UNIVERSITY OF CASSINO AND SOUTHERN LATIUM

Ph.D. in Methods, models, and technologies for engineering
XXXV Cycle

Artificial Intelligence for pollutant recognition applied to smart sensors based on SENSIPLUS

Coordinator
Prof. Fabrizio Marignetti

Ph.D. student
Luca Gerevini

Supervisor
Prof. Mario Molinara

A.A. 2021/2022

“The greatest enemy of knowledge is not ignorance,
it is the illusion of knowledge.”
Stephen Hawking

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Prof. Mario Molinara that supports and helped me alongside all the Ph.D. A special thank goes to Prof. Luigi Ferrigno who helped to make me grow both from a scientific and personal point of view. I would like to thank Prof. Francesco Fontanella that contributed to increasing my scientific knowledge, and all the computer science and electronic measurements research groups of the University of Cassino and Southern Latium. I would like to also thank Roberto Simmarano, CEO of the Sensichips s.r.l. for which I was employed during the first two years of my doctorate, which allowed me to pursue my Ph.D. Thanks to all my Ph.D. fellow students who have become valuable friends, for giving me the opportunity to work in a very friendly and stimulating environment. A special thank goes to Carmine Bourelly who has worked alongside me, supporting me in my research activities. A special thank goes also to Gianni Cerro whose precious advice taught me a lot. Finally, I would like to thank all my family: my parents, my sister, and my lovely girlfriend, for supporting and putting up with me throughout the Ph.D. years.

ABSTRACT

The problem of detecting pollutants in wastewater is of fundamental importance for public health and security. In a fast-developing paradigm, such as the one represented by the Smart City, mapping wastewater systems by detecting pollution sources is a required task that if properly treated helps protect ecosystems and may allow for recovering energy, recoverable material, and nutrients. Generally, this is made by involving the usage of laboratory-based analyses performed by expert professionals that ends to result in high costs and time spending approach which does not allow for timely prevention of environmental disasters. In this regard, to allow an effective prevention activity, a large number of distributed measurement systems are required. In this Ph.D. thesis are presented two possible solutions based on Machine Learning (ML) techniques using the same sensing part. The proposed measurement systems are based on the so-called Smart Cable Water (SCW) sensor, a multi-sensor based on SENSIPLUS technology developed by Sensichips s.r.l. More in detail, one solution is aimed at the development of an end-to-end IoT-ready system for the recognition of a set of substances able to reduce the false positive samples by distinguishing the outlier from the interest ones. In this regard, the system is composed of three functional blocks: a Finite State Machine (FSM) to correctly detect the substance's passage, an anomaly detection classifier to reject all the outlier samples, and a multiclass classifier to correctly recognize the given substance. It is important to note that the capability to distinguish between outlier (not of interest) and inlier (of interest) substances drastically improves the classification performance, especially in terms of false positive rates. An extensive experimental campaign on different contaminants has been carried out to train machine learning algorithms suitable for low-cost and low-power Micro Controlled Unit (MCU). The obtained results demonstrate an excellent classification ability, achieving an accuracy of more than 95% on average, and are a reliable "proof of concept" of a pervasive IoT system for distributed monitoring. The other solution is aimed at the development of an edge computing classifier to be implemented abord the SCW sensor. In this regard has been used the Principal Component Analysis (PCA) decomposition to project the acquired data from a 10-dimensional space to a 3-dimensional one. Next, has been developed an ad-hoc classifier capable to distinguish contaminants in the projected 3-dimensional space. To learn the best classifier's parameters has been used an evolutionary algorithm. The proposed system achieved the best accuracy of 83%, outperforming the other state-of-art systems compared. The novelty of the proposed system lies in the usage of an evolutionary algorithm for the optimization of the parameters of a novel PCA-based classification algorithm capable to detect and recognize a set of wastewater pollutants.

TABLE OF CONTENTS

ABSTRACT I
TABLE OF CONTENTS II
List of Figures V
List of Tables IX
List of Algorithms XI
List of Equations XIII
INTRODUCTION 1
CHAPTER 1. Sensichips System 2
1.1 SENSIPLUS chip 2
1.2 SENSIPLUS Applications 3
1.2.1 Smart Cable Water 3
1.2.2 Smart Cable Air 5
1.2.3 Battery Cell Management Unit 6
1.2.4 Microanalytical Tool 8
1.3 Software API 9
1.3.1 API Level 0 10
1.3.2 API Level 1 11
1.3.3 API Level 2 11
1.4 Developed Application 11
1.4.1 Winux 12
1.5 Embedded API 19
CHAPTER 2. Contaminants Detections and Recognitions Using Machine Learning Techniques 22
2. State of the Art 22
2.1 Scientific Contribution 23
2.2 Measurement SetUp 24
2.3 Features Selection 26
2.4 Data Collection 28
2.4.1 Data Set 30
2.5 Classification System 32

2.5.1 Data Processing....33
2.5.2 Classification System....39
2.5.3 Learning Procedure....42
2.6 Experimental Results....44
2.6.1 Anomaly Detection Results....44
2.6.2 Multiclass Classifier Results....46
2.6.3 Entire System Results....46
2.7 Discussion....48
2.8 Conclusion and future developments....49
CHAPTER 3. Contaminants Detections and Recognitions Using Ad-Hoc Algorithms 52
3. State of the Art....52
3.1 Cone Based Algorithms....54
3.1.1 System Architecture....54
3.1.2 Classification Model....58
3.1.3 Evolutionary Algorithms....59
3.1.4 Results....61
3.1.5 Related Problems....64
3.2 Line Based Algorithms....65
3.2.1 System architecture....66
3.2.2 Data Transformation....66
3.2.3 Classification Model....67
3.2.4 Results....69
3.2.5 Further Work....72
3.2.6 Polar coordinates....72
3.2.7 Smart Initialization....74
3.2.8 Results....75
3.3 Dynamic Labeling....84
3.3.1 Obtained Results....85
3.4 Conclusion....90
CHAPTER 4. Test On Real Field....92
4. Introduction....92
4.1 Borgo Piave (Latina) Tests....95
4.2 East Rome Tests....99
4.3 Further Tests....103
4.4 Conclusion....106

Appendix A.. 108
References.. 114

LIST OF FIGURES

Figure 1 SENSIPLUS simplified block diagram. 3
Figure 2 Smart Cable Water 3
Figure 3 Smart Cable Air 5
Figure 4 Battery Cell Management Unit 6
Figure 5 Microanalytical Tool 8
Figure 6 IoTMAT 8
Figure 7 Software API Architecture 10
Figure 8 Debug tab view. 12
Figure 9 Modify Configuration dialog view. 13
Figure 10 Log view. 13
Figure 11 Connected chips list massage after a connection establishment. 14
Figure 12 Batch tab view. 14
Figure 13 Batch tab with an XML configuration file loaded. 15
Figure 14 XML file containing a possible EIS measurements configuration. 16
Figure 15 Batch tab view a detailed description. 17
Figure 16 EIS measurements information. 17
Figure 17 POT staircase measurements information. 18
Figure 18 Classification Systems results information. 19
Figure 19 Measurement SetUp 24
Figure 20 Randles equivalent circuit. 25
Figure 21 Smart Cable Water IDEs. 26
Figure 22 Randles at low frequency 27
Figure 23 Randles at high frequency 28
Figure 24 Data Set Structure, Exp0, 1, …, 9 means respectively acquisition 0, 1, …, 9 of all substances. 31
Figure 25 Classification System 32
Figure 26 Finite State Machine 35
Figure 27 Finite State Machine Flow Chart 35
Figure 28 Behavior of the old baseline tracking system. 36
Figure 29 Behavior of the new baseline tracking system. 37

Figure 30 Entire System Flow Chart..... 41
Figure 31 Multiclass Classifier Results 47
Figure 32 Entire System Results..... 48
Figure 33 Background substance sphere..... 55
Figure 34 System Architecture..... 56
Figure 35 Representation of a cone with internal points..... 57
Figure 36 Confusion matrix on test data..... 63
Figure 37 LB system architecture. 66
Figure 38 Line-based Model. 68
Figure 39 Entire 3-D classification model. 69
Figure 40 Confusion Matrix CB system. 70
Figure 41 Confusion Matrix LB system 71
Figure 42 Spherical LB model 3-D view. 73
Figure 43 Confusion matrix PB system. 76
Figure 44 Confusion matrix LB system. 76
Figure 45 Confusion matrix PB system with LDA. 78
Figure 46 Overall average fitness. 79
Figure 47 Single substance average fitness. 80
Figure 48 Confusion matrices ML algorithmes. 83
Figure 49 Comparison between gene's DL and SL strategies. 85
Figure 50 Comparison results between DL and LB model's best results. 87
Figure 51 DL best results. 89
Figure 52 Wastewater treatment plant, BorgoPiave Latina. 93
Figure 53 Via Castelbottaccio East Rome. 93
Figure 54 Best KNN model used during real scenario test..... 94
Figure 55 Borgo Piave System tests. 95
Figure 56 Experimental environment..... 96
Figure 57 SCW positions. 96
Figure 58 Air bubbles from water turbulence effects. 97
Figure 59 Lifting pump system interferences. 97
Figure 60 Solid garbage. 98
Figure 61 BorgoPiave tested substances. 98
Figure 62 East Rome System test. 99
Figure 63 East Rome test: green circle represents the sensing manhole, while red circles represent the spiking manholes respectively positioned at 50m, 75m, and 150m from the sensing manhole. 100

Figure 64 Measurement system prototype, developed for East Rome tests. 101
Figure 65 East Rome human activities interferences. On the Right is shown the resistance over the gold IDE at 78kHz, and on the left resistance of the platinum IDE at 78 kHz..... 102
Figure 66 Finite State Machine parameters. 102
Figure 67 Test in the chemical laboratory of RaCIS in Rome. 103
Figure 68 Anzio Monumento Caduti Due Guerre Mondiali real test location. The red circle indicates the spiking manhole, while the green is the sensing one. The manholes were located 10 meters apart. 103
Figure 69 Anzio real test. 104
Figure 70 Beurbach wastewater treatment plant tests. Red Circles indicate the two spots where SCW was installed. 105
Figure 71 XML file structure with an example showing the measurement flow.108
Figure 72 XML attributes related to the experiment and sensor tags. 109
Figure 73 XML attributes related to the eis and pot tags. 109

LIST OF TABLES

Table 1 Smart Cable Water Specifications 4
Table 2 Smart Cable Air Specifications..... 6
Table 3 Battery Cell Management Unit Specifications..... 7
Table 4 Microanalytical Tool Specifications 9
Table 5 Synthetic Waste Water Composition 29
Table 6 DataSet Samples for different classes..... 38
Table 7 Global results for an MLP using the old baseline tracking method. 38
Table 8 Global results for an MLP using the new baseline tracking method. 38
Table 9 Anomaly Detection Models Parameters 41
Table 10 Multiclass Classification Model Parameters..... 42
Table 11 Best Results Anomaly Detection 45
Table 12 Fold 0 Results 45
Table 13 Cone Performance in terms of accuracy, F-score, and true-positives (TP), true-negatives (TN), false-positives (FP), and false-negatives (FN). 62
Table 14 The values of the parameters used in the experiments..... 62
Table 15 Comparison results..... 64
Table 16 Percentages of substances 64
Table 17 Evolutionary algorithm parameters..... 70
Table 18 Comparison results..... 72
Table 19 Classifier parameters..... 82
Table 20 Comparison results..... 82
Table 21 DL strategy with a different number of genes results. 86
Table 22 Dynamic Labeling comparison results..... 89

LIST OF ALGORITHMS

Algorithm 1 Training Procedure 42
Algorithm 2 Test Procedure 43
Algorithm 3 Online Classification Procedure 43
Algorithm 4 Evolutionary algorithm. 61
Algorithm 5 Line-based fitness function. 65
Algorithm 6 Individual initialization. 74

LIST OF EQUATIONS

Equation 1 33
Equation 2 34
Equation 3 34
Equation 4 55
Equation 5 58
Equation 6 58
Equation 7 58
Equation 8 59
Equation 9 60
Equation 10 67
Equation 11 67
Equation 12 68
Equation 13 68
Equation 14 73
Equation 15 74

INTRODUCTION

Water quality is a crucial factor regarding human life; nowadays, about 23% of global diseases can be related to pollutants in air and water environments [1]. For those reasons, many researchers worldwide focus on developing tools for automatic contaminant detection in water and air. Water pollution [2], in this main research context, represents a worldwide concern, also regarding drinkable tap water [3] as confirmed by the World Health Organization (WHO), which has estimated that this problem plagues about two billion people. For that reason, water quality monitoring is increasingly involving researchers from different fields of interest, from artificial intelligence [4] [5] to sensors [6] and data processing [7]. This concern also includes wastewater which represents the main focus of this Ph.D. thesis.

In fact, in a fast-developing paradigm, such as the one represented by the Smart City, mapping wastewater systems by detecting possible pollution sources is a required task that if properly treated helps protect ecosystems and may allow recovering energy, recoverable material, and nutrients. Usually, water quality monitoring is generally made by laboratory-based analyses performed by expert professionals. Because of the cost and the time required by those kinds of analyses, this approach does not allow for the timely prevention of environmental disasters. In this regard, to allow an effective prevention activity, a large number of distributed measurement systems are required.

Although on the one hand, these measurement systems are available today and guarantee excellent measurement accuracy and great reliability in detecting polluting substances. On the other hand, their usage is limited by a high cost. In this context, the usage of low-cost and low-power microsensor systems becomes fundamental.

Moreover, in the context of wastewater monitoring, where the environmental conditions are pretty complex due to the high-water turbidity, sensor degradation, and so on, it’s crucial to combine low costs with good measurement accuracy, as well as good reliability. Typically, systems with these kinds of features are suitable for the paradigms of the Internet of Things (IoT) [8] [9] [10] [11] as well as those of edge and fog computing [12] [13] to perform early analysis and detection in the actual scenario. Both IoT and edge computing benefit from the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques to effectively analyze and exploit the

information contained in the generated data [14] [15] [16]. At this point, it is clear that the problem of detecting pollutants in water with non-invasive and low-cost sensors is an open question.

This Ph.D. thesis will present two possible solutions based on the same measurement system. In particular, the proposed systems are based on a proprietary embedded IoT-ready Micro-Analytical Sensing Platform (MASP) of 1.5mW power absorption and 1.5 x 1.5 mm of the size called Smart Cable Water (SCW) and based on the SENSIPLUS microchip. The SCW is a smart sensor endowed with six interdigitated electrodes (IDEs) covered by specific sensing materials to allow the differentiation between different pollutants. Both SCW and SENSIPLUS microchips have been developed and designed by Sensichips s.r.l. company (see Chapter 1 for more details).

More in detail, one solution is aimed at the development of an end-to-end IoT-ready system for the recognition of a set of substances of interest. From the literature emerges that in the water analysis field, one of the open issues is related to the lack of an anomaly detection system. Thus in a real context, a classification system that ignores the anomalies makes the system itself unusable due to the number of false positives. Thus, the developed system (deeply described in Chapter 2) is capable of detection and recognition of a set of substances (considered dangerous and indicative of an anomalous use of the wastewater) by rejecting all the outlier samples. In particular, the system is composed of three functional blocks:

- a Finite State Machine (FSM) meant to correctly detect a substance's passage;
- an anomaly detection classifier trained in order to be able to reject all the outlier samples;
- a multiclass classifier to correctly recognize the given substance.

It is important to note that the capability to distinguish between outlier (not of interest) and inlier (of interest) substances drastically improves the classification performance, by reducing the number of false positive samples and making the system suitable to be used in a real scenario. The results show an average accuracy greater than 95%, demonstrating an excellent classification capability. Moreover, the promising results can be considered a reliable "proof of concept" of a pervasive IoT system for distributed monitoring.

The other solution, presented in Chapter 3, is aimed at the development of a system for detecting and recognizing wastewater pollutants based on an ad-hoc lightweight algorithm suitable for the IoT and edge-computing paradigms. To make the system as edge computing suitable as possible, the 10-dimensional features space has been

compressed to a 3-dimensional space by using well-known decomposition techniques like for example the principal component analysis (PCA), linear discriminant analysis (LDA), etc. Regarding the classification system, a straightforward, geometrical-based model that can be implemented using very few hardware resources has been developed. As for the best model parameters, on the other hand, they were learned through the use of evolutionary algorithms. It's important to note that in the Ph.D. work, the developed model has been constantly improved by achieving the best accuracy of 83%. Finally, in order to validate the goodness of the obtained results, the ad-hoc system has been compared with the other state-of-art ones showing that the developed system outperforms the state-of-the-art ones.

Furthermore throughout the Ph.D., to validate and improve the effectiveness of the system depicted in Chapter 2, have been performed many tests on the real field. In particular, two main scenarios have been considered:

- the wastewater treatment plant of Acqualatina in Borgo Piave (Latina, Italy);
- a series of manholes situated in Via Castelbottaccio (East Rome, Italy).

Chapter 4 contains a detailed description of all the tests performed on the real field. It is crucial to highlight the importance of having had the opportunity to perform real field tests allowing us to be able to develop an end-to-end reliable and robust system able to correctly recognize a given set of substances.

CHAPTER 1. SENSICHIPS SYSTEM

Sensichips is a little firm involved in the learning microsensors fields, located in Anzio Italy. It was founded in 2011 by experienced entrepreneurs from the Semiconductors industry and Arescosmo SpA, a company active in Aerospace and Defence. With its technology, Sensichips wants to set new performance standards and enable new applications in all that fields that involve sensors such as robotics, electrification, wearable, healthcare automation internet of everything. The team is formed by engineers and researchers in the field of microelectronics, firmware, artificial intelligence, material science, and electrochemistry. It is located in the Design Centers of Pisa and Cassino, while at the headquarters of Anzio, there is the Chemical Laboratory, Application Development, Administration, and Management.

1.1 SENSIPLUS chip

One of the leading technologies designed and developed by Sensichips is the SENSIPLUS. The SENSIPLUS is a microelectronics platform for multiple heterogeneous sensors integrated into a single chip or miniature Multi-Sensor Microsystems (MSM) and represents the core business of Sensichips. The objective is to combine heterogeneous functional materials with CMOS microelectronics into a single-chip sensor.

SENSIPLUS is a versatile System on a Chip (SoC) for sensor interfacing, designed and fabricated in a 0.18m CMOS process. A simplified block diagram of the SENSIPLUS chip is shown in Figure 1. The SENSIPLUS is designed to accomplish the widest range of measurements that can be performed on physical and chemical systems. At the core of the chip, there is a highly precise and versatile analytical microchip that includes a Wideband Electrical Impedance Spectrometer (EIS) based on a frequency programmable Lock-In Amplifier (LIA) and a Potentiostat in a miniature 3x3 mm chip with 2mW power consumption when all functions are active.

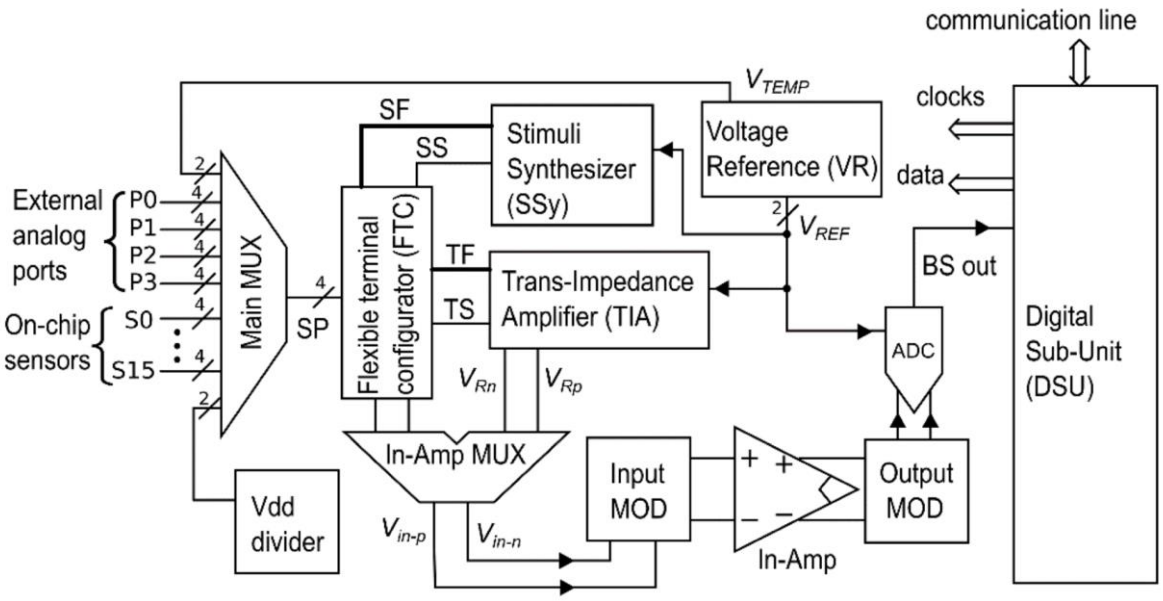


Figure 1 SENSIPLUS simplified block diagram.

1.2 SENSIPLUS Applications

In this section, I would like to give an overview of all the applications based on the SENSIPLUS technologies developed by Senschips s.r.l. It is worth specifying that during my Ph.D. I mainly worked with the Smart Cable Water (SCW) sensor platform, but as a Sensichips employee, I worked with the other applications as well.

1.2.1 Smart Cable Water

Figure 2 Smart Cable Water

Sensichips have developed Smart Cable Water (SCW) to create a multi-sensor microsystem (MSM) to monitor for the presence of toxic chemicals (TICs), pollutants, hydrocarbons, and organics in water. The measurement principle is mainly based on impedance spectroscopy measurements over six different coated interdigitated electrodes (IDEs): five small IDEs on the front face coated with Gold, Copper, Silver, Nickel, Palladium (dimensions 3 mm by 7 mm each) and one Platinum IDE (dimensions 12 mm by 8 mm) on the backside (see Figure 2). A CMOS bandgap temperature sensor, a light emitter, and a light photo-diode sensor are also available on the board.

At the core of SCW, there is the SENSIPLUS, the Sensichips microsensor platform that can interrogate on-chip and off-chip sensors with its versatile and accurate Electrical Impedance Spectrometer (EIS) and Potentiostat. Analytics performed with EIS allow for the exploitation of RedOx dynamics of catalytic noble metals to aid chemical discrimination plus measurement of conductivity and permittivity spectra.

The on-chip Potentiostat can be used for a number of different Voltammetric or Amperometric measurements and real-time discrimination of pollutants. Table 1 shows the main specifications related to the SCW.

ELECTRICAL	
Supply Voltage	1.5 – 3.6V
Max Current	0.4mA continuous when reading on-chip sensors with EIS
Size	12x15mm, 3mm thickness
Interface	I^2C or SENSIBUS, single data wire multidrop sensor array cable interface, 1.5-3.6V
Unique Identifier	OTP 48bits Unique Device Identifier, 16bits User Defined
ELECTRICAL IMPEDANCE SPECTROSCOPY	
Frequency	3.1mHz to 1.2MHz
Vpp output sinewave	156mV to 2.8Vpp
Coherent demodulation	1^{st}, 2^{nd}, or 3^{rd} harmonic
Output	Reciprocal of real or imagery component
Wide Measurement Range	From ohms to 100MΩ

Table 1 Smart Cable Water Specifications

1.2.2 Smart Cable Air

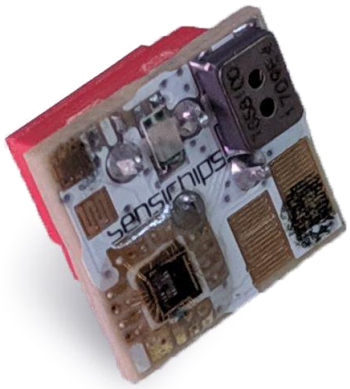


Figure 3 Smart Cable Air

Other than the SCW, Sensichips has developed Smart Cable Air (SCA) to create a compact size, about 8x9 mm, and a low-power solution to monitor air quality (see Figure 3). The SCA is a multi-sensor microsystem (MSM) mainly used to monitor for the presence of toxic chemicals (TICs), pollutants, volatile organic compounds (VOCs), and flammable gases in the air. At the core of the board can be found a SENSIPLUS and, as for SCW, the measurements are mainly based on the Electrical Impedance Spectrometer (EIS).

This measurement technique allows the exploitation of chemisorption or ReDox dynamics of the sensitive film to aid gas discrimination. Furthermore, the EIS allows the derivation of an R/C equivalent circuit of the sensor to decouple components that drift from the ones that represent the response to the gas.

Several SCAs can be installed on the same cable for the continuous monitoring of a large area, which in union with the versatile analytical instruments of the onboard IDEs, makes the SCA an excellent experimental board. Table 2 shows the main specifications related to the SCA.

ELECTRICAL	
Supply Voltage	1.5 – 3.6V
Max Current	0.4mA continuous when reading on-chip sensors with EIS
Size	8.6×7.8mm
Interface	I3C or SENSIBUS, single data wire multidrop sensor array cable interface, 1.5-3.6V
Unique Identifier	OTP 48bits Unique Device Identifier, 16bits User Defined
ELECTRICAL IMPEDANCE SPECTROSCOPY	
Frequency	3.1mHz to 1.2MHz
Vpp output sinewave	156mV to 2.8Vpp
Coherent demodulation	1st, 2nd, or 3rd harmonic
Output	Reciprocal of real or imagery component
Wide Measurement Range	From ohms to 100MΩ

Table 2 Smart Cable Air Specifications

1.2.3 Battery Cell Management Unit

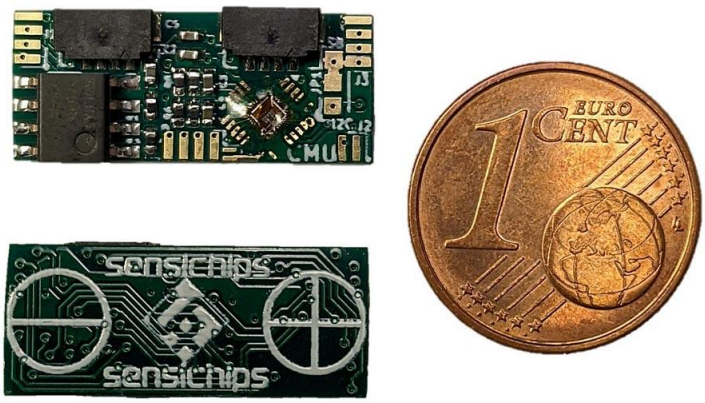


Figure 4 Battery Cell Management Unit

The Battery Cell Management Unit (CMU) has been designed by Sensichips for battery engineers and researchers (see Figure 4). The CMU allows multiple measurements of supercapacitors and battery cells while in operation. The CMU is

based on SENSIPLUS that allows measuring 1st, 2nd, and 3rd harmonics of a Potentiostat/Galvanostat along with on-chip sensors for Temperature, Relative Humidity, and Gassing.

The CMU board includes the Anode and Cathode electrodes that allow a 4 contacts measurement of the cell's internal impedance and tab temperature. Furthermore, the CMU is chemistry independent and can be used with Li-ION and Li-Fe-PO_4 battery packs and with 3b generation of LNMO cells, Supercapacitors, and Fuel Cell combinations. Table 3 shows the main specifications related to the CMU.

ELECTRICAL	
Supply Voltage	1.5-4.5V directly powered by the battery cell
Max Current	20mA when reading EIS at best accuracy
Size	20×8.5mm, 2.3mm thickness
Interface	Isolated (UL1577 rating) I3C or SENSIBUS, single data wire bus
Unique Identifier	OTP 48bits Unique Device/Cell Identifier, 16bits User Defined
ELECTRICAL IMPEDANCE SPECTROSCOPY	
EIS	Four contacts EIS measurement of internal Cell Impedance
Frequency	40Hz to 100KHz
Ipp output sinewave	0 to 20mApp
Coherent demodulation	1st, 2nd, or 3rd harmonic
Output	Reciprocal of real or imagery component
Measurement Resolution	1mΩ with 200μΩ accuracy at 1KHz

Table 3 Battery Cell Management Unit Specifications

1.2.4 Microanalytical Tool

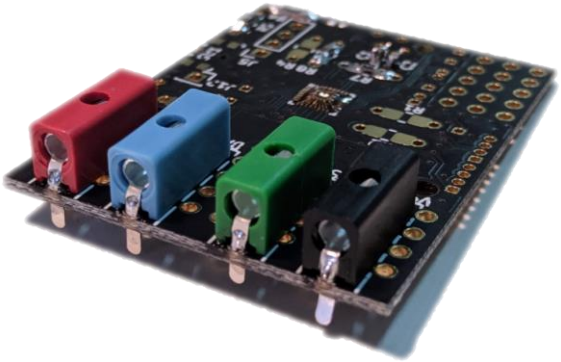


Figure 5 Microanalytical Tool

The Microanalytical Tool (MAT), shown in Figure 5, is a multifunctional measurement instrument that can be used in the laboratory or on the field for continuous online monitoring. The Standalone version can be powered by USB or Battery, and data connectivity includes WiFi, Bluetooth, or USB. The MAT is based on the SENSIPLUS chip and, thanks to the 2mm banana connectors, allows:

- Precision LCR Meter
- Electrochemical Impedance Spectrometer
- Potentiostat and Galvanostat

The main specifications of the MAT can be found in Table 4. One of the main applications of the MAT is represented by the IoTMAT (see Figure 6) which can be used for: precision laboratory instruments, in-field or on-line data logger, networked and remote operation, etc.

Figure 6 IoTMAT

ELECTRICAL	
Supply Voltage	Single Cell Li-Ion Battery or USB cable
Max Power	MAT Standalone: 500mW with WiFi communication MAT Tethered: 3.5mW for EIS measurement
Size	MAT Standalone: 63x34x10mm MAT Tethered: 42x32x7mm
Interface	MAT Standalone: USB, WiFi, Bluetooth MAT Tethered: SENSIBUS
LCR METER	
Frequency	3.1mHz to 1.2MHz
Vpp output sinewave	156mV to 2.8Vpp
Calibration	Open/Short/Load compensation support
Output	Reciprocal of real or imagery component
Wide Measurement Range	From ohms to 100MΩ
ELECTRICAL IMPEDANCE SPECTROSCOPY	
Frequency	3.1mHz to 1.2MHz
Vpp output sinewave	156mV to 2.8Vpp
Coherent demodulation	1^{st}, 2^{nd}, or 3^{rd} harmonic
Output	Reciprocal of real or imagery component
Wide Measurement Range	From ohms to 100MΩ

Table 4 Microanalytical Tool Specifications

1.3 Software API

All the software APIs, which concern the SENSIPLUS ecosystem, have been designed and developed in collaboration with the University of Cassino and Southern Latium. The API library is structured in three main architectures levels:

- **API Level 0**: provides methods for communication between the host and the SENSIPLUS chip.
- **API Level 1**: provides methods for decoding high-level calls into bytes that will be sent directly to the chip.
- **API Level 2**: provides all the methods available for different types of analytical measurements.

For a more detailed description of the API architecture see Figure 7. Obviously, during this Ph.D. period, I have personally contributed to the design and development of new API features either form a researcher's point of view by implementing artificial intelligence algorithms or data processing procedures, that as a Sensichips employee by implementing new features or improving the already existing ones.

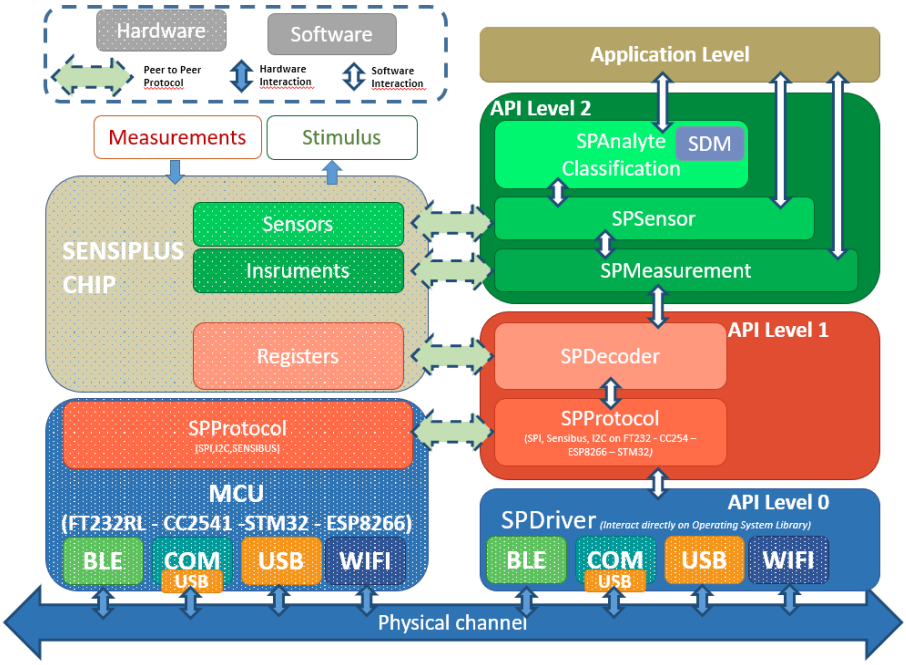


Figure 7 Software API Architecture

1.3.1 API Level 0

API Level 0 takes care of managing the interface driver between the host and the SENSIPLUS. This architectural level allows to send and receive information to and from the chip, thanks to the abstraction of the specific hardware interface (I_2C, SPI, or SENSIBUS).

The primary method of this API level is the *sendData()* that allows to send and receive bytes to/from the SENSIPLUS chip.

1.3.2 API Level 1

API Level 1 takes care to decode high-level instructions, which come from API Level 2, into structured bytes that will be sent to the chip through API Level 0. This level also implements a communication abstraction with the chip directly or through an intermediate bridge device. If the chip directly interfaces with the Host, this abstraction is not implemented.

1.3.3 API Level 2

API Level 2 provides all the methods necessary to perform all the available analytical measurements. For each type of measurement, there are two different methods: *setMeasureType* and *getMeasureType*. Where the *setMeasureType* is needed to set the initial chip state and internal parameter that will be used to perform a given measure type, while the *getMeasureType* request to the chip one or more measures returning to the caller the corresponding numeric values.

1.4 Developed Application

In this section, I want to describe the Winux application developed by Sensichips s.r.l. with the collaboration of the University of Cassino and the Southern Latium in order to perform the desired measurements with the previously depicted SENSIPLUS devices (SCW, SCA, CMU, IoTMAT).
During my Ph.D. work, to implement and improve the developed end-to-end system, I personally worked on the Winux application either by implementing completely new features (like, for example, all the classification systems), that by improving the already existing ones.

1.4.1 Winux

The Winux application has been developed in order to be as much as possible versatile either from the measurement experiments settings point of view, that from the SENSIPLUS interaction. In this regard, the application is mainly divided into two tabs:

- Debug
- Batch

Debug Tab

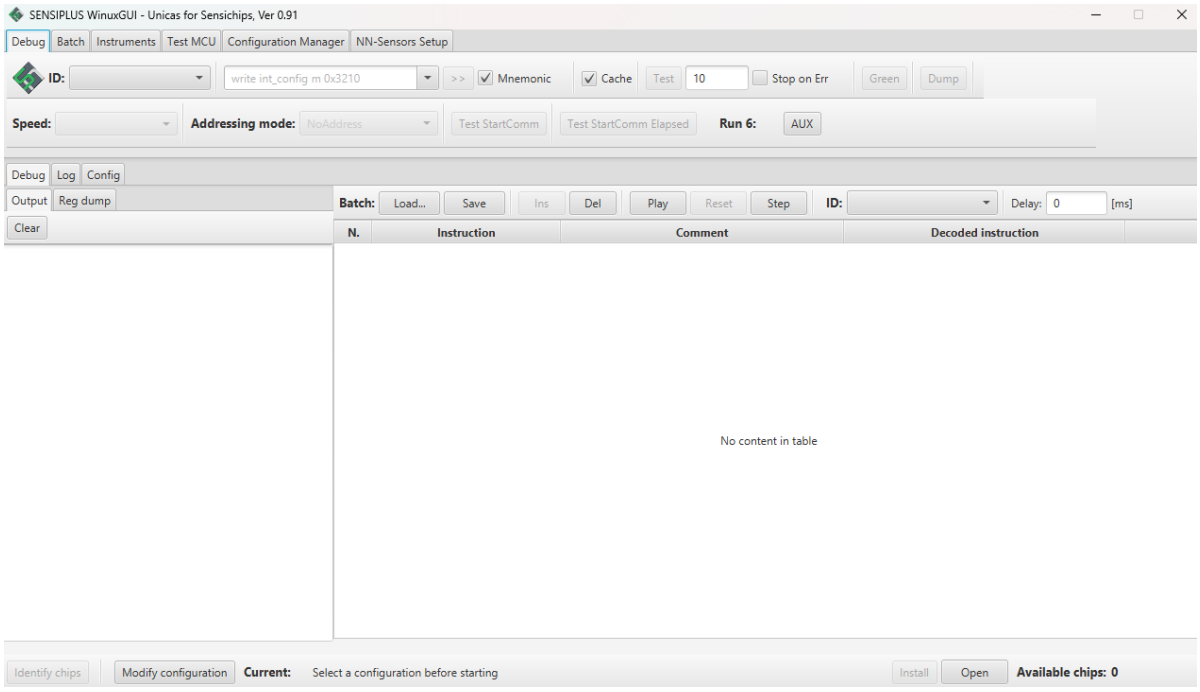


Figure 8 Debug tab view.

The main Debug features are meant to allow the user to direct communicate with the SENSIPLUS chip, by sending/receiving single instructions. Furthermore, allow the user to connect with the given device in order to start a measurement experiment (see Figure 9). Finally, it is capable to take a trace of all the instructions sent to the SENSIPLUS chip during an experiment (see Figure 10).

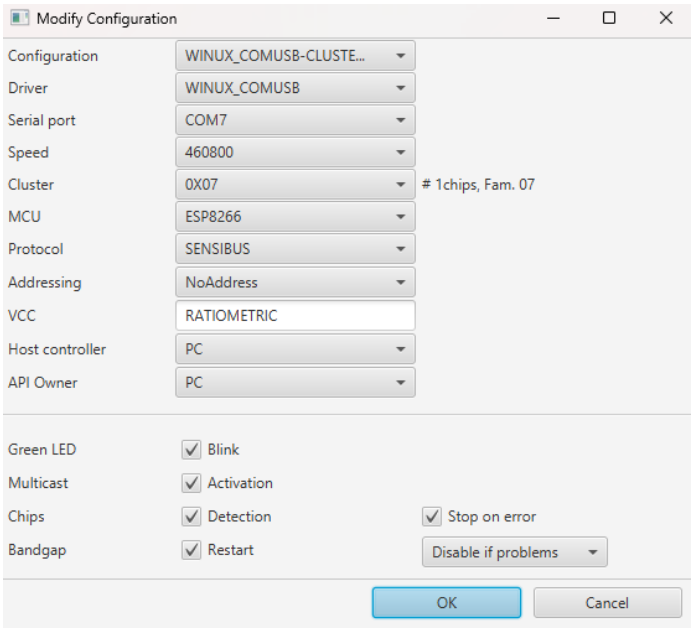


Figure 9 Modify Configuration dialog view.

As can be seen from Figure 9 from the Modify Configuration dialog it's possible to select the given device (from the Configuration's combo box) and sets all the given parameters to allow the connections.

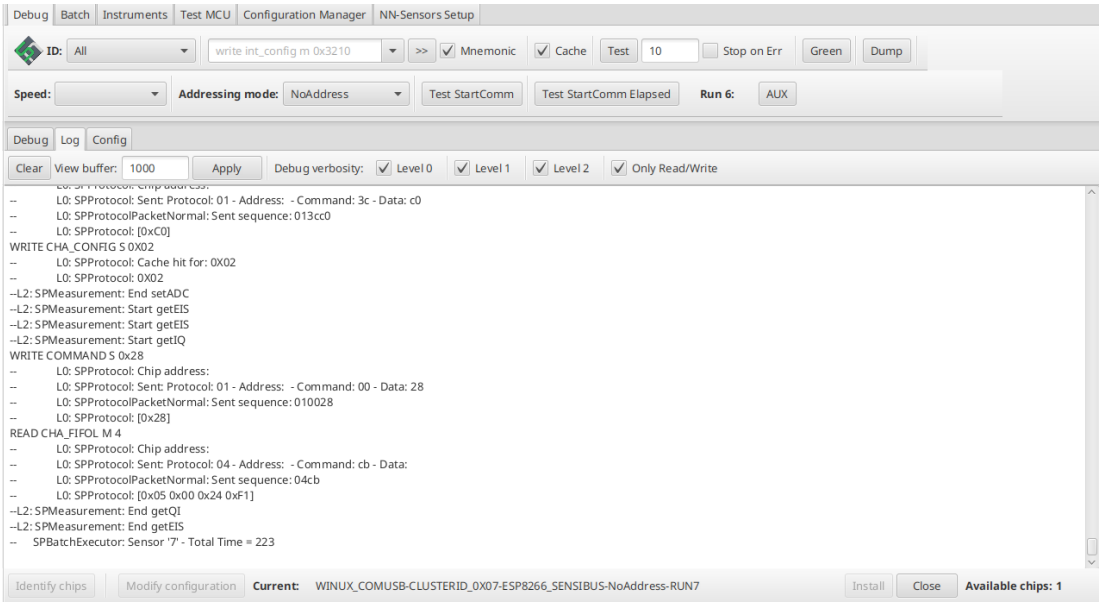


Figure 10 Log view.

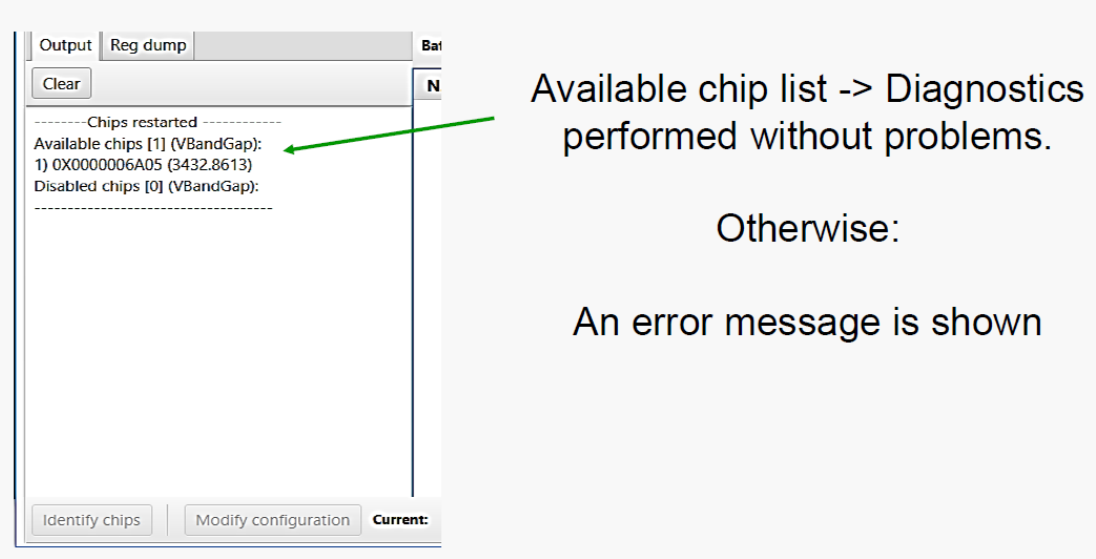


Figure 11 Connected chips list massage after a connection establishment.

Once the connection with the device is established, the application will show an output message with the list of all the connected chips (see Figure 11).

Batch Tab

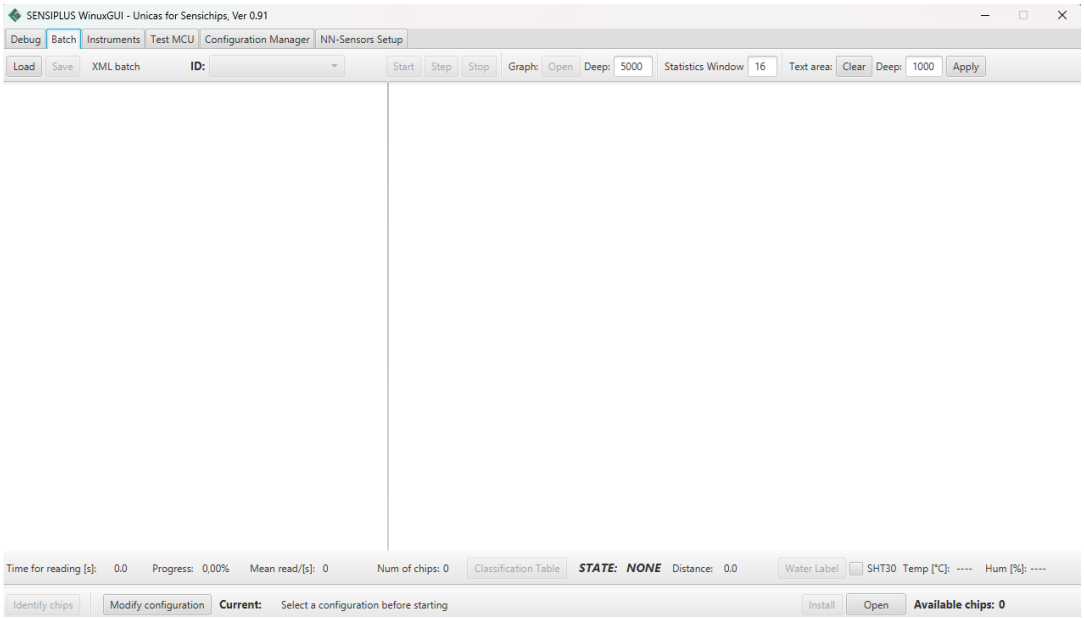


Figure 12 Batch tab view.

The Batch tab is meant to allow the user to set the desired measurement experiment configurations, furthermore, is capable to show the classification system results, all the

physical quantities measured plots, and information. In particular, from this view, it's possible to load an XML file containing all the information needed from the application to execute a given measurement experiment autonomously.

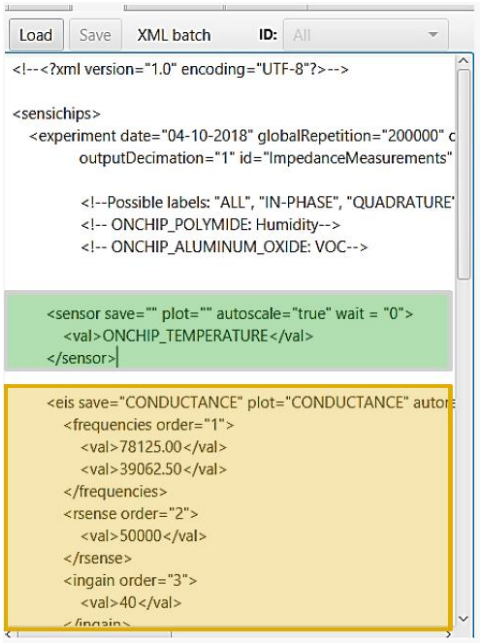


Figure 13 Batch tab with an XML configuration file loaded.

Figure 13, for example, shows an XML file containing two measurement configurations:

1. Sensor: ONCHIP_TEMPERATURE (the green one)
2. EIS measurement (the orange one)

A possible example of an XML file is shown in Figure 14.

```
<sensichips>
    <experiment delay="0" globalRepetition="1000"
    id="pippo" outputDecimation="1" >

        <eis save="ALL" plot="CAPACITANCE, RESISTANCE, QUADRATURE"
                 autorange="TRUE" autoscale="false" waitSET = "0" SETwaitGET = "0"  filter="1" >
                    <frequencies order="1">
                        <val>78125.00</val>
                    </frequencies>
                     <rsense order="2">
                        <val>5000</val>
                     </rsense>
                     <ingain order="3">
                        <val>40</val>
                     </ingain>
                     <outgain order="4">
                        <val>7</val>
                     </outgain>
                     <dcbiasP order="5">
                        <val>0</val>
                     </dcbiasP>
                     <dcbiasN order="6">
                        <val>0</val>
                     </dcbiasN>
                     <contacts order="7">
                        <val>TWO</val>
                     </contacts>
                     <modevi order="8">
                        <val>VOUT_IIN</val>
                     </modevi>
                     <harmonic order="9">
                        <val>FIRST_HARMONIC</val>
                     </harmonic>
                     <inport order="10">
                        <val>PORT_HP</val>
                     </inport>
                     <outport order="11">
                        <val>PORT_HP</val>
                     </outport>
                    <SequentialMode order="12">
                        <val>0</val>
                    </SequentialMode>
                <phaseShift order="13">
                    <val>Quadrants, 0, IN_PHASE</val>
                </phaseShift>
        </eis>
    </experiment>
</sensichips>
```

Figure 14 XML file containing a possible EIS measurements configuration.

For more details about the XML file tags refer to Appendix A.

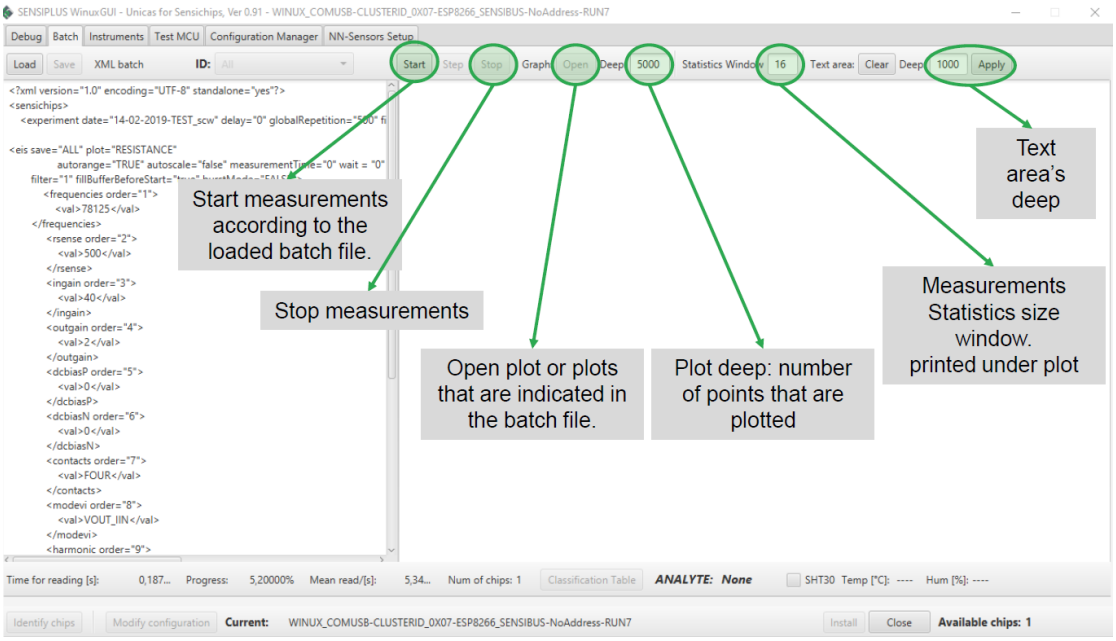


Figure 15 Batch tab view a detailed description.

Regarding all other settings contained in the Batch tab view refer to Figure 15. Figure 16 and Figure 17 represent a possible output from an EIS measurement (Figure 16) and a POT one (Figure 17). As can be seen there are different information that can be retrieved as statistical ones, graphical representation, and a text area with all the acquired samples.

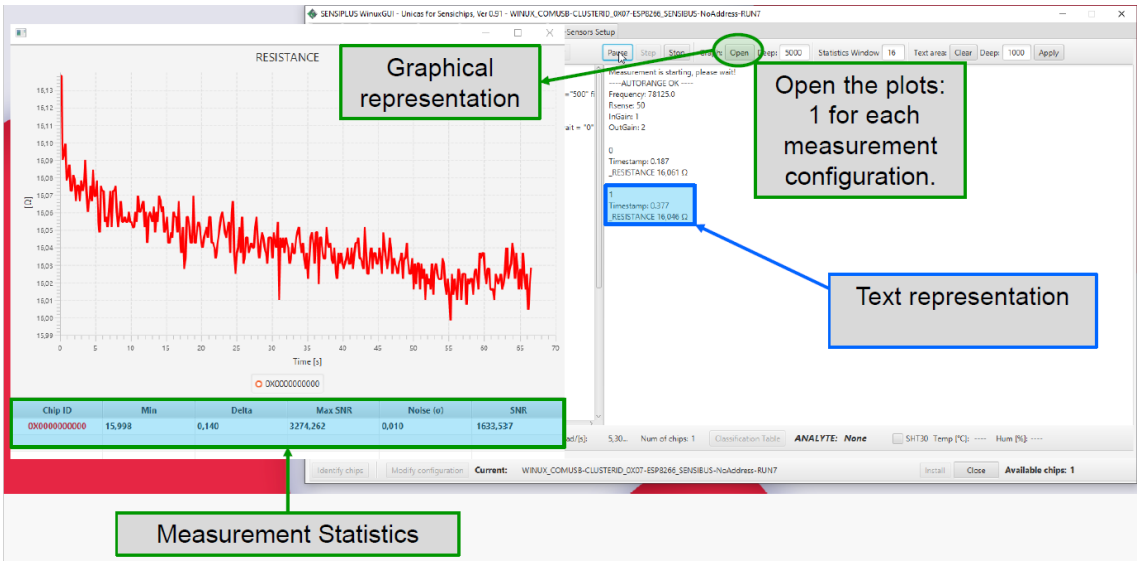


Figure 16 EIS measurements information.

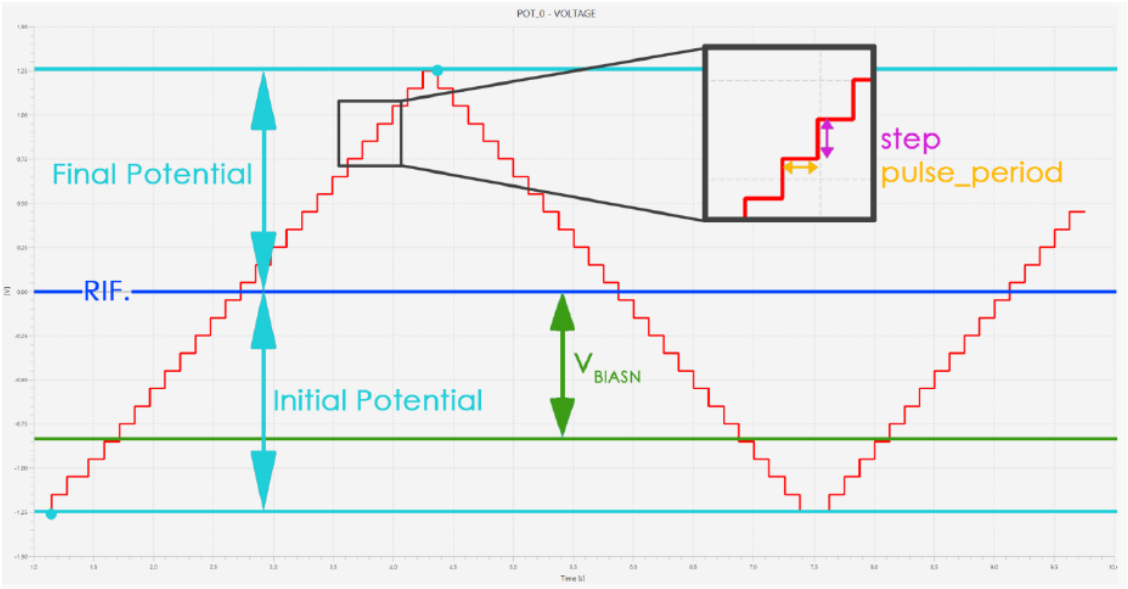


Figure 17 POT staircase measurements information.

Lastly, from the Batch view, it is possible to show the results of the classification system. Obviously, all the measurements and classifications system information is contained in the given XML loaded file. From Figure 18 can be seen all the Classification system information, in particular, the three highlighted columns (Concentration, Confidence, and Reliability) show the output class information.

It is important to note that the Concentration column is still a "work in progress" feature and so at the time I'm writing it has not been implemented yet. Regarding Confidence, the column reports the probability that the given output class is the current flowing one. The Reliability column, instead, is meant to give an idea of the possible truthfulness of the obtained result, showing the reliability (in terms of accuracy) obtained during the learning phase. For more details about the others classification system information refer to Section 2.5.

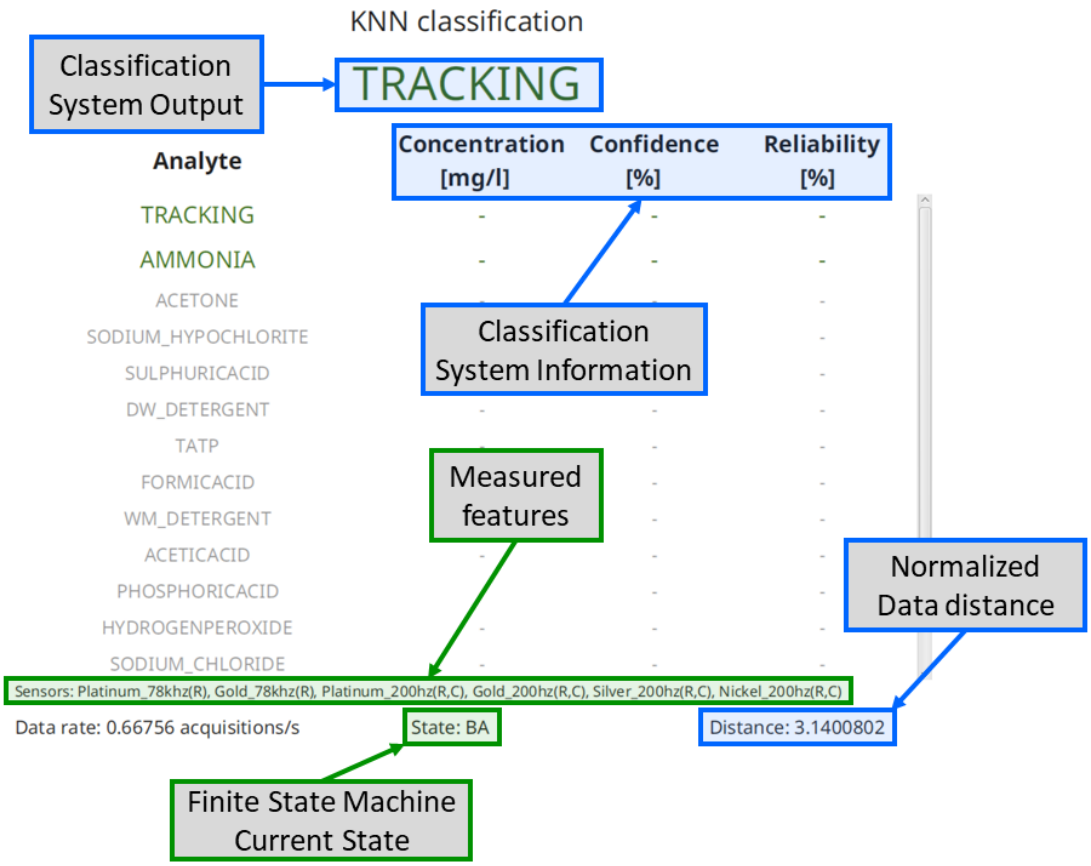


Figure 18 Classification Systems results information.

1.5 Embedded API

Among the main objectives of the Sensichips system, particularly of the SENSIPLUS project, there is the capability to elaborate the measured data by exploiting the methodologies and techniques of artificial intelligence combined with the necessity to keep low consumption. For that reason, during this Ph.D., I've started to work on implementing the Software API on the ESP-32 MCU to exploit all the edge computing advantages, such as the speed up given by the capability to process the data on the MCU itself. For that reason, the first step that has been made is porting the software API described in Section 1.3 into the C language in order to be executed by the MCU itself. During the entire porting work, the main goal has been to make the whole API's modules as efficient and lightweight as possible.

At the same time as the porting, tests were started on various artificial intelligence algorithms to understand which algorithm would give the best performance in terms of energy consumption and computational impact.

CHAPTER 2. CONTAMINANTS DETECTIONS AND RECOGNITIONS USING MACHINE LEARNING TECHNIQUES

In this chapter, I will present the development of an end-to-end identification system (from sensing to classification) capable of detecting a predefined set of substances considered dangerous and indicative of an anomalous use of wastewater. The developed system follows the paradigms of the Internet of Things (IoT) benefiting from the application of Artificial Intelligence (AI) and Machine Learning (ML) techniques to exploit, as much as possible, the information from all the acquired data. The proposed measurement system is based on a proprietary embedded IoT-ready Micro-Analytical Sensing Platform (MASP) of 1.5 mW power absorption and 1.5 x 1.5 mm of the size called Smart Cable Water (SCW) and based on the SENSIPLUS microchip (see Chapter 1 for more details). The SCW and SENSIPLUS microchips have been developed and designed by Sensichips s.r.l. company.

2. State of the Art

In the scientific research field, many activities have been carried out related to sensing technology, developing a sensor able to respond to the presence of a given contaminant selectively, and data analysis with the purpose of achieving detection and classification. Given the complexity of the applications and given the difficulties in using laboratory instruments to perform reliable water quality monitoring the main target of several works, [17] [18] is represented by those systems that are reliable, compact, low cost, and low power. The desired capability of a sensor to respond differently to different substances is quite rare both in commercial and ad-hoc sensors.

A commonly adopted solution is the usage of a sensor array. Sensing technologies for water pollution monitoring have been widely studied in the scientific literature, e.g., see the extensive reviews in [19] [20] [21].
The proposed methods rely on applying electrodes of different metals [22] or being covered by sensing films [23]. Measurements from sensors are usually based on a frequency domain technique, known as Electrochemical Impedance Spectroscopy (EIS) [24] [25] [26], which evaluates the response of an electrochemical cell to a low amplitude sinusoidal perturbation.
Given the large amount of data acquired, data analysis for contaminants classification is typically based on Artificial Intelligence (AI) approaches. For example, in [27], Artificial Neural Network (ANN) and Principal Component Regression are used to estimate nitrate concentration in groundwater, while in [22] partial least square discriminant analysis is applied to detect explosive precursors in wastewater.
In [28] the authors describe an AI approach for water quality monitoring. Another example can be found in [29], where the authors combine an AI algorithm with fractional derivative methods and the main algorithm adopted for machine learning is the Support Vector Regression Model (SVR).
Also, deep learning solutions have been proposed. In [30], for example, Long Short Term Memory (LSTM) and Convolutional Neural Networks (CNN) were adopted for the detection and classification of chemical substances in sea water.

2.1 Scientific Contribution

From the review of the state of the art related to the application of machine learning to water analysis, it emerges that the problem of anomaly detection has not been addressed. In a classification system, ignoring the anomalies means making it unusable in a real context, as the system would not be able to react correctly to substances not taken into consideration during the training phase, potentially generating false positives. In a wastewater analysis scenario, the open issues are mainly related to the complex and expensive equipment often required, unsuitable for the IoT and pervasive paradigm, and the lack of an anomaly detection step. This work proposes a possible solution to both of these issues.
In terms of IoT readiness, is proposed the adoption of the SENSIPLUS chip, a proprietary device developed by the Italian company Sensichips s.r.l., which has been proven to be effective in reliable measurements for pollutant detection in air and water [31] [32] [33] [34] [35]. The SENSIPLUS chip, together with a commercial Micro

Control Unit (MCU), becomes a low-power, low-cost, and IoT-ready miniaturized sensing platform.

The second issue is tackled with a double-stage classification system: an anomaly detector and a multiclass classifier, starting from the idea that only a closed set of substances are interesting while others are simply interferants and do not need to be classified. The anomaly detection allows stating if the analyzed substance can be one of interest or something else (for simplicity, unknown). Whenever such a module declares that the substance is not an anomaly, the multiclass classifier module is activated, and its computational burden is included in the system load. The combination of both modules permits having a substantial false positive reduction while keeping a very high accuracy value for the substances of interest. The combination of the developed platform and the new concept of supervised double-stage classification represents the main contribution of this work to the state of the art.

2.2 Measurement SetUp

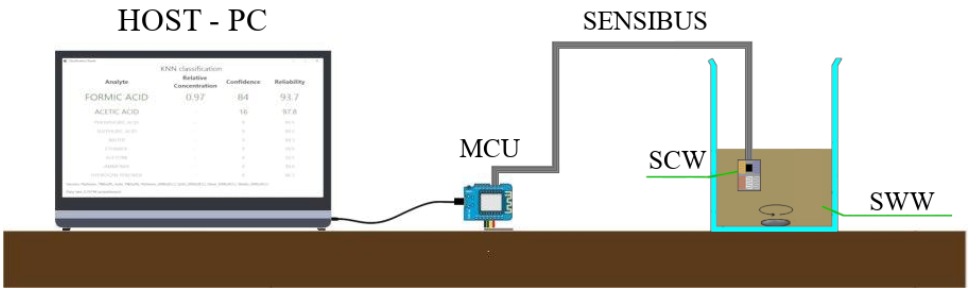


Figure 19 Measurement SetUp.

Figure 19 shows the overall measurement system, composed of a measurement chain, a magnetic stirrer, and a 300 ml beaker. The measurement chain is based on a proprietary IoT-ready ecosystem of Sensichips s.r.l. composed of a sensor layer named Smart Cable Water (SCW) which basically is a tiny analytical sensing platform of 1.5mW power absorption. The SCW is directly connected to an MCU (that in our case is an ESP32/ESP8266) which connects the sensor layer with the host PC. At the core of the SCW, there is the SENSIPLUS, the Sensichips micro-chip capable to interrogate on-chip and off-chip sensors with its versatile and accurate Electrical Impedance Spectrometer (EIS). In this way, it is possible to perform measurements working with

multiple sensors, and in particular, the SCW system has been endowed with 6 Interdigitated Electrodes (IDEs). The magnetic stirrer equipped with a 25 mm anchor simulates the water flow. It is worth noting that all measurements have been performed under the same water-stirring conditions. More in detail, the anchor rotation has been set to 50 rpm. It is important to say that the anchor rotation speed has been chosen to avoid air bubbles in 100 ml of water trying to minimize measurement noise. Finally, in order to have any measure independent from the others, the beaker has been cleaned with soap and fresh water.

The physical principle adopted to achieve the goal of detecting and recognizing a given set of substances of interest is to exploit the RedOx dynamics of catalytic noble metals. The electrical equivalent circuit of two electrodes flooded in a water solution is named Randles and it is represented in Figure 20. As can be seen from the electrical circuit, each electrode is mainly represented by a double layer capacitance Cd and a faradic resistance Rf that take into account the interface between the water solution (called bulk) and the electrode itself, for that reason, it depends on the electrode composition, geometry, bulk composition, etc. The equivalent resistance of the bulk, named Re, mainly depends on the bulk composition and the electrode area.

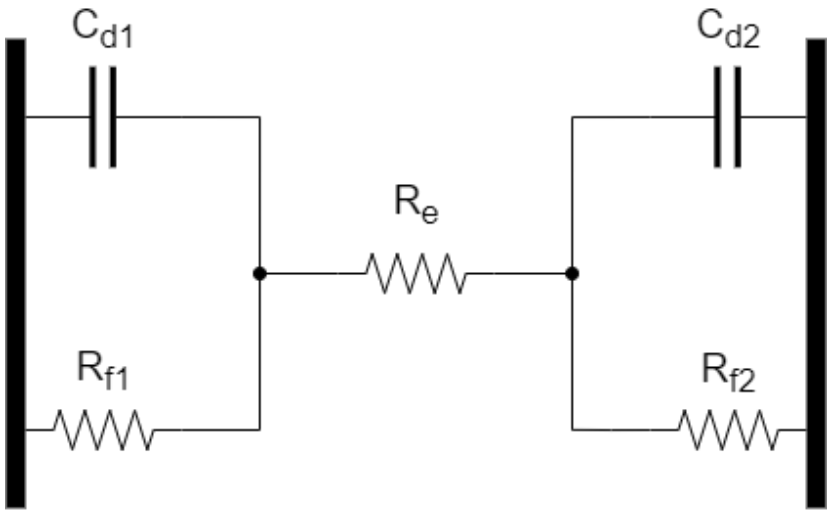


Figure 20 Randles equivalent circuit.

For that reason, it was decided to functionalize the six IDEs of the SCW by coating them with different metals:

- Gold
- Copper
- Silver
- Nickel
- Palladium
- Platinum

Of which the first five (from Gold to Palladium) IDEs are 3 x 7 mm each, while the last one (Platinum) is 12 x 8 mm (see Figure 21).

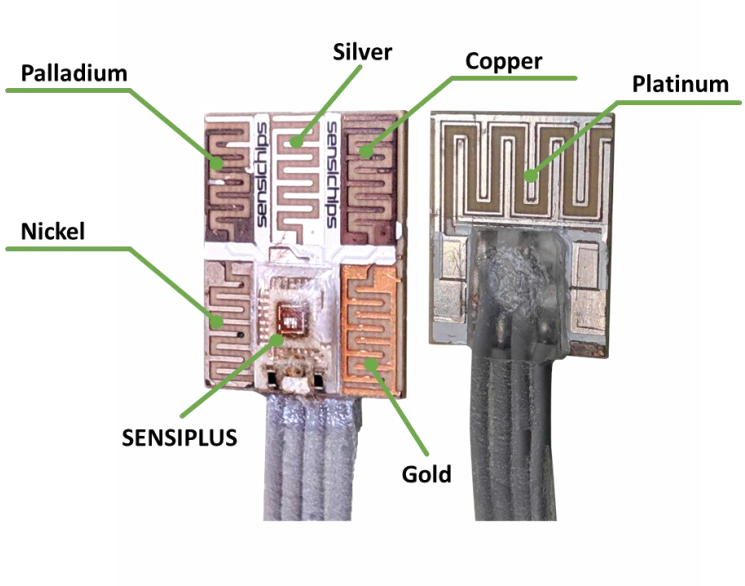


Figure 21 Smart Cable Water IDEs.

2.3 Features Selection

One of the main problems related to Machine Learning is related to feature selection. Feature selection is primarily focused on removing noninformative or redundant predictors from the model and for that reason, according to the electrical equivalent circuit described in the previous section, we choose to collect the following features:

- Resistance at 78KHz frequencies, for the Gold and Platinum IDEs.
- Resistance and Capacitance at 200Hz frequencies, regarding the Gold, Platinum, Silver, and Nickel.

Regards the Palladium and Copper IDEs, they have not been used. It is worth specifying that regards the palladium there were problems related to the coating which made the IDE unusable. As far as copper is concerned, instead, based on a preliminary qualitative campaign it has emerged that the Copper IDE's measurement has resulted to have less sensitive to the contaminants under test, and, therefore, Resistance and Capacitance values have been accordingly discarded.
The main idea behind the chosen features regards the different behavior that the equivalent circuit has at low and high frequencies.

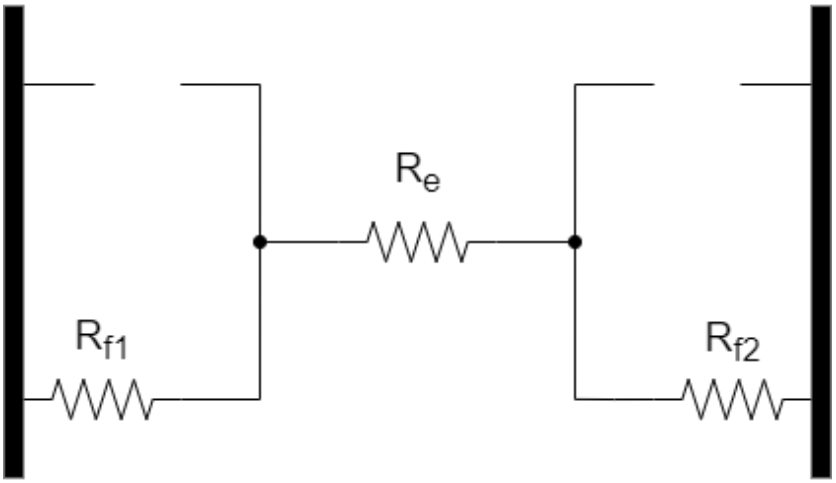


Figure 22 Randles at low frequency

In particular, at the low frequencies, the two capacitance Cd have a high impedance and can be represented as an open circuit (see Figure 22) and so the measurements depend both on the faradic resistance that on the bulk resistance (Re). It is important to note that, as depicted in the previous section (Section 2.2), the faradic resistance mainly depends on the electrode composition, area, and geometry. For that reason, to get a contribution from all the compositions and geometry, all the available IDEs (except the Copper and Palladium ones) have been used.

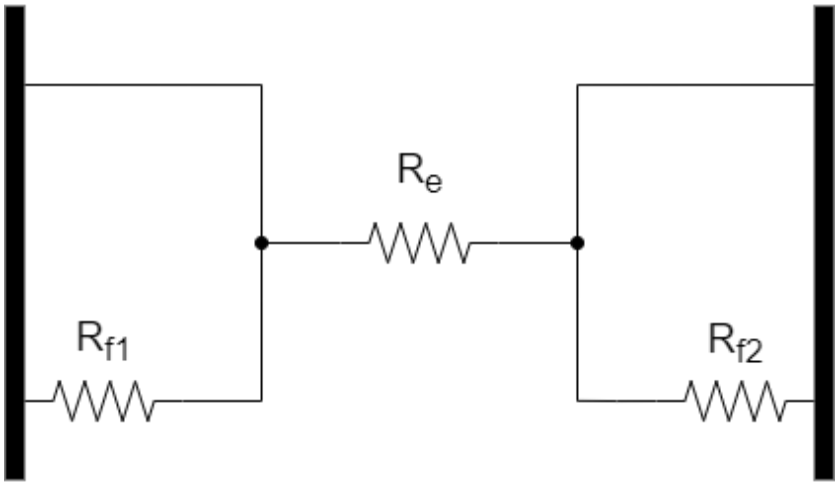


Figure 23 Randles at high frequency

At the high frequencies, otherwise, according to the Randles circuit the two Cd present a low impedance and for that reason can be seen as a short circuit (see Figure 23) and so the measurements depend mainly on the bulk resistance. In this case, unlike what was said on low frequencies, the measurement mainly depends on the electrode's surface area. For that reason, have been used only two IDEs: Platinum which is the one with the greatest surface area, and one (Gold) from the three remaining IDEs that share the same surface area.

2.4 Data Collection

The main goal of the proposed system is to be able to detect and recognize a given set of substances that flows in the wastewater. Ideally, the first step to do in order to achieve the expected result is to build a training set by acquiring all the data set measurements directly in a controlled drain of a sewage network. However, this is not a viable solution mainly from two points of view:

- Measurements point of view: all measurements should be taken from the same and reliable conditions however due to the instability typical of the sewage background environmental composition, it is impossible to reach an acceptable level of reliability conditions.
- Health point of view: due to the presence of viruses, bacteria, and other dangers, operating directly in the sewage network would represent biological hazards.

To solve the listed problems, we decide to use synthetic wastewater (SWW) with the purpose of simulating the sewage composition. The adopted recipe to produce the SWW is inspired by a simplified version of the one created in [36]. Moreover, in order to better reproduce a real wastewater scenario, the pH of every batch of the SWW has been corrected accordingly to [37] measurements of the real wastewater. For a more detailed chemical composition of the Synthetic Waste Water refer to Table 5.
Given the complexity of the real scenario, where every day there are plenty of different substances that flow through the water, in order to build a robust system capable to face all those kinds of problems, the collected dataset has been divided into two main groups: the substances of interest (group 1) and a set of interferent ones (group 2).

Table 5 Synthetic Waste Water Composition

Compounds	**(mg/l)**
Fertilizer	91.74
Ammonium Chloride	12.75
Sodium Acetate Trihydrate	131.64
Magnesium Hydrogen Phosphate Trihydrate	29.02
Monopotassium Phosphate	23.4
Iron (II) Sulfate Heptahydrate	5.80
Starch	122.00
Milk Powder	116.19
Yeast	52.24
Soy Oil	29.02

The following list contains the substances of Group 1:

- Acetic Acid
- Acetone
- Ethanol
- Ammonia
- Formic Acid
- Sulphuric Acid
- Hydrogen Peroxide
- Synthetic Waste Water

While regards the Group 2 substances there are:

- Sodium Hypochlorite
- Sodium Chloride
- Dish Wash Detergent
- Wash Machine Detergent
- Nelsen

We have chosen this type of substance as interferences since they are most likely among the ones that can be found most frequently in wastewater since they come mainly from domestic activities.

Regarding the data collection, it is mainly divided into two phases:

- *Warm-Up phase:* in this phase, in order to let all sensors stabilize, the first 600 samples are acquired in only SWW.
- *Measurement phase:* after the warm-up phase, the substance of interest is spilled in the SWW and, to record the entire sensor's evolution after the injection, another 1000 samples are acquired.

Regarding the measurement procedure, as depicted in Section 2.2, all the acquisitions start with 100 ml of Synthetic Wastewater inside a 300 ml beaker with an anchor used to simulate water flow, with a constant rotation speed of 50 rpm. After the first 600 samples (Warm-Up phase) the given substance is injected into the beaker using a siring. It is important to note that the measurement quantities have been chosen according to the discriminant capabilities of the sensors.

2.4.1 Data Set

For each substance, ten acquisitions of 1600 samples obtained through the measurement procedure as mentioned above have been collected, obtaining 16000 samples overall.

For evaluation purposes, the k-Fold Cross-Validation procedure has been adopted. Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. Its application generally results in a less biased or less optimistic estimate of the model efficiency than other methods, such as a simple train/test split. Usually, the first step in k-fold Cross-Validation is the random shuffle of the collected data.

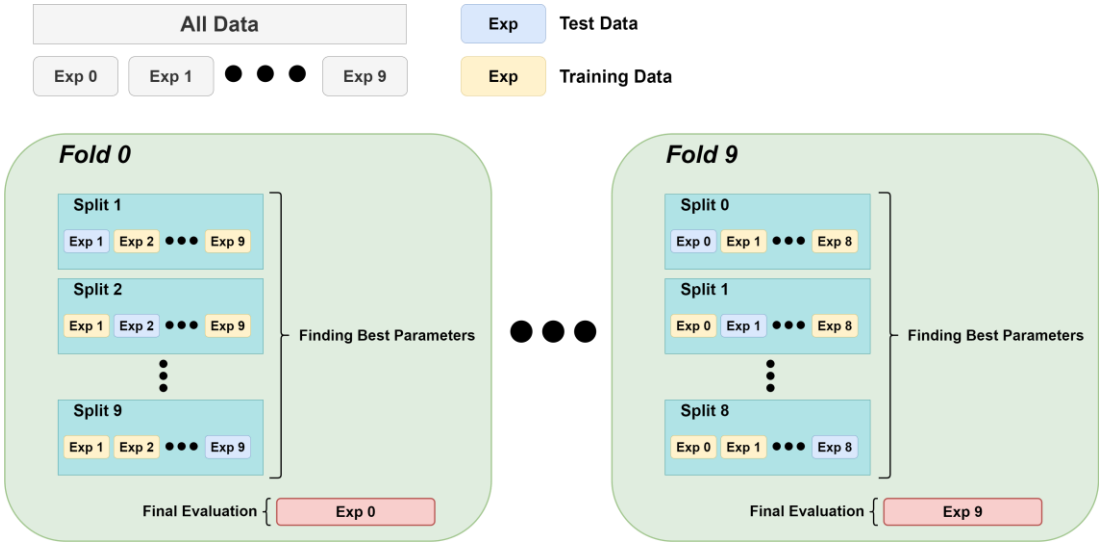


Figure 24 Data Set Structure, Exp0, 1, …, 9 means respectively acquisition 0, 1, …, 9 of all substances.

In our case, taking into account that measures belonging to the same experiment are strongly correlated, has been preferred to assume as a unit for k-fold an entire acquisition (1600 samples) of all the substances.

In order to find the best anomaly detection and multiclass classifiers model to use for the entire system, the entire Data set has been organized in ten Fold (Fold 0, Fold 1, ..., Fold 9). Each Fold contains nine additional Split (Split 0, Split 1, ..., Split 9) and one final evaluation test. The given Split is organized like the following:

- Training data: used to train both anomaly detection and multiclass classifier model.
- Test data: used to find the best model's hyperparameters of both anomaly detection and multiclass classifier.

For the final evaluation concern, it is composed of whose samples are not contained either in the Training data or in the Test data of all the Splits related to the given Fold. In order to keep things clear, has been used a fixed nomenclature: the number inside the given Fold's name indicates the experiment (data acquisition) used to perform the final evaluation, while the number inside the given Split indicates the experiment used for the related Test data. For the Training data concern, it is composed of all the experiments except the one used for the related Test Set and the one used for the final evaluation that, as said before, contains data that is unseen from both the Training and Test data of the related Fold.

For example, Fold 0 contains the Split from 1 to 9, excluding Split 0 since experiment 0 of all the substances is used to build the final evaluation test. The Test data of the

Split 1 is made of experiment 1 of all substances while the Training data is made of all the remaining experiments (excluding experiment 1 used for the Test and experiment 0 used for the final evaluation).
In this way, the Test data of the Split 2 is made by experiment 2, while the related Training data will exclude experiments 2 and 0, and so on. Regards the final evaluation of Fold 0, is made by experiment 0 of all substances.
See Figure 6 for a graphical representation of how the Data Set has been organized, starting from the structure of the Folds to the single Splits. It is worth specifying that in Figure 6 Exp 0, Exp 1, ..., Exp 9 means respectively acquisition 0, 1, ..., 9 of all substances.
Finally, it is worth specifying that regards the multiclass classifier and anomaly detection models the training, test, and final evaluation set ratio during the learning phase was respectively: 80%, 10%, and 10% with respect to the samples belonging to the Group 1 substances. Furthermore, to properly validate and test the learned anomaly detection models, the split's test set and the final evaluation data have been polluted with outlier points taken from the substances belonging to Group 2.

2.5 Classification System

In this section, the entire developed classification system is presented.
The proposed system can be schematized in two main phases:

- Data Processing
- Classification System

As can be seen in Figure 25, the data processing phase is composed of a Finite State Machine (FSM) that takes care to normalize the input data (that comes directly from the sensors) and detect if a substance has been spilled, and give it as input to the classification system that takes care to recognize the given substance.

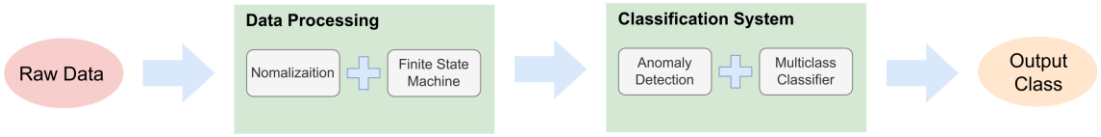


Figure 25 Classification System

It is important to notice that the developed system, as already said before, needs to be capable to work in a real wastewater scenario. In this kind of complex environment, there are many problems to be faced. Firstly, the system needs to be robust with respect

to all measurement degradations resulting from interaction with such an aggressive environment, for example, sensor drift, noise, etc.
Secondly, another problem that must be faced, is the number of different substances that everyday flows in a sewage network. For that reason, developing a system capable to recognize a given set of substances is not enough. Most likely, this kind of system will end up confusing one of the countless substances that flow into the wastewater with one of the sets of interest, generating a wrong classification, and consequently, a false alarm.
To solve these problems depicted so far has been decided to interpose a finite state machine and an anomaly detection system before the multiclass classification system, with the aim to reject all the unknown substances (outliers) and all those signals deriving from the interaction with the environment.

2.5.1 Data Processing

The Data Processing phase is formed by a Finite State Machine, which aims to build a robust baseline signal to normalize the raw data coming directly from the sensors' measurements and then pass the normalized sample to the detection phase.
The baseline signal is generated by the union of the FSM with the application of an Exponential Moving Average (EMA) according to the following equation:

$$\boldsymbol{b}_t = \begin{cases} \boldsymbol{s}_t, & t = 0 \\ \boldsymbol{b}_{t-1}, & t > 0, S \in \{BS, BSP\} \\ \alpha \boldsymbol{s}_t + (1-\alpha) \cdot \boldsymbol{s}_{t-1}, & t > 0, S \in \{WT, BA, BT\} \end{cases}$$

Equation 1

Where WT, BA, BT, BSP, and BS are the states of the FSM that are, respectively: Wait (WT), Baseline Acquisition (BA), Baseline Tracking (BT), Baseline Suspended (BSP), and Baseline Stopped (BS).

Regarding the EMA, the α parameter is given by $\frac{1}{EMA_c}$, while s_t is a vector filled by the sensors' data that comes directly from the measurements. EMA_c is the EMA coefficient that, in our case, has been empirically set to 25. Regarding the normalization value, it is given by the following formula:

$$\boldsymbol{f}_t = \frac{\boldsymbol{s}_t}{\boldsymbol{b}_t}$$

Equation 2

Where the $\boldsymbol{f}_t$ is the normalized features vector, while s_t is the sensors data vector and the $\boldsymbol{b}_t$ is the baseline signal computed as described by Equation 1.

Figure 26 shows the entire FSM system. In particular, t is the current time sample, while τ is a threshold that, in our case, has been empirically set equal to 0.05. Regards the d_t parameters, it represents the Euclidean distance between the normalized features vector $\boldsymbol{f}_t$ and the unit vector $\boldsymbol{u}$ (a vector of ones) in a 10-dimensional space that is the size of the vector $\boldsymbol{s}_t$ (see Equation 3).

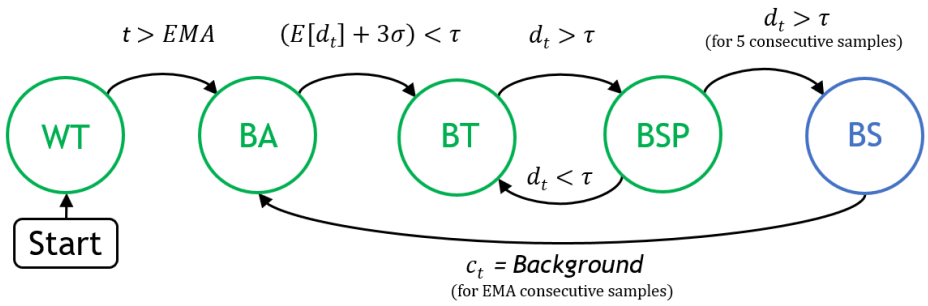
$$\boldsymbol{d}_t = \|\boldsymbol{f}_t - \boldsymbol{u}\|$$

Equation 3

The reason behind the choice to compute the distance with respect to the unit vector is given by Equation 2, indeed it is clear that the vector $\boldsymbol{f}_t$ when $\boldsymbol{b}_t$ is equal to $\boldsymbol{s}_t$ is the unit vector. For that reason, the Euclidean distance has been computed so that when $\boldsymbol{b}_t$ is perfectly tracking the signal $\boldsymbol{s}_t$, the distance $\boldsymbol{d}_t$ results to be equal to zero.

As shown in Figure 27, the state of the FSM can change according to a given rule. The main idea behind the defined rules is the purpose of building a robust baseline capable of coping with the main problems related to data processing, for example, the variability between sensors/chips, sensor drift, environmental noise, interferences, etc. In this sense, the first two states (WT and BA) guarantee that the baseline is not affected by noise and/or interferences.

Once we are in the BT state, we first must grant that single measurements spike is not confused as a substance passage, and this is done with the rules between the BT, BSP, and BS states then, once we are in the BS state, we can pass the normalized features vector to the classification system.



Symbols	
t	time sequence
d_t	distance of s_t from b_t
EMA	Exponential Moving Average coefficient = 25
τ	the threshold (0.05)
c_t	classification of sample t
$E[d_t]$	the mean over last EMA
σ	the std dev over last EMA sample

States		Output
WT	WaiT	BKG
BA	Baseline Acquisition	BKG
BT	Baseline Tracking	BKG
BSP	Baseline SusPended	BKG
BS	Baseline Stopped	The sample is forwarded to the classification module

Figure 26 Finite State Machine

The entire system depicted so far is shown in Figure 27.

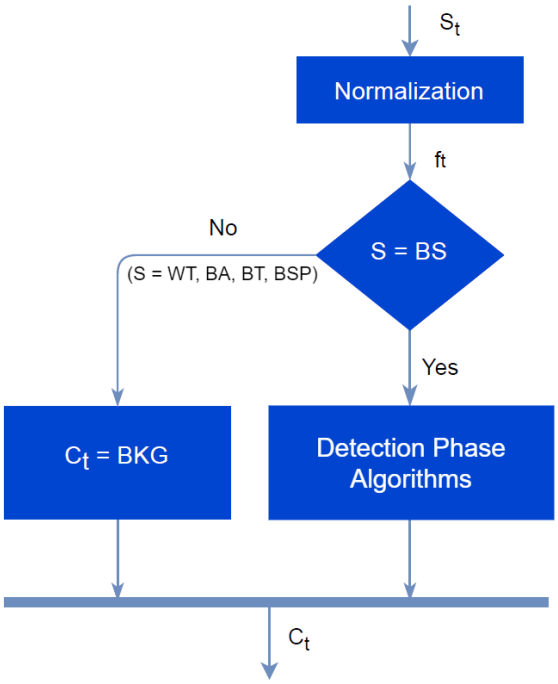


Figure 27 Finite State Machine Flow Chart

Where S indicates the state of the FSM, C_t is the classification of the sample at the time t and BKG is the background substance.
It is important to show why it was necessary to design a finite state machine in order to be able to build a robust baseline. Indeed, before the FSM, the baseline was computed through a simple Exponential Moving Average (EMA) according to the following equation:

$$\boldsymbol{b}_t = \begin{cases} \boldsymbol{s}_t, & t = 0 \\ \alpha \boldsymbol{s}_t + (1-\alpha) \cdot \boldsymbol{s}_{t-1}, & t > 0 \end{cases}$$

The main problems with this type of baseline are that:

- Is unable to adequately track the sensor drift.
- Is unable to represent a good reference value when there is the presence of a substance.
- It works in counter-phase during the wash-out (when the substance goes away).

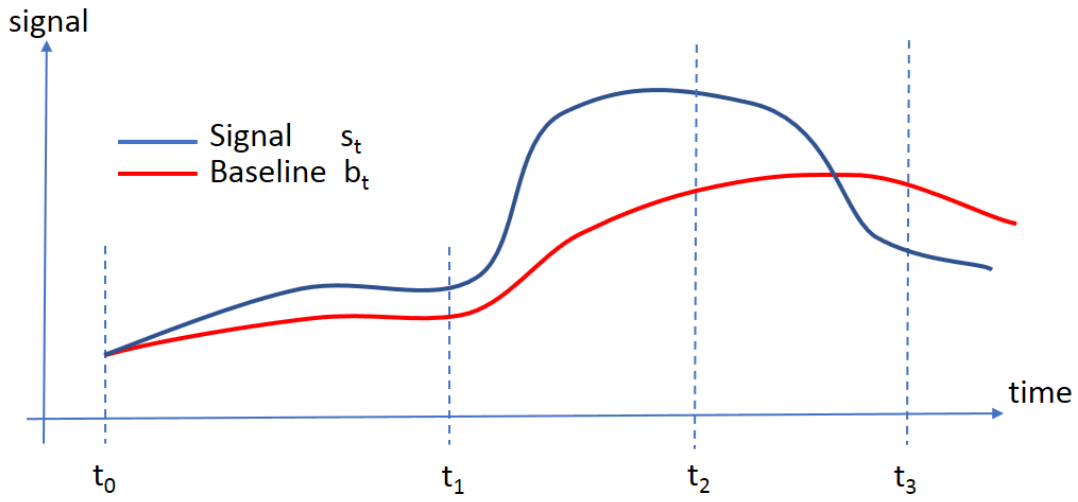


Figure 28 Behavior of the old baseline tracking system.

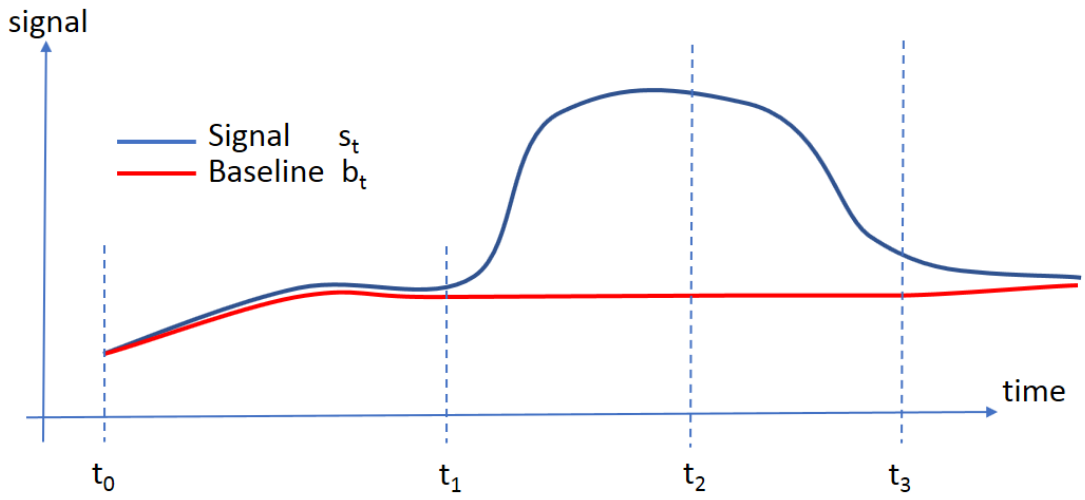


Figure 29 Behavior of the new baseline tracking system.

Figure 28 and Figure 29 represent the behaviors of the old baseline and the new baseline. In particular, four sectors can be identified:

- $[t_0, t_1]$: when the acquisition starts and there is only the background substance (SWW in our case).
- $[t_1, t_2]$: when a given substance is injected and the sensors start to react.
- $[t_2, t_3]$: when the substance is going away.
- $[t_3,\ t_\infty)$: when the substance "disappears".

As can be seen from Figure 28, in the first sector the baseline is a bit distant from the sensor signal. The distance increases during the first part of the second sector (that is good for classification purpose) but, at some point, start to vanish because the baseline keeps updating by trying to track the sensor signal (that is bad). Finally, in the last two sectors, it generates an anti-phase behavior. The described behaviors are not good from the classification point of view because all the oscillations of the features vector $\boldsymbol{f}_t$ (see Equation 2) worsen the performance of the classification system.

As can be seen from Figure 29, most of the problems related to the EMA baseline have been solved by the usage of an FSM. In this regard during the Ph.D. studies a comparison between the two, previously described, baseline tracking methods have been made [38].

In particular, the comparison study has been made by using a Multi-Layer Perceptron (MLP) network. The used dataset is described in Table 6, the results obtained with the old system are reported in Table 7, and the ones obtained with the newly are shown in Table 8. To notice that Table 7 and Table 8 have reported the Size of the Hidden Layer

(SHL), Mean global accuracy (M), Standard Deviation over the global accuracy (SD), and Synthetic Wastewater accuracy (SWW).

Table 6 DataSet Samples for different classes

Substances	Class	Samples			
		Total	Training	Validation	Test
Ethanol	(1)	9371	5183	3174	1014
Sodium Hypochlorite	(2)	9122	5059	3073	990
Sulphuric Acid	(3)	9162	5079	3071	1012
Dish Wash Detergent	(4)	9172	5078	3081	1013
Sodium Chloride	(5)	9162	5078	3071	1013
Synthetic Wastewater	(6)	9165	5079	3072	1014
Total		55154	30556	18542	6056

Has can be seeing the best results obtained with the old baseline tracking system have been achieved by an MLP with 16 hidden neurons. The best mean accuracy was 82.64% with a 1.0541% of standard deviation and an accuracy of 98.40% for the SWW. Regards the new solution, the best results have been obtained by an MLP with 16 hidden neurons. In this case, the best mean accuracy is significantly improved, by reaching 97.20% with a 0.443% of standard deviation. Finally, the SWW achieved a 99.99% of accuracy.

Table 7 Global results for an MLP using the old baseline tracking method.

SHL	M	SD	SWW
16	**0.8264**	**0.10541**	**0.9840**
32	0.7795	0.10832	0.9362
64	0.8226	0.11651	0.9512

Table 8 Global results for an MLP using the new baseline tracking method.

SHL	M	SD	SWW
16	**0.9720**	**0.0443**	**0.9999**
32	0.9720	0.0443	0.9981
64	0.9719	0.0443	0.9989

The obtained results show that the new baseline tracking method improves the performance of the classification both in terms of global accuracy that in terms of a false positive reduction filter by not confusing the sensor's noise and or drift caused by the SWW with any of the trained substances.

2.5.2 Classification System

Obviously, in a real scenario where plenty of substances flow in the sewerage network, it is crucial to be able to distinguish between the substances of interest and the other ones.

In this sense, the main goal of this phase is to determine if the given detected substance is one of interest, and then to be able to correctly predict the name of the substance. The classification phase is basically divided into two main parts:

- Anomaly Detection
- Multiclass Classification

Let’s see more in detail.

Anomaly Detection

Regards the anomaly detection algorithms, we can mainly distinguish them in two approaches:

- Outlier Detection
- Novelty Detection

In the outlier detection algorithms, the training data contains a small portion of the outlier’s samples. In this case, the estimators try to fit the regions where the training data is the most concentrated, ignoring the deviant observations.

In the novelty detection algorithms, the training data are not polluted with outlier samples. In this context, we want to determine whether a new observation is an outlier or an inlier. In this sense, an outlier is also called a novelty.

Our case is best represented by the novelty detection approaches according to our data set and the application field. This is because, in our application field, we want to discard all those substances that are usually present in the sewage system, and we want to recognize only the substances of interest that represent a minimum part of the

substances that can be found in the wastewater. To have as complete a point of view as possible, we trained and tested anomaly detection models built with both novelty and outlier approaches:

- Novelty Detection: One-class SVM, Local Outlier Factor (LOF) and KNN
- Outlier Detection: Elliptic Envelope and Isolation Forest

All the algorithms have been taken from the sci-kit learn library [39] except for the KNN, which has been taken from the Python Outlier Detection (PyOD) library [40].
As described in Section 2.4.1 we have organized the entire data set into ten Folds, each of which contains nine Training phases made by training and a validation set. For the outlier detection data set concern, it is the same as depicted in Section 2.4.1 with the addition of some outlier samples in the training set (about 10%).

Multiclass Classification

Regards the multiclass classification, instead, we have trained and optimized the accuracy of a KNN on the described data set. It's important to notice that, unlike anomaly detection, the training and validation sets are formed with only the samples of the substances of interest.
In both cases, anomaly detection and multiclass classification, the grid search approaches have been chosen to optimize the models' hyperparameters. All the model's parameters are detailed in Table 9 and Table 10.
Finally, the entire system, composed of the data processing and the classification system, is shown in Figure 30. Eventually, to find out the best anomaly and multiclass classifier model, the ten Folds of the data set (see Section 2.4.1 for more details) has been used. Once the best model of each classifier has been found, the entire system has been tested over the test data.
Now, that the entire system has been depicted, it is important to point out that the proposed classification system does not relay over any pattern nor trajectory recognition, time series, or, in other words, it is time-independent. This feature allows us to build an IoT-ready system capable of detecting and recognizing, the given spilled substance based only on the current sample, as shown in Figure 30.
In this sense, we can refer to our system as an IoT-ready platform for real-time pollutant spilling detection.

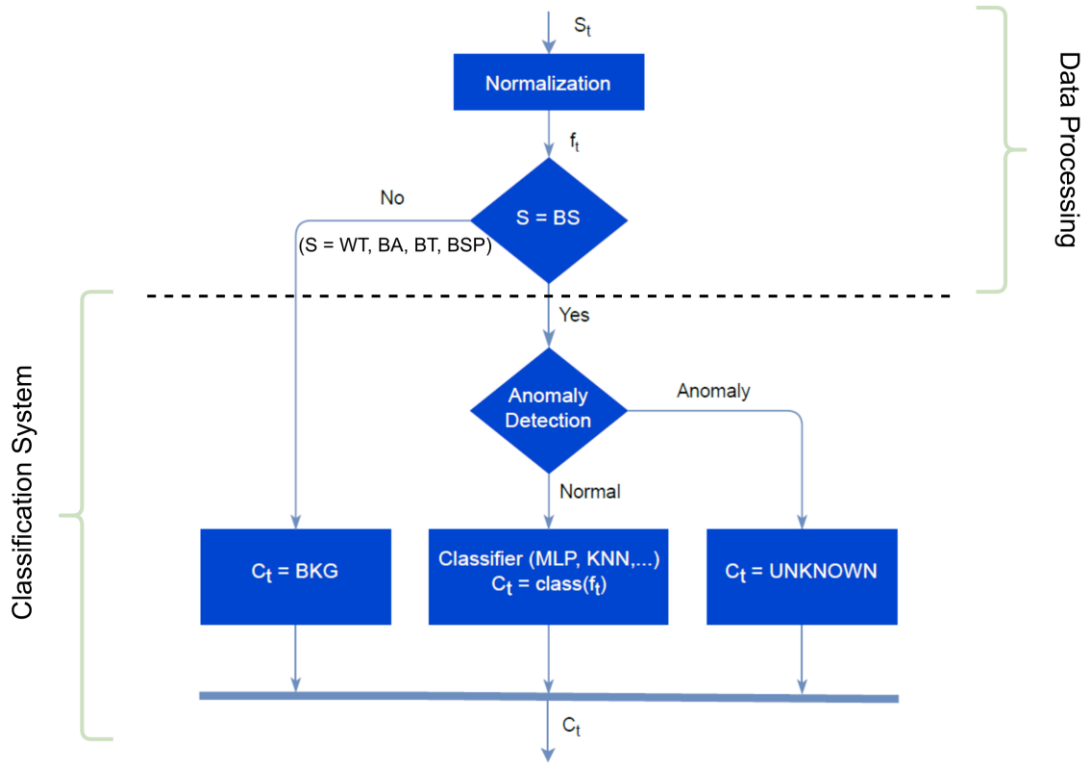


Figure 30 Entire System Flow Chart

Table 9 Anomaly Detection Models Parameters

Classifier	Parameters	
KNN	Contamination	[0.01, 0.05, 0.1, …, 0.5]
	N neighbors	[10, 100, 200, …, 500]
SVM	ν	[0.01, 0.05, 0.1, …, 0.5]
	Kernel	Radial basis function
	γ	[auto, scale, 0.01, 0.05, 0.15, …, 1.0]
Local Outlier Factor	Contamination	[0.01, 0.05, 0.1, …, 0.5]
	N neighbors	[10, 100, 200, …, 500]
Elliptic Envelope	Contamination	[0.01, 0.05, 0.1, …, 0.5]
Isolation Forest	Contamination	[auto, 0.01, 0.05, 0.1, …, 0.5]
	N estimators	[50, 100, 150, …, 500]

Table 10 Multiclass Classification Model Parameters

Classifier	Parameters	
KNN	Algorithm	Ball tree
	N neighbors	[10, 100, 150, …, 500]
	Weights	[uniform, distance]

2.5.3 Learning Procedure

Regards the training and test phase, Algorithm 1 and Algorithm 2 show the pseudo-code of the Training and Test procedure. Regards the training procedure, as already described in the previous sections, it is the same for both anomaly detection and multiclass classifiers. For the test procedure concern, instead, it has been built in order to be able to test either the entire system (anomaly detection and multiclass classifier) rather than only the multiclass classifier one. For that reason, the Test procedure takes as input extra parameters "doAnomaly" which serves to decide if the test must be performed over only the multiclass model (case doAnomaly = FALSE) or on both anomaly detection and multiclass models (case doAnomaly = TRUE). In the latter case, the online classification procedure (Algorithm 3) is called. It is worth specifying that Algorithm 3 represents the procedure implemented on the end-to-end system in order to perform online tests of the entire system.

Algorithm 1 Training Procedure

Algorithm 1: Training procedure

input: A dataset $\mathcal{F}$ representing a single Fold, list of classifiers to train, hyperparameters
output: Best Classifier

begin
  $\mathcal{F}_n = normalizeDataSet(\mathcal{F})$;
  for clf in $classifiers$ **do**
    $X_{train}, Y_{train} = loadTrainigData(\mathcal{F}_n)$;
    $X_{validation}, Y_{validation} = loadValidationData(\mathcal{F}_n)$;
    for $param$ in $hyperparameters$ **do**
      $clf.set\_params(param)$;
      $clf.fit(X_{train})$;
      $Y_{pred} = clf.predict(X_{validation})$;
      $Accuracy.append([clf, evaluate(Y_{pred}, Y_{validation})])$;
  $clf_{best} = getBestClf(Accuracy)$;
  return clf_{best};
end

Algorithm 2 Test Procedure

Algorithm 2: Test procedure

input: A TestSet $\mathcal{T}$, best anomaly model ($anly$), best multiclass model (clf), doAnomaly
output: [Accuracy, CM]

begin
  groundtruth = $getGroundtruth(\mathcal{T}_n)$
  for $sample$ in $\mathcal{T}_n$ **do**
    if $doAnomaly$ **then**
      $outClass.append(onlineClassidication(sample)$;
    else
      $sample_n = normalize(sample)$;
      $state = getFsmState(sample_n)$;
      if $state \neq BS$ **then**
        $outClass.append$("BKG");
      else
        $outClass.append(clf.predict(sample_n))$;
  $ConfusionMatrix = evaluate(outClass,$ groundtruth);
  return $[Accuracy, CM]$;
end

Algorithm 3 Online Classification Procedure

Algorithm 3: Online Classification procedure

input: Sample $\mathcal{S}$, anomaly detection classifier (anly), multiclass classifier (clf)
output: predicted class ($outClass$)

begin
  $\mathcal{S}_n = normalize(\mathcal{S})$;
  $state = getFsmState(\mathcal{S}_n)$;
  if $state \neq BS$ **then**
    $outClass =$ "BKG";
  else
    if $anly.predict(\mathcal{S}_n) =$ inlier **then**
      $outClass = clf.predict(\mathcal{S}_n)$;
    else
      $outClass =$ "UNKNOWN";
  return $outClass$;
end

2.6 Experimental Results

Regarding the experimental results, to test the entire system over the test set for each case (anomaly and multiclass classifier) has been chosen the best model. The following sections are reported the obtained results.

2.6.1 Anomaly Detection Results

Regards the anomaly detection models, the best results have been obtained in the Fold 0 case. As described in Section 2.4.1, in the case of Fold 0, all the experiments between 1 and 9 have been used as training and validation sets. In contrast, experiment 0 of all substances was used for the final evaluation test.

The best results are reported in Table 11, while the mean plus the standard deviation (STD) of the best algorithms, obtained over all the Splits of the Fold 0 data set are reported in Table 12. As can be noticed from the result tables, the reported algorithms can achieve almost the same results, both in the best cases rather than all the Splits of Fold 0. For that reason, it is not possible to easily declare a winner. Thus, since the application field of the proposed system best fits the novelty detection approaches, to test the entire system has been used the One-class SVM classifier. Finally, to statistically validate the obtained results, the Wilcoxon rank-sum test (α = 0.05) has been performed. Table 12, indeed, also shows the p–value of the Wilcoxon test. From the table, it is possible to see that the performance differences between the three algorithms that best perform (One-Class SVM, Elliptic Envelope, and Isolation Forest) are not statically significant (p-value > 0.05). Regarding the Local Outlier Factor (LOF) and the KNN algorithm, it is possible to notice that the p-value is < 0.05, highlighting a statistical difference in the obtained results. Finally, it is worth noticing that the Wilcoxon test has been performed by evaluating all chosen figures of merit (Accuracy, F1 Score and MCC).

Table 11 Best Results Anomaly Detection

Approach	Algorithm	Accuracy	F1 Score	MCC	Parameters		Split
Novelty	One-Class SVM	95.46	0.8868	0.8675	ν	0.01	6
					Kernel	rbf	
					γ	0.45	
	KNN	59.82	0.0938	-0.2485	contamination	0.45	1
					N neighbors	10	
	LOF	86.24	0.7639	0.7123	contamination	0.01	7
					N neighbors	400	
Outlier	Elliptic Envelope	95.45	0.8864	0.8671	contamination	0.05	6
	Isolation Forest	95.47	0.8872	0.8679	contamination	0.1	4
					N estimators	350	

Table 12 Fold 0 Results

Algorithm	Accuracy	F1 Score	MCC	p-value
One-Class SVM	93.58 ± 1.71	0.8474 ± 0.0352	0.8115 ± 0.0497	-
KNN	56.51 ± 2.26	0.0902 ± 0.0018	-0.2762 ± 0.0188	5.6e-6
LOF	82.01 ± 3.46	0.7137 ± 0.0382	0.6519 ± 0.0463	5.6e-6
Elliptic Envelope	93.25 ± 1.57	0.8448 ± 0.0311	0.8069 ± 0.0445	0.4860
Isolation Forest	94.63 ± 1.30	0.8689 ± 0.0277	0.8423 ± 0.0391	0.7317

The reported figures of merit, have been computed by the following formulas:

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

$$F1\ Score = 2\frac{precision \cdot recall}{precision + recall} \quad where \quad \begin{cases} precision = \dfrac{TP}{TP + FP} \\ recall = \dfrac{TP}{TP + FN} \end{cases}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Where the True Positive (TP) is all the outlier samples classified as an outlier, True Negative (TN) are all the inliers classified as inlier, False Positive (FP) are all the inlier samples classified as an outlier, and False Negative (FN) are all the outlier classified as an inlier.

2.6.2 Multiclass Classifier Results

Regards the multiclass classifier model, the best result obtained in the Fold 0 has been achieved by the KNN algorithm, using a number of neighbors (N) equal to 10 and adopting uniform weights. The obtained accuracy is equal to 99.37%. While in terms of mean accuracy and standard deviation over the nine folds contained in Fold 0, the obtained result is (91.0 ± 5.7) %.

2.6.3 Entire System Results

Two main tests have been done to highlight the benefits obtained from anomaly detection followed by the multiclass classifier for the entire system concerns.
Once with only the multiclass classifier and once with anomaly detection plus the multiclass classifier. The obtained results are shown in Figure 31 and Figure 32.
As can be seen from the two confusion matrices, the outlier substances used are:

- Dish Wash Detergent (DW_DETERGENT)
- Nelsen (INT_NELSEN)
- Washing Machine Detergent (WM_DETERGENT)
- Sodium Chloride (SODIUM_CHLORIDE)
- Sodium Hypochlorite (SODIUM_HYPOCHLORITE)

In the case of the Multiclass classifier standalone, the outlier substances get erroneously confused, with one of the known ones generating many false positive alarms. To solve this problem, as described in the previous sections, before the multiclass classifier has been added, an anomaly detection system is capable of working as a false positive reduction filter (based on what is described in the Anomaly Detection section). As reported in Figure 32, with the addition of the anomaly detection system, most outlier samples get correctly labeled as "UNKNOWN".

More precisely, 79.4% of the outlier samples have been correctly labeled as "UNKNOWN", while the remaining 20.6%, which represents all the sodium hypochlorite samples, get mostly confused with the hydrogen peroxide (according to what is shown in Figure 31).

Finally, since the obtained results (see Table 11 and Table 12) show almost the same performance across the used algorithms, and since the application field of the proposed system is best represented by the novelty detection approach, the reported result has been used by the One-class SVM classifier.

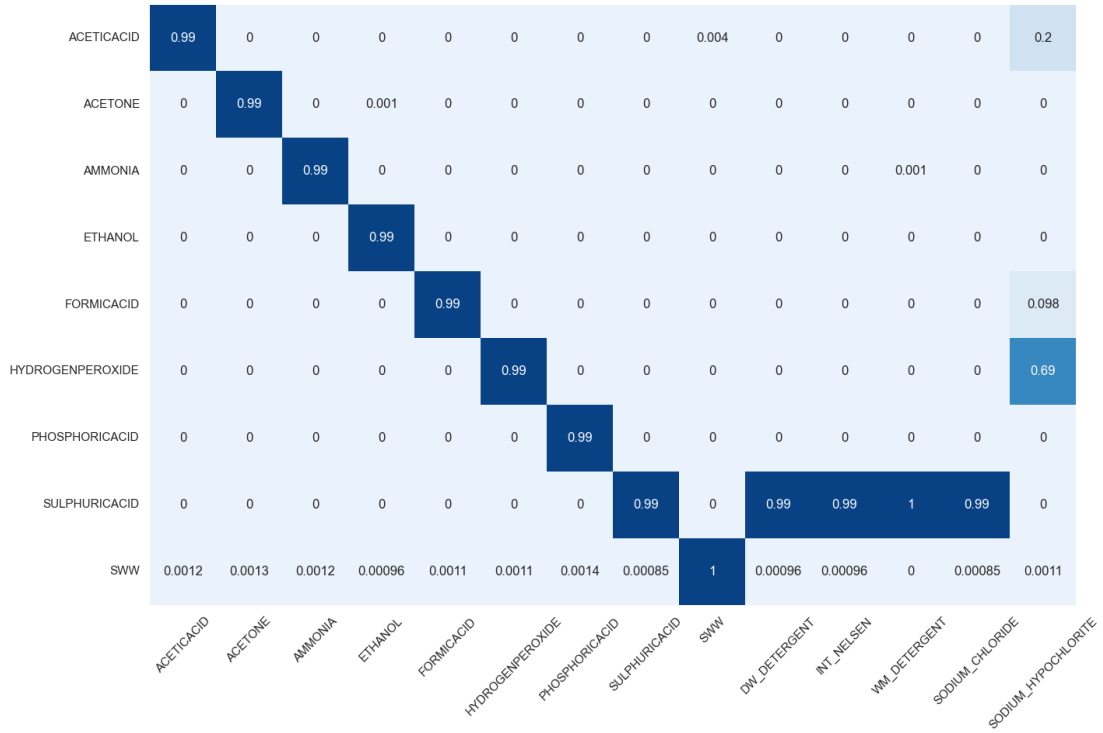


Figure 31 Multiclass Classifier Results

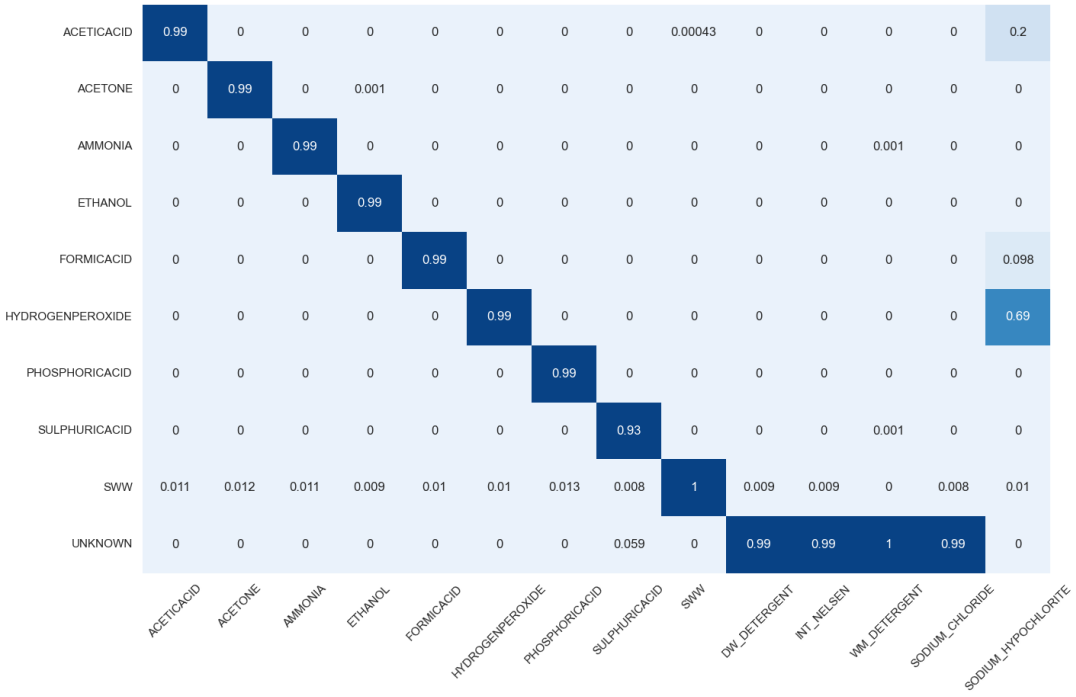


Figure 32 Entire System Results

2.7 Discussion

From the Experimental Results section can be seen that given an outlier sample as input to a multiclass classifier, the output will for sure falls under one of the known classes. This behavior led to generating a number of false alarms equal to the number of outlier samples, making the system useless in a real scenario application.

The results shown in Figure 31, clarify the drawbacks of using only the multiclass classifier system to recognize a given substance. In this case, indeed 100% of the outlier samples represented by the dish wash detergent, Nelsen, washing machine detergent, sodium chloride, and sodium hypochlorite has been mainly confused with the sulphuric acid and the hydrogen peroxide generating a great number of false alarms. For sure a multiclass classifier, used alone, cannot reject any of the outlier samples and for that reason, to solve this kind of behavior has been introduced an anomaly detection module as false alarms filter. Table 11 and Table 12 show the results obtained by the anomaly detection system. As can be seen, the performance obtained by the One-Class SVM, Elliptic Envelope, and Isolation Forest classifier are pretty similar, this means that the three algorithms can be equally used. Moreover, to statistically validate the obtained results, the Wilcoxon rank-sum test (α = 0.05) has

been performed, and the results, shown in Table 12, didn't show any significant differences. Furthermore, the chosen figure of merit shows that the anomaly detection algorithms are able to correctly distinguish between the outlier samples and the normal ones.

At this point putting the anomaly detection, and multiclass classifier systems together has been possible to reach the results shown in Figure 32. The results show as the anomaly detection system has been able to reject most of the outlier's samples (around the 80%) by labeling it as "UNKNOWN". As can be seen, all the sodium hypochlorite samples have been mainly confused with hydrogen peroxide. Even though this is a behavior that worsens the system's performance, it can still be considered an acceptable behavior.

Indeed, even if the two substances are chemically different (i.e., sodium hypochlorite is a polar substance while hydrogen peroxide is nonpolar) they have a similar oxidation potential: 1.6V for the sodium hypochlorite and 1.75V for the hydrogen peroxide [41]. Moreover, among all the substances of interest, hydrogen peroxide is the only compound that can be considered a strong oxidant (in a range that goes from +3V for the oxidants, to −3V for the reducers). This similarity is particularly evident with the measurements at 78kHz. For all those reasons, the confusion between sodium hypochlorite and hydrogen peroxide can be considered acceptable.

For the substance of interest concern, instead, Figure 32 shows as the multiclass classifier results (see Figure 31) are substantially maintained. Indeed, the overall accuracy reached by the entire system over the substances of interest is around 98.44% with a 0.93% of accuracy loss with respect to the multiclass classifier system alone (99.37%). With the consideration made, we can say that the improvement made by putting an anomaly detection system before the multiclass classifier one, has been proven. The entire system indeed has been capable to reject around the 80% of outlier samples and correctly recognize around 98.44% of inlier samples.

Finally, given a real scenario application, it is clear to understand how the presence of an anomaly detection module is of vital importance for the utility of the system itself.

2.8 Conclusion and future developments

In conclusion, the proposed work has been meant to develop a stable and robust detection system capable of working in an aggressive environment like the one represented by the sewage network. The complex environment implies that many

different substances can be present, even those whose danger level is not significant and therefore not to be detected by the proposed system.
Nevertheless, adopting a classical supervised ML approach, whatever substance would be recognized as one of those belonging to the training set. The important novelty carried out and proven to be effective in this work is implementing a two-stage scheme to reduce false alarms by keeping the classification accuracy very high strongly.
To do that, a Finite State Machine with the intent to filter, process, and normalize the measured sensors data, and a classification system was built. The classification system is divided into two main parts, one represented by an anomaly detection classifier (in our case, the One-class SVM), that rejects all the samples belonging to the unknown substances, and one represented by the KNN multiclass classifier to recognize the given substance belonging to those of interest. From the obtained results, shown in Section 2.6, it can be seen that the developed system works as supposed, drastically reducing the classification errors given by the outlier samples and keeping accuracy on the substances of interest higher than 0.93 in all considered cases.
Regards future developments, sure, the system has to be optimized and improved to be as much as possible suitable for a real scenario application. For that reason, many tests must be performed in a real scenario (see Chapter 4 for more details) both to validate the promising results obtained in the laboratory activity and to enforce and improve the generalization property of the system itself.
Furthermore, the sodium hypochlorite confusion shown in Figure 32, as discussed in Section 2.7, suggests to us that substances with some common chemical properties could be confused by the anomaly detection system. A possible way to reduce this phenomenon as much as possible would be to investigate an optimum set of orthogonal features that can exploit the chemical differences to maximize the overall system performance.

CHAPTER 3. CONTAMINANTS DETECTIONS AND RECOGNITIONS USING AD-HOC ALGORITHMS

This Chapter presents the development of a system for detecting and classifying pollutants based on a lightweight algorithm suitable for the IoT and edge-computing paradigms. The system is based on a classifier suitable to be implemented aboard the so-called Smart Cable Water (SCW) sensor, a multi-sensor based on SENSIPLUS technology developed by Sensichips s.r.l.

The SCW endowed with six interdigitated electrodes (IDEs) is a smart sensor covered by specific sensing materials to allow the differentiation between different pollutants. To make the system as edge computing suitable as possible, has been used decomposition techniques (e.g., PCA, LDA, etc.) to compress the obtained data passing from a 10-dimensional space to a 3-dimensional space. Regarding the classification system, a straightforward model that can be implemented using very few hardware resources has been developed. As for the model parameters, on the other hand, they were learned through the use of evolutionary algorithms.

The remainder sections are organized as follows: Section 3 discusses the state of the art; Sections 3.1, 3.2, and 3.3 describe the three proposed approaches based on the Evolutionary Algorithms developed during the Ph.D. work.

3. State of the Art

Environmental pollution monitoring has the attention of many researchers and technical communities. They are mainly proposing new emerging sensors able to reliably detect pollutants by minimizing the costs, energy consumption, and size by developing new network technologies, communication standards, and new methods for data analysis. Zhuiykov [19] proposes a review of the emerging technologies for

water quality parameter monitoring. In this regard, sensors based on electrodes made with different metals [22], electrodes covered by sensing films [23], and optical sensors [42] represent the solution mainly adopted by the scientific literature. In [26] electrochemical impedance spectroscopy (EIS) is exploited for detecting Escherichia coli in river water samples. For the data analysis concern, many researchers are exploiting the advantages offered by Artificial Intelligence and Machine Learning (ML) [34] [43] [22] [35].

In particular, ML techniques are often preprocessed using Principal Component Analysis (PCA). It is important to notice that PCA is often used as a preprocessing step for dimensionality reduction before the classification. It is generally not used to develop an ad-hoc classifier as in this Ph.D. work. For example, [44] Lotfi and Keshavarz proposed a novel approach where PCA is used to reduce the number of features to detect tumors in gene expression microarray data.

In [45], instead, a comparison between PCA and SVM in fault classification for complicated industrial processes is proposed. In this regard, the experimental results showed that the PCA offers a higher classification rate for this multi-class classification case with much less computational effort based on the results obtained from the Tennessee Eastman challenge process whereas SVM classification takes longer and gets less accurate classification results.

Finally, similar approaches have been used for face recognition [46] and intrusion detection problems [47]. In most of the proposed approaches in the literature, PCA is used primarily to reduce the number of available features and has never been used to develop an ad-hoc classifier.

Evolutionary computation (EC) based algorithms have proven to be effective in solving many real-world problems characterized by large and non-linear search spaces [48] [49] [50] [51]. They have also been used to improve the performance of the basic PCA algorithm. In [52] an approach for classifying hyperspectral images has been proposed. In particular, an approach in which feature selection and extraction were combined by using a Genetic Algorithm (GA) to select the features to give as input to the PCA. Eventually, the new transformed features were provided as input to the adopted classifier.

In [53] the GA is used to identify the optimal PCA components for the automated identification of different dementia syndromes from PET images data. Then the extracted feature was given in input to an SVM classifier. The proposed approach was based on eigenvector selection and weighting reaching a test accuracy of 90% on a dataset of 210 clinical cases.

Finally, in [54] a novel PCA and GA approach for human face recognition is proposed. In this case, the PCA was used for extracting features from images with the help of covariance analysis to generate eigen components of the images, whereas the GA was used for dimensionality reduction. The proposed approach was able to achieve an accuracy of 96% on the Japanese Female Facial Expression (JAFFE) dataset. Therefore, EC-based approaches have been widely used to improve the results of the PCA procedure by suitably selecting or modifying the principal components provided by the standard algorithm. However, to the best of my knowledge, an EC-based algorithm has never been used to learn the parameters of a classification model in the feature space provided by the PCA.

3.1 Cone Based Algorithms

The proposed work represents the first step that has been made to build an EA-based system capable of recognizing a given set of substances spilled in wastewater. It is worth specifying that the data collection and the data set structure are the same that have been already described in Sections 2.4 and 0, with the only exception regards the set of substances used: Acetic Acid, Ammonia, Phosphoric Acid, Sulphuric Acid, and Synthetic Waste Water (Sww). Regards the features selection techniques and the measurement setup, have been extensively detailed in Sections 2.3 and 2.2.

3.1.1 System Architecture

The developed system is divided into two main modules. The first uses a PCA procedure to transform the input normalized data, consisting of ten electrical measures, i.e., resistance and capacity values (see Section 2.3 for additional details), into 3-D space. The main goal of this module is to simplify the original data by identifying a few uncorrelated features that maximize the data variability. This data transformation allows for the building of a simple classification model that can be implemented on very low hardware resources. It is important to mention that the implemented PCA is based on the randomized truncated SVD method [55]. Regarding the data normalization, it was performed with respect to the baseline computed as the mean value over the first 600 samples (*worm-up* samples refer to Section 2.4 for more details). In particular, the last 1000 samples of each acquisition used to build the dataset, have been normalized following Equation 4.

$$\boldsymbol{n} = \frac{\boldsymbol{x}_i}{\boldsymbol{b}_i} - 1.0 \qquad \forall\, i \in [1,2,\ldots,10]$$

Equation 4

Where $\boldsymbol{x}_i$ represents the set of the last 1000 samples of the ith acquisition, $\boldsymbol{b}_i$ is the baseline of the given acquisition and $\boldsymbol{n}$ are the normalized data.

Finally to centering the data into the origin of the references system, has been subtracted 1.0 from the obtained value. It is worth noting that since the first 600 samples have been taken with the only presence of the background substance (SWW), the background normalized samples will lay around the origin.

The output data of the first module represents the input of the second one, which consists of a *C* binary classifier, where *C* represents the number of substances to be distinguished. Each classifier, except for the background class (SWW), is based on a simple geometrical model consisting of a 3-D cone whose vertex coincides with the origin of the *xyz* references system. Regarding the SWW samples, given the normalization approach, they will stay all around the origin of the 3-D space, and for that reason have been choosing to use a sphere centered in the origin of the reference system. In this way, the sphere can be seen as a threshold on the distance of a given point P from the origin, so if a given point P falls inside the sphere, it is labeled as belonging to the SWW (see Figure 33).

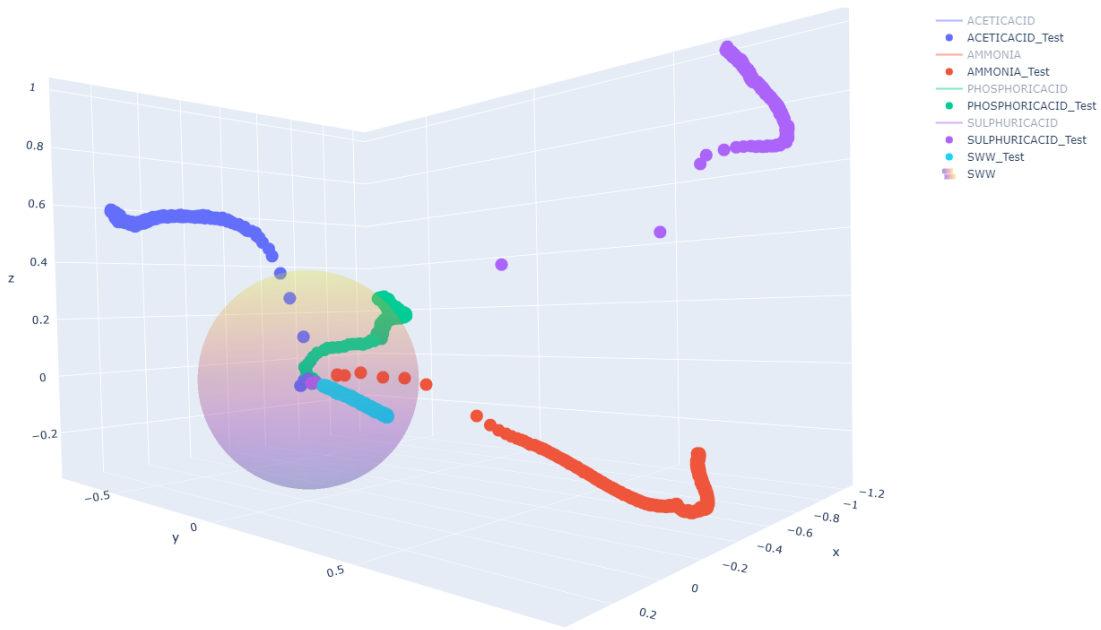


Figure 33 Background substance sphere.

This model uses the one-versus-all strategy and so given a cone Γ trained to detect the substance γ, the points internal to Γ are labeled as belonging to γ, whereas the external ones are labeled as non γ points, i.e. belonging to one of the other substances to be detected (see Figure 35). It is worth noting that the cone Γ is uniquely determined by four parameters (see Section 3.1.2 for more details) with the internal/external points providing positive/negative values for the equation. The architecture of the entire system is shown in Figure 34.

The idea behind the proposed system is to build a very simple model capable to be implemented on sensors. Indeed, once the model parameters have been learned, new data can be classified by performing a few calculations, which do not require executing any software but can be implemented directly on the available sensor hardware.

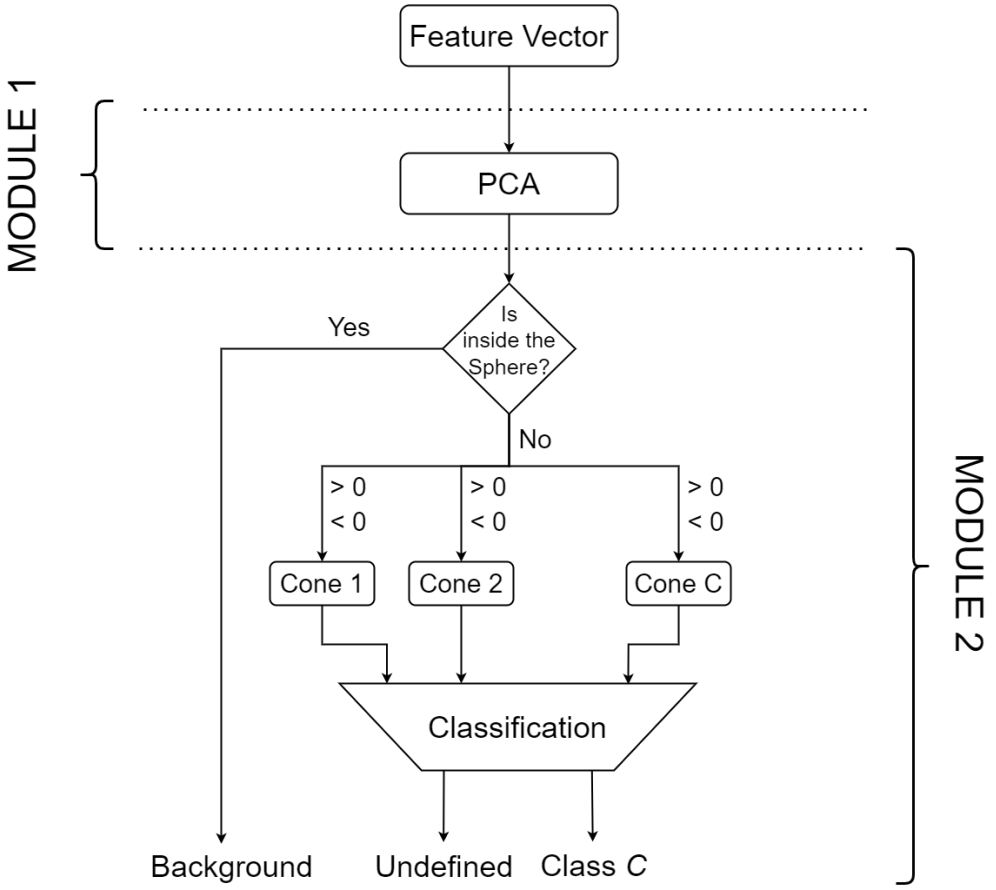


Figure 34 System Architecture

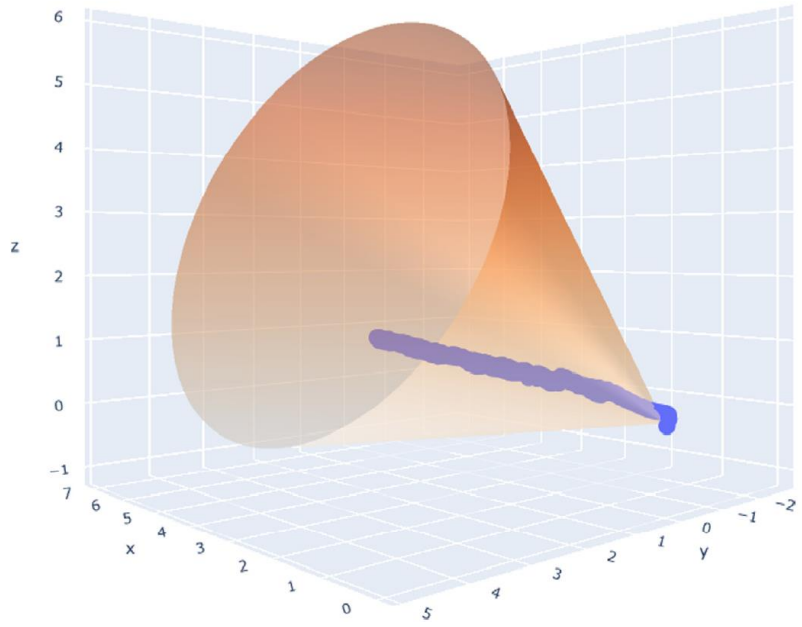


Figure 35 Representation of a cone with internal points.

Indeed, the PCA algorithm, once learned, is basically represented by an $N \times 3$ matrix, where N is the number of features extracted that in our case is ten. For that reason, a new sample $\boldsymbol{x}$ that has to be classified can be projected into the 3-Dimensional space at runtime by performing $N \times 3$ multiplications and sums.

Looking at Figure 34, after the transformation into the 3-D space the three features are used to solve the C cone equations (for more detail see Section 3.1.2), one for each learned class (substance). Those equations need $C \times 3$ multiplications and sums, where C is the number of learned substances to be discriminated. At this point, three scenarios can be faced:

1. Only an equation provides a positive value for the sample $\boldsymbol{x}$;
2. More than one equation provides a positive value for $\boldsymbol{x}$;
3. No equation provides a positive value $\boldsymbol{x}$.

In case 1, $\boldsymbol{x}$ is assigned to the class related to the equation that provides the positive value. In case 2 the distance between $\boldsymbol{x}$ and each of the bisectors of the Γ cones involved is computed, and $\boldsymbol{x}$ is assigned to the class represented by the cone whose bisector is the nearest one. Finally, in case 3 $\boldsymbol{x}$ is still recognized by the system as data representing a pollution state, but the sample is labeled as "undefined".

Given a dataset $\mathcal{D}$, the task of learning the cone Γ that best classifies the given substance γ, from the other ones can be seen as an optimization problem where the objective function to maximize is $F_s(\mathcal{D},\ l, m, n, \alpha)$. In particular, the chosen objective

function is the F-score computed on the cone represented by the parameters l, m, n and α, achieved on $\mathcal{D}$. Since the EAs have been proven effective in solving hard and non-linear problems, they have been used as a tool to learn the devised classification model. Finally, it is important to note that the training set $\mathcal{D}_t$ used during the learning process, has been decimated (and so only 100 samples per substance have been used).

3.1.2 Classification Model

The classification model is based on a geometrical cone where each substance γ is represented by a cone Γ whose vertex coincides with the *xyz* reference system origin in the 3-D space obtained by the PCA, with the internal points classified as belonging to γ and the external ones belonging to the other substances. To check if a given 3-D point *P* falls inside a cone Γ, firstly, the angle θ between *P* and the axis of Γ is computed, then the value of the angle θ is checked, and if it is less than the opening angle α of the cone Γ means that *P* falls inside it, otherwise it is outside.
Starting from the equation that defines the axis of a generic cone:

$$r: \begin{cases} x = x_0 + lt \\ y = y_0 + mt \\ z = z_0 + nt \end{cases} \quad with \quad t \in \mathbb{R} \implies v_r = (l, m, n)$$

Equation 5

Where (x_0, y_0, z_0) are the coordinates of the point P_0 representing the vertex of the cone, that in our case coincides with the origin of the reference system. At the same time (*l*, *m*, *n*) is the component with respect to the base {*i*, *j*, *k*} of a parallel vector to *r*. Moreover, the unit vector v_s between *P* and P_0 is:

$$v_s = \overrightarrow{P_0 P_1} = P_1 - P_0 = (x_1 - x_0, y_1 - y_0, z_1 - z_0)$$

Equation 6

Then given two unit vectors:

$$v_r = li + mj + nk$$
$$v_s = l'i + m'j + n'k$$

Equation 7

the cosine of the angle θ between v_r and v_s can be computed according to the following formula:

$$\cos(\theta) = \frac{v_r \cdot v_s}{\|v_r\|\|v_s\|} = \frac{ll' + mm' + nn'}{\sqrt{l^2 + m^2 + n^2}\sqrt{l'^2 + m'^2 + n'^2}}$$

Equation 8

once the value of θ has been found, if $\theta \leq \alpha$ then the data sample represented by P is assigned to the substance γ represented by the cone Γ.

3.1.3 Evolutionary Algorithms

In the last few decades, computer science researchers have widely studied the effectiveness of the natural mechanism. Indeed natural selection is, basically, the result of competition among different living beings for the resources available and needed to survive. Individuals with the features to survive better in their environment are more likely to pass their genes to the next generation. This simple but very interesting mechanism originated a new computation paradigm, Evolutionary Computation (EC). EC-based algorithms have proved their effectiveness by solving complex problems e.g., NP-hard ones, in which optimal solutions must be found in very huge search spaces.

In this regard, since we can see the problem of finding the optimal parameters for a given substance's cone in the 3-D space as an optimization problem where we want to maximize the objective function $F_s(Ð, l, m, n, \alpha)$, the generational evolutionary algorithm has been used. In particular, the algorithm starts by generating a population of P individuals, each made of four real variables (the cone parameters to be learned). Within the individuals, the ith variable is initialized by randomly generating a number in the range $[m_i, M_i]$ by using the uniform distribution function, where m_i and M_i are respectively the minima and maximum values of the ith variable.

At this point, the fitness of the P individuals is computed, and a new population is generated. The e individuals are just copied into the new population; this strategy ensures that the best individuals found along the evolutionary process are not lost. Then the remaining $(P - Ð)/2$ couples of individuals are selected by using the tournament method.

Afterward, the uniform crossover is applied to each of the selected couples with the probability p_c. Next, the mutation operator, with a probability of p_m, is applied. The p_m value has been computed to apply the modification of only one chromosome element,

corresponding to $0.251/l_c$ where l_c is the chromosome length. This value has been suggested in Ochoa [56] as the optimal mutation rate below the error threshold of replication. Finally, the newly generated individuals are added to the new population and the depicted process is repeated for N_g generations. The implemented evolutionary algorithm is shown in Algorithm 4. Further details about the evolutionary algorithms can be found in [57].

Regards the fitness function for the *i*th substance is defined by a dataset $\mathcal{D}_i$ suitably build, where the samples of the *i*th substance are labeled as belonging to class 1, and the remaining ones are labeled as 0. Then, given an individual *I*, its fitness is computed according to the formula:

$$F_s(\mathcal{D}, l_I, m_I, n_I, \alpha_I) = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

$$precision = \frac{N_{tp}}{N_{tp} + N_{fp}}$$

$$recall = \frac{N_{tp}}{N_{tp} + N_{fn}}$$

Equation 9

Where l_I, m_I, n_I and α_I are the cone parameters encoded by *I*, N_{tp} represents the number of samples belonging to the *i*th class correctly classified (i.e., true positive), N_{fp} is the number of samples erroneously classified as belonging to the *i*th class (i.e., false positive) and N_{fn} is the number of samples not correctly classified (i.e., false negative). Thus, F_s measure the accuracy of the cone represented by *I* on the training set $\mathcal{D}_i$.

Algorithm 4 Evolutionary algorithm.

input: list of parameters (Table 3), a data set $\mathcal{T}$
output: best individual found

begin
randomly *initialize* a population of $\mathcal{P}$ individuals;
evaluate the fitness of each individual;
$g = 0$;
while $g < N_g$ **do**
  copy the best e individuals in the new population;
  for $i = 0$ to $\mathcal{P}/2 - e$ **do**
    select a couple of individuals;
    replicate the selected individuals;
    if flip(p_c) **then**
      apply the crossover operator on the selected individuals;
    if flip(p_m) **then**
      perform the mutation on the offspring;
  evaluate the fitness of each individual;
  replace the old population with the new one;
  update the best individual found so far;
  $g = g + 1$;
return the best individual found;
end

The function flip(p) returns the value 1 with a probability p and the value 0 with a probability $(1 - p)$.

3.1.4 Results

To test the developed system have been used four substances: acetic acid, ammonia, phosphoric acid, sulphuric acid, and sww represent our background substance. A balanced dataset who has been extensively detailed in the Section 0 has been used. Since the proposed approach needs to evolve a cone for each substance using a one-vs-all strategy, each substance, has been built ad-hoc dataset, where all the samples of the given substance have been labeled as the target class, while the remaining ones as non-target.

Furthermore, the entire dataset has been split into two statistically independent sets: the first is made of 90% of the available samples used to learn the best cone's parameters; the second is made of the remaining 10% of the samples used to test the learned models. Totally, for each substance, to learn the best models, twenty runs have been made. For each run, the parameters of the best individuals have been stored as

the best solution provided by that run. It is worth specifying that all the reported results have been computed by averaging those of the twenty runs.
Table 13 reports the obtained cone performance in terms of accuracy and F-score, while Table 14 reports the evolutionary algorithm parameters used for all the experiments. These parameters have been found after preliminary tests.

Table 13
Cone Performance in terms of accuracy, F-score, and true-positives (TP), true-negatives (TN), false-positives (FP), and false-negatives (FN).

Substance	Accuracy	F-Score	TP	TN	FP	FN
Acetic Acid	0.84	0.54	4982	38484	6087	2447
Ammonia	0.89	0.48	2677	43528	1043	4751
Phosphoric Acid	0.88	0.47	2627	43353	1218	4802
Sulphuric Acid	0.91	0.64	4312	42850	1722	3117
Sww	0.88	0.43	2297	43650	922	5132

Table 14
The values of the parameters used in the experiments

Parameter	Symbol	value
Population size	p	100
Crossover probability	p_c	0.6
Tournament size	T	5
Elitism	e	2
Mutation probability	p_m	0.25
Mutation range	m_r	0.1
Number of Generations	N_g	500

To analyze the performance obtained by the cones found with the EA has been taken into account: accuracy, F-score (see Equation 9), true-positives (TP), true-negatives (TN), false-positives (FP), false-negatives (FN). From Table 13 can be seen that Sulphuric Acid has achieved the best accuracy. These results show that the proposed system is effective in distinguishing each substance. For the F-score values concern, these are far from optimal (1.0), meaning that the system can be improved by future updates (see Section Line Based Algorithms 3.2 and 3.3). To test the effectiveness of

the proposed system in distinguishing the various contaminants and the sww as well has been built the confusion matrix on the test samples (see Figure 36).

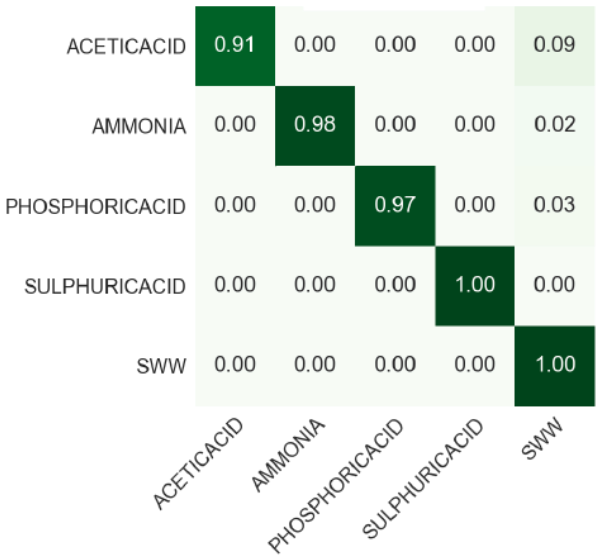


Figure 36 Confusion matrix on test data.

The results reported in Figure 36 show that the sww (as well as sulphuric acid) is correctly distinguished, and this means that no false alarms are provided by the system, in other words, the sww is never confused with a polluted state. It is also worth noting that substance samples are never confused. Finally, the overall accuracy of the system is 96.5%.

To best tests, the effectiveness of the system, a comparison between the proposed approach and more traditional machine learning algorithms has been made. In particular, the cone-based system results have been compared with those achieved by: AdaBoost (AB-J48), bagging (BAG), convolutional neural network (CNN), Support Vector Machine (SVM), Multilayer Perceptron (MLP), random forest (RF), voting (vote) and xgboost. For these classifiers, except the CNN, the implementation provided by the WEKA open-source machine learning software [58] has been used.

Regarding the parameters, the default ones provided by WEKA have been used. The CNN, instead, applies a single convolutional layer with 64 unidimensional kernels of size 3x1. After a batch normalization layer, two fully connected layers with 256 hidden neurons have been added, interleaved with a dropout fixed to 0.5. For the fully connected layers, relu and soft-max have been used as activation functions, respectively. The results shown in Table 15 show that the proposed system outperforms all the classifiers taken into account for the comparison.

Table 15 Comparison results.

Classifier	Accuracy
AB-J48	85.04
BAG	80.23
CNN	83.59
SVM	89.56
MLP	86.42
RF	85.51
Vote	88.21
Xgboost	82.12
Our method	**96.50**

3.1.5 Related Problems

Additional investigation has been made regarding the behavior of the proposed system with respect to the undefined samples. To this aim, has been analyzed the composition of these test samples and calculated for each contaminant as well as the sww the percentage w.r.t. the total number of undefined samples (2600) and the total number of samples for the given contaminant (1300 for each contaminant). Table 16 reports the obtained results. As can be seen, the acetic and phosphoric acid samples amount to about 80% of the test samples that are unclassified, i.e., they fall outside the related cone. Furthermore, these two substances are mostly unclassified, respectively 89% and 71%. Regarding ammonia, about 30% of the test samples are undefined.

Table 16 Percentages of substances

Substance	w.r.t. undefined	w.r.t. substance
Acetic Acid	44.5	89.0
Ammonia	15.5	31.0
Phosphoric Acid	34.5	71.0
Sulphuric Acid	5.5	11.0
Sww	0.0	0.0

These analyses suggest that the proposed cone-based system had some representation limits that need to be improved (see the following sections). However, the results of the sww confirm the potentiality of the proposed approach showing that the proposed system never provides false alarms.

3.2 Line Based Algorithms

As said in Section 3.1 the cone-base algorithm was the first step in the building of a lightweight EA-based classification system. After the results obtained with the cone-based algorithm, the proposed approach, named the line-based (LB) algorithm, has improved the system's performance by changing the classification model. In particular, each substance is now represented by a straight line passing through the origin of the transformed 3-D reference system, and each point is assigned to its nearest axis, according to the Euclidean distance. Unlike the cone-based (CB) system, a multiclass classifier has been implemented straightforwardly, avoiding all the problems belonging to the one-versus-all technique (e.g., N distinct and independent training set, labeling conflicts between cones, etc.).
Furthermore, the proposed approach allowed us to simplify even further the classification model, indeed, in a 3-D space, a straight line passing through the origin is represented by only three parameters (four in the case of cones). It is important to specify that regards the implementations of the evolutionary algorithm, except for the fitness function, it is the same adopted with the cone-based system, so refer to Section 3.1.3 for more details. For the fitness function concern, it is depicted in Algorithm 5.

Algorithm 5 Line-based fitness function.

input: A dataset $\mathcal{T}$ consisting of $N_{\mathcal{T}}$ points in the 3-D space, an individual I representing C lines
output: Accuracy achieved by I on $\mathcal{T}$

begin
  $N_c = 0$;
  for $i = 1$ to $N_{\mathcal{T}}$ **do**
    compute the distance between $P_i \in \mathcal{T}$ and each of the lines of I
    assign P_i to the nearest line r_n
    if $label(P_i) == label(r_n)$ **then**
      $N_c = N_c + 1$;
  return $N_c/N_{\mathcal{T}}$;
end

3.2.1 System architecture

As proposed in the CB-based system, even the LB model is a classification system divided into two steps. Firstly, a PCA transformation is applied over the 10-dimensional input data, to project them into a 3-D space. Secondly, the transformed 3-D data are classified by a simple geometrical model: a straight line passing for the origin of the 3-D reference system. Figure 37 shows the entire LB system architecture.

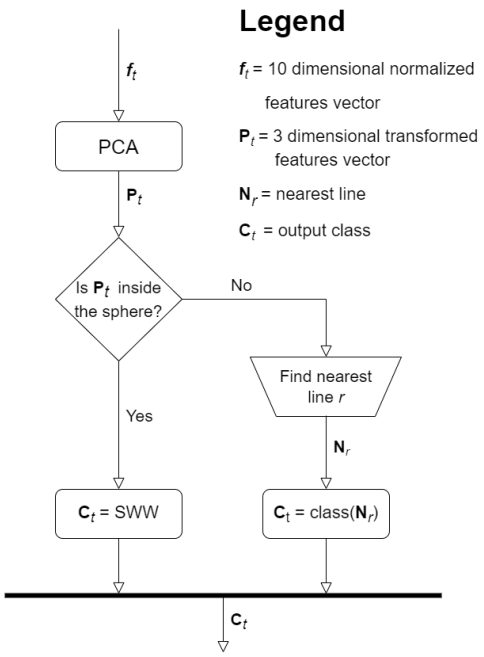


Figure 37 LB system architecture.

3.2.2 Data Transformation

The PCA data transformation aims to project the 10-dimensional input data into a 3-D space. As discussed in Section 3.1.1, the main goal of the data transformation is to allow the design of a simple and lightweight classification model. As well as done in the cone-based algorithm, the PCA decomposition has been performed via the eigen decomposition of the covariance matrix.

3.2.3 Classification Model

The multi-class classification system models the *C* class (one for each substance), except the background one (SWW in our case), with a straight line passing through the origin of the *xyz* reference system (see Figure 38). In particular, each line *r* is defined by three parameters (*l, m, n*) see Equation 5 for more details. In this way, a point in the 3-D space is labeled as belonging to the class associated with the nearest line.

As regards the classification of the SWW, as well as has been done in the cone-based system, has been used a sphere centered in the origin of the reference system, which basically is a threshold on the distance of a given point from the origin (see Figure 33).

The reason behind the choice changing from the previous classification model based on *C* geometrical cones Γ to one based on *C* lines *r* is to reduce the system's computational complexity and to resolve the problems related to the CB system depicted in Section 3.1.5. Indeed, with the line-based model, a classification task consists of only $C \times 3$ multiplications and sums, removing all the operations related to the cone-based system in the case a point *P* falls within the volume of more than one cone (say *n*). In this case, indeed, additional $n \times 3$ multiplications and sums have to be computed. Furthermore, the line-based model resolves all the problems related to the so-called "undefined" points. The entire classification model can be seen in Figure 39.

Regarding the classification model, more in detail, a point *P* in a 3-D space is assigned to belong to a given class according to a two-step procedure:

i. The distance between *P* and each of the *r* lines is computed.
ii. Point *P* is labeled with the class associated with the nearest line.

The following equations are applied to find the distance between a point P and a line r. Starting from the parametric equation of a line in the 3-D space reported in Equation 5, the plane α orthogonal to *r* that passes through the point *P* can be computed starting from the Cartesian equation of a plane in the 3-D space:

$$\alpha: ax + by + cz + d = 0$$

Equation 10

At this point the orthogonal plane to the line *r* that pass-through *P* is defined by:

$$lx + mx + nz - \left(lx_p + my_p + nz_p\right) = 0$$

Equation 11

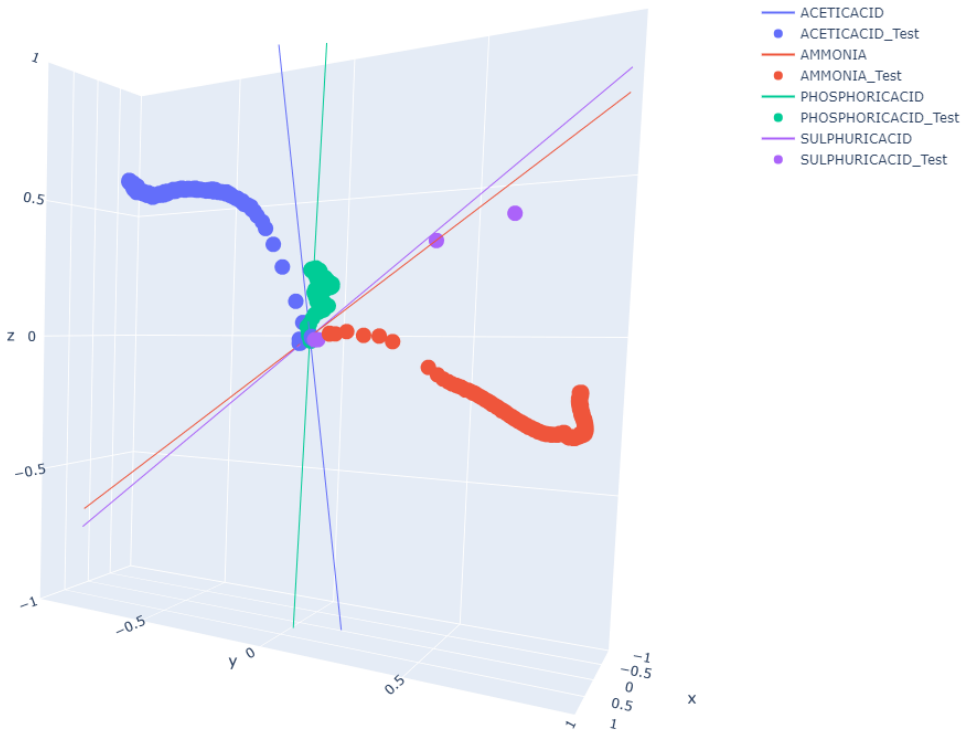


Figure 38 Line-based Model.

Substituting (x, y, z) with (x_r, y_r, z_r), Equation 11 can be resolved with respect to the parameter t (see Equation 5). Next, point $H(x_H,\ y_H,\ z_H)$ given by the intersection between the plane α and the line r can be computed as following:

$$H: \begin{cases} x_H = x_0 + lt \\ y_H = y_0 + mt \\ z_H = z_0 + nt \end{cases}$$

Equation 12

Finally, the Euclidean distance between P and r results to be:

$$d(P,r) = d(P,H) = \sqrt{(x_P - x_H)^2 + (y_P - y_H)^2 + (z_P - z_H)^2}$$

Equation 13

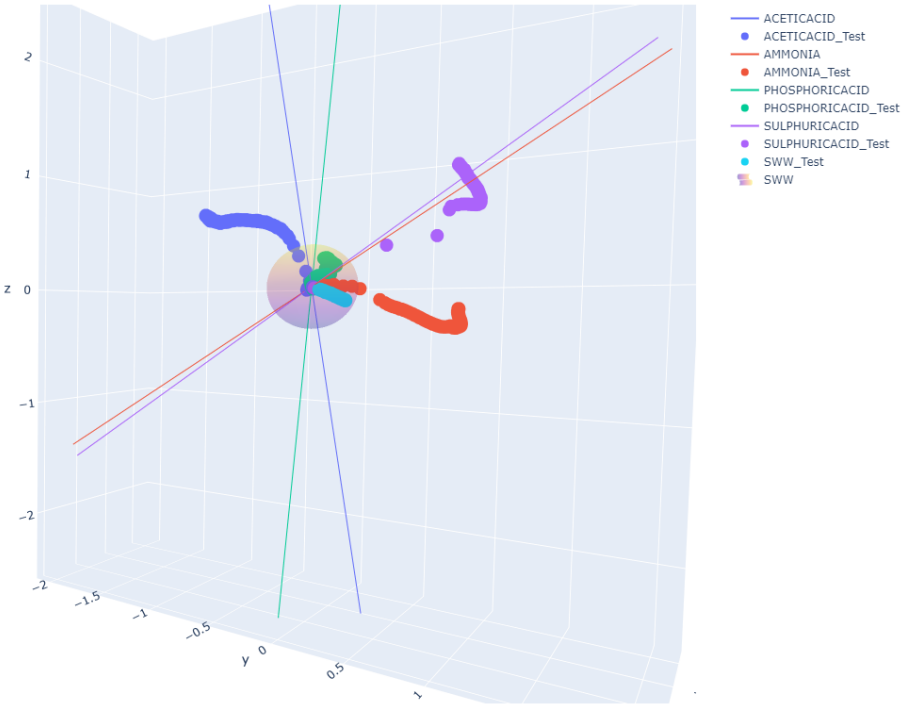


Figure 39 Entire 3-D classification model.

3.2.4 Results

To evaluate the improvement given by the line-based model regarding the cone-based one has been used the ten-fold cross-validation strategy. For this reason, the built dataset has been split into two sets:

- Training set: containing 90% of the samples of all substances except the SWW.
- Test set: containing the remaining samples and the SWW samples.

In particular, the training set has been used to compute the individual's fitness of the EA and, more in detail, for each fold, has been performed twenty runs and at the end of each run, the parameters encoded into the individual with the best fitness were stored. The used EA parameters are shown in Table 17.

At this point, the results obtained on the test set have been compared with those obtained with the cone-based model.

Table 17 Evolutionary algorithm parameters.

Parameter	Symbol	value
Population size	p	100
Crossover probability	p_c	0.6
Tournament size	T	5
Elitism	e	2
Mutation probability	p_m	0.08
Mutation range	m_r	0.1
Number of Generations	N_g	500

Figure 41 shows the confusion matrix obtained on the test data by the line-based (LB in the following) system, while Figure 40 reports the results of the cone-based (CB in the following) system.

Figure 40 Confusion Matrix CB system.

Figure 41 Confusion Matrix LB system

Looking at the reported confusion matrices, it seems that the obtained performances are similar. It is important to notice that the results related to the CB system (Figure 40) have been computed by excluding all the undefined samples.

For that reason, for the sake of simplicity,

Table 16 is reported in the following contains all the information related to the undefined points.

Substance	w.r.t. undefined	w.r.t. substance
Acetic Acid	44.5	89.0
Ammonia	15.5	31.0
Phosphoric Acid	34.5	71.0
Sulphuric Acid	5.5	11.0
Sww	0.0	0.0

Now, looking at the table above, it should be clear to see the improvements in the LB system. Indeed, the line-based algorithm correctly identifies most of the substances and, most importantly, does not present any undefined points. At this point considering that the CB system on one hand has been capable of achieving similar performance on the labeled substances, while on the other hand producing a large number of undefined

samples, it can be possible to say that the LB system brings a big improvement with respect to the CB one.
Finally, Table 18 shows an updated version of the comparison results previously reported in Table 15.

Table 18 Comparison results.

Classifier	Accuracy
AB-J48	85.04
BAG	80.23
CNN	83.59
SVM	89.56
MLP	86.42
RF	85.51
Vote	88.21
Xgboost	82.12
CB system	96.50
LB system	**99.06**

3.2.5 Further Work

Further work has been made on the LB system. In particular, has been made a study to evaluate the effectiveness of the usage of the polar coordinates rather than the cartesian one used by the LB model; another study concerns the usage of the Linear Discriminant Analysis (LDA) rather than the PCA and lastly has been evaluated a different initialization procedure for the initial population of the evolutionary algorithm named smart initialization.
Firstly, regarding the dimensionality reduction technique, a comparison between the PCA and the Linear Discriminant Analysis (LDA) algorithms has been made and secondly, a further classification model simplification has been completed.

3.2.6 Polar coordinates

The main idea behind using the polar coordinates rather than the cartesian ones is to simplify the computational complexity of the training process by reducing the space solution in which the EA must find the best solution.

In particular, the idea is to represent each substance with a line r defined by the polar coordinates θ, ϕ, and ρ. However, since the ρ parameter specifies the distance from the origin that for classification purposes, is irrelevant, it can be taken as a constant value, and so only the θ and ϕ parameters must be evolved. In this way, the computational complexity of the LB model is decreased further. Indeed, while the Cartesian LB model has to evolve and store $C \times 3$ coefficients, in the polar LB (PB in the following) version, only $C \times 2$ parameters need to be evolved and stored.

The main idea behind this simplification is to improve the performance of the classification system by reducing the solution space in which to search for the best parameters. Regards the classification model's equations, they are the same as depicted in Section 3.2.3 with the only exception that the parameters (l, m, n) must be computed as:

$$\begin{cases} l = \rho \sin \emptyset \cos \theta \\ m = \rho \sin \emptyset \sin \theta \\ n = \rho \cos \emptyset \end{cases} \quad with \quad \begin{array}{l} \rho \in [0, +\infty] \\ \theta \in [0, 2\pi) \\ \emptyset \in [0, \ \pi] \end{array}$$

Equation 14

Figure 42 shows a 3-D view of the entire model. The evolutionary algorithm and the fitness function refer respectively to the one depicted in Algorithm 4 and Algorithm 5; regarding the EA parameters they are the same reported in Table 17.

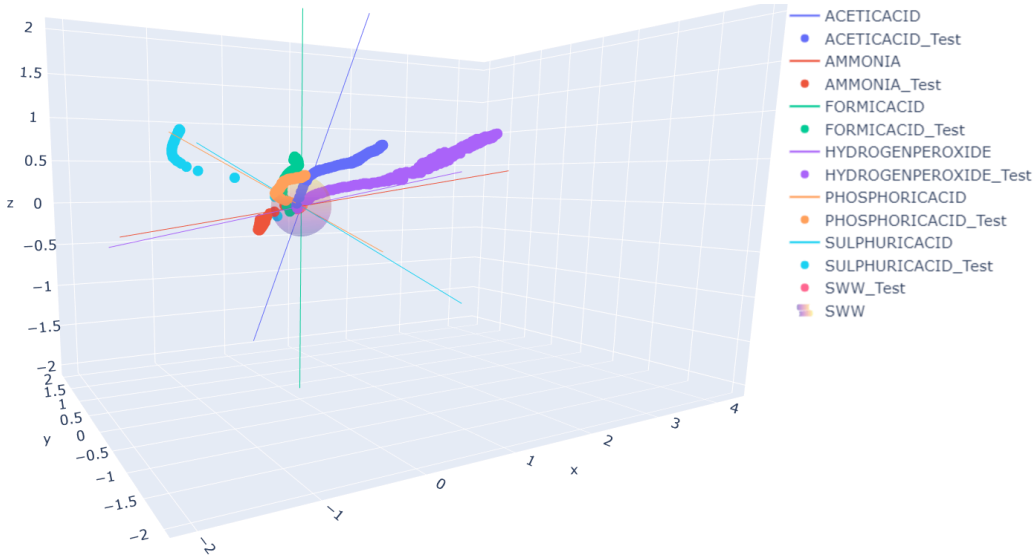


Figure 42 Spherical LB model 3-D view.

3.2.7 Smart Initialization

Smart initialization is a different initialization procedure for the initial population of the evolutionary algorithm. In particular, this procedure initializes a given fraction of the individuals in the initial population using the information provided by the training set. Thus, given an individual to be initialized, each couple of real values (the parameters θ and ϕ) representing an axis a is initialized by randomly choosing a sample s from the training set belonging to the class (substance) of a and setting the values in such a way that a passes through s, see Algorithm 6 for more details.

Algorithm 6 Individual initialization.

input: A dataset $\mathcal{T}$ consisting of $N_{\mathcal{T}}$ points in the 3-D space, an individual I representing C lines to be initialized
output: I initialized

begin
  $N_c = 0$;
  for $i = 1$ to C **do**
    randomly select from $\mathcal{T}$ a sample s_i belonging to the class i
    compute θ_i and ϕ_i in such a way that the i-th axis of I passes through s_i
  return I ;
end

To obtain the parameters θ and ϕ that represents the axis a, starting from the sample s, since each sample is represented by the x, y, and z coordinates the parameters θ and ϕ can be computed according to the following formula:

$$\emptyset = \cos^{-1}\frac{z}{\rho} \quad \text{with} \quad (x,y,z) \neq (0,0,0)$$

$$\theta = \begin{cases} \frac{\pi}{2} & \text{if} \quad x = 0, y > 0 \\ \frac{3\pi}{2} & \text{if} \quad x = 0, y < 0 \\ \text{not defined} & \text{if} \quad x = 0, y = 0 \\ \tan^{-1}\frac{y}{x} & \text{if} \quad x > 0, y \geq 0 \\ \tan^{-1}\frac{y}{x} + 2\pi & \text{if} \quad x > 0, y < 0 \text{ or if } x < 0, y > 0 \\ \tan^{-1}\frac{y}{x} + \pi & \text{if} \quad x < 0, y \leq 0 \end{cases}$$

Equation 15

3.2.8 Results

To evaluate the proposed system's effectiveness, three sets of experiments have been performed using the substances listed previously. The results of the PB system with those achieved with the cartesian one have been compared in the first set of experiments. In the second set, the two decomposition techniques LDA and PCA have been tested and have evaluated the smart initialization. Lastly, in the third set has been compared the PB system's obtained results with those of four well-known and widely used classification algorithms.

The substances used during all the set of experiments are listed in the following:

- Acetic Acid (AA)
- Ammonia (AMM)
- Phosphoric Acid (PA)
- Hydrogen Peroxide (HP)
- Formic Acid (FA)
- Sulphuric Acid (SA)
- Synthetic Waste Water (SWW)

First experiment results

To evaluate the improvement brought by using the polar coordinates has been compared the results obtained by the PB with those of the previously described LB system. In particular, the confusion matrices reported in Figure 43 and Figure 44 show the results obtained respectively with the PB and LB system. As can be seen, both approaches confused very few pollutants samples with the SWW, confirming the effectiveness of developing a system capable of detecting pollutants in water.

Whereas in terms of overall accuracy, it is clear that the PB approach outperformed the LB one, indeed the PB's best overall accuracy achieved was 0.86, while the LB one was 0.42. The PB results (Figure 43) show confusion between phosphoric and acetic acid and between formic and acetic acid. In contrast, the LB one (Figure 44), in addition to the acid's confusion, shows a confusion between the hydrogen peroxide and the formic acid.

It is important to notice that, while acid-acid confusion can be considered "acceptable", cause sensors tend to respond to the acids with similar patterns, peroxide-acid confusion is much less acceptable. Those results, together with the one related to the

overall achieved accuracy, confirm that the PB approach is more effective than the LB one in discriminating the pollutants analyzed in this experiment.

Figure 43 Confusion matrix PB system.

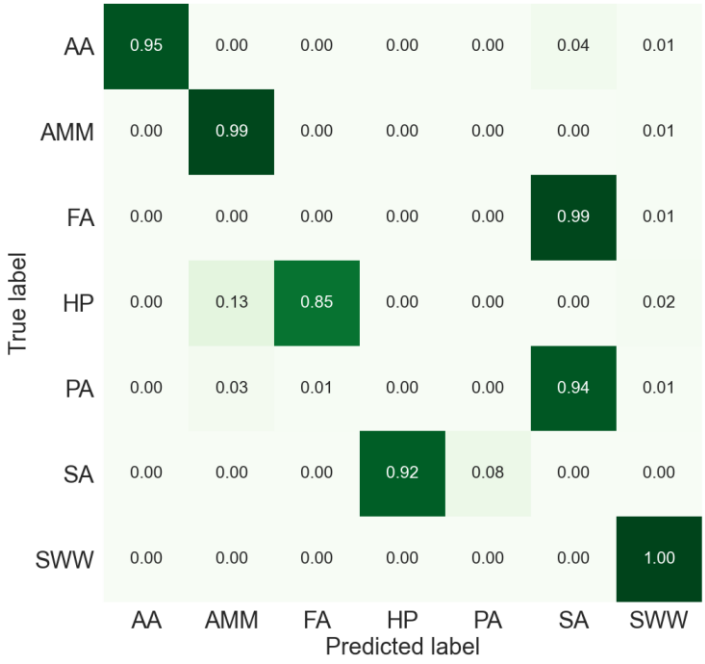


Figure 44 Confusion matrix LB system.

Second experiment results

In the second set of experiments, the results obtained by using the LDA technique rather than the PCA have been compared, and has been evaluated the results obtained by using information from the training set to initialize the individuals in the initial population.

PCA vs LDA

The LDA algorithm is a dimensionality reduction technique that uses a supervised procedure to find a linear combination of the features in the original space that allow a better class separation. To evaluate the effectiveness of this transformation, the results obtained with both LDA and PCA approaches have been compared to reduce the feature space from 10 to 3D space. Figure 45 shows the results obtained using the LDA algorithm. Comparing the PCA results (see Figure 43) with that shown in Figure 45 can be seen that PCA outperformed LDA, both in terms of overall accuracy (PCA: 0.86, LDA: 0.71) that in terms of pollutants confusion with SWW (see last columns of confusion matrices).

It is worth noting that the confusion between pollutants still allows the end-user to be warned about the presence of a dangerous substance, whereas confusing a pollutant with SWW does not allow any warning. PCA confused very few percentages of pollutants with SWW, whereas LDA confused 60% of phosphoric acid with SWW.

As for the inter-pollutant confusion, we can observe that LDA achieved a peroxide-acid confusion which is less acceptable than acid-acid confusion as discussed in the First experiment results subsection.

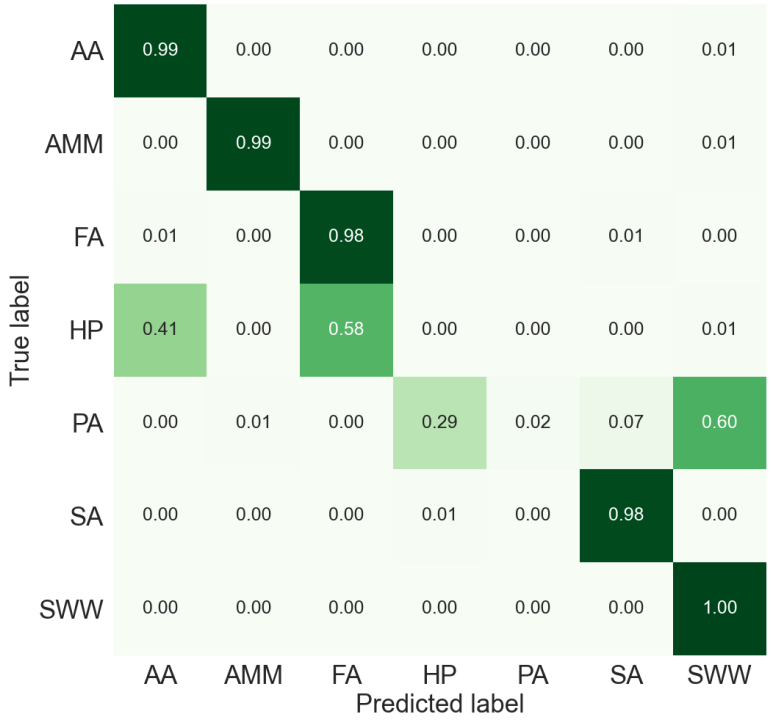


Figure 45 Confusion matrix PB system with LDA.

Smart Initialization

To test the effectiveness of the proposed new initialization strategy, three different values for the initial population fraction to initialize according to the procedure depicted in Algorithm 6 have been tested: 0.05, 0.10, and 0.15.

To evaluate the improvement brought by the “smart initialization” have been analyzed and compared the population's average fitness (i.e., the training accuracy) along the evolution for the three tested values and without the smart initialization procedure (0.0). Figure 46 shows the results of the comparison.

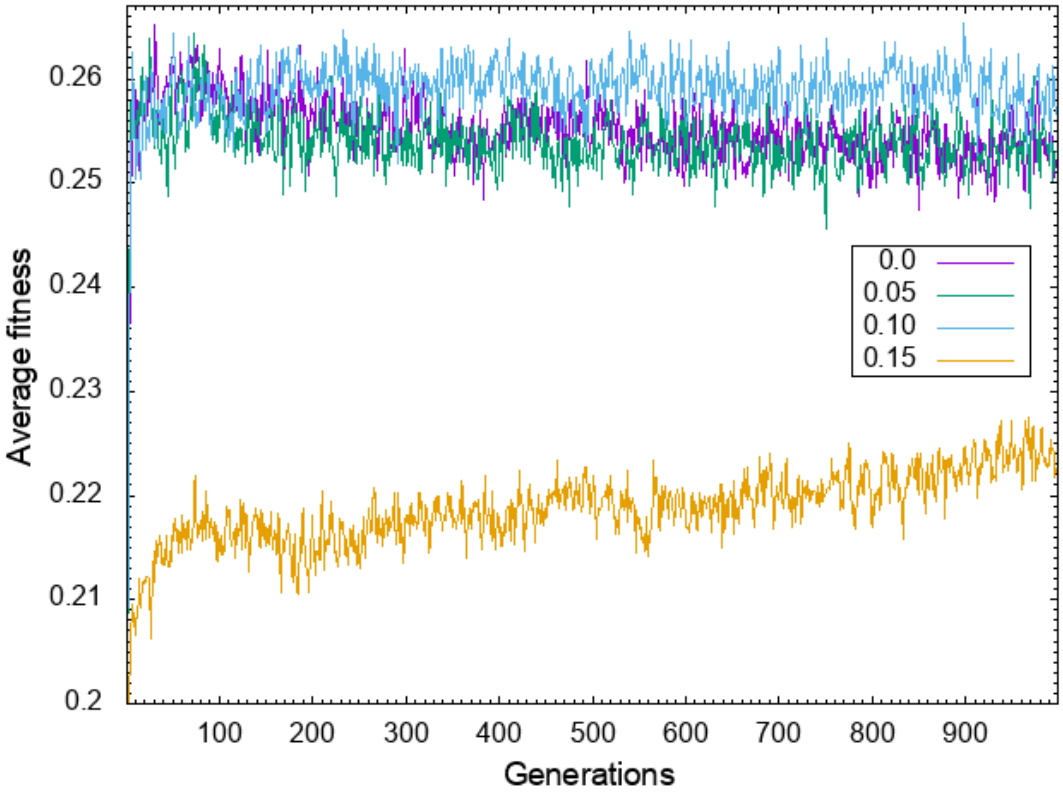
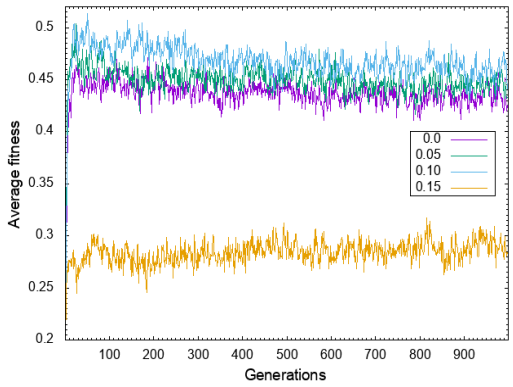


Figure 46 Overall average fitness.

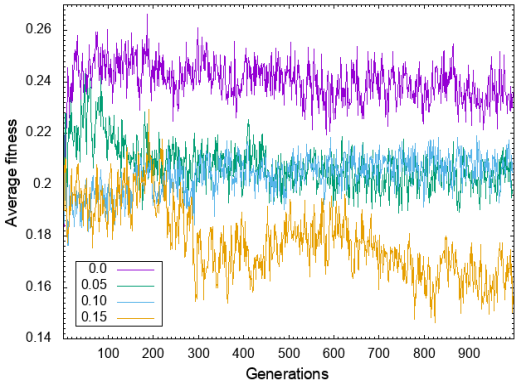
As can be seen, the best performance was achieved by the values 0.0, 0.05, and 0.10, while the value 0.15 performed much worse than the other ones. From the obtained results can be concluded that:

i. A too-high fraction of initialized individuals limits the exploration ability of the evolutionary algorithm.
ii. Low values of initialization do not allow any improvement.
iii. There is a value that allows a small but significant improvement.

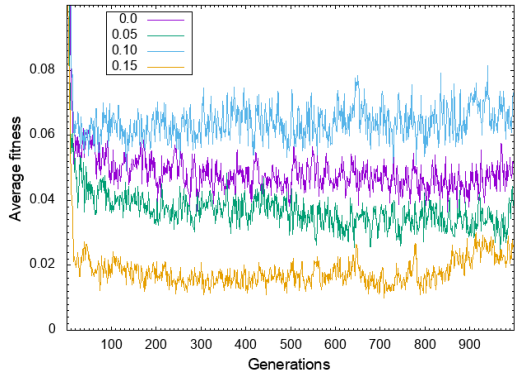
Other than the overall average fitness has been evaluated, even the average fitness along the evolution for the single substance (see Figure 47). From the results, for most of the substances, except for the sulphuric acid, the value 0.15 performed worse than the value of the others. Regarding the value 0.10, instead, the performance is slightly better than the others except for the one obtained with the formic acid; in this case, the 0.0 value (random initialization) was the best performing.



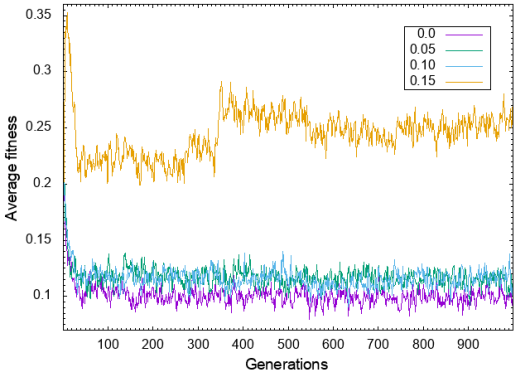
(a) Ammonia



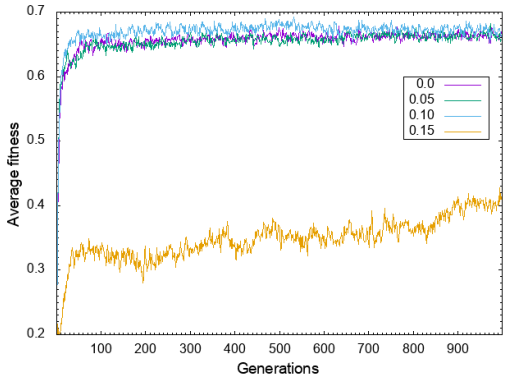
(b) Formic Acid



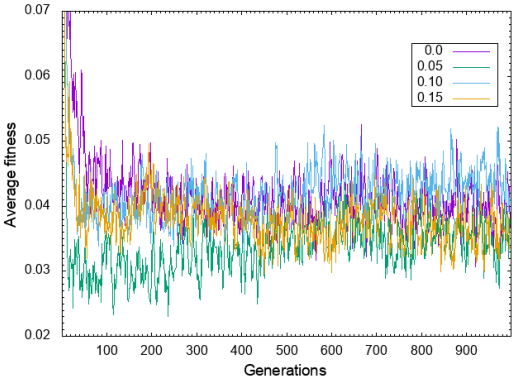
(c) Hydrogen Peroxide



(d) Sulphuric Acid



(e) Acetic Acid



(f) Phosphoric Acid

Figure 47 Single substance average fitness.

Third experiment results

The third experiment was meant to evaluate the effectiveness of the proposed approach by comparing the results obtained with the PB approach with those achieved by using four well-known and widely used classification algorithms:

1. Decision Tree (DT)
2. Nearest Neighbor (KNN)
3. Neural Networks (NN)
4. Support Vector Machines (SVM)

Table 19 reports the values of the classifier parameters used for the comparison. For the sake of a fair comparison, the same training and test procedure has been used for our evolutionary algorithm (performing 30 runs). The Wilcoxon rank-sum test ($\alpha = 0.05$) has been performed to validate the comparison results. The obtained results are reported in Table 20. The results report the average accuracy and the related standard deviation computed over the 30 runs, the *p*–value of the Wilcoxon test, and the performance achieved on the best run.

From Table 20 can be seen that the difference between the results of machine learning (ML) algorithms and those of our system is not statistically significant. However, looking at the best accuracy, the proposed approach outperforms the ML ones.

Figure 48 reports confusion matrices computed over the best run of the ML algorithms. As can be seen, there are similar behaviors to those exhibited by our system, indeed there is confusion between acids e.g., between formic (#3), phosphoric (#5), and acetic (#1) acids.

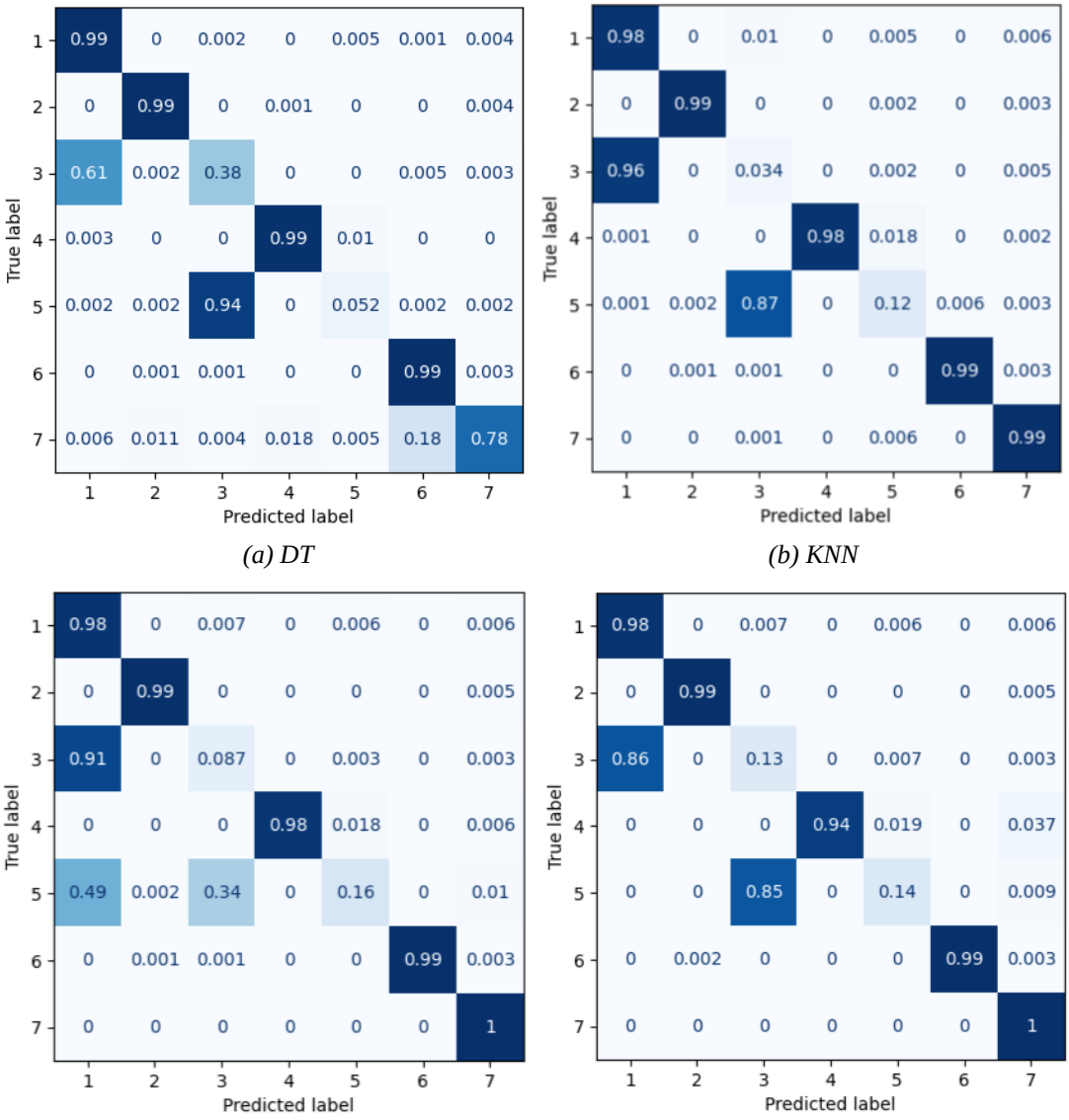
In contrast, regarding the confusion between the other substances with the SWW, only the DT had a significant percentage of confusion (about 22%), which makes this algorithm much less performing than the other ones.

Table 19 Classifier parameters.

Classifier	Parameter	value
DT	Confidence factor	0.25
	Minimum #instances per leaf	2
KNN	K	3
	Distance	Euclidian
NN	Learning rate	0.3
	Momentum	0.2
	Hidden Neurons	8
	Epochs	500
SVM	Kernel	RBF
	C	1.0
	γ	0.5

Table 20 Comparison results.

Classifier	Avg	Std	*p*	Best
Our system	0.69	0.13	—	0.87
SVM	0.73	0.01	0.16	0.74
MLP	0.73	0.01	0.11	0.74
DT	0.69	0.02	0.92	0.74
KNN	0.71	0.01	0.67	0.73



(a) DT *(b) KNN*

(c) NN *(d) SVM*

Figure 48 Confusion matrices ML algorithmes.

3.3 Dynamic Labeling

Even though the PB model has been capable to solves most of the problems related to the LB and CB models, during the Ph.D. thesis, another strategy aimed at further improvement has been evaluated: Dynamic Labeling (DL). It is worth specifying that even the DL model bases the class representation on lines in the 3-D space. The main idea behind the proposed strategy is that the genes (i.e., lines) which make up the individuals are not a priori associated whit any of the available classes (as was the case in the previously presented models), but their labeling occurs at runtime, more in details, after that each sample in the training set has been assigned to its nearest gene. In particular, the gene's labeling association procedure is divided into two main steps:

1. Each sample in the training set is assigned to the nearest gene (i.e., line) of the given individual to be evaluated. Then the Euclidean distance in the feature space (3-D in our case) is computed (as already described in the previous sections). After this step, p_i ($p_i \geq 0$) samples will have been assigned to the nearest *i-th* gene. Finally, the genes with $p_i > 0$ will be referred to as valid, while the remaining genes with $p_i = 0$ will be ignored.
2. Each valid gene is, then, labeled with the class most widely represented among the patterns that have been assigned to it.

For example, we can imagine a case in which four classes need to be recognized. In that case, each individual is made up of four genes. Now if we look at Figure 49 we can distinguish two possible cases: *(a)* where the DL strategy has been used, and *(b)* where the static labeling (SL) strategy was adopted.

As can be noted with the DL approach *(a)* the gene G_1 is labeled as the c_4, G_2 as c_1, G_3 as c_3, and G_4 as c_2, while with the SL the association is a priori defined: $G_1 \rightarrow c_1$, $G_2 \rightarrow c_2$, and so on. It is important to note that the DL, furthermore, allows the EA to automatically find the most proper number of genes for the considered classification problem. Yet, genes DL allows for the relaxation of a strong constraint due to the a priori labeling of the genes (as is the case of Figure 49 (b)).

Indeed, suppose the generic case in which the data contains *c* classes, in the genes SL approach each individual will also contain *c* a priori labeled genes. The constraint imposed over the gene's labels reduces a factor (*c!*) the number of solutions to be considered as a possible solution to the problem.

It is clear that with an SL strategy among the (*c!*) possible permutations of a set of *c* genes, only one is considered a good solution, while the remaining ones are considered bad solutions. This is caused by the a priori gene's labeling.

Instead using a gene's DL strategy each individual is evaluated according to the fitness function value associated with the given *c* class. It's clear that by increasing the number of *c* classes the DL strategy outperforms, even more, the system performance obtained by the SL approach.

	G_1	G_2	G_3	G_4
c_1	30	**55**	0	18
c_2	0	16	28	**80**
c_3	10	0	**72**	0
c_4	**60**	29	0	2

(a) DL strategy

	G_1	G_2	G_3	G_4
c_1	**30**	55	0	18
c_2	0	**16**	28	80
c_3	10	0	**72**	0
c_4	60	29	0	**2**

(b) ST strategy

Figure 49 Comparison between gene's DL and SL strategies.

3.3.1 Obtained Results

In order to prove the effectiveness of the proposed DL approach has been made two sets of experiments. In the first sets, to prove the improvement given by the proposed strategy, a comparison between the DL model using the Cartesian coordinates and the corresponding LB one has been made. The second set of experiments, on the other hand, aims to test the robustness of the new approach by using the polar coordinates and substantially increasing the number of substances used and by comparing the obtained results with those achieved by the well-known and widely used ML algorithms.

First Set of Experiments

In the first set of experiments, a comparison between the DL strategy and LB one using the cartesian coordinates has been made.

Regarding the evolutionary algorithm, dataset structure, preprocessing techniques, and fitness function used within the DL strategy they are conceptually the same as already well discussed in the previous sections.

To evaluate the effectiveness of the new DL approach have been performed different tests by changing the maximum number of genes that each individual can hold between 3, 3.5, 4, 4.5, and 5 times the minimum number of genes, which was equal to the number of classes.

The idea is to exploit the capability of the gene's DL to be able to find the best combination of genes in order to optimize the fitness function value. For each test performed, has been applied the cross-validation technique over 10-fold. Table 21 reports the obtained result in terms of mean accuracy over the 10-fold.
Figure 50, instead, reports the comparison between the best results obtained by the DL approach and the LB one. As can be seen from the shown results the DL strategy obtains the best accuracy of 81.4% outperforming the one obtained by the LB (42%).

Table 21 DL strategy with a different number of genes results.

	Max number of genes Accuracy (%)				
	3	3.5	4	4.5	5
Fold 1	19.9	25.8	34.8	27.5	36.5
Fold 2	49.7	41.8	60.2	36.8	51.1
Fold 3	37.7	41.5	39.9	46.3	61.3
Fold 4	71.7	74.2	72.9	73.7	74.7
Fold 5	43.9	57.5	56.5	48.9	51.3
Fold 6	67.3	75.2	60.7	68.7	64.4
Fold 7	**76.7**	54.1	64.4	**73.9**	**77.9**
Fold 8	66.2	43.9	**77.3**	71.3	76.7
Fold 9	58.9	71.8	72.8	71.1	75.8
Fold 10	69.7	**81.4**	71.5	66.0	46.9
Mean	56.17	56.72	61.10	58.42	**61.66**

(a) DL best results *(b) LB best results*

Figure 50 Comparison results between DL and LB model's best results.

Second Set of Experiments

This set of experiments aims to evaluate the effectiveness of the PB model's technique in conjunction with the usage of the DL strategy. Thus, in those experiments has been used all the previously presented techniques: have the polar coordinates and smart initialization. To prove the effectiveness brought by the DL has been increased the number of substances under test and a comparison with the well-known and wieldy used ML algorithms have been performed.

It is worth specifying that the substances used during all those experiments are listed in the following:

1. Acetic Acid (ACT)
2. Ammonia (AMM)
3. Dish Wash Detergent (DWD)
4. Formic Acid (FMC)
5. Hydrogen Peroxide (HDP)
6. Nelsen (NLS)
7. Phosphoric Acid (PSP)
8. Sodium Chlorite (SCH)

9. Sulphuric Acid (SPH)
10. Washing Machine Detergent (WMD)
11. Synthetic Waste Water (SWW)

Figure 51 shows the best results reached by the DL approach using a max of 3.5 number of genes per class and 15% of smart initialization. Note that regarding the choice of the best value for the number of maximum allowed genes and the best value of the smart initialization fraction, previous tests have been performed. It is important to note that, given the large number of substances used, the obtained results have exceeded all expectations achieving an overall accuracy of the 89,73%.

Moreover, looking the Figure 51 can be seen that there are mainly three major confusions:

- 40% of the Phosphoric Acid samples have been confused with Formic Acid.
- 26% of the Washing Machine Detergent has been confused with Dish Wash Detergent.
- 18% of the Dish Wash Detergent has been confused with Sodium Chlorite.

As already discussed in Section 3.2.5, the confusion showed by the DL system can be considered completely "acceptable", indeed in the first case (PSP - FMC) we have confusion between two acids, while in the second (WMD – DWD) and third (DWD - SCH) cases there is a confusion between two non-acid substances.

Finally, to evaluate the effectiveness of the proposed approach, the obtained results have been compared with those achieved by using seven well-known classification algorithms:

- Supported Vector Machine (SVM)
- Multilayer Perceptron (MLP)
- Decision Tree (DT)
- K-Nearest Neighbor (KNN)
- Bagging
- Random Forest
- Adaboost

The comparison is reported in Table 22. The results show the average accuracy, the standard deviation computed over the 30 runs, the *p*–value of the Wilcoxon test (α = 0.05) and the performance achieved on the best run.

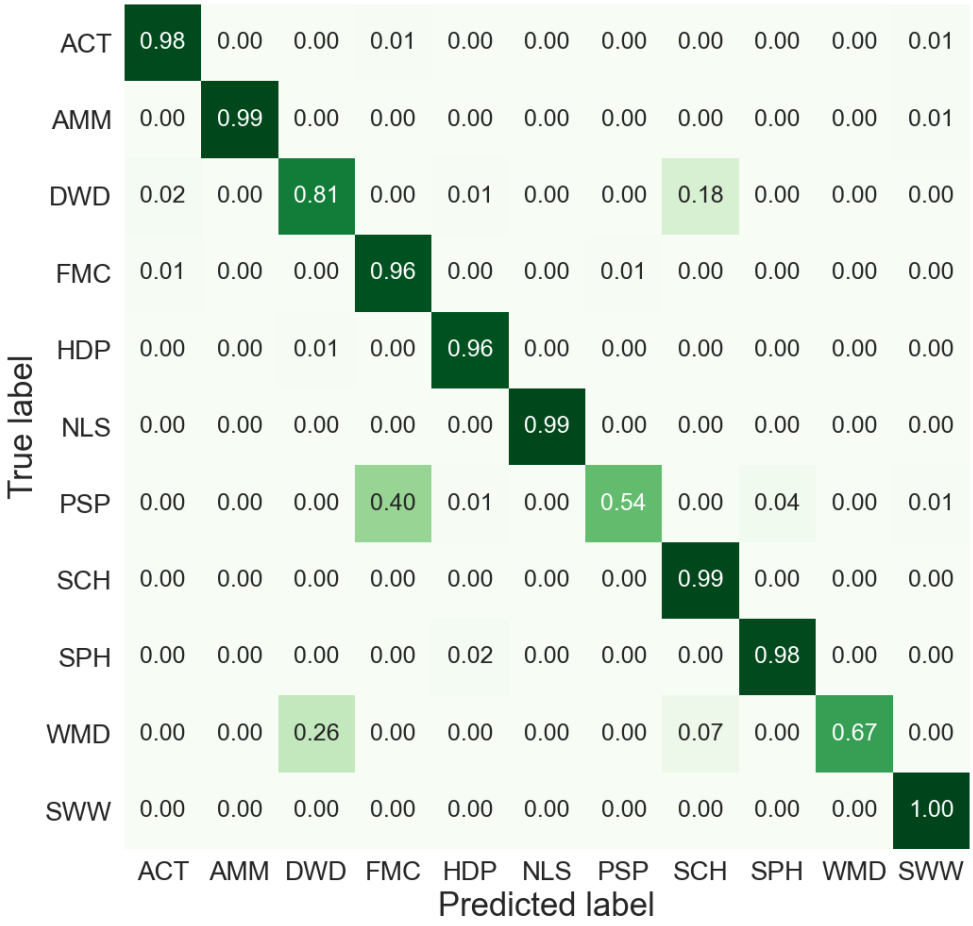


Figure 51 DL best results.

Table 22 Dynamic Labeling comparison results.

Classifier	Avg	Std	*p*	Best
Our system	0.72	0.05	—	0.83
SVM	0.67	0.03	1.04e-04	0.74
MLP	0.70	0.04	0.0584	0.78
DT	0.55	0.05	5.5e-11	0.63
KNN	0.57	0.03	5.5e-11	0.63
BAGGING	0.57	0.04	1.33e-10	0.65
RANDOM FOREST	0.67	0.04	2.68e-04	0.73
ADABOOST	0.31	0.11	3.02e-11	0.51

The results show that the proposed systems tend to have an overall performance better than the machine learning ones. Even looking at the best case, our system outperforms the other ones.

3.4 Conclusion

Water pollution is a worldwide concern and therefore, it is crucial to find reliable and low-cost technologies for continuous and diffused monitoring of wastewater.
In this Ph.D. work, I've presented a system for water pollutant classification to be implemented on the multi-sensor microcontroller SENSIPLUS, where input data were first projected into a 3-D space and then classified using simple geometrical models. The aim was to implement a pollutant classification system able to work even with the few computational resources available on cheap microcontrollers. In this study, I've presented a further development of those approaches. This development allowed us to (i) improve the effectiveness of the IoT-based system; (ii) reduce the number of computational resources needed.
The obtained results proved the effectiveness of the proposed solutions to improve the performance of our system. The results also confirmed that our system can be compared, and sometimes outperforms, some state-of-the-art classification algorithms. Furthermore, with the implementation of the dynamic labeling strategy presented in [57], has been possible to reach further improvement both in terms of the number of output classes and in terms of overall performance.
Even though the obtained results show that the proposed system is capable of correctly distinguishing between a set of substances, an interesting future work would be to implement the system on an MCU in order to perform some tests on the real scenario to evaluate the system capability to work in a real context.

CHAPTER 4. TEST ON REAL FIELD

4. Introduction

It is important to say that part of the activity conducted during my Ph.D. thesis work was part of the European project named SYSTEM. SYSTEM is the three-year innovation action awarded to a consortium led by Fondazione FORMIT addressing the challenge of the topic "Integration of detection capabilities and data fusion with utility providers' network" (SEC-10-FCT-2017) included in the 2016-2017 Work Programme "Secure societies Protecting freedom and security of Europe and its citizens" of Horizon 2020. SYSTEM started on 1 September 2018 and aims at developing and testing a customized sensing system for hazardous substances detection in complementary utility networks and public spaces. The proposed innovative monitoring and observing of fused data sources will be tested and eventually adapted in six urban areas, while carefully tracking Ethics and Legal aspects as well as managing confidential information. To achieve these aims, a wide set of skills and capabilities has been considered key to success, determining the large partnership working on the project, made by partners cooperating with more than ten stakeholders supporting the project activities.

For that reason, all the research activity depicted in the previous sections has been conducted with different tests on the field in a real savage network scenario. The main goal of the real tests was to validate the developed system in terms both of accuracy and robustness. Mainly, have been made two main sets of experiments on a real scenario: one at the wastewater treatment plant of Acqualatina in Borgo Piave (Latina, Italy) (see Figure 52), and the second set on a series of manholes situated in Via Castelbottaccio (East Rome, Italy) in collaboration with ACEA (see Figure 53).

Other minor real scenario tests have been made during my Ph.D. and they will be discussed in Section 4.3. It is worth noting that during all the tests have been used different KNN models, trained on different sets of substances and in different conditions, but the one that best performs, in terms of accuracy and robustness, over different scenarios is reported in Figure 54. Regards the classification system is the

one depicted in Section 2.5 except for the anomaly detection system that at the time of testing had not yet been implemented. It is important to underline that both the classification system and the preprocessing module have been constantly developed, improved, and implemented alongside the evidence gathered during the real tests.

Figure 52 Wastewater treatment plant, BorgoPiave Latina.

Figure 53 Via Castelbottaccio East Rome.

Global confusion matrix for K-NN *Global accuracy:* *90.23%*

(mean and std on 10 fold experiments)

AMMONIA		ACETONE		SODIUM HYPOCHLORITE		SULPHURIC ACID		SWW		DW DETERGENT		TATP		FORMIC ACID		WM DETERGENT		ACETIC ACID		ETHANOL		PHOSPHORIC ACID		HYDROGEN PEROXIDE		SODIUM CHLORIDE				
1		*2*		*3*		*4*		*5*		*6*		*7*		*8*		*9*		*10*		*11*		*12*		*13*		*14*		<-	Classified as	
mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std			
99,0	**0,5**	0,0	0,0	0,0	0,0	0,0	0,0	0,7	0,3	0,0	0,0	0,3	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	99,0	*1*	**AMMONIA**
0,0	0,0	**74,3**	**28,2**	0,0	0,0	0,0	0,0	0,8	0,2	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	24,8	28,1	0,0	0,0	0,0	0,1	0,0	0,0	74,3	*2*	**ACETONE**
0,0	0,0	0,0	0,0	**94,6**	**14,6**	0,0	0,0	0,4	0,0	0,0	0,0	5,0	14,6	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	94,6	*3*	**SODIUM HYPOCHLORITE**
0,2	0,2	0,0	0,0	0,0	0,0	**89,3**	**29,8**	0,4	0,2	0,0	0,0	0,2	0,2	9,9	29,5	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	89,3	*4*	**SULPHURIC ACID**
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	**99,6**	**0,4**	0,0	0,0	0,4	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,1	0,1	0,0	0,0	0,0	0,0	99,6	*5*	**SWW**
0,4	0,7	0,0	0,0	0,1	0,1	0,1	0,4	8,8	25,7	**57,2**	**34,1**	0,5	1,5	0,4	1,0	32,3	30,9	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	1,0	57,2	*6*	**DW DETERGENT**
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	3,0	0,7	0,0	0,0	**95,2**	**5,6**	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	1,8	5,3	0,0	0,0	0,0	0,0	95,2	*7*	**TATP**
0,4	0,7	0,0	0,0	0,0	0,1	9,9	29,7	6,2	17,3	0,0	0,0	0,3	0,3	**75,2**	**36,9**	0,0	0,0	0,1	0,3	0,0	0,0	7,8	17,0	0,0	0,1	0,0	0,0	75,2	*8*	**FORMIC ACID**
0,1	0,1	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,1	10,2	18,2	0,1	0,1	0,0	0,0	**89,0**	**18,2**	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,5	0,8	89,0	*9*	**WM DETERGENT**
0,2	0,7	0,0	0,0	0,1	0,1	0,0	0,0	10,0	27,9	0,0	0,0	0,1	0,1	0,4	0,9	0,0	0,0	**87,6**	**28,6**	0,0	0,0	1,5	3,1	0,1	0,1	0,0	0,0	87,6	*10*	**ACETIC ACID**
0,0	0,0	28,4	38,7	0,0	0,0	0,0	0,0	10,0	28,5	0,0	0,0	0,0	0,1	0,0	0,0	0,0	0,0	0,0	0,0	**60,7**	**42,5**	0,1	0,2	0,7	1,9	0,0	0,0	60,7	*11*	**ETHANOL**
0,9	2,7	0,0	0,0	0,0	0,0	0,0	0,0	1,1	0,5	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	7,7	23,1	0,0	0,0	**90,2**	**22,7**	0,0	0,0	0,0	0,0	90,2	*12*	**PHOSPHORIC ACID**
0,0	0,1	1,6	2,4	3,0	9,0	0,0	0,0	15,2	24,9	0,0	0,0	4,1	7,7	0,0	0,0	0,0	0,0	0,0	0,0	2,8	4,5	0,1	0,1	**73,3**	**32,4**	0,0	0,0	73,3	*13*	**HYDROGEN PEROXIDE**
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,1	0,1	0,1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	**99,6**	**0,1**	99,6	*14*	**SODIUM CHLORIDE**

Numbers:

```
* 10      experiments for each substance
* 1600    measure for each experiment (1.5 seconds per measure)
* 600     measure with SWW (background) and 1000 with substance
```

The confusion matrix is obtained on:

```
* 10000   measures for each substance
* 94000   measures for SWW
* 224000  measures for the entire Test Set (1000 * 10 * 14 + 600 * 10 * 14)
```

Figure 54 Best KNN model used during real scenario test

4.1 Borgo Piave (Latina) Tests

The first set of experiments has been conducted in Borgo Piave at the wastewater treatment plant of Acqualatina. Figure 56 shows the experimental environment. In particular, the given substance under test was spilled from the spiking manhole (at the bottom right corner). Then, after 60 m pipe long were installed the SCW system (sensing well). For the acquisition procedure concerns, firstly the SCW where positioned inside the sensing well. Secondly, the measurement system starts the acquisition and waits until the Finite State Machine reaches the Baseline Tracking state. At this point, the given substance was spilled from the spiking manhole. It's important to note that the spilled quantities were around 2 to 5 liters and have been chosen according to a qualitative campaign.

Figure 55 Borgo Piave System tests.

During all the performed tests been faced many problems of different natures. The first problem to be faced was the choice of the best spot in which to install the SCW sensor in order to be able to detect and recognize the spilled substances. After different tries shown in Figure 57 with red circles, the best position capable to maximize the classification accuracy has been found at the exit of the pipe, represented by the green circle. One of the problems related to the red circles' positions, was the heavy presence of the air bubbles. Indeed, by reproducing a similar scenario in the laboratory has been

possible to measure the effect that air bubbles have on the measurements (see Figure 58). Another problem that has been faced concerns interference due to a lifting pump system located upstream of the spiking manhole shown in Figure 56. In particular, has been noted that leaving the SCW flooded in the wastewater without spilling external substances, the response of the sensors was affected by the activity of the pumps. Figure 59 shows the interference measured during the activity of the hydraulic system. Unfortunately, in this case, a solution has not yet been found and so in order to be able to test the classification system has been necessary to turn off the lifting pump system during the measurements.

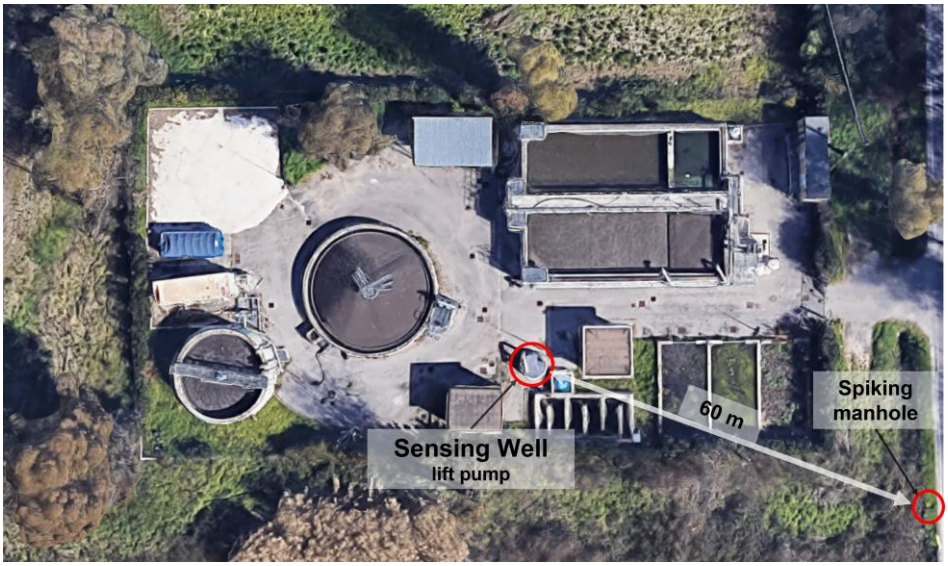


Figure 56 Experimental environment

Figure 57 SCW positions.

Figure 58 Air bubbles from water turbulence effects.

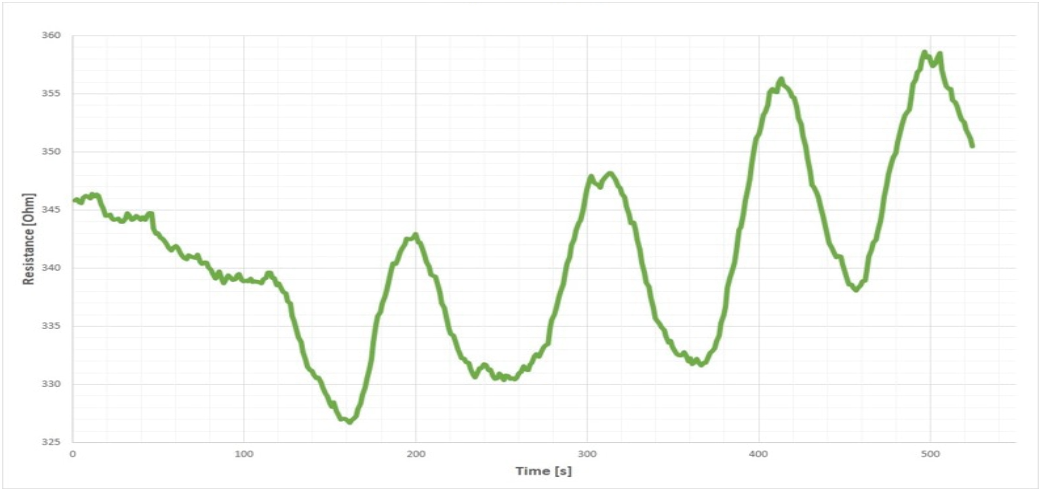


Figure 59 Lifting pump system interferences.

Another open problem related to the Borgo Piave setups was the large presence of solid garbage (see Figure 60) in the wastewater that can alter the sensor's response.
All other kinds of problems like those represented by the heavy environment noise, external activity interferences like industrial or domestic activity, and all the problems related to a different flow rate of water caused, for example, by heavy rain have been resolved by implementing the data pre-processing techniques depicted in the Section 2.5.1.
In this context, many substances have been tested: Phosphoric Acid, Sodium Hypochlorite, Acetic Acid, Formic Acid, Ammonia, and Hydrogen Peroxide (see Figure 61).

Figure 60 Solid garbage.

Figure 61 BorgoPiave tested substances.

4.2 East Rome Tests

Figure 62 East Rome System test.

The second set of experiments has been conducted in Via Castelbottaccio at East Rome. In this case, the scenario was less complex, and much closer to a real use case, than the one related to the wastewater treatment plant in Borgo Piave due to the absence of any lifting pumps in the vicinity, due to the water flow rate that was much lower than the one related to the Acqualatina plant and, for that reason, even the presence of solid wastes was reduced.

Unlike the tests performed at Borgo Piave in Latina, in this case, there was a sensing manhole and three different spiking manholes positioned at three different distances: 50m, 75m, and 150m (see Figure 63). As for the Borgo Piave experiments, the SCW was installed inside the sensing manhole (green circle) while the substance under test was spilled from one of the spiking manholes (red circles). In this context, the spilled quantities were around 1 to 3 liters chosen according to some preliminary tests.

During the tests has been spilled different substances, for example, sodium hypochlorite, acetone, formic acid, sulphuric acid, etc. at different distances in order to evaluate the effect of the dilution over the capacity of the system to detect and recognize the given substance.

In order to be able to install the SCW inside the sensing manhole, has been developed a measurement system prototype shown in Figure 64. As can be seen, the measurement

system is composed of a white box with an IP56 waterproof certificate, a Raspberry Pi4, a GSM hat for Raspberry Pi based on SIM7600E-H, a 20000mAh power bank, an ESP32 board connected to the SCW via a 10m SENSIBUS cable and finally two antennas.

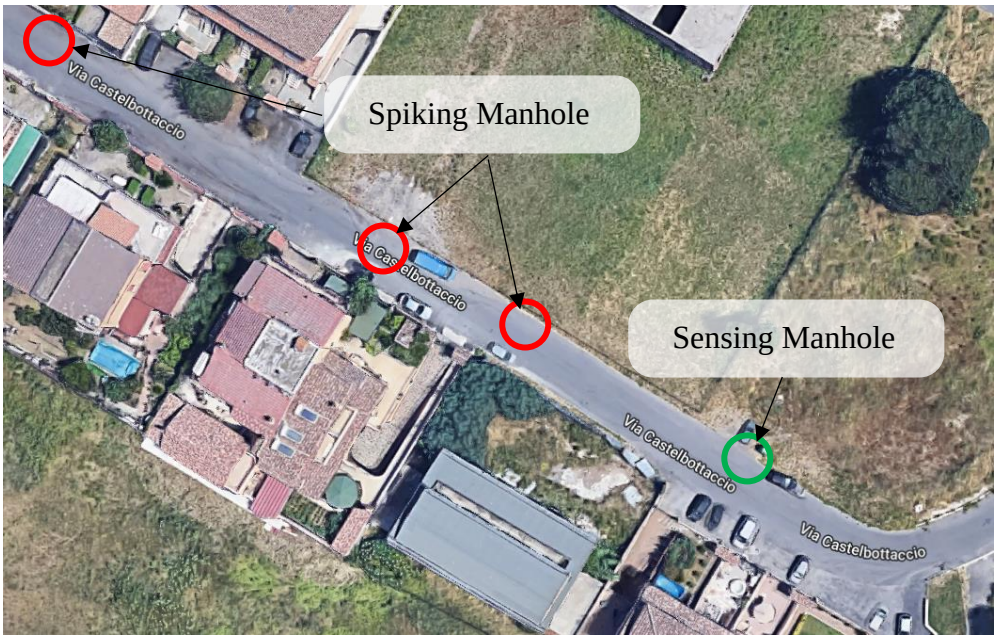


Figure 63 East Rome test: green circle represents the sensing manhole, while red circles represent the spiking manholes respectively positioned at 50m, 75m, and 150m from the sensing manhole.

In particular, on the Raspberry Pi4 was installed the Winux application depicted in Section 1.4.1, the GSM hat was meant to transfer all the acquired data, the ESP32 act as a communication bridge between the Raspberry Pi and the SCW board, the two antennas were needed to improve the signal quality and lastly, the power bank was necessary to power the entire system (Raspberry Pi, MCU, SCW, and the GSM hat).

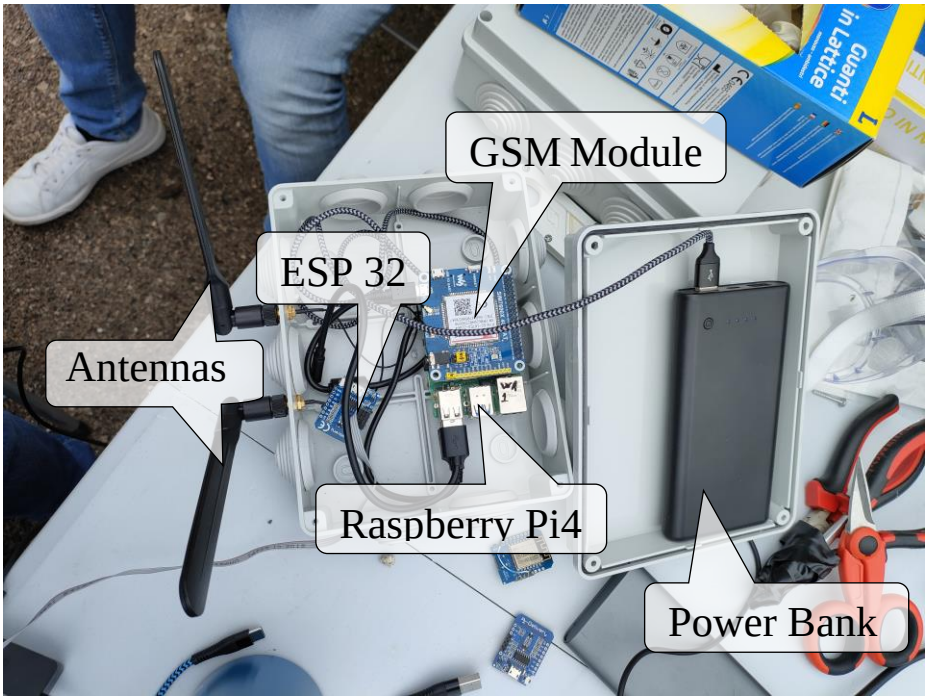


Figure 64 Measurement system prototype, developed for East Rome tests.

In this context, we faced two main problems, one related to the positioning inside the sensing manhole of the SCW board, and one related to some interference caused by household activities.

Regards the positioning of the SCW inside the manhole, the problem was to find a reliable spot capable to keep all the SCW's IDEs flooded into the wastewater; this was necessary since the flow rate of the wastewater inside the sewers network when there wasn't any kind of activities, was really poor. To solve this problem, we decided to anchor the SCW to the bottom of the sewer pipe by using a little metallic bar. In this way has been possible to make tests without care about the state of the wastewater flow rate.

Another problem was related to the interferences caused by human activity. Indeed, looking at the red circles depicted in Figure 65 can be seen the effect of those activities on the sensor's response. Fortunately, at the time of the tests, this problem was already addressed by implementing the data pre-processing module based on a finite state machine (see Section 2.5.1 for more information).

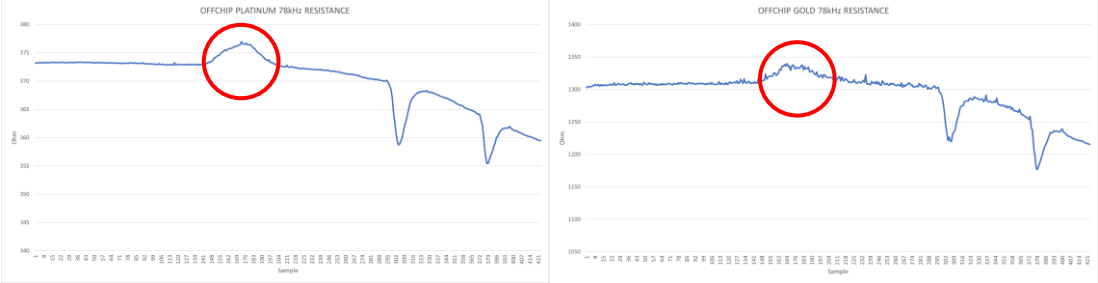
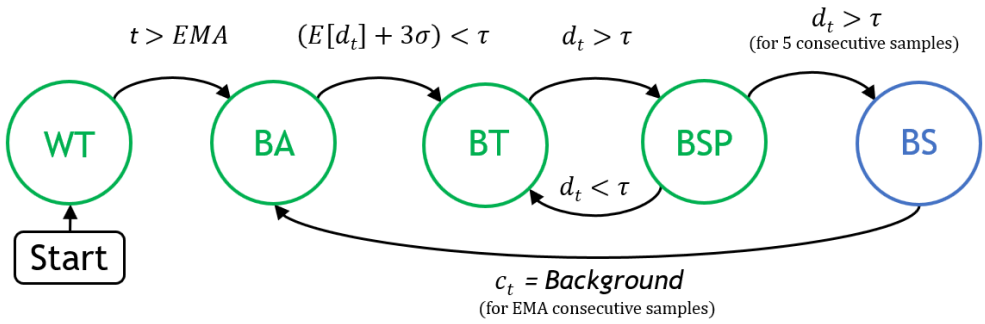


Figure 65 East Rome human activities interferences. On the Right is shown the resistance over the gold IDE at 78kHz, and on the left resistance of the platinum IDE at 78 kHz.

Figure 65, shows the effect of the same interference over two different features: gold IDE resistance at 78kHz and platinum IDE resistance at 78kHz; the others two sensors' responses shown are related to two successive sodium hypochlorite spills.

The only action that has been necessary to correctly reject all those kinds of interferences, was to correctly tune the finite state machine parameters (EMA_c and τ) see Figure 66.



Symbols	
t	time sequence
d_t	distance of s_t from b_t
EMA	Exponential Moving Average coefficient = 25
τ	the threshold (0.05)
c_t	classification of sample t
$E[d_t]$	the mean over last EMA
σ	the std dev over last EMA sample

States		Output
WT	WaiT	BKG
BA	Baseline Acquisition	BKG
BT	Baseline Tracking	BKG
BSP	Baseline SusPended	BKG
BS	Baseline Stopped	The sample is forwarded to the classification module

Figure 66 Finite State Machine parameters.

4.3 Further Tests

In addition to the tests depicted in the previous section, other tests have been made. One was performed at the chemical laboratory of the RaCIS located in Caserma Salvo D'Acquisto Comando Polifunzionale Arma dei Carabinieri Roma shown in Figure 67. In this test has been made some demonstrative experiments using the following substances: Ammonia, Hydrogen Peroxide, Sulphuric Acid, Phosphoric Acid, Acetic Acid, Sodium Hypochlorite, and Nelsen.

Figure 67 Test in the chemical laboratory of RaCIS in Rome.

Finally, other tests on a real field, where I wasn't personally present, have been made. One test has been made in Anzio at the Monumento Caduti Due Guerre Mondiali, where there were two manholes ten meters apart, one where the substances have been spilled (spiking manhole) and one where the SCW was installed (sensing manhole) shown in Figure 68.

Figure 68 Anzio Monumento Caduti Due Guerre Mondiali real test location. The red circle indicates the spiking manhole, while the green is the sensing one. The manholes were located 10 meters apart.

Figure 69 Anzio real test.

Figure 69 shows how the SCW was installed inside the manhole. In this test has been used the Ammonia, Sulphuric Acid, and Hydrogen Peroxide.

Another test has been made at the wastewater treatment plant located in Beuerbach (Frankfurt). This was one of the first tests on the real field and for that reason, only the Ammonia and Sulphuric Acid substances have been used.

In these experiments, two SCWs have been installed into two successive spots see Figure 70 for more details.

Figure 70 Beurbach wastewater treatment plant tests. Red Circles indicate the two spots where SCW was installed.

4.4 Conclusion

In conclusion, from all the evidence and all the experience obtained from the depicted real tests has been possible to build a reliable and robust system capable of correctly detecting and recognizing a given substance in a real application. It is essential to note the importance of having had the opportunity to perform various tests in the real field and in a different context, which allowed us to face many real problems that, otherwise, would hardly have emerged during laboratory tests.

All this gave us the possibility of being able to develop an end-to-end system that, starting from the acquisition of the single data, was able to first process and then classify the measured sample. Even though some of the faced problems have been resolved, there are still many open problems that need to be addressed before being able to install the proposed system in a continuous wastewater monitoring station. Between the major problems that need to be resolved there is, for sure, the necessity to develop a module capable to manage two or more successive substance spills without giving the system the necessary time to return to a state capable to wait for the next spill (Baseline Tracking see Figure 66) and to the SCW's sensors the time to clean itself by the particles of the last spilled substance.

Another aspect that has to be taken into account and has to be faced is to evaluate the system behavior during continuous monitoring sessions, in this sense, an endurance test has never been performed. For a continuous monitoring system, it is important to collect as much as possible information regarding the possible problems that can be faced in this kind of context.

After all the considerations that have been made, based on all the information gathered during the real tests and, in particular from the last tests performed at the wastewater treatment plant in Brogo Piave (Latina) and the East Rome ACEA's manholes, we can reasonably state that the proposed system can represent a viable end-to-end solution for continuous wastewater monitoring.

APPENDIX A

Winux XML measurements configuration file

As already described in Section 1.4.1 from the Batch tab view it is possible to load a given XML file containing all the measurement information to allow the Winux application to execute the measurement session autonomously. Figure 71 shows the XML file structure and the related measurement flow result.

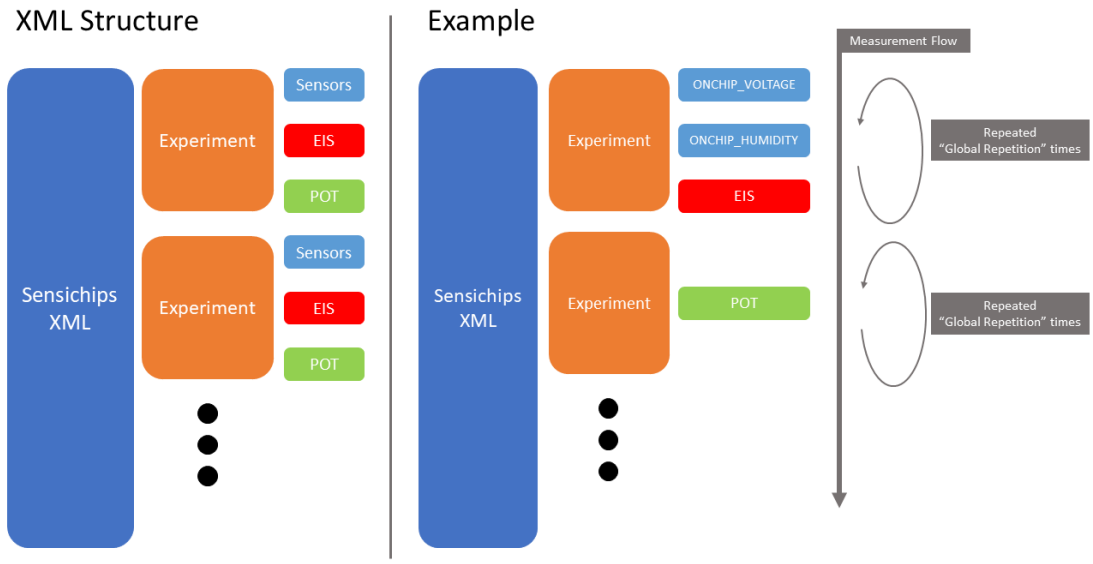


Figure 71 XML file structure with an example showing the measurement flow.

As can be seen, each XML can be composed of different experiments, and each experiment can be made by different kinds of measurements. Moreover, all the measurements of a given experiment are performed "Global Repetition" (see Figure 72) times. Figure 72 and Figure 73 shows in detail the allowed XML attributes related to the experiment, sensor, eis, and pot tags.

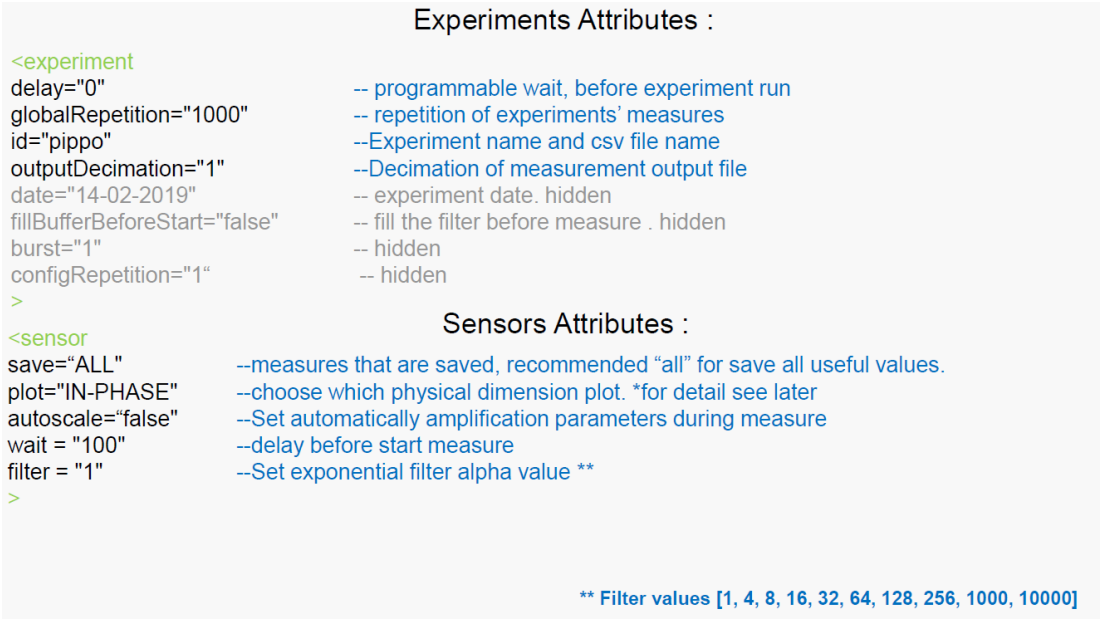


Figure 72 XML attributes related to the experiment and sensor tags.

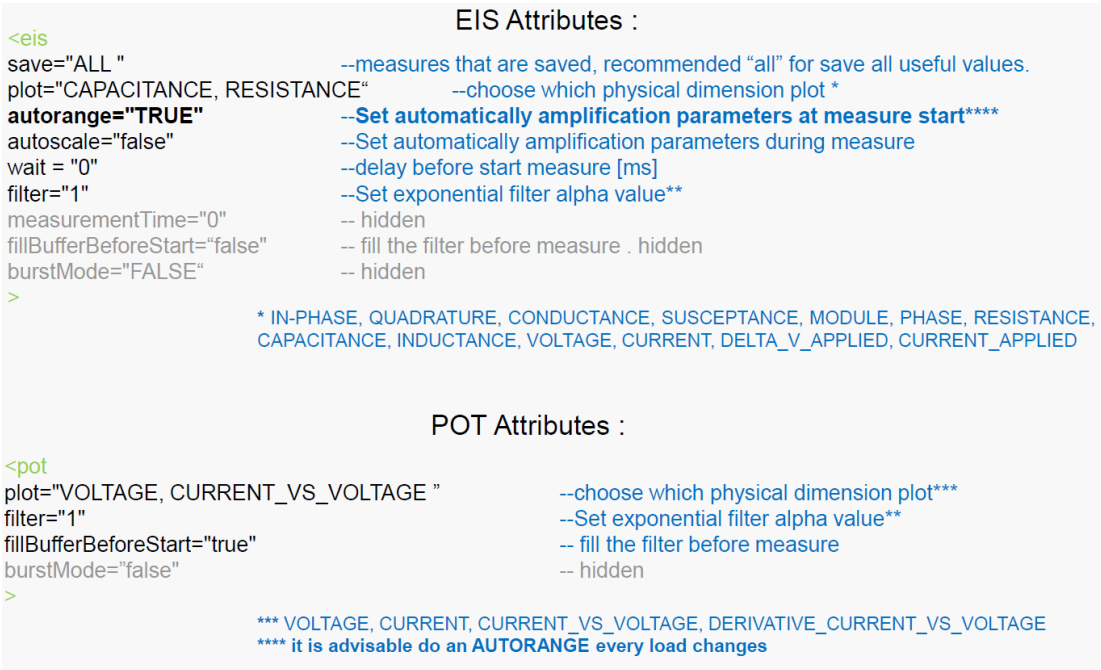


Figure 73 XML attributes related to the eis and pot tags.

A commented XML file of an EIS measurement is reported in the following.

```
<sensichips>
    <experiment date="14-02-2020" globalRepetition="5000" configRepetition="1" burst="1"
     fillBufferBeforeStart="FALSE" outputDecimation="1" id="FileName" delay="0" >
            <!-- Autorange sets Amplification parameters automatically, filter is a moving
                 average windows size-->
            <eis save="ALL" plot="RESISTANCE" autorange="TRUE" autoscale="false"
                 measurementTime="0" waitSET="0" SETwaitGET="0" filter="1"
                 fillBufferBeforeStart="true" burstMode="FALSE">
                    <!-- Measurement frequency [Hz]-->
                    <frequencies order="1">
                            <val>78125.00</val>
                    </frequencies>
                    <!-- Amplification parameter, allowed value [50, 500, 5000, 50000] -->
                    <rsense order="2">
                            <val>50</val>
                    </rsense>
                     <!-- Amplification parameter, allowed value [1, 12, 20, 40] -->
                     <ingain order="3">
                            <val>1</val>
                    </ingain>
                    <!-- Peak-to-peak amplitude of the output Sinewave, allowed value
                         [0 to 7] -->
                     <outgain order="4">
                            <val>7</val>
                    </outgain>
                    <!-- add a programmable DC offset to DASF terminal, allowing value to
                        [-2048 to 2047] -->
                     <dcbiasP order="5">
                            <val>0</val>
                    </dcbiasP>
                    <!-- add a programmable DC offset to VSCMF terminal, allowed
                        value [-32 to 31] -->
                     <dcbiasN order="6">
                             <val>0</val>
                     </dcbiasN>
                     <!-- number of contacts available [TWO, FOUR] -->
                     <contacts order="7">
                             <val>TWO</val>
                     </contacts>
```

```
                    <!-- Measurement Mode allowed [Vout_Iin, Vout_Vin, Iout_Iin,
                        Iout_Vin] -->
                    <modevi order="8">
                            <val>VOUT_IIN</val>
                    </modevi>
                    <!-- Selects the demodulation frequency allowed
                        [FIRST_HARMONIC, SECOND_HARMONIC, THIRD_HARMONIC] -->
                    <harmonic order="9">
                            <val>FIRST_HARMONIC</val>
                    </harmonic>
                    <!-- Measurement PORT allowed [PORT0 to PORT11, PORT_HP,
                        PORT_EXT1 to PORT_EXT3, PORT_EXT1_1 to PORT_EXT3_1,
                        PORT_TEMPERATURE, PORT_VOLTAGE, PORT_LIGHT, PORT_DARK,
                        PORT_NA, PORT_SHORT, PORT_OPEN] -->
                    <inport order="10">
                            <val>PORT_HP</val>
                    </inport>
                    <!-- Measurement PORT allowed [PORT0 to PORT11, PORT_HP,
                        PORT_EXT1 to PORT_EXT3, PORT_EXT1_1 to PORT_EXT3_1,
                        PORT_TEMPERATURE, PORT_VOLTAGE, PORT_LIGHT, PORT_DARK,
                        PORT_NA, PORT_SHORT, PORT_OPEN] -->
                    <outport order="11">
                            <val>PORT_HP</val>
                    </outport>
                    <SequentialMode order="12">
                            <val>0</val>
                    </SequentialMode>
                    <!-- add a delay to the Demodulation Channel -->
                    <phaseShift order="13">
                            <val>Quadrants, 0, IN_PHASE</val>
                    </phaseShift>
            </eis>
        </experiment>
</sensichips>
```

A commented XML file of a Sensor measurement is reported in the following.

```
<sensichips>
        <experiment globalRepetition="100000" outputDecimation="1" id="TestTemp" delay="0" >
                <sensor save="" plot="" autoscale="true" wait = "100" filter = "1">
                        <val>ONCHIP_TEMPERATURE</val>
                </sensor>
        </experiment>
</sensichips>
```

A commented XML file of a POT measurement is reported in the following.

```
<sensichips>
        <experiment globalRepetition="500" outputDecimation="1" id="TEST" delay="0" >
                <!-- POSSIBLE PLOT: VOLTAGE - CURRENT - CURRENT_VS_VOLTAGE -->
                <pot plot="VOLTAGE, CURRENT, CURRENT_VS_VOLTAGE" filter="1"
                        burstMode="true" >
                        <!-- Possible values: LINEAR_SWEEP - STAIRCASE - SQUAREWAVE -
                        NORMAL_PULSE - DIFFERENTIAL_PULSE - POTENTIOMETRIC - CURRENT -->
                         <type>STAIRCASE</type>
                        <rsense>500</rsense>
                        <ingain>1</ingain>
                         <port>PORT_HP</port>
                        <contacts>TWO</contacts>
                        <!-- range: [-1250,1250] -->
                        <initial_potential>-1250</initial_potential>
                        <!-- range: [-1250,1250] -->
                        <final_potential>1250</final_potential>
                        <step>30</step>
                        <pulse_amplitude>75</pulse_amplitude>
                        <pulse_period>100</pulse_period>
                        <alternative_signal>true</alternative_signal>
                </pot>
        </experiment>
</sensichips>
```

REFERENCES

[1] D. T. Jamison, J. G. Breman, A. R. Measham, G. Alleyne, M. Claeson, D. B. Evans, P. Jha, A. Mills and P. Musgrove, Disease Control Priorities in Developing Countries, The International Bank for Reconstruction and Development / The World Bank, 2006.

[2] P. K. Goel, Water Pollution: Causes, Effects and Control, New Age International, 2006.

[3] A. J. Whelton, L. McMillan, M. Connell, K. M. Kelley, J. P. Gill, K. D. White, R. Gupta, R. Dey and C. Novy, "Residential Tap Water Contamination Following the Freedom Industries Chemical Spill: Perceptions, Water Quality, and Health Impacts," *Environmental Science & Technology*, vol. 49, p. 813–823, January 2015.

[4] S. K., T. V. S., M. S. Kumaraswamy and V. Nair, "IoT based Water Parameter Monitoring System," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, 2020.

[5] Z. Sun, N. B. Chang, C. F. Chen, C. Mostafiz and W. Gao, "Ensemble Learning via Higher Order Singular Value Decomposition for Integrating Data and Classifier Fusion in Water Quality Monitoring," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 3345-3360, 2021.

[6] J. Cleary, D. Maher, C. Slater and D. Diamond, "In situ monitoring of environmental water quality using an autonomous microfluidic sensor," in *2010 IEEE Sensors Applications Symposium (SAS)*, 2010.

[7] A. C. D. S. Júnior, R. Munoz, M. D. L. A. Quezada, A. V. L. Neto, M. M. Hassan and V. H. C. D. Albuquerque, "Internet of Water Things: A Remote Raw Water Monitoring and Control System," *IEEE Access*, vol. 9, pp. 35790-35800, 2021.

[8] L. Atzori, A. Iera and G. Morabito, “The Internet of Things: A survey,” *Computer Networks,* vol. 54, pp. 2787-2805, 2010.

[9] R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar and B. Qureshi, “An Overview of IoT Sensor Data Processing, Fusion, and Analysis Techniques,” *Sensors,* vol. 20, 2020.

[10] D. S. Rathee, K. Ahuja and A. Nayyar, “Sustainable future IoT services with touch-enabled handheld devices,” *Security and Privacy of Electronic Healthcare Records: Concepts, paradigms and solutions,* 2019.

[11] A. Nayyar and V. Puri, “Smart farming: IoT based smart sensors agriculture stick for live temperature and moisture monitoring using Arduino, cloud computing & solar technology,” 2016.

[12] W. Shi and S. Dustdar, “The Promise of Edge Computing,” *Computer,* vol. 49, pp. 78-81, May 2016.

[13] A. Kaur, P. Singh and A. Nayyar, “Fog Computing: Building a Road to IoT with Fog Analytics,” in *Fog Data Analytics for IoT Applications: Next Generation Process Model with State of the Art Technologies*, S. Tanwar, Ed., Singapore, Springer Singapore, 2020, p. 59–78.

[14] A. Bernieri, L. Ferrigno, M. Laracca and M. Molinara, “An SVM Approach to Crack Shape Reconstruction in Eddy Current Testing,” in *2006 IEEE Instrumentation and Measurement Technology Conference Proceedings*, 2006.

[15] G. Cerro, M. Ferdinandi, L. Ferrigno, M. Laracca and M. Molinara, “Metrological Characterization of a Novel Microsensor Platform for Activated Carbon Filters Monitoring,” *IEEE Transactions on Instrumentation and Measurement,* vol. 67, pp. 2504-2515, 2018.

[16] G. Cerro, M. Ferdinandi, L. Ferrigno and M. Molinara, “Preliminary Realization of a monitoring system of Activated Carbon Filter RLI based on the SENSIPLUS® microsensor platform,” 2017.

[17] J. Li and S. Cao, “A Low-cost Wireless Water Quality Auto-monitoring System,” *International Journal of Online and Biomedical Engineering (iJOE),* vol. 11, p. pp. 37–41, May 2015.

[18] W. Schmidt, D. Raymond, D. Parish, I. G. C. Ashton, P. I. Miller, C. J. A. Campos and J. D. Shutler, “Design and operation of a low-cost and compact

autonomous buoy system for use in coastal aquaculture and water quality monitoring," *Aquacultural Engineering,* vol. 80, p. 28–36, 2018.

[19] S. Zhuiykov, "Solid-state sensors monitoring parameters of water quality for the next generation of wireless sensor networks," *Sensors and Actuators B: Chemical,* vol. 161, pp. 1-20, 2012.

[20] M. H. Gholizadeh, A. M. Melesse and L. Reddi, "A comprehensive review on water quality parameters estimation using remote sensing techniques," *Sensors,* vol. 16, p. 1298, 2016.

[21] S. N. Zulkifli, H. A. Rahim and W.-J. Lau, "Detection of contaminants in water supply: A review on state-of-the-art monitoring technologies and their applications," *Sensors and Actuators B: Chemical,* vol. 255, pp. 2657-2689, 2018.

[22] C. Desmet, A. Degiuli, C. Ferrari, F. S. Romolo, L. Blum and C. Marquette, "Electrochemical Sensor for Explosives Precursors' Detection in Water," *Challenges,* vol. 8, 2017.

[23] J. K. Atkinson, M. Glanc, M. Prakorbjanya, M. Sophocleous, R. P. Sion and E. Garcia-Breijo, "Thick film screen printed environmental and chemical sensor array reference electrodes suitable for subterranean and subaqueous deployments," *Microelectronics International,* April 2013.

[24] A. M. Syaifudin, K. P. Jayasundera and S. C. Mukhopadhyay, "A low cost novel sensing system for detection of dangerous marine biotoxins in seafood," *Sensors and Actuators B: Chemical,* vol. 137, p. 67–75, 2009.

[25] X. Li, K. Toyoda and I. Ihara, "Coagulation process of soymilk characterized by electrical impedance spectroscopy," *Journal of Food Engineering,* vol. 105, p. 563–568, 2011.

[26] P. Geng, X. Zhang, W. Meng, Q. Wang, W. Zhang, L. Jin, Z. Feng and Z. Wu, "Self-assembled monolayers-based immunosensor for detection of Escherichia coli using electrochemical impedance spectroscopy," *Electrochimica Acta,* vol. 53, pp. 4663-4668, 2008.

[27] G. Charulatha, S. Srinivasalu, O. Uma Maheswari, T. Venugopal and L. Giridharan, "Evaluation of ground water quality contaminants using linear regression and artificial neural network models," *Arabian Journal of Geosciences,* vol. 10, p. 128, 20 March 2017.

[28] N. S. Gunda, S. Gautam and S. Mitra, “Artificial Intelligence for Water Quality Monitoring,” *ECS Meeting Abstracts,* Vols. MA2018-02, p. 1997–1997, July 2018.

[29] X. Wang, F. Zhang and J. Ding, “Evaluation of water quality based on a machine learning algorithm and water quality index for the Ebinur Lake Watershed, China.,” *Scientific Reports,* vol. 7, pp. 12858-12858, 2017.

[30] S. N. Dean, L. C. Shriver-Lake, D. A. Stenger, J. S. Erickson, J. P. Golden and S. A. Trammell, “Machine Learning Techniques for Chemical Identification Using Cyclic Square Wave Voltammetry,” *Sensors,* vol. 19, 2019.

[31] A. Bria, G. Cerro, M. Ferdinandi, C. Marrocco and M. Molinara, “An IoT-ready solution for automated recognition of water contaminants,” *Pattern Recognition Letters,* vol. 135, pp. 188-195, 2020.

[32] C. Bourelly, A. Bria, L. Ferrigno, L. Gerevini, C. Marrocco, M. Molinara, G. Cerro, M. Cicalini and A. Ria, “A Preliminary Solution for Anomaly Detection in Water Quality Monitoring,” in *2020 IEEE International Conference on Smart Computing (SMARTCOMP),* 2020.

[33] M. Molinara, M. Ferdinandi, G. Cerro, L. Ferrigno and E. Massera, “An End to End Indoor Air Monitoring System Based on Machine Learning and SENSIPLUS Platform,” *IEEE Access,* vol. 8, pp. 72204-72215, 2020.

[34] G. Betta, G. Cerro, M. Ferdinandi, L. Ferrigno and M. Molinara, “Contaminants detection and classification through a customized IoT-based platform: A case study,” *IEEE Instrumentation & Measurement Magazine,* vol. 22, pp. 35-44, 2019.

[35] M. Ferdinandi, M. Molinara, G. Cerro, L. Ferrigno, C. Marroco, A. Bria, P. Di Meo, C. Bourelly and R. Simmarano, “A Novel Smart System for Contaminants Detection and Recognition in Water,” in *2019 IEEE International Conference on Smart Computing (SMARTCOMP),* 2019.

[36] I. Nopens, C. Capalozza and P. A. Vanrolleghem, “Stability analysis of a synthetic municipal wastewater,” *Department of Applied Mathematics Biometrics and Process Control, University of Gent, Belgium,* 2001.

[37] H. Janna, “Characterisation of Raw Sewage and Performance Evaluation of Al-Diwaniyah Sewage Treatment Work, Iraq,” *World Journal of Engineering and Technology,* vol. 4, p. 296–304, 2016.

[38] A. Bria, L. Ferrigno, L. Gerevini, C. Marrocco, M. Molinara, P. Bruschi, M. Cicalini, G. Manfredini, A. Ria, G. Cerro, R. Simmarano, G. Teolis and M. Vitelli, "A False Positive Reduction System For Continuous Water Quality Monitoring," in *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2021.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, p. 2825–2830, 2011.

[40] Y. Zhao, Z. Nasrullah and Z. Li, "PyOD: A Python Toolbox for Scalable Outlier Detection," *Journal of Machine Learning Research*, vol. 20, pp. 1-7, 2019.

[41] P. Vanýsek, "ELECTROCHEMICAL SERIES," 2010.

[42] T. P. Lambrou, C. C. Anastasiou, C. G. Panayiotou and M. M. Polycarpou, "A Low-Cost Sensor Network for Real-Time Monitoring and Contamination Detection in Drinking Water Distribution Systems," *IEEE Sensors Journal*, vol. 14, pp. 2765-2772, 2014.

[43] P. Bruschi, G. Cerro, L. Colace, A. De Iacovo, S. Del Cesta, M. Ferdinandi, L. Ferrigno, M. Molinara, A. Ria, R. Simmarano, F. Tortorella and C. Venettacci, "A Novel Integrated Smart System for Indoor Air Monitoring and Gas Recognition," in *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2018.

[44] E. Lotfi and A. Keshavarz, "Gene expression microarray classification using PCA–BEL," *Computers in Biology and Medicine*, vol. 54, pp. 180-187, 2014.

[45] C. Jing and J. Hou, "SVM and PCA based fault classification approaches for complicated industrial process," *Neurocomputing*, vol. 167, pp. 636-642, 2015.

[46] M. O. Faruqe and M. A. M. Hasan, "Face recognition using PCA and SVM," in *2009 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*, 2009.

[47] X. Xu and X. Wang, “An Adaptive Network Intrusion Detection Method Based on PCA and Support Vector Machines,” in *Advanced Data Mining and Applications*, Berlin, 2005.

[48] C. De Stefano, F. Fontanella and C. Marrocco, “A GA-Based Feature Selection Algorithm for Remote Sensing Images,” in *Applications of Evolutionary Computing*, Berlin, 2008.

[49] N. D. Cilia, C. De Stefano, F. Fontanella and A. Scotto di Freca, “Variable-Length Representation for EC-Based Feature Selection in High-Dimensional Data,” in *Applications of Evolutionary Computation*, Cham, 2019.

[50] C. De Stefano, F. Fontanella, G. Folino and A. S. di Freca, “A Bayesian Approach for Combining Ensembles of GP Classifiers,” in *Multiple Classifier Systems*, Berlin, 2011.

[51] N. D. Cilia, C. De Stefano, F. Fontanella, S. Raimondo and A. Scotto di Freca, “An Experimental Comparison of Feature-Selection and Classification Methods for Microarray Datasets,” *Information*, vol. 10, 2019.

[52] L. Ying, G. Yanfeng and Z. Ye, “Hyperspectral Feature Extraction using Selective PCA based on Genetic Algorithm with Subgroups,” in *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, 2006.

[53] Y. Xia, L. Wen, S. Eberl, M. Fulham and D. D. F. Feng, “Genetic algorithm-based PCA eigenvector selection and weighting for automated identification of dementia using FDG-PET imaging,” 2008.

[54] F. Mahmud, M. E. Haque, S. T. Zuhori and B. Pal, “Human face recognition using PCA based Genetic Algorithm,” in *2014 International Conference on Electrical Engineering and Information & Communication Technology*, 2014.

[55] N. Halko, P. G. Martinsson and J. A. Tropp, “Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions,” *SIAM Review*, vol. 53, pp. 217-288, 2011.

[56] G. Ochoa, “Error Thresholds in Genetic Algorithms,” *Evolutionary Computation*, vol. 14, pp. 157-182, June 2006.

[57] L. P.cordella, C. De Stefano and F. Fontanella, “Evolutionary prototyping for handwriting recognition,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 21, November 2011.

[58] I. H. Witten and E. Frank, “Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations,” *SIGMOD Rec.,* vol. 31, p. 76–77, March 2002.