selectivity while using a single resistive gas sensor.

REFERENCES

[1] United States Environmental Protection Agency. [Online]. Available: http://www.epa.gov/iaq/voc.html (visited on 03/03/2022).

[2] World Health Organization. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health (visited on 03/03/2022).

[3] C. D. Natale, A. Macagnano, E. Martinelli, R. Paolesse, G. D'Arcangelo, C. Roscioni, A. Finazzi-Agrò, A. D'Amico, "Lung cancer identification by the analysis of breath by means of an array of non-selective gas sensors," Biosensors and Bioelectronics, vol. 18, no. 10, pp. 1209-1218, 2002.

[4] M. Phillips, "Breath test for dedection of various diseases", United States Patent, 6540691, 2003.

[5] R. Fend, C. Bessant, A. Williams, A. C. Woodman, "Monitoring haemodialysis using electronic nose and chemometrics," vol. 19, no. 12, pp. 1581-1590, 2003.

[6] B. Karlak, "Real time bad breath diagnosis and trasmission for telemedicine," IJSIT Lecture Note of International Conference on Intelligent Knowledge System, vol. 1, no. 1, 2004.

[7] T. Nakamato, A. Iguchi, T. Mariizumi, "Vapor supply method in odor sensing system and analysis of transient sensor responses," Sensors and Actuators B: Chemical, vol. 71, no. 3, pp. 155-160, 2000.

[8] E. Göl, E. Karabudak, "Mini-review: Ball type phthalocyanines: similarities and differences from mono phthalocyanines," Mini Reviews in Organic Chemistry, vol. 16, no. 5, pp. 410-421, 2019.

[9] K. E. van Keulen, M. E. Jansen, R. W. M. Schrauwen, J. J. Kolkman, P. D. Siersema, "Volatile organic compounds in breath can serve as a noninvasive diagnostic biomarker for the detection of advanced adenomas and colorectal cancer," Alimentary Pharmacology and Therapeutics, vol. 51, no. 3, pp. 334–346, 2020.

[10] K. M. Tripathi, T. Y. Kim, D. Losic, T. Thanh Tung, "Recent advances in engineered graphene and composites for detection of volatile organic compounds (VOCs) and non-invasive diseases diagnosis," Carbon, vol. 110, pp. 97-129, 2016.

[11] Y. Saalberg, M.Wolff, "VOC breath biomarkers in lung cancer," Clinica Chimica Acta, vol. 459, pp. 5–9, 2016.

[12] V. S. Bhati, M. Hojamberdiev, M. Kumar, "Enhanced sensing performance of ZnO nanostructures-based gas sensors: A review," Energy Reports, vol. 6, pp. 46–62, 2020.

[13] R. Ridhi, G. S. S. Saini, S. K. Tripathi, “Sensing of volatile organic compounds by copper phthalocyanine thin films,” Materials Research Express, vol. 4, no. 2, 025102, 2017.

[14] B. Wanga, Z. Li, X. Zuoa, Y. Wu, X. Wang, Z. Chen, C. He, W. Duan, J. Gao, “Preparation, characterization and NO2-sensing properties of octa-iso-pentyloxyphthalocyanine lead spin-coating films,” Sensors and Actuators B: Chemical, vol. 149, no. 2, pp. 362–367, 2010.

[15] A. Altındal, Ö. Kurt, A. Şengül, Ö. Bekaroğlu, “Kinetics of CO_2 adsorption on ball-type dicopper phthalocyaninethin film,” Sensors and Actuators B: Chemical, vol. 202, pp. 373–381, 2014.

[16] M. S. Ağırtaş, A. Altındal, B. Salih, S. Saydam, Ö. Bekaroğlu, Synthesis, “Characterization, and electrochemical and electrical properties of novel mono and ball-type metallophthalocyanines with four 9,9-bis (4-hydroxyphenyl) fluorene,” Dalton Transactions, vol. 40, no. 13, pp. 3315–3324, 2011.

[17] R. D. Yang, T. Gredig, C. N. Colesniuc, J. Park, I. K. Schuller, W. C. Trogler, A. C. Kummel, “Ultrathin organic transistors for chemical sensing,” Applied Physics Letters, vol. 90, no. 26, p. 263506, 2007.

[18] V. E. Bochenkov, G. B. Sergeev, “Sensitivity, selectivity, and stability of gas-sensitive metal-oxide nanostructures,” in Metal Oxide Nanostructures and Their Applications, American Scientific Publishers, 2010, vol. 3, ch. 2, pp. 31-52.

[19] S. M. Majhi, G. K. Naik, H.J. Lee, H. G. Song, C. R. Lee, I. H. Lee, Y. T. Yu, “Au@NiO core-shell nanoparticles as a p-type gas sensor: novelsynthesis, characterization, and their gas sensing properties with sensing mechanism,” Sensors and Actuators B: Chemical, vol. 268, pp. 223–231, 2018.

[20] X. Zou, J. Zhao, S. Wu, X. Huang, “Vinegar classification based on feature extraction and selection from tin oxide gas sensor array data,” Sensors, vol. 3, no. 4, pp. 101–109, 2003.

[21] T. Eklov, P. Martensson, I. Lundstrom, “Selection of variables for interpreting multivariate gas sensor data,” Analytica Chimica Acta, vol. 381, no. 2-3, pp. 221-232, 1999.

[22] K. Yan, D. Zhang, “Feature selection and analysis on correlated gas sensor data with recursive feature elimination,” Sensors Actuators B: Chem, vol. 212, pp. 353-363, 2015.

[23] A. Gramm, A. Schütze, “High performance solvent vapor identification with a two sensor array using temperature cycling and pattern classification,” Sensors Actuators B: Chem, vol. 95, no. 1-3, pp. 58-65, 2003.

[24] R. Ionescu, E. Llobet, “Wavelet transform based fast feature extraction from temperature modulated semiconductor gas sensors,” Sensors Actuators B: Chem, vol. 81, no. 2-3, pp. 289-295, 2002.

[25] J. Yan, F. Tian, Q. He, Y. Shen, S. Xu, J. Feng, K. Chaibou, “Feature extraction from sensor data for detection of wound pathogen based on electronic nose,” Sensors and Materials, vol. 24, no.2, pp. 57–73, 2012.

[26] R. Kaur, R. Kumar, A. Gulati, C. Ghanshyam, P. Kapur, A. P. Bhondekar, “Enhancing electronic nose performance: A novel feature selection approach using dynamic social impact theory and moving window time slicing for classification of Kangra orthodox black tea (Camellia sinensis (L.) O. Kuntze),” Sensors and Actuators B: Chemical, vol. 166–167, pp. 309–319, 2012.

[27] E. Martinelli, C. Falconi, A. D’Amico, C. Di Natale, “Feature extraction of chemical sensors in phase space,” Sensors and Actuators B: Chemical, vol. 95, no. 1-3, pp. 132–139, 2003.

[28] M.PadillaI, I. Montoliu, A. Pardo, A. Perera, S. Marco, “Feature extraction on three way enose signals,” Sensors and Actuators B, vol. 116, no. 1-2, pp. 145–150, 2006.

[29] J. Vitola, F. Pozo, D. A. Tibaduiza, M. Anaya, “A sensor data fusion system based on k-nearest neighbor pattern classification for structural health monitoring applications,” Sensors, vol. 17, no. 2, p. 417, 2017.

[30] M. Pardo, G. Sberveglieri, “Classification of electronic nose data with support vector machines,” Sensors and Actuators B: Chemical, vol. 107 no. 2, pp. 730-737, 2005.

[31] A.K. Srivastava, “Detection of volatile organic compounds (VOCs) using SnO2 gas-sensor array and artificial neural network,” Sensors and Actuators B: Chemical, vol. 96, no. 1-2, pp. 24-37, 2003.

[32] B. Tudu, A. Jana, A. Metla, D. Ghosh, N. Bhattacharyya, R. Bandyopadhyay, “Electronic nose for black tea quality evaluation by an incremental RBF network,” Sensors and Actuators B: Chemical, vol. 138, no. 1, pp. 90-95, 2009.

[33] J. Yan, X. Guo, S. Duan, P. Jia, L. Wang, C. Peng, S. Zhang, “Electronic nose feature extraction methods: A Review,” Sensors, vol. 15, no. 11, pp. 27804-27831, 2015.

[34] H.Li, D. Luo, Y. Sun, H. GholamHosseini, “Classification and identification of industrial gases based on electronic nose technology”, Sensors, vol. 19, no. 22, p. 5033, 2019.

[35] P. Jia, F. Tian, Q. He, S. Fan, J. Liu, S. X. Yang, “Feature extraction of wound infection data for electronic nose based on a novel weighted KPCA,” Sensors and Actuators B: Chemical, vol. 201, pp. 555–566, 2014.

[36] C. Kong, S. Zhao, X. Weng, C. Liu, R. Guan, Z. Chang, “Weighted Summation: Feature extraction of farm pigsty data for electronic nose,” IEEE Access, vol. 7, no. 99, p. 1, 2019.

[37] K. A. Ngo, P. Lauque, K. Aguir, “Identification of toxic gases using steady-state and transient responses of gas sensor array,” Sensors and Materials, vol. 18, no. 5, pp. 251-260, 2006.

[38] O. Güldali, “Deep feature transfer from deep learning models into machine learning algorithms to classify COVID-19 from chest X-ray images,” M. S. Thesis, Dept. Math. Eng., Istanbul Technical Univ., Istanbul, Turkey, 2021.

[39] N. J. Nilsson, “Preliminaries,” in Introduction to Machine Learning: An early draft of a proposed textbook, Robotics Laboratory Department of Computer Science Stanford University, Stanford, CA, 1998, ch. 1, p. 1.

[40] Machine Learning: A Quick Introduction and Five Core Steps. [Online]. Available: https://centricconsulting.com/blog/machine-learning-a-quick-introduction-and-five-core-steps/. (visited on 03/03/2022).

[41] S. Theodoridis, K. Koutroumbas, “Classifiers based on bayes decision theory,” in Introduction to Pattern Recognition: A Matlab Approach, 4th ed., Burlington, USA: Academic Press, 2009, ch. 1 p. 1.

[42] Supervised Machine Learning (SVM). [Online]. Available: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ (visited on 03/03/2022)

[43] C. M. Bishop, “Introduction” in Pattern Recognition and Machine Learning, New York, USA, Springer, 2006, ch. 1, pp. 3.

[44] F. Nielsen, Introduction to HPC with MPI for Data Science, Springer, 2016.

[45] E. Alpaydın, “Introduction,” in Introduction to Machine Learning, 2nd ed., Cambridge, Massachusetts London, England, The MIT Press, 2010, ch. 1, p. 13.

[46] J. D. Kelleher, B. M. Namee, A. D’Arcy, Fundamentals Of Machine Learning For Predictive Data Analytics, The MIT Press, Cambridge, Massachusetts, London, 2015.

[47] Understanding Bias and Variance [Online]. Available: https://medium.com/@jason9389/understanding-bias-and-variance-in-relation-to-features-in-supervised-learning-13f7846ff10e (visited on 03/03/2022)

[48] G. Hackeling, “From Linear Regression to Logistic Regression,” in Mastering Machine Learning with scikit-learn, 1st ed., Birmingham, UK, Packt Publishing, 2014, ch. 7, pp. 76-79.

[49] C. Cortes, V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[50] S. Theodoridis, K. Koutroumbas,”Support vector machines,” in Pattern Recognition, 2nd ed., Boston, USA, Academic Press, 2003, ch. 3, sec. 3.6, p.77.

[51] Support Vector Machine Algorithm. [Online]. Available: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm (visited on 03/03/2022).

[52] J. Mrva, S. Neupauer, L. Hudec, J. Sevcech, P. Kapec, “Decision support in medical data using 3D decision tree visualisation,” E-Health and Bioengineering Conference (EHB), pp. 1–4, 2019.

[53] M. H. Chiu, Y. R. Yu, H. L. Liaw, L. C. Hao, “The use of facial micro-expression state and Tree-Forest Model for predicting conceptual-conflict based conceptual change,” In Science Education Research: Engag-ing Learners for a Sustainable Future (ESERA eProceeding); ESERA: Kaunas, Lithuania, vol.50, pp. 184–19, 2016.

[54] A. Dumankaya, “Applying machine learning approaches to price prediction,” M. S. Thesis, Dept. Computer Eng., Ankara Yıldırım Beyazıt Univ., Ankara, Turkey, 2021.

[55] B. Ay Türe, “Design of predictive maintenance model using artificial intelligence methods,” M. S. Thesis, Dept. Computer Eng., Istanbul Commerce Univ., Istanbul, Turkey, 2021.

[56] Ş. Abdurrahmanoğlu, A. Altındal, M. Bulut, Ö. Bekaroğlu, “Synthesis and electrical properties of novel supramolecular octa-phthalocyaninato-dicobalt(II)-hexazinc(II) and dicobalt(II)-dimeric-phthalocyanine with six ferrocenylimin pendant groups,” Polyhedron, vol. 25, no. 18, pp. 3639–3646, 2006.

[57] S. M. Scott, D. James, Z. Ali, “Data analysis for electronic systems,” Microchimica Acta, vol. 156, no. 3, pp. 183-207, 2006.

[58] B. Tudu, B. Kow, N. Bhattacharyya, R. Bandyopadhyay, “Normalization techniques for gas sensor array as applied to classification for black tea”, International Journal on Smart Sensing and Intelligent Systems, vol. 2, no. 1, pp. 176–189, 2009.

[59] S. Zhang, C. Xie, D. Zeng, Q. Zhang, H. Li, Z. Bi, “A feature extraction method and a sampling system for fast recognition of flammable liquids with a portable e-nose,” Sensors and Actuators B: Chemical, vol. 124, no. 2, pp. 437-443, 2007.

[60] C. Distantea, M. Leo, P. Sicilianoa, K. Persaud, “On the study of feature extraction methods for an electronic nose,” Sensors and Actuators B: Chemical, vol. 87, no.2, pp. 274–288, 2002.

PUBLICATIONS FROM THE THESIS

Conference Papers

1. S. Aksoy, M. Özavşar "Feature extraction methods from gas sensor data," 3th International Congress on Statistics Mathematics and Analytical Method, Istanbul, Turkey, p. 77, 12 -13 March 2020.

2. S. Aksoy, M. Özavşar, "Classification of VOC vapors using machine learning algorithms," 8th International Congress on Fundamental and Applied Sciences, Antalya, Turkey, p. 158, 19 -21 October 2021.