UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS
Département d'informatique et d'ingénierie

Iterative learning framework for modeling multimodal systems

by

Juan Carlos Dávila Mesa

THESIS SUBMITTED AS PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE
DOCTORATE DEGREE IN SCIENCE AND INFORMATION
TECHNOLOGY

November 22, 2019

UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS
Département d'informatique et d'ingénierie

Iterative learning framework for modeling multimodal systems

par

Juan Carlos Dávila Mesa

THÈSE PRÉSENTÉE COMME EXIGENCE PARTIELLE DU DOCTORAT EN SCIENCES ET TECHNOLOGIES DE L'INFORMATION

Membres du jury

Président	Dr. Ahmed Lakhssassi, UQO.
Directeur de thèse	Dr Marek Zaremba, UQO.
Codirectrice de thèse	Dre Ana-Maria Cretu, UQO
Examinateur externe	Dr Adam Krzyzak, Université Concordia.
Examinatrice interne	Dre Rokia Missaoui, UQO.

Le 22 novembre 2019

Combien acquérir la sagesse vaut mieux que l'or! Combien acquérir l'intelligence est préférable à l'argent!
Proverbes16:16

How much better is it to get wisdom than gold! and to get understanding rather to be chosen than silver!
Proverbs16:16

Acknowledgements

Grateful thanks are due to the following individuals set out below for their continuous support during this amazing journey. To my lovely wife, Claudia Marcella and my children Lina and Juan Diego for encouraging me in those difficult moments. To my mother, Ana Adelina and my sister Andrea, for include me in their prayers. I want to express my gratitude to Dr. Marek Zaremba and Dr Ana-Maria Cretu for their invaluable contributions and guidance to the planning, design, and development of this work. Their continued support and financial assistance were essential to help me with reaching my academic goals.

To you all, thank you again!

"The universe is made of stories not of atoms", Muriel Rukeyser.

TABLE OF CONTENT

List of tables

List of figures

List of acronyms

AIC	Akaike information criterion
AUC	Area under the curve
BIC	Bayesian information criterion
CHL-A	Chlorophyll type a
CRISP-DM	Cross-Industry Standard Process for data mining
DOC	Dissolved organic carbon
DWT	Discrete wavelet transform
Erf	Gauss error function
FIR	Finite impulse response
FLH	Fluorescent line height
HAR	Human activity recognition
HPLC	High-performance liquid chromatography
ICA	Independent component analysis
IFOV	Instantaneous field of view
IMU	Inertial Measurement Unit
KNN	k-nearest neighbor
MAD	Median Absolute Deviation
MAE	Mean absolute error
MBR	Maximum Band Radio
MCI	Maximum chlorophyll index
MERIS	Medium Resolution Imaging Spectrometer
MII	Model improvement index
MODIS	Moderate Resolution Imaging Spectro-radiometer
MSE	Mean squared error
NaN	Not-a-number
NIR	Near-infrared
NN	Neural network
PCA	Principal component analysis

POC	PARTICULATE ORGANIC CARBON
RBF	RADIAL BASIS FUNCTION
RL_TOSA	TOP-OF-STANDARD-ATMOSPHERE RADIANCE REFLECTANCE
RMSE	ROOT MEAN SQUARED ERROR
ROC	RECEIVER OPERATING CHARACTERISTIC
RPY	ROLL, PITCH, YAW
RS	REMOTE SENSING
SEAWIFS	SEA-VIEWING WIDE FIELD-OF-VIEW SENSOR
SEMMA	SAMPLE, EXPLORE, MODIFY, MODEL AND ASSESSMENT
SNR	SIGNAL-TO-NOISE RATIO
SVD	SINGULAR VALUE DECOMPOSITION
SVM	SUPER VECTOR MACHINE
TSS	TOTAL SUSPENDED SOLIDS
UTM	UNIVERSAL TRANSVERSE MERCATOR
WBF	WAVELET BASIS FUNCTIONS

Résumé

En pratique, les données collectées par plusieurs capteurs, dans différentes périodes, fusionnées dans un seul jeu de données produisent des fonctions de distribution. Souvent, ces dernières ne peuvent pas être analysées efficacement sur une hypothèse uni-modale. La modélisation de systèmes multimodaux à l'aide de méthodes conventionnelles, telles que les méthodes empiriques, semi-empiriques, semi-analytiques, quasi-analytiques ou analytiques peut dégrader considérablement la précision des modèles empiriques.

Dans ce travail, nous visons à développer un processus automatisé pour modéliser des systèmes multimodaux appliquant un cadre itératif basé sur l'apprentissage automatique. Cette nouvelle technique peut être appliquée à un large éventail de problèmes de régression et de classification notamment avec des applications dans diverses branches d'ingénierie. La méthodologie proposée, appliquée sur un jeu de données, utilise un processus itératif qui, après avoir déterminé le nombre de modes, extrait successivement les meilleurs candidats d'entrainement appartenant à chaque mode. Ensuite, elle classifie le jeu de données en classes binaires et sélectionne de manière itérative de nouveaux ensembles de données étiquetées.

La méthode proposée peut être décrite, succinctement, comme une séquence itérative de procédures de classification et de régression. Elle améliore la fonction de prédiction d'un classificateur donné et, par conséquent, le modèle de données résultant. Nous validerons et démontrerons l'efficacité de la méthode proposée en abordant deux problèmes complexes dans lesquels la multimodalité des données affecte l'extraction de modèles précis. Dans notre premier problème, nous avons estimé la concentration de chlorophylle existant dans le lac Winnipeg dans la province canadienne de Manitoba- un exemple de problème de régression classique.

Notre méthode a démontré que l'introduction d'un mécanisme itératif de sélection de l'échantillon améliore la précision du modèle de prédiction de la concentration de chlorophylle.

Dans le deuxième problème, nous nous sommes concentrés sur la reconnaissance des activités de locomotion humaine telles que marcher, être debout, s'asseoir et se coucher. Cette expérience est basée sur des enregistrements par plusieurs accéléromètres sans fil. Ce problème classique de modélisation d'un système dynamique à partir de données multi capteurs est un exemple de problème de classification. Dans ce scénario, notre mécanisme itératif de sélection des échantillons a amélioré l'exactitude de la classification, tout en accélérant le processus d'entraînement et en minimisant le problème de surapprentissage

Abstract

In practice, data collected by multiple sensors, in different timeframes and merged into a single dataset, produce data distribution functions that often cannot be efficiently analyzed when using strictly a unimodal hypothesis. Modeling multimodal systems using conventional methods, such as empirical, semi-empirical, semi-analytical, quasi-analytical or analytical can markedly degrade the precision of empirical models.

In this work, we aim to develop an automated process for modeling multimodal systems by applying an iterative framework based on machine learning. This novel technique can be applied in a broad spectrum of regression and classification problems with application in various engineering problems. The proposed methodology, applied on a given dataset, uses an iterative process that, after determining the number of modes, extracts successively the best training candidates belonging to each mode, classifies the given dataset into binary classes and iteratively selects new, expanded sets of labeled data. The proposed method can be succinctly described as an iterative sequence of regression and classification procedures that improves the prediction function in a given classifier and consequently in the resulting data model. We validate and demonstrate the efficacy of the proposed method by addressing two complex problems in which data multimodality affects the extraction of precise data models.

In our first experiment, we estimate the chlorophyll concentration occurring in large aquatic areas - an example of classical regression problems. In this case, a wide range of chlorophyll concentration and different types of waters with contrasting optical properties are combined with the interaction of multiple components in the optical data flow, making this problem a difficult one from the standpoint of developing a precise and robust regression models over the entire range of spectral frequencies. Our method has demonstrated that the introduction of an iterative sample selection mechanism improves the accuracy of the chlorophyll concentration model.

In a second experiment, we focus on recognizing human locomotion activities such as walk, lie, sit and stand using readings recorded by multiple wireless acceleration sensors, a classical problem of multi-sensor analysis of a dynamic system and an example of a classification problem. In this experiment, our iterative sample selection mechanism has improved the classification accuracy, while at the same time speeding up the training process and minimizing the problem of overfitting.

CHAPTER I: INTRODUCTION

1.1 Modeling multimodal systems

The integration of multiple sources of information coming simultaneously from multiple sensors can broaden the knowledge about the complex interaction between the variables involved in modeling of the physical phenomenon under observation. In order to obtain an accurate model, the multimodality of the data sources and data distributions should often be considered and properly dealt with in the modeling process.

The multimodality phenomenon can be produced by multiple factors, such as simple covariate shift, prior probability shift, sample selection bias, imbalanced data, domain shift and source component shift [1] and it often occurs when a physical process under observation (energy exchange, mass transfer, variations of optical properties, etc.) is described by different physical quantities, for which measurements are obtained by sensors operating on multiple measuring principles and technologies [2]. Finding a consistent and robust data model is especially challenging when the diversity of information sources is coupled with a large observation time. In practice, information obtained from multiple data acquisition missions, using multiple sensors and technologies, and processed by different teams in different time frames, is frequently merged in a single dataset, producing a data distribution function that cannot be efficiently modeled under a unimodal hypothesis.

Modeling complex systems using conventional methods (e.g. empirical, semi-empirical, semi-analytical, quasi-analytical or analytical) does not always lead to the best precision of the system, especially in the presence of statistical multimodality. Machine learning provides an excellent approach to deal with the aforementioned problem because the system behavior is learned from data empirically, using discrepancies between predictions and the model being trained from data [10], to improve the prediction performance and the model accuracy. It also provides a broad number of options to build data models when an adequate and complete theoretical data model is difficult to obtain due to the number of

variables, the variable interaction, and when the spatiotemporal interdependence is complex [11,6,12]).

1.2 Objectives and contributions

The overall objective of this research is to develop an automated process for modeling multimodal systems using an iterative machine learning approach, which can be applied in abroad number of engineering problems solving both regression and classification tasks. We also aim to reach five specific objectives:

- To allow the modeling algorithm to develop models that span the whole input domain (as opposed to piecewise models).
- To enhance the level of robustness to variations in the quality of input data.
- To integrate a mechanism based on a multimodal hypothesis in order to assess the occurrence of multimodality.
- To validate the proposed approach when modeling multimodal systems in regression problems using spatial data.
- To validate the proposed approach when modeling multimodal systems in classification problems using temporal data.

Building on the general data-driven iterative learning methodology, the thesis solves two complex technical problems: the empirical assessment of chlorophyll type a (chl-a) concentration using in-situ measurements (an example of a regression problem, i.e. the problem of predicting a continuous quantity output, using a spatial data set), and the human locomotion classification using readings recorded by body-worn sensors (an example of classification problem, i.e. the problem of predicting a discrete class label output, using a temporal data set). As such, in contrast to most traditional modeling methods, this work brings an important contribution by providing a method for solving a broad scope of technical problems, offering at the same time the advantage of a reduced number of the training samples and consequently, a reduced time required to build the analytical data model.

1.3 Motivation

This thesis presents a general adaptive machine learning-based solution that deals with data structure complexity, capable of resolving a large class of classification and regression problems. The main motivation for the research came from a large-area environment monitoring problem consisting in the estimation of chl-a concentration in inland waters. The acquisition of multispectral information by the use of remote sensing equipment with different spatial, temporal, spectral and radiometric characteristics is a typical example in which there is a high probability of the occurrence of data multimodality. Building analytical data models for the estimation of chlorophyll concentration is an especially challenging task due to the complex interaction of biophysical variables existing in the ecosystem, and the elaborated sensor data processing procedures. From the perspective of monitoring water quality in in-land areas, chl-a is frequently used as indicator of the ecological health of aquatic environments, which is of key importance to sustain human economic activities like fishing, agriculture and human consumption.

For the purpose of monitoring large areas, the spectral information is generally collected by using remote sensing technology due to its cost-effectiveness and accuracy, comparing with technologies relying on in-situ data acquisition, which are demanding both in terms of resources and time [3]. By using optical satellite imagery, the level of chl-a concentration is determined by the amount of phytoplankton biomass, the latter being responsible to produce distinct changes in watercolor, and its effect can be detected using optical properties of the incident light or the reflectance [4]. This reflectance, used in the form of indices derived from the shape of the spectral characteristics, is combined in band ratios to build data models using the regression analysis.

In order to extend the investigation of multimodal systems into a classification type of problem we addressed the issue of human locomotion recognition by using readings obtained from wireless wearable sensors. Wireless wearable sensor technologies are gaining interest in research communities due to the increased availability of significantly miniaturized electronic components, with low power consumption, which makes them the

standard data acquisition equipment for a variety of applications related to human activity recognition in indoor and outdoor environments. Wireless sensors are an excellent solution for gathering information in health rehabilitation, respiratory and muscular activity assessment, sports and safety applications [5], allowing users to perform natural execution of any physical activity. When collecting information from multiple wireless wearable sensors to recognize human locomotion activities (e.g., in popular Fitbits), readings are affected by various factors, such as sensor data alignment, data losses, and noise, among other experimental constraints, all introducing a bias in the resulting data model [6, 7]. This situation is even more challenging when solving multi-class classification problems [8], because samples with different class membership can be found in the same spatial region, increasing the complexity when using traditional modeling methods.

These two complex technical problems have driven this research and provided implementation and testing platforms for the proposed system modeling solution, such as reported in [13-19].

1.4 Thesis outline

This thesis contains eight chapters. After the Introduction, Chapter 2 presents the multimodality problem, focusing on large-area monitoring systems and multi-sensor analysis of dynamic systems. In Chapter 3, a literature review and state-of-the-art in multimodal modeling and machine learning are presented. The research objectives and contributions are described in detail in Chapter 4. An overview of the proposed methodology is presented in Chapter 5. The application of the proposed approach in solving two kinds of engineering problems (regression and classification) is illustrated in Chapters 6 and 7, and conclusions are presented in Chapter 8.

CHAPTER II: THE MULTIMODALITY PROBLEM

Modeling multimodal systems is affected by miscellaneous problems occurring in the data acquisition process, for instance the measurement uncertainty, outliers, sensor alignment, noise and data correlation among other factors that impact negatively the input data quality, biasing the resulting data model. In practice, the collected data - obtained from multiple sensors in different timeframes - are merged into a single dataset, producing a data distribution function, which often cannot be evaluated by using a unimodal hypothesis. This is because datasets are often not coincident in both time domain (nonstationary data) and frequency domain as a result of time delays in the process of data acquisition and information processing. The combination of statistical modality of data related to the data distribution and the aforementioned data acquisition considerations make the analysis and design of data models a challenging endeavor. The presence of multimodality in the data, and the resulting fluctuation of the data distributions in the datasets associated with each modality can result in a substantial degradation of the precision of empirical models, as further discussed in Sections 2.3 and 2.6.

2.1 Modeling multimodal systems

Depending on the data application, modeling multimodality systems can be approached by using deterministic or stochastic modeling solutions [20]. A deterministic approach is when the resulting data model depends on the observed phenomena and it does not consider the complexity of the inputs or its underlying processes. A stochastic approach uses random features drawn from a possible data distribution to conduct multiple simulations. While both these modeling approaches present interesting advantages, neither of them is a good candidate for modeling when data structures change dynamically or when theoretical models are difficult to obtain [6, 11, 12].

The key factor to build a solution that considers the complexity of multimodal variable interactions is the determination of the grade of modality (number of modes) presented in

system. There are two general modeling approaches, when the problem is reduced to a unimodal case, and when the problem is modeled as a composition of several unimodal problems. In the first case, the problem is reduced to finding the dominant single mode by using traditional dimension reduction procedures, such as principal component analysis (PCA) or independent component analysis (ICA) [20]. This is a valid approach when problems are not subject to inherent high dimensionality; otherwise, statistical analysis methods or optimized models must be used, such as in the technique based on kernel density estimates introduced by Bernard W Silverman, commonly used to determine the number of modes in the input domain [21-25]. Some of the most popular methods to estimate multimodality use histograms, kernel density estimates, and mixture models (Bayesian and Markov) [21]. Table 1 summarizes some of the techniques used to determine the grade of multimodality.

Determining the grade of multimodality	Algorithms
Problem is reduced to a unimodality case.	• PCA or ICA [23] • Histograms, kernel density estimates and mixture models [26] • Silverman's test [21-25]
The excess mass approach	• Silverman's test [24] modified.

Table 1. Modality assessment

The principal component analysis (PCA) algorithm is a dimension reduction tool that can be used to reduce a large set of variables to a small set that still contains most of the information from the large set. The idea is to transform a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible and each succeeding component accounts for as much of the remaining variability as possible. For a given dataset, PCA can deconstruct the distribution using eigenvectors and eigenvalues. The eigenvectors are linear combinations of the original variables and

weighted by their contributions to explain the variance in an orthogonal dimension. These eigenvectors (direction) exist in pairs with their eigenvalues, which are numbers that measure the amount of variation in the given dataset. The PCA method was implemented in our framework, as presented further detailed in Chapters 6 and 7 in order to reduce the dimensionality in our datasets. Figure 1 illustrates the use of PCA to transform high data (3 dimensions) to low dimension (2 dimensions) [144].

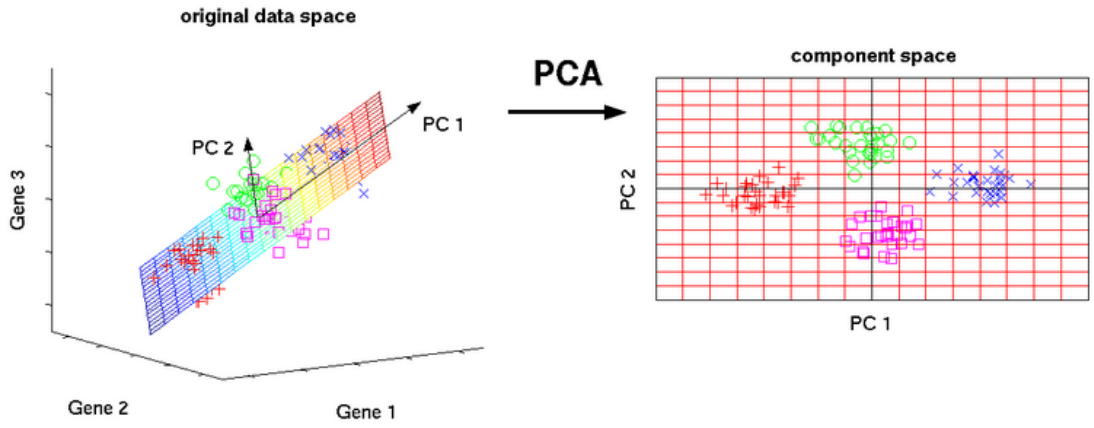


Figure 1. Three-dimensional gene data reduced to a two-dimensional space using PCA [27]

Like PCA, the independent component analysis (ICA) is another statistical technique intended to deconstruct the input data into a set of vectors, which are independent components for the given data. All components are equally important and mutually independent. This characteristic makes ICA an ideal solution to deal with cases when input domain is noisy, and when features cannot be correlated due to non-Gaussian, non-linear and non-stationary conditions [28].

Although ICA is similar to PCA, the ICA algorithm is computationally demanding, especially in terms of memory capacity, that's why it was not implemented in our framework. In many problems, it is common to face the issue of latent variables and the way to associate collected values to those latent variables. In our second experiment, for example, we found cases when some sensors were not recording information and dataset values where registered as not-a-number (NaN). In these cases, the mixed model strategy

is the most convenient way to deal with this problem. In mixed models, the latent variable corresponds to the mixture component. In general, a mixture model assumes that data are generated by the following process: sample latent variables and then sample the observables x from a distribution which depends on the latent variables, called z, formulated as follows:

$$p(z, x) = p(z)p(x|z) \quad (1)$$

where $p(z)$ is always a distribution and $p(x : z)$ can take a variety of parametric forms, for example a Gaussian distribution, in this case the resulting model is referred to as a "mixture of Gaussians" [29]. A reference to this method is found during the assessment of the multimodality of the chlorophyll concentration in Chapter 6. Finally, the Silverman's test is a methodology to determine modality based on kernel estimates. The term "kernel" defines a special type of probability density function (PDF) with the additional following characteristics: non-negative, real-valued, even and its integral over its support set must be equal to 1 [30]. Some of the most popular kernels are as follows: triangular, parabolic and Gaussian. The problem with this method occurs when the input domain includes a mixture of component distributions with low and high variance, resulting in low accuracy performance. To cope with this problem, auxiliary methods such as the dip test and the excess mass estimates [31,32] can be deployed.

2.2 Multimodality in remote sensing

As stated in the introduction, this thesis aims to propose, implement and validate an automated process for modeling multimodal systems using an iterative machine learning approach in regression problems using spatial data as well as in classification problems using temporal data. The following sections explain the issue of large-scale environment and human motion monitoring, the technology behind the monitoring process, and how this technology impacts the modality of the acquired datasets.

The first implementation problem dealt with in this thesis is defined and approached in terms of solving a regression problem in the context of the remote sensing technology.

Remote sensing is a science of acquiring information about the earth's surface without being in contact with it. This is made possible by sensing and recording the reflected or emitted energy. Figure 2 illustrates the basic components required in a remote sensing system. The process starts when a source of energy produces the light required to illuminate a target of interest. The electromagnetic energy, in form of radiation, travels from the source of energy to the target, interacting with it. Depending on the target properties, the resulting reflecting energy travels from the target to the sensor that is not in direct contact with the target interacting with the atmosphere as depicted in Figure 2.

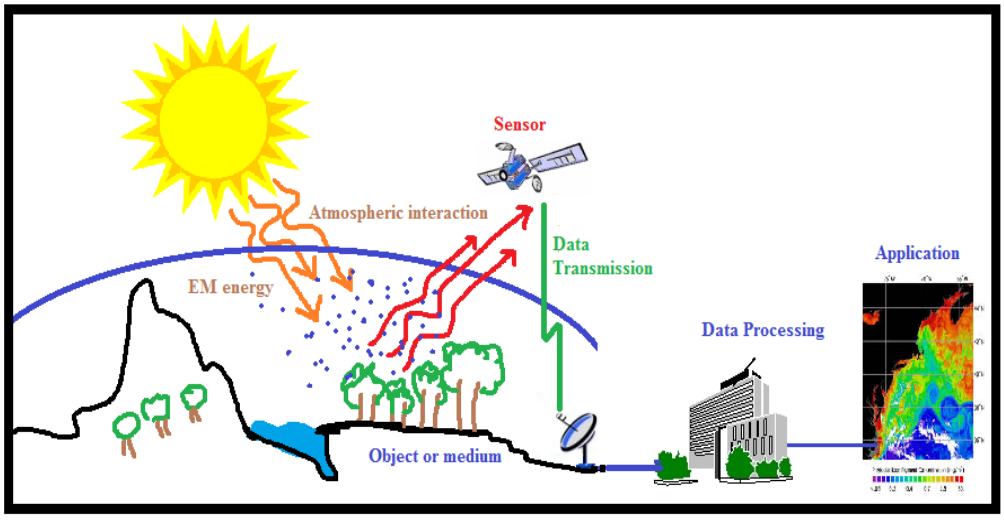


Figure 2 Remote sensing components

The energy emitted or scattered from the target is collected and recorded by the sensor. During this period of traveling, the light, which comes from the source to the target of interest and from the target to the sensor, interacts with the atmosphere as it passes through it . The readings are transmitted to earth stations in which data are processed and converted into images. This process ends when the resulting images are interpreted to extract information about the target of interest to solve a particular problem. However, repeated observations of a given area over time produce radiometric inconsistency due to changes in sensor calibration, differences in illumination and observation angles, and variations in atmospheric effects [83], resulting in a data model with a degraded predictive power.

Remote sensing technologies play a key role in environmental monitoring and modeling, impacting multiple disciplines, such as hydrology, meteorology, forestry and geography. They are widely used for monitoring water quality over large aquatic basins due to the cost and flexibility advantages associated to their deployment. Some of the best-known technologies are the low-resolution Sea-viewing Wide Field-of-view Sensor (SeaWiFS), Moderate Resolution Imaging Spectro-radiometer (MODIS), launched in 2002 on the Aqua satellite, and the Medium Resolution Imaging Spectrometer (MERIS), launched in 2002 on the ENVISAT platform. Figure 3 illustrates three images obtained from MERIS, MODIS Agua and SeaWiFS sensors. In this thesis, we use empirical data collected by MODIS and MERIS to estimate chl-a concentration.

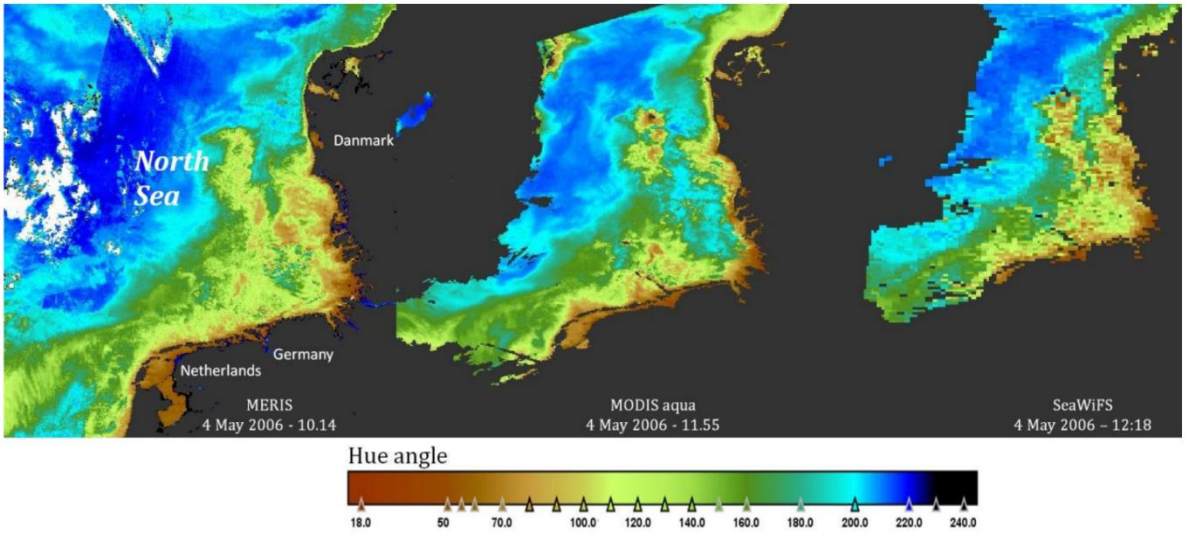


Figure 3. True color classification of natural waters using Hue angle image processing of MERIS, MODISA and SeaWiFS (left to right). Area North Sea, date 4 May 2006 [84]

Remote sensing readings are subject to bias because electromagnetic energy is affected by the atmospheric and target interactions that produce reflection, refraction, scattering and absorption, thus distorting the collected data. Raw data contain additive atmospheric noise, which must be de-noised by a radiometric correction process, as presented in Figure 4. The source of radiometric noise depends on the sensor technology, the imaging mode, and the way it is used to capture the image [89]. Another source of data distortion is the variation of the viewing geometry (sensor-earth). In this case, a geometric correction process is required. Some sources of this type of errors are as follows: the variation of altitude, the

relief displacement, and nonlinearity in the sweep of the sensor instantaneous field of view (IFOV). Once data acquisition process is completed and the information is recorded, the image data and auxiliary data are used to produce the final application product.

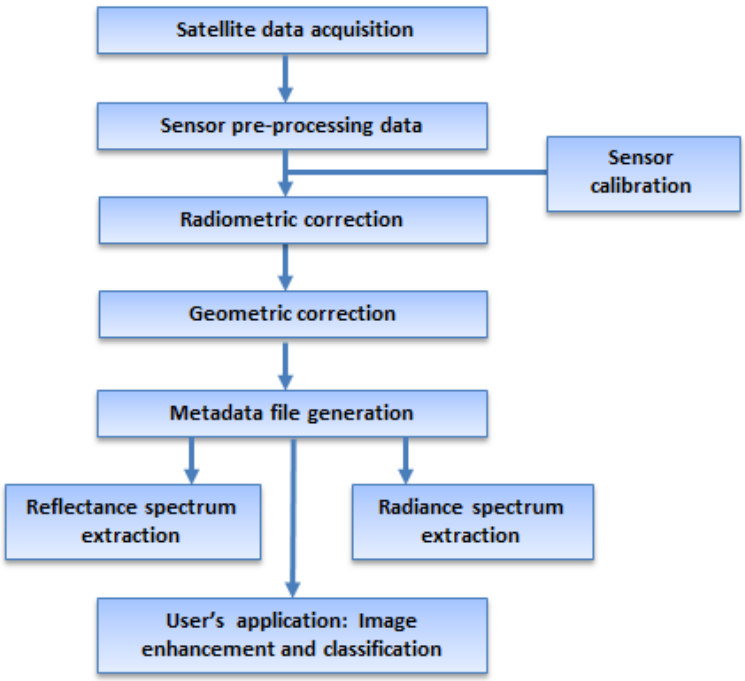


Figure 4. Remote sensing data processing

Remote sensing of water constituents requires a careful atmospheric correction since more than 90% of the upward directed radiance at the satellite altitude comes from the atmosphere, including direct sunlight and skylight, which are specularly reflected from the water surface. Small errors in determining the optical properties of the atmosphere may induce large errors in the retrieval of water constituent concentrations [90]. Figure 5 shows the atmospheric correction process for medium resolution imaging spectrometer (MERIS) sensor. The various elements presented in this figure are described in [90] as follows:

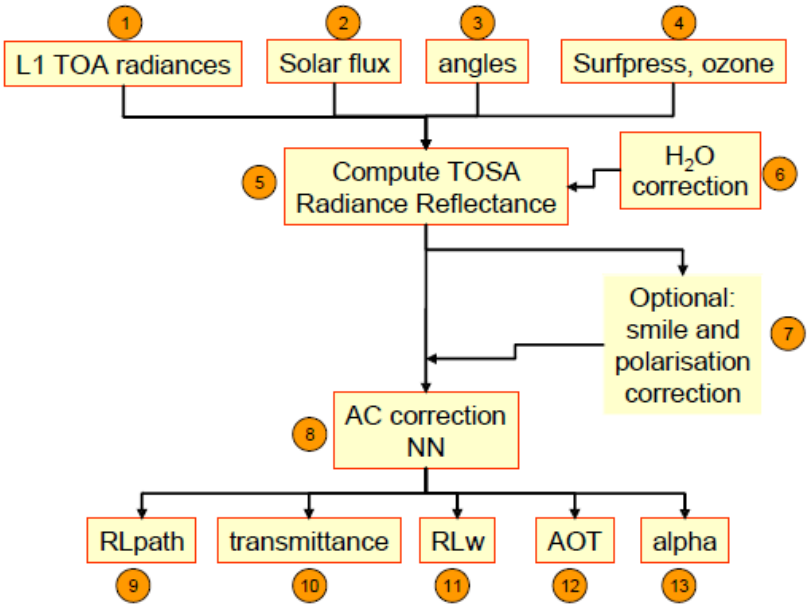


Figure 5. Atmospheric correction process for MERIS sensor [90]

Inputs (1 – 4): are the input values to the procedure obtained from the MERIS image pixel by pixel, except for the solar flux. The angles are converted into Cartesian coordinates to avoid the angle problems around the nadir angles.

Compute TOSA (5): is a module which computes the top-of-standard-atmosphere radiance reflectance (RL_TOSA) using the deviation of the atmospheric pressure and ozone concentration from the standard values, i.e. 1013 hPa and 350 Dobson units (DU), respectively. This module considers also the altitude of the target lake in the pressure calculation.

Water Correction (6): is the module for the correction of the influence of water vapor on band 9 (708 nm). It uses the standard algorithm as implemented in the instrument processing facility (IPF).

Optional (7): optional procedures for correcting or reducing the camera boundary problem and for considering the polarization in the atmosphere.

AC correction (8): The atmospheric correction neural network (NN), which considers the influence of aerosols, thin cirrus clouds, sun and sky glint and the water leaving radiance.

Outputs (9 – 12): output of the NN: (9) path radiance reflectance, i.e. radiance entering the sensor from all sources above the water surface, (10) transmittance, (11) water leaving radiance reflectance, (12) aerosol optical thickness for 4 wavelengths.

Aerosol angstrom (13): coefficient alpha computed from the aerosol optical thicknesses at 443 and 865 nm. Depending on the application type, readings are also affected by data values fluctuating from one location to another or over the time at the same location producing outliers that are recorded in the collected data. These outliers are categorized in [91] as follows:

Unusual time series snippets: We can identify Earth observation data patterns such as diurnal or seasonal cycles which are relatively stable. However, there exist snippets in a time series that deviate from the stable pattern. That is the case in the images in Figure 6 where the duration of high brightness temperatures each summer is relatively stable. However, one snippet, the high brightness temperature, persisted longer than usual. This error could be caused by either unusual natural events or reading errors [91].

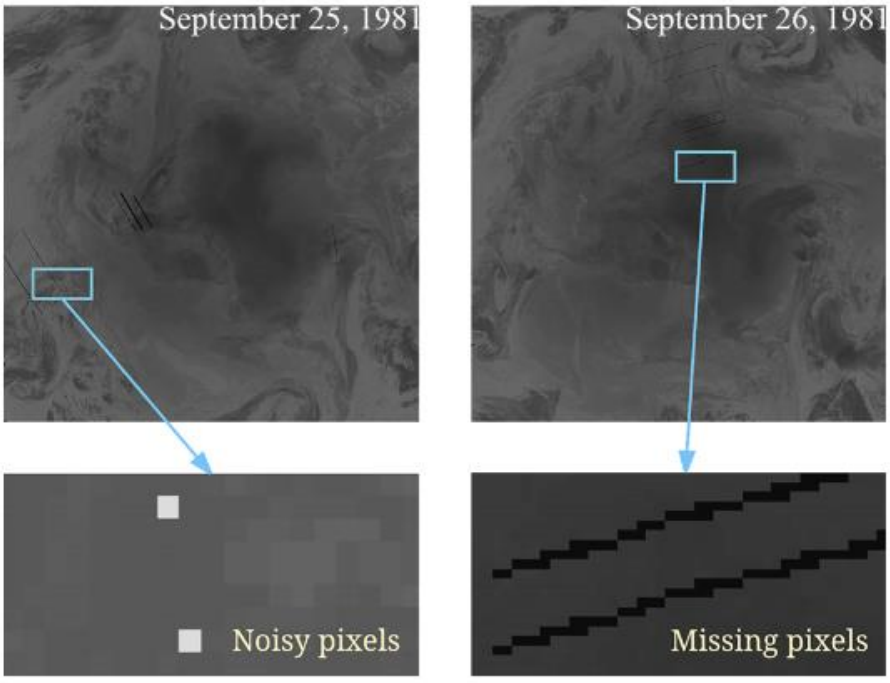


Figure 6. Unusual time series on two different sampling dates [91]

Level shifting: Figure 7 also shows a situation when a group of adjacent pixels significantly increases or decreases, causing a temporal discontinuity that may appear normal when viewed spatially at a specific time and can only be discovered when viewed as a time series at a given location [91].

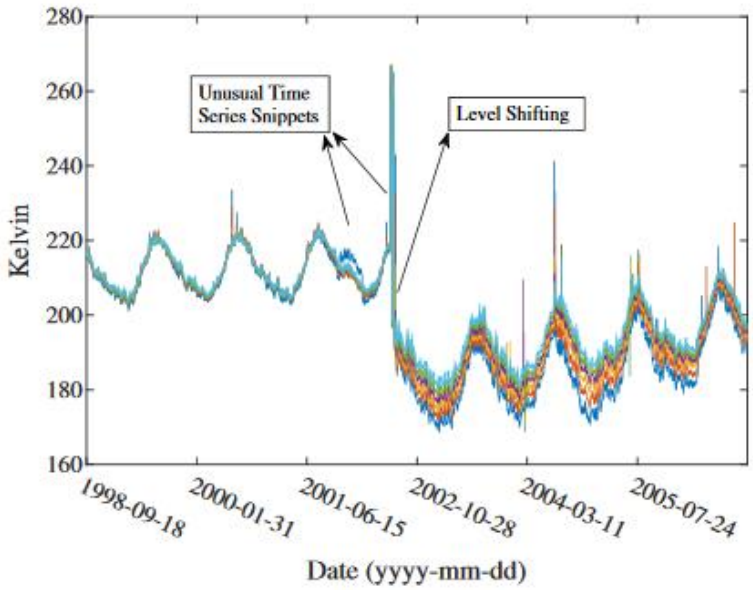


Figure 7. Unusual time series and level shifting [91]

Local spatial outlier: As shown in Figure 8, pixel A is an outlier with respect to its neighbors, but normal when viewed globally. Pixel B has the same value as pixel A but is not a local spatial outlier [91].

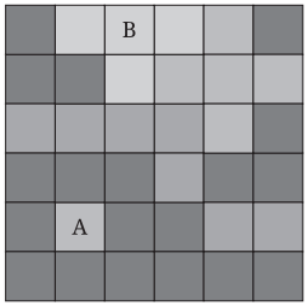


Figure 8. An illustration of objects and local spatial-temporal neighborhoods. Pixel A and B have the same value [91]

The occurrence of level shifting and unusual time series snippets in readings obtained in our experiments produced statistical modality issues considered during the design of our methodology (see Table 5 and 6, section 3.5, Chapter III).

2.3 Wearable wireless sensors for recognizing human activity

The second scenario in which the automated process for modeling multimodal systems using an iterative machine learning approach is validated is a classification problem in the context of human motion monitoring using wearable sensors. This section reviews the sensor system used in human locomotion recognition and how this technology impacts the modality of the acquired datasets.

The new wearable technology used to recognize human activity is becoming extremely attractive to customers in a wide area of applications, ranging from fitness to clinical monitoring, for both indoor and outdoor environments. These applications allow users to achieve a natural execution of any physical activity, while providing good results in multiple practical applications, such as health rehabilitation, respiratory and muscular activity assessment, sports and safety applications [5]. Wearable sensor technologies are gaining interest in research communities due to the use of significantly miniaturized electronic components, with low power consumption. Currently, wearable sensor solutions include devices with the capacity to register, amplify, process and transmit information about the target of interest, reducing direct human intervention and allowing integration between body-worn sensor and ambient sensors with excellent applications in monitoring patients or athletes, while providing more efficacy during assessment. A wide offer of wireless sensors is available, such as accelerometers, gyroscopes, barometers and other devices with low power consumption. With a growing market calculated in 560 million units a year by the end of 2021, this market is widely dominated by accelerometer sensors, which use the design principle of Newton's law and Hooke's law [61]. Designed to sense one, two or three axes, accelerometers use the piezoelectric effect in crystal structures, or capacitors for sensing changes in capacitance. In both cases an accelerative force interacts with crystals or capacitances generating a voltage from the applied stress, and the accelerometer interprets the voltage to determine the corresponding velocity and orientation. Other type of sensors can make use of hot air bubbles, piezo-resistive effect and light [81].

The interpretation of data collected by such sensors, when characterizing the type of activities being executed by a user, is still a significant challenge. The main problems related to modeling multimodal systems in wireless sensors are: the complexity of human activities (i.e. certain activities contain similar gestures), the extraction of relevant features, data loss that characterizes any wireless transmitter, and complex data pre-processing required to deal with factors related to data alignment. Other problems such as data losses, experimental constraints and noise inherent in the collected measurements [54, 6] depreciate the data quality and the final model's accuracy [6]. The non-ergodicity of the acquisition process from acceleration sensors, will result in poor performance [80] and searching patterns becomes a challenging process [82].

The new wearable technology used to recognize human activity is becoming extremely attractive to customers in a wide area of applications, ranging from fitness to clinical monitoring. A wide offer of wireless sensors is available, such as accelerometers, gyroscopes, barometers and other devices with low power consumption. As the data we are using in the context of this thesis is coming from accelerometers, we will focus the discussion on this category of wearable sensors only. An accelerometer is an electromechanical device that measures acceleration forces [81].

One of the most typical accelerometers is the piezoelectric. A basic diagram is exhibited in Figure 9. This device uses a quartz crystal or a polycrystalline ceramic material. Due to the Newton's second law of motion, the mechanical stress produced by the acceleration force acting against the material is equal the change in the electrical charge within the material [86]. As presented in the Figure 9, signal leads connected to the piezoelectric material are connected to a circuit to make the signal suitable for display or recording. Typical accelerometers are constructed to monitor multiple axes. For example, to determine two-dimensional movement, a 2-axis unit is required, and to monitor three-dimensional positioning, a 3-axis unit will be required. Most smartphones typically make use of three-axis models, whereas cars simply use only a two-axis to determine the moment of impact. The sensitivity of these devices is quite high as they are intended to measure

even very minute shifts in acceleration. The more sensitive the accelerometer, the more easily it can measure acceleration [85].

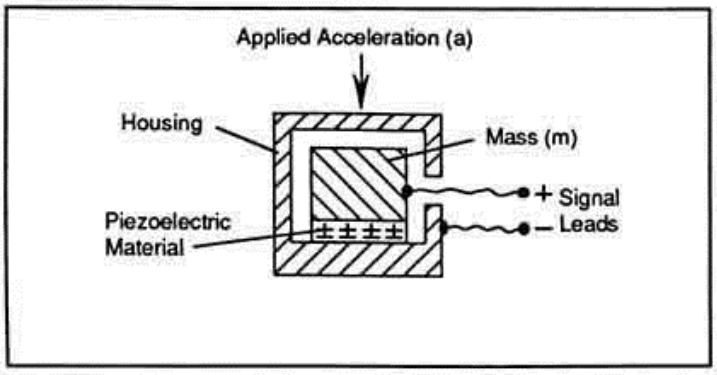


Figure 9. Piezoelectric accelerometer [86]

The interpretation of the data collected by such sensors when characterizing the type of activities being executed by a user still brings serious challenges to developers [54], as described in Table 6, section 3.5 in Chapter III.

In practice, the user understands their test requirements well. However, data analysis runs into difficulty when matching the test requirements with available accelerometer models. There exists, then, a need for a comprehensive description and explanation of accelerometer specifications that manufacturers routinely use [87]. As the data we are using in the context of this thesis was acquired using piezoelectric accelerometers, key specifications used to describe piezoelectric accelerometers [88] are important to define, essentially because these parameters are source of introducing statistical multimodality, i.e. the measurement range represents level of acceleration supported by the sensor's output signal specifications, typically specified in ±g (gravity, where $g = 9.8\frac{m}{s^2}$). This is the greatest amount of acceleration the sensor can measure and accurately represent as an output. The sensitivity defines the ideal, straight-line relationship or ratio between the sensor's electrical output to mechanical input (e.g., gray dashed line in Figure 10). Sensitivity is specified at a particular voltage and is typically expressed in units of millivolts per gravity [mV/g] for analog-output accelerometers.

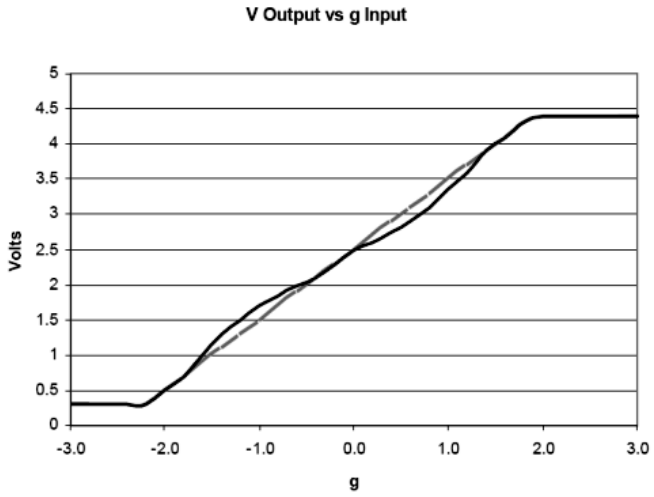


Figure 10. Piezoelectric sensor response [88]

In general, the cumulative effect produced by collecting information from this type of accelerometers, especially with different frequency response, sensitivity and range of operation parameters could produce issues such as the level shifting, distortion and clipping in the output signal that is detected as statistical multimodality during the data acquisition process.

2.4 Conclusions

In practical problems, multimodality in data sources produces a cumulative effect in the accuracy and quality of the final data models. For example, in the context of remote sensing, optical properties from different sensors might introduce marked differences on readings from a target under observation. Assessing the grade of modality (number of modes) in a multimodal system becomes a key factor to build a data model that considers the interaction of multiple input variables, while dealing with different levels of noise originated from a wide range of technical issues such as outliers, level shifting and usual time series snippets that could be merged with other data sources. Similarly, in the context of human recognition using multi-sensor systems, the identification tasks require processing of a large amount of sensor data as well as considering the influence by a varying degrees of human activities, over the period of observation, of the noise produced by the wireless components, of the bandwidth limitation and of the sensor's technical specifications, making the construction of a precise and robust data model challenging.

CHAPTER III: STATE OF THE ART

The main objective of the state-of-the-art review in this chapter is the exploration of the most used data analysis methodologies, with the focus on the data learning process. Given a strong component of the thesis related to the verification of the presented methodology through the solution of two distinct engineering problems, a broader perspective of analyzing and designing intelligent engineering systems is also included. Encouraged to explore the most used data analysis methodologies accepted by industry and scholars, we have included section 3.1 to explain the data learning process, which provides a reference to understanding the problem of the random and false discoveries. Essentially, data preparation and problem understanding were challenging activities in our experiments, mainly because raw datasets were affected by the lack of ground-truth information, non-ergodicity, scarcity, data overlapping, and in some cases by excessive noise. In section 3.2, we provide a brief overview of machine learning methods and other learning methodologies currently used including the iterative learning and its taxonomy, focusing on the concept of the training sample selection and the way it can improve performance prediction and the resulting data model precision. In section 3.3 we present the metrics used to measure the model performance of our method. Finally, in section 3.4, we review the main challenges encountered when applying machine learning in remote sensing and human recognition problems.

3.1 Data mining architectures

One of the most used methodologies in data science and highly recommended to improve the structure of the process of problem analysis the Cross-Industry Standard Process for data mining (CRISP-DM). CRIPS-DM has been widely adopted by many industries as a tool for data analysis because it is soundly based on the practical, real-world experience of how people conduct data mining projects. The methodology consists in a cycle of six phases, which are shown in Figure 11. The sequence of the phases is not rigid; the outcome of each phase determines the next task or phase to be performed, and arrows indicate the most important and frequent dependencies between the phases [48].

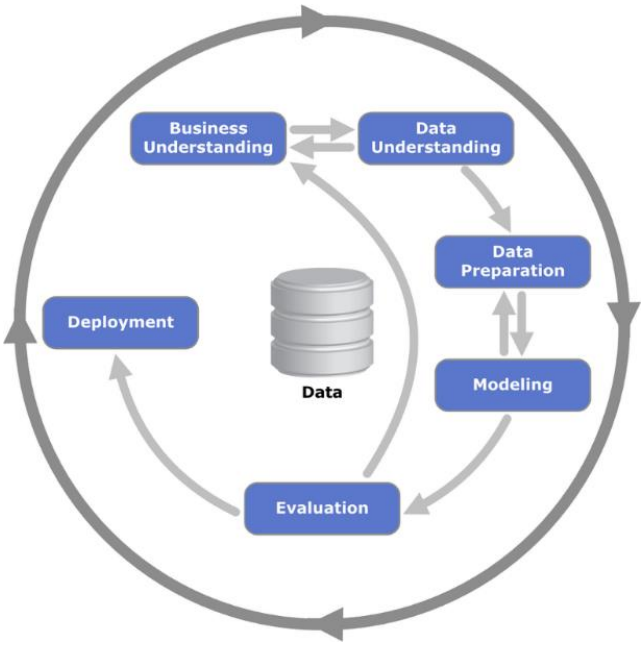


Figure 11. Phases of the CRISP-DM reference model [48]

The first phase of CRISP-DM or Business Understanding or problem understanding focuses on understanding the objectives and requirements; the outcome will be a preliminary plan designed to achieve the objectives. The data understanding phase starts with the initial data collection, checking their quality, exploring of data, and becoming familiar with the problem, getting the insights on the data as well as to form hypotheses regarding the hidden information.

The data preparation phase includes all activities required to construct the final dataset, which serves as an input to the modeling tool in the next step. Data preparation tasks are likely to be performed multiple times and not in any prescribed order. Tasks include table, record, and attribute selection, as well as transformation and cleaning of data. In the modeling phase, various modeling techniques are selected and applied on the cleaned data, and their parameters are calibrated to optimal values. Once the model is built, it is important to determine if the results meet the original objectives, reviewing those aspects that have not been sufficiently considered. This phase is called evaluation. At the end of this phase, a decision on the use of the data mining results should be reached. The model also defines the development phase that will depend on the requirements.

The deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process across the problem space. In this phase, the resulting data models are applied and used to set up for continuous mining of the data [48]. It can be noticed that CRIPS-DM matches the general, sequential, waterfall-type learning process exhibited in Figure 12.

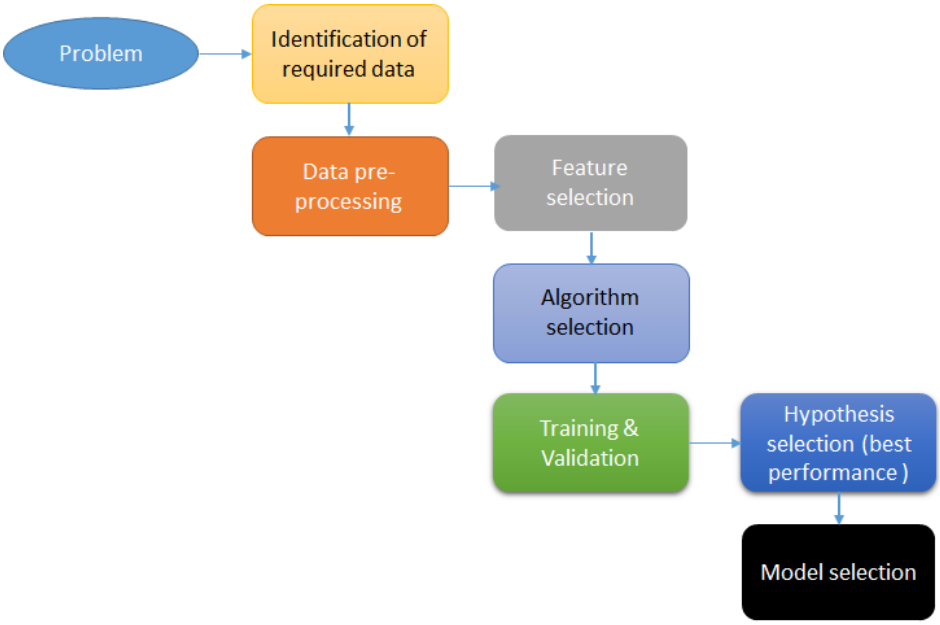


Figure 12. General learning process

In Figure 12, the 'Identification of the required data' process determines the nature of the target and the problem complexity. Data pre-processing and feature selection determine the size of the samples, attempting to maintain quality, while removing spurious combination between features and their interrelationships.

Sometimes, construction of new features and their combinations may improve the accuracy of the model, thus leading to the creation of more concise and accurate classifiers. Some of the most typical problems encountered in this process are related to cleaning the data [34, 35], missing data issues [36], data formatting [34], data transformation and data reduction [37]. Machine learning – arguably the most important stage - uses computational

methods and experience to improve performance or to make accurate predictions. Validation and selection of best performing hypothesis will lead us to the completion of the model selection process.

Another methodology frequently used in data mining is called SEMMA, acronym that stands for Sample, Explore, Modify, Model and Assessment, illustrated in Figure 13. It was commercially implemented in 2008 [49]. This method presents five distinct stages of knowledge discovery as follows [50]:

Sample: This is where a portion of a large dataset (big enough to contain the significant information yet small enough to manipulate quickly) is extracted. For optimal cost and computational performance, some (including the SAS Institute) advocate a sampling strategy, which applies a reliable, statistically representative sample of the full-detail data that will be used during training (used for model fitting), validation (used for model assessment and to prevent overfitting) and testing (to review model performance and generalization).

Explore: After sampling data, the next step is to explore them visually or numerically to identify inherent trends or groupings. Exploration helps to refine and to redirect the discovery process, allowing the user to search for unanticipated trends and anomalies in order to gain a better understanding of the dataset.

Modify: This step aims to create, select and transform the variables upon which to focus the model construction process. Based on the discoveries in the exploration phase, one may need to manipulate data to include information such as the grouping, significant subgroups, or to introduce new variables. Because data mining is a dynamic, iterative process, this step is required to update data mining methods or models when new information becomes available.

Model: Once data are prepared, models are built to explain patterns in the data. The acceptable model performance depends on searching for combinations of variables that

reliably predicts the desired outcome. Modeling techniques in data mining include artificial neural networks, decision trees, rough set analysis, support vector machines, logistic models, and other statistical models, such as time series analysis, memory-based reasoning, and principal component analysis.

Assess: In this step, we evaluate the usefulness and the reliability of findings from the data mining process, assessing how well models perform. A common means of assessing a model is to apply it to a portion of dataset put aside (and not used during the model building) during the sampling stage.

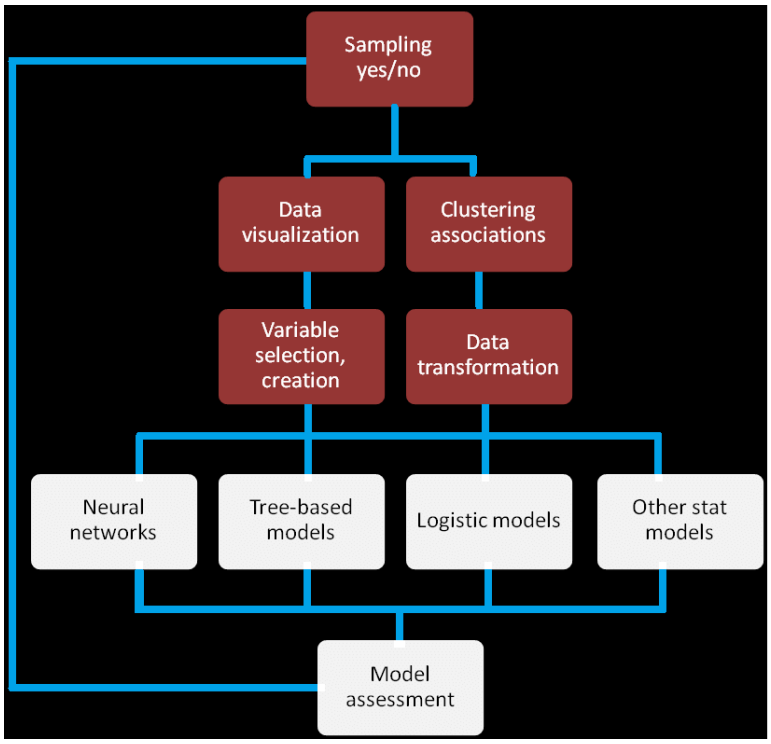


Figure 13. Phases of the SEMMA reference model [50]

Table 2 shows each of the three methods previously described and their equivalent processes.

General learning process	CRISP-DM	SEMMA
Identification of required data	Data understanding	Sample and explore
Data preparation and feature selection	Preparation phase	Modify
Selection and parameter tuning	Modeling process	Model
Training and Validation	Evaluation process	Assess

Table 2. Comparison of standard data mining processes

Once the data is prepared and the features are selected, the next phase consists in choosing an appropriate algorithm (learner) for the task. This phase is critical, because it helps to control and optimize the model accuracy.

Almost every algorithm comes with a large number of settings or hyper-parameters that must be specified [38]. The hyper-parameter tuning is in general a difficult problem. The hyper-parameters are considered as the parameters of a learning algorithm, for example in the *k*-nearest neighbor algorithm (KNN), the value of integer *k* is a hyper-parameter. According to [39], the process of finding the best-performing model from a set of models that were produced by different hyper-parameter settings is called model selection and it is often determined by the bias-variance trade-off used to fixe hyper-parameters in the learning algorithm. For example, a small training dataset produces high bias and low variance when using Naïve Bayes and that represents low computational time, which is an advantage over low bias and high variance in classifier like KNN.

In machine learning, parameter tuning, training and model validation are important tasks and they are needed to maximize the model accuracy. In real problems, it is critical to measure the performance achieved by a learning algorithm. Let us consider a scenario for a supervised learning algorithm. Three datasets are required during the learning process: training set, validation set and testing set. The validation set is used to avoid the phenomenon called overfitting [40]. Once the learner is trained on the training set, the resulting data model is tested on the testing set. The learner's performance is measured by comparing the predicted labels with unseen samples (which were not available during the training process). Only accuracy measured on an independent test set is a fair estimate of

accuracy on the whole population [40]. However, we might wonder about the number of examples needed on each set to learn successfully and how to split them. The most common methods are as follows: dataset split and cross-validation.

Dataset split: aims to achieve low variance over the model parameters. This technique divides the input domain randomly in training-validation and testing sets. A common practice is to start with 80%-20% split (80% training and 20% test).

Cross-validation: A common practice to exploit the label data for both model selection and training is called *n*-fold-cross-validation. This technique is extensively used, mainly because the amount of labeled data is often too small to set aside a validation sample since that would leave an insufficient amount of training data. The process defines a vector of free parameters of the algorithm denoted as θ. The method consists of first randomly partitioning a given sample *S* of m labeled examples into n subsamples, or folds. The *i*th fold is thus a labeled sample $\big((x_{i1}, y_{i1}), .., (x_{mi}, y_{mi})\big)$ of size m_i. Then for any $i \in [1, n]$, the learning algorithm is trained on all folds but the *i*th fold to generate a hypothesis h_i, and the performance of h_i is tested on the *i*th fold as shown in Figure 14. The parameter value θ is evaluated based on the average error of the hypothesis h_i, which is called cross validation error. This quantity is computed as [41]:

$$\hat{R}cv(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i} L(\, h_i\big(x_{ij}\big), y_{ij}) \tag{2}$$

The folds are chosen to have equal size, that is $\boldsymbol{m_i = \frac{m}{n}}$ for all $\boldsymbol{i \in [1, n]}$. In this formula, *L* represents the loss function.

Figure 14. Partitioning of training data into 5 folds

The special case in *n*-fold-cross-validation where *n=m* is called leave-one-out cross validation (the value of *n* is fixed to m, where m is the size of the dataset), since at each iteration exactly one instance is left out of the training sample. In general, leave-one-out is computationally costly, because it requires to train *n* times in samples of size *m*-1 [41]. In general, the appropriate choice of the number of folds is subject to a trade-off between bias (how well the model can approximate the data) and variance (model's ability to respond to new data) and it might depend on the size of the dataset in some cases.

Some of the algorithms are sensitive to the hyper-parameter settings and their selection could be challenging because the relationship between parameters and model performance is intrinsic and vague [42], imposing a deep "searching" of the parameter space for the optimum values that will produce the lowest variance. Some typical solutions are as follows [43]: grid search [44, 45], random search [45, 46] and Bayesian optimization [47].

Grid search is the most basic hyper-parameter tuning method. This technique aims to build a model from which global optimum parameters could be computed. Random search is widely seen in optimization algorithms and it differs from grid search in that discrete parameter values are not provided; instead, statistical distribution for each parameter is provided from which values may be randomly sampled. The reason to prefer this technique over the grid search is because in many cases, parameters are not equally important [45, 46]. Bayesian optimization is a powerful strategy for finding the extrema of objective functions that are expensive to evaluate. It is applicable in situations where one does not

have a closed-form expression for the objective function, but where one can obtain observations (possibly noisy) of this function at sampled values.

Machine learning does not deal with the interdependency of the input variables or their interaction with the environment around the target or system under observation, but deals with the system's behavior, learning data structures empirically, using loss functions to express discrepancies between predictions and the model being trained from data [3]. Machine learning is defined as computational methods using experience to improve performance or to make accurate predictions [51]. Indeed, the amount of data and its quality are essentials to reach the level of precision and learning generalization required to perform valid predictions (including low generalization error and entropy) by a learner. In general, the most common problems tackled in machine learning are:

- Classification: involves assigning a category (labels) to each item; it could be simple (binary classification) or complex (multi-class classification).
- Regression: implies predicting a real value for each item; the penalty for an incorrect prediction depends on the magnitude of the difference between the true and predicted values.
- Ranking: involves ordering items according to descending order of relevance.
- Clustering: involves partitioning items into homogeneous regions.
- Dimensionality reduction or manifold learning implies transforming an initial representation of data into a lower dimensional representation, while preserving some properties of the original representation.

Machine learning can be summarized as learning a function *f* that maps inputs variables *x* to output variables *y*, $y = f(x)$. The goal is to learn about the target by mapping the function from training data [37]. Based on how the function is learned (by making different assumptions), we identify two groups of algorithms: *parametric* and *nonparametric*. In the first group, the algorithm uses a well-known form, such as a line (linear regression), a sigmoid curve (logistic regression) or a Gaussian bell (linear discriminant analysis), to adjust its mapping function. Therefore, the learning model performs prediction with a set

of parameters of fixed size to define, for example, a probability density function described by two parameters (mean and standard deviation), independent of the number of training samples and making them faster to learn from data with a limited amount of training samples. A disadvantage is the poor fit due to dependency on choosing a functional form that matches with the target function. Some well-known parametric algorithms are logistic regression and linear discriminant analysis [52]. In the second group, the algorithm relies on data and no assumptions are required regarding the variable dependency. In other words, the algorithm is free to learn any functional form from the training data with some ability to generalize to unseen data [52]. However, it might be a drawback because the algorithm will require more training data to better estimate the target function and consequently slowing down the learning time. The best-known algorithms in this group are *k*-Nearest Neighbors, decision trees and neural netrworks. In addition, when working with machine learning two learning categories are used to deal with classification, regression and clustering type of problems: supervised and unsupervised. Table 3 exhibits the well-known algorithms applied to those problems.

Machine Learning		
Supervised learning		**Unsupervised learning**
Classification	**Regression**	**Clustering**
Support Vector machines	Linear Regression, General Linear Model	K-means, K-Medoids, Fuzzy C-means
Discriminant Analysis	Support Vector Regression, Gaussian Process Regression	Hierarchical
Naïve Bayes	Ensemble Methods	Gaussian Mixture
Nearest Neighbor	Decision Trees	Hidden Markov Model
Neural Networks	Neural Networks	Neural Networks

Table 3. Typical machine learning algorithms [39]

In our thesis, we capitalize on the property of nonparametric algorithms of not making assumptions about the underlying functions, it allows us to apply our framework in a wide spectrum of classification and regression problems. We used clustering techniques as described in section 5.3.

When the learner receives a limited series of labeled and unlabeled examples, semi-supervised learning can be used for training to make a prediction on unseen examples [6, 10, and 11]. Typically, a fraction of the labeled data (as labeled data is not always available in practical problems) is used during the training process that is combined with unlabeled data that is less expensive and takes less effort to acquire. Many scholars define this category as halfway between supervised and unsupervised learning. To apply this category of learning, the problems must meet three assumptions [53]: the smoothness (continuity) assumption: "If two points x_1, x_2 in a high-density region are close, then so should be the corresponding outputs y_1, y_2". The continuity assumption applies for both classification and regression problems. The cluster assumption: because of the previous assumption, in classification problems, "if two points x_1, x_2 are in the same cluster, then they are likely to be of the same class $y_1 = y_2$". The cluster assumption can also be formulated as follows: "The decision boundary should lie in a low-density region", this is known as low density separation assumption. The third assumption is named the manifold assumption: "High-dimensional data lie roughly on a low-dimensional manifold". By using this assumption, the learning algorithm can essentially operate in a space of corresponding dimension without having to pay the overload of computing processing. This assumption is useful for classification and regression. In general, the semi-supervised and supervised learning are both used in classification, regression and ranking problems [41].

Machine learning has applicability in text classifications, natural language processing, speech recognition, optical character recognition (OCR), computational biology application, computer vision tasks, fraud detection, games, unassisted vehicle control, medical diagnosis, remote sensing and information extraction systems.

3.2 Other learning methodologies

Aside from previous learning categories, we present other learning techniques that are not necessarily oriented to task-driven (supervised) or data-driven (unsupervised) architectures, albeit, they are powerful in a broad number of problems.

3.2.1 Iterative learning

One of the key aspects in iterative learning is the ability to extract training samples from previous instances and then use them to improve task performance in the next iteration. This implies updating a learning function with the best result, therefore improving the prediction model. There are many fields of application for iterative learning methodologies, for example in the control of robotic arms [125], when work is required to perform the same action repeatedly with high precision. An iterative process reduces the classification error and generates a rule of prediction that increasingly improves the learned function.

A technique that has influenced the landscape of machine learning since its inception in early 90s [54, 55] is boosting. In literature, we find various examples of boosting applications, especially in problems related to text recognition, control, data de-noising and model accuracy. To improve the efficiency of iterative learning, it is necessary to trade-off between the selection of the optimal set of parameters for the weak classifier and the form of the loss function (a method of evaluating how well the algorithm models the given data, e.g. mean squared error, likelihood loss, log, etc.) [56]. A weak classifier is a learning algorithm capable of producing classifiers with probability of error strictly (but only slightly) less than that of random guessing (0.5, in the binary case). On the other hand, a strong classifier is able (given enough training data) to yield classifiers with arbitrarily small error probability [57]. Since a loss function plays an important role in statistical inference, the relation between the loss function and the prediction performance is widely studied in statistics and machine learning communities.

In last decade some useful loss functions for classification problems have been proposed, for example, the *hinge loss* for support vector machines, and the *exponential loss* for Adaboost, among others [58]. In some cases, the boosting model can take advantage of using the weak learners at multiple resolutions. Two solutions are proposed by [59]: (1) using model-driven multi-resolution, achieved by varying the complexity of the classification boundary, providing a systematic procedure that increases the complexity of the weak learner as the boosting iterations progress, and thus reducing the over-fitting

problem; or (2) using a data-driven multi-resolution by considering the data (not the model) at multiple resolutions during each iteration in the boosting algorithm. The selection of the weak learners for the boosting algorithm can best fit the current resolution and as the additive modeling iterations progress, the modeling resolution is increased. A complete solution presented by [59] included the combination of AdaBoost (model-driven solution) and LogistBoost (data-driven solution). The AdaBoost algorithms build a hypothesis $\mathcal{H}$ that is a linear combination of weak or base hypothesis h_t [60], thus $\mathcal{H}$ can be of the form:

$$\mathcal{H}(x) = \text{sign}(\sum_t \alpha_t \, h_t(x)) \tag{3}$$

where α_t is a weight or confidence value $\in \mathcal{R}$. At the model level, iterative learning plays an important role helping to define fitting parameters and tuning hyper-parameters. The gradient descent algorithm uses an iterative learning methodology very useful to deal with the fitting parameter definition. The algorithm calculates the loss achieved by a model with a given set of parameters and then alters those parameters to reduce the loss. It repeats this process until that the loss cannot substantially be reduced further [35]. When tuning hyper-parameters, for example in SVM, two essential parameters must be defined, namely cost (C) and gamma (γ). The intention is to identify an optimal combination that minimizes the loss by evaluating iteratively the performance of the given hyper-parameter combination using cross-validation. Figure 15 shows a typical classification problem that is solved using AdaBoost.

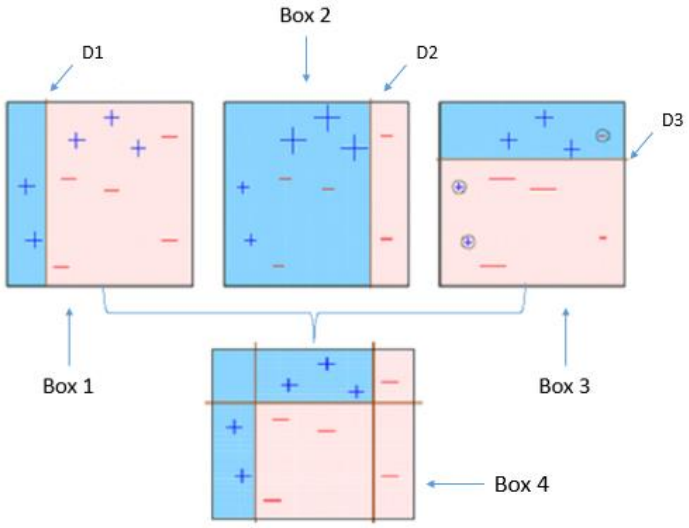


Figure 15. AdaBoost for classification problems [146]

In Box 1, data have been assigned equal weights and the first decision stump (D1) has been applied to classify them as plus (+) or minus (-). The incorrect samples will now carry more weight, in this case D1 has misclassified three (+). In Box 2, D2 has misclassified three (-). In Box 3, D3 has misclassified one (-) and two (+). The Box 4 uses previous individual weak learners D1, D2 and D3 to build a most accurate predictor. The model would continue adjusting the previous error obtained until building the most accurate predictor [146]. This type of algorithms is known to be very sensitive to outliers and noisy data and being disadvantageous in applications as those aimed in this proposal, i.e. remote sensing or recognition of locomotion using acceleration sensors. In our case, the prediction mechanism was optimized by hybridizing the data model adjustment with both correct and incorrect predictions from previous predictors.

3.2.2 On-line learning

Instead of learning from a training set and then testing on a test set, the on-line learning scenario intermixes both training and testing in multiple rounds. On each round, the classifier receives an unlabeled training point, makes predictions and verifies whether its prediction is correct or incorrect. The model is adjusted on-the-fly with correct predictions and then used to next rounds. The objective is to reduce the cumulative error (loss) over all rounds. Performance in on-line learning is measured using a mistake model and the notion

of regret. The on-line learning algorithms are particularly attractive in modern applications since they form a very interesting solution for large-scale problems. By processing one sample at a time, these algorithms are more practical than batch algorithms, because they take an initial guess model and then pick up one observation from the training population and recalibrate the weights on each input parameter, as presented in Figure 16 [61]

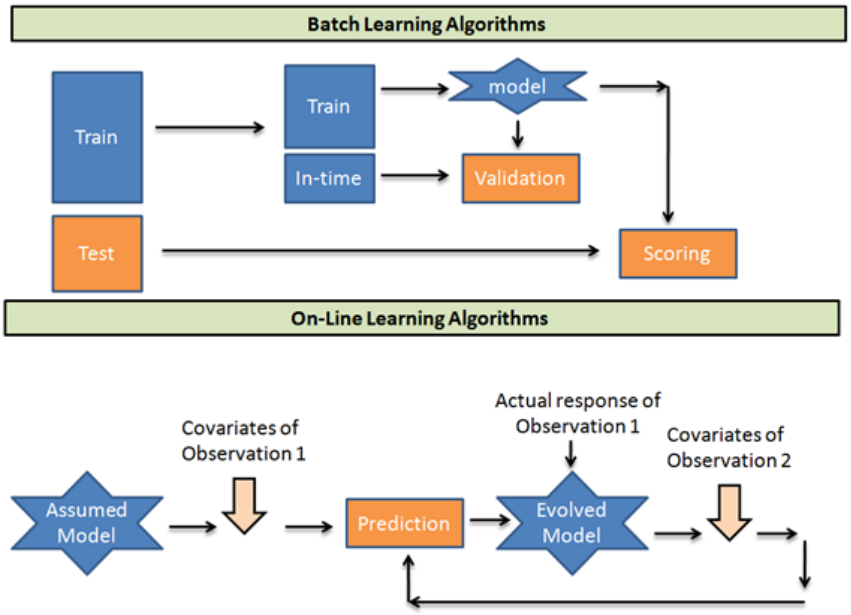


Figure 16. Batch learning algorithms vs On-line algorithms [61]

3.2.3 Reinforcement learning

The principle of reinforcement learning is to collect information through a course of actions by interacting with the environment. In response to an action, the agent or learner, receives two types of information: the current state in the environment and a real-value or reward, which is specific to the task and its corresponding goal. In this technique, there is no fixed distribution according to which instances are drawn. The choice of a policy defines the distribution, therefore, its choice is a very sensitive issue, since it will impact the rewards to be received. Reinforcement learning is widely connected to control theory, optimization and cognitive sciences. Figure 17 shows a diagram of the scenario of reinforcement learning.

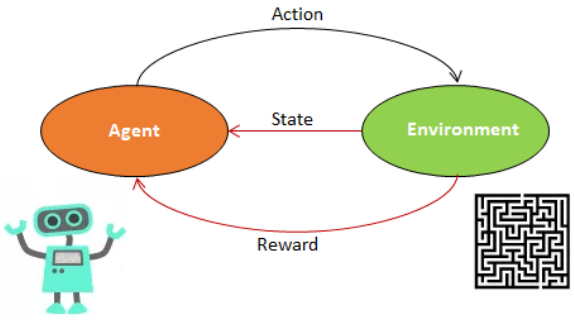


Figure 17. Reinforcement learning scenario [62]

The objective of the agent is to maximize his reward and therefore to determine the best course of action or policy to achieve such objective. However, the information he receives from the environment is only the immediate reward related to the action taken, therefore no future or long-term reward feedback is provided by the environment. An important aspect to be considered in this technique is the concept of delayed reward or penalties. The agent faces the dilemma of getting more information from the environment by exploring unknown states and the rewards or exploiting the information already collected to optimize his reward. This concept is known as exploring vs exploiting trade-off [63].

3.3 Model accuracy metrics

The most common metrics for model evaluation in regression and classification problems with machine learning are presented in Table 4.

Regression	Classification
Mean Absolute Error	Accuracy
Mean Squared Error	Precision, Recall
Root Mean Squared Error	F-score, AUC
Coefficient of determination	Receiver Operating Characteristic (ROC)

Table 4. Data model evaluation metrics

The regression metrics aim to evaluate and compare the real and the estimated values and determine the accuracy of the data model in prediction. Classification metrics evaluate the prediction performance of the algorithm. Let us review the most common regression metrics [79]:

Mean absolute error (MAE): measures the difference between two continuous variables and it can only be compared between models whose errors are measured in the same units. It is usually similar in magnitude to root mean squared error, but slightly smaller. Using the following notations: $\hat{y}_i$: predicted value, y_i: real value, n: dataset size, i {1,..,n}), the mean absolute error can be computed as:

$$MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n} \quad (4)$$

Mean squared error (MSE): assesses the quality of a predictor and is calculated as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \quad (5)$$

Root mean squared error (RMSE*)*: measures the error rate of a regression model. However, it can only be compared between models whose errors are measured in the same units. The formula to compute this error is as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}} \quad (6)$$

Coefficient of determination (R^2): summarizes the explanatory power of the regression model and is computed from the sums of squares terms as:

$$R^2 = \frac{\sum(\hat{y} - \bar{y})^2}{\sum(y - \bar{y})^2} = 1 - \frac{\sum(y - \hat{y})^2}{\sum(y - \bar{y})^2} \quad (7)$$

R^2 describes the proportion of variance of the dependent variable explained by the regression model.

The most common metrics to evaluate performance in classification problem (binary case) are accuracy, precision, recall, F-score and ROC. For a binary problem, a class instance can take only two possible values: positive or negative (0,1). The instances correctly predicted by the classifier are called true positive (TP) or true negatives (TN). Contrary, for those instances wrongly predicted by the classifier, the class instances are called false positive (FP) or false negative (FN), respectively.

Accuracy: is defined as the percentage of correct predictions that is determined by the ratio between the number of correct classification samples and the total number of samples. Accuracy (A_{cc}) is computed as:

$$A_{cc} = \frac{1}{n}[\sum_{i=1}^{n-1} 1(y_i = \hat{y}_i)] \times 100\% \quad (8)$$

where *n* is the number of samples, y_i and $\hat{y}_i$ are actual and predicted labels, and *i* an integer {1,..,n}. Due to its simplicity, it has restrictions to provide a thorough analysis of the algorithm behavior, for example, it is good when various classes in the input domain are nearly balanced. However, it would not be helpful to use it in problems with presence of a dominant class. For this reason, it is convenient to use the confusion matrix, which can contain a summary of prediction results with count values for true positive, true negatives, false positive and false negative [37]. The Acc can be defined in these terms as:

$$Acc = \frac{TP+TN}{TP+FP+FN+TN} \quad (9)$$

Precision: indicates how many samples where classified correctly among samples classified as positive. It is the ratio of true positive (TP) divided by the sum of the TP and false positives (FP).

$$\text{Precision} = \frac{TP}{TP+FP} \quad (10)$$

Recall or Sensitivity: It is the ratio between TP divided by the sum of TP and false negatives (FN). It is called sensitivity in binary classification.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{11}$$

F-Score: considers both recall and precision and it is a good way to evaluate how a classifier behaves. It is computed as:

$$\mathrm{F}_\beta = \frac{\text{Precision}*\text{recall}*(1+\beta^2)}{\text{Precision}+\text{recall}*\beta^2} \tag{12}$$

where β is a parameter that controls the importance given to the precision and recall. When equal importance is given to both metrics, then $\beta = 1$, therefore, $F_1 - \text{measure}$ is defined as:

$$\mathrm{F}_1 = \frac{2*\text{Precision}*\text{recall}}{\text{Precision}+\text{recall}} \tag{13}$$

Receiver operative characteristic (ROC): It is a tool used to evaluate discriminate effects among various methods. To plot the ROC curve, it is necessary first to obtain sensitivity and specificity values from data under consideration and normalize them into the same equal interval [80]. Specificity (SP) is defined as follows:

$$SP = \frac{TN}{FP+TN} \tag{14}$$

The ROC (probability curve) plots the TP rate (recall) against the FP rate, providing also a way to compare two classifiers with each other by measuring the area under the curve (AUC). Figure 18 shows the ROC AUC of a classifier. If the classifier is 100% correct, would have a ROC AUC of 1 represented by the curve plotted in blue, in this case two classes are distinguished. The red line represents a situation when the model does not have the capacity to distinguish between two classes.

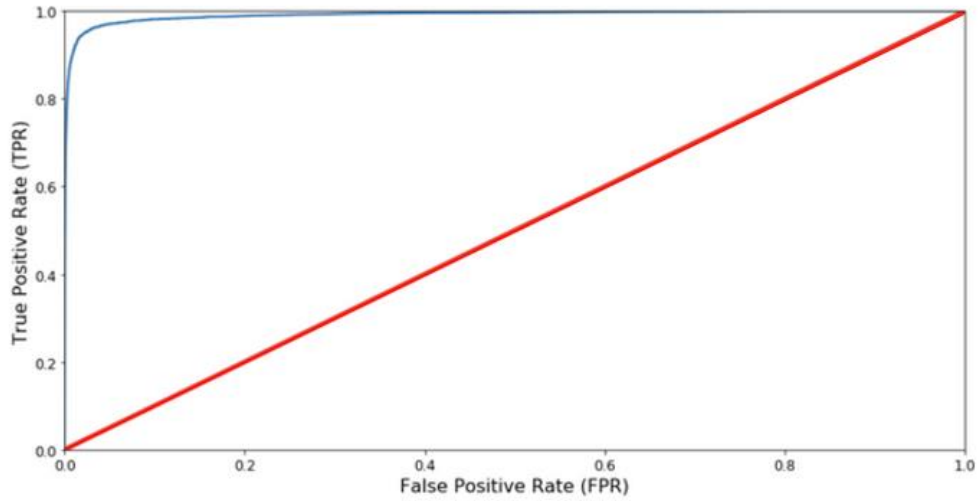


Figure 18. ROC AUC. A classifier that is 100% correct would have a ROC AUC of 1 [147]

The following section describe the use of machine learning techniques for the two scenarios that we are exploring in this project, namely remote sensing and wearable sensors.

3.4. Machine learning applications in remote sensing and wearable sensors

In the area of remote sensing, the applications of machine learning are diverse and include different domains such as trace gases, aerosol products, vegetation indices, ocean products, characterization of rock mass, liquefaction phenomenon, ground motion parameters and the interpretation of the remote sensing images [64]. Some examples are the estimation of the typhoon rainfall over ocean using multivariable meteorological satellite data [65], monitoring of water quality using remote sensing [66], mapping of base-metal deposits [67], image thresholding for landslide detection [68] and soil moisture distribution analysis [69]. Machine learning application in remote sensing dates back to the 90's with Huang and Jensen [70], [71], who built a knowledge-based data solution using minimal input from human experts and then created decision trees to infer the rules from the human input for the expert system. The generated rules were used at a study site on the Savannah River and the study demonstrated that results yielded the highest accuracy compared to conventional methods at that time [70]. In general, machine learning becomes an important tool to solve various problems in remote sensing like [72]:

a. Complexity in data/image fusion or merging for higher spatial and temporal resolution.
b. Feature extraction of different environmental quality images.
c. Cloud contamination, image reconstruction and cross-mission data merging
d. The design of integrated detection support systems.

Classification maps are the main product of remote sensing image processing. In the last years, data-driven approaches have gained relevance in the remote sensing community and non-parametric methods have demonstrated good performance [73].

There are several characteristics of geoscience applications that present a challenge and may limit the usefulness of traditional machine learning algorithms for knowledge discovery. They occur primarily in three situations [74]: first, there are some inherent challenges arising from the nature of geoscience processes. For example, objects that have amorphous boundaries in space and time, showing non-stationary highly multi-variate characteristics, and often involved in interesting but rare events. Second, data have multiple resolutions of space and time, and are impacted by varying degrees of noise, are incomplete, and present uncertainties. Third, in the case of supervised learning, the small sample size (e.g., small number of historical years with adequate records) becomes a challenge, as does the scarcity of standard ground truth in most geoscience applications.

These problems are discussed in more detail in Chapter 6. Authors in [74] report three major categories of challenges for applying machine learning techniques in geoscience and remote sensing: challenges inherent to the geoscience process, challenges related to data collection and challenges related to the paucity of samples and ground truth information. These are summarized in Table 5.

A second type of application involving machine learning in this thesis is related to the recognition of human locomotion by processing information recorded with wireless wearable sensors. Body activity recognition using the wearable sensor technology has drawn more and more attention over the past few decades. The complexity and variety of

body activities makes it difficult to fast, accurately and automatically recognize body activities [75].

Challenge type	Constraint
Geoscience process	Objects with amorphous boundaries Spatiotemporal structure High dimensionality Heterogeneity in space and time Interest in rare phenomena
Data acquisition	Multi-resolution data Noise, incompleteness and uncertainty
Paucity of samples and ground truth	Small sample size Paucity and ground truth information

Table 5. Machine learning challenges in geoscience and remote sensing [74]

The application of machine learning techniques over data obtained via wearable sensors applications is described in [54]. The authors reported on human activity recognition systems based on supervised learning approaches, with overall accuracy between 84% and 97.5%, in applications related to exercise analysis and monitoring of patients with heart disease, diabetes and obesity, with data gathered on a daily or weekly basis. The authors also reported applications based on semi-supervised learning techniques with an overall accuracy up to 96.5%. Some of these results were obtained by using a training dataset containing 2.5% of the total amount of data and employing multi-graph algorithms and support vector machines (SVM) combined with multiple eigen-spaces. This approach is close to our approach, since we also make use of eigenvalues (scores) produced by principal component analysis (PCA). Other learning techniques, like decision trees, Bayesian and neural networks, fuzzy logic, Markov models and boosting [76] have also shown significant potential in wearable sensing, especially when dealing with problems like segmentation (determined by the variability and the periodicity produced by human activity) and classification [54, 77]. In human activity recognition, data collection with

varieties of sensors is preceded by other data analytics phases such as pre-processing, data segmentation, extraction of salient and discriminative features, and finally classification of activity details [78]. Although the research on activity recognition is beneficial from the perspective of wireless sensors' unobtrusiveness and deployment flexibility, it also faces major challenges [77], as presented in Table 6.

Tables 5 and 6 exhibit the major challenges for machine learning applications using spatiaotemporal information registered by remote and acceleration sensors. In general, in this type of problems, the accuracy of data models will strongly depend on the selection and application of an adequate data preparation process that mitigates the negative effects produced by the constraints presented in Tables 5 and 6.

Challenge type	Constraints
Data acquisition process	Noise, incompleteness and uncertainty Industry manufacturing standards Feature extraction Activity signal pre-processing
Inherent to phenomenon	Motion during transition period between two activities. Insufficient training set Model training Location and orientation of the wearable sensor
User dependent model	Subject sensitivity

Table 6. Machine learning challenges in wearable wireless sensors [77, 78]

The complex problem of recognizing human activity has motivated different groups of researchers to benchmark different real-world, multi-mode, non-stationary scenarios with wearable sensing solutions. As mentioned in Chapter 2, machine learning provides an excellent approach to improve model accuracy, based on data structures that might dynamically change, while dealing with complex and large datasets acquired from a particular environment [12]. One critical problem found in the design of data solutions for

recognizing human locomotion is the limitations (e.g. noise, jitter, interference, etc) introduced in the data acquisition process and their repercussions during the selection of the training dataset [54]. Figure 19 shows the taxonomy of human activity recognition (HAR) systems discussed by [54]. Challenges and other details were discussed in section 2.5

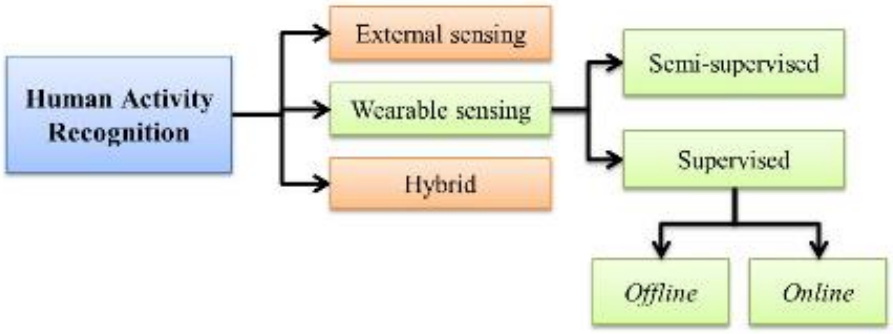


Figure 19. Taxonomy of HAR system [54]

3.5 Conclusions

In this chapter, we described the learning process and the types of challenges faced when using different machine learning methodologies in the design of multimodal systems. We found the use of three types of learning frameworks: CRISP-DM, SEMMA and general learning process, which have been deployed to improve the process of problem analysis, mitigating the problem of having random or false discoveries that bias the expected results. We also discussed classical and also other machine learning methodologies such as iterative learning, on-line learning and reinforcement learning, placing special emphasis on the iterative learning. Finally, we presented the most relevant challenges found in machine learning when modeling data solutions in geoscience and remote sensing as well in the recognition of human locomotion by using wearable wireless sensors.

CHAPTER IV: OBJECTIVES AND CONTRIBUTIONS

This work aims to develop a machine learning approach to automate the process of modeling multimodal systems, with application in both regression and classification problems. To demonstrate the validity of the proposed iterative learning framework, we solve two problems:

a. The estimation of chlorophyll using data extracted from remote sensing platforms.
b. The recognition of human locomotion activities using data collected using wireless wearable sensors.

In this context, this thesis has a general objective and four specific objectives as presented below.

4.1 Research objectives

4.1.1 General objective

The ultimate objective of this research is to develop an automated process for modeling multimodal systems using an iterative machine learning approach. We aim at building an iterative process that progressively adjusts the previous error found by the learner until obtaining the most accurate data model on each iteration. The objective of our research work further extends to the verification of the developed process by solving engineering tasks that are representative of the range of problems that can be successfully addressed by the proposed approach. The first problem that will be investigated is the issue of building analytical data models for the estimation of chlorophyll concentration –a challenging task due to the complex interaction of biophysical variables affecting the accuracy of the model, and the elaborated sensor data processing procedures. The second problem investigates the problem of building a data model in a non-stationary environment, when human locomotion recognition is done by using acceleration sensors readings. Two major issues are addressed in this problem, the difficulty to deal with motion during transition period between two activities and the presence of noise that alters the resulting readings.

4.1.2 Specific objectives

a) Integration of a mechanism based on a multimodal hypothesis to assess the occurrence of multimodality.

Our intention is to integrate in our approach a mechanism based on multimodal hypothesis, which determines the occurrence of multimodality, instead of reducing the problem to a unimodal hypothesis such as proposed in [93, 94, and 95]. In order to reach this goal, we propose to make use of the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) to weight contributions provided by each modality.

b) Development of regression models spanning the entire input domain

We aim to improve the precision of the model by developing an iterative process that combines the regression analysis with data classification in order to effectively deal with multi-resolution data spanning the whole input domain. These type of technical problems are presented in [74].

c) Enhancement of the level of robustness to variations in the quality of input data

The data acquisition process plays an important role when extracting training samples and when the variations in the quality of the acquired data have a negative impact on the feature extraction process and the resulting data model. We thus aim at mitigating the influence of the noise present during the process of data acquisition by using an iterative learning process that extracts only the best training samples from previous instances and then use them to improve the task performance in the next iteration.

d) Validation of the proposed approach when modeling multimodal systems in regression problems

We propose the application of our approach in solving the problem of the assessment of chl-a concentration using in-situ measurements in Lake Winnipeg in Manitoba, Canada and optical datasets collected for MODIS and MERIS in a series of lake surveys carried out in the years 2002–2004.

e) Validation of the proposed approach when modeling multimodal systems in classification problems using temporal data.

We propose the application of our approach in solving the problem of classifying human locomotion activities, such as walk, stand, lie and sit using readings acquired from body–worn sensors available in the open source dataset Opportunity [6].

4.2 Contributions

The proposed modeling framework covers a wide spectrum of problems, primarily regression and classification, and improves the accuracy of the data model regardless the size and quality of the dataset [74, 77, 78]. The proposed approach reduces significantly the training size and the time needed to build accurate data models.

The proposed data-driven architecture combines unsupervised (regression and clustering) with supervised iterative learning to identify, through an iterative process, the selection of the best candidate training samples. Indeed, introducing a sample selection mechanism improves the model accuracy and brings a two-fold benefit: reduction of the training process time, and minimization of the problem of overfitting and of the complexity of the classifier.

The proposed methodology can be adapted to practically any classification or regression problem. In such case, changes in the pre-processing phase can be easily implemented by using any of the data mining solution discussed in Chapter 3, such as CRISP- DM or SEMMA.

One of the most promising applications of our framework are multiple areas of environment monitoring, management and control, thanks to the reduction of the operation costs required when collecting *in-situ* samples and the shortening of the intervals between data gathering processes whilst speeding up the analysis of the collected information with a quasi-real time response.

Our framework has also a potential for extensive application in human locomotion recognition and particularly in monitoring the elderly with limited range of motion and the athletes to follow up on their performance. This is because our framework is a user-dependent data model, thus becoming a valid option for clinical treatment, where diagnostics are customized according to user's needs.

CHAPTER V: METHODOLOGY

This chapter presents an overview of the proposed methodology, which has been developed in order to reach the thesis objectives. The methodology, in its core, consists in using an iterative classification process that extracts successively the best training candidates belonging to each mode, classifies the given dataset into binary classes and selects new, expanded sets of labeled data. The models generated for each class and their joint error are compared with the error of the previous set of models and the model with the lowest misclassification error is designated as the resulting data model.

In general, our methodology focuses on generating datasets associated with each statistical modality of the given dataset, spanning the entire instance space for each modality. The core component in our methodology is therefore the training dataset extraction process, which ensures a high level of robustness to variations in the quality of input data and consequently leads to an improvement in the data model accuracy. We can distinguish four building blocks of our methodology:

- The assessment of the use of the multimodal hypothesis in building a precise model from the application dataset. To validate this part, we use Akaike information criterion (AIC) and the Bayesian information criterion (BIC).
- The mechanism to select candidates from the input data, used later in building a training dataset, which can optimize the learner's prediction process.
- The iterative classification process, which successively classifies the data and selects new, expanded sets of labeled data.
- The model selection with the lowest error or misclassification rate.

The general architecture, which includes the multimodal system modeling methodology, is presented in Figure 20 and described in sections 5.1 through 5.4.

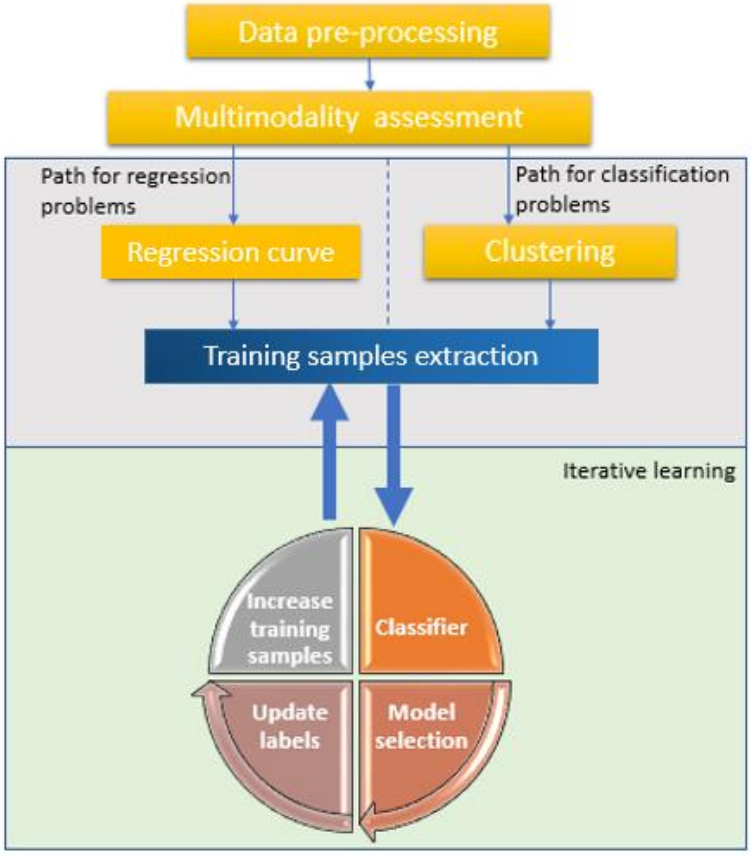


Figure 20. Block diagram for proposed framework

Overall, four core processes can be distinguished in our general modeling framework: the data pre-processing process is required to prepare raw data by removing noise, outliers and spurious values, etc. The multimodality assessment process allows us to validate the occurrence of multimodality in the given dataset. Depending on the problem type (linear regression or classification), we use an appropriate partitioning mechanism that extracts the initial training dataset (made up of the best candidates). This phase is called the training samples extraction process. The iterative learning process consists in classifying data successively into binary classes, using a portion of the training dataset in order to generate a hyperplane that defines a separation curve used to determine the re-labeled data further for estimation. In each iteration, the resulting data model is stored, and its labeled data used into the next iteration. The process ends when all training samples are completely used by the iterative learning phase. The model selection focuses on getting the lowest

modeling or misclassification error produced by each of the resulting models found in each iteration.

The implementation of this general framework follows the CRISP-DM standard presented in Chapter III, section 3.1. In our model, the business understanding phase corresponds to the multimodality assessment process, the data preparation phase is equivalent to the training dataset extraction process, and the evaluation and modeling phases with the proposed iterative learning process.

5.1. Data pre-processing

This initial process includes data cleaning, data formatting, and problem-specific data transformations. The data pre-processing step is also required to select relevant features. In order to deal with the problem of high dimensionality, techniques such as PCA and singular value decomposition (SVD) are applied in this phase. In problems related to multi-sensor analysis of dynamic systems, additional steps such as timestamping in a consistent manner, resampling, filtering and de-noising of the raw data are required to enhance the precision of the resulting data models.

5.2 Multimodality assessment

The multimodality assessment process determines the number of modes existing in the given data distribution. We use Akaike information criterion (AIC) and the Bayesian information criterion (BIC) to detect the optimal number of modes. The process starts by creating a Gaussian mixture model, and continues by increasing the number of modes until getting the minimum value of AIC and/or BIC; the minimum value determines the quality of the best model. In our work, BIC was used to agree on the results obtained through AIC. That is because BIC penalizes the model complexity more severely than AIC.

The Akaike information criterion is a tool used to measure the model quality based on the maximized likelihood estimate of a data model. The idea of AIC is that a chosen model is correct if it can sufficiently describe any future data with the same distribution. In our work, AIC was used to compare model performance produced by statistic modalities. The

value of AIC for a given model is a measure of the loss of information which results from the use of the model to explain a particular variable or pattern [149]. The AIC is defined as follows:

$$AIC = -2(loglikelihood) + 2K \tag{15}$$

where *K* is the number of estimated parameters included in the model and log-likelihood of the given data. The lowest AIC value will indicate the best model among all models specified for the dataset [149]. In our work, AIC was used to determine the quality of each model over Gaussian distributions produced by each statistical modality (i.e., bi-modal in the case of our first experiment as shown in Chapter VI), therefore, the most accurate model will have the smallest AIC value. We used an AIC score defined as follows [150]:

$$AIC = \log \mathfrak{f} + \frac{2\mathrm{k}}{N} \tag{16}$$

where $\mathfrak{f}$ is the loss function, K is the number of estimated parameters, and N is the number of values in the estimation dataset. The loss function is given by [150]:

$$\mathfrak{f} = det\left[\frac{1}{N}\sum_1^N \varepsilon((t,\theta_N))(((t,\theta_N)))^T\right] \tag{17}$$

where N is the number of values in the estimation dataset, $\varepsilon((t,\theta_N)$ represent the prediction error given θ_N estimated parameters. A more robust criterion but not necessarily better than AIC is the Bayesian information criterion (BIC). In general terms, BIC uses the same principle of the optimal loglikelihood function value. However, it includes a penalty function that depends on the sample size. BIC is formalized as [151]:

$$BIC = -2(logLikelihood) + K * \log(N) \tag{18}$$

where *K* is the number of estimated parameters and *N* is the number of samples. The lowest BIC score is produced by the most accurate model. In equation (18) the term $K * \log(N)$ is known as the penalty term that grows with the number of samples.

5.3 Training samples extraction: the initial partition

Once the number of modes in the data set is determined, our process continues with the initial partition process, which extracts the training samples to be used during the training process. For regression problems, the initial partition consists in generating a regression curve (linear, exponential, and polynomial) from the given data, and then extracting the samples with the largest Euclidean distances that are measured between each data sample of the given data and the regression curve, as depicted in Figure 21. This strategy is called policy layer, where the largest distances are defined as those resulting distances that are larger than mean plus the standard deviation of all the Euclidian distances on the given data. Figure 21 illustrates a bimodal system with two classes. Samples coloured in yellow are labelled as Class 1, dots in blue are labelled as Class 2. Dots in red represent the regression curve.

The same principle is used for classification problems. However, the *policy layer* is determined by the distance between each sample in the input data and its cluster's centroid instead of using a regression curve. Figure 22 exhibits the initial partition for classification problems. The number of centroids is defined according to the number of modes determined in the modality assessment process (see Chapter VI, section 6.5 and Chapter VII, section 7.6)

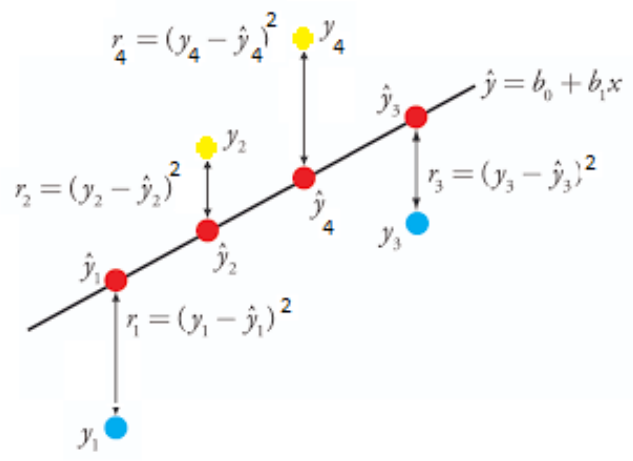


Figure 21. Initial partition using linear regression for a bimodal data distribution. Yellow dots belong to Class 1 and blue dots belong to Class 2. The regression curve is represented by red dots.

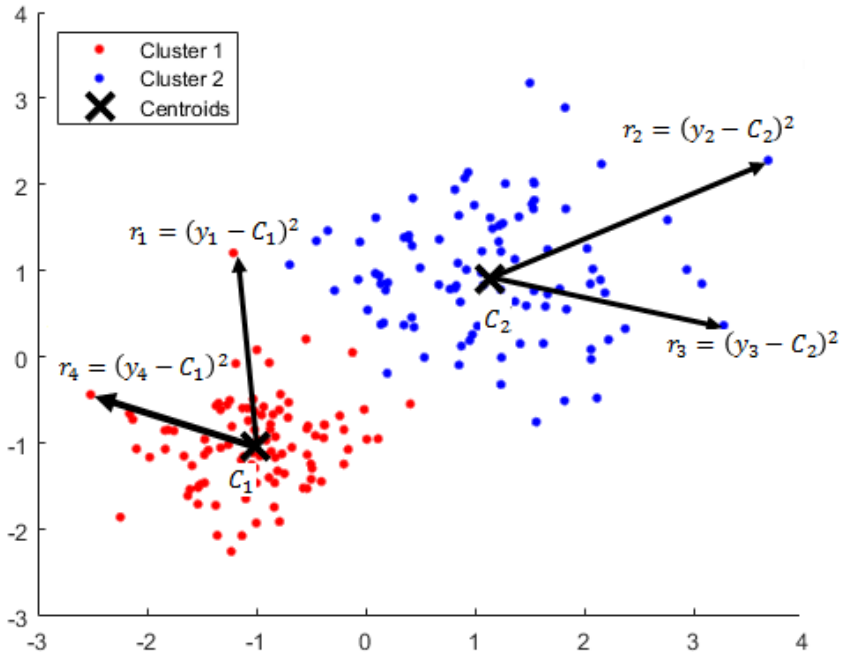


Figure 22. Initial partition using centroids. Red dots belong to cluster 1 and blue dots to cluster 2.

5.4 Iterative learning process

The aim is to build a process that classifies data successively into binary classes and selects new, expanded sets of labelled data. This process can be described as follows: The learner, a support vector machine algorithm (SVM), receives a fraction of the training samples found in the initial partition (5.3). The size of this training subset depends on the size of the training dataset and the number of desired iterations:

$$\text{Trainig subset size} = training\ dataset\ size * threshold \tag{19}$$

where threshold is $\frac{1}{\#\ of\ iteration}$, a parameter used to control the number of samples in each iteration (see Chapter 6, Figure 30). The number of iterations, noted as *m* is defined in section Chapter VI, section 6.9. The training subset is used to select the best combination of hyper-parameters such as regularization parameter (*C*), tolerance (ε) and kernel parameters (e.g. γ) via cross-validation. The resulting model is used to predict over the unseen data. This process is repeated, by increasing the size of the training subset, until reaching the size of the training dataset of the initial partition.

By comparing the data model generated by a current iteration and its joint error with the error of the previous set of models, we take advantage of the concept of the maximum margin in SVM. In this case, only those class members with single class membership can define a better line of separation in reference to those members where class membership is difficult to determine, for example in regions with high data density.

The important feature of SVM, as opposed to probabilistic type classifiers, is that the discriminative power of the classifier is defined by a set of support vectors, i.e., samples located close to the separating hyperplane, and not on the parameters of unknown distributions of multi-class data. This strategy corresponds to the maximum margin principle.

A two-class SVM can be defined as follows: let $\overrightarrow{x_i} \in R^d$ be a multidimensional empirical input vector, and $y_i \in \{-1,1\}$ the label of the class which is assigned to each input vector.

The problem consists in assigning a label to each vector according to its class, -1 or +1. The space R^d is split into two regions by a hyperplane

$$\overrightarrow{x_i}.\vec{w} + b = 0 \tag{20}$$

where $\vec{w}$ is normal to the hyperplane and b is a constant. If x_+ is a sample labeled as 1 and x_- is labeled as -1, the widest margin that produces the optimal separation of positive and negative examples is defined as finding the maximum of:

$$(\overrightarrow{x_+} - \overrightarrow{x_-})\left(\frac{\vec{w}}{||\vec{w}||}\right) = \frac{2}{||\vec{w}||} \tag{21}$$

subject to $y_i(\overrightarrow{x_i}.\vec{w} + b) - 1 = 0\ \forall i$. Lagrange multipliers and the Wolfe theorem are used to solve the previous problem [122]. The optimal margin L_d is obtained from:

$$L_{\mathrm{d}} = \frac{1}{2}||\vec{w}||^2 - \sum_{i=1}^{l} \alpha_i(\mathrm{y_i}(\overrightarrow{\mathrm{x_i}}.\vec{\mathrm{w}} + \mathrm{b}) - 1) \tag{22}$$

where:

$$\vec{w} = \sum_{i=1}^{l} \alpha_i \mathrm{y_i}\,\overrightarrow{\mathrm{x_i}} \text{ and } \sum_{i=1}^{l} \alpha_i y_i = 0,\ \alpha_i > 0 \tag{23}$$

It can be noticed that the optimal margin depends on dot products produced by all samples in the input space. This solution works fine when samples in the input space are linearly separable. The problem of linear separability is dealt with by applying the "kernel trick" [123]. By introducing kernel functions, a non-linear problem defined in the input space can be transformed into a linear problem in a high-dimensional feature space. Consequently, by operating in the kernel space

$$k\left(\overrightarrow{\mathrm{x_i}}, \overrightarrow{\mathrm{x_j}}\right) = \varphi(\overrightarrow{\mathrm{x_i}}).\varphi\left(\overrightarrow{\mathrm{x_j}}\right) \tag{24}$$

the optimal margin will be reduced to:

$$L_d = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j \mathrm{y_i y_j} k\left(\overrightarrow{\mathrm{x_i}}, \overrightarrow{\mathrm{x_j}}\right) \tag{25}$$

The margin analysis is a vital part in the modeling framework because by fixing the decision rule in each iteration it determines the way samples are classified as in Figure 23. In each iteration the margin is adjusted according to the labels that are obtained in the previous iteration. In the first iteration, they are obtained from the initial partition. The semi supervised learning strategy consists, therefore, in finding a subset of best candidate samples, which are most distant to the partitioning curve and produce a maximally large margin. While iterating, the learning process produces new, larger sets of labeled data, a new separation line with a narrower margin, and an updated data separation between the two classes.

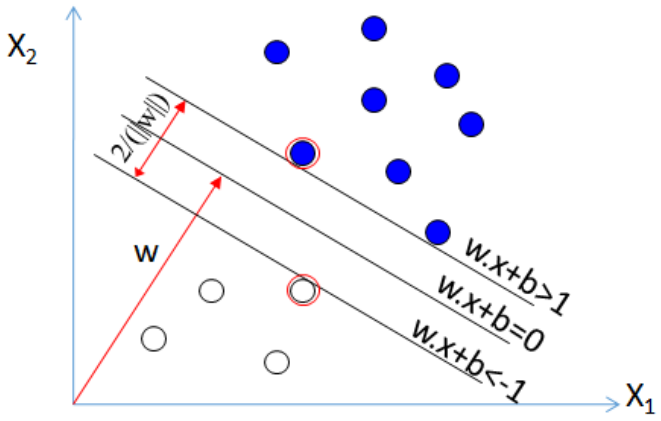


Figure 23 Margin L_d is adjusted on each iteration

This approach is shown in Figure 24 for an iterative process applied to a bimodal data distribution. Lines in red illustrates the separation plan generated by a linear SVM classifier. Note that the model is progressively adjusted by increasing the number of training samples as found in section 5.3. These samples (circled in orange) are the ones with the largest Euclidean distances identified during the initial partition. The model with the lowest misclassification rate is the winner model.

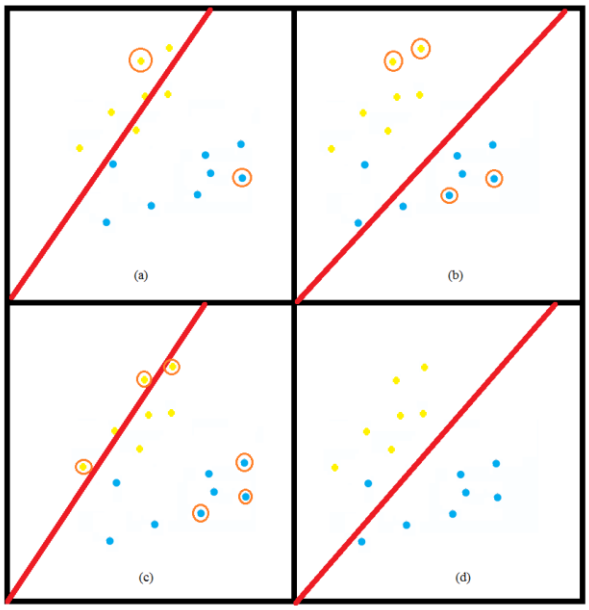


Figure 24. Iterative learning process in three iterations. Red line defines the separation hyperplan generated by training candidates circled in orange (a,b,c). The best model is shown in (d)

The general pseudo-code is shown in Figure 25. The *SelectClassifierParameters* procedure, displayed in Figure 25, takes place after performing the partitioning procedure and uses a uniform selection grid to find the best combination of parameters. These parameters can be extracted from a *k*-fold cross validation procedure, with *k*=5, and thus using four subsets for training and one subset for testing. This strategy allowed us to set a range of values for *C* and γ organized in a grid of values to enable the selection the pair of parameters which has the lowest training error. In our case, we use $\mathrm{C} = (2^{-5},..,2^{7})$ and $\gamma = (2^{-5},..,2^{7})$.

Input: Var_{n-1}, Var_n ; ***Output***: *Estimation model*

Method:

Pre-process data source
Normalize-Variables (Var_{n-1}, Var_n) *// Values [0,1] or [1,-1]*
TestMultimodality(Var_{n-1}, Var_n)
N= Create Partitions
// Based on linear regression, polynomial, exponential partitioning. Clustering partitioning for classification problems
ReducetoBinomialClass //If problem is related to multi-class classification
For 1=1: N
    InitialLabel(Class1,...,Class2)
    ECM=ExtractCandidates (Class1,Class2)// Extract candidate members
    SelectClassifierParameters (ECM)//
    TrainClassifier // Using best candidates (k-fold cross-validation)
    TestClassifier
    DetermineFinalLabel(MatrixOfLabels)
    SaveModel(N)
End for
SelectBestModel(SaveModel)

Figure 25. General pseudo-code for the proposed framework

5.5 Conclusions

Our proposed mechanism of selection, based on the largest Euclidian distance, measures the distance between each sample and a given partition curve (in regression) or cluster's centroids (in classification), locating the samples with a unique class- membership regions, such as shown in Figure 23. This concept combined with the SVM maximum margin definition, is applied to build a decision function that is fully specified by the training dataset, and able to identify those samples that "matter" in defining the separating line and consequently improving the prediction mechanism.

It is worth mentioning that we take advantage of hybridizing in our approach both unsupervised and supervised learning methods using key learning concepts from each of them: learning how to extract the structure from the given data (unsupervised), and learning

to predict outputs from the input data (supervised). By extracting a most coherent training dataset, we also optimize the time required for training - a serious disadvantage for non-parametric algorithms because of the need for a significant number of samples during the training phase. More implementation details are presented in Chapters VI and VII.

CHAPTER VI: SOLVING A REGRESSION PROBLEM

In this chapter we present the application of our modeling framework in the problem of estimating chlorophyll concentration in large aquatic areas using remote sensing technologies, a problem that has challenged researchers for years due to the complex interactions of biophysical variables and their direct impact on the accuracy of the model.

Obtaining precise models in environment monitoring is important. The fact that we are obtaining data from different remote sensing (RS) sources, often from different data acquisition missions and different time periods, makes the environment modeling process especially prone to statistical multimodality. A wide range of chlorophyll concentration values and different types of waters with contrasting optical properties, combined with the interaction of multiple components in the optical data flow make this environment monitoring problem especially difficult from the standpoint of developing precise and robust regression models.

6.1 Monitoring of chlorophyll concentration in large aquatic areas

Chlorophyll is a color pigment, the molecule of which are used as photoreceptors in the process of photosynthesis. Chlorophyll type a (chl-a) is one of six different chlorophylls, and it is the primary molecule responsible for photosynthesis. It is found in plants, algae, and oxygenic photosynthetic organisms (phytoplankton) that sustains all terrestrial life by producing oxygen.

From the standpoint of the quality of inland waters, chl-a is frequently used as the indicator of the ecological health of aquatic environments. Nutrient enrichment, occurring in many water basins surrounded by the agriculture land, causes water eutrophication by reducing the oxygen levels and increasing the concentration of organic matter [96]. Therefore, chl-

a concentration can act as a water eutrophication indicator [97], tracking phytoplankton biomass [98].

Precise levels of chl-a concentration can be estimated with in-situ methods, such as fluorometry and chromatography, or using remote sensing, which uses different techniques to retrieve the chl-a concentration from selected spectral bands, for example red and near infra-red regions of the light spectrum. Remote sensing technologies are widely used for monitoring water quality over large aquatic basins due to the cost and flexibility advantages as compared with in-situ methods. However, they require the use of sophisticated data processing techniques.

Full global coverage for ocean color products was first provided by the low-resolution Sea-viewing Wide Field-of-view Sensor (SeaWiFS). Two medium-resolution sensors have been widely used for determining chl-a concentration in in-land waters: The Moderate Resolution Imaging Spectroradiometer (MODIS), launched in 2002 on the Aqua satellite, and the Medium Resolution Imaging Spectrometer (MERIS), launched in 2002 on the ENVISAT platform. Recent improvements applied to the MODIS products are discussed in [99].

The level of concentration of chl-a is determined by the amount of phytoplankton biomass that produces distinct changes in water color by absorbing and scattering the incident light [4]. In the open-ocean Case-1 water, chl-a concentration can be derived from the blue and green spectral bands. Case 1 is that of a high concentration of phytoplankton compared to other particles. In contrast, a suspension of nonliving material with a zero concentration of pigments is called the Case 2 water [148]. Overlapping absorptions by dissolved organic matter limit the utility of the blue spectral region. Therefore, in turbid productive water, spectral algorithms which are based on the reflectance in the red and the near-infrared (NIR) spectral regions are often preferable [100].

An advantage of the MERIS sensor is the availability of a spectral channel at 708 nm, commonly used to detect fluorescence wavelength peaks, which are emitted by the

phytoplankton's pigment at that band [101]. MODIS has significant advantages in the number of bands in the IR portion. However, the absence of a spectral channel at 708 nm makes it less reliable for applications in turbid and productive inland waters [102]. The study reported in [103] advocates the use of some improved approaches, such as vicarious cross-calibration method, to improve the poor noise value of MODIS shortwave IR (SWIR) bands [103]. An improved SWIR iterative algorithm for MODIS data when applied to monitor the water quality in Lake Taihu was presented in [104].

6.2 Modeling chl-a estimation using remote sensing techniques

To estimate chl-a concentration from optical satellite imagery, indices derived from the shape of the spectral characteristics are used. Two examples of empirical models based upon reflectance band ratios are the ocean chlorophyll 2 algorithm (OC2v4) and the ocean chlorophyll 4 algorithm (OC4v4), which are frequently used with reflectance data collected by SeaWiFS [105]. The OC2v4 algorithm uses two bands (490, 555 nm), located in the blue and green portion of the light spectrum. The chl-a concentration is predicted from the band ratio $R_s(490)/R_s(555)$ by plugging the obtained value into a fourth order polynomial function. In a similar manner, OC4v4 uses four bands (443, 490, 510, and 555 nm) [106]. The chl-a concentration is assessed by determining the maximum band radio (MBR) produced by $R_s(443)/R_s(555)$, $R_s(490)/R_s(555)$ or $R_s(510)/R_s(555)$ [107].

The utility of reflectance data collected in the red (610 nm - 680 nm) and the near-infrared (NIR) spectral regions (790 nm – 890 nm) was demonstrated in [108, 109]. The chlorophyll concentration can be estimated by NIR-Red parametric models [4], such as the two and three band models are:

$$Chl(\lambda) = f\left(\frac{R_{\lambda_3}}{(R_{\lambda_1})}\right) \tag{26}$$

$$Chl(\lambda) = f\left(R_{\lambda_3}\left(\frac{1}{R_{\lambda_1}} - \frac{1}{R_{\lambda_2}}\right)\right) \tag{27}$$

Two algorithms directly related to the chl-a concentration phenomenon have proven their usefulness: the maximum chlorophyll index (MCI) and the fluorescent line height (FLH).

The spectral model based on MCI or FHL associates the chlorophyll concentration with the height of the reflectance peak produced in the wavelength where the fluorescence is emitted by the phytoplankton's pigment [110] and at nearby bands, where the fluorescence phenomena is reduced or absent. These peaks are detected in bands 709 nm (MCI-MERIS) [111] and 673 nm (FLH-MODIS) [112]. In general, MCI and FHL, also called line height algorithms, are based on the reflectance spectral response that is retrieved by a linear interpolation of two baseline bands, and can be expressed as follows [113]:

$$Chl(\lambda) = R_{(\lambda)} - R_{(\lambda^-)} - \left\{R_{(\lambda^+)} - R_{(\lambda^-)}\right\} \times \left[\frac{\lambda-\lambda^-}{\lambda^+-\lambda^-}\right] \quad (28)$$

where $R_{(\lambda)}$ is the value of the reflectance in a central wavelength (λ) band, and λ^+ and λ^- are the neighbor bands preceding and succeeding the central wavelength band. MCI indicates the presence of chl-a against a scattering background by correlating the height of the peak at the 709 nm band with a linear baseline defined by radiances at the wavelengths of 681 nm and 753 nm [114]:

$$MCI = R_s(709) - R_s(681) - \left[\frac{(709-681)}{(753-681)}(R_s(753) - R_s(681))\right] \quad (29)$$

Though MCI applies primarily to the MERIS satellite sensor, a spectral response based on FLH can be obtained by computing MERIS bands at 680.5 nm, 664 nm and 708 nm [115]. In the case of MODIS, FLH can be determined by using bands 13, 14 and 15, and can be calculated as follows [116, 117]:

$$FLH = R_s(678) - R_s(748) + \left[\frac{748-678}{748-667}\right]\left(R_s(667) - R_s(748)\right) \quad (30)$$

The precision of measuring the chl-a concentration depends not only on the selection of the appropriate index, but also on such factors as water turbidity, depth, and temperature. In deep ocean waters, the optical properties are more directly affected by phytoplankton and the observed spectral response can be more easily related to the concentration of chl-a [118]. When observations are done over inland waters (Case-2 waters), the observed spectral response can be markedly affected by other water constituents, such as the

concentration of total suspended solids (TSS), dissolved organic carbon (DOC) and particulate organic carbon (POC). Their concentrations do not necessarily co-vary with the chl-a concentration. Thus, retrieving water constituent absorption coefficients in turbid and hypereutrophic waters from remote sensing reflectance data is a challenging issue [119].

Apart from the problem of optical characteristics of the water, there are some important practical restrictions intervening in the process of building precise data-driven models. The nonstationary character of environmental phenomena, the often longtime of data acquisition missions, biases introduced by pre-processing procedures, and the use of different types of sensors can generate datasets with multimodal statistical distributions. Data multimodality can markedly degrade the precision of empirical models. Another general problem in verifying environmental models, in this case, water quality models, is the scarcity and high cost of obtaining the ground-truth information for different water types and conditions.

6.3 Data pre-processing

Our analysis was performed on two optical datasets that were collected in Lake Winnipeg, Manitoba, Canada, from locations in the zone UTM easting 487130 - 684992 and UTM northing 5598980 - 5965192. Lake Winnipeg, covering an area of 24,514 square kilometers, is diverse both geographically and in terms of optical characteristics of its water. The South Basin and the east shore of the North Basin are turbid regions, with high concentrations of suspended solids. Widespread plankton blooms have developed mostly in the remaining parts of the North Basin. High levels of dissolved organic carbon concentrations (DOC) occur near the mouths of tributary rivers, especially those draining from the agriculture region to the east of the lake. *In situ* chl-a measurements were acquired in a series of lake surveys carried out in the years 2002–2004 during the months of June and August, using the equipment on board of MV Namao operated by the Lake Winnipeg Research Consortium.

The chlorophyll biomass was estimated by Fluoroprobe (manufacturer: bbe-Moldaenke, Germany) at 5-minute intervals. The chl-a concentration was analytically determined by

using a high-performance liquid chromatography (HPLC) method [120]. The spectral measurements were performed by using an ASD FieldSpec spectrometer, with the capacity to record radiance in 1.4 nm -wide bands from 330-1050 nm at 1 second intervals. Water samples were taken using a van Doorn sample bottle roughly 0.1-0.3 m below the surface of the lake at intervals of approximately 6-7 km [121]. Collected optical data were sampled to the wavelengths corresponding to MERIS and MODIS satellites. The resulting data were grouped in two sets of 148 measurements each: the first one in a range of 16 MODIS bands (412 nm to 940 nm) and the second one in a range of 15 MERIS bands (412.5 nm to 900 nm). The chl-a concentration varies from a minimum of 0.4 mg/m^3 to a maximum of 133.6 mg/m^3. We processed the collected optical information using a transformation process that takes the optical radiance indices from the light bands where chl-a is mainly detected, converting them into light reflectance band ratios, in our case using two parametrical models: the maximum chlorophyll index (MCI) and the fluorescent line height (FLH). Raw data is plotted in Figures 26a and 26b; and data transformation is shown in Figures 27a and 27b. Figure 26 also exhibits reflectance values collected for MODIS and MERIS.

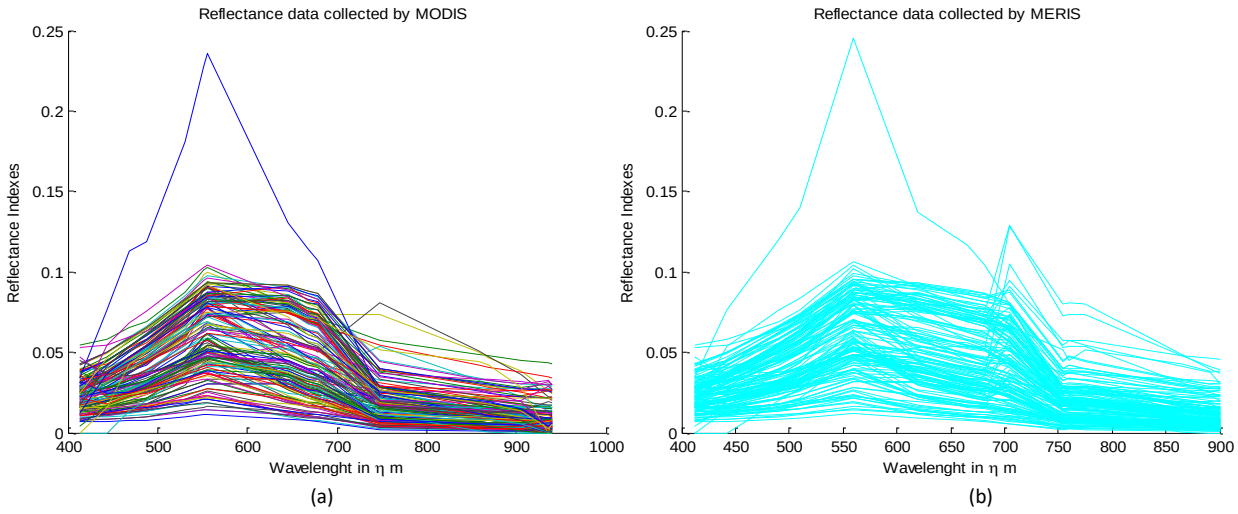


Figure 26. Data transformation using a) FHL for MODIS and b) MCI MERIS

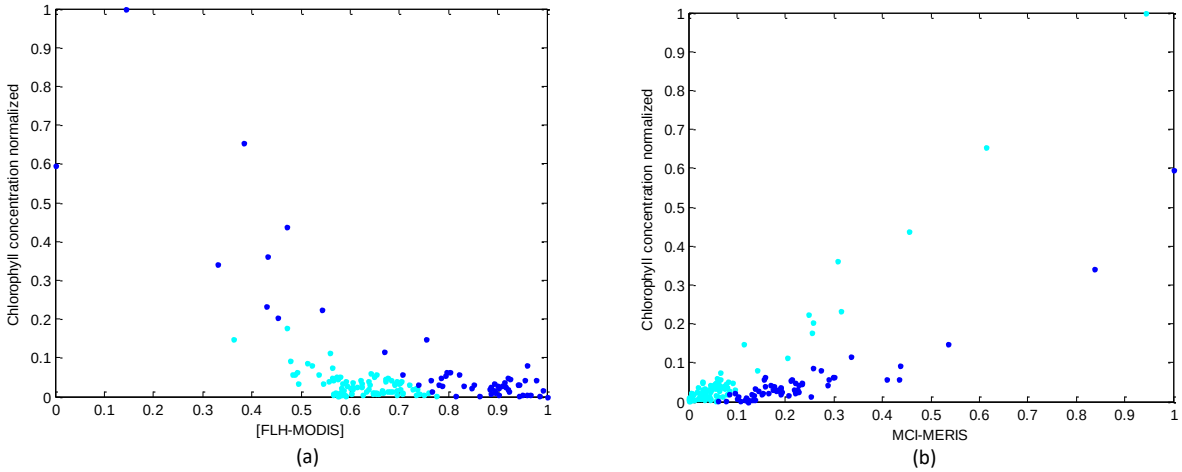


Figure 27. Reflectance indexes collected in Lake Winnipeg. (a) Spectral shape response for MODIS. (b) Spectral shape response for MERIS

6.4 Multimodality assessment

Once the pre-processing process has taken place, we identified a bimodal distribution as it is observed in Figure 27. Points colored in magenta and blue represent two different modalities found in the data distribution. This distribution is produced when MCI and FLH algorithms are applied to the input data domain. In this problem, we are interested in deploying a mechanism that determines the occurrence of multimodality based on statistical analysis. The analysis of the data carried out in the next paragraphs addresses the multimodality issue in more detail.

Assuming a Gaussian distribution for each dataset, we built Gaussian mixture models (GMM), aiming to obtain a generative probabilistic model discribing the distribution of the data (see Figure 28), instead of hyper-spherical clusters with the same radius. In other words, we do not need to standardise the input variables. In order to assess the level of modality that would produce the best model, the dataset was tested against two criteria, the Akaike information criterion and the Bayesian information criterion as described in Chapter 5, section 5.2. Since both AIC and BIC are likelihood maximization criteria, the model with a lower AIC/BIC score value is preferred, because negative contributions from the likelihood are greater than positive contributions from the parameters. These results of the modality analysis are presented in Table 7.

	MODIS		MERIS	
Modality	**AIC**	**BIC**	**AIC**	**BIC**
Unimodal	-364,74	-349,75	-467,88	-452,89
Bimodal	-728,71	-695,74	-874,85	-841,89
Trimodal	-762,84	-711,88	-950,08	-899,13
Quadmodal	-791,12	-722,18	-960,61	-891,68

Table 7. AIC and BIC score

Table 7 shows resulting scores reported for MODIS and MERIS when applying AIC and BIC. From the table we observe that a bimodal assumption enhances the goodness on each data distribution (MODIS and MERIS) with respect of the unimodal distribution. According to the Akaike theory, the most accurate model will have the smallest AIC score. Like AIC, BIC uses the optimal loglikelihood function value and penalizes more complex models [155], hence the most accurate model will have the smallest BIC score value. The best results are shown in Table 7 (see values in red).

Technically, the best scores are found in the Quad-modal distribution. However, as shown in Figure 28, a few samples are located in and outside of the region circled in blue. Particularly for MODIS (Figure 28.a) and MERIS (Figure 28.b) we found just, two and three samples respectively. Comparing these populations with the amount of samples significantly higher, in the regions circled in red and magenta, we conclude that the quadmodal distribution assumption does not describe the whole input domain precisely enough. In this context, the samples located in the blue regions are not necessarily considered as those of different water types. Indeed, the difference between bimodal, trimodal and quadmodal distributions (28,41) is marginal comparing with difference between unimodality and bimodality models (363,97). This analysis, therefore, helps us to confirm the presence of two dominant classes along the data input domain, such as shown by the number of samples in the regions within the red and magenta circles and observed in the Figure 28.

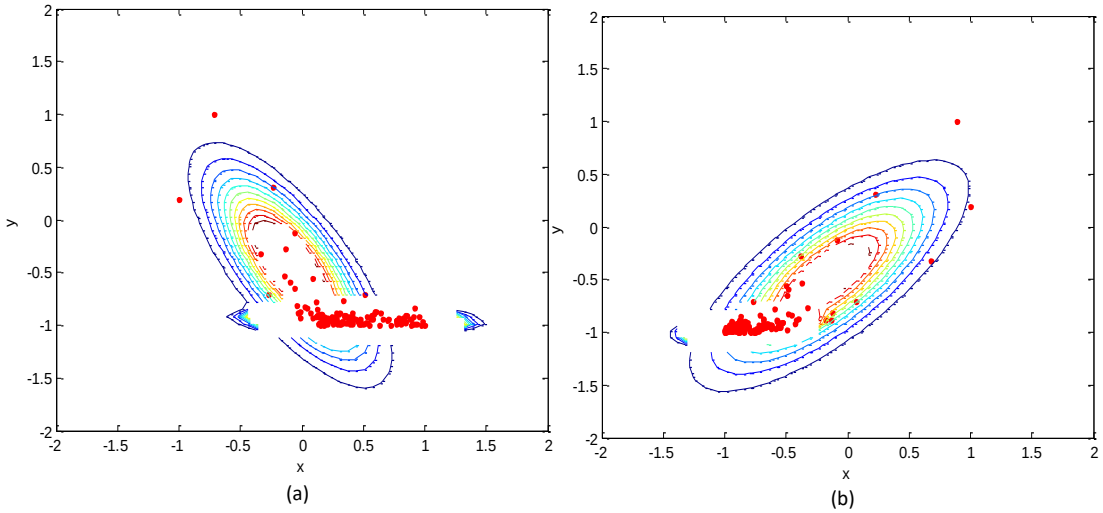


Figure 28. Probability density function (pdf) for Gaussian mixture distributions, for normalized values: a) Chl-a (y) vs FLH (x), and b) Chl-a (y) vs MCI (x).

6.5 Training samples extraction

As explained in Chapter V, section 5.3, the automated modeling process receives two input datasets denoted as follows: the first one $\mathcal{X} \in \mathbb{R}^n$ is a multidimensional vector of reflectance indices, which are produced by parametric algorithms like MCI and FLH. The second input denoted as $\mathcal{Y} \in \mathbb{R}$ is a unidimensional vector of empirical values of chl-a concentrations distributed according to a distribution $\mathcal{D}$ over $\mathcal{X}$ defined by f: $\mathcal{X} \rightarrow \mathcal{Y}$. Once modality level has been assessed, in this case a bimodal distribution, the process continues with the initial partition of the dataset.

The initial set of labeled data consists of the best candidates, which are the instances most distant from an initial partition curve, in this case the regression curve produced by input variables $\mathcal{X}, \mathcal{Y}$. This partitioning process is explained as follows.

The proposed partition mechanism extracts the datasets used for building regression models by producing a separation curve that is further used to determine the re-labeled data. Since our objective is not to obtain piece-wise models, it is important to span the separation line along the whole input domain. Three partitioning mechanisms are analyzed: linear, polynomial and exponential. The resulting regression curves determine the initial

labels. For simplicity, samples situated above the curve are labeled as "1" (mode 1), while those below the curve as "0" (mode 2).

Figure 29 shows the initial separation of the MERIS dataset as produced by each partitioning scheme. At the same time, the initial partition curve serves as an initial unimodal model, which the final model results are compared with. Table 8 summarizes the coefficients of determination (R^2) obtained at the stage of the initial partition. The same analysis can be extended to the MODIS dataset.

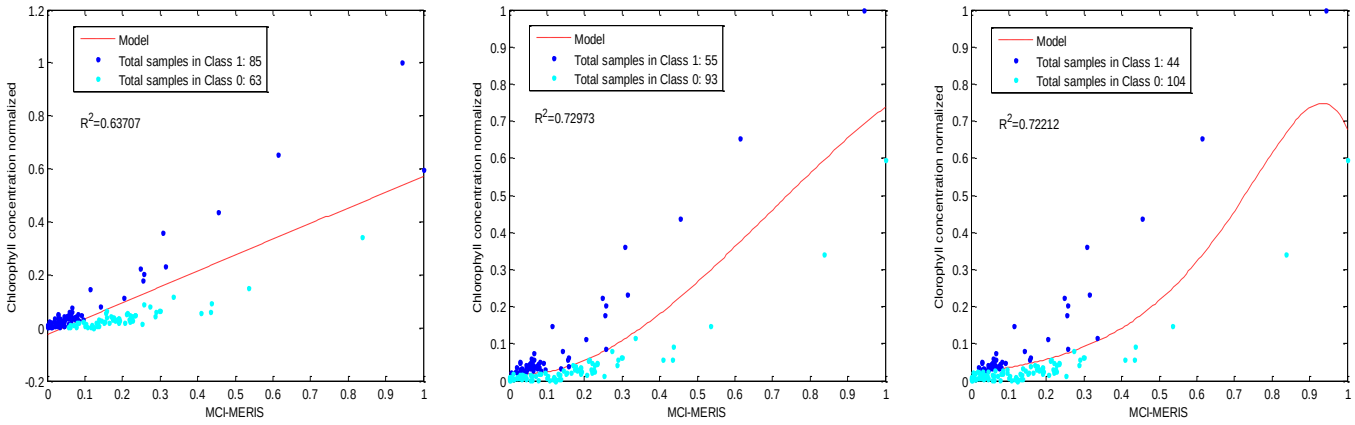


Figure 29. Initial partitioning for MERIS based on regression analysis a) linear, b) polynomial and c) exponential.

PM mechanism	Index	# Samples in Mode 1	# Samples in Mode 2	R^2
Linear	FHL-MODIS	56	92	0.322
Polynomial	FHL-MODIS	70	78	0.657
Exponential	FHL-MODIS	62	86	0.612
Linear	MCI-MERIS	85	63	0.637
Polynomial	MCI-MERIS	55	93	0.730
Exponential	MCI-MERIS	60	88	0.719

Table 8. Initial data partitioning for MERIS and MODIS

6.6 Iterative learning process

As presented in Chapter V, section 5.4, once the initial labels are defined, the process continues with an iterative classification routine. Typical unsupervised learning methods, such as k-means algorithms, cannot be used in this case, since they limit the resulting clusters to some segments of the input domain. Supervised learning strategies require reference samples. Therefore, in absence of a reference dataset, we are confronted with the problem of defining a suitable set of samples. Our approach provides a solution with two core features. First, those samples that are most distant to the partitioning curve are assumed to have the greatest probability to belong to the class as determined by the curve, and second, we adopt a support vector machine (SVM) classifier, a nonparametric algorithm, as the partitioning mechanism operating at the subsequent stages of the learning process. The reasons for this selection are: the robustness against the outliers, controlled by cost *C* (The algorithm is especially effective in problems where a number of dimensions is high) and the flexibility of parameter adjustments (In our case, we use both linear and non-linear partitioning mechanisms).

6.7 The training set selection: the policy layer

Let $R_s\,(i)$ be the resulting unidimensional residual vector that is independently produced by each mode. $R_s\,(i)$ is given by:

$$R_s\,(i) = \left| f\big(x(i)\big) - \hat{y}(i) \right| ; \qquad i = [1, \dots, 148] \in \mathcal{N} \tag{31}$$

where $f\big(x(i)\big)$ is the value produced by the initial partition, and $\hat{y}(i)$ is the value produced by the current partition curve. Let f_s be a ranking function, based on the mean $\overline{R_s}\,(i)$ and the standard deviation $\sigma(R_s\,(i))$ independently applied to each mode, and defined as follows:

$$f_s\big(R_s\ (i)\big) = \begin{cases} S_1(i):\ R_s\ (i) \geq \overline{R_s}\ (i) + \sigma \\ S_2(i):\ \overline{R_s}\ (i) \leq R_s\ (i) < \overline{R_s}\ (i) + \sigma \\ S_3(i):\ \overline{R_s}\ (i) - \sigma \leq R_s\ (i) < \overline{R_s}\ (i) \\ S_4(i):\ R_s\ (i) < \overline{R_s}\ (i) - \sigma \end{cases} \tag{32}$$

The datasets $V_{s1,Class1}\ (i)$ and $V_{s1,Class2}\ (i)$ contain the items with the largest Euclidian distance from the partition curve. These data are considered as the best candidates and are included in the training dataset. Thus, three components are associated: the distance, the sample $f\big(x(i)\big)$ and its label $y = \{1,0\}$. Since datasets $V_{s1,Class1}$ and $V_{s1,Class2}$ may be imbalanced [124], we used a strategy that includes a mechanism to control the number of training candidates assigned to each class. The size of the training dataset depends on the minimum number of members in $V_{s1,Class1}$ and $V_{s1,Class2}$:

$$TrainingL_{y=\{1,0\}} = \min\{length\big(V_{s1,Class1}\big), length\big(V_{s1,Class2}\big)\} \tag{33}$$

Equation (33) produces the same number of samples per class in the training set. The same strategy is applied to the second training set, using $V_{s2,Class1}(i)$ and $V_{s2,Class2}\ (i)$ in the iterative classification routine. With each iteration we increment the training dataset by a portion of the extended training set. During each round, the algorithm calculates the resulting coefficient of determination $R^2_{1,2}$ defined as follows:

$$R^2_{1,2} = \left[\frac{\frac{\sum_{i=1}^{m}(y_{1,i}-y^*_{1,i})^2}{m} + \frac{\sum_{i=m+1}^{n}(y_{2,i}-y^*_{2,i})^2}{(n-m)}}{\frac{\sum_{i=1}^{n}(y_i-\overline{y_i})^2}{n}}\right] \tag{34}$$

where m is the number of members in the first class, $(n-m)$ is the number of members in the second class, and n is the number of input data. The resulting labels and model parameters are stored for determining the final labeling.

6.8 Model selection

The final classification results are used to obtain the empirical models. The quality of the models is assessed according to the value of its correlation coefficients. Once the iterative learning process is completed, the algorithm builds a *matrix of labels* with those models that satisfy the condition $R_{1,2}^2 \geq R^2$. Each row contains the labels assigned to each sample during the iteration process. The next task is to determine the most probable label for each sample. A probabilistic function *LabelValue* weighs each row in the matrix of labels, scoring the 1s and 0s:

$$LabelValue(j)_{l=1} = \frac{\sum_{i=1}^{n}[x(i)=1]}{n} = \alpha_j, j \in [1,..,m] \quad (35)$$

$$LabelValue(j)_{l=0} = 1 - \alpha_j = \beta_j, j \in [1,..,m] \quad (36)$$

where n is the number of labels in each row, m is the number of rows, j is the jth sample, and i is the ith column. Once the scoring process is done, the label matrix is reduced to an $m \times 2$ matrix

$$\begin{bmatrix} \alpha_1 & \beta_1 \\ : & : \\ \alpha_m & \beta_m \end{bmatrix} \quad (37)$$

The values produced in (35) and (36) determine the label that is assigned to a particular sample. The label assignation process is performed based on a linear range that is defined by three approximately equal segments. The first segment [0-0.35], positively negative, is used to label samples as "0". The second segment [0.35-0.65] defines an uncertainty region. The last segment [0.65-1], positively positive, is used to label samples as "1". In the case of four rounds of classification, the task of labeling the samples in the uncertainty segment is limited practically to breaking the parity. In our experiments, the samples with an even number of "0"s and "1"s were labeled according to the results of the classification performed using the Initial Partitioning curve. By adopting an uncertainty segment labeling scheme, a consistent approach is offered also for a higher number of classification rounds. The complete algorithm is presented in Figure 30.

Input:
MultibandIndices matrix (X), In_situ chl-a vector (Y)

Output:
Chl-a estimation model

Method:

```
SelectIndex= (if 1 then MCI, 0 then FHL)
ChlaFeature=ExtractFeature(MultibandIndices, SelectSensor, SelectIndex)
[x,y]=LoadFeature(ChlaFeature , In_situ Chl-a)
[xnorm, ynorm]=Normalize(x,y)
Bimodal = testBimodality(xnorm,ynorm)
    if Bimodal then
      for j=1:n Partition
        [Class1,Class2,R^2]=PerformPartitionClass(xnorm,ynorm, j)
        TrainingSamples=ExtractCandidates [Class1,Class2]
        YLabelsInit=InitialLabel(xnorm, ynorm)
          m=(TrainingL_{y={1,0}}) * threshold
          while m<= σ(R_s (i))
            MatrixOfLabels=IterSVM(YLabelsInit,TrainingSamples,m,R^2)
            DetermineFinalLabel(MatrixOfLabels);
            SaveModel(j)
            m=m+threshold*σ(R_s (i))
          end while
      end for
      selectBestModel(SaveModel)
    else
        "Bimodality is not present in the problem."
    end
```

Figure 30. Partitioning algorithm

6.9 Model performance evaluation

The model performance is measured using the root mean squared error and the coefficient of determination R^2. To demonstrate the performance of the algorithm, we compared our results with those obtained by applying classical linear and non-linear regression methods.

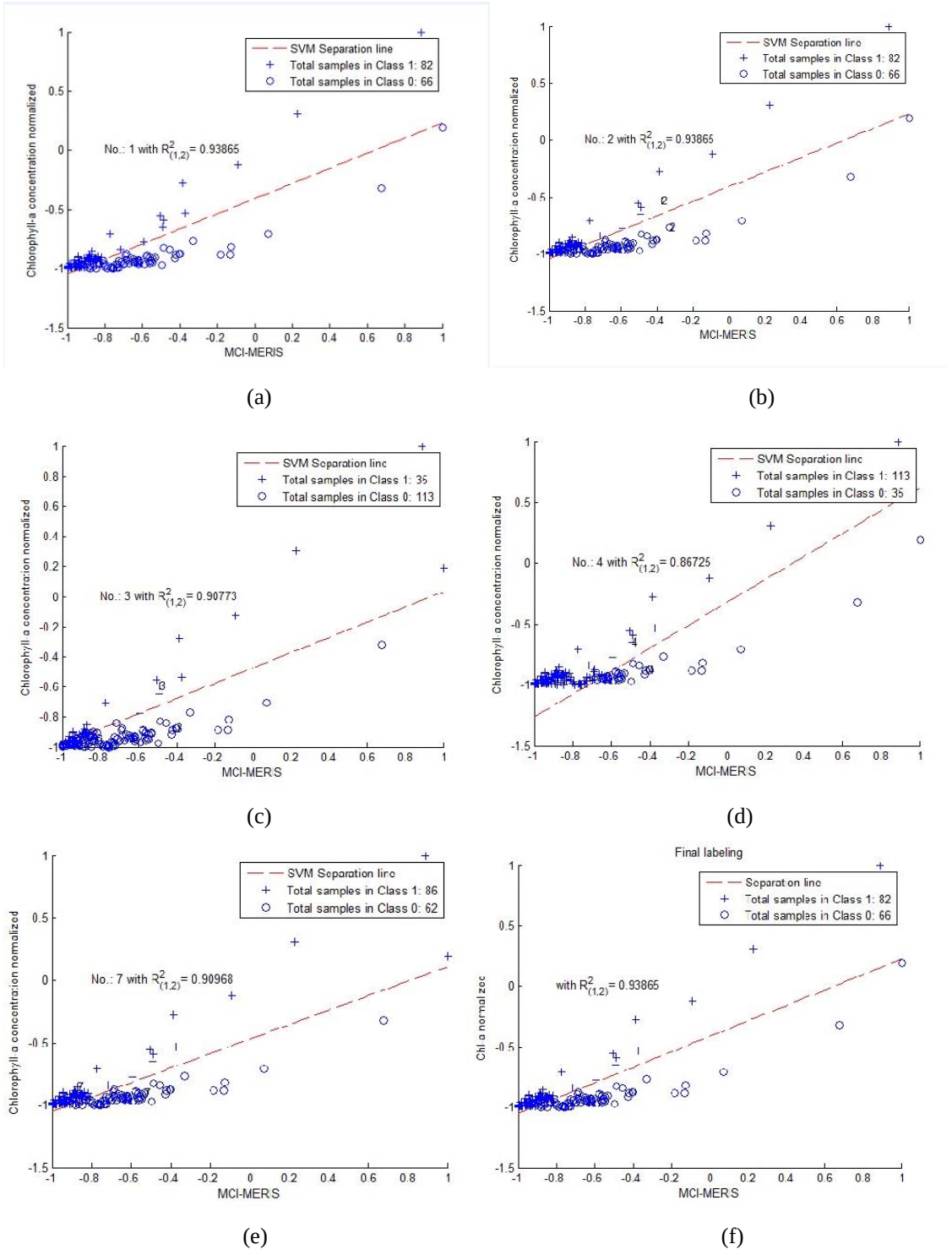


Figure 31. A sequence of 7 iterations using a linear partition presented in Figure 29. The resulting training sets are used on each iteration (a,b,c,d and e) to determine the best model (f).

Figure 31 exhibits normalized chl-a in-situ values versus the maximum chlorophyll index (MCI) obtained for the MERIS data. The samples above the regression curve are considered members for class '1'and labelled as plus '+', otherwise class '0' and labelled as 'o'. During the initial partitioning, we found the coefficient of correlation ($R^2 = 0{,}32$) obtained when using linear regression, see Chapter VI, section 6.5, Table 8. Next, we select the training set based on measuring the Euclidean distance between each sample and the regression line (policy layer). Next, we use the resulting training samples to evaluate the data model on each iteration as shown in Figure 31 a,b,c,d,e. Finally, using the resulting matrix of weights, we calculate the probability of the label each sample must have and generate the final model as in Figure 31.f. The iterations 5 and 6 were arbitrary omitted in order to allow the reader to better trace the process and to enhance the legibility of the content of the Figure 31.

6.10 Experimental results

The experiments were performed on the dataset described in section 6.4 . The whole solution was implemented in MatLab 2016 and the SVM classifier was integrated from the LibSVM library, version 3.20 [152]. In order to estimate the chl-a concentration, three experiments were performed, based on the three partitioning mechanisms discussed in section 6.5: linear regression, non-linear polynomial and exponential regression. In an additional experiment, the impact of different kernels on the performance of the proposed partitioning mechanisms was investigated.

Our framework was applied over MODIS and MERIS datasets independently. Table 9 shows the resulting values when applying linear regression as a partitioning mechanism. We present two modes and the number of samples associated to each modality. The number of iterations are fixed as $m=(TrainingL_{y=\{1,2\}}) * threshold$, where m is the number of training samples, $TrainingL_{y=\{1,2\}}$, defined in equation (32) and a $threshold$, as defined in (15). Through trial and error, we concluded that using a threshold of $0.25\ \sigma$ provided us the best control over the number of iteration the algorithm can run. In this case we have four iterations until reaching the condition $m <= \sigma(R_s\ (i))$, where $R_s\ (i)$ is defined

in equation (31). Based on the resulting training size, the $threshold$ must be updated, such as in Chapter VII, section 7.5, where threshold was identified as 0.125 σ. On each iteration, we select a new, expanded sets of labeled data, producing a coefficient of determination $\boldsymbol{R_{1,2}^2}$ defined in equation (34), which measures the performance of the model. In the Table 9, the "Final labeling" row displays the resulting data model after applying the model section mechanism described in section 6.8.

MCI-MERIS				FLH-MODIS		
Number of iterations	**Mode I**	**Mode II**	$\mathbf{R_{1,2}^2}$	**Mode 1**	**Mode II**	$\mathbf{R_{1,2}^2}$
1	136	12	0.7468	50	98	0.7944
2	136	12	0.7468	55	93	0.7769
3	91	57	0.9061	50	98	0.8006
4	90	58	0.9072	50	98	0.8006
Final labelling	**91**	**57**	**0.9061**	**51**	**97**	**0.7974**

Table 9. Results obtained for a linear regression partitioning

We can observe that using traditional linear regression, the coefficient of determination calculated for MERIS-MCI was 0.637 and 0.32 for MERIS-FLH (see Table 8). Comparing with values obtained with our method, the model obtained from the iterative learning process provides a significant improvement, especially in the case of FLH-MODIS.

In the second experiment, we used a partitioning mechanism based on a non-linear regression with a cubic polynomial function (x, x^2, x^3). The results are shown in Table 10.

MCI-MERIS				FLH-MODIS		
Number of iterations	**Mode I**	**Mode II**	$\mathbf{R^2_{1,2}}$	**Mode I**	**Mode II**	$\mathbf{R^2_{1,2}}$
1	126	22	0.9391	54	94	0.9002
2	126	22	0.9391	54	94	0.9002
3	126	22	0.9391	31	117	0.9111
4	126	22	0.9391	31	117	0.9111
Final labelling	**121**	**27**	**0.945**	**32**	**116**	**0.9111**

Table 10. Results obtained when applying a non-linear regression partition (cubic polynomial function)

We notice an important improvement on the data model performance produced when implementing our method. For MCI-MERIS, the coefficient of determination was improved from 0.73 to 0.945, while for FLH-MODIS it was improved from 0.68 to 0.91.

In the third experiment, when the exponential $(\alpha e^{\beta x})$ nonlinear regression was applied as the partitioning mechanism, the results are shown in Table 11.

MCI-MERIS				FLH-MODIS		
Number of iterations	**Mode I**	**Mode 2**	$\mathbf{R^2_{1,2}}$	**Mode 1**	**Mode 2**	$\mathbf{R^2_{1,2}}$
1	138	10	0.8656	87	61	0.8872
2	135	13	0.8754	89	59	0.8777
3	102	46	0.9487	89	59	0.8777
4	115	33	0.9183	89	59	0.8777
Final labelling	**96**	**52**	**0.9621**	**72**	**76**	**0.8903**

Table 11. Results obtained when applying non-linear regression partition (exponential function)

As shown in Table 11, the values reported for both FLH-MODIS and MCI-MERIS are better than those obtained via non-linear regression using exponential functions. When assessing FLH-MODIS, we obtained an improvement of 35.01% on the accuracy of the model. Likewise, an improvement of 17% was obtained on the resulting model for the MCI-MERIS data. In our three experiments, the obtained results have demonstrated the advantage of applying nonlinear partitioning mechanisms. A marked increase in the model

performance can be observed especially for MODIS data. It is important to note that we do not perform the experiment on MCI-MODIS since MCI bands are not available in MODIS.

The selection of the type of the kernel is an issue that should be considered in the SVM classification. In our fourth experiment, the impact of selecting different kernel functions on model precision was assessed. Apart from the linear kernel, the kernel functions listed in Table 12 were used to analyze the classifier's performance. In this experiment, we tested the following additional kernels: Radial Basis Function (RBF), sigmoid and polynomial.

Kernel	Function
Linear	$k(x_i, y_j) = x_i^T y_i$
Polynomial	$k(x_i, y_j) = (\gamma x_i^T y_i + r)^d, \gamma > 0$
RBF	$k(x_i, y_j) = e^{-\gamma \left\| x_i - y_j \right\|^2 d}, \gamma > 0$
Sigmoid	$k(x_i, y_j) = \tanh(\gamma x_i^T y_i + r)$

Table 12. Kernel functions

Table 13 shows the implementation of our method with the SVM classifier using different kernels and different partitioning methods.

Kernel	Partition mechanism (PM)		
	$R^2_{1,2}$ Linear	$R^2_{1,2}$ Polynomial	$R^2_{1,2}$Exponential
FHL-MODIS			
Linear	0.7974	0.8073	0.8647
RBF	0.4970	0.8822	0.8954
Sigmoid	0.7959	0.8410	0.8938
Polynomial	0.7937	**0.9111**	**0.9035**
MCI-MERIS			
Linear	0.9361	0.9556	0.9203
RBF	0.9065	0.9548	0.9183
Sigmoid	0.9369	**0.9619**	**0.9621**
Polynomial	0.8571	0.945	0.8864

Table 13. Results of our learning process using four kernels with three different partitioning methods

We observed that the kernel behavior is different in both dataset distributions. For FHL-MODIS, the polynomial kernel produced the highest values; however, the sigmoid kernel was even better in the case of MCI-MERIS. The improvement in the model performance was reached when we changed the input data feeding strategy of the classifier. In our initial experiments, the SVM classifier was fed with the same number of positive and negative examples, with the focus on alleviating the problem of the dominant class and the resulting biased data model.

With the objective to improve the model generalization capability, we modified our initial feeding strategy by considering the class member imbalance, giving us some level of freedom in the data set to vary independently. This resulted in slightly different results, but also in more robust models. It should be noted that the issue of imbalanced training sets has been extensively researched in the area of machine learning, for example in [154]. The best resulting models (polynomial and sigmoid) are illustrated in Figures 32 to 35.

Figure 32 shows classification results for the MERIS data when our framework uses a normalized MCI index with a sigmoid kernel. Figure 33 shows the resulting comparison between estimated and observed models for MERIS.

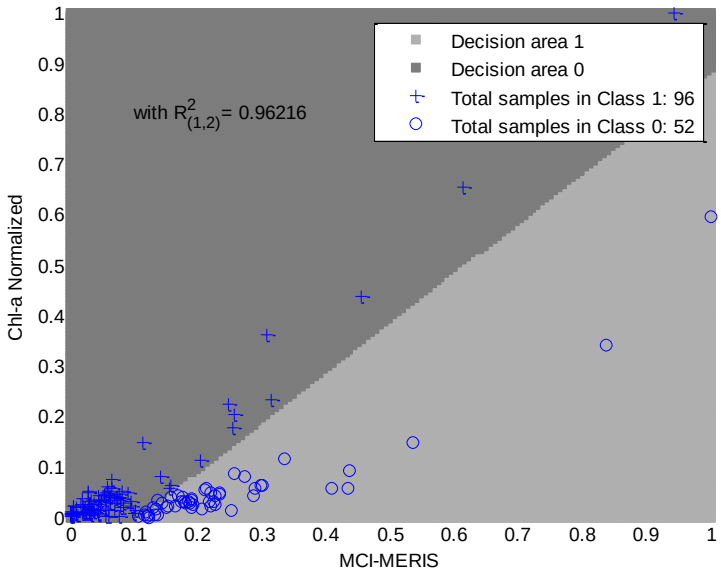


Figure 32. Final classification for MERIS data with an exponential partitioning and a sigmoid kernel.

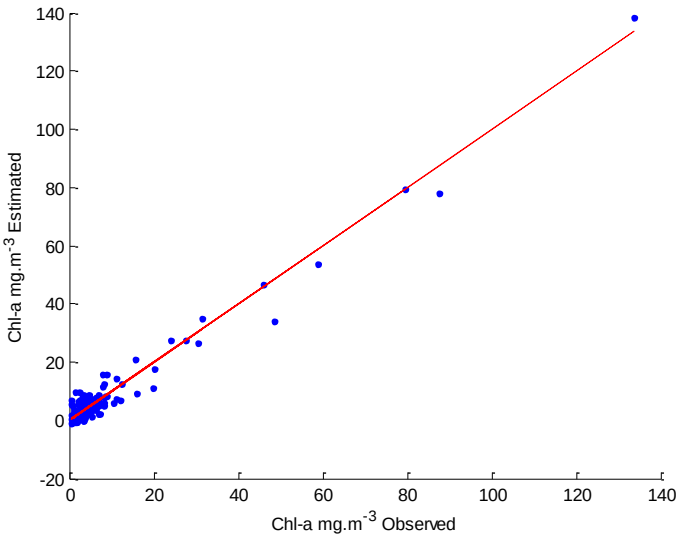


Figure 33. Estimated chl-a vs. observed chl-a values for MERIS

Similarly, classification results for the MODIS data using normalized FHL with a polynomial kernel are illustrated in Figure 34, while Figure 35 shows the resulting comparison between the estimated and observed models for MODIS.

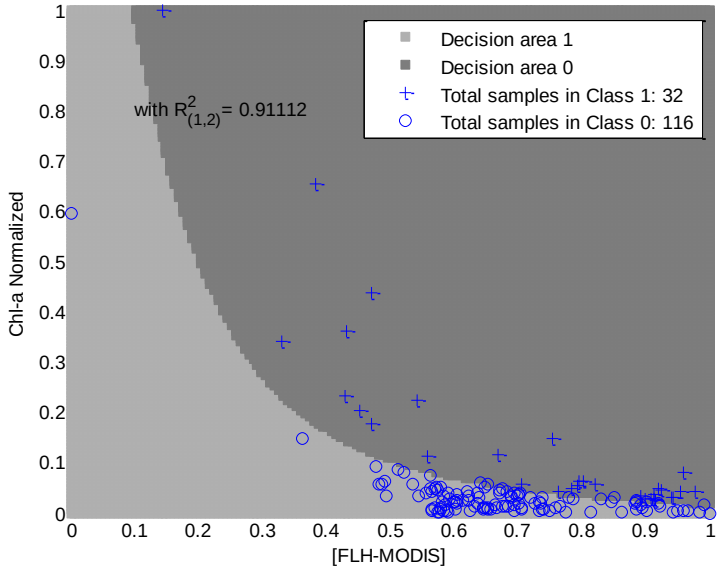


Figure 34. Final classification for MODIS with a cubic polynomial partitioning

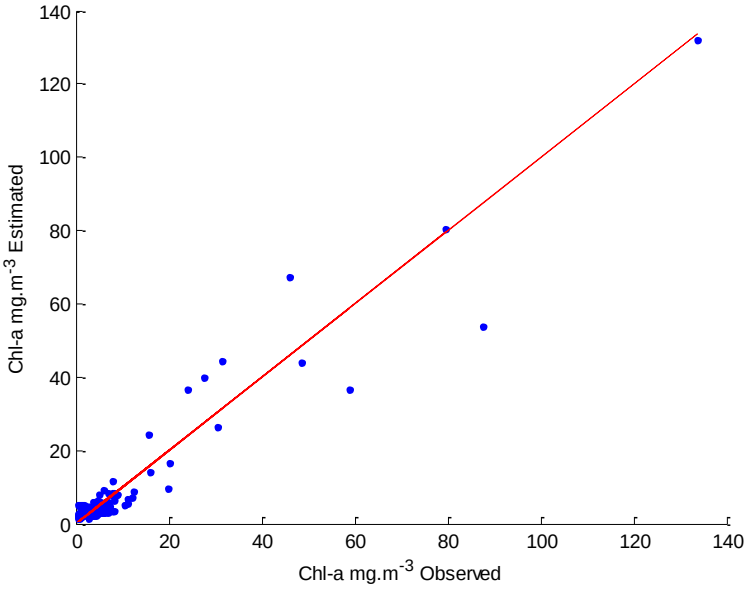


Figure 35. Estimated chl-a vs. observed chl-a values for MODIS

Finally, Table 14 summarizes the improvement obtained when using our method with different initial partitioning mechanisms.

Kernel	Partition mechanism		
	Linear [%]	Polynomial [%]	Exponential [%]
MODIS			
Linear	48	15	25
RBF	18	23	28
Sigmoid	47	18	28
Polynomial	47	25	29
MERIS			
Linear	30	23	20
RBF	27	22	20
Sigmoid	30	23	24
Polynomial	22	22	17

Table 14. Model improvement indices

The model improvement index (MII) applied in Table 14 to evaluate the results produced by using different kernels on each PM is calculated as follows:

$$MII = (R_{1,2} - R_{PM}) * 100\% \tag{38}$$

where R_{PM} is the coefficient of determination produced by each partition, and $R_{1,2}$ is the final coefficient of determination produced by our method.

The final models for both modalities are as follows. Let $\hat{y}_0$ be the predicted value of chl-a in class 0, obtained from the final labeling procedure, and $\hat{y}_1$ in class 1. For MERIS, where *X* values correspond to MCI indices, the two models are:

$$\begin{cases} \hat{y}_{0,i} = -232679.296 * e^{(5.355*X_i)} + 232678.747 * e^{(5.362*X_i)} \\ \hat{y}_{1,i} = 3.166 * e^{(49.366*X_i)} - 14.747 * e^{(-234.240*X_i)} \end{cases} \quad (39)$$

with X varying from (-1.20×10^{-4}) to 6.52×10^{-2}.

For MODIS, where X values correspond to FHL indices, the obtained models are:

$$\begin{cases} \hat{y}_{0,i} = 25.798 - 9193.272 * X_i + 1.427 \times 10^7 * {X_i}^2 - 7.634 \times 10^7 * {X_i}^3 \\ \hat{y}_{1,i} = 5.861 - 2423.418 * X_i + 6.0921 \times 10^5 * {X_i}^2 - 5.447 \times 10^7 * {X_i}^3 \end{cases} \quad (40)$$

with X varying from (-7.4×10^{-3}) to 6.2×10^{-3}.

6.11 Conclusions

Two important components are required in machine learning: the data and the algorithm. From the data perspective, we introduced the concept of policy layer, which uses the largest Euclidean distances, calculated from each data sample of the given data and the regression curve obtained in the initial partitioning. This strategy allowed us to identify samples with unique and strong class membership, thus alleviating the risk of using samples with dual membership and consequently unsatisfying outcomes.

From the perspective of the algorithm, our framework capitalizes on the combined use of unsupervised and supervised techniques, allowing us to exploit the best features from each technique and to produce the highest quality and accuracy in the resulting data models. This fact is important in environmental management, when the need for robust and precise data models is essential to develop standards and reliable value references for environmental legislation and environmental risk management.

The proposed iterative learning algorithm was validated using two independent multispectral datasets (data from MERIS and MODIS satellites) collected by the Lake Winnipeg Research Consortium, Canada. Our methodology significantly improved the resulting data model. The final labeling process resulted in new regression models with a lower value of residual sum of squares (unexplained variance) as shown in Tables 9 through 11, as compared to the classical regression method used in the initial partitioning as shown in Table 8. When modeling spectral information from MODIS, the improvement

was of the order 20.25% to 40% on average, whereas the MERIS model improvement was on average of the order of 20.25% to 27.25%.

Our iterative process successfully dealt with two practical restrictions: the lack of ground truth information regarding the water type classification and the absence of a suitable label set.

In our experiments, the classifier was fed with imbalanced positive and negative training data sets. Using this framework, we also improved the generalization power of the models, obtaining more robust data models. It should be noted that the issue of imbalanced training sets has been extensively researched in the area of machine learning. A specific implementation of the proposed framework may incorporate, if desired, one of several techniques of dealing with imbalanced data sets as found in [154].

CHAPTER VII: SOLVING A CLASSIFICATION PROBLEM

This chapter demonstrates the applicability of our methodology in a non-stationary environment when temporal data coming from multiple wireless sensors are used for human locomotion recognition. As presented in Chapter II, section 2.3, two major issues are addressed in this problem, the difficulty to deal with the motion during the transition period between two activities and the presence of noise that alters the resulting readings. When monitoring a multimodal system using information acquired by wireless sensors, the designer must address problems associated with several sensor-related factors, such as data alignment, data losses, and noise, among other experimental constrains. This situation represents a challenge because the multimodality influences the input data quality, deteriorating the resulting model accuracy.

The problem of recognizing human activity is solved here by adapting the framework presented in Chapter V. Our challenge – apart from a proper modification of the learning framework - is to classify low-intensity human locomotion activities, such as walking, standing, lying and sitting. To accomplish this, we modify the data pre-processing phase to handle the temporal signals. First, we use a timestamp function to organize the given dataset in a coherent time-event order. Second, we introduce a two-stage consecutive filtering approach used to enhance the quality of the given data, thus minimizing the effect of spurious data that could otherwise interfere with the classification process.

7.1 Dataset description

To demonstrate the capabilities of the proposed framework on classification problems, we adapted it to work on temporal data acquired from body–worn sensors, using the open source dataset Opportunity in view of recognizing human activity. The Opportunity dataset has been previously used as a benchmarking reference for modeling different systems, such as labeling large robot-generated activity datasets [125], sensors relocation due to replacement or slippage [126,127], dynamic sensor selection with power minimization

[128], and other application-related initiatives [129]. According to the technical description of the Opportunity project [130], the body-worn sensors used in this experiment are as follows: twelve 3-axial acceleration sensors and seven inertial measurement units – IMUs (i.e. Xsens MT9, technical specifications in reference [130]). The location of these units is summarized in Table 15 [6]. The dataset has a total of 58 dimensions including the time stamp. Each device senses the acceleration in three perpendicular axes, recording the acceleration values at the sampling rate of 30 Hz. Records are labeled according to four primitive classes, namely walk, lie, sit and stand. The signal acquisition protocol is performed under a pre-established scenario with six experimental sessions (or runs), performed independently by each of four users. The extracted dataset contains a total of 869,387 samples, which are distributed as follows: 234,661 samples for user 1; 225,183 samples for user 2; 216,869 samples for user 3, and 192,674 samples for user 4.

Sensor type	Left foot	Right foot	Up right knee	Low right knee	Hip	Back	Right forearm	Left forearm	Right arm	Left arm	Right hand	Left hand	Right wrist	Left wrist	**Total**
IMU	1	1				1	1	1	1	1					**7**
3-axial			1	1	1	1	2	2			1	1	1	1	**12**

Table 15. Placement of sensors (as specified in the OPPORTUNITY Activity recognition dataset [6])

Figure 36 shows the first 400 samples collected from user 1 by a 3-axial acceleration sensor placed on the hip. Amplitude values are in gravities ($g = 9.8\frac{m}{s^2}$), and time in seconds.

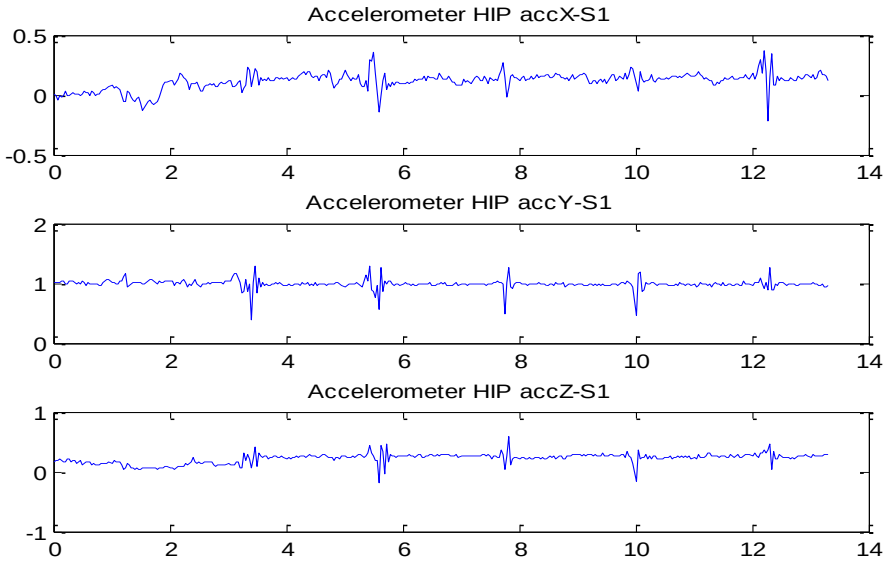


Figure 36. Example of readings collected from user 1 by a 3-axial acceleration sensor placed on the hip.

7.2 Data pre-processing

As presented in Chapter II, section 2.3 and 2.5, wireless wearable sensors are affected by various technical constraints during data acquisition that decrease data quality. The non-ergodicity of the acquisition process, especially when processing signals from acceleration sensors, will result in poor learning performance, affecting applications involving multi-class classification. The proposed pre-processing data procedure address these issues in two steps. The first one consists in excluding values affected by data losses and the noise presented in the communication path. To deal with the problem of missing data, we fused all readings produced by each sensors. We proposed a double filtering stage, using a finite impulse response (FIR) filter to enhance the precision of the given sensor readings. Then, we coupled a second filtering, based on wavelets, to efficiently denoise raw data. The process is explained in the following two sections.

7.3 Finite impulse response filter

In our analysis, high frequency bands are not relevant, since users are not performing routines with high motion intensity like running, jumping or jogging. Moreover, in general, the acceleration signals present a high level of correlation within a limited-length time

window, implying that a FIR filter can be efficiently used in this application [131,132,133]. We use a FIR passband architecture of the order of 40, which is a compromise between the complexity of the signals under observation and the delay introduced by higher orders. We also use cutoff frequencies of 2Hz and 15Hz due to the characteristics of the 3-axial acceleration sensors used, which have sampling frequencies of 32Hz and 64Hz. The frequency of 15Hz meets the Nyquist theorem requirements ($f_s > 2 * f_n$), where f_s is the sampling frequency and f_n corresponds to the motion intensity [134]. The frequency of 2Hz is selected according to criteria presented in [134]. The selected passband provides us with an optimal range of motion intensity, since the recorded motion in this study does not go beyond 15Hz, making it acceptable to perform human motion sensing. Once the FIR filtering is processed, we proceed with the second stage - based on wavelets.

7.4 Wavelet filter

To efficiently denoise raw data, we include a mechanism that guarantees that the resulting classification model is not biased due to the quality of the input data [135]. In general, the acceleration sensors are influenced by several noise sources, such as electrical noise induced by the electronic devices [136], or the noise produced by the wireless communication processes, resulting from the propagation phenomenon and causing distortion in the transmitted signal as mentioned in Chapter II, section 2.3. The noise present in the acceleration sensor measurements has commonly a flat spectrum. It is present in all frequency components, constituting a serious challenge for traditional filtering methods, which by removing sharp features, can introduce distortions in the resulting signal. Decomposition of the noisy signal into wavelets [137] eliminates small coefficients, commonly associated with the noise, by zeroing them, while concentrating the signal in a few large-magnitude wavelet coefficients. Wavelet filtering consists in the decomposition of the signal into wavelet basis functions (WBF) $\psi_{a,b}(t)$ given by [138]:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{41}$$

where $a, b \in \mathbb{R}$ are called scale and position parameters respectively. The wavelet basis is defined by the selection of the previous parameters. Their choice is commonly known as

critical sampling, hence $a = 2^{-j}$ and $b = (k)2^{-j}$, where k and j are integers, will give a spare basis [139]. The function in (41) can be represented in powers of two; this strategy is called dyadic and can be formulated as:

$$\psi_{m,n}(k) = 2^{\frac{-m}{2}}\psi(2^{-m}k - n) \tag{42}$$

where $m, n \in \mathbb{Z}$. By computing an inner product between any given function $f(k)$ and $\psi_{m,n}(k)$, we can obtain the discrete wavelet transform as:

$$DWT(m,n) = \langle f.\psi_{m,n}\rangle = 2^{\frac{-m}{2}}\sum_{k=-\infty}^{\infty} f(k).\psi(2^{-m}k - n) \tag{43}$$

The advantage of having a function represented in wavelets is the flexibility of the mathematical model, defined in the domain of both frequency and time; in the frequency domain via dilation and in the time domain via translation. This feature is helpful also when removing noise, because the main characteristics of the original signal can be more easily preserved. Wavelet de-noising involves thresholding of a range of wavelet coefficients. Setting wavelet coefficients below a specific value (λ) to zero [138] is called hard-thresholding and it can be represented as:

$$f(k) = \begin{cases} k \text{ if } |k| > \lambda \\ 0, \text{otherwise} \end{cases} \tag{44}$$

In addition, if the wavelet coefficients are below the threshold value, they are shrunk, and when the coefficients are above the threshold value, they are scaled. This process is called soft-thresholding and can be represented as:

$$f(k) = \max(0, 1 - \frac{\lambda}{|k|}) \tag{45}$$

In literature, we can find four well-known threshold estimation methods [138], namely the minmax criterion [139], the Square root log (sqtwolog) criterion [139], the Rigrsure criterion [140] and the heursure criterion. In general, the correct selection of the threshold

leads to a better noise suppression; a large threshold value will bias the estimator, while a low value will increase the variance. The thresholding approach selected in this work employs the Sqtwolog criterion, because it guarantees a high signal-to-noise ratio (SNR) with a low mean square error (MSE). The threshold values are calculated by the universal threshold $\sqrt{2 * \ln(.)}$ or $\lambda_i = \sigma_j\sqrt{2\log(N_j)}$, where N_j is the length of the noise at j^{th} scale and σ_j is the Median Absolute Deviation (MAD) at the j^{th} scale given by [138]:

$$\sigma_j = \frac{MAD_j}{0.6745} = \frac{\text{median}(|\omega|)}{0.6745} \tag{46}$$

where ω represents the wavelet coefficients at scale j. The value 0.6745 in (46) is obtained as: $\frac{1}{Erf(0.5)*\sqrt{2}}$, where the Gauss error function (Erf) is computed by integrating the normal distribution. This value will scale the MAD to obtain an approximation for sigma (only for a Gaussian distribution). Figure 37 shows the result when using 2-stage filtering.

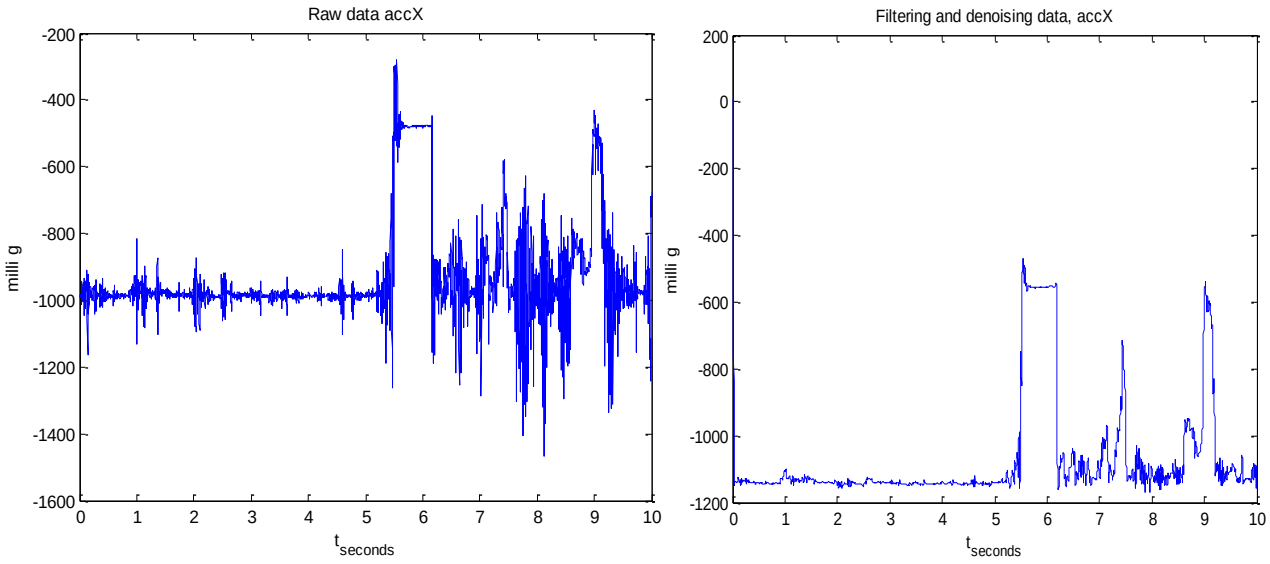


Figure 37. Measurements recorded for user 1 for a 3-axial acceleration sensor located on the up-right knee: (a) raw data and (b) after applying 2-stage filtering

7.5 Training samples extraction

After filtering the raw data, we proceed with the feature extraction and selection process. The aim is to retrieve a set of data with high correlation, allowing us to extract the best candidates for the training dataset [141]. This process focuses on the extraction of kinematics features, such as roll, pitch, yaw (RPY), and the norm of the axial components produced by each of the body-worn sensors. Our first feature set is based on the signal magnitude vector (SMV). At each time instance *j*, the acceleration sensor *k* produces a 3-axial vector, consisting of acceleration values along a system of orthogonal axes $a_{j,k} = (acc_x, acc_y, acc_z) \in \mathcal{R}^3$. For each sensor, we first retrieve the single magnitude vector $|a_{j,k}|$. The second feature set is related to roll, pitch and yaw (RPY) angles, calculated as follows:

$$roll_{j,k} = atan\left(\frac{acc_x}{\sqrt{acc_y + acc_z}}\right);\ pitch_{j,k} = atan\left(\frac{acc_y}{\sqrt{acc_x + acc_z}}\right); yaw_{j,k} = atan\left(\frac{acc_z}{\sqrt{acc_x + acc_y}}\right) \quad (47)$$

Finally, we build a matrix with all axial components produced by all sensors under observation:

$$acc_{x,y,z,k} = \{[acc_{x,k}], [acc_{y,k}], [acc_{z,k}]\} \quad (48)$$

This matrix has $n \times a_{j,k} \times k$ components, where n is the number of samples in each experiment for k sensors in $a_{j,k}$ dimensions. To deal with the absence of some values, we use principal component analysis (PCA) and singular value decomposition (SVD). PCA provides a mechanism to reduce dimensionality, while SVD provides a convenient way to extract the most meaningful data. Combining these techniques, we find data dependency while removing redundancy. PCA [142] and SVD [143] ensure the preservation of the nature of the data and consequently the original data structure in each feature category in the resulting transformed data. When applying PCA, we reduce the problem to only two principal components. Similarly, when SVD is applied, each feature is reduced to two SVD dimensions, as shown in equation (49). The new target function $f_{j,k}()$ is represented as follows:

$$f_{j,k} = f\big(pca\ (RPY), pca(SMV), pca(acc_{x,y,z,k}), svd(RPY), svd(SMV), svd(acc_{x,y,z,k})\ \big) \quad (49)$$

where j corresponds to each observation produced by sensor k. We are therefore reducing our analysis to a function with three attributes $(\mathrm{RPY}, \mathrm{SMV}, \mathrm{acc_{x,y,z,k}})$ using two mathematical methods, PCA and SVD. Our learning framework aims to classify human activities using a single multi-class SVM classifier, such as shown in Figure 38.

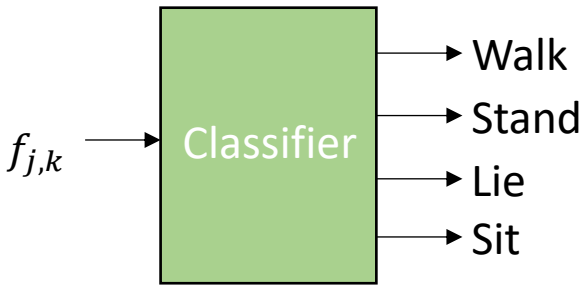


Figure 38. Locomotion recognition using a single classifier with a multimodal input

To achieve this goal, we must deal with two data constrains: 1) the large size of the experimental dataset, containing in many cases overlapping class members and high data density; and 2) the non-ergodicity of the recorded signals. To improve the classification accuracy, while keeping the required processing time at the minimum, features $((\mathrm{f_1}, \mathrm{f_2}), \ldots, (\mathrm{f_j}, \mathrm{f_k}))$ produced by (48) are grouped pairwise to cover all the possible combinations. The candidates for the training dataset are then determined by measuring the Euclidean distance between each class member and the centroid of each distribution of $(\mathrm{f_j}, \mathrm{f_k})$. If the resulting distance is larger than the mean plus the standard deviation of all resulting Euclidean distances, then the class member is considered a candidate for the training set. This process leads to the creation of support vectors, which generate the optimal separation planes to classify the remaining data with only a fraction of the total data presented for each user experiment. The following procedure, illustrated in detail in Figure 39, summarizes the process for the extraction of the training dataset:

1. Select sensor readings recorded (in this case, from the Opportunity dataset [6]), perform time stamping and missing-data imputation.

2. Select band-pass FIR filter (2-15 Hz) and perform wavelet de-noising using Sqtwolog criterion (Figure 37)
3. Perform multimodality assessment
4. Extract kinematics features: signal magnitude vector, roll, pitch, yaw (RPY), and the norm of the axial components produced by each of the body–worn sensors, in order to create the target function $f_{j,k}()$ as indicated in Equation (48). This step will produce twelve features.
5. Build a subset of features (f_j, f_k), where $j = (1,..,11)$ and $k = (2,..,12)$ from target function $f_{j,k}()$ and extract classes presented in subset (f_j, f_k) (Figure 40a).
6. Select a pair of classes $(x_n, x_{,m})$, from subset (f_j, f_k) where $n = (1,..l-1)$ and $m = (2,..l)$ and l is the number of labels in the dataset (in our case four classes corresponding to each locomotion activity), and extract centroids produced by members of each class.
7. Extract the Euclidean distance between each class member in (x_n) and the centroid of the class (x_m). Store the results in a vector of distances $R_{n,m}(j)$:

$$R_{n,m}(j) = \left|\left(x_{n,m}(j)\right) - Centroid_{n,m}\right| \tag{50}$$

where *n* and *m* are the classes of (f_j, f_k), j is a class member and $Centroid_{n,m}$ is the centroid of the class, with respect to the discriminating hyperplane, of the class member under evaluation (Fig. 40b).

8. If the resulting Euclidean distance vector $R_{n,m}(j)$ satisfies condition (51), then the class member is a candidate for the training dataset.

$$R_{n,m}(j) \geq \overline{R_{n,m}} + \sigma(R_{n,m}) \tag{51}$$

where $\overline{R_{n,m}}$ and $\sigma(R_{n,m})$ are the mean and standard deviation of the Euclidean distance vector $R_{n,m}(j)$. The candidate is stored in a vector of candidates (VoC), VoC$(x_{n,m}(j))$ (Figure 40c).

9. Repeat steps 9 to 12 until $n = l - 1$ and $m = l$
10. Repeat steps 7 to 13 until $j = 11$ and $k = 12$.

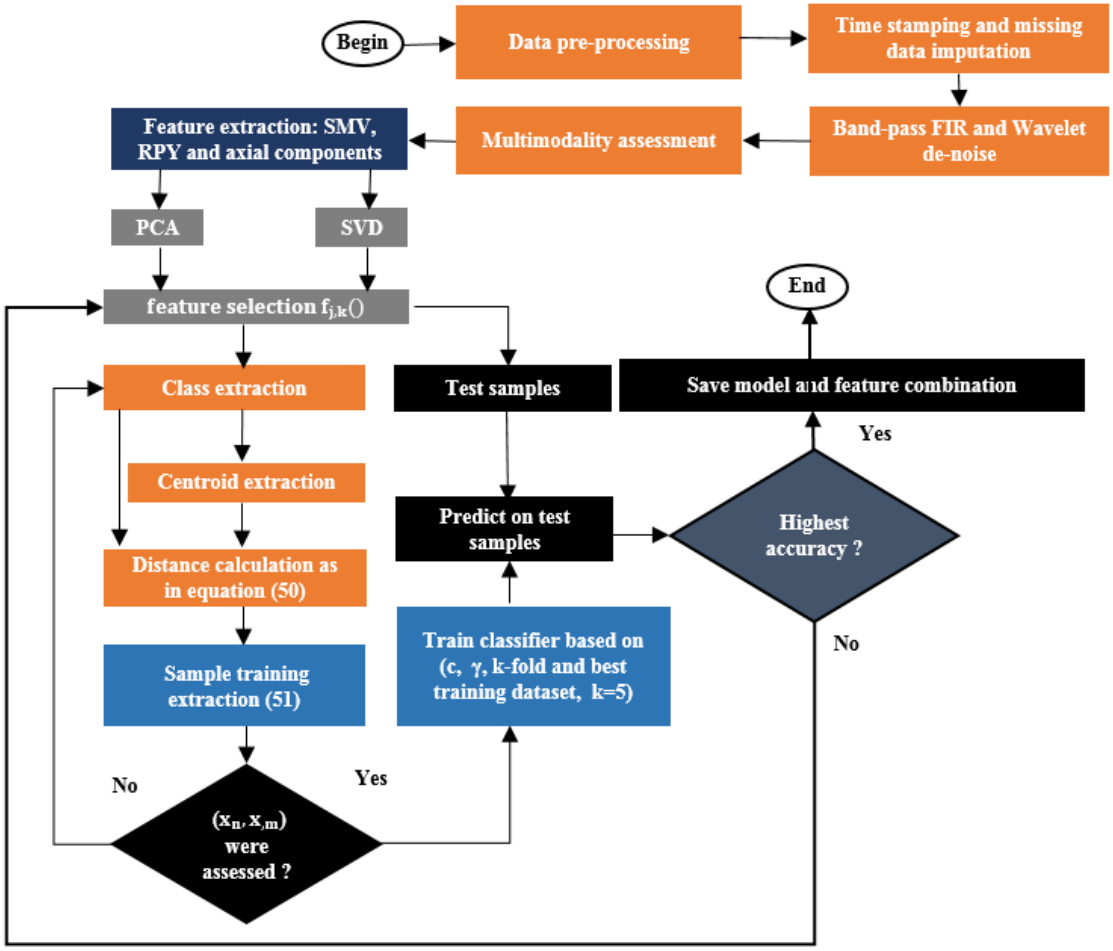


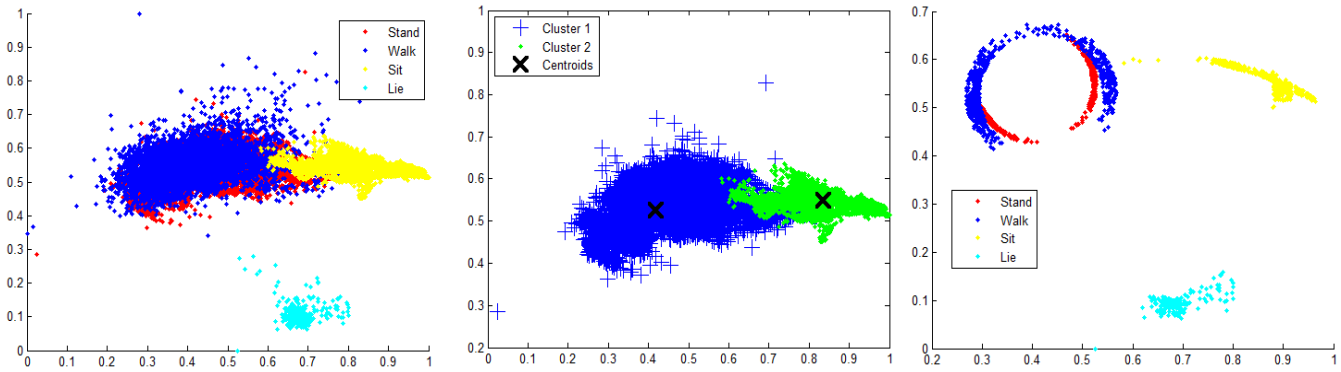
Figure 39. Modified framework for human locomotion recognition

Figure 40a shows the data distribution when PCA is applied to features generated by axial components from the sensor measurements, for example, for the first two PCA components $f_{1,2} = f\left(pca(acc_{x,y,z,k})\right)$. Both components are called scores. The advantage of PCA is that the resulting score does not change the order of the original rows (observations), helping us to preserve the previously assigned labels. In this figure, we also observe a clear separation between the sit (shown in yellow) and the lie (shown in cyan) instances, while the stand (shown in red) and the walk (shown in blue) classes overlap. Permutation of the members from $f_{j,k}$ helps us to find different data distributions from the original data

structure. This provides some distributions with linearly separable data, which decreases the misclassification error rate produced by the multi-class classifier.

Figure 40b represents the extraction of two classes $(x_n, x_{,m})$ from $f_{j,k}$ and their respective clusters. We extract the samples producing the largest Euclidean distances between each of them and their corresponding centroid. The whole set of training samples is extracted by pairing all given classes (*stand*=1, *walk*=2, *sit*=3 and *lie*=4) as follows: (1,2),(1,3),..,(3,4).

Figure 40c shows the resulting VoC($x_{n,m}(j)$) composed by samples that satisfy equation (51), that is:VoC($x_{n,m}(j)$)= $[(Class_1, Class_1), (Class_1, Class_2), \dots, (Class_{n-1}, Class_m)]$, where *n*, *m* = 4. The resulting VoC($x_{n,m}(j)$) provides an effective way to deal with non-separable data (data overlapping). Because the SVM classification depends only on the training samples near the decision boundary, the optimal separation margin will be determined by the separation of the training samples controlled by the cost parameter C. The improvement can be observed by comparing the separation on Figure 40a with Figure 40c, where we notice a strong overlapping of data samples, in particular for the stand, walk and sit classes.



(a) (b) (c)

Figure 40. Training sample extraction results. (a) PCA is applied to $acc_{x,y,z,k}$(data distribution corresponds to the first and second principal components). (b) Classes are extracted in pairs $(x_n, x_{,m})$, centroids are extracted, and Euclidean distances are calculated according to step 6; and (c) Training candidates extracted after applying the policy layer.

7.6 Model selection

Once the best training dataset VoC($x_{n,m}(j)$) is identified, we proceed with the selection of the best classification model using a multi-class SVM classifier with an RBF kernel. The training and testing samples are normalized in the range of 0 to 1. The kernel selection is done based on an experimental performance evaluation with different kernels, e.g. linear, cubic polynomial and sigmoid. The evaluation presented in [18], and confirmed by initial tests on the Opportunity dataset, indicates that RBF kernels consistently produce models with the lowest or close to the lowest misclassification error rates. The selection of the one-versus-all (OVA) classification method reduced our original multi-class problem to a binary classification problem. Designing the SVM classifier requires to find the best combination of the cost and gamma (C, γ,) parameters. By using a grid search (C and γ) and a *k*-fold cross validation process with $k = 5$ (four subsets for training and one subset for testing) we determine the best performing hyper-parameter. This process allows us to find a tradeoff between bias and variance by adjusting C and γ. To find the best C and γ we use a grid search, where $C = (2^{-5}, .., 2^{7})$ and $\gamma = (2^{-5}, .., 2^{7})$. In practical terms, the best combination, in the sense of a high variance and a low bias, is that of large C with small γ.

The resulting model is then used to predict the labels on the testing dataset. Once the classification rate is determined, the algorithm stores the accuracy values, features (f_j, f_k), C, γ and the size of the VoC($x_{n,m}(j)$) and repeats the process until all combinations of (f_j, f_k) are exhausted.

7.7 Model performance evaluation

Model performance is measured by using accuracy measures and the F-score, summarized in Chapter III, section 3.4. Additionally, we will compare our results with values reported by the Opportunity team in section 7.8.2

7.8 Experimental results

The proposed solution, based on iterative learning, is tested in two scenarios, the first one using a single-stage filtering and the other one on a two-stage consecutive filtering. The whole process is presented in Figure 41.

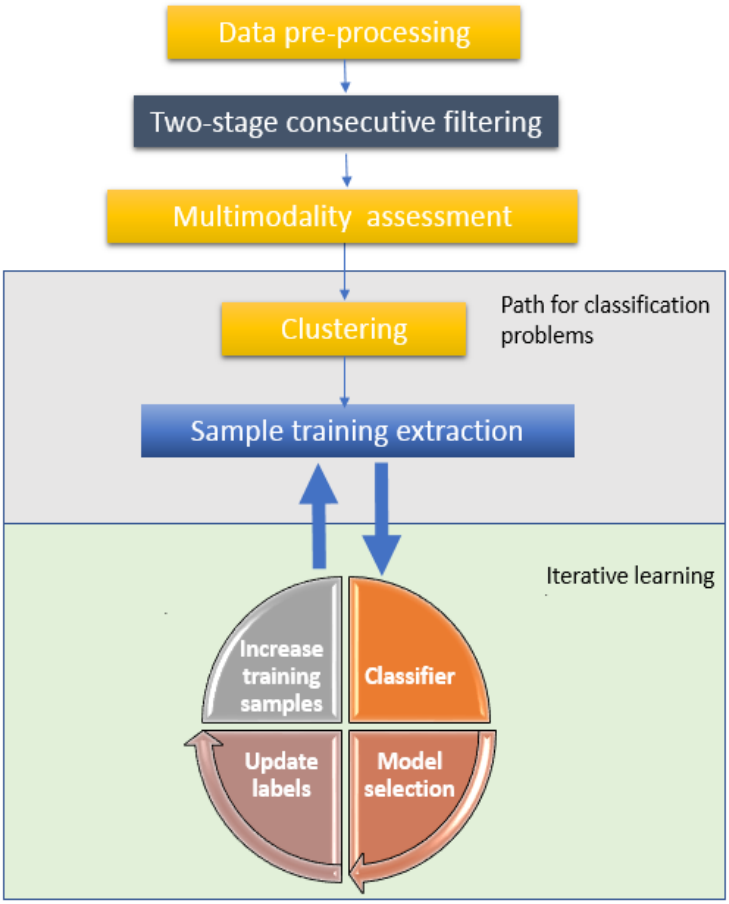


Figure 41. Process block diagram implemented during experiments

7.8.1 Results obtained using single-stage wavelet filtering

The proposed process was evaluated initially using a single wavelet filtering stage in three experiments: two considering the measurements of a sole sensor and one combining the use of various sensors. Two measures were used to validate the results, namely the prediction accuracy (Acc) and the size (as percentage of the total dataset) of the training dataset that was used for classification (Ts):

$$\text{Acc} = \frac{Labels\ correctly\ predicted}{(size\ of\ user's\ dataset)} \times 100\%;\ Ts = \frac{size(R_{n,m})}{(size\ of\ user's\ dataset)} \times 100\% \qquad (52)$$

It is important to note that the values of Acc and Ts depend on the size of the user dataset and the resulting value of $R_{n,m}(j)$ in eq. (51). These values are changing with the number of measurements done for each user in each experiment. Table 16 presents the results when

using only data obtained from the IMU sensors. Table 17 shows the values for Acc and Ts when using data obtained from 3-axial acceleration sensors, and Table 18 when using data obtained when fusing measurements from the 3-axial acceleration sensors and IMU devices in three experiments. The type of the experiments and users are such as described in section 3.1 of [6]. The results obtained by our iterative learning framework are compared with the case in which 80% of total of data are used of each user experiment, which is a common practice when a *k*-fold cross-validation process is performed, with $k=5$. In this case, the samples are randomly selected from the input domain without the selection of best training candidates.

	Experiments					
	Experiment 1 (Acc% /TS %)	**Experiment 2 (Acc% / TS %)**	**Experiment 3 (Acc% / TS %)**	**Experiment 1 (Acc% / 80%)**	**Experiment 2 (Acc% / 80%)**	**Experiment 3 (Acc% / 80%)**
User 1	80 / 4.47	75.36 / 1.19	81 / 3.31	83.92	74.76	80.55
User 2	71.56 / 4.97	47.43 / 11.96	65.23 / 10.18	77.53	77.17	78.31
User 3	70,64 / 5.70	57 / 7.70	73.28 / 0.16	71.46	69.43	75.19
User 4	66.19 / 2.8	61.27 / 2.70	78 /1.86	77.2	74.46	79.88

Table 16. Classification performance for IMU sensors data

	Experiments					
	Experiment 1 (Acc%/ TS%)	**Experiment 2 (Acc% / TS %)**	**Experiment 3 (Acc% / TS %)**	**Experiment 1 (Acc% / 80%)**	**Experiment 2 (Acc% / 80%)**	**Experiment 3 (Acc% / 80%)**
User 1	82./3.03	79.23 / 11.38	83.71 / 9.11	83.12	79.12	80.56
User 2	52.42 / 2.96	50.86/12	57.84 / 1.89	69.9	75	73.56
User 3	69 / 13.16	67.86 / 0.60	76.62 / 3.37	72.09	65.21	77.51
User 4	66 / 1.63	64 / 10.4	77.53 / 3.45	71.59	76.15	87.55

Table 17. Classification performance for 3-axial acceleration sensors data

	Experiments					
	Experiment 1 (Acc%/ TS %)	Experiment 2 (Acc% / TS %)	Experiment 3 (Acc% / TS %)	Experiment 1 (Acc% / 80%)	Experiment 2 (Acc% / 80%)	Experiment 3 (Acc% / 80%)
User 1	80.62 / 7.15	77.21 / 8.3	84.77 / 8.17	81.11	75.92	80.85
User 2	65.85 / 8.78	45.16 / 12.49	66.25/ 0.90	71.54	76.68	74.56
User 3	58.49 /13.93	67.62 / 1.42	70.35 / 2.97	72.30	65.18	77.08
User 4	66.48 / 0.70	66.64 / 11.41	71.54 / 4.14	73.43	75.80	87.38

Table 18. Classification performance for IMU and 3-axial acceleration sensors data

These results are compared graphically in Figure 42 that shows the average accuracy when using two training dataset selection strategies: iterative with a limited number of training samples (in blue) and supervised one with a large number of training samples (in red). One can observe that using on average 7.33% of the dataset for training (Figure 43), the performance achieved is only 7.28% under the performance obtained when the classifier processes a much higher number of training samples.

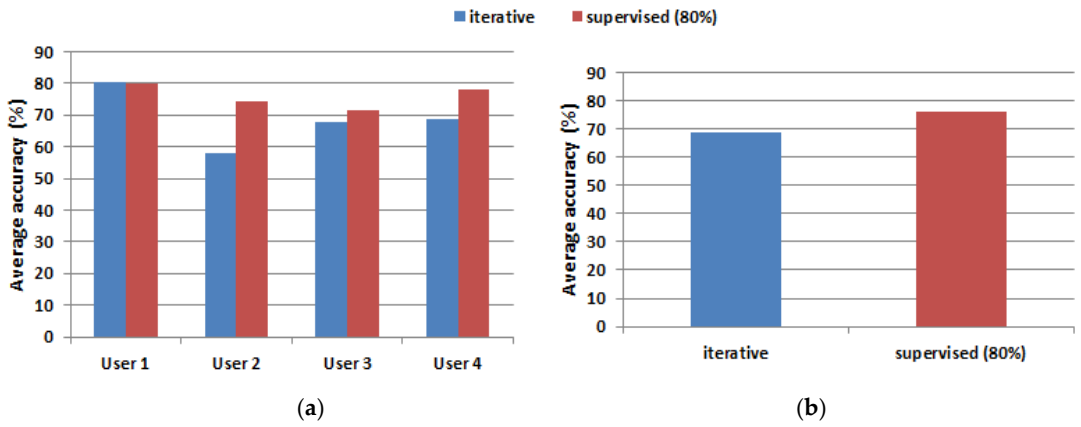


Figure 42. Accuracy comparison: (a) accuracy generated by SVM multi-class classifier on each user; and (b) average accuracy for iterative versus supervised methods

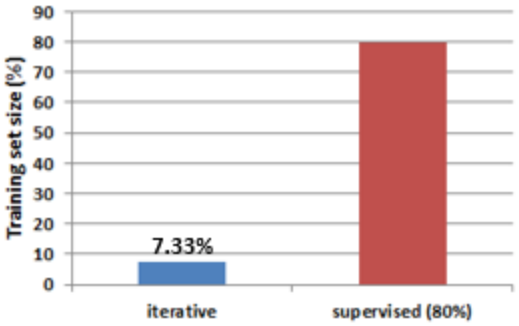


Figure 43. Training size comparison (Iterative only uses on average 7.33% of the input data size)

The use of a smaller training set leads as well to an important decrease in the computation time. The average processing time per user is roughly 35 minutes when using the training with 80% of the dataset (Matlab on a single processor Intel 7, 6 Gb RAM memory). The use of the iterative process leads to a reduction in the average time for processing an experiment to about 5 minutes, which is less than 15% of the time required by the fully supervised process.

7.8.2 Results obtained using two-stage consecutive filtering.

In this section, we present the experimental results when the bandpass FIR filter and subsequently wavelet de-noising are applied on the data collected from IMU sensors, 3-axial acceleration sensors and when fusing measurements from the IMU and 3-axial acceleration sensors (Tables 19 through 21). These values are compared with results obtained in section 8.2.1. As detailed in section 7.3 and 7.4 , it is expected that performance will increase as a result of this two-stage consecutive filtering.

	Experiments					
	Two-stage consecutive filtering			**Wavelet filtering**		
	Experiment 1 (Acc% / TS %)	**Experiment 2 (Acc% / TS %)**	**Experiment 3 (Acc% / TS %)**	**Experiment 1 (Acc% / TS %)**	**Experiment 2 (Acc% / TS %)**	**Experiment 3 (Acc% / TS %)**
User 1	80.23/5.5	79.5/6.05	80/5.9	80 / 4.47	75.36 / 1.19	81 / 3.31
User 2	76/8.19	50.23/13.8	76.91/6.18	71.56 / 4.97	47.43 / 11.96	65.23 / 10.18
User 3	73.55/5.8	68.22/5.68	76/6,01	70,64 / 5.70	57 / 7.70	73.28 / 0.16
User 4	75.62/4.23	67.71/5.11	72.85/13.79	66.19 / 2.8	61.27 / 2.70	78 /1.86

Table 19. Classification performance for IMU sensors data: filtering comparison

	Experiments					
	Two-stage consecutive filtering			Wavelet filtering		
	Experiment 1 (Acc% / TS %)	Experiment 2 (Acc% / TS %)	Experiment 3 (Acc% / TS %)	Experiment 1 (Acc% / TS %)	Experiment 2 (Acc% / TS %)	Experiment 3 (Acc% / TS %)
User 1	81.93/6.05	73.5/6.05	81.48/5.9	82.82 / 3.03	79.23 / 11.38	83.71 / 9.11
User 2	63.25/5	66.53/14	72.50/12.72	52.42 / 2.96	50.86/12	57.84 / 1.89
User 3	68.38/7.4	71.60/5.29	78.44/5.46	69 / 13.16	67.86 / 0.60	76.62 / 3.37
User 4	73.63/6.67	72.07/6.33	79.80/6.03	66 / 1.63	64 / 10.4	77.53 / 3.45

Table 20. Classification performance for 3-axial acceleration sensors data: filtering comparison

	Experiments					
	Two-stage consecutive filtering			Wavelet filtering		
	Experiment 1 (Acc% / TS %)	Experiment 2 (Acc% / TS %)	Experiment 3 (Acc% / TS %)	Experiment 1 (Acc% / TS %)	Experiment 2 (Acc% / TS %)	Experiment 3 (Acc% / TS %)
User 1	87.26/6.28	78/5.47	82.30/6.39	80.62 / 7.15	77.21 / 8.3	84.77 / 8.17
User 2	67.5/7.2	71.50/6.40	75/7.46	65.85 / 8.78	45.16 / 12.49	66.25/ 0.90
User 3	74.45/5.12	70.82/5.40	71.67/5.69	58.49 /13.93	67.62 / 1.42	70.35 / 2.97
User 4	74.20/7.18	73/7.74	81.41/7	66.48 / 0.70	66.64 / 11.41	71.54 / 4.14

Table 21. Classification performance for IMU and 3-axial acceleration sensors data: filtering comparison

In general, we noticed a performance improvement when the framework uses a two-stage consecutive filtering. Deployment of the extra filtering stage generated an increase in the average accuracy. For example, for User 2, an average accuracy of 61.40% is obtained with wavelet filtering (Table 19). An average accuracy of 67.71% is obtained with two-stage consecutive filtering, which corresponds to an improvement of 6.30%.

Similarly, in Table 20, an average improvement of 12.72% can be noticed. Finally, in Table 21, for the same user we obtained an improvement of 12.24%. Results obtained by using a training dataset of 80% of total data are summarized in Tables 22 through 24. Better results are obtained when classification is performed on fused data coming from IMU and 3-axial acceleration sensors.

	Experiments					
	Two-stage consecutive filtering			Wavelet filtering		
	Experiment 1 (Acc% / 80 %)	Experiment 2 (Acc% / 80 %)	Experiment 3 (Acc% / 80 %)	Experiment 1 (Acc% / 80 %)	Experiment 2 (Acc% / 80 %)	Experiment 3 (Acc% / 80 %)
User 1	89.88	77.33	89.33	83.92	74.76	80.55
User 2	84.83	82.36	84.17	77.53	77.17	78.31
User 3	81.79	83.55	85.76	71.46	69.43	75.19
User 4	86.19	84	89.41	77.2	74.46	79.88

Table 22. Classification performance for IMU sensors data: filtering comparison

	Experiments					
	Two-stage consecutive filtering			Wavelet filtering		
	Experiment 1 (Acc% / 80 %)	Experiment 2 (Acc% / 80 %)	Experiment 3 (Acc% / 80 %)	Experiment 1 (Acc% / 80 %)	Experiment 2 (Acc% / 80 %)	Experiment 3 (Acc% / 80 %)
User 1	83.42	79.85	82.36	83.12	79.12	80.56
User 2	69.68	76.05	77.90	69.9	75	73.56
User 3	72.30	69.41	82.33	72.09	65.21	77.51
User 4	76.90	74.36	82.21	71.59	76.15	87.55

Table 23. Classification performance for 3-axial acceleration sensors data: filtering comparison

	Experiments					
	Two-stage consecutive filtering			Wavelet filtering		
	Experiment 1 (Acc% / 80 %)	Experiment 2 (Acc% / 80 %)	Experiment 3 (Acc% / 80 %)	Experiment 1 (Acc% / 80 %)	Experiment 2 (Acc% / 80 %)	Experiment 3 (Acc% / 80 %)
User 1	91.43	79.64	88.32	81.11	75.92	80.85
User 2	74.51	79.93	79.98	71.54	76.68	74.56
User 3	78.97	68.91	82.92	72.30	65.18	77.08
User 4	82.97	78.66	86.85	73.43	75.80	87.38

Table 24. Classification performance for IMU and 3-axial acceleration sensors data: filtering comparison

Figure 44 presents an accuracy comparison between the single-stage approach and the two-stage filtering process.

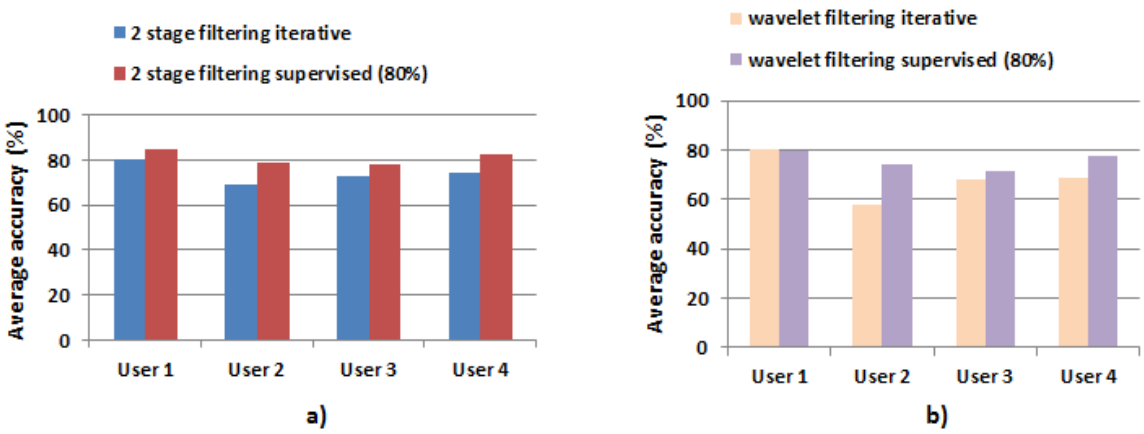


Figure 44. Average accuracy comparison between single-stage and two-stage filtering. (a) Average accuracy when using two-stage filtering and the iterative methodology (in blue) and when using the supervised method (in red); and (b) average accuracy comparison when using a single filtering and the iterative methodology (light yellow); and when using the supervised method (in light purple)

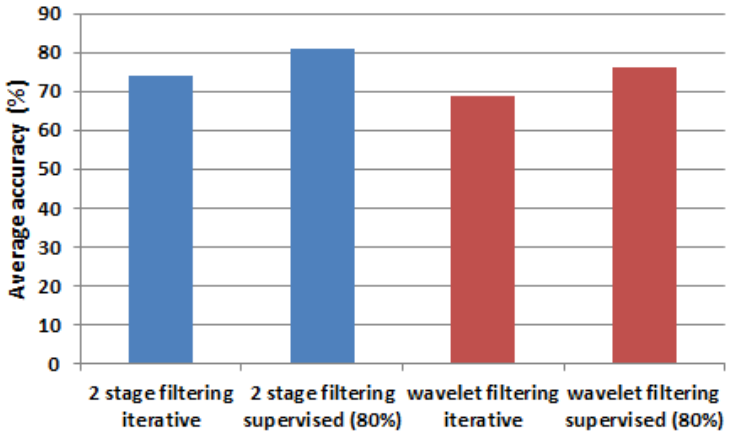


Figure 45. Average accuracy. Bars in blue represent average accuracy when two-stage filtering is used. Bars in red represent the results for single-stage wavelet filtering.

The approach with two-stage filtering, as compared with the wavelet filtering only, generated an accuracy improvement in those experiments where only a fraction of samples was used for training. Overall, the second filtering produced an average accuracy of 74.08% versus 68.76% produced by the single filtering approach, an equivalent of 5.32% of improvement. The model accuracy for user 2 was improved by 6.11% for readings obtained from 3-axial acceleration sensors and by 3.88% when IMU and 3-axial acceleration sensors were fused. The performance was improved by 5.03% for the case of the training size of 80% of the total amount of the input data as shown in Figure 45.

In previous experiments, we presented the results based on how effective the algorithm was in predicting the true values of a label. In this section, we quantify the classification results using the F_1 measure, which takes into account recall and precision metrics (See Chapter III, section 3.3) By applying the F_1 measure to each class, we have [153]:

$$F_1 = \sum_i 2 * \frac{\text{Precision}_i * \text{recall}_i}{\text{Precision}_i + \text{recall}_i} \times w_i \quad (53)$$

where i is the class index, $w_i = \frac{n_i}{N}$, N is the total number of samples, and n_i - the number of samples of the ith class. The results are presented in Table 25.

	Experiments					
	Experiment 1 (F_1 / TS %)	**Experiment 2 (F_1 / TS %)**	**Experiment 3 (F_1 / TS %)**	**Experiment 1 (F_1 / 80%)**	**Experiment 2 (F_1 / 80%)**	**Experiment 3 (F_1 / 80%)**
User 1	0.8506/6.28	0.7669/5.47	0.79/6.39	0.9103	0.7701	0.8786
User 2	0.62/7.22	0.6809/6.40	0.695/7.46	0.7324	0.7821	0.7545
User 3	0. 7283/5.12	0.6756/5.40	0.6346/5.69	0.7835	0.5805	0.8104
User 4	0.6847/7.18	0.6665/7.74	0.7627/7	0.8297	0.7691	0.8234

Table 25. F_1 measure for data fused from IMU and 3-axial acceleration sensors.

Figure 46 presents the F_1 measure for both learning schemes. One can notice a total average difference of 0.075 between the two methods, as compared to 0.0532 in Table 25.

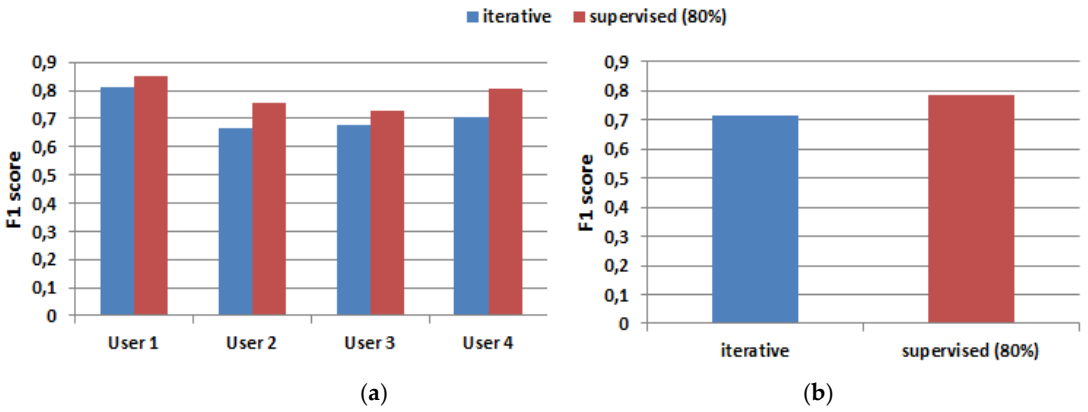


Figure 46. F_1 measure comparison for IMU and 3-axial acceleration sensors fused data. a) Results for each user and b) average F_1 measure.

Finally, the performance of our algorithm was evaluated for each class. Figure 47 shows the average accuracy obtained for each user. One can notice a marked separation between the *sit* and *lie* activities versus *walk* and *stand*. The difficulty in distinguishing *walk* from *stand* stems from the overlapping of data for these two classes. The iterative method produced an average accuracy of 75.4% for the *walk* movement, compared with 82.1% obtained by the supervised method. Similarly, for the *stand* activity, the iterative method produced an average accuracy of 77.06%, which makes a difference of 6.57% with respect of the value obtained by the supervised method (83.63%). However, the classification difference is reduced for the *lie* activity - an average accuracy of 97.57% for the iterative method and 99.18% for the supervised one. When classifying the *sit* activity, the iterative process produced an average accuracy of 91.46% while the supervised method produced 97.27%.

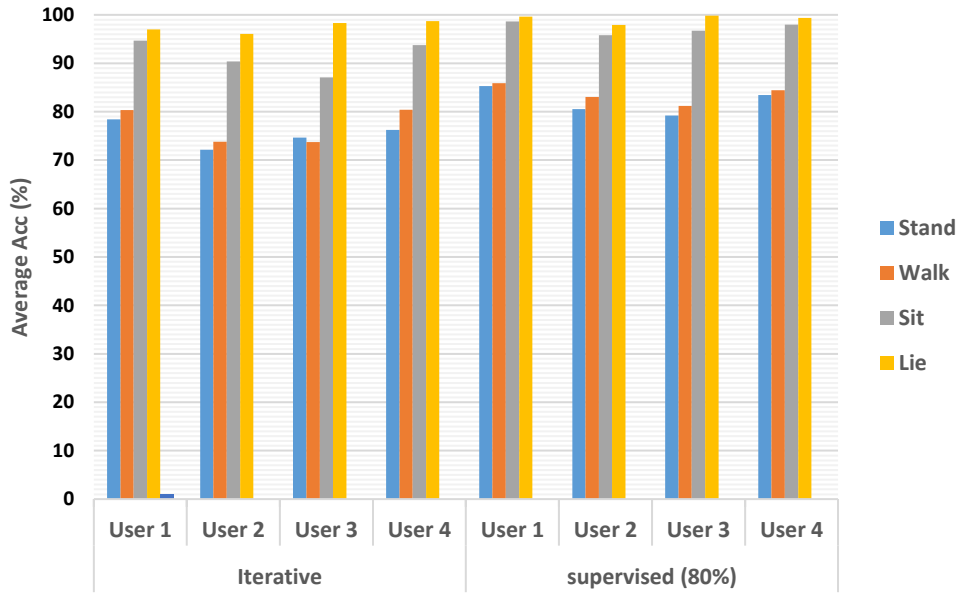


Figure 47. Classification model accuracy comparison between iterative and supervised methods.

Table 26 compares the locomotion classification performance results obtained in [6] with the ones obtained using our method. The results were quantified using the F_1 measure. We focused on those values obtained via SVM classifier, coded as SStar (SVM) and CStar (SVM + 1 NN) in that paper. We also present and compare the results obtained by the

method with the highest classification performance, the 3- Nearest Neighbors (3 NN), reported in the same paper. The results are observed for three users, denoted in [6] by S1, S2 and S3.

Method	User 1 (S1)	User 2 (S2)	User 3 (S3)
SStar [6]	-	0,61	0,68
CStar [6]	-	0,60	0,65
(3NN) [6]	0,85	0,86	0,83
Iterative	0,802	0,66	0,68
Supervised	0,85	0,75	0,72

Table 26 Locomotion classification performance results obtained by [6] and our framework. The results are quantified using the F_1 *measure*

Reviewing results of User 1, our framework demonstrated that using only 6% of the input data, the classification performance was close to 3NN, but higher than those obtained via the SStar and CStar methods. In contrast with Users 2 and 3, our method, using 7% and 5.40% of the input data respectively, shows a higher variance as compared to 3NN, but it is slightly better than SStar and CStar.

7.9 Conclusions

In this section, we have presented the results of the application of our method in the context of recognition of human locomotion using data extracted from 3-axial accelerometers and inertial measurement units (IMU). Our framework takes into account the importance of efficient feature extraction in multimodal sensor data fusion. The deployment of a two-stage filtering process reduced the noise, while providing the classifier with a training set more consistent in terms of the quality of the data.

The algorithm arranges features in pairs to generate the best classification model based on the best feature combination.

The approach, a data-driven iterative learning process reduces the number of samples required for the classification of low-intensity human locomotion activities such as walking, standing, lying and standing, by employing information derived from the

distribution of samples, i.e., data clusters and centroid selection referred in step 5 and 6 of the process presented in section 7.5.

The problem of class overlapping was addressed by the selection of proper values of the parameter C and γ (adopting a kernel RBF) in the SVM classifier in combination with a cross-validation process, i.e., finding a trade-off between bias and variance. Since the classification depends only on training samples near the decision boundary, which is controlled by parameter C (cost), the stronger class membership of each sample in the training set, the more optimal the separation margin (as presented in Chapter V, section 5.4).

The deployment of the extra filtering stage has generated an increase in the average accuracy registered, for example, in the classification models for User 2. Observing the accuracy results in Table 21, we obtained an average accuracy of 61.40%, when we used wavelet filtering. In contrast, we obtained an average accuracy of 67.71%, when we used the two-stage consecutive filtering over the same user -a difference of 6.30%. Similarly, in Table 22 for User2, we obtained an improvement of 12.72%. Finally, in Table 23 for the same user, we obtained an improvement of 12.24%.

Our framework has a potential for extensive application in human locomotion recognition, and particularly in monitoring the elderly, with limited range of motion, or athletes to follow-up their physical performance. This is possible because our framework offers user-dependent data modeling- a valid option for clinical treatment, where diagnostics are customized according to user's needs.

It is worth to mention the superior performance of our method in the recognition of modes of locomotion (F_1) as compared to the values reported in [6], especially taking into account that we used only a fraction of the total input domain. This demonstrates the robustness of our proposed method in terms of the quality of the input data and the effectiveness of the strategy of the training dataset extraction.

CHAPTER VIII: CONCLUSIONS AND FUTURE WORK

A multispectral information collected by the remote sensing equipment with different spatial, temporal, spectral and radiometric characteristics is a typical example when there is a high probability of the occurrence of data multimodality that could be difficult to be modeled by traditional methods. A similar modeling limitation is found in the problem of human locomotion recognition. Indeed, the use of readings obtained from multiple wireless wearable sensors is subject to various issues, such as sensor data alignment, sampling error, data losses, and noise. The method proposed in the thesis aims to automate the process of modeling multimodal systems using an iterative learning machine approach. Our framework focuses on generating datasets associated with each statistical modality of the given dataset, spanning the entire instance space for each modality, while deploying a training dataset extraction process that ensures a high level of robustness to variations in the quality of input data, and consequently leads to an improvement in the data model accuracy.

In regression problems, the proposed data-driven architecture combines unsupervised and supervised learning techniques with classical regression analysis. With a focus on developing precise and robust regression models in regression-type problems, this approach was verified by solving the problem of measuring chl-a concentration in inland waters using remote sensing data. Our method also deals effectively with two practical issues often present in the task of generating accurate and robust models in environment modeling: the scarcity of ground truth information and the absence of a suitable reference label set.

In the problem of using readings obtained from wireless wearable sensors and their application in human locomotion recognition, a two-stage consecutive filtering approach was used to enhance the precision of the acceleration signals. This mechanism provides an effective way to deal with the high data density and non-separable data (overlapping)

because the classification depends only on training samples near to decision boundary, which is controlled by parameter cost (*C*). We improved classification results by including an incremental training dataset, which must be as far as possible to the decision boundary produced by the SVM classifier.

In both problems, the novelty of the proposed method consists in the association of selected data with statistical modality by deploying a process of a consecutive selection of the best candidate samples. In both problems, the multimodality phenomenon occurred as a result of multiple complex interactions between the target variables and their surrounding environments. This required developing a method to work on a multimodal hypothesis instead on focusing in a single hypothesis.

The challenges related to the large percentage of missing data and the noise affecting the measurements were successfully exceeded when applying data fusion with a robust two stage filtering mechanism combined with an iterative learning process. The need for significantly less data entails much shorter computation times. The minimization of the number of samples is an important contribution that allows the user to deal efficiently with an ever-growing number of large datasets. Our framework has demonstrated that introducing a sample selection mechanism it is possible to improve the model accuracy with a two-fold benefit: speed up the training process minimizing the problem of overfitting and reduce complexity of the classifier.

From the practical application perspective, the proposed iterative learning framework offers a powerful solution of a wide spectrum of applications focused on robust modeling of multimodal systems, especially where the model should span the entire input space, instead of providing a piecewise model solution. From the methodological perspective, further work is needed on the process of automating the number of iterations required in both regression and classification problems. The adaptability and flexibility of our solution will be explored in more detail by using benchmark data to compare the SVM classifier, as the optimal sample separation mechanism, with other classifiers.

PUBLICATIONS

Below is a summary of refereed papers published in scientific journals and high-quality international conferences.

1. **Wearable Sensor Data Classification for Human Activity Recognition Based on an Iterative Learning Framework.** Juan Davila, Ana-Maria Cretu, Marek Zaremba Sensors, Vol 17(6), 1287, 2017

The design of multiple applications in human activity recognition, in areas such as healthcare, sports and safety, relies on wearable sensor technologies. However, when making decisions based on the data acquired by such sensors in practical situations, several factors related to sensor data alignment, data losses, and noise among other experimental constrains, deteriorate data quality and model accuracy. To address these issues, this paper presents a data-driven iterative learning framework to classify human locomotion activities such as walk, stand, lie, and sit, extracted from the Opportunity dataset. Data acquired by twelve 3-axial acceleration sensors and seven inertial measurement units are initially de-noised using a two-stage consecutive filtering approach combining a band-pass Finite Impulse Response (FIR) and a wavelet filter. A series of statistical parameters are extracted from the kinematical features, including the principal components and singular value decomposition of roll, pitch, yaw and the norm of the axial components. The novel interactive learning procedure is then applied in order to minimize the number of samples required to classify human locomotion activities. Only those samples that are most distant from the centroids of data clusters, according to a measure presented in the paper, are selected as candidates for the training dataset. The newly built dataset is then used to train an SVM multi-class classifier. The latter will produce the lowest prediction error. The proposed learning framework ensures a high level of robustness to variations in the quality of input data, while only using a much lower number of training samples and therefore a much shorter training time, which is an important consideration given the large size of the dataset.

2. **Iterative Learning for Human Activity Recognition from Wearable Sensor Data** Juan Davila, Ana-Maria Cretu, Marek Zaremba. Proceedings of the 3rd Int. Electronic Conference Sensors Applications, 15–30 November 2016, Sciforum Electronic Conference Series, Vol. 3, S2002, 2016. **ECSA-3 Best Paper Award in 2016**.

Wearable sensor technologies are a key component in the design of applications for human activity recognition, in areas like healthcare, sports and safety. In this paper, we present an iterative learning method to classify human locomotion activities extracted from the Opportunity dataset by implementing a data-driven architecture. Data collected by 12 3D acceleration sensors and 7 inertial measurement units are de-noised using a wavelet filter, prior to the extraction of statistical parameters of kinematical features, such as Principal Components Analysis and Singular Value Decomposition of roll, pitch, yaw and the norm of the axial components. A novel approach is proposed to minimize the number of samples required to classify walk, stand, lie and sit human locomotion activities based on these features. The methodology consists in an iterative extraction of the best candidates for building the training dataset. The best training candidates are selected when the Euclidean distance between an input data and its cluster's centroid is larger than the mean plus the standard deviation of all Euclidean distances between all input data and their corresponding clusters. The resulting datasets are then used to train an SVM multi-class classifier that produces the lowest prediction error. The learning method presented in this paper ensures a high level of robustness to variations in the quality of input data while only using a much lower number of training samples and therefore a much shorter training time, which is an important aspect given the large size of the dataset.

3. **An Iterative Learning Framework for Multimodal Chlorophyll-a Estimation.** Juan Davila and Marek Zaremba, IEEE Transactions on Geoscience and Remote Sensing, Volume 54, Issue: 12, Dec. 2016.

Precise monitoring of the chlorophyll type "a" (chl-a) concentration is critical in determining the level of production of oxygen and, consequently, the health conditions of inland aquatic ecosystems. This paper addresses two important issues in building precise and robust regression models for chl-a concentration from remote sensing (RS) data: the presence of multimodality in the sensor data distribution, and the scarcity of information available to properly label the data. In order to effectively deal with the above issues, we

propose an iterative learning framework (ITEMS – Iterative Transductive Environmental Modeling System) based on the principles of transductive learning that combines data-driven regression-based modeling with an iterative non-linear classification process. The classification procedure, contingent on the maximum margin principle, generates datasets associated with each statistical modality. The classified data are labeled through a process of consecutive selection of the best candidate samples. Different selection mechanisms are discussed. The proposed method was applied in the empirical assessment of chl-a concentration from MERIS and MODIS satellite data and validated by in-situ measurements in Lake Winnipeg in Manitoba, Canada.

4. **Automated modeling of multimodal data processes in remote sensing.** Juan Davila and Marek Zaremba, 15th IFAC/IEEE/IFIP/IFORS Symposium on Information Control Problems in Manufacturing. ISSN 2405-8963, Vol. 48, Issue 3, p 1918-1923, Ottawa, May 2015. **Candidate paper for best student paper award, INCOM 2015**

Automated monitoring of bio-geophysical phenomena, especially those occurring in large areas, requires the use of models obtained from remote sensing data. The interaction of multiple components in the optical data flow and the non-ergodicity of the acquisition process can seriously affect the precision of the models. In order to effectively deal with this situation, we are proposing an iterative semi-supervised learning framework that combines regression analysis leading to the final set of models with an iterative classification process, based on support vector machines (SVM) that generates datasets associated with each statistical modality. This paper presents an application of the proposed method in modeling the concentration of water pollutants, particularly chlorophyll-a, in inland waters using multimodal satellite datasets.

5. **An Integrated Framework for Scarce Data Environment Modeling.** Juan Davila, 15th IFAC/IEEE/IFIP/IFORS Symposium on Information Control Problems in manufacturing, Ottawa, May 2015.

The interaction of multiple variables and their non-ergodicity can seriously affect the precision of models obtained through a data acquisition process. What exacerbates the model development process in practice is the scarcity of both empirical data and the ground-truth information required for statistical learning procedures. In order to effectively

deal with this situation, this article investigates an approach based on an iterative learning framework. This framework employs an iterative classification process, based on support vector machines (SVM) that generates datasets associated with each statistical modality in order to improve the final theoretical model precision.

REFERENCES

[1] F. P. Sarel, Thesis (PDF) on Dataset Shift. In: Land-use classification for optical remote sensing, Electrical, Electronic and Computer Engineering, University of Pretoria, April 2016, pp. 2-7.

[2] L. Gomez-Chova, D. Tuia, G. Moser. Multimodal Classification of Remote Sensing Images: A Review and Future Directions. *Proceeding of IEEE*, August 2015, pp 1-52.

[3] M. Dalla Mura, S. Prasad, F. Pacifici, P. Gamba, Jocelyn Chanussot, et al. Challenges and Opportunities of Multimodality and Data Fusion in Remote Sensing. *Proceedings of the IEEE*, Institute of Electrical and Electronics Engineers, 2015, pp. 1585- 1601.

[4] W.J. Moses et a., "Estimation of chlorophyll-a concentration in case II waters using MODIS and MERIS data — successes and challenges," Environmental Research Letters, vol. 4(4), no. 045005, doi:10.1088/1748-9326/4/4/045005, 2009, pp. 1-8.

[5] S. Patel, H. Park, P. Bonato, L. Chan and M. Rodgers. A review of wearable sensors and systems with application, in rehabilitation. *Journal of Neuro-engineering and Rehabilitation*, 2012, Volume 9, pp. 1-17.

[6] R. Chavarriaga, H. Sagha, A. Calatroni, S. Tejaswi, G. Troster, J. R Millán and D. Roggen. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 2013, Volume 34 (15), pp. 2033–2042.

[7] B. Khaleghi, A. Khamis, F. O. Karray and S. N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 2013, Volume 14(1), pp. 28–44.

[8] H. Qian, Y. Mao, W. Xiang and Z. Wang. Recognition of human activities using SVM multi-class classifier. *Pattern Recognition Letters*. 2010, Volume 31(2), pp. 100-111.

[9] M. Berthold and D. J. Hands. Introduction. In: Intelligent Data Analysis, second edition. *Springer*, Verlang, Berlin, Heidelberg 1999, 2003.

[10] N. Le Roux, Y. Bengio, A. Fitzgibbon. Improving First and Second-Order Methods by Modeling Uncertainty. In: Optimization for Machine Learning. *MIT Press*, 2012, p. 404.

[11] Y. Lou, R. Caruana, J. Gehrke. Intelligible Models for Classification and Regression. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, *ACM*, 2012, pp. 150-158

[12] M. Sugiyama and M. Kawanabe. Introduction and problem formulation. In: machine learning in non-stationary environments. *The MIT Press*. 2012, pp. 3-13.

[13] M. Mohri, A. Rostamizadeh and A. Talwalkar. Introduction and the PAC Learning Framework. In: Foundation of Machine Learning. *The MIT press*. 2012, pp. 1-32

[14] S. Jain and E. Kiner. Iterative learning from texts and counterexamples using additional information. *Journal Machine Learning*. 2011, Volume 84(3), pp. 291-333.

[15] R. Warrier and S. Devasia. Iterative Learning From Novice Human Demonstration for Output Tracking. *IEEE transactions on Human-Machine Systems*. 2016, Volume 46(4), pp. 510-521.

[16] S. Lange and G. Grieser. On the Strength of Incremental Learning. In: algorithmic learning theory. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence). *Springer*, Berlin, Heidelberg. 1999, pp 118- 131.

[17] Y. Xu, A. Fern and S. Yoon. Iterative Learning of Weighted Rule Sets for Greedy Search. *Proceeding of the 20th International Conference on Automated Planning and Scheduling*. 2010, pp 201- 208.

[18] J.C. Davila and M. Zaremba. An Iterative Learning Framework for Multimodal Chlorophyll-a Estimation. *IEEE Transactions on Geoscience and Remote Sensing*. 2016, Volume 54(12), pp. 7299-7308.

[19] Y. Freund and R.E Schapire. A shore Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence*, Volume 14(5), 1999, pp. 771-780.

[20] Y. Wu, K. Chen-Chaun, K. Chang, E. Y. Chang, J.R. Smith. "Optimal Multimodal Fusion for Multimedia Data Analysis", Proceedings of the 12th annual *ACM international conference on multimedia*, New York, October 2004, pp. 572-579.

[21] B.W Silverman, "Using Kernel Density Estimates to Investigate Multimodality", *Journal of the Royal Statistical Society*. Series B (Methodological), 1981, pp. 97-99.

[22] L. Xu, E.J Bedrick, T. Hanson, and C. Restrepo, "A Comparison of Statistical Tools for Identifying Modality in Body Mass Distributions*", Journal of Data Science*, Vol 12, 2014, pp. 175-196.

[23] P. Hall, M. York, "On the Calibration of Silverman's Test for Multimodality", *Statistica Sinica*, Vol 11, 2011, pp. 515-536.

[24] D.W. Müller, G. Sawitski, "Excess Mass Estimates and Tests for Modality", *Journal of the American Statistical Association*, Vol 86, 1991, pp. 738-746.

[25] Z. Abraham, P-N. Tan, "A Semi-Supervised Framework for Simultaneous Classification and Regression of Zero-Inflated Time Series Data with Applications to Precipitation Prediction", *IEEE International Conference on Data Mining Workshops*, December 2009, pp.644-649.

[26] C.E Binding, T.A Greenberg, R.P Bukata, “The MERIS Maximun Chlorophyll Index; its Merits and Limitations for Inland Water Algal Bloom Monitoring”, *Journal of Great Lakes Research Supplement*, Issue 69, 2013, pp. 100-107.

[27] V. M Scholz. Thesis dissertation on approaches to analyze and interpret biological profile data. Max-Planck-Institut f¨ur Molekulare Pflanzenphysiologie, Potsdam University, Postam,Germany, 2006, pp. 15-31

[28] A. Hyvärinen, J. Karhunen, E. Oja. Introduction. In:independent component analysis, *Wiley*, New York, ISBN 978-0-471-40540-5, 2001, pp. 1-11.

[29] WH. Press, SA. Teukolsky, WT. Vetterling, BP. Flannery. "Section 16.1. Gaussian Mixture Models and k-Means Clustering". Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York: *Cambridge University Press*. ISBN 978-0-521-88068-8, 2007, pp. 843-849.

[30] M. Rosenblatt, "Remarks on Some Nonparametric Estimates of a Density Function". *The Annals of Mathematical Statistics*. 27 (3), 1956: 832-837.

[31] J. A. Hartigan, P.M Hartigan. The DIP test of unimodality. *Annals of Statistics*, vol 13, 1985, pp. 70-84.

[32] D. W. Muller and G. Sawitski. Excess mass estimates and tests for modality. *Journal of the American Statistical Association*, Vol 86, 1991, pp. 738-746.

[33] S.B. Kotsiantis, Supervised Machine Learning: A review of Classification Techniques, Department of Computer Science and Technology, University of Peloponnese, Greece. *Informatica*, Vol 31, 2017, pp. 249-268

[34] P. Dorian. The nature of the World and its impact on data preparation. In: Data preparation for data mining. *Academic Press*. San Diego, CA, USA, 1999, pp. 45-87.

[35] J. Osborne. Best practices as you prepare your data collection. In: best practices in Data Cleaning. *Sage Publications*, CA, USA,2011, pp. 17-36

[36] A. Gelman and J. Hill. Missing-data imputation. In: Data Analysis Using Regression and Multilevel/Hierarchical Models. *Cambridge university press*, New York, USA, 2007, pp. 529- 545.

[37] I.H. Written, E. Frank, M. Hall, Algorithms: the basic methods. In: Data mining: practical machine learning tools- 3rd edition, *Elsevier*, MA, USA, 2010, pp. 85.145.

[38] R., Payam, L. Tang, and H. Liu. 2007. On Comparison of Feature Selection Algorithms. In: *proceedings of AAAI Workshop on Evaluation Methods for Machine Learning* II, 34–39.

[39] A. Burkov. Introduction. In: the hundred-page machine learning book. *Andriy Burkov*, ISBN 978-1-9995795-0-0, 2019, pp. 1-7

[40] G Forman and M Scholz. Apples to apples in cross-validation studies: pitfalls in classifier performance measurement. *ACM SIGKDD Explorations*, 2010, 12(1), pp. 49–57.

[41] M. Mohri, A. Rostamizadeh and A. Talwalkar. Introduction and the PAC Learning Framework. In: foundation of machine learning. *The MIT press*. 2012, pp. 1-32.

[42] C. Yao, D. Cai, J. Bu, G. Chen. Pre-training the deep generative models with adaptive hyperparameter optimization. Neurocomputing. *Elsevier*. Vol 247, 2017, pp. 144-155

[43] F. Hutter, L. Kotthoff, J. Vanschoren. Hyperparameter Optimization. In: automated machine learning. Springer Nature Switzerland AG. *Part of Springer Nature*, 2019, pp.3-33

[44] Q. Huang, J. Mao, Y Liu. An Improved Grid Search Algorithm of SVR Parameter Optimization. Communication Technology (ICCT), 2012 *IEEE 14th International Conference on*. May, 2013, pp. 1022- 1026.

[45] J. Bergstra, Y. Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, Vol 13, 2012, pp. 281-305.

[46] S. Gao, W. Cheng. A Partition-Based Random Search for Stochastic Constrained Optimization via Simulation. *IEEE transactions on Automatic Control*, Vol 62, February 2017, pp. 740-752.

[47] E. Brochu, V. M. Cora, N de Freitas. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. *Cornell University Library*, 2010, pp. 1-49.

[48] P. Chapman (NCR), J. Clinton (SPSS), R. Kerber (NCR), T. Khabaza (SPSS), T. Reinartz (DaimlerChrysler), C. Shearer (SPSS) and R. Wirth (DaimlerChrysler). CRSIP-DM 1.0. Step-by-Step datamining guide. CRISP-DM consortium, 2000.

[49] O. Marban, G. Mariscal, J. Segovia, A Data Mining & Knowledge Discovery Process Model, *INTECH Open Science*, ISBN: 978-3-902613-53-0, 2009, pp. 1-18.

[50] D. L. Olson and D. Delen. Data mining process. In: Advanced data mining techniques, *Springer*, Verlag, Berlin Heidelberg, Germany, 2008, pp. 19-34.

[51] T. Mitchell. Machine learning introduction. In: Machine learning, *McGraw Hill*, Redmond, WA, USA, ISBN 0070428077, 1997, pp. 1-19.

[52] S. Russell, P. Norving. Learning from examples. In: artificial intelligence: A Modern Approach. 3rd edition. *Pearson Education limited*, Edingurgh Gate, England, 2014, pp 704-757.

[53] O. Chapell, B. Scholkopf and A. Zien. "A taxonomy for Semi-supervised learning methods", in Semi-Supervised Learning. *The MIT Press*, Cambridge, Massachusetts, London, England, ISBN 0-262-03358-5, 2006, pp. 15-31.

[54] O. D. Lara and M. A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*. Volume 15(3), 2013, pp. 1192-1209.

[55] P. Buhlmann, B. Yu, "Boosting with the $\mathbf{L_2}$ Loss: Regression and Classification", *Journal of the American Statistical Association*, Vol 98, No. 462, June 2003, pp. 324-339.

[56] E. L. Allwien, R. E. Schapire, Y. Singer. "Reducing Multiclass to Binary: A Unifying approach for Margin Classifiers", *Journal of Machine Learning Research*, December 2000, pp. 113-141.

[57] A.J. Ferreira, M.A.T. Figueiredo. Boosting Algorithms: A Review of Methods, Theory, and Applications. In: Zhang C., Ma Y. (eds) Ensemble Machine Learning. *Springer*, Boston, MA, 2012, pp. 35-85.

[58] T. Kanamori, T. Takenouchi, S. Eguchi, N. Murata. The Most Robust Loss Function for Boosting. In: N.R Pal, N. Kasabov, R.K. Mudi, S. Pal, S.K Parui. (eds) Neural Information Processing. ICONIP 2004. Lecture Notes in Computer Science, vol 3316. *Springer*, Berlin, Heidelberg, 2004, pp. 496-501

[59] C.K. Reddy, J-H. Park, "Multi-resolution boosting for classifications and regression problems", Knowledge and Information Systems, November 2011, Vol.29 (2), pp. 435-456.

[60] R.E. Schapire, Y. Freund. Using AdaBoost to minimize training error. In: Boosting: Foundations and Algorithms. *The MIT Press*, Cambridge, Massachusetts, USA, 2012, pp. 53-71.

[61] Wearable Tech Market to Cross 560M Devices Shipped Annually By 2021 http://www.wearabletechworld.com/topics/wearable-tech/articles/418588-wearable-tech-market-cross-560m-devices-shipped-annually.htm
Visited in July 24, 2018).

[62] A. Khatami, S. Mirghasemi, A. Khosravi, C. P. Lim. S. Nahavandi. A new PSO-base approach to flame detection using K-Medoids clustering. Expert Systems with Applications. *Elsevier*. Volume 68, 2017, pp. 69-80.

[63] J. Han, M. Kamber and K. Tung. Spatial clustering methods in data mining: A survey. In J. M. Harvey, & H. Jiawei (Eds.), geographic data mining and knowledge discovery, 2001, pp. 1-29

[64] D.J. Lary, A. H Alavi, A.H. Gandomi, A.L. Walker. Machine learning in geosciences and remote sensing. *Geoscience Frontiers*, ISSN: 1674-9871, Vol 7(1), 2016, pp. 3-10.

[65] L. Chen, K. Yeh, H. Wei, G. Liu. An improved genetic programming to SSM/I estimation typhoon precipitation over ocean. *Hydrological Processes*, 2011, pp. 2573-2583.

[66] L. Chen. A study of applying genetic programming to reservoir trophic state evaluation using remote sensor data. *International Journal of Remote Sensing*, 2003, pp. 2265-2275.

[67] C. Lewkowski, A. Porwal, I. González-Álvarez. Genetic programming applied to base-metal prospectively mapping in the Aravalli Province, *India Geophysical Research Abstracts*, 2010, pp. EGU2010-15171.

[68] P. Rosin, J. Hervas Image thresholding for landslide detection by genetic programming L. Bruzzone, P. Smiths (Eds.), Analysis of Multi-temporal Remote Sensing Images, *World Scientific*, 2002, pp. 65-72.

[69] A. Makkeasorn, N.B. Chang, M. Beaman, C. Wyatt, C. Slater Soil moisture estimation in a semi-arid watershed using RADARSAT-1 satellite imagery and genetic programming, *Water Resources Research*, 2006, pp. 1-15.

[70] X. Shang, & L.A. Chisholm. Classification of Australian native forest species using hyperspectral remote sensing and machine-learning classification algorithms. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol 7(6), doi:10.1109/JSTARS.2013.2282166, 2014, pp. 2481-2489

[71] X. Huang, & J.R. Jensen. A machine-learning approach to automated knowledge-base building for remote sensing image analysis with GIS data. *Photogrammetric engineering and remote sensing*, 1997, pp. 1185-1193.

[72] N-B. Chang, K.Bai, C-F Chen. Integrating multisensor satellite data merging and image reconstruction in support of machine learning for better water quality management. *Journal of environmental management*, 2017, pp. 227-240.

[73] Machine learning in remote sensing data processing. Gustavo Camps-Valls. Image Processing Laboratory (IPL), Universidad de Valencia, Spain http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.475.9806&rep=rep1&type=pdf

Visited in August 8, 2018.

[74] A. Karpatne, I. Ebert-Uphoff, S. Ravela, H. Ali Babaie, and V. Kumar. Machine learning of the geosciences: Challenges and Opportunities. arXiv, *Cornell University Library*. arXiv:1711.04708 [cs.LG], 2017, pp. 1-15

[75] YG Cheng, K. Zhu and Y. Li. "Recognition of human activities using machine learning methods with wearable sensors," *2017 IEEE 7th Annual Computing and Communication Workshop and Conference* (CCWC), Las Vegas, NV, 2017, pp. 1-7.

[76] E.M. Tapia. Using Machine Learning for Real-time Activity Recognition and Estimation of Energy Expenditure. Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning, in partial fulfillment of the requirements for the degree of Doctor of Philosophy. *Massachusetts Institute of Technology*, June 2008.

[77] J. Sunny. Applications and Challenges of Human Activity Recognition using Sensors in a Smart Environment. IJIRST –*International Journal for Innovative Research in Science & Technology*, 2015, pp 50-57.

[78] H. F Nweke, Y.W. Teh, M. Al-garadi, U. R. Alo. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. Expert Systems with Applications, *Elsevier*, Vol 105, September 2018, pp. 233-261.

[79] G. James, D Written. Statistical learning. In: an introduction to statistical learning. Springer, New York, USA, ISBN 978-1-4614-7138-7, 2013, pp. 15- 52.

[80] P. Lorrentz. 2015. Artificial Neural Systems: Principles and Practice. [S.l.]: Bentham Science Publishers, 2015. *eBook Collection* (EBSCOhost), EBSCOhost.
Visted on December 23, 2017.

[81] V. Kaajakari. Accelerometers. In: Practical MEMS. Ville Kaajakari, ISBN 978-0-9822991-0-4, 2009, pp. 33-44.

[82] H. Qian, Y. Mao, W. Xiang and Z. Wang. Recognition of human activities using SVM multi-class classifier. *Pattern Recognition Letters*, 2010**,** pp. 100-111.

[83] D. T. Janzen, A. Fredeen and R. Wheate. Radiometric corrections techniques and accuracy assessment for Landsat data in remoted forested regions. *Canadian Journal of Remote sensing*, 2006, pp.330-340.

[84] H.J. Woerd, M.R. Wernand. True Colour Classification of Natural Waters with Medium-Spectral Resolution Satellites: SeaWiFS, MODIS, MERIS and OLCI. *Sensors*, 2015, pp. 25663-25680.

[85] Live Science Available online: https://www.livescience.com/40102-accelerometers.html. Visited in March 06, 2018.

[86] Engineering 360 https://www.globalspec.com/learnmore/sensors_transducers_detectors/acceleration_vibration_sensing/accelerometers. Visited in March 07, 2018.

[87] Endevco. Practical understanding of key accelerometer specifications. Meggott Smart engineering for extreme environments. https://endevco.com/news/emails/2011_12/tp328.pdf Visited in March 07, 2018.

[88] Analog Devices. http://www.analog.com/en/products/landing-pages/001/accelerometer-specifications-definitions.html Visited in March 07, 2018.

[89] D.A. Stow, A. Hope, et al. Remote Sensing of vegetation and land-cover change in Arctic Tundra Ecosystems, *Remote Sensing of Environment,* Volume 89, Issue 3, 2004, pp. 281-308.

[90] R. Doerffer, H. Schiller. MERIS Regional Coastal and Lake Case 2 Water Project Atmospheric Correction ATBD. GKSS Forschungszentrum Geesthacht GmbH. Doc GKSS-KOF-MERIS-ATBD01, 2008, page 4-42.

[91] Q. Liu, R. Klucik, C. Chen, G. Grant, D. Gallaher. Q. Lv, L. Shang. Unsupervised detection of contextual anomaly in remotely sensed data. Remote Sensing Environment. *Elservier,* vol 202, 2017, pp. 75-87.

[92] L. Qui et al., "Influence of particle composition in remote sensing reflectance and MERIS maximum chlorophyll index algorithm: examples from Taihu Lake and Chaohu Lake," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 6, 2015, pp. 1170-1174.

[93] L. Gomez-Chova, D. Tuia, G. Moser, and G. Camps-Valls. Multimodal classification of remote sensing images: a review and future directions. *Proceedings of the IEEE*, 103(9), 2015, pp. 1560–1584.

[94] C. Pohl, J. L. Van Genderen, "Multisensor image fusion in remote sensing: Concepts methods and applications*", International Journal Remote Sensing*, vol. 19, no. 5, 1998, pp. 823-854.

[95] I. R. Farah. A Multi Views Approach for Remote Sensing Fusion Based on Spectral Spatial and Temporal Information. In Image Fusion, Rijeka, Croatia: InTech, 2011, pp. 43-70.

[96] S. Novoa, et al., "Water quality assessment using satellite-derived chlorophyll-a within the European Directives, in the southeastern Bay of Biscay," Marine Pollution Bulletin, vol. 64, no. 4, 2012, pp. 739-750.

[97] S.I. Allakhverdiev et al., "Redox potentials of primary electron acceptor quinone molecule (Q_a^-) and conserved energetics of photosystem II in cyanobacteria with chlorophyll a and chlorophyll d," *National Academy of Sciences of the United States of America*, vol. 108, no. 19, 2011, pp. 8054-8058.

[98] S. Roy, "HPLC-measured chlorophyll-type pigments during a phytoplankton spring bloom in Bedford Basin (Canada)," *Marine Ecology Progress Series*, vol. 55, 1989, pp. 279-290.

[99] G. Meister and B. A. Franz, "Corrections to the MODIS Agua calibration derived from MODIS Aqua ocean color products," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, No. 14, 2014, pp. 6534- 6541.

[100] Y. Li et al., "Estimation of chlorophyll-a concentration using NIR/Red bands of MERIS and classifications procedure in inland turbid water," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 3, 2012, pp. 3492-3510.

[101] L. Qui et al., "Influence of particle composition in remote sensing reflectance and MERIS maximum chlorophyll index algorithm: examples from Taihu Lake and Chaohu Lake," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 6, 2015, pp. 1170-1174.

[102] W.J. Moses et al., "HICO-based NIR-Red Models for estimating Chlorophyll-a concentration in productive coastal waters," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 4, 2009, pp. 845-849.

[103] J. Chen et al., "A review of some important technical problems of satellite remote sensing of chlorophyll-a concentration in coastal waters," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 5, 2013, pp. 2275-2289.

[104] M. Zhang et al., "A validation study of an improved SWIR iterative atmospheric correction algorithm for MODIS-Aqua measurements in lake Taihu, China," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 8, 2014, pp. 4686-4695.

[105] O'Reilly, et al., "Ocean color chlorophyll algorithms for SeaWiFS," *Journal of Geophysical Research*, vol. 103, no. C11, 2009, pp. 24937- 24953.

[106] V. Barale and M. Gade. Visible & thermal infrared passive/active remote sensing. In: remote sensing of the European seas, Springer Science and Business Media, 2008, pp. 103-108.

[107] O'Reilly et al., "Ocean color chlorophyll-a algorithms for SeaWiFS OC2, and OC4: Version 4" in O'Reilly et al., SeaWiFS Postlaunch Calibration and Validation Analyses, Part 3. NASA Tech. Memo, Vol. 11, *NASA Goddard Space Flight Center*, Greenbelt, Maryland, 2000-206892.

[108] G. Dall'Olmo and A. Gitelson, "Effect of bio-optical parameter variability on the remote estimation of chlorophyll-a concentration in turbid productive waters: experimental results" *Optical Society of America*, vol. 44, no. 3, 2005, pp. 412-422.

[109] A. Gitelson et al., "A simple semi-analytical model for remote estimation of chlorophyll-a in turbid waters: validation," *Remote Sensing Environment*, vol. 112, no. 6, 2008, pp. 3582-3593.

[110] D. Tang, "A global survey of intense surface plankton blooms and floating vegetation using MERIS MCI," in Remote Sensing of the Changing Oceans, *Springer Science & Business Media*, 2011, pp. 99-120.

[111] J. Gower and S. King and P. Goncalves, "Global monitoring of plankton blooms using MERIS MCI" *International Journal of Remote Sensing*, vol. 29, no. 21, 2008, pp. 6209-6216.

[112] V. Barale and M. Gade. Satellite water colour observation in African seas. In Remote Sensing of the African Seas, *Springer Science*, Dordrecht, 2014, pp. 31-54.

[113] R. Doerffer and C. Brockamann, "Approaches and existing algorithms. In: Consensus Case 2 Regional Algorithm Protocols," Brockmann Consult, Report DEL-26(1), 2014, pp. 10-12.

[114] C.E Binding et al., "The MERIS maximun chlorophyll index: its merits and limitations for inland water algal bloom monitoring," *Journal of Great Lakes Research Supplement*, vol. 69, 2013, pp. 100-107.

[115] S.J Palmer et al., "Validation of Envisat MERIS algoriothms for chlorophyll retrieval in a large, turbid and optically-complex shallow lake," *Remote Sensing of Environment*, vol. 157, 2015, pp. 158-169.

[116] M.R. Abbott and R.M. Letelier, "An analysis of chlorophyll fluorescence algorithms for the moderate resolution imaging spectrometer (MODIS)," *Remote Sensing of Environment*, vol. 58, no. 2, 1996, pp. 215-223.

[117] X-G. Xing et al., "An overview of remote sensing of chlorophyll fluorescence," *Ocean Science Journal*, vol. 42, no. 1, 2007, pp. 49-59.

[118] A. Buranapratheprat et al., "MERIS imageries to investigate surface chlorophyll in the upper Gulf of Thailand," *Coastal Marine Science*, vol. 33, no.1, 2009, pp. 22-28.

[119] S. Mishra et al., "Bio-optical inversion in highly turbid and cyanobacteria-dominated waters," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 1, 2014, pp. 375-388.

[120] L. R. Snyder et al. Preparative liquid chromatography. In: introduction to modern liquid chromatography, *John Wiley & Sons*, New York, 2009, pp. 19-83.

[121] G. McCullough, "Chlorophyll Mapping using MODIS/MERIS imagery over Case 2 Waters, Lake Winnipeg," Technical Report, University of Manitoba, 2006.

[122] M. Frank and P. Wolf, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, 1956, pp. 95-110.

[123] B. Schölkopf and A. J. Smola, "Learning with kernels, support vector machines, regularization, optimization, and beyond," *The MIT Press*, United States, 2002, pp. 25-55.

[124] H. He, Y. Ma. Foundations of imbalanced learning. In: Imbalanced Learning. Foundations, algorithms and applications, *John Wile & Sons, Inc*, Hoboken, New Jersey, USA, 2013.

[125] D. Roggen, S. Magnenat, M. Waibel and G. Troster. Wearable Computing: Designing and Sharing Activity-Recognition Systems Across Platforms. *IEEE Robotics and Automation Magazine*.Volume 18(2), 2011, pp. 83-95.

[126] R Chavarriaga, H. Bayati, S.R. Millán. Unsupervised adaptation for acceleration-based activity recognition: robustness to sensor displacement and rotation. *Personal and Ubiquitous computing*. Volume 17, 2013, pp. 479-490.

[127] M. Gjoreski, H. Gjoreski, M. Luštrek and M. Gams. How Accurately Can Your Wrist Device Recognize Daily Activities and Detect Falls?, *Sensors*. Volume 16(6), 2016, pp.1-21.

[128] P. Zappi, D. Roggen, E. Farella, G. Troster and L. Benini. Network-level power-performance trade- off in wearable activity recognition: A dynamic sensor selection approach. *ACM Transactions on Embedded Computing Systems*. 2012, Volume 11(3), article 68, pp.1-30

[129] A complete list with other applications is found at http://www.opportunity-project.eu Visited in March 30, 2017.

[130] D. Roggen, M. Bächlin and J. Schumm. An educational and research kit for activity and context recognition from on-body sensors. *International Conference on Body Sensor Networks*. 2010, pp. 277-282.

[131] F. Taylor. Finite Impulse Response Filter in Digital Filters: Principles and Applications with MATLAB, chapter 6. *Wiley-IEEE press e-book chapters*. 2012, pp. 53-70.

[132] Basics of Instrumentation, Measurement and Analysis, Design of FIR filters. Available on line: http://www.vyssotski.ch/basicsofinstrumentation.html. Visited in March 30, 2017.

[133] H.P. Hsu. Fourier analysis of discrete-time signals. In: Signals and systems, the McGraw-Hill, ISBN 0-07-030641-9, 1995, pp. 227-230.

[134] A. Godfrey, R. Conway, D. Meagher, G. Laighin. Direct measurement of human movement by accelerometry. *Medical Engineering & Physics*. Volume 30, 2008, pp. 1364–1386.

[135] D. Figo, P. C Diniz, D.R. Ferreira and J.M.P Cardoso. Pre-processing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*. Volume 14(7), 2010, pp. 645–662.

[136] F. Levinzon. Fundamental Noise Limit of an IEPE Accelerometer from Piezoelectric Accelerometers with Integral Electronics. *Springer International Publishing Switzerland*. 2015, pp. 107-116.

[137] M. Misiti, Y. Misiti, G. Oppenheim and J-M. Poggi. Guided tour from Wavelet and their applications. *Wiley*. 2007, pp. 1-27.

[138] N. Verma and A.K. Verma. Performance Analysis of Wavelet Thresholding Methods in Denoising of Audio Signals of Some Indian Musical Instruments. *International Journal of Engineering Science and Technology*. Volume 4, 2012, pp. 2047-2052.

[139] B. Vidakovic and P. Mueller. Wavelet for Kids, a Tutorial introduction, Duke University, 1991.

[140] N. K. Al-Qazzaz, S. Ali, S. A. Ahmad, Md. S. Islam and M. I. Ariff. Selection of Mother Wavelets Thresholding Methods in De-noising Multi-channel EEG Signals during Working Memory Task. *IEEE conference on Biomedical Engineering and Science*. 2014, pp. 214- 219.

[141] M. Zhao, C. Fu, L. Ji, K. Tang and M. Zhou. Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with

feature chromosomes. *Expert Systems with Applications*. Volume 38(5), 2011, pp. 5197–5204.

[142] J. Josse and F. Husson. Handling Missing Values in Exploratory Multivariate Data Analysis Methods. *Journal de la Société Française de la Statistique*. Volume 153 (2), 2012, pp. 79-99.

[143] M. Kurucz, A. Benczúr, K. Csalogány. Methods for LargeSscale SVD with Missing Values. *Computer and Automation Research Institute of the Hungarian Academy of Sciences*. 2007, pp. 31-38.

[144] G. R. Naik. "PCA, Kernel PCA and dimensionality reduction" in Advances in principal component Analysis. Springer Nature, Singapore, Singapore, ISBN 978-981-10-6703-7, 2019, pp 19-46.

[145] A. Hyvarinen, J. Karhunen, E. Oja. "What is independent component analysis? In Independent component analysis. John Wiley & Sons, INC. ISBN 0-47-40540-X, 2001, pp. 125- 143.

[146] Quick Introduction to Boosting Algorithms in Machine Learning https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/
Visited in September 15, 2018

[147] Evaluation Metrics for Classification. https://github.com/Donges-Niklas/Classification-Basics/blob/master/Classification_Basics.ipynb
Visited in September 16, 2018

[148] C. D. Mobley, D. Stramski, W. P. Bissett and E. Boss. Optical Modeling of Ocean Water. Is the Case1 Case 2 Classification Still Useful? Oceanography, J*ournal of The Oceanography Society*, Volume 17, Number 2, 2003.

[149] M. J. Mazerolle. Improving data analysis in herpetology: using Akaike's Information Criterion (AIC) to assess the strength of biological hypotheses. Amphibia-Reptilia 27, *Publication of the Societas Europaea Herpetologica*, 2006, pp. 169-180.

[150] L. Ljung. System Identification: Theory for the User, Upper Saddle River, NJ, *Prentice-Hall PTR*, 1999.

[151] G. Box, G. M. Jenkins and G.C.Reinsel. Time Series Analysis: Forecasting and Control. 3rd ed. Englewood Cliffs, NJ, *Prentice Hall*, 1994.

[152] C.-C. Chang and C.-J. Lin, LIBSVM – A library for support vector machines. Available online: http://www.csie.ntu.edu.tw/~cjlin/libsvm/
Visited in October 10, 2016).

[153] M. Sokolova, N. Japkowicz and S. Szpakowicz. Beyond Accuracy, F-score and ROC: a Family of Discriminant Measures for Performance Evaluation. AI 2006: *Advances in Artificial Intelligence*. 2006, pp. 1015-1021.

[154] H. He, Y. Ma, Imbalanced Learning, *Wiley –IEEE Press*, 2013

[155] Box, G. E. P., G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1994.