UNIVERSITÀ DEGLI STUDI
DI GENOVA

DIPARTIMENTO DI INGEGNERIA NAVALE, ELETTRICA, ELETTRONICA E DELLE TELECOMUNICAZIONI

INTERNET AND MULTIMEDIA ENGINEERING

Tesi Magistrale

Ottobre 2021

Supervised learning strategies for wind and rainfall prediction from underwater noise analysis

Emanuele Fava

Relatore: Prof. Ing. Andrea Trucco

Relatrice: Prof.ssa Dott.ssa Annalisa Barla

Correlatore: Prof. Dott. Alessandro Verri

Abstract

The objective of this thesis is to extract meteorological information from ambient ocean noise. In particular, this study aims to use Machine Learning algorithms to obtain rainfall detection and prediction of rainfall intensity and wind speed from underwater acoustic spectra. Those spectra are recorded by a passive aquatic listener (PAL) in the Mediterranean Sea near Genoa, Liguria, over the timeframe of June 2011 to May 2012. The entire dataset is composed of 18193 hourly-averaged acoustic spectra. So the aim of the thesis is to demonstrate the efficiency of Machine Learning techniques in underwater acoustic signal analysis for what concerns rainfall and wind intensity information with respect to state-of-the-art empirical methods. The proposed techniques permit to obtain good results with an RMSE (Root Mean Square Error) value related to rainfall prediction of about 0.48 mm/h, 1.15 m/s for what concern wind and detect precipitations greater than 1 mm/h with 90% probability, keeping the false alarm probability below 0.5%.

1. Introduction 6
1.1. General framework and objectives 6
1.2. State-of-the-art 7
1.3. Available dataset 11
1.4. Contribution summary and innovation 13
1.5. Organization 14
2. Materials and methods 16
2.1. Methods for features extraction 16
2.1.1. Sparse Dictionary Learning 16
2.1.2. MFCC 18
2.1.3. GTCC 21
2.2. Supervised Learning 22
2.2.1. Linear Regression and Least Square 23
2.2.2. Polynomial regression 25
2.2.3. Linear Discriminant Analysis 27
2.2.4. Logistic Regression 28
2.2.5. Support Vector Machine 30
2.2.6. Random Forest 32
2.3. Cross-Validation 34
3. Rainfall Detection 37
3.1 Algorithm analysis 38
3.2. In-depth results analysis and comparisons 41
3.3. Performance of the literature algorithms 48
3.4. Final analysis and comparisons 51
4. Wind prediction 54
4.1 Algorithm analysis 55
4.2. In-depth results analysis and comparisons 60
4.3. Performance of the literature algorithms 67
4.4. Final analysis and comparisons 69
5. Rainfall prediction 71

5.1 Algorithm analysis..... 72
4.2. In-depth results analysis and comparisons..... 75
5.3. Performance of the literature algorithms..... 83
5.4. Final analysis and comparisons..... 85
6. Conclusions..... 87

1. Introduction

1. Introduction

1.1. General framework and objectives

In some cases the role of noise signal analysis could be very useful in situations in which the aim is to understand natural phenomena and their characteristics in a given moment. In today's technological world it's essential to implement solutions that could collect data and analyse them in order to extract information useful to increase the awareness about a given situation of interest or for future research studies. One of those cases is underwater acoustic signals analysis. Underwater acoustic noise could contain information for estimating meteorological parameters, for example rainfall intensity and wind speed. There are many situations, in the marine environment, in which that kind of measurements are essential, for instance for navigation or meteorological monitoring, and could be impossible or not suitable to install and use surface sensors, such as pluviometer and anemometer; this is the case, for example, of harsh environments or dense navigation traffic areas. Also there are many challenging scenarios in which it could be even difficult to use satellite transmission and it is needed to store information on board. A particular example could be the underwater navigations of submarines, in this case the underwater vehicle must need to know the meteorological situation on the surface if it wants to emerge in a safe way. Another example could be the collection of weather informations before a military marine operation in a particular area. A series of sono-buoys could be dropped from helicopters in a given section of sea in order to monitor meteorological conditions and understand well what kind of scenario will be faced. Changing the topic completely, another need could be the monitoring of natural phenomena, especially in relation to climate change and risk prevention. In some situations, data may have to be collected in very harsh environments such as polar waters and the coverage that satellites offer at the higher latitudes of the polar environment is reduced. Weather surveillance radars, operating along the coast, surface rain gauges and anemometers, installed on oceanographic fixed or mobile platforms, also present critical issues that make it difficult to deploy these devices on a large scale. Another solution for meteorological monitoring could be the usage of Synthetic Aperture Radar (SAR). The advantage of using SAR is the ability to work night and day in all meteorological conditions and with high spatial resolution, but costs and complexity can increase a lot. The objective of this thesis is to use underwater acoustic spectra for extracting meteorological informations. As said before, this could be important in scenarios in which collecting weather knowledge could be difficult or not suited with other traditional systems. The approach followed by the thesis is totally different from most state-of-the-art studies. In the thesis Machine Learning algorithms were applied to the dataset available trying to predict rainfall and wind intensity and with the objective of understanding which methods and techniques could better be able to model, from a mathematical point of view, the meteorological phenomena considered. So a deep analysis of different kinds

of preprocessing and Machine Learning algorithms was made in order to find the best ones. This kind of approach is based on the analysis of all the features available that compose the collected spectra. In the past, the approach followed by state-of-the-art studies was very different, in this case in fact the models implemented were based on empirical formulas built on the analysis of few frequencies of the spectra. Underwater acoustic has a long tradition of calculating some physical quantities (for example the speed of sound in water) using empirical equations trying to obtain simple models with a global validity. That's why also for the estimation of wind and rainfall, equations of this type has been proposed, obviously using a limited number of variables (i.e. the values of the spectrum at some frequencies chosen with the utmost care). But the results were not very accurate nor global. In this thesis a Machine Learning regression is proposed to exploit all the spectrum's information. To demonstrate the advantage of this approach, it is necessary to compare the results of the thesis and those of the traditional methods, applied to the same data. The experimental dataset used during the thesis study was collected and provided by the Italian CNR. It is made of 18193 spectra and every collected spectrum is composed of a series of frequency bins that discretize the frequency space in a given range; 64 frequencies in a range from 0.1 to 50 kHz. The underwater acoustic noise was recorded, at a depth of 36 m, by an oceanic recorder, based on Passive Aquatic Listener (PAL) technology. So the inputs available are composed by the dataset of 18193 spectra and by a pair of labels associated to each spectrum. In fact for each spectrum collected, the real measurements of rainfall in mm/h and wind intensity in m/s were collected by two meteorological instruments installed in the same site where spectra were collected. The rainfall intensity was measured with a Vaisala RAINCAP Sensor, composed of a Vaisala Weather Transmitter WXT520, and the hourly average wind speed was computed using measurements from a WindSonic 2D anemometer. Instead the output is composed by the predicted measurements of rainfall and wind intensity, again respectively in mm/h and m/s, for what concerns the prediction analysis part and by the rain/no rain label considering the rainfall detection part. So in summary, this thesis aims to study the possibility of applying Machine Learning algorithms to obtain rainfall detection, so to detect the presence or not of precipitation, and rainfall and wind prediction, thanks to which it is possible to estimate the rainfall intensity measured in mm/h and wind speed in m/s. The idea is to demonstrate and to show that the application of Machine Learning (ML) techniques could produce good results in this particular case, in which acoustic spectra are used as ambient measurements.

1.2. State-of-the-art

During the analysis of the methods proposed by this thesis, comparisons with the previous state-of-the-art algorithms are made. Past methods proposed are substantially based on empirical formulas and analysis of few frequencies. For

example, for what concerns wind prediction in Vagle90 [1] the wind speed is calculated through a model that uses only the sound level at 8 kHz (SPL_8).

$$U = \frac{10^{SPL_8/20}+104.3}{53.91} \quad (1.1)$$

where U is wind speed (m/s) and SPL_8 is the acoustic intensity at 8 kHz (dB relative to 1 μPa^2 Hz^{-1}). SPL8 is divided by 20 because in this equation the wind speed is computed through the peak pressure of the acoustic wave. In Nystuen11 [8], instead, a third-order equation is used. The model is fitted on the data with the objective of reducing the bias and offset of the previous methods.
SPL8 (dB relative to 1 μPa^2 Hz^{-1}) again is used.

$$U = a_3 SPL_8^3 + a_2 SPL_8^2 + a_1 SPL_8 + a_0 \quad (1.2)$$

The coefficients (a_3, a_2, a_1, and a_0) correspond to the values: (0.0005; 20.0310; 0.4904; 2.0871). In Pensieri15 [2] one linear equation is calculated for low SPL_8 value and one quadratic for high SPL_8 value. As for the previous models, sound pressure level at 8 kHz is used.

$$U = 0.1458 \cdot SPL_8 - 3.146 \qquad 30 < SPL_8 < 38,$$
$$U = 0.044642 \cdot SPL_8^2 - 3.2917 \cdot SPL_8 + 63.016$$
$$38 \leq SPL_8 < 60.$$
(1.3)

Finally, in Cazau19 [3] a method based on an outlier-robust nonlinear regression model (O-R regression model) is proposed for wind speed prediction. The regression model has the following second-order polynomial form that follows [2].

$$U = a_2 \cdot SPL_8^2 \mp a_1 \cdot SPL_8 + a_0 \quad (1.4)$$

with (a_2, a_1, a_0) equal to (0.027418, 1.8705, 37.9).
In order to briefly explain outlier removal based on Cook's distance it can be said that in statistics, that kind of distance is commonly used to estimate the influence of a data point when performing a least-squares regression analysis. In a practical ordinary least squares analysis, Cook's distance indicates influential data points that are particularly worth checking for validity. In [3] study, Cook's distance threshold is chosen through an optimization process. This distance is optimized so that the coefficient α in the regression equation $U = \alpha \cdot U_{gt}$ tends to 1, where U_{gt} is the wind speed ground truth.

Considering rainfall prediction, Ma05[4] computes an acoustic rainfall rate algorithm using a simple power-law relationship between sound intensity and rainfall (in mm/h), which can be written in the form of:

$$SPL_5 = \alpha \cdot R^{\beta} \quad (1.5)$$

where SPL_5 (dB relative to 1 $\mu Pa^2 Hz^{-1}$) is the sound intensity at 5 kHz, R is the rainfall intensity (mm/h), and α and β are empirically determined parameters. These parameters are determined at 5 kHz and have values of 42.5 and 15.4, respectively. Taking the 10 $\log_{10}$ of the previous equation, this becomes:

$$R = 10^{\left(\frac{SPL_5 - \alpha}{\beta}\right)} = 10^{\left(\frac{SPL_5 - 42.5}{15.4}\right)} \quad (1.6)$$

Considering rainfall prediction in [2], SPL_5 is analysed and two exponential equations are estimated, one for drizzle and one for rain and heavy rain.

$$R = 10^{\left(\frac{SPL_5 - 64.402}{25}\right)} \quad drizzle,$$
$$R = 10^{\left(\frac{SPL_5 - 65.645}{17.86}\right)} \quad rain\ and\ heavy\ rain.$$
(1.7)

In Nystuen08 [5] the $\log_{10}$ of R (the rainfall rate in mm/h) is estimated as a linear equation:

$$log_{10}(R) = b_0 SPL_5 + b_1 \quad (1.8)$$

Again SPL_5, the sound level at 5 kHz (dB relative to 1 μPa^2 Hz^{-1}), is used for the prediction. The values of the coefficients (b_0 and b_1) are empirically calculated: (0.0325, -1.4416), respectively. In a completely different way from the previous methods, in Taylor20 [6] Machine Learning methods to predict rainfall intensity and wind speed are proposed. CatBoost and Random Forest models are used for this purpose and are applied to hourly-averaged acoustic spectra. For what concerns rainfall detection in [1], the spectral slopes between 3 and 8 kHz and between 3 and 19.5 kHz are compared to specific thresholds to achieve an indication of the precipitation presence.

$$S(19.5) - S(3) > -13.25\ OR\ S(8) - S(3)\ >\ -6.82$$
(1.9)

Where the notation S(f_k) is introduced to indicate the sound spectral level of underwater noise, measured in dB re 1 $\mu Pa^2 Hz^{-1}$, at the frequency f_k expressed in kHz. In [4] rainfall is detected if at least one of the following three conditions is verified. The third condition being specific for drizzle:

$$S(21) + 2.35\ S(5.4)\ >\ 194$$
$$S(21)\ >\ 48\ AND\ S(5.4)\ >\ 53$$
$$S(21) >\ 44\ AND\ S(21) - 0.7\ S(8.3)\ >\ 14$$
(1.10)

In Nystuen15[7] and Nystuen11[8] rainfall is detected if at least one of the following four conditions is verified. The third condition being specific for drizzle and the fourth for rain with high wind:

$$S(20) - 0.75\ S(5)\ >\ 5\ AND\ S(5)\ \leq\ 70$$

$$S(8)\ >\ 60\ AND\ Q(2,8) > \theta\ AND\ S(20)\ >\ 45$$

$$S(8) <\ 50\ AND\ Q(8,15) > -5\ AND\ S(20) >\ 35\ AND$$
$$S(20) >\ 0.9\ S(8)$$

$$\{S(20) + 0.1144\ S^2(8) - 12.728\ S(8) > -307\ AND\ Q(2.8) > \theta$$
$$AND\ S(20) + 0.1\ S^2(8) - 11.5\ S(8) < -281\ AND$$
$$51 < S(8) < 64\}$$
(1.11)

where Q(f_1,f_2) is the spectral slope, in dB/decade, between the frequencies f_1 and f_2 (expressed in kHz):

$$Q(f_1, f_2) = \frac{S(f_1) - S(f_2)}{log_{10}(f_1) - log_{10}(f_2)} \quad (1.12)$$

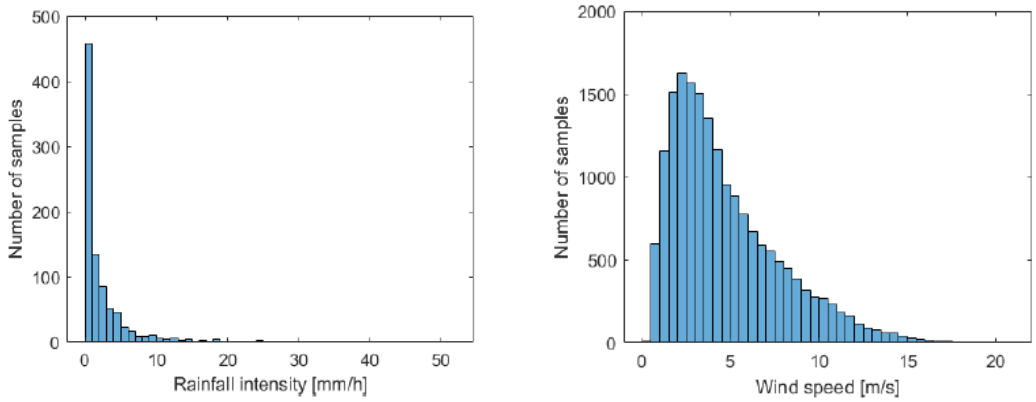
θ = –18 dB/decade in [8] and θ = –13 dB/decade in [7].

In [6] for rainfall detection supervised models are applied to hourly-averaged acoustic spectra, extending the analysis to all the frequency bins instead of only few frequencies and slopes. For the detection task, a binary classifier is built through the CatBoost algorithm, setting the lower bound for rainfall intensity equal to 1 mm/h.

1.3. Available dataset

The material used for the development of the thesis contribution is composed by a dataset of underwater acoustic spectra and rainfall and wind intensity at sea surface collected from 17 June 2011 to 6 September 2013 (with a few breaks, approximately 1.5 months overall) by apposite sensors installed on the meteo-oceanographic observatory W1M3A, moored on a deep-sea bed of 1,200 m, about 80 km off the Ligurian coast, in the northwestern part of the Mediterranean Sea. The rainfall intensity was measured with a Vaisala RAINCAP Sensor, composed of a Vaisala Weather Transmitter WXT520, placed on the upper part of the buoy trellis, at about 10 m. The hourly average wind speed was computed using measurements from a WindSonic 2D anemometer installed on the same trellis on the observatory at 10 m above sea level. The underwater acoustic noise was acquired by a dedicated oceanic recorder, based on Passive Aquatic Listener (PAL) technology, clamped to the body of the platform at a depth of 36 m. This device is designed to operate unattended at sea for a long period of time powered by an internal battery, and to acquire an average of seven acoustic noise snapshots per hour. Each snapshot consists of a time series of 4.5 s, sampled at 100 kHz, which is processed on board to obtain a spectrum composed of 64 frequency bins, with a resolution of 0.2 kHz from 0.1 to 3 kHz and 1 kHz from 3 to 50 kHz. The spectra of the snapshots acquired in one hour (at an average interval of about 9 minutes from each other) were averaged, producing a mean spectrum that is included in the acoustic dataset used in this thesis for prediction and detection. In the entire period of operation, 18193 hourly-averaged acoustic spectra were collected and are available for processing. The rain gauge measured precipitation greater than 0.1 mm/h in 876 of the 18,193 hours considered. The maximum rainfall intensity measured was 51.5 mm/h. and the distribution of the observed intensities is shown in Fig. 1.1(a). The wind speed is ranged between 0.4 and 20.7 m/s and its distribution shown in Fig. 1.1(b). Finally, the tracks of the Automatic Identification System (AIS) used on ships reveal how many of them transited near the buoy in the period of data acquisition. Considering a circle with a radius of 5 km, centered at the position of the buoy, the number of hours in which at least one ship crossed the circle is 1,999, of which 78 are characterized by the presence of rain. Additionally, Fig. 1.2 shows the average spectra

of the underwater noise measurements grouped per Beaufort wind class. The variation of some spectral features as the wind speed increases is evident. [11]



(a) (b)

Fig. 1.1 Distribution of rainfall intensity (a) and wind speed (b) in the period from 17 June 2011 to 10 October 2012.

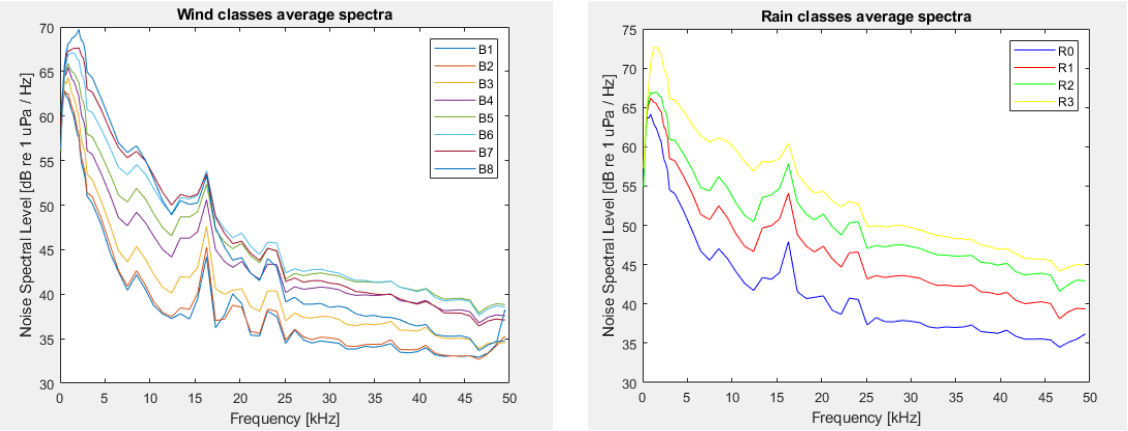


Fig. 1.2 Average spectra of the underwater noise present in the acoustic dataset. The grouping of the spectra is based on the wind classes (left) and rain classes (right).

1.4. Contribution summary and innovation

So, in summary, the study carried out by this thesis was composed by a first part of analysis of the spectra and of the data available. The description of the features and the comprehension of what are the peculiarities of the data are essential for proceeding into the next phases. Then, another important step was the features extraction thanks to algorithms such as Dictionary Learning or Mel-Frequency Cepstral Coefficients (MFCCs) and Gammatone Cepstral Coefficients (GTCCs) processings. So the aim of the initial part is to understand what are the features that can be used for the supervised learning analysis and what could be the ones more suitable for the objective of the thesis, in order to obtain the best performances. Then there is the first supervised learning part in which the objective is to do rainfall detection. The spectra are divided in 2 categories identified by two labels: rain and not rain. The 64 frequency bins are given in input to some Machine Learning algorithms in order to perform detection. The results of the various algorithms are compared between them and between the state-of-the-art methods. In the rainfall detection case the spectra are given in input without any kind of preprocessing on the values. Then, there is the part of rainfall and wind prediction. Again supervised methods are used, but in this case the objective is to make regression and to obtain as output rainfall intensity values in mm/h and wind speed in m/s. So the labels are the real values themselves in mm/h for rain and m/s for wind corresponding to each spectrum. In this case the idea is to give as input to the supervised techniques spectra processed in three different ways. In the first case the preprocessing is done using a Dictionary Learning algorithm, in the second case with MFCC processing and finally with GTCC processing. Then polynomial regression (a form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modelled as an n degree polynomial in x) is used and the results are compared between them and with state-of-the-art algorithms. The great innovations carried out by this thesis is the idea of using Machine Learning algorithms to do prediction of rainfall and wind values and rainfall detection, instead of using previous empirical algorithms based on signal analysis. As said before, the previous methods for detection and prediction in the state-of-the-art are based on the analysis of few frequencies and on the estimation of empirical formulas. In ML cases all the frequencies are taken in consideration together and the potential of this kind of analysis is greater because it is not limited to only 1 or 2 features, so a more complete model could be fitted and could better be able to describe the phenomena in analysis. Using a pool of information on the dataset more complete, the model learned will be obviously more precise and give better results. Also another objective of the thesis is to show that the use of averaged spectra could give good results. In fact every spectrum that will be used in this study is calculated as

the hourly average of a series of short-terms spectra (that can be also called ‘instantaneous’ and indicate data gathered over some minutes) acquired in one hour at an average interval of about 9 minutes from each other. So another objective of the thesis is to demonstrate that hourly-averaged spectra can be used with Machine Learning techniques and can give better performances than those achieved by previous empirical methods fed by short-term data. The usage of hourly averaged spectra could be useful in situations in which there is the need of mitigating extra noise non related to rainfall and wind (for example due to ships passages) or in applications in which it is important to reduce data transmission or also on board memory occupancy. In fact, in some scenarios it is difficult to use satellite transmission continuously and, in some cases, data have to be stored on board.

1.5. Organization

The thesis initially describes, in Chapter 2, the various methods used, divided in categories. The first category contains the methods for preprocessing and so Mel-Frequency Cepstrum Coefficients (MFCCs), Gammatone Cepstral Coefficients (GTCC) and Dictionary Learning (DL). Then supervised algorithms are described. First, the supervised learning methods analysed are the ones used for rainfall detection: Linear Discriminant Analysis, Logistic Regression, Support Vector Machine and Random Forest. Then, the techniques used for rain and wind prediction are described: Linear Regression and Polynomial regression. At the end of this part the thesis will focus the attention on the results and on the proceeding used for obtaining them. For what concerns the results, they will be organized in: rainfall detection results (Chapter 3), wind speed prediction results (Chapter 4) and rainfall intensity prediction results (Chapter 5). All of those three chapters will be divided in: algorithm analysis, in-depth results analysis and comparisons, performance of the literature algorithms and final analysis and comparisons. So results are accompanied by a depth comparison with state-of-the-art methods. Finally, there will be conclusions and possible future research in Chapter 6.

2. Materials and methods

2. Materials and methods

2.1. Methods for features extraction

The feature extraction is the transformation (linear or nonlinear) of the original feature space in n dimensions in a space of different dimension m, usually lower than the original one. One of the objectives of reducing the dimension of the original features could be the reduction of the computational effort. When the number n of the features is very large, the model used for regression or classification could have some issues because of the dimensionality of the problem ("curse of dimensionality"). Increasing n, the computational complexity of a model increases and so this involves an increasing of computation time and sometimes also of memory occupation. Obviously the disadvantage is that sometimes reducing the dimension of the feature space involves loss of information. Then another motivation to change feature space could be the use of another type of features more effective for what concerns the interpretation of data and phenomena. For example in some cases the use of biologically inspired features could be very useful for doing automatic speech recognition tasks. This is the objective of using MFCCs and GTCCs instead of the original frequency bins features. Considering the use of Dictionary Learning the advantage could be the decreasing of the computational effort but at the same time the compression of the initial information could gain the performances of the models analysed.

2.1.1. Sparse Dictionary Learning

Sparse coding can be defined as a representation learning method. It's objective is to obtain a sparse representation of the input data as a linear combination of basic elements. The basic elements that have to be computed are called *atoms* and they compose a *dictionary*. Atoms in the dictionary are not required to be orthogonal. Given the input dataset X = [x_1 , . . . , x_k] , $x_i \in R^d$ we wish to find a dictionary D $\in R^{d\,x\,n}$: D = [d_1 , . . . , d_n] and a representation R = [r_1 , . . . , r_k] ,$r_i \in R^n$ such that both

$\left||X - DR|\right|_F^2$ is minimized and the representations r_i are sparse enough. This can be formulated as the following optimization problem:

$$\underset{D \in C, r_i \in R^n}{argmin} \sum_{i=1}^{K} \|x_i - Dr_i\|_2^2 + \alpha \|r_i\|_0, where\ C \equiv \{D \in R^{dxn} :\ \|d_i\|_2 \leq 1\ \forall i = 1, \ldots., n\}, \alpha > 0$$

(2.1)

C is required to constrain D in order to not have atoms with arbitrarily high values allowing low (but non-zero) values of r_i. α is a parameter used to control the trade off between the sparsity and the minimization error. In general the minimization problem described is not convex because of the ℓ_0-"norm" and it is an NP-hard problem. If n<d, the dictionary is defined "undercomplete", if n>d it is considered "overcomplete". Usually the complete dictionary case n=d does not provide any improvement and so it isn't considered. Undercomplete dictionaries case represent a situation in which the new input data lies in a lower-dimensional space and so it can be an important tool for dimensionality reduction. In this specific case dimensionality reduction based on dictionary representation can be utilized for particular tasks such as data analysis or classification. Instead, considering the overcomplete case, it is possible to obtain redundant atoms and multiple representations of the same signal but also an improvement in sparsity and flexibility of the representation.

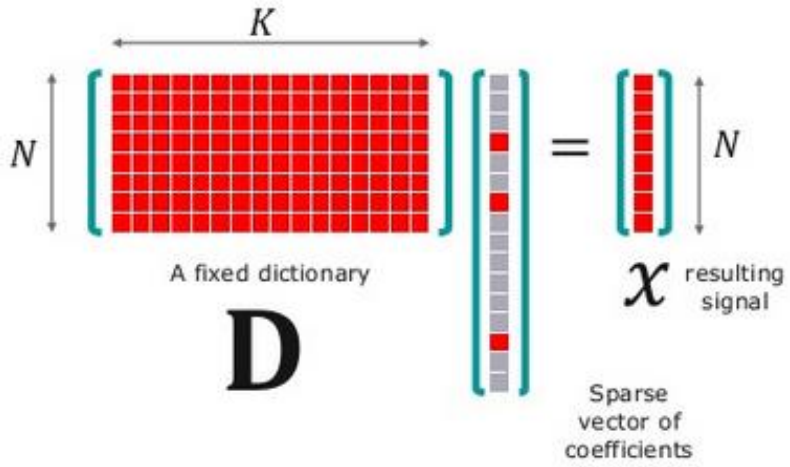


Fig. 2.1 Graphical example of how a signal *x* can be decomposed using matrix D and the correspondent vector of coefficients

2.1.2. MFCC

Mel Frequency Cepstral Coefficents (MFCCs) are a kind of feature derived from short-term power spectrum of a sound and are computed using linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency. The MFCC feature extraction technique includes different phases: windowing the signal, applying the DFT, taking the log of the magnitude, warping the frequencies on a Mel scale and finally applying the inverse DCT. In general this kind of coefficients are used in speech recognition.

Frame blocking and windowing

Consider the case in which the signal is slowly time-varying or quasi-stationary. For stable acoustic characteristics, for example for speech, signal needs to be examined over a sufficiently short period of time. Therefore, analysis must always be carried out on short segments across which the signal is assumed to be stationary. Short-term spectral measurements are typically carried out over 20 ms windows, and advanced every 10 ms. Advancing the time window every 10 ms enables the temporal characteristics of individual sounds to be tracked, and the 20 ms analysis window is usually sufficient to provide good spectral resolution of these sounds, and at the same time short enough to resolve significant temporal characteristics. The purpose of the overlapping analysis is that each sound of the input sequence would be approximately centered at some frame. On each frame, a window is applied to taper the signal towards the frame boundaries. Generally, Hanning or Hamming windows are used. This is done to enhance the harmonics, smooth the edges, and to reduce the edge effect while taking the DFT on the signal.

DFT spectrum

Each windowed frame is converted into magnitude spectrum by applying DFT.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}}; \qquad 0 \leq k \leq N-1 \quad (2.2)$$

where N is the number of points used to compute the DFT.

Mel spectrum

Mel spectrum is computed by passing the Fourier transformed signal through a set of band-pass filters known as Mel-filter bank. A Mel is a unit of measure based on the human ears perceived frequency. It does not correspond linearly to the physical frequency of the tone, as the human auditory system apparently does not perceive pitch linearly. The Mel scale is approximately a linear frequency spacing below 1 kHz and a logarithmic spacing above 1 kHz.

The approximation of Mel from physical frequency can be expressed as

$$f_{Mel} = 2595 log_{10}(1 + \frac{f}{700}) \qquad (2.3)$$

where f denotes the physical frequency in Hz, and f_{Mel} denotes the perceived frequency. Filter banks can be implemented in both time domain and frequency domain. For MFCC computation, filter banks are generally implemented in the frequency domain. The center frequencies of the filters are normally evenly spaced on the frequency axis. However, in order to mimic the human ears perception, the warped axis, according to the nonlinear function given in Eq. 2.3 above, is implemented. The most commonly used filter shaper is triangular, and in some cases the Hanning filter can be found. The triangular filter banks with Mel frequency warping is given in Fig. 2.2

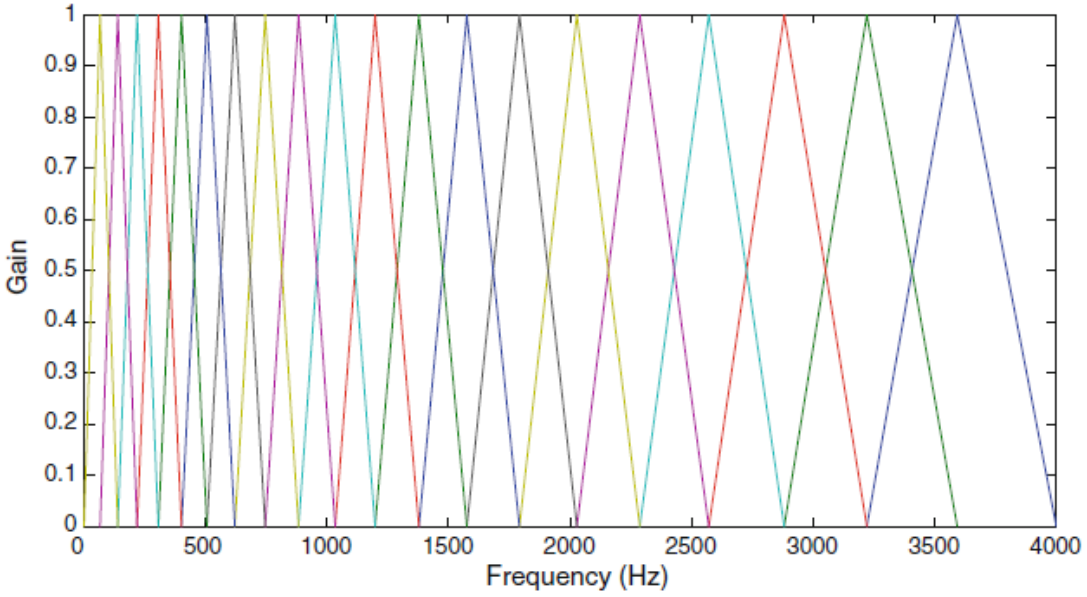


Fig. 2.2 Example of triangular filter banks used in MFCC analysis [13]

The Mel spectrum of the magnitude spectrum X(k) is computed by multiplying the magnitude spectrum by each of the triangular Mel weighting filters.

$$s(m) = \sum_{k=0}^{N-1}[|X(k)|^2 H_m(k)]\,; \qquad 0 \le m \le M-1 \qquad (2.4)$$

where M is the total number of triangular Mel weighting filters. H_m(k) is the weight given to the k-th energy spectrum bin contributing to the m-th output band and is expressed as:

$$|H_m(k)| = \begin{cases} 0, & k < f(m-1) \\ \dfrac{2\big(k - f(m-1)\big)}{f(m) - f(m-1)}, & f(m-1) \le k \le f(m) \\ \dfrac{2(f(m+1) - k)}{f(m+1) - f(m)}, & f(m) < k \le f(m+1) \\ 0, & k > f(m+1) \end{cases}$$

with m ranging from 0 to M-1

(2.5)

Discrete cosine transform (DCT)

The DCT is applied to the transformed Mel frequency coefficients to produce a set of cepstral coefficients. Prior to computing DCT, the Mel spectrum is usually represented on a log scale. This results in a signal in the cepstral domain with a quefrency peak corresponding to the pitch of the signal and a number of formants representing low quefrency peaks. Since most of the signal information is represented by the first few MFCC coefficients, the system can be made robust by extracting only those coefficients ignoring or truncating higher order DCT components. Finally, MFCC is calculated as:

$$c(n) = \sum_{m=0}^{M-1} log_{10}(s(m)) cos\left(\frac{\pi n(m-0.5)}{M}\right)\,; \qquad (2.6)$$

where c(n) are the cepstral coefficients, and C is the number of MFCCs. Traditional MFCC systems use only 8–13 cepstral coefficients. [13]

2.1.3. GTCC

Gammatone cepstral coefficient computation process is a biologically inspired modification of MFCC. The two processes are very similar. The signal is first windowed into short frames. Subsequently, the GT filter bank is applied to the signal's discrete Fourier transform (DFT) and finally the log function and the discrete cosine transform (DCT) are applied. The computational cost is almost equal to the MFCC case. The gammatone filter-bank aims to model the frequency analysis of the cochlea in the inner ear and so to mimic the structure of the peripheral auditory processing stage. The idea is to simulate the motion of the basilar membrane within the cochlea as a function of time, the output of each filter is the frequency response of the basilar membrane at a single place. The energies of the filter outputs is a sort of biologically driven version of the power spectrum used to compute the new features.

The gammatone function is defined in time domain using the g(t) impulse response:

$$g(t) = at^{n-1}cos(2\pi ft + \varphi)e^{-2\pi bt} \quad (2.7)$$

where n is the order of the filter which largely determines the slope of the filter's skirts; b is the bandwidth of the filter and largely determines the duration of the impulse response; a is the amplitude; f is the filter centre frequency; Φ is the phase. Usually, in order to provide a satisfactory fit to the human auditory filter shapes, the impulse response of the gammatone function of order 4 can be used, derived by Patterson and Moore (1986). Glasberg and Moore (1990) have summarized human data on the equivalent rectangular bandwidth (ERB) of the auditory filter with the function:

$$ERB = 24.7(4.37 \cdot 10^{-3}f + 1) \quad (2.8)$$

That is the so-called equivalent rectangular bandwidth (ERB), a measure used in psychoacoustics, with the aim of approximating the bandwidths of the filters in human hearing. The gammatone filter bank is defined in such a way that the filter center frequencies are distributed across frequency in proportion to their bandwidth, known as the ERB scale. The ERB scale is approximately logarithmic, on which the filter center frequencies are equally spaced and so each filter has constant unitary bandwidth on an ERB frequency scale. [14]

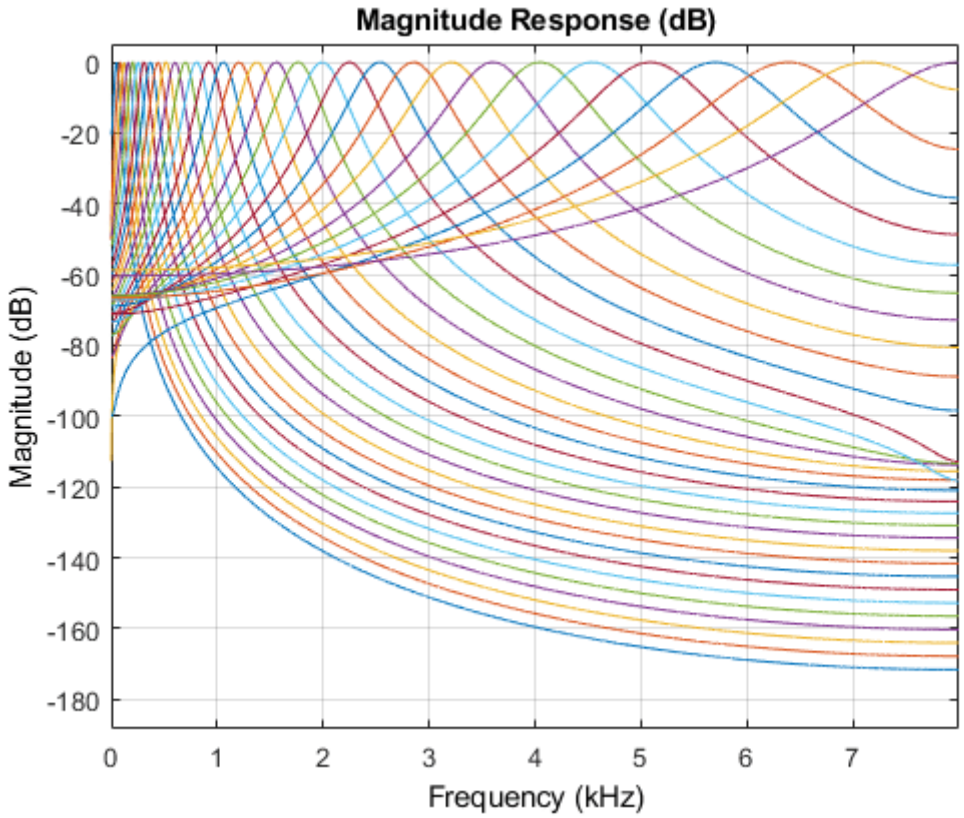


Fig. 2.3 Example of filter banks used in GTCC analysis

2.2. Supervised Learning

Supervised learning is a particular category of Machine Learning. It is characterized by the use of labeled datasets to train the algorithms with the objective of classifying data or predict outcomes accurately. Supervised learning uses a training set, that includes inputs and correct outputs, to fit models. Then the algorithm measures precision and accuracy in prediction of the learned model using a test set. Supervised learning is composed of two types of problems, classification and regression:

- **Classification** has the objective of dividing test data into classes. It recognizes patterns in the dataset and tries to draw some conclusions on how those entities

should be labeled or defined. Labels are discrete numbers, corresponding classes or categories. Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbor, and random forest

- **Regression** is used to understand the relationship between dependent and independent variables. Labels are continuous numbers, such as measurements values. Linear regression, logistic regression, and polynomial regression are popular regression algorithms.

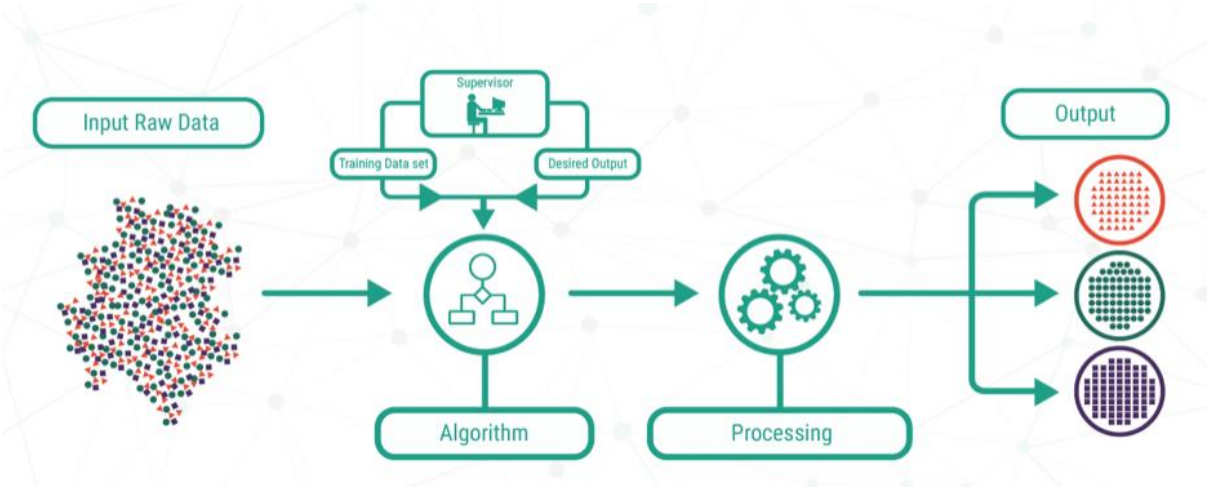


Fig. 2.4 Scheme representing classical data processing, in which supervised learning algorithms are used

2.2.1. Linear Regression and Least Square

Linear regression is one of the simplest methods for supervised learning. Given a vector of inputs $X^T = (X_1, X_2, \ldots, X_p)$, it predict the output Y via the model

$$\hat{Y} = \widehat{\beta_0} + \sum_{j=1}^{p} X_j \widehat{\beta_j}$$

(2.9)

The term $\widehat{\beta_0}$ is the intercept, also known as the bias in Machine Learning. Often it is convenient to include the constant variable 1 in X, include $\hat{\beta}_0$ in the vector of coefficients $\hat{\beta}$, and then write the linear model in vector form as an inner product

$$\hat{Y} = X^T\hat{\beta}$$

(2.10)

In the (p + 1)-dimensional input–output space, (X, $\hat{Y}$) represents a hyperplane. The relationship can be represented as a function over the p-dimensional input space, $f(X)$= X^Tβ is linear, and the gradient $f'(X) = \beta$ is a vector in input space that points in the steepest uphill direction. In order to fit the linear model using a training set there are several methods and the most popular is the least squares method. In this case, the coefficients β is calculated in order to minimize the, so called, residual sum of squares

$$RSS(\beta) = \sum_{i=1}^{N}(y_i - x_i^T\beta)^2 \quad (2.11)$$

RSS(β) is a quadratic function of beta. Its minimum always exists, but may not be unique. It can be written

$$RSS(\beta) = (y - X\beta)^T(y - X\beta) \ (2.12)$$

where X is an N × p matrix with each row an input vector, and y is an N-vector of the outputs in the training set. Differentiating w.r.t. β we get the normal equations

$$X^T(y - X\beta) = 0 \quad (2.13)$$

If X^TX is nonsingular, then the unique solution is given by

$$\hat{\beta} = (X^TX)^{-1}X^Ty \quad (2.14)$$

and the fitted value at the i-th input x_i is $\hat{y}_i$ = $\hat{y}(x_i)$ =$x_i^T\hat{\beta}$. At an arbitrary input x_0 the prediction is $\hat{y}(x_0)$ = $x_0^T\hat{\beta}$. The entire fitted surface is characterized by the p parameters $\hat{\beta}$. Intuitively, it seems that we do not need a very large data set to fit such a model. [12]

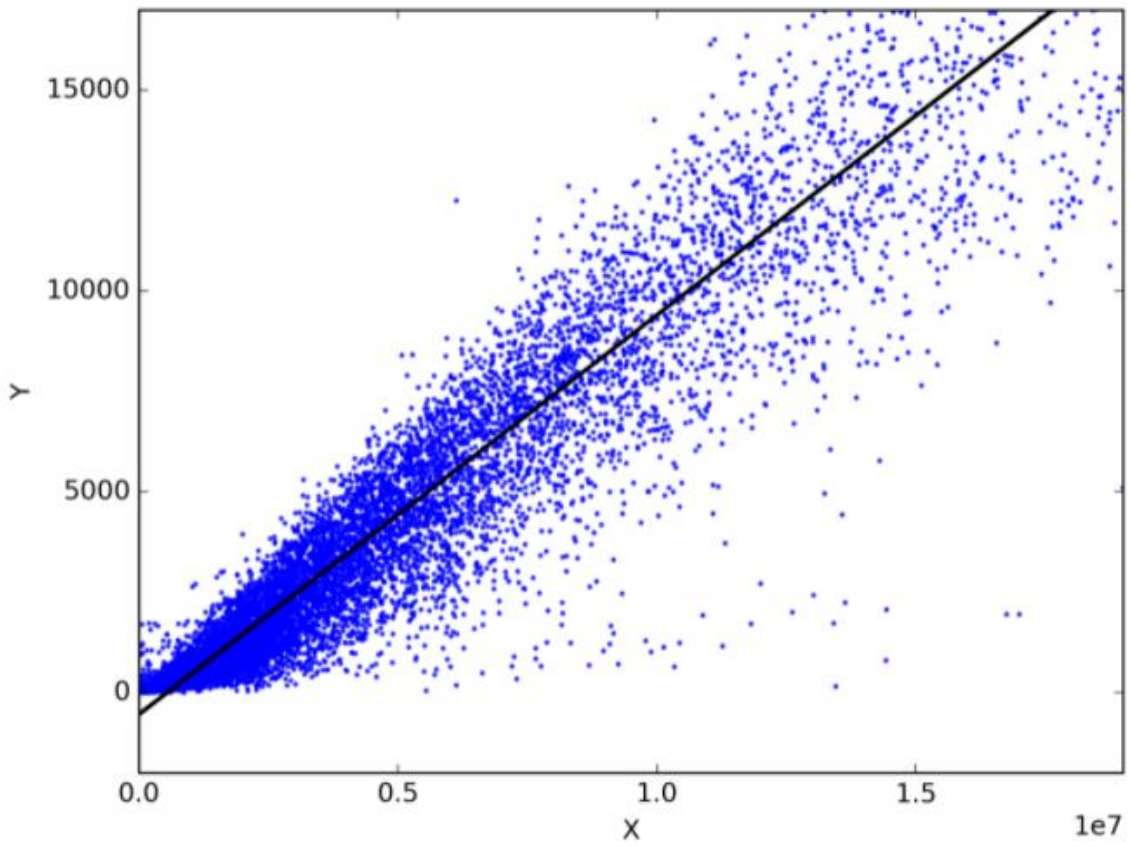


Fig. 2.5 Example of linear regression

2.2.2. Polynomial regression

In the case of polynomial regression most of the considerations made in Linear Regression are still valid. In fact a model is considered linear if it is linear in parameters. For example the models

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$
(2.15)

and

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1\ x_2 + \varepsilon$$
(2.16)

are also linear models. Polynomial models aim to describe situations in which the relationship between dependent and independent variables is curvilinear or to

approximate a complex nonlinear relationship. Considering Polynomial models in one variable of the k-th order we have

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_k x^k + \varepsilon \qquad (2.17)$$

If $x_j = x^j, j = 1,2,\ldots,k$ then the model is multiple linear regressions model in k independent variables $x_{1,} x_{2,} \ldots, x_k$. The linear regression model $y = X\beta + \varepsilon$ can be adapted and used for representing the polynomial regression model. As said before, the techniques for fitting linear regression models can be used for fitting the polynomial regression model.

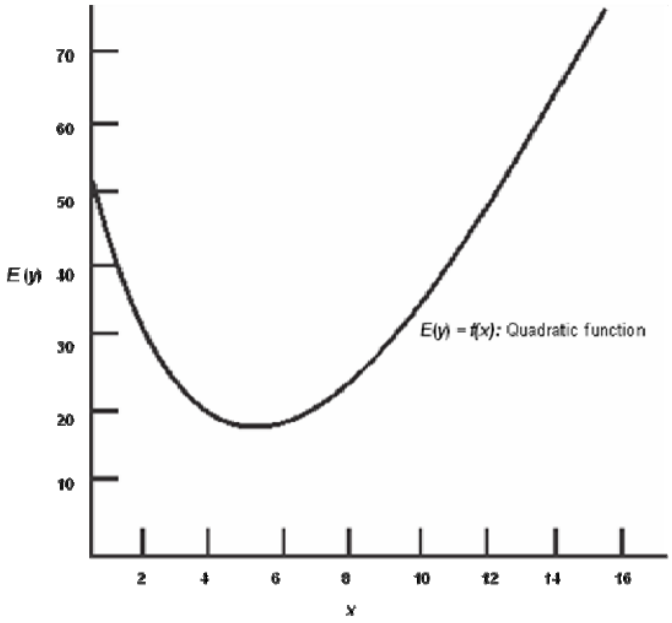


Fig. 2.6 Example of quadratic function [15]

A model is 'hierarchical' if it contains the terms x, x^2, x^3, x^3 etc. in a hierarchy. For example, the model $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \varepsilon$ is hierarchical as it contains all the terms up to order four. The model $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_4 x^4 + \varepsilon$ is not hierarchical as it does not contain the term of x^3. It is expected that all polynomial models should have this property because only hierarchical models are invariant under linear transformation. In some cases, the need for the model may be more complex. For example, the model $y = \beta_0 + \beta_1 x_1 + \beta_{12} x_1 x_2 + \varepsilon$ uses a two-factor interaction, provided by the cross-product term. [15]

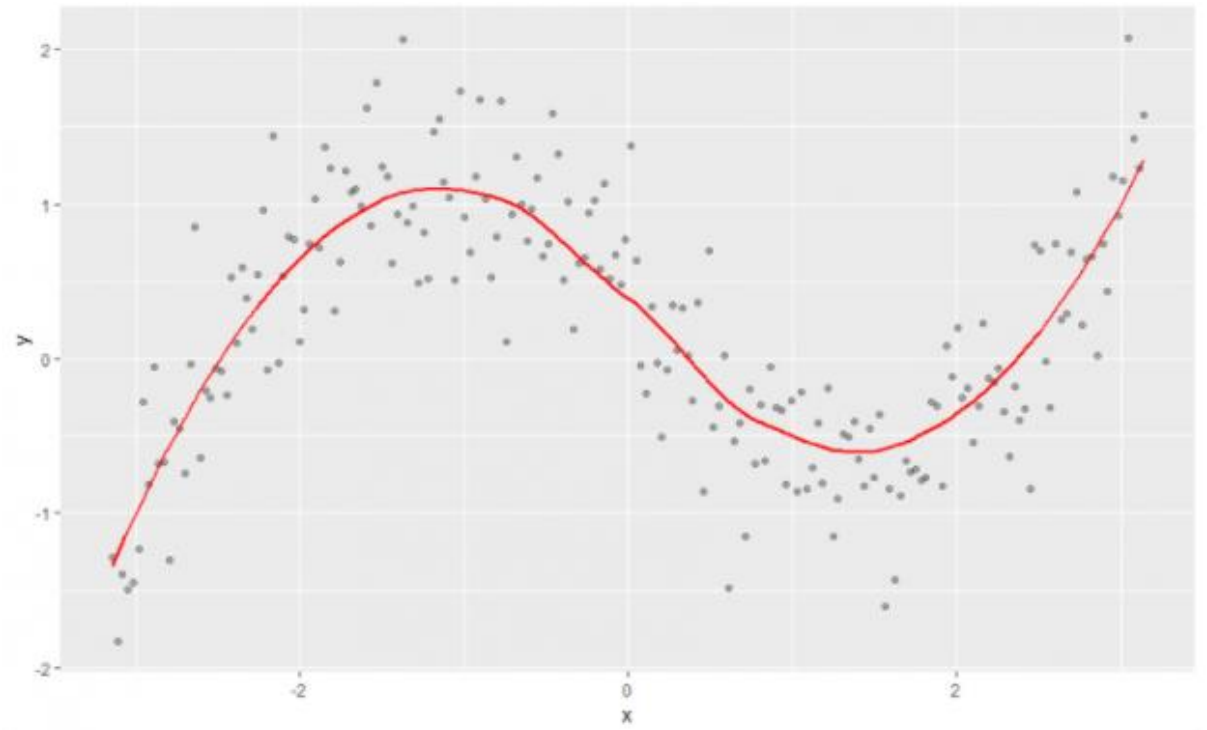


Fig. 2.7 Example of polynomial regression

2.2.3. Linear Discriminant Analysis

Linear discriminant analysis (LDA) is usually used to classify patterns between two classes, but there are some extensions to use LDA as multiple class classifier. The basic assumption of LDA is that classes are linearly separable. So, multiple linear discrimination function representing hyperplanes in the feature space are computed in order to distinguish the classes. For example if there are two possible classes, the LDA model traces a hyperplane and projects the data onto this hyperplane. The separation of the two categories is maximized. This hyperplane is created using the two criteria simultaneously:

•Maximizing the distance between the means of two classes.
•Minimizing the variation between each category.

Linear Discriminant Analysis (LDA) is a traditional method, based on decision theory and Bayes theorem, in which the probability density functions for the samples belonging to the +1 and −1 classes are assumed to be multivariate Gaussian with mean vectors μ_{+1} and μ_{-1}, respectively, and the same covariance matrix . The knowledge of these class-conditional densities, $f_{x|+1}(x)$ and $f_{x|-1}(x)$, together with the prior probabilities for the two classes, $P(-1)$ and $P(+1)$, makes computation of the class

posterior probabilities possible for a given sample $\boldsymbol{x}$, $P(+1 \mid \boldsymbol{x})$ and $P(-1 \mid \boldsymbol{x})$. Specifically, the probability for the +1 class given sample $\boldsymbol{x}$ is:

$$P(+1|x) = \frac{f_{x|+1}(x)P(+1)}{f_{x|+1}(x)P(+1)+f_{x|-1}(x)P(-1)} \quad (2.18)$$

Sample $\boldsymbol{x}$ is assigned to the +1 class if this probability exceeds 0.5, to the −1 class otherwise. Although the 0.5-threshold is optimum in terms of overall classification accuracy, a different value can be set to modify the balance between P_d (Probability of Detection) and P_{fa} (Probability of false alarm), and, therefore, the tuning of the threshold allows the tracing of the detector's ROC (receiver operating characteristic) curve. The samples belonging to the training set are used in LDA to estimate the mean vectors, the covariance matrix and the prior probabilities mentioned above. LDA is a linear classification method because membership of sample $\boldsymbol{x}$ can be equivalently assigned working on the log-odds function (i.e., the logarithm of the ratio between (+1 | $\boldsymbol{x}$) and (−1 | $\boldsymbol{x}$)), which is a linear equation in $\boldsymbol{x}$. [11]

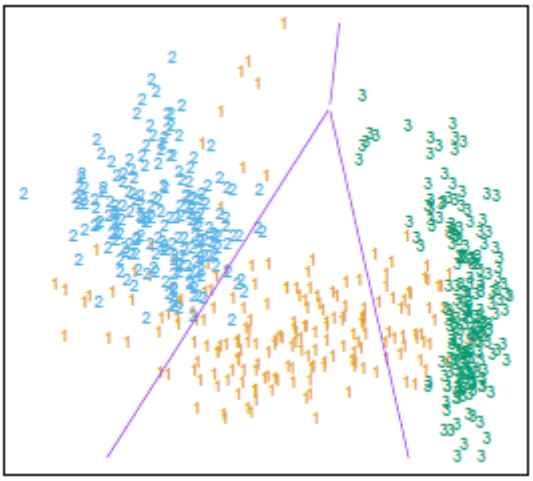


Fig. 2.8 The plot shows some data from three classes, with linear decision boundaries found by linear discriminant analysis.[12]

2.2.4. Logistic Regression

The Logistic Regression (LR) model assumes the log-odds function to be a linear function in $\boldsymbol{x}$ and derives the equations for the class posterior probabilities without introducing any assumption about the class-conditional density functions. The model has the objective to model the posterior probabilities of the K classes via linear

functions in x, while at the same time ensuring that they sum to one and remain in [0, 1]. The model has the form

$$log\ \frac{Pr(G=1|X=x)}{Pr(G=K|X=x)} = \beta_{10} + \beta_1^T x$$
$$log\ \frac{Pr(G=2|X=x)}{Pr(G=K|X=x)} = \beta_{20} + \beta_2^T x$$
$$\vdots$$
$$log\ \frac{Pr(G=K-1|X=x)}{Pr(G=K|X=x)} = \beta_{(K-1)0} + \beta_{K-1}^T x$$

(2.19)

where G is our predictor.
The model is specified in terms of K − 1 log-odds or logit transformations (reflecting the constraint that the probabilities sum to one). Although the model uses the last class as the denominator in the odds-ratios, the choice of denominator is arbitrary in that the estimates are equivariant under this choice. A simple calculation shows that

$$Pr(G=k|X=x) = \frac{exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{l=1}^{K-1} exp(\beta_{l0} + \beta_l^T x)}, k = 1, \ldots\ldots, K-1,$$
$$Pr(G=K|X=x) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T x)}$$

(2.20)

and they clearly sum to one.
To emphasize the dependence on the entire parameter set θ = $\{\beta_{10}, \beta_1^T, \ldots, \beta_{(k-1)0}, \beta_{k-1}^T\}$, we denote the probabilities Pr(G = k|X = x) = p_k(x; θ). When K = 2, this model is especially simple, since there is only a single linear function. It is widely used in biostatistical applications where binary responses (two classes) occur quite frequently. For example, patients survive or die, have heart disease or not, or a condition is present or absent.

In the binary case, the probability for the +1 class given the sample $\boldsymbol{x}$ results:

$$P(+1|x) = \frac{1}{1+exp(\beta_o+\beta^T x)} \quad (2.21)$$

i.e., a sigmoid function whose parameters β_0 and β can computed by maximizing a conditional log- likelihood function. The maximization is achieved through an iterative procedure in which the training set samples are exploited and the Newton-Raphson algorithm is typically applied to find the root of the first derivative. As in LDA, sample

$\boldsymbol{x}$ is assigned to the +1 class if (+1 | $\boldsymbol{x}$) is greater than 0.5, to the −1 class otherwise, but different threshold values can be used to trace the detector's ROC curve and change the balance between P_d and P_{fa}. [12]

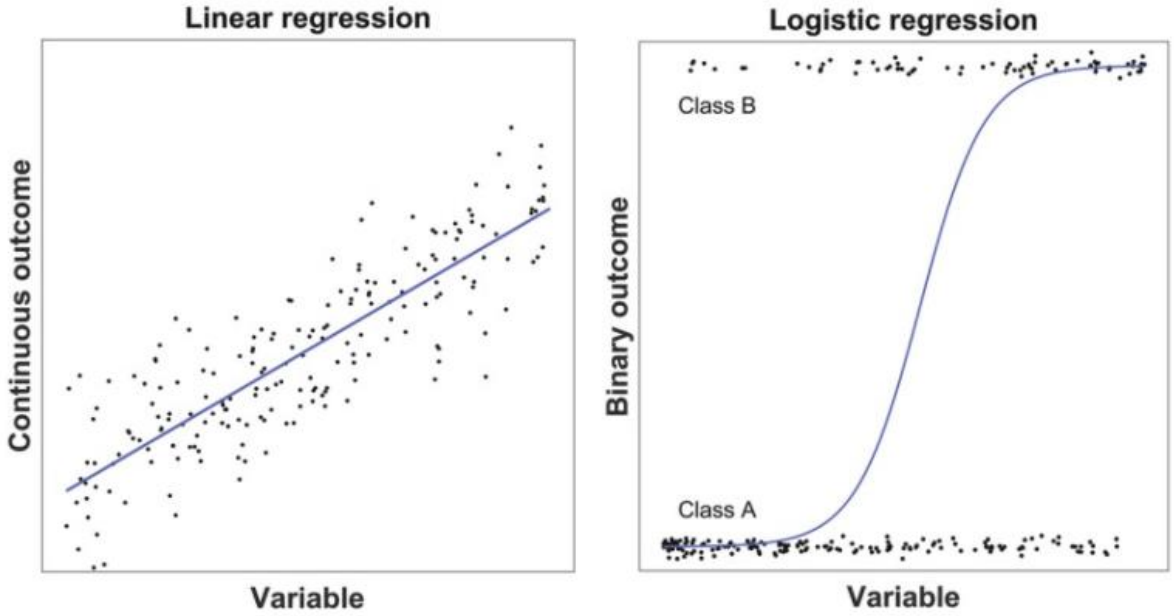


Fig. 2.9 Examples of predictive modeling (blue line) for a continuous outcome using linear regression and for a binary outcome using logistic regression. The predictions in the logistic regression are rounded to either class A or B using a threshold (0.5 by default)

2.2.5. Support Vector Machine

Support Vector Machine (SVM) indicates a popular approach to classification based on two key ideas:

- Exploiting the information represented by the samples at the interface between classes
- Extending from linear to non linear classification through kernel functions. SVM is the most widespread example of a kernel machine, i.e. Machine Learning methods based on kernel functions.

Our training data consists of N pairs (x_i, y_i) with 0<i<N+1 and $x_i \in R^p$ and $y_i \in \{-1, 1\}$.

A Support Vector Machine (SVM) assigns sample $\boldsymbol{x}$ to one of the two classes based on the score of the discriminant function:

$$h(x) = \sum_{i=1}^{L} \alpha_i y_i K(x_i, x) + b \quad (2.22)$$

L is the number of support vectors, that are the samples with nonzero coefficients α_i where $K(\cdot,\cdot)$ is a kernel function and the coefficients α_i and b are optimized by solving a quadratic programming problem. This optimization problem exploits the samples of the training set and includes a parameter C that bounds the range for α_i: 0< α_i<C, i=1,2,…,L. Sample $\boldsymbol{x}$ is assigned to the +1 class if $h(\boldsymbol{x})$ is positive, to the −1 class otherwise. As for previous methods, different threshold values can be used to trace the detector's ROC curve and change the balance between P_d and P_{fa}. In the SVM literature the most commonly adopted kernels are the linear function, polynomial function of order q and Gaussian radial basis function (RBF), defined, respectively, as:

$$K(x_i x) = x^T x_i$$
$$K(x_i, x) = (1 + x^T x_i)^q$$
$$K(x_i, x) = exp(\frac{-||x-x_i||^2}{2\sigma^2}) \quad (2.23)$$

where σ^2 is a specific parameter of the RBF kernel.
The choice of C and σ^2 (if the case) requires specific attention and, possibly, an optimization stage. In addition, although not strictly necessary, all features of the dataset samples are often preliminarily standardized, so that each of them has a zero mean and a unitary variance. This operation makes features insensitive to the scales on which they are measured and favors numerical stability in the solution of the quadratic programming problem mentioned above. [11]

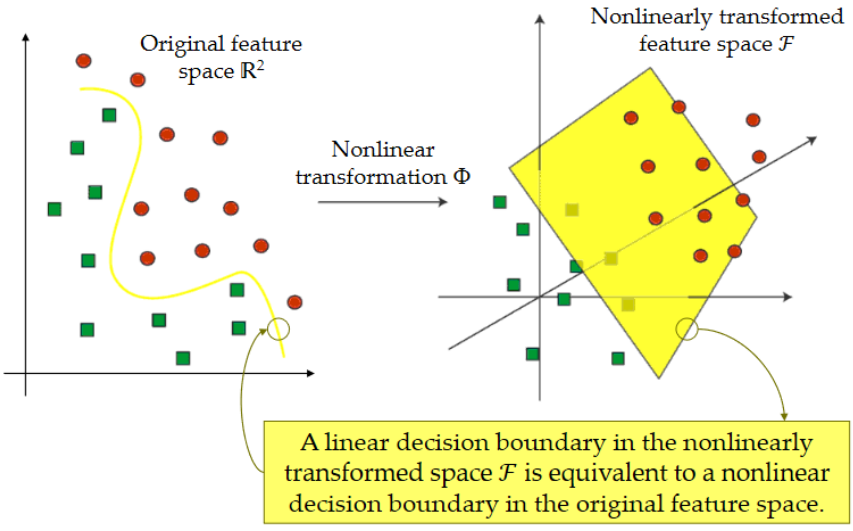


Fig. 2.10 Example of non linear transformation from an original R^2 space to non linear transformed space.

For what concerns the computational aspects, the quadratic problem of the training of an SVM has a unique global optimum, a desirable property that is not shared by other approaches. Also it can be shown that training an SVM is equivalent to minimizing an

upper bound on the probability of error, which is also related to the generalization capability of the classifier. The set of support vectors is usually a small subset of the training set, i.e., the discriminant function, expressed as a kernel expansion, gives a “sparse” representation of the training data. This property helps preventing overfitting.

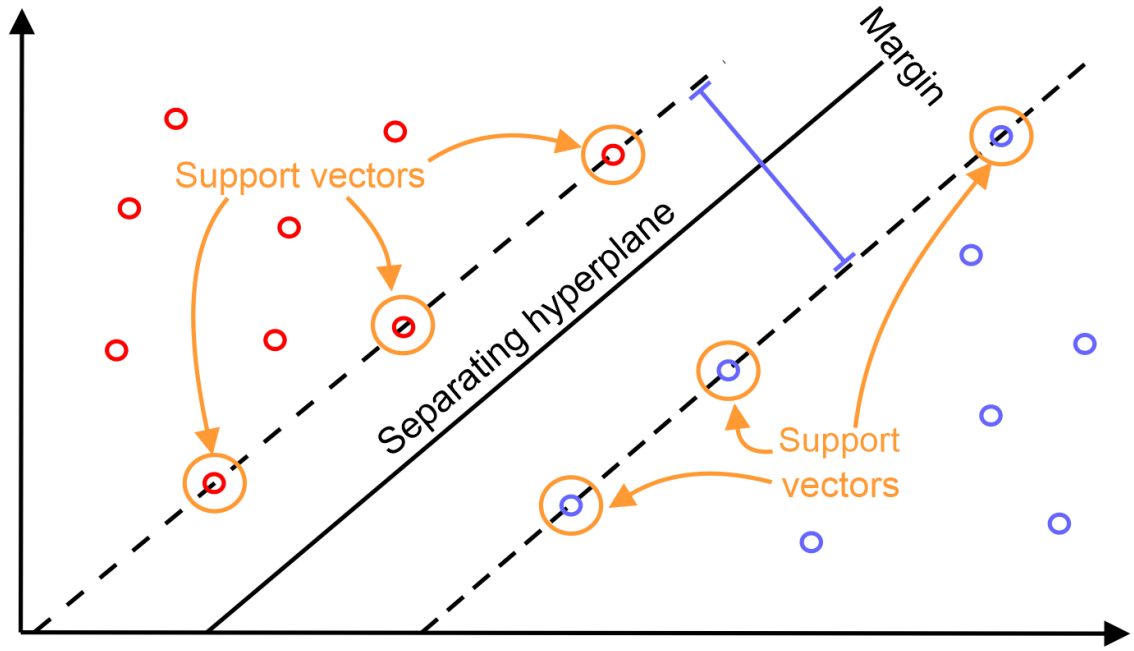


Fig. 2.11 A simple representation of how SVM works. It can be seen the support vectors for blue and red classes lying on the two hyperplanes parallel to the so called ‘separating hyperplane’ and the ‘margin’ that is the distance between the two dashed hyperplanes.

2.2.6. Random Forest

Random Forest (RF) is an ensemble model that aggregates the predictions individually achieved by many decision trees, separately trained on a subset of samples randomly chosen from the training set. A decision tree is an acyclic connected graph where each node represents a decision rule (called split) related to a single feature that leads to the partition of data in two groups. To automatically set the structure and splits of a decision tree, Classification and Regression Trees (CART) is an algorithm widely adopted in which a new node is created by identifying the feature that yields the best split in terms of a pre-selected metric.

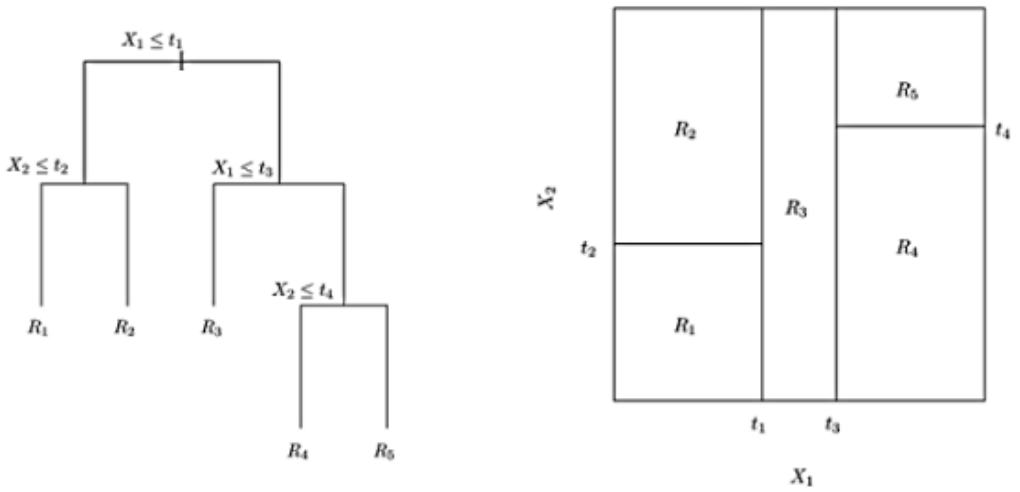


Fig. 2.12 Example of a CART (left) with the related partition of the feature space (right)

In an RF model B trees are generated and trained in an independent and identically distributed way by performing, for each tree T_b, b = 1, …, B, the following steps :

- a subset of L samples is drawn randomly from the training set, uniformly and with replacement (this means that some samples are taken more than once, others are not chosen at all)

- such a subset is used to grow the tree T_b, for each node of which a pool of m features is selected (at random and uniformly from the d features) and used to identify the best feature and the best decision rule to split the node into two daughter nodes

- the previous step is repeated until at least one of the predefined stopping criteria is satisfied.

When all the B trees are generated, an unknown sample $\boldsymbol{x}$ is classified as follows: the sequence of decision rules of the b-th tree is applied to $\boldsymbol{x}$ in such a way that the corresponding class prediction $\hat{y}^b(\boldsymbol{x})$ is reached (namely, +1 or −1); the predictions from all the trees of the RF are used to compute a score.

$$g(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{y}^b(x) \quad (2.24)$$

Sample $\boldsymbol{x}$ is assigned to the +1 class if ($\boldsymbol{x}$) is positive, to the −1 class otherwise. Threshold values different from zero can be used to trace the detector's ROC curve and tune the balance between P_d and P_{fa}. Although the setting of B and m does not critically affect performance, it deserves some investigation, recalling that these two parameters affect the computational burden. [11]

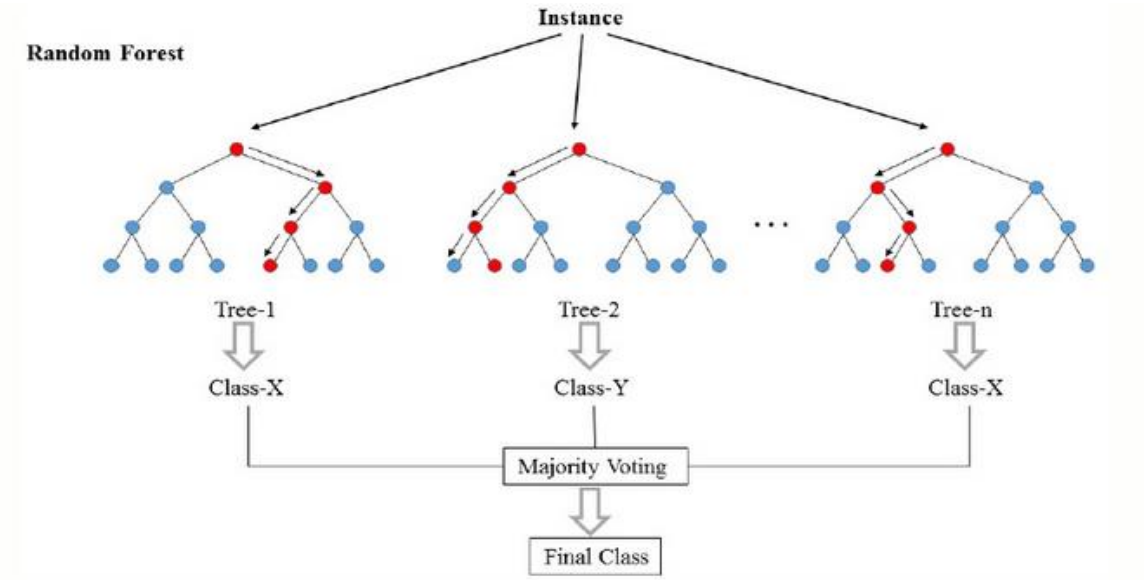


Fig. 2.13 Random forest illustration example in which n trees are used and major voting criterion is considered

2.3. Cross-Validation

The evaluation of the learned models in Supervised Learning is a very important step. Usually, the dataset is divided in training and testing sets. The training set is used to learn the new model and the testing set is used to test and evaluate the model. The performances can be evaluated using error metrics estimating the accuracy of the fitted model, such as Mean Square Error or Percentage Error. This simple concept can be extended in order to be more rigorous using the *K-fold Cross Validation(CV)* method. In fact, using K-Fold CV a given dataset is divided into *K* sections/folds where each fold is used as a testing set. For example, consider a scenario of 5-Fold cross validation (K=5), dataset is split into 5 folds. In the first iteration, the first fold is used to make the test of the model and the remaining folds are used to train the model. In the second iteration, the second fold is used as the testing set and the remaining folds are used as the training set. This process is repeated until each fold of the 5 folds have been used as the testing set.

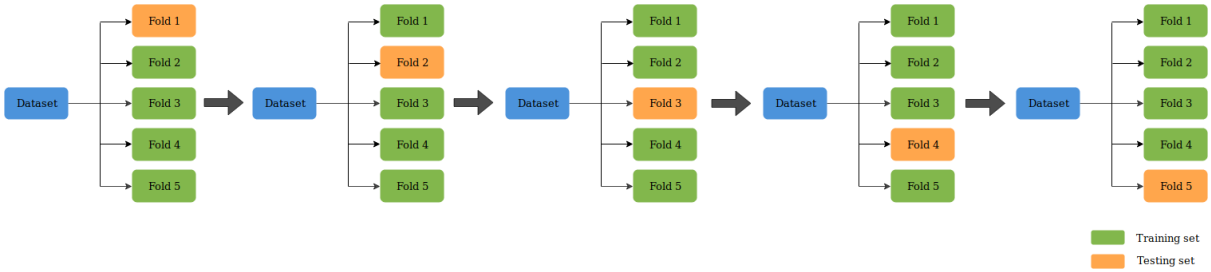


Fig. 2.14 Graphical explanation of cross validation algorithm with 5 folds

3. Rainfall Detection

3. Rainfall Detection

As said in the Introduction Chapter, underwater noise analysis allows estimating parameters of meteorological interest, difficult to monitor with other devices, especially in polar waters. Rainfall detection is a fundamental step of acoustical meteorology toward quantifying precipitation. To date, this task has been conducted with some success by using few frequency bins of the noise spectrum, combining their absolute values and slopes into some inequalities. Unfortunately, these algorithms do not perform well when applied to spectra obtained by averaging multiple noise recordings made over the course of an hour. The use of supervised, statistical learning models allows the use of all the frequency bins in the spectrum, exploiting relationships that are difficult to identify by a human observer. Among the different models tested, a binary classifier based on random forest performed well with moderate computational load. Using a data set consisting of over 18000 hourly-averaged spectra (approximately 25 months of at-sea recordings) and comparing the results with measurements from a surface-mounted rain gauge, the proposed system detects precipitations greater than 1 mm/h with 90% probability, keeping the false alarm probability below 0.5%. This system has demonstrated remarkable robustness as performance is achieved without excluding spectra corrupted by passing ships or high winds. So in this chapter rainfall detection topics will be treated. In the first part there will be the analysis of the algorithms and Machine Learning techniques used for the processing of the spectra and fitting of the models such as, Random Forest, SVM, using different kinds of kernel functions, Linear Regression and LDA. In this part tools are compared in order to show what is the best one. To this aim some indices or variables are calculated in order to make comparisons, for example the probability of false alarm, the probability of detection, or also the OA (overall accuracy) and the AUC (area under the ROC curve). Then in the second part of the chapter results are compared and explained with a series of plots, in order to analyse in deeper what are the outputs and to interpret them in the best way. In the last part of the chapter, the state-of-the-art algorithms are considered and their results are shown in order to make some comparison between these algorithms and supervised ones. In particular it will be analysed rule-based algorithms (called in this way because the detection is performed by decision rules using empirical formulas), such as those presented in Vagle *et al.*, 1990 [1], Ma *et al.*, 2005 [4], Nystuen et al., 2011 [8] and Nystuen *et al.*, 2015 [7], but also supervised ones such as Taylor21[6].

3.1 Algorithm analysis

Before entering in the details of the Machine Learning techniques that have been used, it is necessary to understand what kind of algorithm is used in order to validate the results and output of the various models. In this thesis aiming to assess and compare the detection performance of the statistical models, a 10-fold cross-validation with stratification in dataset partitioning is used. So every time the dataset is divided in training and test parts, data are selected in a random way but based on their class (Fig. 3.1). In this way, when a test set, representing the 10% of the complete dataset, is selected randomly, it's sure that for each class of rain or wind about the 10% will be put in the test set and 90% in training set. In addition, for the SVM approach feature standardization is applied, the constant is set equal to 1.0, according to common practice, and the variance of the Gaussian RBF kernel, after some tests, is set equal to 8.0. For the parameters of the RF, B = 100 trees and m = 22 features are used, although a change of these values in even rather broad ranges does not significantly affect the performance obtained. Data processing is performed using Matlab and, in particular, the Statistics and Machine Learning toolbox. Additionally, it has been verified that results perfectly consistent with those shown below can be obtained using the scikit-learn library for the Python programming language.

				WIND											
				Beaufort scale (intervalli in m/s)										> 1000	
				B0	B1	B2	B3	B4	B5	B6	B7	B8		> 100	
				<0,3	0,3-1,5	1,5-3,3	3,3-5,8	5,8-8	8-10.8	10.8-13.9	13.9-17.2	17.2-20.7	tot	> 10	
RAIN	intervalli in mm/h	No rain (R0)	0-0,1	0	1756	5553	4779	2884	1599	596	140	10	17317		
		Light (R1)	0,1-2,55	0	9	70	129	136	171	98	26	0	639		
		Moderate (R2)	2,55-7,55	0	0	7	15	47	50	43	16	1	179	876	tot. con pioggia
		Heavy (R3)	7,55-50	0	0	3	5	10	18	15	7	0	58		
			tot	0	1765	5633	4928	3077	1838	752	189	11	18193		

Fig. 3.1. Subdivision of the dataset in classes of wind and rainfall intensity. For what concerns rain, 4 intervals are considered, from R0 (No rain) to R3 (Heavy rain). For what concern wind, Beaufort scale is considered and so 9 classes are used, from B0 to B8.

The ROC curves obtained from the trained models are shown in Fig. 3.2 and their performance is summarized in Table 3.I, where the P_d value for which P_{fa} = 0.01 is reported. In the cross-validation procedure, one of the K folds into which the dataset has been split, in turn, is not used for training but is instead used for testing. At the end of the procedure and after setting the decision threshold, it is possible to calculate P_d and P_{fa} in each fold used for the test. The data inserted in Table 3.I are the average and standard deviation of the P_d and P_{fa} values calculated on each fold. The average P_{fa} is 0.01 since the threshold is set precisely to achieve this result. The threshold values used for each classifier are also included in Table 3.I and should be read recalling that the optimal threshold values (i.e., those values that maximize OA) are: 0.5 for LDA and LR; 0 for SVM and RF. The change of the kernel function for the SVM classifier does not significantly alter the performances, although the linear case shows a lower detection ability and the Gaussian case reports the worst *AUC* figure. The *OA* values are all greater than 0.97, but this finding has little relevance because it is strongly influenced by the correct classification of non-rainy samples (probability 0.99, P_{fa} being 0.01) which are by far the most numerous. Overall, the best option among the models considered is the RF classifier because it achieves the best performance figures, shows a stability better than that of SVM classifiers with polynomial or RBF kernels, requires a computational load lower than that of such SVM classifiers, and is not appreciably affected by changes in the parameter setting. Accordingly, in the remainder of this chapter, further analysis and performance comparisons will be carried out with reference to the RF-based classifier. Indeed, the goal is not to identify the best model, but rather to demonstrate that the Machine Learning approach is well suited for rainfall detection also in case of drizzle phenomena, characterized by low rainfall intensity.

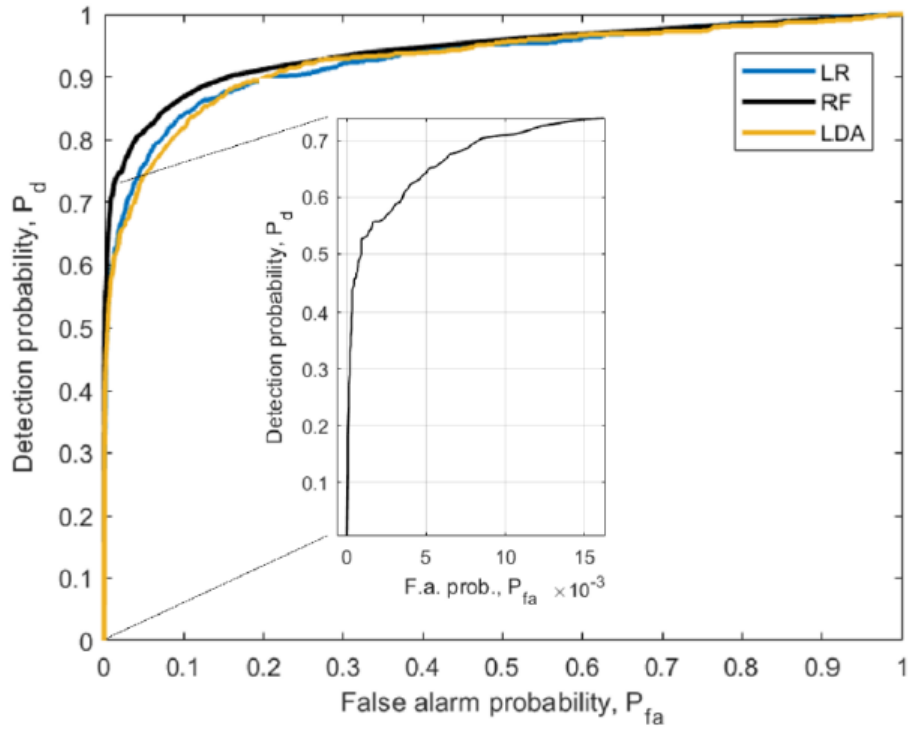
	P_d	P_{fa}	*OA*	*AUC*	*TV*
LDA	0.583 ± 0.053	0.010 ± 0.003	0.970	0.926	0.38
LR	0.597 ± 0.058	0.010 ± 0.003	0.971	0.928	0.33
SVM, linear	0.667 ± 0.041	0.010 ± 0.003	0.974	0.931	-0.67
SVM, polynomial, q=2	0.702 ± 0.060	0.010 ± 0.002	0.976	0.936	-0.51
SVM, Gaussian RBF	0.703 ± 0.062	0.010 ± 0.003	0.976	0.897	-0.71
RF	0.708 ± 0.054	0.010 ± 0.002	0.977	0.941	-0.42

Table 3.I. Detection probability, false alarm probability, overall accuracy, *OA*, and area under the ROC curve, *AUC*, for the supervised learning models. For the probabilities, the average ± the standard deviation is reported. For each classifier, the threshold value *TV*, used to obtain the average P_{fa} equal to 0.01, is reported.

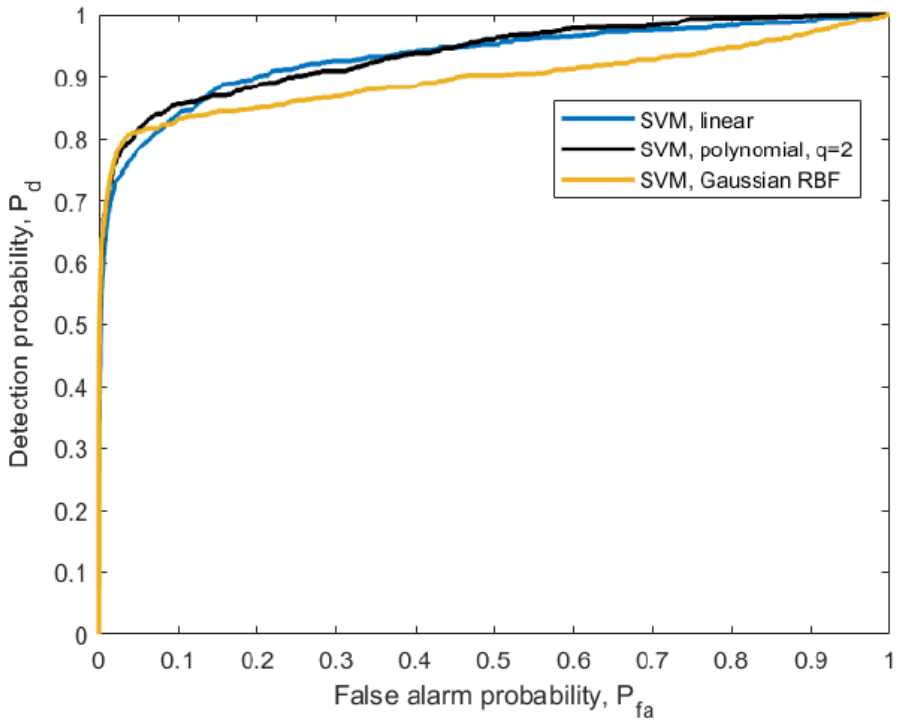
To delineate the desired performance of the rain detector, it is necessary to recall that rainfall is present in 5% of the one-hour periods included in the dataset and that the precipitation limit which distinguishes between rainy and non-rainy hourly-averaged spectra is particularly low (i.e., 0.1 mm/h). In this scenario, it is strictly necessary that the false alarm probability be very low, while a detection probability not too close to one may be acceptable. Consequently, the performance of a detector cannot be considered acceptable if P_{fa} exceeds 0.01.

3.2. In-depth results analysis and comparisons

In Fig. 3.2 (a), the zoom of the ROC curve for the RF model demonstrates that P_d remains greater than 0.6 even if P_{fa} is reduced by as much as 0.0035. More precisely, the following probability pairs, {P_d, P_{fa}}, lie on that curve: {0.661, 0.006}, {0.644, 0.005}, {0.623, 0.004}, {0.588, 0.003}. The ability of the classifier to detect the precipitation can be analysed as a function of the rainfall rate, as shown in Fig. 3.3. In this case, P_d is estimated using the hourly samples in which the cumulated rainfall, measured by the rain gauge on the platform in one hour, is equal to or greater than a value G. The P_d curves shown in Fig. 3.3 are related to three choices of the threshold value, leading to different P_{fa}: 0.010, 0.005 and 0.003. P_d increases rapidly with G, reaching, respectively, 0.921, 0.897 and 0.876 for G = 1 mm/h. Although the probabilities of detection of the three detectors show significant differences for $G < 2$ mm/h, for rainfall intensities higher than this value the three detectors provide similar P_d. It is therefore possible to design acoustic detectors capable of detecting rainfall of intensity greater than 2 mm/h with a probability greater than 0.9, while keeping a false alarm probability of 0.003. The sharp P_d increase with G observed in Fig. 3.3 shows that the missed detections are mainly related to drizzle phenomena characterized by low precipitation intensity. This relation is confirmed by the average of the rainfall intensities recorded by the surface rain gauge when the precipitation is detected or missed by the underwater acoustic device. Among the 876 acoustic samples collected in rainy conditions (with intensity greater than or equal to 0.1 mm/h), the RF-based classifier correctly detects 620 of them (70.8%) and misses the remaining 256 samples (29.2%). The average rainfall intensity measured for the detected samples is 2.98 mm/h, whereas the average intensity for the missed samples is 0.71 mm/h. The small fluctuations that the curves in Fig. 3.3 show, especially for G greater than 2 mm/h, are mainly due to the limited number of samples available to train the RF model (in the training phase) and to estimate the detection probability (in the test phase). The number of acoustic samples with rainfall greater than 3 mm/h is about 200, while the number of those with rainfall greater than 4 mm/h is reduced to less than 150.



(a)



(b)

Fig. 3.2 ROC curves for the supervised learning models listed in Table 3.I. (a) LDA, LR and RF, with a zoom for the RF model. (b) SVM with three kernel functions. [11]

Moving from missed detections to false alarms, an analysis of wind distribution provides some interesting insights. Fig. 3.4 shows the wind speed histograms for rainy samples correctly detected (620 samples), non-rainy samples correctly classified (17145 samples), and non-rainy samples raising false alarms (172 samples, corresponding to P_{fa} = 0.01). The average wind speeds for these three categories are, respectively, 8.5, 4.6 and 9.3 m/s. It is evident that the false alarm samples present a wind distribution more similar to that of the rainy samples than to that of non-rainy samples.

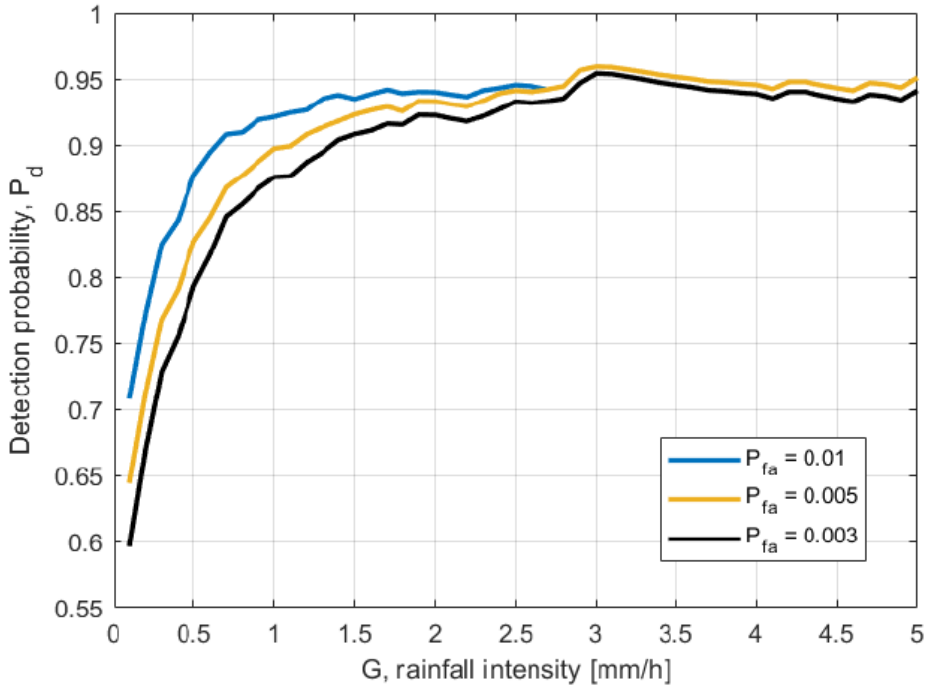


Fig. 3.3 Detection probability for the rainy samples with a rainfall intensity greater than or equal to G. Three RF-based classifiers, with different false alarm probabilities, are considered.[11]

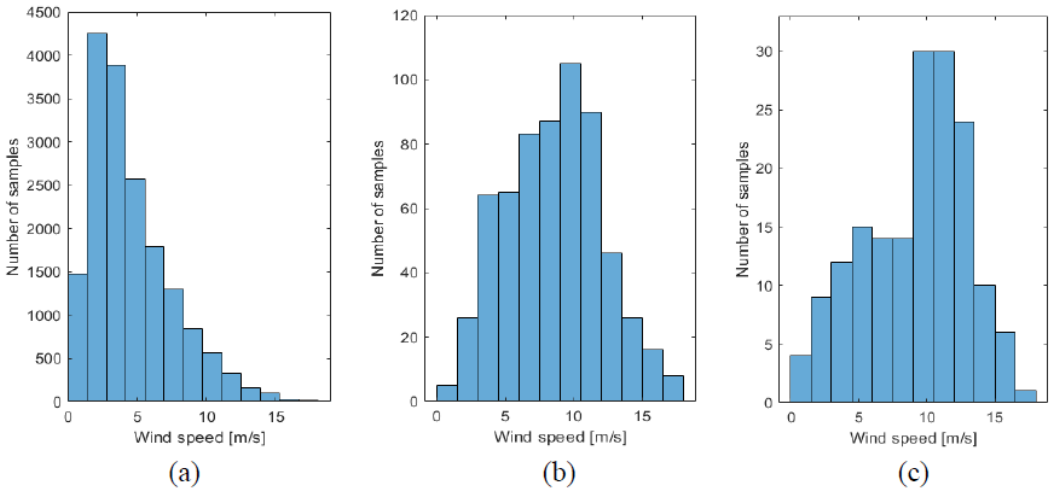


Fig. 3.4 Histograms of wind speed for: (a) non-rainy samples correctly classified; (b) rainy samples correctly detected; (c) non-rainy samples raising false alarms.

However, the histogram of non-rainy samples shows that there are over a thousand samples with wind speed greater than 10 m/s that are correctly classified. To analyse this issue in detail, Fig. 3.5 shows the estimated P_{fa} when the samples for which the wind speed is greater than W, $W \in$ [0.1, 10] m/s, are considered. The three detectors already examined in Fig. 3.3 are included. Notwithstanding the considerable rise of the P_{fa} with increasing wind speed, the probability of correct classification for non-rainy samples remains satisfactory (e.g., for W = 10 m/s, P_{fa} increases from 0.01 to 0.08, but the probability of correctly classifying a non-rainfall sample is still high: 0.92). Therefore, the wind-related similarity only partially explains why the detector is misled and false alarms occur.

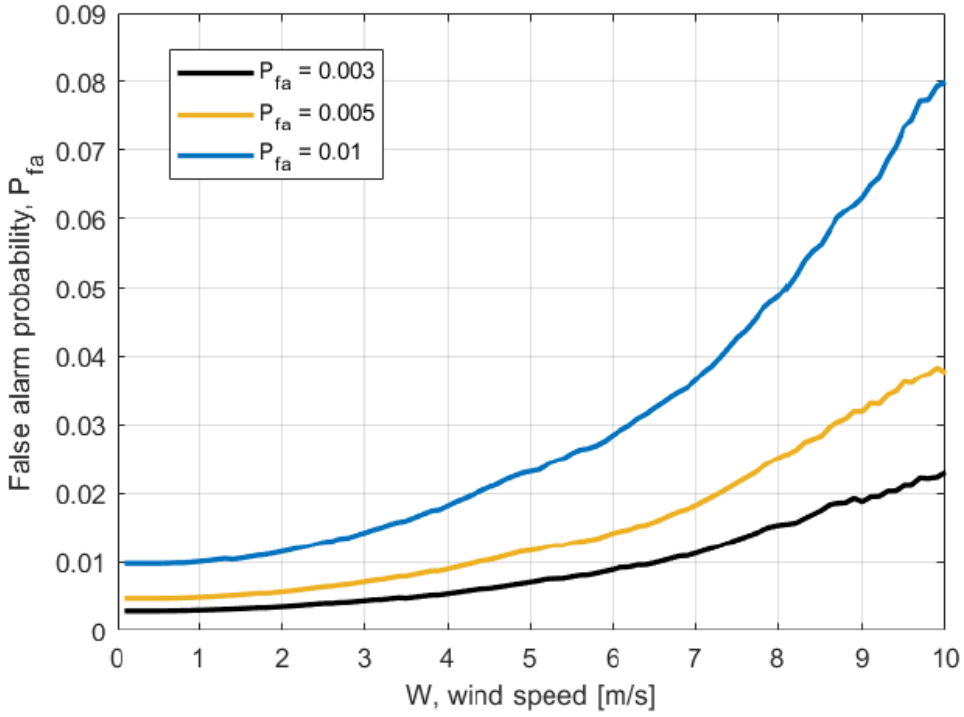


Fig. 3.5 False alarm probability for non-rainy samples with a wind speed greater than W. Three RF-based classifiers with different probabilities of false alarm (on the entire dataset) are considered. [11]

Finally, the performance of the RF-based detector during the period of data collection is examined in Fig. 3.6, where the height of the bars indicates the rainfall intensity measured by the rain gauge and the colours distinguish samples correctly detected (light blue bars) from missed alarm samples (orange bars). Samples raising false alarms are inserted as white bars with black edges, and an arbitrary height of 2 is set for them. The two zoom panels show the typical behaviour that characterizes the 25-month span of data collection with good uniformity. 1999 out of the 18193 dataset samples are characterized by the passage of a ship within 5 km of the platform during the observation hour. These samples are not discarded and are used, like all others, to train and test the statistical model. It is verified *a posteriori* that the P_d and P_{fa} values estimated on these samples do not differ significantly from those already reported, thus supporting the robustness of the proposed detector. Fig. 3.7 shows a further zoom of a portion of the 25-month span introduced in Fig. 3.6, in which the one-hour intervals at which a ship passage occurred are indicated by a black diamond. Again, it can be verified that there is no correlation between the ship passages and false alarms or missed detections. Since the rainy samples most susceptible to missed detection are those characterized by modest precipitation intensity, it is reasonable to expect that a P_d of 70.8% would allow the detection of rainy samples with the greatest impact in terms of cumulative precipitation. Fig. 3.8 compares the cumulative rainfall profiles over the 25-month span obtained by considering all rainfall events measured by the rain gauge (876 hourly samples) or only those detected by the proposed underwater acoustic system (620 hourly samples). It can be emphasized that although the detected rainy samples are 70.8% of the total, these samples, at the end of the

observation period, gather 91.0% of the cumulative precipitation (i.e., 1851 mm out of a total of 2035 mm). The performance achieved by the RF-based detector acting on hourly-averaged spectra can also be compared with those obtained by other underwater acoustic systems [2,3,4] acting on short-term spectra. By using data in Fig. 3.3, it is immediately possible to observe that the proposed system, at the same P_{fa} values and rainfall intensities, always provides a significantly higher detection capability. Moving from short-term spectra to hourly-averaged spectra the performance obtained from the detection algorithms used in [2,3,4] worsen. As a result, the supervised learning models adopted in this thesis achieve a detection performance significantly better than those obtained from rule-based detection algorithms and better even than that obtained from the binary classifier proposed in [6]. Another useful comparison is with the rainfall detection capability of a weather radar installed on Mount Settepani, located at about 1400 m above sea level, about 87 km away from the buoy, covering the area of investigation and the data of which were used in [2]. In that thesis, rainfall detection by radar at the W1M3A observatory is characterized by P_{fa} = 0.009 accompanied by P_d = 0.728 for G = 0.1 mm/h and P_d = 0.846 for G = 1 mm/h. The data in Fig. 3.3 show that the performance of the proposed acoustic system is very close to that of radar: slightly worse for G = 0.1 mm/h and slightly better for G = 1 mm/h. However, it is important to emphasize the qualitative nature of this comparison because the radar performance refers to a time period of about 11 months [2], thus significantly shorter than the 25-month period considered in this study. [11]

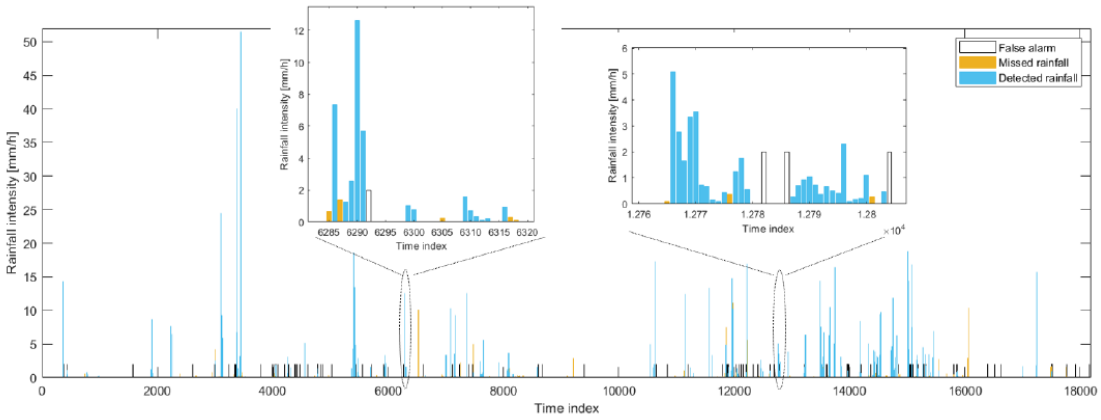


Fig. 3.6 Rainfall intensity during the 18193 hours of observation (one sample per hour; about 25 months of data collection), with indication of detected rainy samples (620 hours), missed rainy samples (256 hours), false alarm samples (172 hours). The zoom panels show two examples of the occurrence of the three cases on a fine scale. [11]

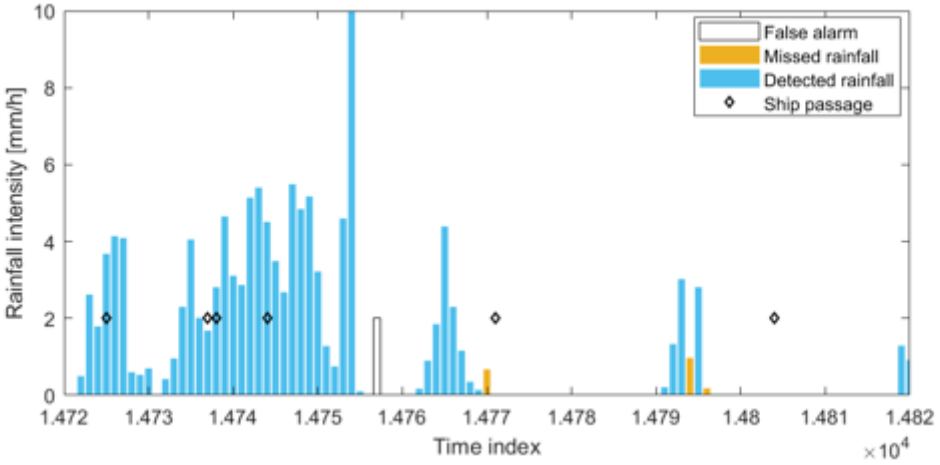


Fig. 3.7 Zoom of a portion of the 25-month span shown in Fig. 3.6, with the indication (black diamond) of the one-hour intervals at which a ship passage occurred. [11]

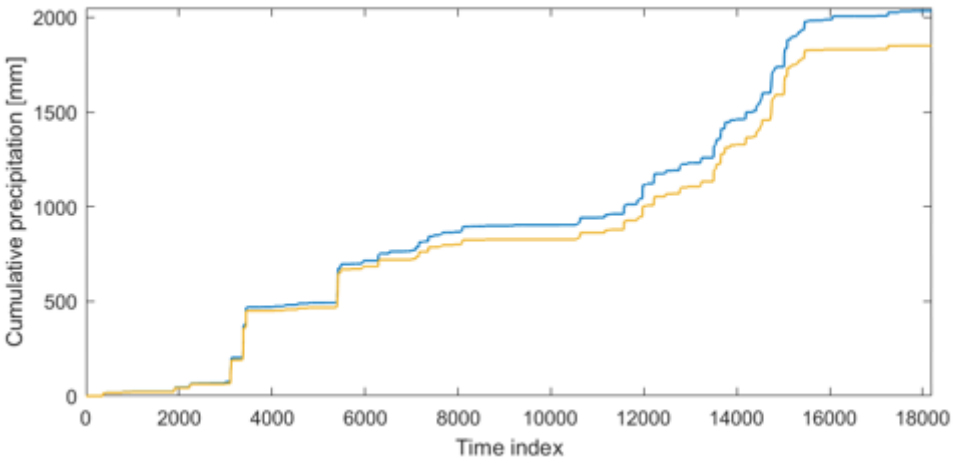


Fig. 3.8 Cumulative rainfall profiles over the 25-month span obtained by considering all rainfall events measured by the rain gauge (light blue line) or those detected by the underwater system (golden line). [11]

3.3. Performance of the literature algorithms

The application of the algorithms introduced in the first chapter to the available dataset provides the results summarized in Table 3.II. It is important to recall that these algorithms were designed to detect rainfall using short-term acoustic spectra, whereas in this thesis they are applied to hourly-averaged spectra.

The algorithms applied are:

-Vagle90 [1]:

$$V1\colon S(19.5) - S(3) > -13.82 \quad OR \quad S(12.5) - S(3) > -10.54 \quad OR \quad S(8) - S(3) > -6.82$$

$$V2\colon S(19.5) - S(3) > -13.25 \quad OR \quad S(8) - S(3) > -6.82$$

(3.1)

-Ma05 [4]:

$$S(21) + 2.35\,S(5.4) > 194$$
$$S(21) > 48 \; AND \; S(5.4) > 53$$
$$S(21) > 44 \; AND \; S(21) - 0.7\,S(8.3) > 14$$

(3.2)

- Nystuen15 [7] and Nystuen11 [8]:

$$S(20) - 0.75\,S(5) > 5 \; AND \; S(5) \leq 70$$
$$S(8) > 60 \; AND \; Q(2,8) > \theta \; AND \; S(20) > 45$$
$$S(8) < 50 \; AND \; Q(8,15) > -5 \; AND \; S(20) > 35 \; AND \; S(20) > 0.9\,S(8)$$

$$\{S(20) + 0.1144S^2(8) - 12.728S(8) > -307 \; AND \; Q(2.8) > \theta \; AND \; S(20) + 0.1S^2(8) - 11.5S(8) < -281 \; AND \; 51 < S(8) < 64\}$$

(3.3)

where Q(f1,f2) is the spectral slope, in dB/decade, between the frequencies f1 and f2 (expressed in kHz):

$$Q(f_1, f_2) = \frac{S(f_1) - S(f_2)}{log_{10}(f_1) - log_{10}(f_2)} \quad (3.4)$$

θ = –18 dB/decade in [8] and θ = –13 dB/decade in [7].

	P_d	P_{fa}
Vagle90 [1] – V1	0.900	0.630
Vagle90 [1] – V2	0.880	0.570
Ma05 [4]	0.300	0.001
Nystuen11 [8]	0.671	0.116
Nystuen15 [7]	0.586	0.094

Table 3.II. Probabilities of detection, P_d, and false alarm, P_{fa}, for the rule-based algorithms applied to hourly-averaged spectra.

The algorithms in [1] achieve high P_d, but this is accompanied by excessive P_{fa}. A bias in hydrophone sensitivity cannot be the cause of the problem, because the quantities compared with thresholds in Vagle90 formula are subtractions between measurements. One option to make the algorithms more selective is to arbitrarily increase the threshold values, modifying the V2 rule, as follows:

$$S(19.5) - S(3) > -13.25 + \delta \quad OR \quad S(8) - S(3) > -6.82 + \delta \qquad (3.5)$$

where $\boldsymbol{\delta} > 0$. Varying the value of $\boldsymbol{\delta}$ between 0 and 5, the ROC curve in Fig. 3.9 is obtained. The P_{fa} reduction is obtained but, unfortunately, it is accompanied by a significant P_d decrease. When a similar modification is applied to the V1 rule, the performance is worse since the ROC curve is always below that shown in Fig. 3.9 for the V2 rule. What is more, the algorithms in [4], [7] and [8] do not provide satisfactory performance, because P_d is too low, as for [4], or P_{fa} is too high, as for [7] and [8]. The performance of these algorithms can be optimized by considering potential errors in hydrophone sensitivity. To do this, the values $S(f_k)$ in equations from (3.2) to (3.8), for whatever f_k, are replaced by $S(f_k)+\varepsilon$, where ε is intended to compensate a sensitivity bias. Varying ε between –10 and 10 dB re 1 μPa^2 Hz^{-1}, the ROC curves shown in Fig. 3.9 are obtained. This comparison clearly evinces that the rule-based algorithm

achieving the best performance (with the discussed correction) is the one proposed in [4]. In particular, for ε = 2 dB re 1 μPa^2 Hz^{-1}, a detection probability P_d = 0.521 is accompanied by P_{fa} = 0.010. The corrections introduced fail to reduce the false alarms of the other rule-based algorithms [1,7,8] to acceptable values: P_{fa} always remains significantly higher than 0.01.

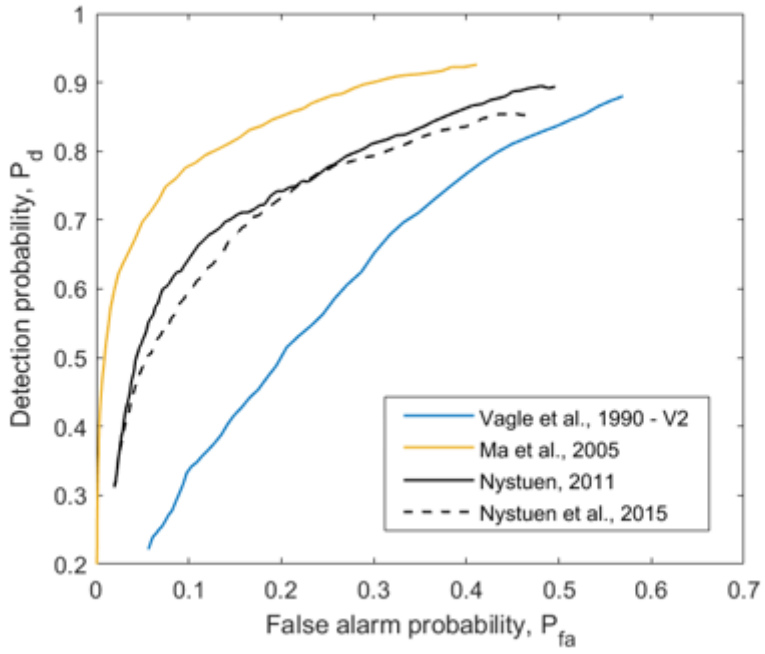


Fig. 3.9 ROC curves obtained by varying threshold values [1] and hydrophone sensitivity [4,7,8] in the rule-based algorithms listed in Table 3.II. [11]

So the rainfall detection by rule-based algorithms taken from the literature have not provided satisfying performance on this type of spectrum, Machine Learning methods have shown that the detection can be carried out successfully. The linear classifiers (i.e., LDA and LR) perform moderately better than the best rule-based algorithm, increasing the probability of detection to about 0.6. A further advantage is offered by the SVM and RF classifiers for which probability of detection exceeds 0.7. Instead in the solution proposed by [6] Machine Learning methods are used to predict rainfall intensity and wind speed using all the frequency bins of the underwater noise spectrum as input data, in order to exploit implicit relationships that are not evident to the human observer. In fact in [6] Machine Learning methods are applied for rainfall detection, using hourly-averaged spectra as input data. Unfortunately, detection is limited to precipitation intensities greater than 1 mm/h and the performance obtained over a one-year period is worse than that reported in [2], where a rule-based prediction method

[8] was fed with short-term data collected by the same equipment and over the same time period used in [6]. For the detection task, a binary classifier is built through the CatBoost algorithm, setting the lower bound for rainfall intensity equal to 1 mm/h. When the detector is applied to the available one-year dataset (through the cross-validation scheme), a P_{fa} = 0.0332 and a P_d = 0.811 are obtained: a poorer performance than that obtained in [2] using the same dataset but exploiting short-term data in place of hourly-averaged data.[11]

3.4. Final analysis and comparisons

In this analysis, kernel-based and ensemble-learning models have demonstrated the best performance among the experimented supervised classifiers. In particular, the RF-based binary classifier has shown a satisfactory balance between computational burden and performance, reaching a detection probability greater than 90% when precipitation exceeds 0.7 mm/h and P_{fa} is 1% or, alternatively, when precipitation exceeds 1.4 mm/h and P_{fa} is 0.3 %. This level of performance is slightly better than that obtained by a weather radar operating in the experiment area, and therefore the proposed method represents a promising alternative to obtain an estimate of rainfall intensity in areas where environmental constraints do not allow the installation of rain gauges or radar systems. This is even more noteworthy in polar areas, where global warming is changing the hydrological cycle of those regions, thus increasing rainfall with respect to snow precipitation. While the presence of high wind, especially above 10 m/s, induced a noticeable increase in the probability of false alarm, the performances did not undergo significant alterations in the hours in which a ship transited in the area where the underwater measurement device was placed. Similarly, no fluctuations in performance were observed on a seasonal basis, attributable to varying underwater propagation conditions. It is worth recalling that rainfall detection was based on the amount of precipitation accumulated over the course of an hour, and it is not possible to determine whether this amount is due to transient, intermittent or continuous rain. Although very promising, supervised learning models require a training phase that necessitates extensive collection of underwater acoustic spectra, accompanied by concomitant precipitation measurements to be used as ground truth. On the other hand, this is also partially necessary for rule-based algorithms that need specific calibrations to account for geographic location and hydrophone sensitivity.

The possibility of using the trained detector in geographic areas other than the one in which the training data were collected is a topic for future investigation. However, it is reasonable to assume that in similar environmental settings, a trained detector can continue to operate successfully. The performance obtained working on averaged spectra suggests that Machine Learning models may also be advantageous for rain detection using short-term acoustic spectra.

4. Wind prediction

4. Wind prediction

As said in the Introduction Chapter the prediction of the wind intensity values is a very important and discussed topic that could have many applications from meteorology to navigation, in particular when the situation in which a system has to work is difficult or not suitable to install and use surface sensors, such as anemometers. In the past wind prediction was the objective of many scientific researches, but in many cases the results were not so good and not good performances were obtained. In particular this happened because the old state-of-the-art empirical formulas were not so suited for this kind of regression. In this study Machine Learning algorithms are applied and the results of the previous state-of-the-art algorithms are improved. Previous methods for prediction are based on the analysis of few frequencies and on the estimation of empirical formulas, instead in supervised learning all the frequencies are taken in consideration together and so the potential of this kind of analysis is greater because it is not limited to few features. The use of supervised learning gets satisfying results applying polynomial regression with Dictionary Learning preprocessing. Using a data set consisting of over 18000 hourly-averaged spectra the proposed algorithms obtain a RMSE (Root mean square error) of 1.15 m/s, a very low bias (ME, mean error) and a MAE (Mean average error) of 0.80 m/s. The structure of the algorithm is the same as the rainfall prediction case, because the research part was studied in parallel. This happened because rainfall and wind predictions show a sort of similarity or parallelism for what concern results related to methods used. In the first part preprocessing algorithms and Machine Learning tools for regression that were used in the thesis will be described such as, Linear Regression, Polynomial Regression, Dictionary Learning, MFCC and GTCC. In this part results are shown and algorithms are compared. Some indices or indicators are used to make comparisons between them, such as RMSE, ME or MAE. In the second part of the chapter results are compared using figures and plots describing well the information that could be analysed from the outputs of the processing. In the last part of the chapter, the state-of-the-art algorithms are considered and their results are shown in order to make some comparisons. In particular it will be analysed algorithms based on empirical formulas such as those presented in Vagle90[1], Nystuen11[8], Pensieri15[2], Shaw78[9], Cazau19 [3] but also supervised ones such as Taylor21[6].

4.1 Algorithm analysis

Before starting to describe in detail the techniques and algorithms used to obtain the results shown in Table 4.I, it is important to describe the validation methods used to choose the optimal hyperparameters and so the optimal models. In the table, the best results for each technique are shown. In this study aiming to compare prediction performance of the statistical models, the first step of obtaining the best hyperparameters is to use a cross validation. The dataset is first divided in 10-fold with stratification in dataset partitioning. So at every cycle the dataset is divided in training (9 fold) and test parts (1 fold) in a cyclical way. Then considering the training set, that is composed of 9 folds, at each cycle 8 folds compose the training set and one the validation set. So every cycle evaluates one particular set of hyperparameters and at the end of the 9 repetition on the validation the best set of hyperparameters are chosen and used to evaluate the performances using the initial 10-fold training/test set (Fig. 4.1). The data for a fold are selected in a random way but based on classes, in this way when a test set, representing the 10% of the complete dataset, is selected randomly, it is sure that for each class of rain or wind about the 10% will be put in the test set and 90% in training set. Differently from rainfall detection the technique used showed great changes when hyperparameters are subject to slight and even low variations; it is clear that that kind of model is very sensitive and in this case it is important that parameters are chosen in a precise way. Data processing is performed using the scikit-learn library for the Python programming language.

				WIND											
				Beaufort scale (m/s)											> 1000
				B0	B1	B2	B3	B4	B5	B6	B7	B8			> 100
				<0,3	0,3-1,5	1,5-3,3	3,3-5,8	5,8-8	8-10.8	10.8-13.9	13.9-17.2	17.2-20.7	tot		> 10
RAIN	mm/h	*No rain (R0)*	0-0,1	**0**	**1756**	**5553**	**4779**	**2884**	**1599**	**596**	**140**	**10**	17317		
		Light (R1)	0,1-2,55	**0**	**9**	**70**	**129**	**136**	**171**	**98**	**26**	**0**	639		
		Moderate (R2)	2,55-7,55	**0**	**0**	**7**	**15**	**47**	**50**	**43**	**16**	**1**	179		876
		Heavy (R3)	7,55-50	**0**	**0**	**3**	**5**	**10**	**18**	**15**	**7**	**0**	58		
			tot	0	1765	5633	4928	3077	1838	752	189	11	18193		

Fig. 4.1 Subdivision of the dataset in classes of wind and rainfall intensity. For what concerns rain, 4 intervals are considered, from R0 (No rain) to R3 (Heavy rain). For wind, Beaufort scale is considered and so 9 classes are used, from B0 to B8.

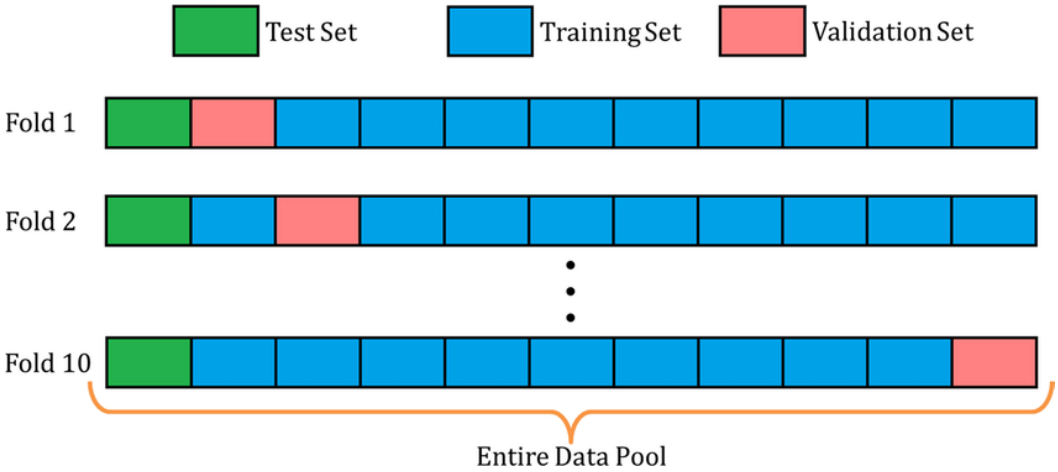


Fig. 4.2 10-Fold cross validation graphical example

	RMSE [m/s]	*MAE [m/s]*	*ME [m/s]*
Linear Regression (64 frequency bins)	2.45 (±0,2)	0.95 (±0,03)	-0.001
Linear Regression (30 DL coefficients)	2.30 (±0,1)	0.92 (±0,03)	0.002
Polynomial Regression (16 DL coefficients/ Degree 3)	1,15 (±0,03)	0.80 (±0,01)	-0,00005
Polynomial Regression (64 frequency bins- degree 2)	1,27 (±0,05)	0.85 (±0,02)	-0,00003
MFCC(15 coefficients/ Degree 2)	1,28 (±0,05)	0.86 (±0,02)	-0,00002
GTCC(10 coefficients/ Degree 3)	1,27 (±0,04)	0.85 (±0,02)	-0,00003

Table 4.I. Performance of the predictors. For RMSE, MAE, ME

In Table 4.I the best results for each technique are shown. It can be noted that initially a simple linear method was used just to obtain a preliminary result to be compared with the followings. It was immediately clear that the usage of Dictionary Learning preprocessing and of the coefficient collected in the so called Code matrix permits the algorithm used to achieve better results than the ones obtained applying directly the frequency bins. In fact using the simple linear regression with 30 coefficients as features decreases the RMSE of 0.15 m/s, even if the results remain not good. The best results are obtained for polynomial regression using 16 dictionary coefficients and third degree. First the DL coefficients are extracted using as input all the 64 bins, then a polynomial preprocessing is applied to the new dataset 18193x16 and a simple linear regression is used to fit the model. The performances strongly increase and the RMSE is 1.15 m/s, with a reduction of 1.3 m/s from the previous linear model. Also the MAE value decreases to 0.80 m/s from 0.95 m/s in the previous case. Good results are shown by applying a bank of filters to extract MFCC or GTCC and fitting a polynomial model. First a bank of filters is applied to the 64 bins, using classical MFCC or GTCC computation, and a number of coefficients corresponding to the number of filters in the bank is generated. At this point polynomial regression is applied and models are fitted. The RMSE values are 1.28 m/s for MFCC and 1.27 m/s for GTCC. In the first case 15 coefficients and a second degree polynomial is used and in the second case 10 coefficients and a third degree polynomial is considered. The results are satisfying but worse than the previous one and at the same time are good to show and demonstrate that techniques already used for speech recognition could be applied also for wind prediction with good performances. In general, all the models show unbiased behaviour and MAE results simply reflect RMSE performances. The use of a smaller number of features with respect to the 64 bins is useful also for obtaining good performances in terms of computational time and computation effort. In fact the time to obtain a model using polynomial regression with 64 bins is very high. For what concern the Dictionary Learning preprocessing it was useful to understand what could be the optimum parameters to be applied in order to fit a dictionary model for the extraction and calculation of the coefficients. The main idea was to estimate the best α regularization parameter in order to obtain the 'best' Code and Dictionary matrices from the DL algorithm. In order to understand which could be the meaning of the concept of 'best' matrices, it is useful to remember that multiplying Code matrix and Dictionary matrix the result is an approximation of the original dataset. So the best α (alpha) corresponds to the matrices from which the best approximation of the original dataset is obtained. Obviously considering the number of atoms, it is simple to understand that increasing it the approximation will be better and better, and it is also clear from the plot in Fig. 4.3. But for what concern that parameter, corresponding to the number of coefficients per spectra, it is better to optimize it considering directly the results of the cross validation process explained before. In Fig. 4.3 it is plotted a 3D surface representing the MSE between the original dataset and the approximation obtained multiplied by the Code and the Dictionary calculated by the DL model versus the parameters alpha and number of atoms. The MSE is calculated between the spectra in linear scale and not directly in dB scale. The unit of measure of the MSE, hence, is μPa^4. As said before, in general increasing the number of atoms the approximation is better. For what concern the alpha parameter the best results are obtained using alpha with a low value. So for all the following calculations the regularization parameter is taken equal to 1 in order to obtain Code and Dictionary matrices more similar to the optimum ones.

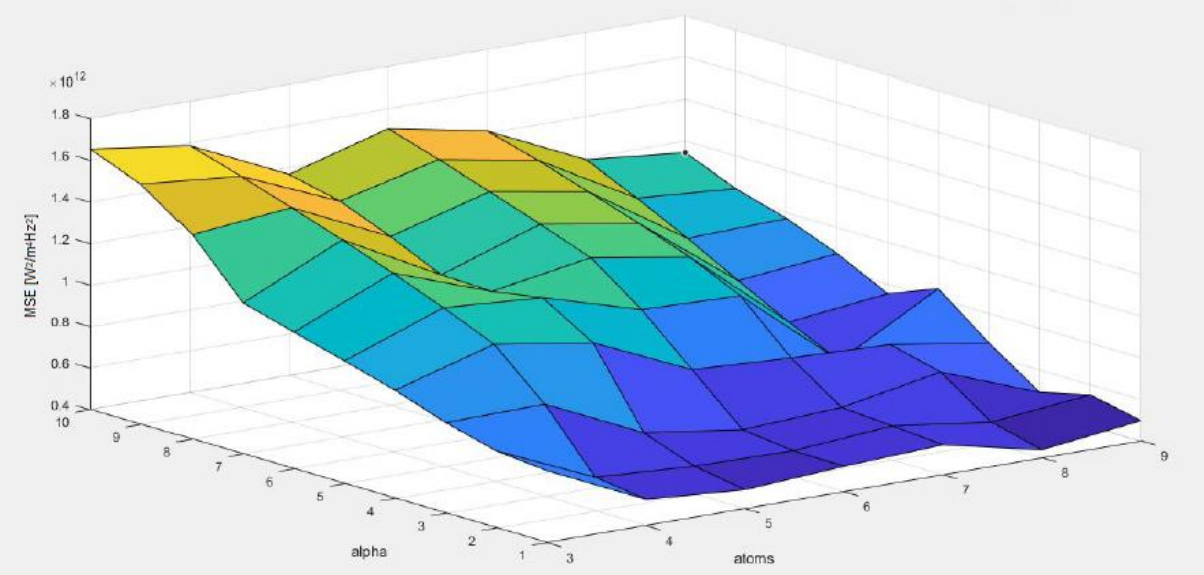


Fig. 4.3 Approximation error between the original dataset and the reconstructed one versus alpha and the number of atoms.

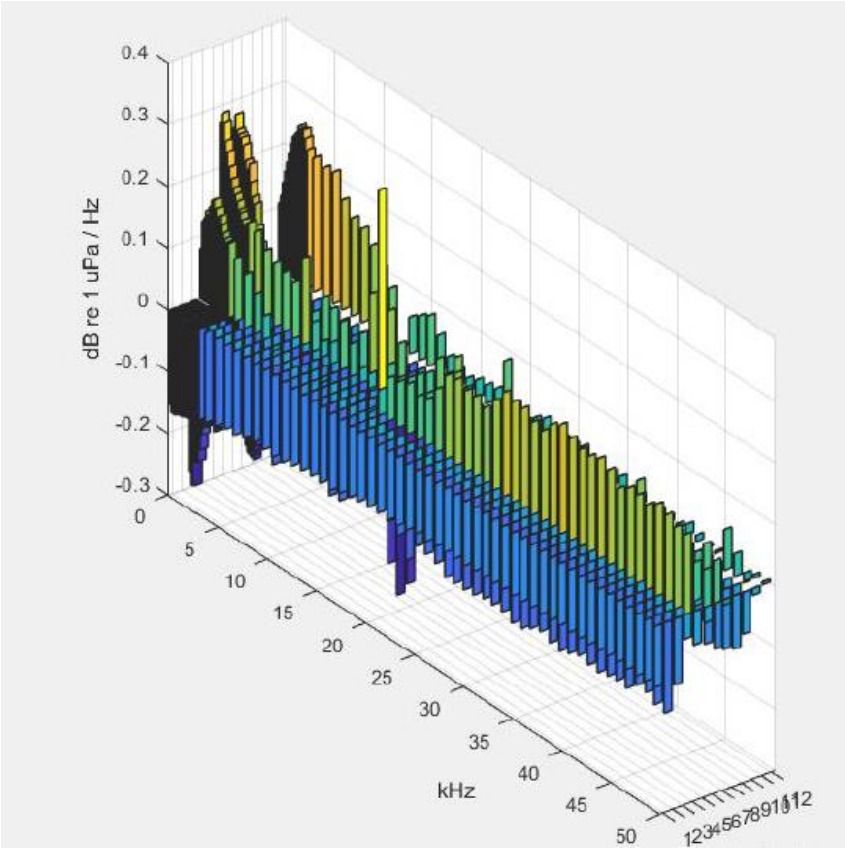


Fig. 4.4 12 dictionary spectra example in 3D representation

The usage of the DL coefficients as features instead the direct usage of the frequency bins is the key for the increasing of performance obtained for polynomial regression. The decreasing of the number of features and the compression of the information helps to obtain a more clear comprehension of the underlying scheme that binds data features to classes of rain and wind. In Fig. 4.5 there are three tables each one represents one of three averaged coefficients divided in classes obtained fitting a DL model with alpha equal to 1 and atoms equal to 3. In the figure the classes composed by a consistent number of samples are underlined in orange and only those classes are considered for that general analysis. Considering the first coefficient values in the underlined classes it can be noted that increasing the rainfall intensity the values increase in modulus. Instead considering wind intensity the coefficients have greater values in modulus in the medium classes and the values decrease for low intensity and high intensity classes, like a sort of Gaussian distribution. More complex is the second coefficient case, in which the behaviour looks exactly opposite of the preceding case. In fact the lowest and negative values are in the center classes and the values increase moving to the right and to the left. The distribution is a sort of reversed Gaussian. Analyzing the third coefficient table, it is clear that the values increase in a slightly linear way, increasing the rainfall intensity and wind speed.

	0	1	2	3	4	5	6	7	8
0	0	-308.119	-307.438	-330.529	-363.299	-380.171	-384.581	-377.712	-361.949
1	0	-360.625	-353.431	-372.443	-389.394	-394.836	-393.991	-386.274	0
2	0	0	-418.744	-412.259	-428.337	-423.829	-410.551	-398.481	-378.848
3	0	0	-381.035	-452.579	-468.37	-435.977	-423.263	-390.349	0

	0	1	2	3	4	5	6	7	8
0	0	24.4414	26.4227	18.6676	5.5116	-0.232562	1.6384	14.0462	28.8667
1	0	-0.778272	2.20809	-3.66306	-7.48256	-6.63627	-1.00179	10.0103	0
2	0	0	-29.7374	-22.8225	-27.739	-20.7987	-8.62474	1.73299	24.6233
3	0	0	-6.94266	-36.6418	-43.7434	-26.2748	-8.28846	12.3511	0

	0	1	2	3	4	5	6	7	8
0	0	-0.184472	1.34583	6.46324	11.6128	16.9357	23.4454	29.3348	33.7811
1	0	8.70335	6.69716	9.82176	16.3401	20.5424	26.084	32.1387	0
2	0	0	19.3696	19.2322	17.9068	23.3105	28.9815	32.3122	48.8356
3	0	0	7.16564	22.5675	28.4497	26.1443	37.9273	37.9104	0

Fig. 4.5 Averaged DL coefficients values divided per classes of wind and rain

4.2. In-depth results analysis and comparisons

The objective of this part will be to underline and analyse in a detailed way the results obtained. The results in analysis will be related to the optimum algorithm, i.e. Dictionary Learning preprocessing, extraction of 16 coefficients and using them as features given in input to fit a polynomial regression model. In addition, predictions with a value <0 m/s are carried to 0 m/s. In Fig. 4.7 there are two tables with values of RMSE and Absolute Percentage Error divided in classes. In the first table, considering only classes in which a consistent number of samples are present, it can be noted that the lowest values are related, in general, to classes in which there is a greater number of samples. For example, considering no rain case and B2 wind class, we have 5553 samples and a RMSE of 0.98 m/s. In part it is clear that considering a great number of spectra the model is well fitted for that class and can learn very well how to better recognize that kind of wind intensity. From another point of view it's also clear that having spectra without rain contribution could be simpler to discriminate between different wind classes or intensities. So the combination of those two characteristics, i.e. absence of rain noise and great number of samples available, keeps to good results and it is clear observing all the classes corresponding to no rain. B2 has the best results, but also B3 and B4 respectively with RMSE equal to 1.04 m/s and 1.09 m/s give very good performances and correspond to the 3 classes with more samples. Decreasing the number of spectra the RMSE increases, B1 has a RMSE of 1.47 m/s and B6 of 1.45 m/s. B5 again shows very good results with a RMSE of 1.18 m/s. When the rain contribution increases, also the number of samples per class decreases, so the RMSE values are higher. The worst values in classes with consistent numbers of spectra are represented by samples in (R2, B3) with RMSE value of 2.5 m/s and (R3, B6) with RMSE equal to 2.9 m/s; obviously related to low numbers of samples and high rain noise contribution. Instead (R2,B6) shows good values 1.14 m/s despite the theoretically non favorable conditions.

	0	1	2	3	4	5	6	7	8
0	0	1,4672	0,9759	1,0415	1,0872	1,1815	1,4485	2,1558	5,1142
1	0	2,9161	1,6640	1,4747	1,5705	1,5354	1,3431	1,3221	0
2	0	0	3,7122	2,5268	2,1414	1,4422	1,1405	1,7194	1,6264
3	0	0	0,9274	4,3058	2,8993	1,6958	2,8961	3,1749	0

	0	1	2	3	4	5	6	7	8
0	0	0	30,3774	18,5173	12,2263	9,9419	10,0283	11,5927	22,9501
1	0	0	51,6974	26,0025	16,8906	11,8641	8,5692	6,9590	0
2	0	0	134,4750	45,2958	23,8654	11,4264	7,5746	8,5506	9,1118
3	0	0	27,4087	78,9301	37,0431	14,0378	16,0686	14,4607	0

Fig. 4.7 Wind RMSE divided per classes of wind and rain in the first tab; Wind Absolute Percentage Error divided per classes of wind and rain in the second tab

The same consideration could be made for results in the second table. This table shows the Absolute Percentage Error values for wind >1.5 m/s divided per class. It is clear that computations for very low values of wind speed are neglected because of low prediction errors could bring to very high errors in percentage. Increasing the wind speed values, and so classes, the error decreases, because in general errors in absolute value don't increase so much with the increasing of speed and so the absolute percentage error decreases. Very good values again are shown by zero rain contribution, in class B5 with percentage equal to 9.9% and in class B6 equal to 10.0%. But the best results are shown for rain R1 and wind in B7 with a percentage of 6.9% and B6 with a error of 8.6% or also for R2 and wind in B6 with a value of 7.6% and B7 with 8.6%. This is a good result because it shows that high values for wind speed don’t lead to high prediction percentage errors, and so the error does not increase strongly. So the model can be considered stable and in some way shows some linearity in the increasing of the errors with respect to values to be predicted.

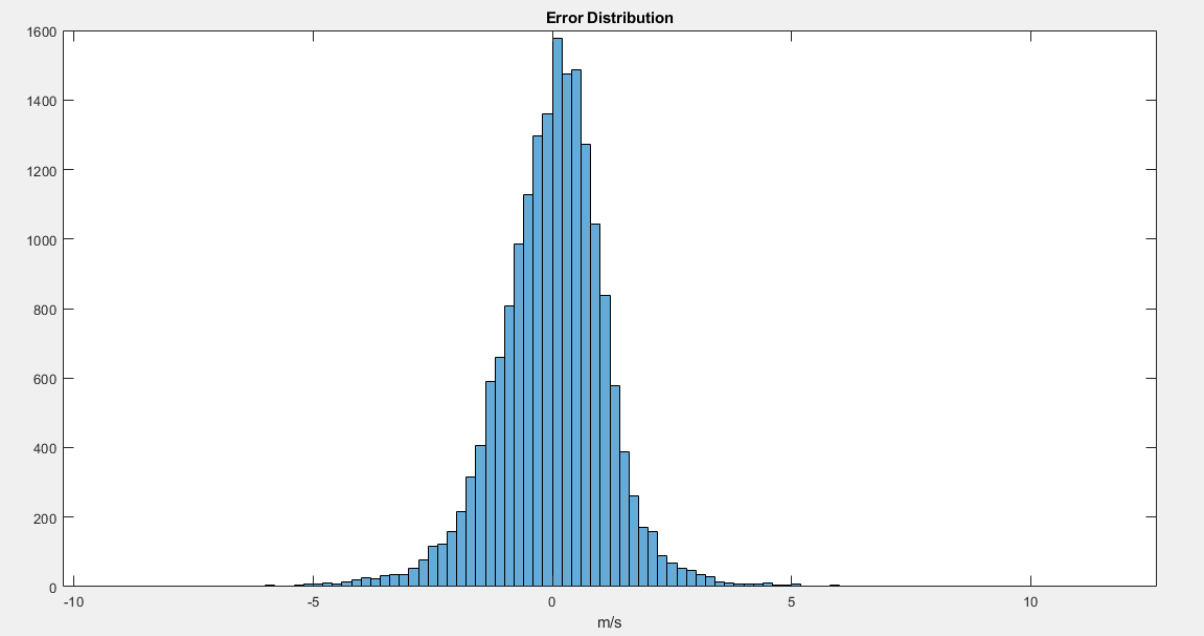


Fig. 4.8 Wind Error Distribution plot

In Fig. 4.8 it is shown the distribution of the errors. We know from Table 4.I that the Mean Error is very low and so the model could be considered unbiased. This new plot is important in order to understand how the error is distributed. It is simple to note that the distribution is a sort of Gaussian with low variance and so quite strict and centered in zero. The fact that the distribution is symmetrical and centered in zero is good to understand that there is no a privileged direction for what concerns prediction errors and so a very similar number of errors is related to negative and positive values. The fact that the bias is low and RMSE is quite small is a good indicator for the fact that the model fitted is a good model. A good bias-variance tradeoff is an indicator of good generalization properties and so a good probability in having no overfitting, but also not even an underfitting situation. So the model is not too simple and not too complex and the wind phenomenon is modelled, and also predicted, in a good way.

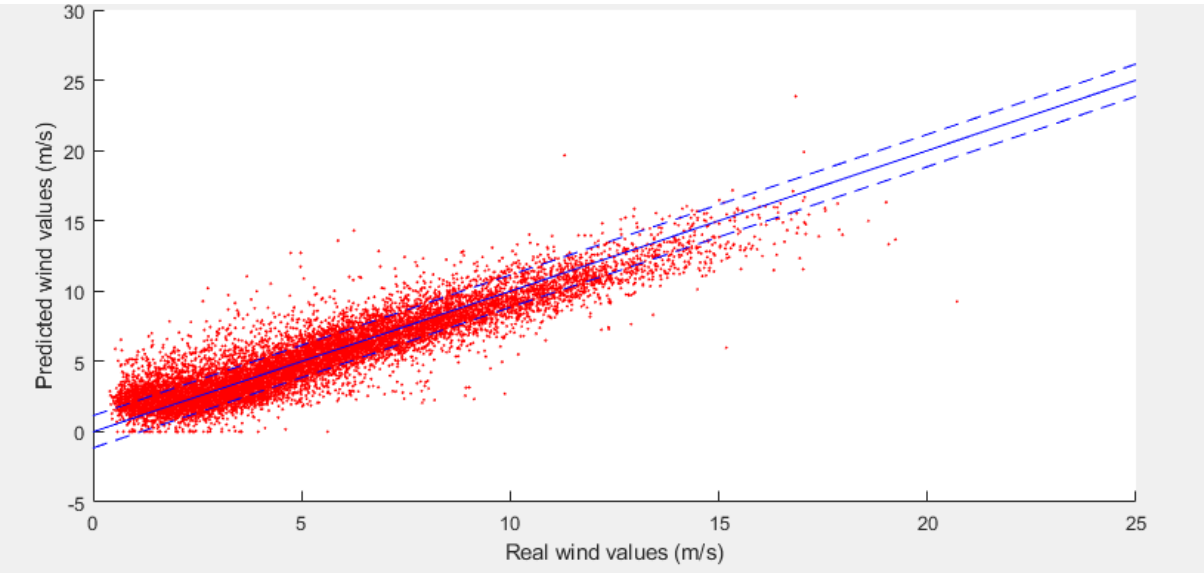


Fig. 4.9 Scatter plot with Real values versus Predicted values, the continuous line represents the bisector and the two dashed lines are the bisector shifted by ±RMSE

In the plot depicted in Fig. 4.9 predicted values versus real wind speed are represented. The continuous line is the bisector line describing the perfect prediction and the two dashed lines (bisector shifted by ±RMSE) are parallel to the bisector. It is clear that the distribution of the red dots follows very well the black line and the dots inside the dashed lines are the majority. Inside the two dashed lines there is a very great number of samples, about the 85%. Again the 3D distribution of the dots is a sort of Gaussian symmetric with respect to the continuous line and with a low variance value. That kind of behaviour is good in terms of precision of the prediction model. In Fig. 4.10 we can see the same scatter plot with a different scale and two lines, the blue one is the bisector and the green one is the line that best fits the predictions and it is clear that the two lines are very near each other. The plots in Fig. 4.11 (a) and 4.11 (b) can be used to make some considerations. The samples predicted through the thesis technique are much less dispersed than those predicted by [1], [18] and [2]. The number of samples for which the predicted wind speed is noticeably incorrect (especially in terms of overestimation) is reduced. The bisector and the best fit line are similar and near only in [2] and [6], but in the other two models they are very distant. The results made by [6] are similar to the one obtained by the thesis algorithm for what concern dispersion of the samples and similarity between the two lines (best fit line and bisector), but in the [6] case a low number of samples is depicted and the plot is cleaned from points too distant from the bisector. Instead in Fig. 4.10 all 18193 samples are shown.

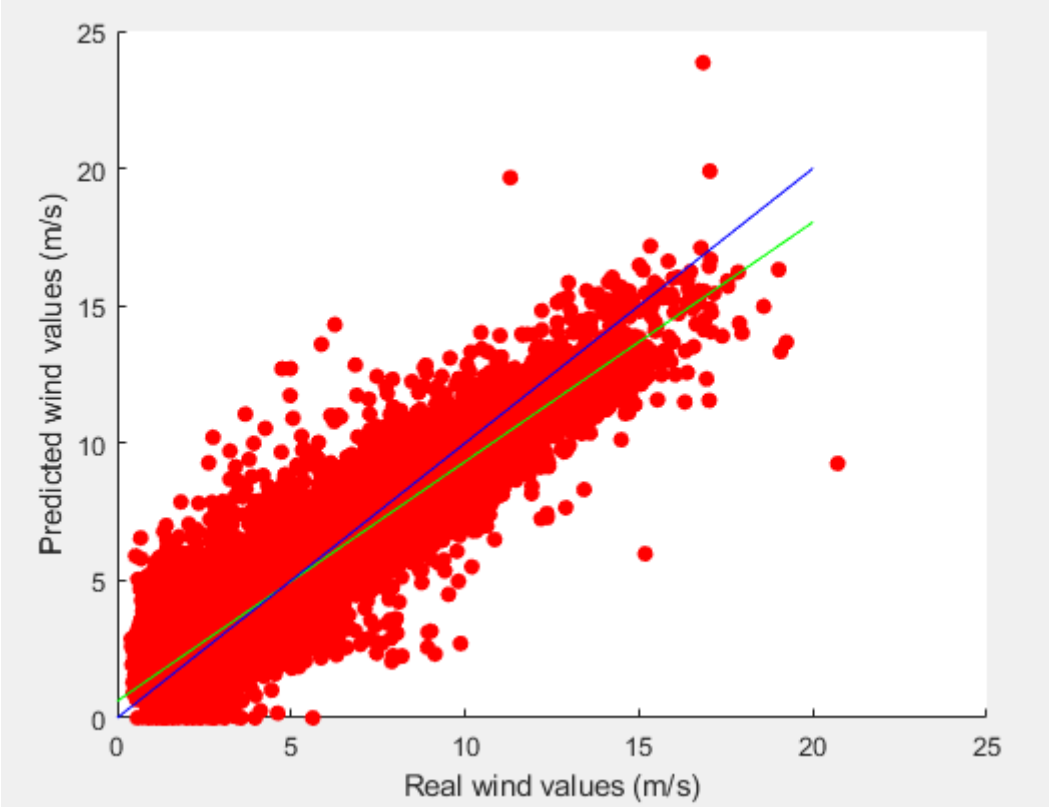


Fig. 4.10 Scatter plot with Real values versus Predicted values, the blue line represents the bisector and the green line is the best fit line.

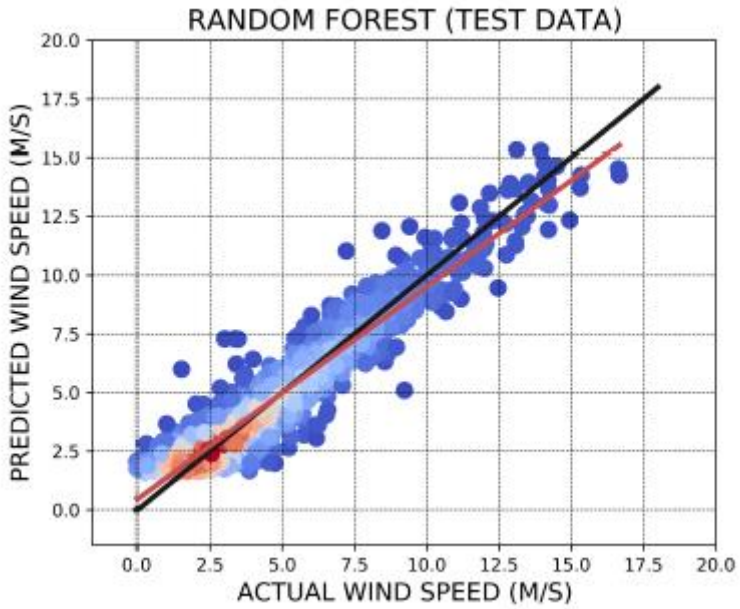


Fig. 4.11 (a) Scatter plot with Real values versus Predicted values from [6], the black line represents the bisector and the red line is the best fit line.

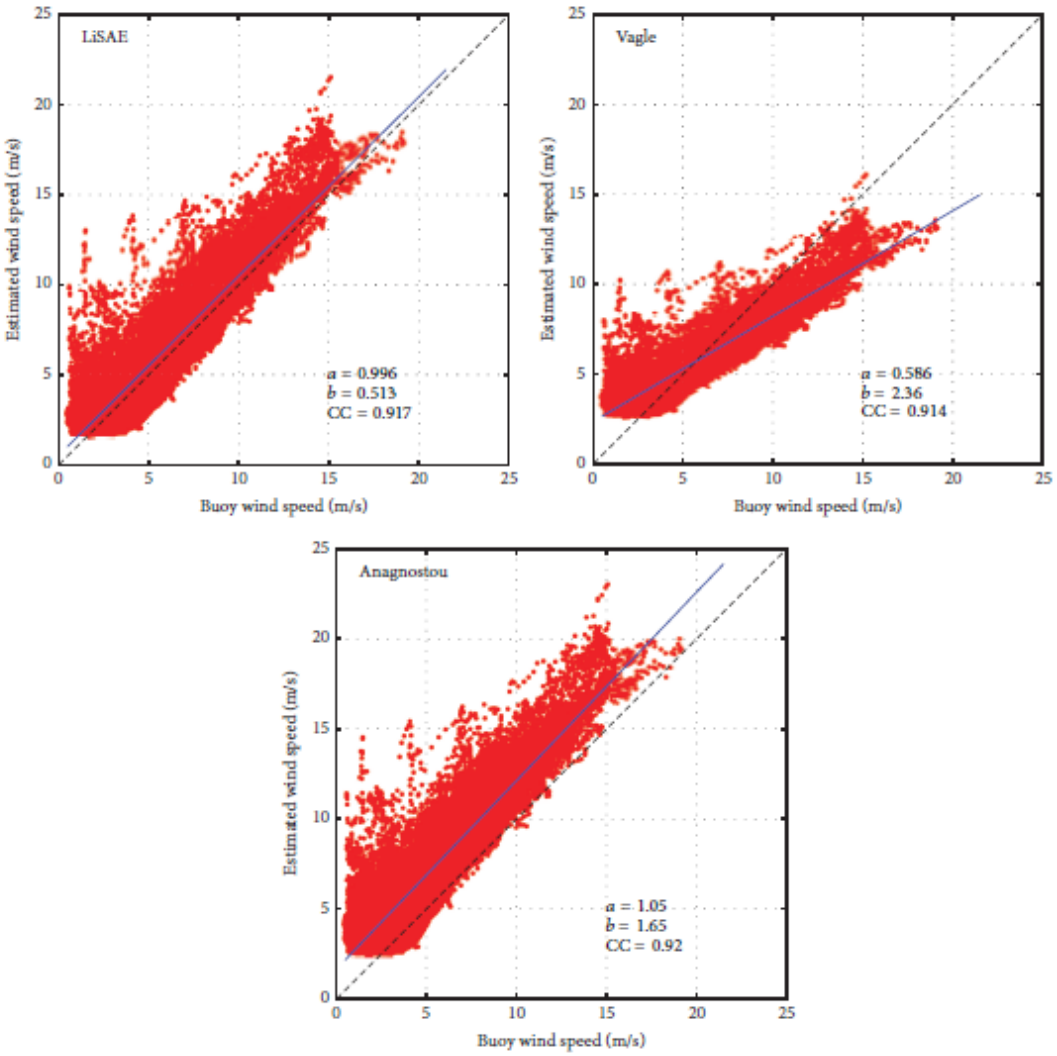


Fig. 4.11 (b) Scatter plots with Real values versus Predicted values from [2], the black dashed line represents the bisector and the blue line is the best fit line.

An important way to understand if the model is a good estimator is to plot, as in Fig. 4.12, in a chronological way all the 18193 real wind label values and the predicted ones together. In the plot all the samples together are depicted and it can be preliminary understood that the predictions, in general for all the analysis period, follow very well the real wind speed. Obviously it is important to show some figures with higher resolution in order to see and describe the details. This is done in Fig. 4.13. In this figure we can note in detail that the predictions are quite precise and follow very well the real wind speed trend. A common error that happens is when the model predicts a fast increasing or decreasing of the speed but in reality the behaviour is quite smoothed. On the other side another kind of error is related commonly to rapid

changes of the real wind status in particular when the wind speed increases rapidly. In general this error could occur both when there are peaks both related to rapid increasing or decreasing cases. So it is clear that the model shows in some situations difficulties, but in general the behaviour of the prediction is satisfying.

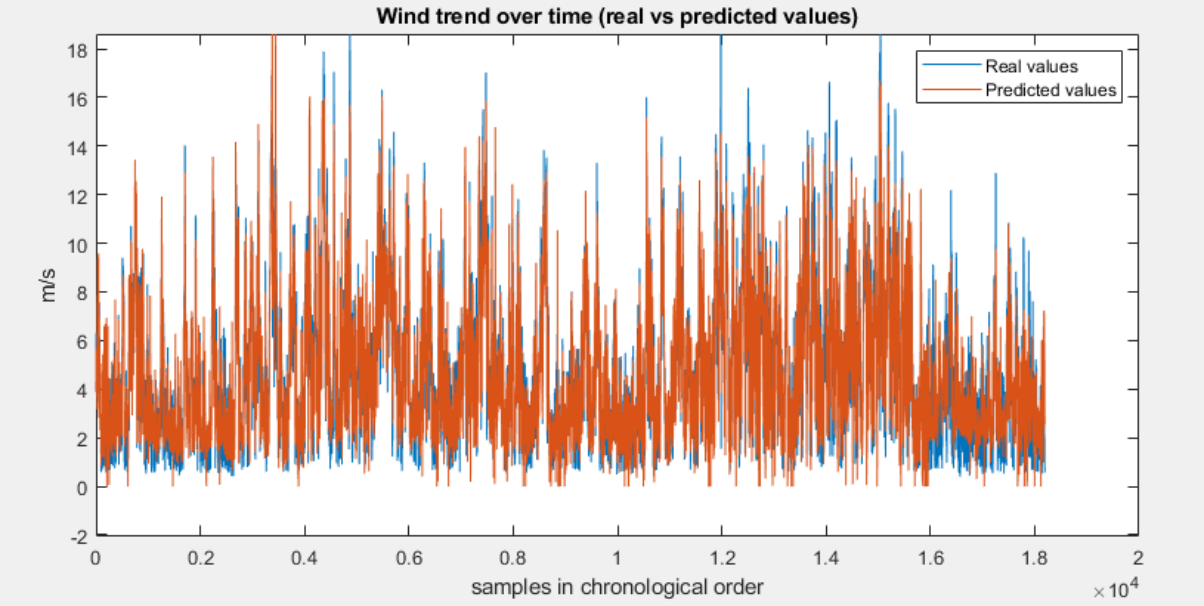
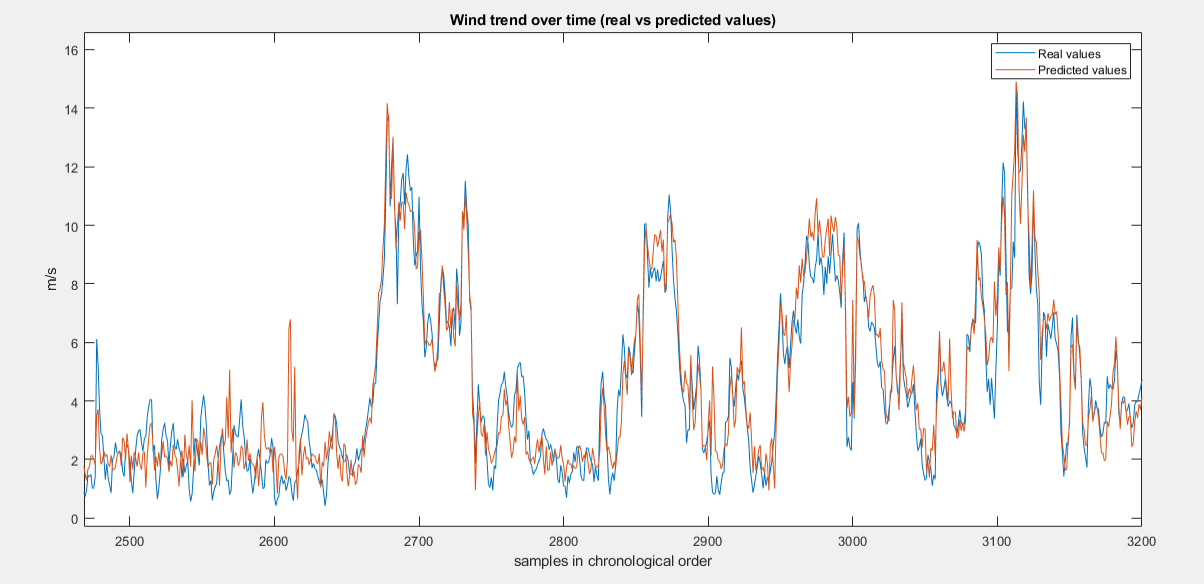


Fig. 4.12. Comparison between Real and Predicted wind speeds for the 18193 spectra of the dataset.



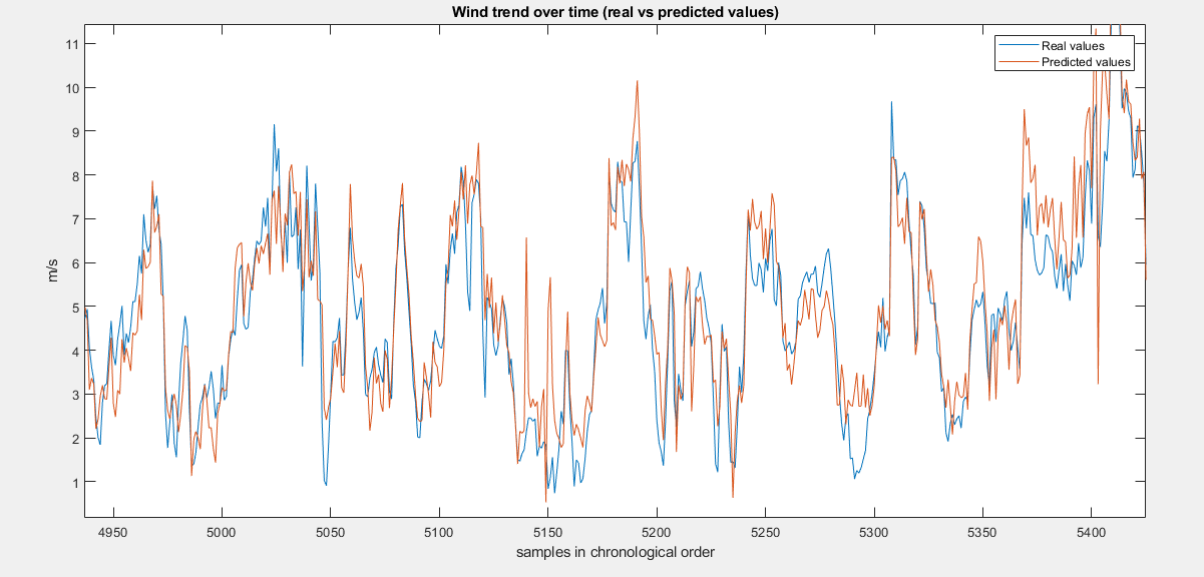


Fig. 4.13. Two detail fragments of the comparison between Real and Predicted wind speeds for the 18193 spectra of the dataset

4.3. Performance of the literature algorithms

In this paragraph of the chapter the objective will be to show and analyse the algorithms introduced in the Introduction Chapter that are part of the state-of-the-art related to wind prediction. The most important techniques were applied to the available dataset and the results are summarized in Table 4.II. It is important to recall that most of these algorithms were designed to predict wind using short-term acoustic spectra, whereas in this study they are applied to hourly-averaged spectra. In order to obtain results comparable to the one proposed in Table 4.I, the same identical 10-Fold cross validation scheme is used, for supervised techniques, and the same indices are computed.

	RMSE [m/s]	*MAE [m/s]*	*ME [m/s]*
Vagle90[1]	2.02	1.36	0.28
Nystuen11[8]	3.41	1.82	1.58
Pensieri15[2]	2.43	1.25	0.45
Cazau19[3]	4.96	2.03	-0.01
Taylor21[6]	1.11	0.78	0.02

Table 4.II. Performance of the state-of-the-art predictors. For RMSE, MAE, ME.

The wind prediction techniques in literature based on empirical formulas have not provided good performance on averaged spectra and Machine Learning methods have shown the best performances. The polynomial regression with 16 DL coefficients performs better than the best algorithm, decreasing the RMSE of about 2.3 m/s for what concerns [8], about 0.9 m/s for [1] and 1.3 m/s for [2]. The same happens for MAE values, that decreases of about 1 m/s for [8] and of about 0.5 m/s for the other two. The same could be said for the algorithms using MFCC or GTCC that decreases the RMSE of about 2.1 m/s for what concern [8], about 0.8 m/s for [1] and 1.1 m/s for [2]. Not good solutions are proposed by [3]. In that case RMSE is 4.96 m/s and MAE value increases to 2.03 m/s. Again those performances are very far from those achieved by the regression model proposed by this thesis. The same happens for example in the solution proposed by [6] in which Machine Learning methods are used to predict wind speed using all the frequency bins of the underwater noise spectrum as input data. Again in [6] Machine Learning methods are applied for wind prediction using hourly-averaged spectra as input data. In that case Random Forest with 50 trees was used. The performances are quite similar. RMSE is 1.11 m/s and MAE value is 0.78 m/s, but it shows a greater bias value. That kind of comparison is essential to understand that the thesis results are consistent and correct, in fact different techniques applied to a similar dataset carry to the same result.

4.4. Final analysis and comparisons

The analysis done in this chapter is aimed to compare the performance of the wind prediction techniques described previously, including fixed empirical equations, models fitted to the data and supervised learning techniques, when applied to averaged spectral data. The results provided by all the models fitted to the data are obtained through a 10-fold cross-validation. The prediction accuracy metrics for each method applied to the available dataset are reported in Table 4.I and 4.II and their analysis leads to some observations. First, the polynomial regression with DL preprocessing shows significantly lower error values, considering RMSE and MAE, than the prediction techniques that use a single feature, both fixed or fitted to the data. Also good results are obtained using speech recognition techniques such as GTCC and MFCC together again with polynomial regression. The results in Table 4.II, for the state-of-the-art models that use only one feature, were obtained using frequency at 8 kHz, as in most of the literature. This analysis confirms the superiority of the prediction performed using a multiplicity of features and regression methods that belong to the Machine Learning domain. Considering [6] results are very similar to the thesis model, and it shows almost equal RMSE and MAE, but high ME. So at the end it can be concluded that the thesis analysis on wind prediction demonstrates that the application of ML techniques to averaged spectra could produce good performances. In fact previous algorithms for prediction in the state-of-the-art, applied to the same dataset composed of 18193 spectra, obtained not good results or in general worse than the supervised models. A possible idea could be to understand if the thesis predictor applied to spectra collected in different seas could obtain again good performances, but it is reasonable to assume that it would continue to operate successfully. The performance obtained working on averaged spectra suggests that Machine Learning models may also be useful for wind prediction using short-term acoustic spectra.

5. Rainfall prediction

5. Rainfall prediction

In the Introduction Chapter the importance of predicting rainfall intensity was highlighted through the applications that could be implemented thanks to this kind of analysis, both in navigation and in meteorological studies. For example for underwater navigations of submarines, in which it's needed to know the meteorological situation on the surface or also in situations in which is important to monitor natural phenomena, for example in relation to risk prevention. But in the past rainfall regression state-of-the-art algorithms had not good results and not good methodologies. This happened because they were based on algorithms not so adapted for this kind of analysis. The idea is to improve that results and so in the thesis rainfall prediction analysis based on Machine Learning algorithms is considered. The usage of supervised learning obtains good performances, in particular applying polynomial regression and Dictionary Learning together as in the previous wind prediction case. Using the same available dataset consisting of 18193 hourly-averaged spectra the best proposed algorithm obtains an RMSE of 0.48 mm/h and a MAE of 0.08 mm/h. In the first part the preprocessing algorithms and Machine Learning tools for regression used will be described such as, Linear Regression, Polynomial Regression, MFCC and GTCC preprocessings. In this part results are shown and algorithms are compared. Some indices or indicators are used to make comparisons between them, such as RMSE, ME or MAE. In the second part of the chapter results are compared using figures and plots describing well the information that could be analysed from the outputs of the processing. In the last part of the chapter, the state-of-the-art algorithms are considered and their results are shown in order to make some comparisons. In particular some algorithms based on empirical formulas will be analysed such as those presented in Pensieri15[2], Nystuen08[5] and Ma05[4] but also supervised learning models such as Taylor21[6].

5.1 Algorithm analysis

In Table 5.I the best results for all the algorithms tested are shown. The same validation methods used to choose the optimal hyperparameter in the wind prediction case are used again. Hence summarizing the algorithm for validation already explained in the preceding chapter, the dataset is first divided in 10-fold with stratification based on classes. So at every cycle the dataset is divided in training (9 fold) and testing parts (1 fold) in a cyclical way. Then considering the training set, that is composed of 9 folds, at each cycle 8 folds compose the training set and one the validation set. So every cycle evaluates one particular set of hyperparameters and at the end of the 9 repetition on the validation the best set of hyperparameters is chosen and used to evaluate the performances using the initial 10-fold training/test set (Fig. 5.1). The data for a fold are selected in a random way but based on classes, in this way when a test set, representing the 10% of the complete dataset, is selected randomly, it is sure that for each class of rain or wind about the 10% will be put in the test set and 90% in training set. Again, as for wind prediction, the techniques used are very sensitive to hyperparameters changes; so it is important that parameters have to be chosen in a precise way. Data processing is performed using the scikit-learn library for the Python programming language.

				WIND											
				Beaufort scale (intervalli in m/s)										> 1000	
				B0	B1	B2	B3	B4	B5	B6	B7	B8		> 100	
				<0,3	0,3-1,5	1,5-3,3	3,3-5,8	5,8-8	8-10.8	10.8-13.9	13.9-17.2	17.2-20.7	tot	> 10	
RAIN	intervalli in mm/h	No rain (R0)	0-0,1	0	1756	5553	4779	2884	1599	596	140	10	17317		
		Light (R1)	0,1-2,55	0	9	70	129	136	171	98	26	0	639		
		Moderate (R2)	2,55-7,55	0	0	7	15	47	50	43	16	1	179	876	tot. con pioggia
		Heavy (R3)	7,55-50	0	0	3	5	10	18	15	7	0	58		
			tot	0	1765	5633	4928	3077	1838	752	189	11	18193		

Fig. 5.1. Subdivision of the dataset in classes of wind and rainfall intensity. For what concerns rain, 4 intervals are considered, from R0 (No rain) to R3 (Heavy rain). For what concerns wind, Beaufort scale is considered and so 9 classes are used, from B0 to B8.

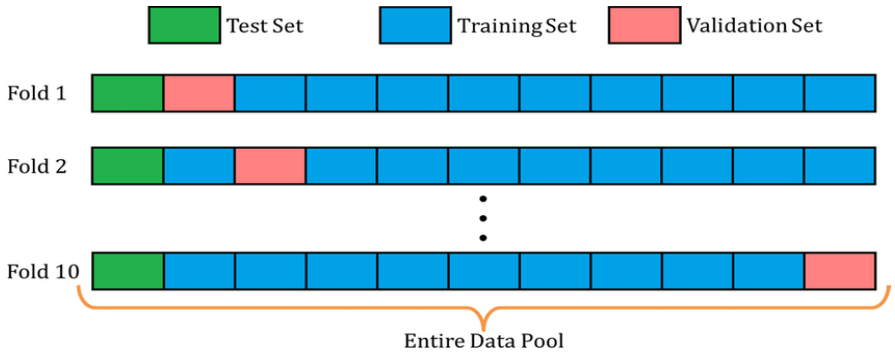


Fig. 5.2 10-Fold cross validation graphical example

	RMSE [mm/h]	*MAE [mm/h]*	*ME [mm/h]*
Linear Regression (64 frequency bins)	0.89 (±0,22)	0.3 (±0,05)	0.2
Linear Regression (30 DL coefficients)	0.78 (±0,19)	0.28 (±0,04)	-0.1
Polynomial Regression (12 DL coefficients/ Degree 3)	0,50 (±0,12)	0.14 (±0,02)	-0.0709
Polynomial Regression (12 DL coefficients/ Degree 3) with threshold 0.6 mm/h	0.48 (±0,12)	0.08 (±0,01)	0.0006
Polynomial Regression (64 frequency bins-degree 2)	0,7 (±0,25)	0.2 (±0,06)	-0.09
MFCC(20 coefficients/ Degree 3)	0,58 (±0,15)	0.2 (±0,06)	0.1
GTCC(10 coefficients/ Degree 3)	0,59 (±0,14)	0.19 (±0,05)	0.1

Table 5.I. Performance of the predictors. For RMSE, MAE, ME

The procedure used during the study is the same used in the wind prediction case. The studies proposed by the thesis in the prediction cases are led in parallel, because wind and rain show a strong parallelism for what concerns results and algorithms. So first a simple linear method was used just to obtain a preliminary result useful for comparisons and starting point. Then, Dictionary Learning preprocessing was introduced before linear regression and the usage of the DL coefficients of the Code matrix was useful to obtain better performances. In fact, if linear regression, that uses the frequency bins, has an RMSE of 0.89 mm/h, the performance increases and with 30 coefficients the RMSE becomes 0.78 mm/h. A good improvement but not the best possible. Then polynomial regression using 12 dictionary coefficients and third degree was introduced. First the DL coefficients are extracted using as input all the 64 bins, then a polynomial preprocessing is applied to the new dataset 18193x12 and a simple linear regression is used to fit the model. Putting to 0 mm/h all predictions lower than zeros the RMSE became 0.50 mm/h and the MAE 0.14 mm/h. A further improvement of the performances is obtained using a threshold of 0.6 mm/h. In this case predictions lower than 0.6 mm/h are put to 0 mm/h. So the RMSE decreases to 0.48 mm/h and MAE to 0.08 mm/h. Again as for wind prediction suboptimal results are shown by applying a bank of filters to extract MFCCs or GTCCs and fitting a polynomial model. First a bank of filters is applied to the 64 frequency bins, then, using classical MFCC or GTCC computation, a number of coefficients corresponding to the number of filters in the bank is generated. At this point polynomial preprocessing is applied and models are fitted. The results are good, but not the best performances are obtained; the RMSE is 0.58 mm/h for MFCC and 0.59 mm/h for GTCC. At the same time those two models are good to show and demonstrate that techniques already used for speech recognition could be applied also for rainfall prediction with good performances. In general all the models shown unbiased behaviour and MAE results simply reflect RMSE ones. The reduction of features with respect to the 64 bins is useful to achieve good performances in terms of computational time and computation effort. In fact the time to obtain a model using polynomial regression with 64 bins is very high. Considering Dictionary Learning preprocessing the same considerations done in the previous chapter can be made. The same process and analysis for the parameter alpha was made and is valid for the rainfall prediction case. So, as said before, in general increasing the number of atoms the approximation is better. For what concern the alpha parameter the best results are obtained using alpha with a low value. For all the following calculations the regularization parameter is taken equal to 1 in order to obtain Code and Dictionary matrices more similar to the optimum ones.

4.2. In-depth results analysis and comparisons

The objective of this part will be to underline and analyse in a detailed way the results obtained. The results in analysis will be related to the algorithm that implements Dictionary Learning preprocessing, extracts 12 coefficients and uses them as features to fit a polynomial regression model of third degree. In addition, predictions with a value <0 mm/h are carried to 0 mm/h. Again as in the previous chapter, two tables in Fig. 5.3 represent the RMSE divided per class and the Absolute Percentage Error for rain >2.55 mm/h divided per class. In the second table only values greater than a certain threshold are computed, this is because too low values of rainfall could make the percentage error grow exponentially and so are not suited for that kind of index evaluation. Considering the first table and in particular classes with a consistent number of samples, it can be noted that in general increasing the number of spectra, performances increase. In part it is clear that considering a great number of spectra the model is well fitted for that class and can learn very well how to better recognize that kind of rainfall intensity. Increasing the rainfall intensity the RMSE values increase and if for (R0,B1) or (R0,B3) there are very good results, 0.22 mm/h and 0.20 mm/h respectively, for (R1, B4) the value increases to 0.8 mm/h and the worst results are for (R3,B5), 4.6 mm/h, and (R2,B4), 2.0 mm/h. The availability of samples in some way precludes the possibility to have uniform RMSE in all classes, but performances decrease rapidly increasing the rainfall intensity and then also decreasing the number of spectra. Differently from what can be observed in wind prediction case, increasing the other component intensity, in this case increasing wind speed, there is not an important decreasing of performances. Also when the wind intensity is high there isn't a consistent increasing of RMSE but the values remain stable and the error seems to depend only on rain classes and number of samples. Regarding the results in the second table, the values are quite regular and uniform, again it can be concluded that the results for rain are not dependent on wind intensity, in fact changing wind class there is not an important increasing or decreasing in the values reported. The fact that the percentage error remains quite constant when rainfall intensity increases is good, because it means that high errors are done in high value samples and small errors in small value samples.

	0	1	2	3	4	5	6	7	8
0	0	0,2166	0,2263	0,2015	0,2244	0,3540	0,4999	0,7743	1,3912
1	0	1,3025	0,9710	0,6172	0,8093	0,9550	0,5704	1,7159	0
2	0	0	1,9309	2,5590	2,0323	1,7934	2,0410	1,9967	0,2418
3	0	0	8,5720	4,5614	6,7254	4,6273	5,9763	12,0694	0

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	0	37,4245	52,9907	40,0550	30,4366	38,4593	37,8299	4,1052
3	0	0	71,2919	39,1742	35,3795	37,5127	35,9438	54,7688	0

Fig. 5.3 Rain RMSE divided per classes of wind and rain in the first tab (a); Wind Absolute Percentage Error divided per classes of wind and rain in the second tab (b)

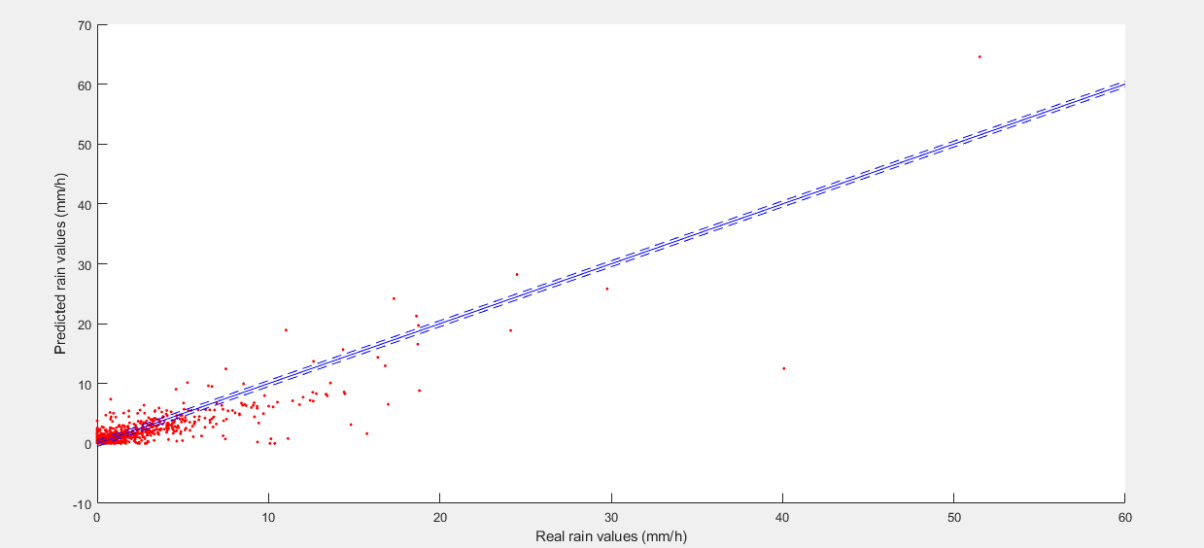


Fig. 5.4 Scatter plot with Real values versus Predicted values, the continuous line represents the bisector and the two dashed lines are the bisector shifted by ±RMSE

In Fig. 5.5 it can be analyse a scatter plot in which every red point represents real values versus predicted ones. The continuous line is the bisector line and the two dashed lines are bisectors shifted by ±RMSE. Observing the plot in detail in Fig. 5.6 there is a great number of points inside the two dashed lines (remember that the samples >0 mm/h are only 1277) and sparse samples outside that are the minority.

Increasing the intensity of rainfall the behaviour becomes more confuse, samples are more outside the double dashed lines and dots are not more near the bisector representing the perfect prediction line. This is in practice the visualization of what was observed in Fig. 5.3 (a), in which increasing the rainfall intensity the RMSE increases. It is also important to understand that the majority of points are very near or equal to zero (16916) and that points are well grouped near the bisector line and inside the two RMSE lines. Samples with higher intensity are very sparse and so it is obvious that the algorithm could have difficult to predict them but also it is good that high errors are made in high intensity samples and low errors in low intensity samples.

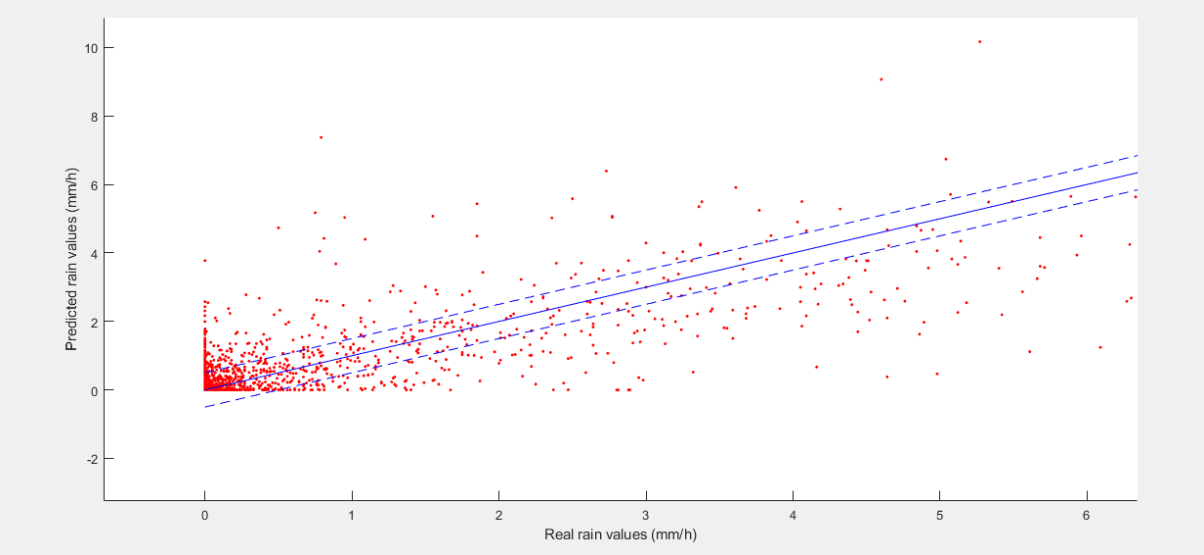


Fig. 5.5 Detail of Fig. 5.4

In Fig. 5.6 and Fig. 5.7 the same scatter plot is depicted with blue line representing bisector and the green line that is the best fit line. Observing Fig. 5.7 with high resolution it can be seen that for samples approaching zero the two lines are very near and they diverge for high value samples. Again the same consideration made before for low and high rainfall measurements can be said and so the regression model proposed is better fitted on low rainfall values. Although in general the divergence between the two lines doesn't increase so much and they are quite near in all cases.

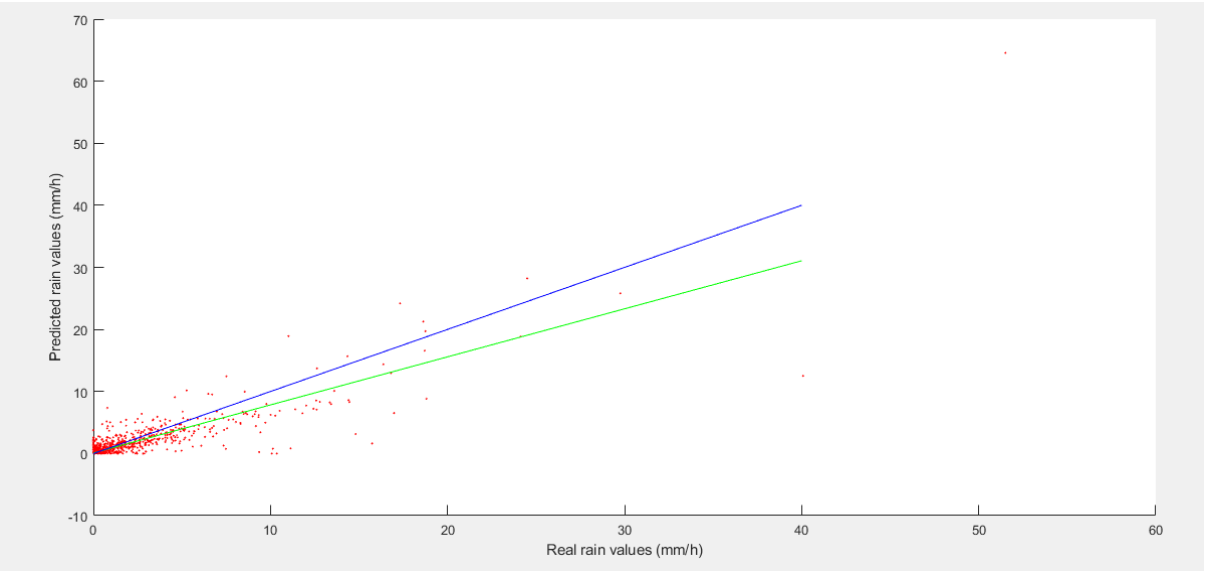


Fig. 5.6 Scatter plot with Real values versus Predicted values, the blue line represents the bisector and the green line is the best fit line.

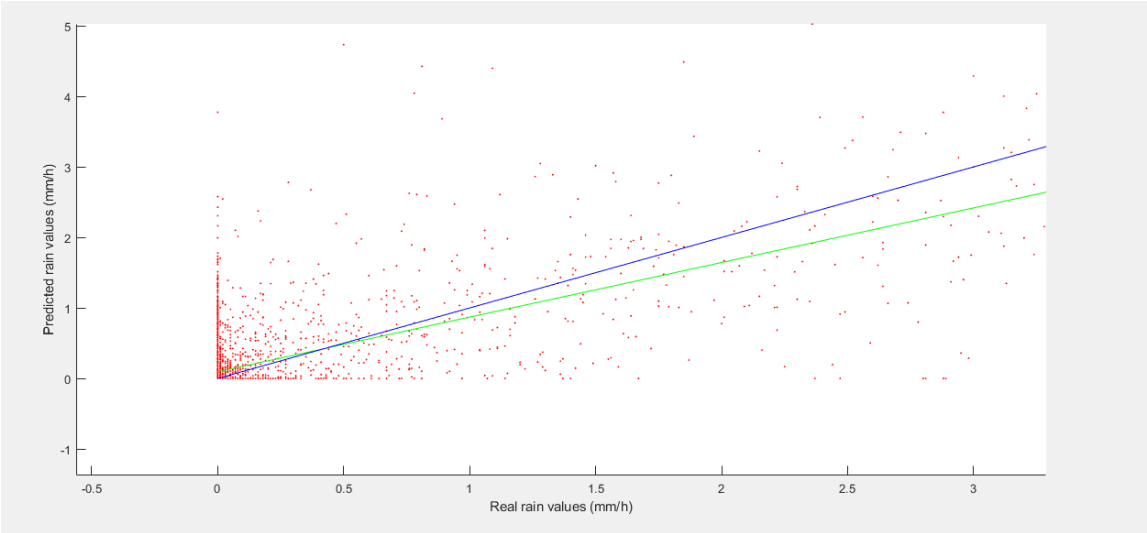


Fig. 5.7 Detail of Fig. 5.6.

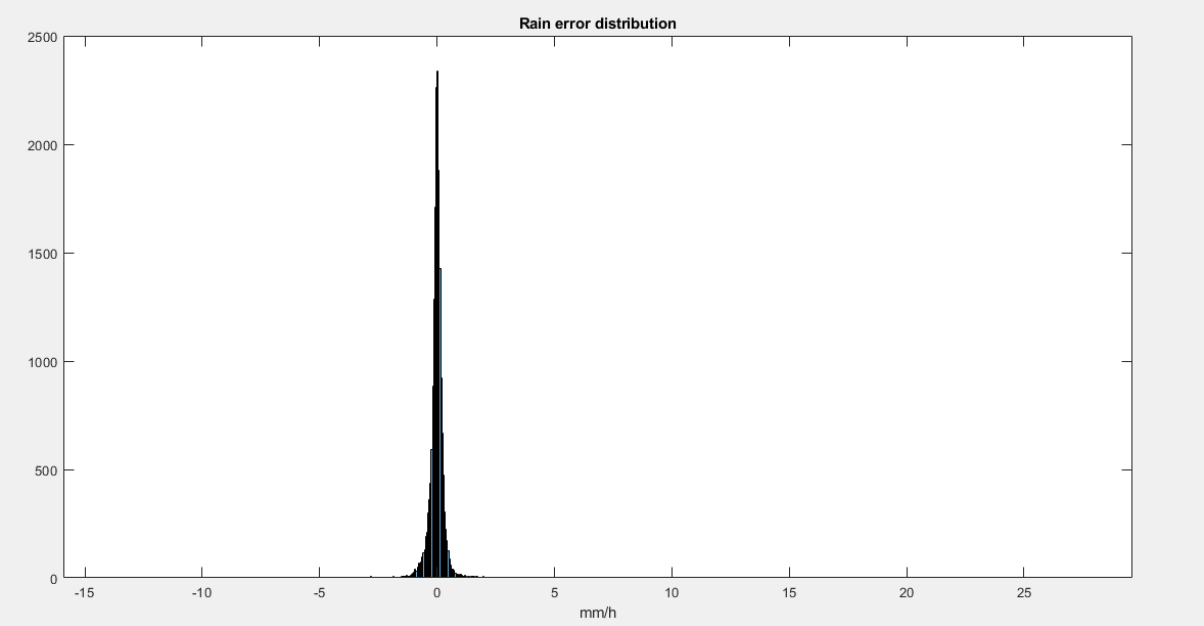


Fig. 5.8 Rain Error distribution plot

In Fig. 5.8 it is shown the distribution of the rain errors. It is clear from Table 5.I that the model fitted can be considered unbiased, because a very low ME is observed. It is simple to understand that the shape of the curve is a Gaussian with very low variance, and so very strict, and with centre in zero. The distribution is symmetrical and this kind of characteristic is good to understand that there is no privileged direction in overestimation or underestimation and so a very similar number of errors is related to negative and positive values. Also the low variance value indicates low error values in general. The fact that the bias is low and RMSE is quite small is a good indicator for understanding that the model fitted is a good model and in particular that good bias-variance trade off is obtained. So the model fitted is not too simple and not too complex and at the same time there is no overfitting or underfitting. In Fig. 5.9 it can be seen the rainfall trend over time for real values and predictions. From this figure it can be observed only the trend from a general point of view and not in detail. The general real trend is well followed. In Fig. 5.10 there is a most detailed image and it can be well analysed the small oscillations of samples near 0 mm/h. It is clear that there are a lot of small oscillations in prediction that don’t permit to precisely follow the real trend. From this kind of considerations was born the idea of using a threshold (for example of 0.6 mm/h) under which predictions are put to 0 mm/h. And in Fig. 5.11 it can be seen that the oscillation problem is quite solved and now also the trend for low values is better followed. This kind of reasoning is well explained and again demonstrated by Fig. 5.12, 5.13 and 5.14. In these three plots cumulative rainfall trends are depicted using different threshold values. It's clear from Fig. 5.12 that the curve representing cumulative trend of real values is not well followed by the one obtained with predictions using thr=0 mm/h. The two curves diverge and the predicted curve is overestimated because of that kind of small oscillations described before. Then imposing a threshold

of 0.5 mm/h in Fig. 5.13 and 0.6 mm/h in Fig. 5.14 better results are obtained and the two curves are overlapped in a quite precise way, in particular in the second case. From Table 5.I is clear that the model with thr=0.6 mm/h shows very good properties and results also for RMSE, MAE and ME.

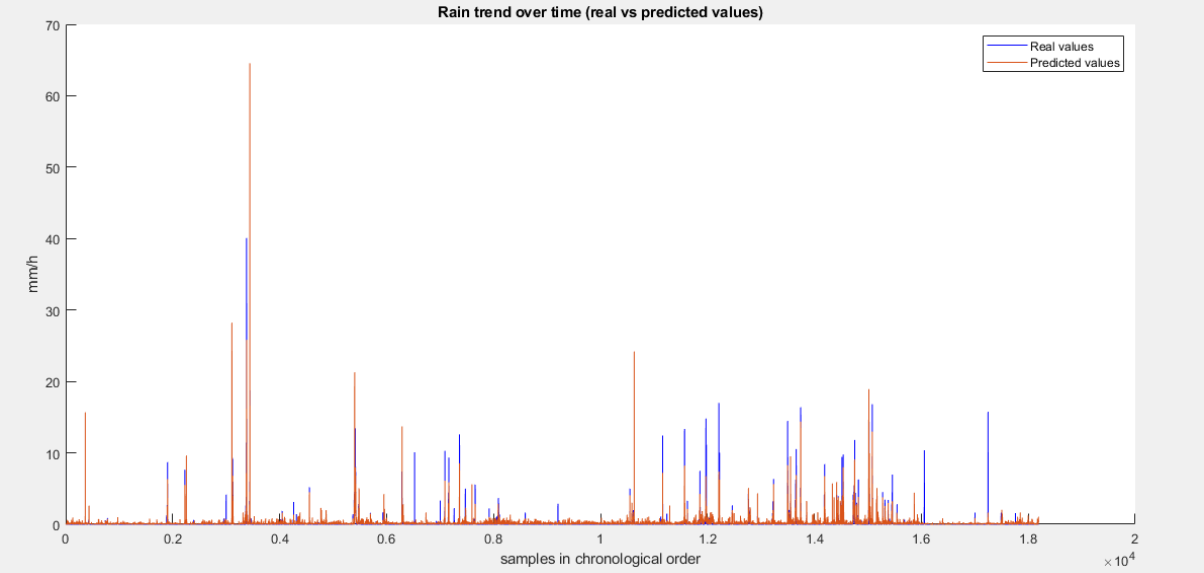


Fig. 5.9. Comparison between Real and Predicted rainfall intensities for the 18193 spectra of the dataset.

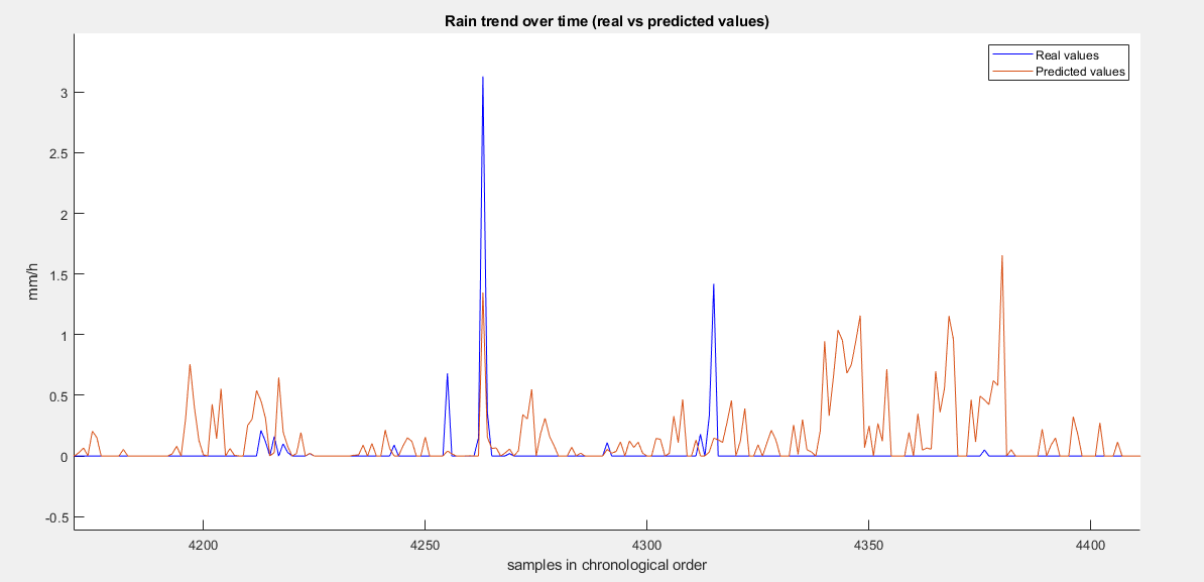


Fig. 5.10 Details of Fig.5.9.

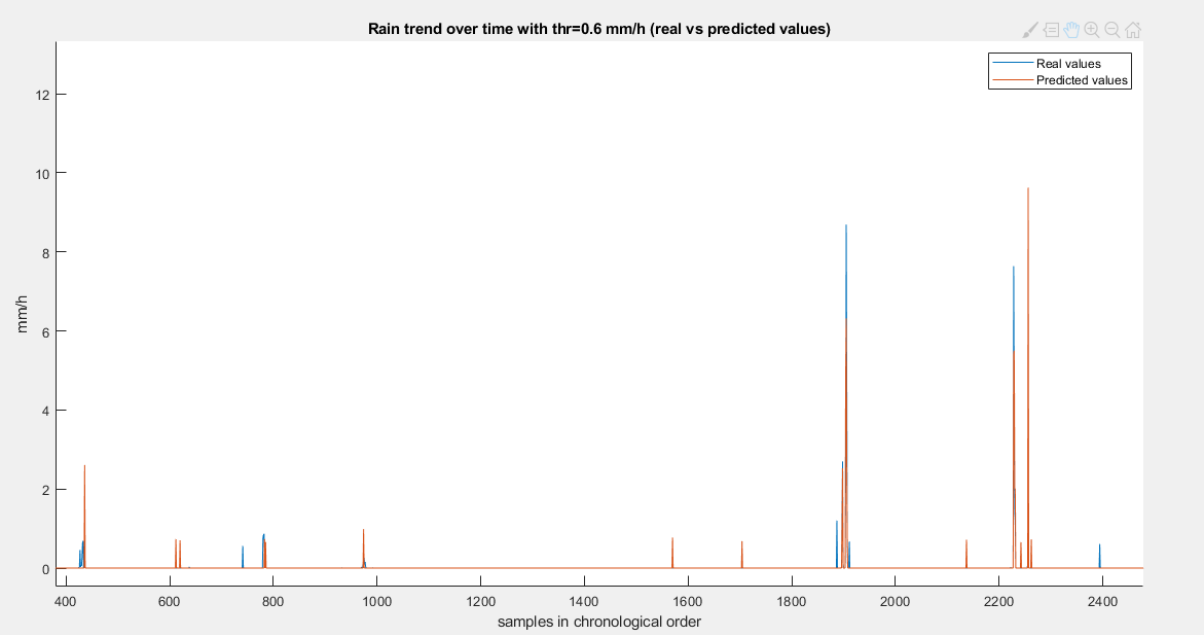


Fig. 5.11 Detailed comparison between Real and Predicted rainfall intensities for the 18193 spectra of the dataset with thr = 0.6 mm/h.

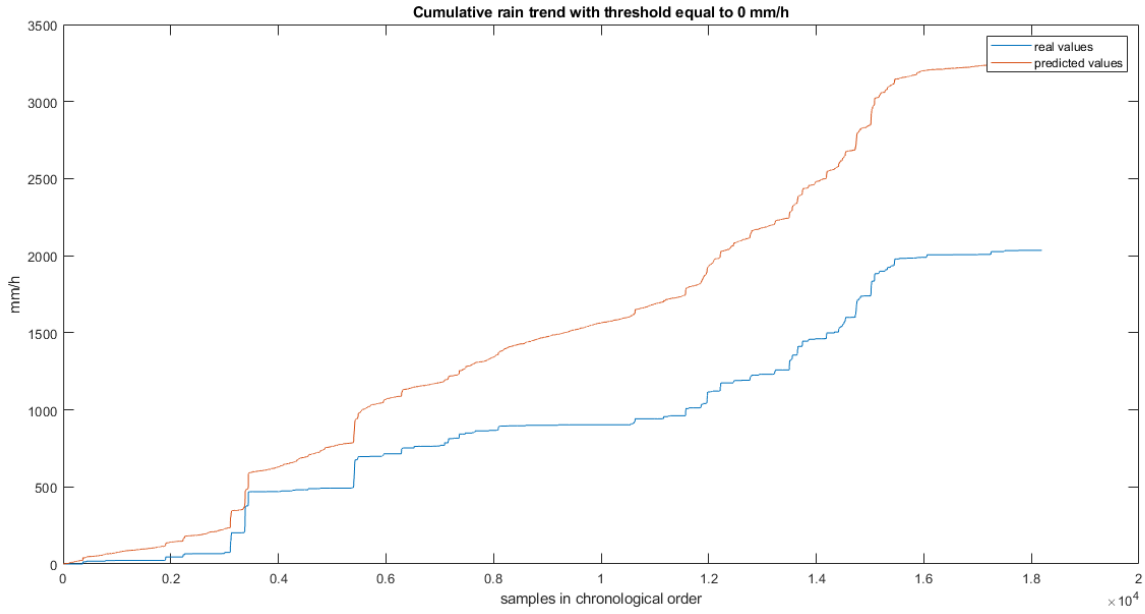


Fig. 5.12 Cumulative rainfall trend for Real values and Predicted values with thr=0 mm/h.

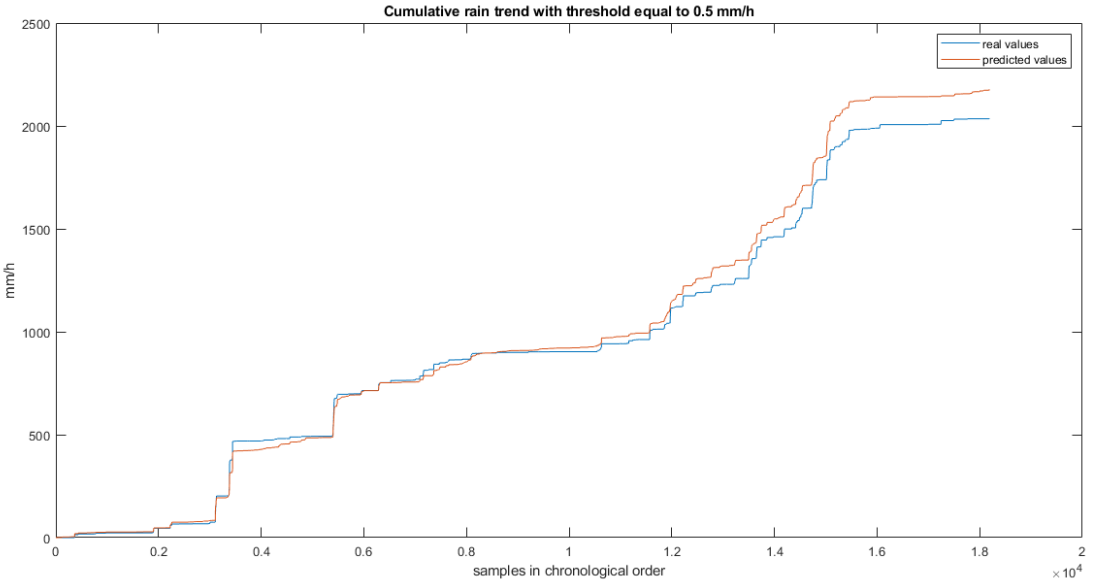


Fig. 5.13 Cumulative rainfall trend for Real values and Predicted values with thr=0.5 mm/h.

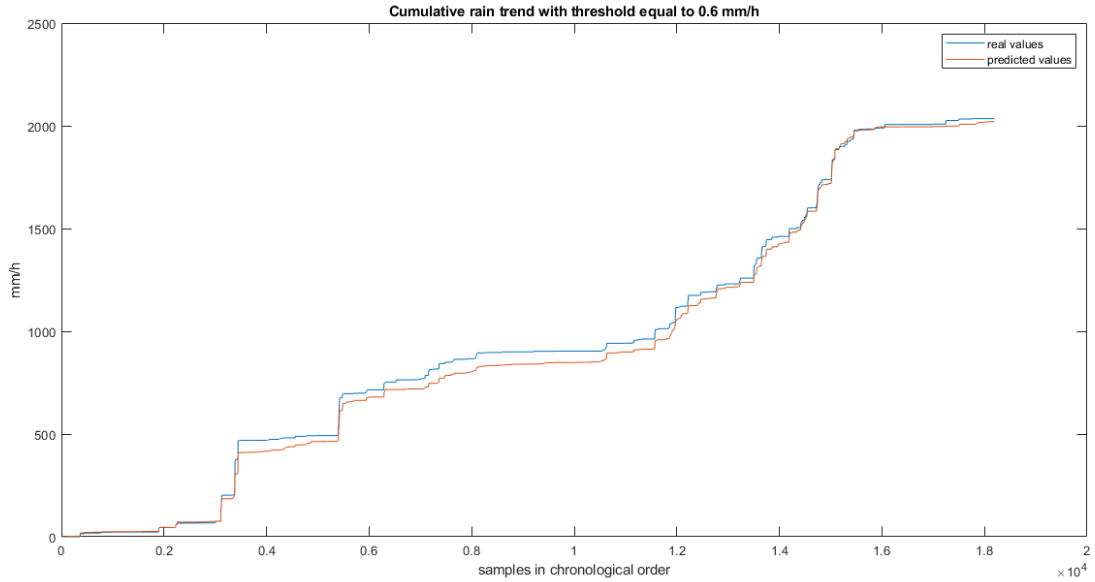


Fig. 5.14 Cumulative rainfall trend for Real values and Predicted values with thr=0.6 mm/h.

5.3. Performance of the literature algorithms

In this part of the chapter some state-of-the-art models are used to compute performances to be compared with the data contained in Table 5.I. In the rainfall prediction case there is a relatively low number of formulas or techniques in state-of-the-art. In fact that kind of analysis during the previous years was considered very difficult and results were not satisfactory for possible applications. Also in many studies data were cleaned by 'difficult' samples in order to obtain better results. In this case that kind of comparison is made computing the same indices of Table 5.I using the entire dataset available and using the same cross validation schemes, in the case of supervised learning techniques. As seen in the introduction part, [5], [2], [4] and [6] studies will be analysed. In the first three cases empirical formulas based on one frequency (5 kHz) are computed. Considering [2] study there are two formulas, one valid for drizzle and another for rain >1 mm/h. For [5] and [4] there is only one formula. For what concern [6], Random Forest predictor is used again as for wind prediction but results are calculated considering only samples >0 mm/h

	RMSE [mm/h]	MAE [mm/h]	ME [mm/h]
Nystuen08[5]	1.6	1.37	1.27
Pensieri15[2]	0.98	0.32	0.12
Ma05[4]	3.95	2.9	2.8
Taylor21[6] (>0mm/h)	1.82	0.85	0.25

Table 5.II. Performance of the state-of-the-art predictors. For RMSE, MAE, ME.

Making a comparison between results in Table 5.I and 5.II, it is clear that all the predictors analysed by the thesis showed better results than the empirical formulas proposed by [4], [5] and [2]. [4] shows the worst results with an RMSE of 3.95 mm/h and a very large bias. Also [5] has not good performances, but better than [5]. [2] shows a lower bias, but RMSE is high, comparable with linear regression done in the simplest way (using 64 bins directly). In order to compare the thesis model with the one proposed by [6] it was needed to consider and compute the performances of the polynomial regression using 12 DL coefficients for samples with rainfall values greater than zero. In that particular case the RMSE increases to 1.76 mm/h and the MAE to 0.83 mm/h. Considering the RMSE, the performances are strongly better than the one proposed by [6]. The bias of the thesis model is 0.28 mm/h and so very similar to the correspondent in [6]. This could be considered a very good result, in fact, observing Fig. 5.15, it is clear that the dataset is composed of a relatively very low number of samples > 0 mm/h. So it is simple to understand that a model could have difficulties in

predicting high value samples because it has less opportunities to learn how to discriminate them. Also a consistent reduction in RMSE from other supervised learning in literature is a great result for what concern the effectiveness of polynomial regression and Dictionary Learning.

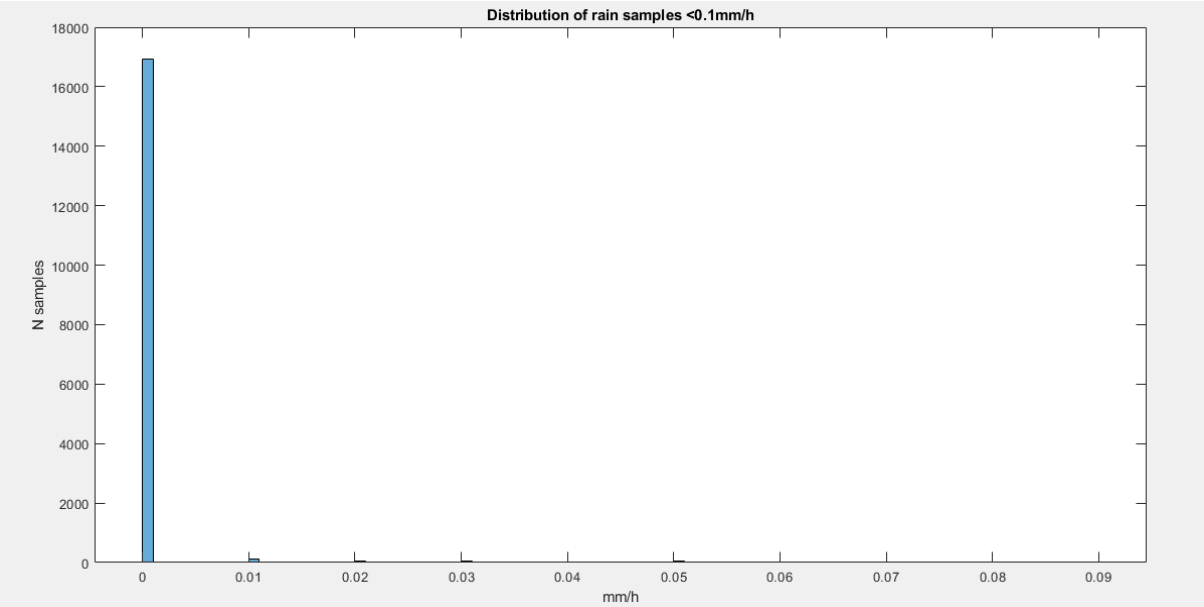


Fig. 5.15 Distribution of rain samples with intensity <0.1 mm/h

5.4. Final analysis and comparisons

This study has the objective of analysing the performances of the rainfall prediction algorithms described in the preceding paragraphs, including fixed empirical equations and supervised learning techniques, when applied to averaged spectral data. Some important considerations can be summarized; the polynomial regression with DL preprocessing has lower errors than all other techniques and RMSE and MAE values obtained by the thesis predictor are better than those obtained by state-of-the-art algorithms that use only the frequency at 5 kHz as feature. Good results are obtained using speech recognition techniques such as GTCC and MFCC, again with polynomial regression. Considering [6] result values are very similar to the thesis predictor, and it shows greater RMSE and MAE, but smaller ME. The study kept by the thesis shows and demonstrates that the predictions performed using all features available and supervised regression methods significantly improve state-of-the-art schemes performances, in particular the ones using empirical formulas. From a certain point of view supervised learning algorithms needs a training phase and a consistent number of underwater acoustic spectra has to be used but clearly in some way this is also partially necessary for other kinds of algorithms. For example when it is necessary to find optimum coefficient of an equation used for predictions and some kind of calibrations are needed. In general, it is important to recall that, as said in the rainfall detection part, no fluctuations in performance were observed on a seasonal basis and rainfall prediction was based on the amount of precipitation accumulated over the course of an hour, and it is not possible to determine whether this amount is due to transient, intermittent or continuous rain. A possible idea could be to observe dataset collected in different environments and understand if the thesis scheme obtains again good performances, but it is reasonable to assume that it would continue to operate successfully. As for wind prediction case, the performance obtained working on averaged spectra suggests that Machine Learning models may also be useful for prediction using short-term acoustic spectra.

6. Conclusions

6. Conclusions

This thesis concerned the possibility of detecting precipitation, from drizzle phenomena to events of high intensity, and predicting rainfall intensity and wind speed using the underwater acoustic noise spectra obtained from the average of the instantaneous spectra acquired, at various times, over the course of an hour. Since each sample is representative of an entire hour, to maintain sufficient temporal coverage it was necessary to analyze all the spectra acquired, even those altered by the passage of ships and other concurrent noises [11]. State-of-the-art methods for detection and prediction consider only few frequencies and try to compute empirical formulas. The Machine Learning algorithms proposed by the thesis are more accurate because they consider all the frequencies available together. The potential of this analysis is very high because it is not limited to only 1 or 2 features, so a more complete model could be fitted and could better be able to describe the phenomena in analysis. In this way better results are obtained. Also, the thesis demonstrates that averaged spectra can be used to obtain good results even better than the usage of short-terms spectra in state-of-the-art. Considering first the performances obtained in the rainfall detection part, it can be said that the literature methods didn't have obtained satisfying performances using the dataset available in the thesis. Instead Machine Learning methods have obtained good performances, better than previous ones and with the capability of having satisfactory results for possible applications. The best performances are obtained by kernel-based and ensemble-learning models, among the experimented supervised classifiers. In particular, the RF-based binary classifier has obtained a good tradeoff between computational burden and performance. For this classifier the P_d is greater than 90% when precipitation exceeds 0.7 mm/h and P_{fa} is 1% or, alternatively, when precipitation exceeds 1.4 mm/h and P_{fa} is 0.3%. This kind of method represents a good alternative to obtain rainfall detection in areas where environmental constraints do not allow the installation of rain gauges or radar systems, in fact its performances are also better than the ones obtained by a weather radar operating in the experiment area. In some cases with high wind above 10 m/s the probability of false alarm can increase, but there are no important alterations in results during passages of ships in the area near the dataset spectra were collected. Also no alterations in results are related to seasonal basis. Considering the wind prediction case, the best performances are obtained using Dictionary Learning coefficients with polynomial regression. In particular, using 16 coefficients and a polynomial of third degree. The performances increase with respect to state-of-the-art algorithms and the

RMSE is 1.15 m/s. Also MAE value decreases to 0.80 m/s and ME is very near zero. Suboptimal results are obtained also using a bank of filters to extract MFCCs or GTCCs and using them as input of a polynomial model. In the first case 15 coefficients and a second degree polynomial are applied and in the second case 10 coefficients and a third degree polynomial are used. The RMSE is 1.28 m/s for MFCC and 1.27 m/s for GTCC. Analysing also the plot (Fig. 4.9) in which predicted values versus real wind speed are depicted, it can be noted that the samples follow very well the bisector line and the dots inside the two lines representing the bisector shifted by ±RMSE are about the 85% of the total. So they are grouped around the perfect prediction line represented by the bisector. Representing in a chronological way all the 18193 wind real values and the predicted values together as in Fig. 4.12, it can be seen that the prediction trend follows very well the real one in time, both considering the general trend and if, as in Fig. 4.13, detailed situations are considered. Considering the rainfall prediction case, the best results are obtained again using Dictionary Learning coefficients and polynomial regression. 12 Dictionary Learning coefficients and a third degree polynomial were used. Then all predictions lower than zeros are put to 0 mm/h. RMSE has a value of 0.50 mm/h and the MAE 0.14 mm/h. A further increase of the performances can be achieved by introducing a threshold of 0.6 mm/h, instead of the threshold of 0 mm/h. So the RMSE decreases to 0.48 mm/h and MAE to 0.08 mm/h. Again, as for wind prediction, also good results are shown using a bank of filters to extract MFCCs or GTCCs and fitting a polynomial model. The RMSE is 0.58 mm/h for MFCC and 0.59 mm/h for GTCC. In the first case 20 coefficients and a third degree polynomial are applied and in the second case 10 coefficients and again a third degree polynomial are used. Analyzing, as in the wind prediction case, scatter plot (Fig. 5.4) in which every red point represents real values versus predicted ones again it is quite clear that a great number of points are inside the two bisectors lines shifted of ±RMSE and a low number of samples are outside from these lines. So points are well grouped around the perfect prediction line represented by the bisector and the samples follow it very well. Increasing the intensity of rain, the behavior becomes less precise and in general high errors are computed for high values. Considering the rainfall trend over time for real values and predictions from a general point of view (Fig. 5.9) the real trend is well followed. Considering a more detailed image it can be noted that there are small oscillations near 0 mm/h. In order to solve this kind of problem the threshold at 0.6 mm/h was considered and it can be seen that the oscillation problem is quite solved and then also the trend for low values is better followed. The performance obtained working on averaged spectra suggests that Machine Learning models may also be advantageous for detection and prediction using short-term acoustic spectra. So a possible idea for future research could be to extend the studies done in this thesis to instantaneous spectra. Also similar considerations could be done for what concern the usage of temporal spectra and so in this way extending the Machine Learning analysis from frequency to time domain. Another objective could be to understand if the Machine Learning techniques that have been used in this thesis and the models fitted using the available dataset can obtain good results also with data collected in other parts of the world and in seas with different characteristics in terms of salinity,

temperature and maybe with different depth. In the future, the performance of the prediction system should be assessed when it is operated at a different point from that where the data used for training were acquired. The performance drift when the prediction system is trained and operated under environmental conditions different than those used in this thesis (e.g., shallow water, polar water, etc.) should also be investigated. An additional idea for future research could be the so called compound. The technique consists in averaging the spectra gathered in successive instants before proceeding with the wind speed or rainfall intensity prediction. Spectral compounding from a few tens of minutes up to a few hours have been used in the state-of-the-art. A lesser adopted option is prediction compounding, in which wind speed predictions produced in successive instants are averaged. In the state-of-the-art [16, 7] the compounding of instantaneous predictions was applied over an interval of three hours, while the prediction compounding, over very long intervals, was applied downstream of a spectral compounding over an interval of an hour [1, 17].

REFERENCES

[1] Vagle, S., Large, W. G., Farmer, D. M. (1990). An evaluation of the WOTAN technique of inferring oceanic winds from underwater ambient sound. Journal of Atmospheric and Oceanic Technology, 7(4), 576-595.

[2] Pensieri, S., Bozzano, R., Nystuen, J. A., Anagnostou, E. N., Anagnostou, M. N., et al. (2015). Underwater acoustic measurements to predict wind and rainfall in the Mediterranean Sea. Advances in Meteorology, 2015(612512), 1-19.

[3] Cazau, D., Bonnel, J., Baumgartner, M. (2019). Wind speed prediction using acoustic underwater glider in a near-shore marine environment. IEEE Transactions on Geoscience and Remote Sensing, 57(4), 2097-2106.

[4] Ma, B. B., Nystuen, J. A. (2005). Passive acoustic detection and measurement of rainfall at sea. Journal of Atmospheric and Oceanic Technology, 22(8), 1225-1248.

[5] Nystuen, J. A., Amitai, E., Anagnostou, E. N., Anagnostou, M. N. (2008). Spatial averaging of oceanic rainfall variability using underwater sound: Ionian Sea rainfall experiment 2004. The Journal of the Acoustical Society of America, 123(4), 1952-1962.

[6] Taylor, W. O., Anagnostou, M. N., Cerrai, D., Anagnostou, E. N. (2021). Machine Learning Methods to Approximate Rainfall and Wind From Acoustic Underwater Measurements (February 2020). IEEE Transactions on Geoscience and Remote Sensing, 59(4), 2810-2821.

[7] Nystuen, J. A., Anagnostou, M. N., Anagnostou, E. N., Papadopoulos, A. (2015). Monitoring Greek seas using passive underwater acoustics. Journal of Atmospheric and Oceanic Technology, 32(2), 334-349.

[8] Nystuen, J. A. (2011, June). Quantifying physical processes in the marine environment using underwater sound. In Proceedings of 4th Underwater Acoustics& Measurements conference (pp. 20-24).

[9] Shaw, P. T., Watts, D. R., Rossby, H. T. (1978). On the prediction of oceanic wind speed and stress from ambient noise measurements. Deep Sea Research, 25(12), 1225-1233.

[10] Anagnostou, M. N., Nystuen, J. A., Anagnostou, E. N., Nikolopoulos, E. I., Amitai, E. (2008). Evaluation of underwater rainfall measurements during the Ionian Sea rainfall experiment. IEEE Transactions on Geoscience and Remote Sensing, 46(10), 2936-2946.

[11] Trucco, A., Bozzano, R., Fava, E., Pensieri, S., Verri, A., Barla, A. (2021) A supervised learning approach for rainfall detection from underwater noise analysis. IEEE Journal of Oceanic Engineering, 1-13

[12] T. Hastie, R. Tibshirani, J. Friedman (2001) The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics

[13] K.S. Rao and Manjunath K.E., (2017) Speech Recognition Using Articulatory and Excitation Source Features, SpringerBriefs in Speech Technology

[14] Rahana Fathima, Raseena P.E., (2013) Gammatone Cepstral Coefficient for Speaker Identification International Journal of Scientific & Engineering Research, Volume 4, Issue 10

[15] Shalabh, IIT Kanpur, Regression Analysis, Polynomial Regression Models, Chapter 12

[16] Riser, S. C., Nystuen, J., Rogers, A. (2008). Monsoon effects in the Bay of Bengal inferred from profiling float-based measurements of wind speed and rainfall. Limnology and Oceanography, 53(5part2), 2080-2093.

[17] Vakkayil, R., Graber, H. C., Large, W. G. (1996, September). Oceanic winds predictd from underwater ambient noise observations in SWADE. In OCEANS 96 MTS/IEEE Conference Proceedings (Vol. 1, pp. 45-51).

[18] M. N. Anagnostou, J. A. Nystuen, E. N. Anagnostou, A. Papadopoulos, and V. Lykousis, “Passive aquatic listener (PAL): an adoptive underwater acoustic recording system for the marine environment,” *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 626-627, supplement, pp. S94–S98, 2011.

Ringraziamenti

Vorrei dedicare questo piccolo spazio a coloro che hanno contribuito alla realizzazione di questa Tesi di Laurea.
Vorrei innanzitutto ringraziare il Prof. Andrea Trucco, Relatore di questa Tesi, che mi ha seguito, passo dopo passo, in questo percorso con la grande pazienza e passione che lo contraddistingue.
Proseguo ringraziando la Prof.ssa Annalisa Barla, Relatrice della Tesi, e il Prof. Alessandro Verri, Correlatore, per il tempo dedicatomi e per tutti i consigli e suggerimenti che hanno contribuito a migliorare il lavoro svolto.
Infine, ringrazio mia madre che mi sostiene e crede in me sempre, in ogni momento, da quando sono nato e anche prima.

"Si Deus pro Nobis…"