ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Electrical, Electronics and Communications Engineering
(32nd cycle)

Novel Neural Approaches to Data Topology Analysis and Telemedicine

Vincenzo Randazzo

* * * * * *

Supervisors
Prof. Eros Pasero, Supervisor
Prof. Giansalvo Cirrincione, Co-supervisor

Doctoral Examination Committee:
Prof. Alessandro Vinciarelli, Referee, University of Glasgow
Prof. Roberto Tagliaferri, Referee, Università degli studi di Salerno
Prof. Marco Badami, Politecnico di Torino
Prof. Vitoantonio Bevilacqua, Politecnico di Bari
Prof. Anna Esposito, Università della Campania "Luigi Vanvitelli"
Prof. Salvatore Vitabile, Università degli studi di Palermo

Politecnico di Torino
September 22, 2020

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . .
Vincenzo Randazzo
Turin, September 22, 2020

Summary

The conventional approach to artificial intelligence and data mining is tied up to input distribution learning and, more generally, to understand the laws underlying data independently of the input at hand. To this purpose, the most common used tools are the neural networks, which are often employed as black boxes for extracting information from data. Once found the proper architecture, neural networks better map the data manifold than human-designed models, especially if the input distribution is non-linear or is embedded in a high dimensional space. Very often, in a context where the Internet of Things (IoT) has become pervasive and tons and tons of data are produced every instant, it is easy to think the best approach is to gather as much data as possible and, then, use deep learning. The idea is that collecting and aggregating a huge amount of data from different sensors, would yield the needed information. Unfortunately, recently the focus is more on achieving amazing performance on classification/regression tasks rather than understanding the reasons behind them. A quite exhaustive example is deep learning, which seems the solution to most of the open problems once collected enough data, but does not have a theoretical model behind. For example, it automatically extracts features from data while performing training; but, what are the extracted features? Moreover, handling a deep learning neural model, requires a lot of data and computation power; is it really required to use such a great computational power, time and efforts just because the dataset is huge?

This thesis tackles the lack of formalism and the black box approach by providing a scientific framework for analysing data and understanding their topology before classification. In this sense, neural networks are used both to explore data manifold and to determin which architecture is better tailored for a problem.

Before choosing an architecture, it would be better to understand data. The input space is analysed using both linear and non-linear methods to estimate its intrinsic dimensionality; understanding the input space can drive the performance analysis and unveil data patterns, which can then be used to guide training, e.g. in the deep learning, and to select the best feature set.

Both unsupervised and supervised architectures have been employed; the former is used for clustering data into unknown groups, the latter for classifying them into predefined classes. The choice of the proper approach is done w.r.t. different applications, e.g. online learning, data projection or telemedicine. Both stationary and non-stationary input distributions are examined. When needed, new neural networks (onCCA, GCCA, G-EXIN, GH-EXIN) have been designed for exploiting input data topology and preserving it during training.

Supervised learning performance has been analysed by studying the classification results as input features change. Deep learning automatically extracts features and provides good classification outcomes, but it is a black box and its results cannot be interpreted in a theoretical framework. On the other side, shallow neural networks need a human-based feature engineering phase prior to their training, but it is possible to interpret their outcomes w.r.t. the input features. The proposed approach combines these two techniques for exploiting their advantages by means of a correlation analysis between the deep layers and the best performing feature set of the classical approach. In this sense, by understanding which are the features automatically extracted by the deep technique, it would be possible to give an interpretation, i.e. an explanation, of its results.

Public available databases have been used in order to compare performance with state of the art on a common benchmark. At the same time, data have been collected at the *Neuronica* and *Polito*^{BIO}*Med* laboratories of Politecnico di Torino in order to validate the quality both of the proposed approach and of the new designed and built devices. The input data can be grouped in three main categories: non-stationary, stationary and IoT. The former regards input distributions that change over time, e.g. jump, and has been exploited for machine prognostic. On the other side, stationary data experiments have been used to handle medical and hierarchical applications; in this sense, the aim was to explore data internal structure and to discover new patterns. Finally, in a real case scenario, an application to telemedicine has been studied: new wireless wearable devices, the ECG WATCH and the VITAL-ECG, have been developed to acquire and monitor vital signs, such as heart rate. The proposed approach has been used to diagnose possible heart diseases and to design a biometric identification system based on electrocardiogram.

Acknowledgements

Attending a PhD was a complex decision to make and live with. Every single day is a challenge hoping that, at the end, some meaningful results for your research topic will be achieved and time was not wasted.

Pushing yourself always at the limit is not an easy task, but I was lucky enough to find two huge beacons, my supervisors, who guided, counselled and, above all, believed in me. These years would not be so amazing without all the chats, deadlines, (night) Skype calls, lunches and dinners, conferences, articles and projects (to be written in a day/night), courses and bureaucracy (which never ends), (almost never accepted) coffees, jokes and, above all, friendship and passion for the research, we shared together.

I would like to thank all the colleagues and students I worked with at Neuronica Lab in these years; two special awards go to Nancy and Jacopo for putting up with me all days (inside and outside the Politecnico), and to Rahul, Gabriele and Pietro for sharing their research paths with me and for having answered my inquiries any time I asked (with no regard to the time zone). Without all of you, this PhD would not have been the same.

Totò, Manfredi, Marco, Valeria, Antonio, Guglielmo, Serena, Maria, Claudia, Gaspare, Francesco and the rest of my friends, thanks for always being there for me. You believe in me and my skills, even when I doubt of myself.

One last thought full of love to my Mom and Dad, my heroes in this brave world.

To the loving memory of my grandparents, who have always encouraged me in struggling for knowledge and science.

Stealing Kant's words: treat neural networks always at the same time as an explorative tool and never simply as an end.

It may seem the hardest path but remember that with ambition, patience and perseverance even «gutta cavat lapidem».

Contents

List of Tables

List of Figures

Chapter 1

New Pathways to the Neural Field

The conventional approach to artificial intelligence and data mining is tied up to input distribution learning and, more generally, to understand the laws underlying data, independently of the input at hand. To this purpose, the most common used tools are the neural networks; indeed, by combining different layers together with activation and error functions, it is possible to extract information from data. In this sense, frequently, neural networks are employed as black boxes, see Fig. 1.1.

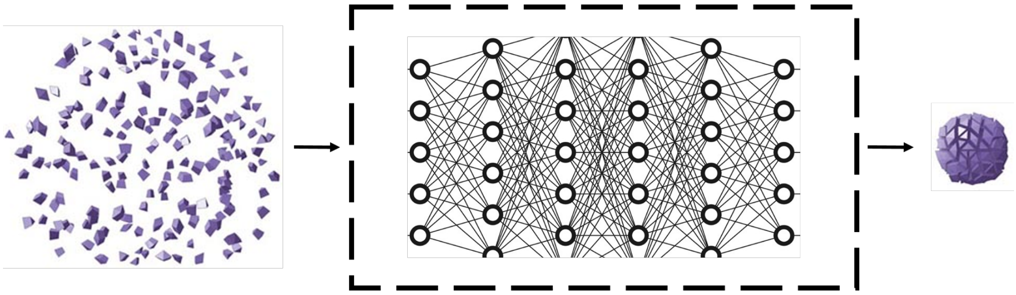


Figure 1.1: Extracting information from data

A quite exhaustive example is deep learning. Each day journals and newspapers publish new results based on this technique. Apparently, it seems to be the solution to most of the open problems; once collected enough data, they can be fed them into a neural system, which, after some training and network optimization, will understand data and yield the proper results. The most impressive ones have dealt with images, e.g. face recognition, natural language processing and healthcare [1]. These are just few examples in which deep learning has been successfully applied; fields like gaming, self-driving cars, fault detection, structural monitoring have also been studied by means of this technique.

The tendency is to take old issues and try to solve them with a neural approach, which is good because it opens new perspectives on the state of the art problems; indeed, if training data are not affected by sufficient noise and given the proper architecture, neural networks better map the data manifold than human-designed models, especially if the input distribution is non-linear or is embedded in a high dimensional space. Unfortunately, as a matter of fact, a lot of researchers, especially those that do not have a proper artificial intelligence background, are focused more on achieving amazing performance on classification/regression tasks rather than understanding the reasons behind them. Actually, deep learning performance may derive from overfitting. To train a deep model, a lot of data are required; generally, the more the better. It is possible that, given so many examples as a training set, they will span over most part of the input space. As a consequence, the system will learn more the data, i.e. overfitting, rather than the process. This hypothesis seems to be confirmed by the fact that, in some cases, changing just few pixels in a test image will lead the neural net to provide a completely wrong output, i.e. a label, like misclassifying a cat for a dog. In this sense, it can be said that current deep learning systems are bad at reasoning and abstraction and this is why they need huge amounts of data to learn simple tasks.

For sake of completeness, it must be underlined that deep learning does not have a theoretical model behind it. Transfer learning is an example (see Fig. 1.2). Neural networks trained for some tasks, e.g. imaging, are used (after retraining only some layers) in a completely different field, e.g. speech recognition, and they work. But why? Why data coming from completely different environments can be addressed by the same network, i.e. the same architecture and the same weights? Maybe, thanks to its depth, the system is able, layer after layer, to abstract information towards its comprehension. As a consequence, it may be argued that such an approach mimics the brain ability to learn, maintain and organize knowledge. Indeed, hearing a dog barking or seeing him doing it, will trigger the same concept in a human mind, i.e. a barking dog, independently of the input, either visual or audio, that fired it. In this sense, transfer learning could make perfect sense; the external layers tackle the problem at hand and need to be retrained accordingly, while the network core handles the learning process, i.e. the knowledge, and can be shared across multiple applications. However, these are just hypotheses because deep learning way of functioning has not been uncovered yet. It works, it can achieve amazing performance in some fields, but science is not able to demonstrate why.

A peculiar characteristic of deep learning is its ability of automatically extracting features from data while performing training. The main advantage of such an approach is obvious, no feature engineering is needed, which requires human interaction and, consequently, can lead to bias the results if it is wrongly performed. Moreover, skipping a phase, i.e. the feature extraction, speeds up the recall; a data can be fed directly to the network without preprocessing, and its associated output can be produced instantaneously. In this sense, slicing windows on the input can be avoided, further boosting the process. But, what are the extracted features and why most of them can be shared with transfer learning?

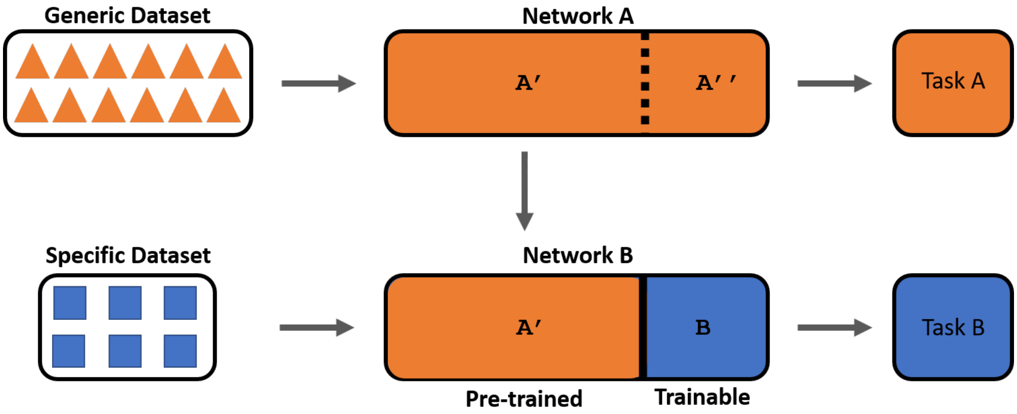


Figure 1.2: Transfer learning: A' represents the knowledge and is task-indipendent; A" and B are task-specific.

On the other hand, to store and efficiently train a deep network with a lot of weight vectors on a big dataset, i.e. huge number of training iterations, for several epochs, a lot of data and computation power are, usually, required. The common strategy employs fast GPUs, with GB of embedded dedicated RAM, and big server farms in order to cut down the time required for each training. Of course, the more and better the hardware, the higher the cost. Is it really required to use such a great computational power, time and efforts just because the dataset is huge? The answer is negative; as an example, consider a dataset made of 2 quadrillion of rows and 1 million of columns, as shown in Fig. 1.3. Here, half of the samples are made by row vectors of all ones and the other half of all zeros. Barely looking at the dimensionality of the dataset, 2P x 1M of integers (2 bytes), i.e. around 4 ZB (2 x 10^{21} x 2 bytes), it is easy to think that the best solution would be using a deep network with several layers and thousands of filters. Indeed, even if the input is very highly dimensional (10^6), its obvious that, given all the columns in a vector have the same value, either 0 or 1, its intrinsic dimensionality is quite small. As a consequence, just a single neuron is able to perfectly separate the input into the two classes, saving time, effort and money. Of course, this is trivial example but, the underlying idea is not. Very often, in a context where the Internet of Things (IoT) has become pervasive [2] and tons and tons of data are produced every instant, it is easy to think the best approach is to gather as much data as possible and, then, use deep learning. The idea is that the collection and aggregation of a huge amount of data from different sensors, would yield the needed information. Also, combining different sources, i.e. data fusion, may provide new insights and new perspectives, which could lead to novel inferences on the problem at hand. In this sense, the distinct sources of information, i.e. the sensor data, need to be uncorrelated, otherwise, it will just increase the input space without providing new information to the system but uselessly complicating its training.

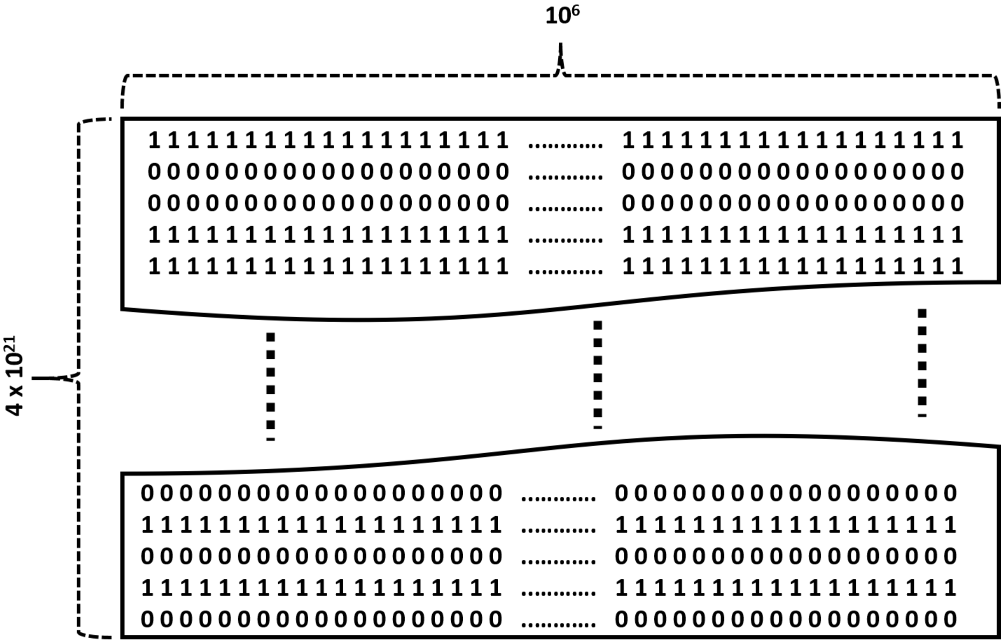


Figure 1.3: Silly database example

In the above discussion, the underlying assumption is that is always possible to find the proper deep architecture within a reasonable amount of time. However, it is not so obvious; actually, it is quite the opposite. When adding a new layer can improve performance? Unfortunately, at the state of the art, no building science exists to create and optimize a neural architecture for one task; as a consequence, the process is not straightforward and may never converge to a proper solution. Trials and errors follow each other by means of adding/removing layers, increasing/decreasing kernel sizes, varying optimization techniques, e.g. dropout or batch normalization, etc. It is like a chef making new recipes who tries different ingredients, quantities and cooking techniques until the result is satisfying. Each attempt, i.e. each network revision, will need a full retraining, which, as explained, is quite time-consuming; considering performing it several times, the overall training process can be incredibly long.

1.1 The proposed approach

Prior to choose an architecture, it would be better to understand data you are working on. In this sense, neural networks can be used to explore data manifold to determine which architecture is better tailored for a problem. Fig. 1.4 shows the process. The input space is analysed using neural methods to estimate its intrinsic dimensionality. First, a linear analysis is performed by means of Principal Component Analysis (PCA) [3] and the intrinsic dimensionality is estimated using Pareto chart [4] as the number δ_{PCA} where the cumulative total variance explained by the components, less-than or equal-to δ_{PCA}, reaches a value greater than 90%. Then, a more complex non-linear study is conducted by means of the Curvilinear Component Analysis (CCA) [5]. To begin, data are projected in a subspace whose dimensionality is equal to δ_{PCA} and the quality of this operation is evaluated by examining its associated *dy-dx* diagram; then, if the projection is not satisfactory, data are projected to different subspaces according to *dy-dx*. The smallest dimension δ_{CCA}, corresponding to a still acceptable diagram, represents the intrinsic dimensionality of the input dataset. Indeed, understanding the input space can drive the performance analysis and unveil data patterns, which can then be used to guide training, e.g. in the deep learning.

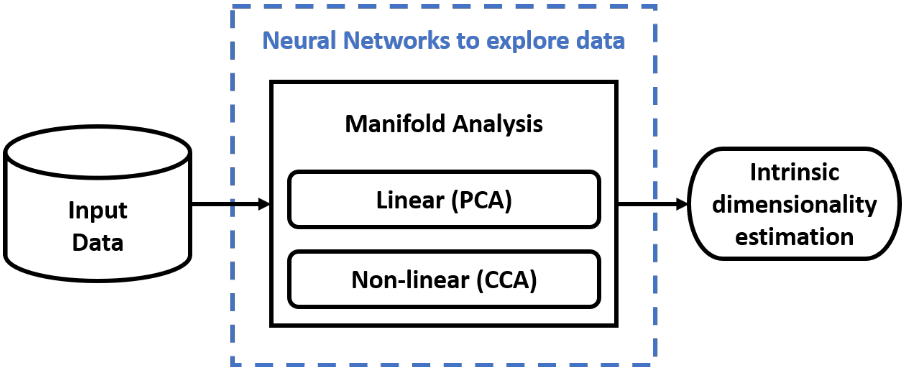

Neural Networks to explore data
Manifold Analysis
Linear (PCA)
Non-linear (CCA)
Input Data
Intrinsic dimensionality estimation

Figure 1.4: Neural networks for intrinsic dimensionality estimation

The intrinsic dimensionality gives just the first glimpse on the input dataset and can be used to drive the feature extraction phase and to determine the size of the neural system input layer, i.e. the quantity of features to be fed. Three approaches have been studied and compared to check their performances and how the intrinsic dimensionality varies after feature extraction. The simpler technique considers the raw data as meaningful in themselves, i.e. not to perform feature extraction; the input is divided into windows, whose size depends on the problem at hand (e.g. a full heartbeat), and the corresponding training set is constructed. An alternative approach considers the statistical time-evolution of the input signal, e.g. its mean and variance, as significative for dealing with the input dataset; in this sense, the focus is on non-stationarity, i.e. the temporal changes, of the input distribution. Finally, a third scenario exploits algorithms as Pisarenko [6] or MUSIC [7] for estimating frequencies of signals from noise-corrupted measurements.

In order to let the system learn the input, the resulting training set is fed to a specific neural network, which can be either unsupervised or supervised; the former is used for clustering data into unknown groups, the latter for classifying them into predefined classes. The choice of the proper approach is done w.r.t. different applications, e.g. online learning, data projection or medical analysis. Both stationary and non-stationary input distributions are examined. When needed, new neural networks (onCCA, GCCA, G-EXIN, GH-EXIN) have been designed for exploiting input data topology and preserving it during training. Fig. 1.5 displays the whole process.

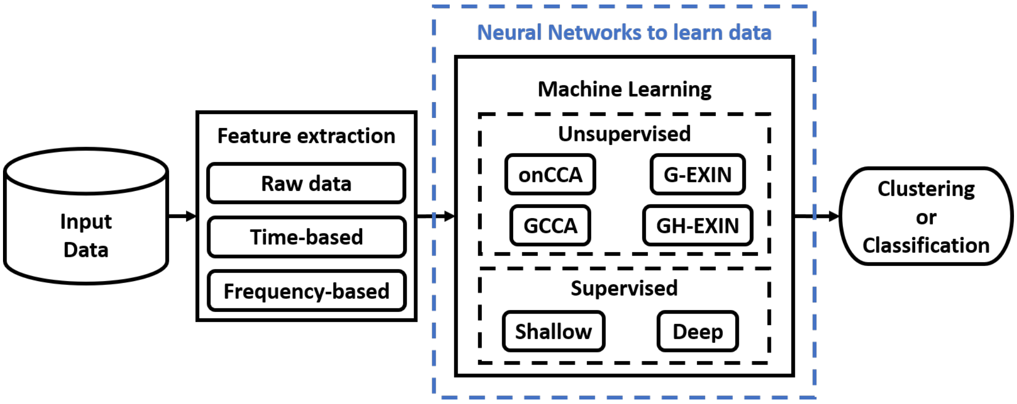


Figure 1.5: Neural networks for learning input data

Supervised learning performance has been analysed by studying the classification results as input features change. Deep learning is able to automatically extract features from data and provide good classification outcomes, but it has to be treated as a black box and the results cannot be interpreted in a theoretical framework. On the other side, classical neural networks, such as shallow ones, need a human-based feature engineering phase prior to their training; due to the network simplicity w.r.t. deep models, it is possible to interpret its outcomes and to relate them with the input features. The proposed approach, see Fig. 1.6, combines these two techniques to exploit their advantages. First, deep learning is trained to reach a good classification performance P; then, P is used as a benchmark to evaluate and guide classical neural network training and feature selection (orange arrow). Once the model reaches a satisfactory performance, the features F, extracted in the engineering phase, are sought in the deep learning model by means of a correlation analysis between F and the deep network layers (blue arrow). In this sense, by understanding which are the features automatically extracted by the deep technique, it would be possible to give an interpretation, i.e. an explanation, of its results.

The proposed method has been tested with several experiments on different datasets depending on the application at hand, as shown in Fig. 1.7. Public available databases have been used for comparing performances with state of the art on a common benchmark. At the same time, data have been collected at the *Neuronica* and $Polito^{BIO}Med$ laboratories of Politecnico di Torino to validate

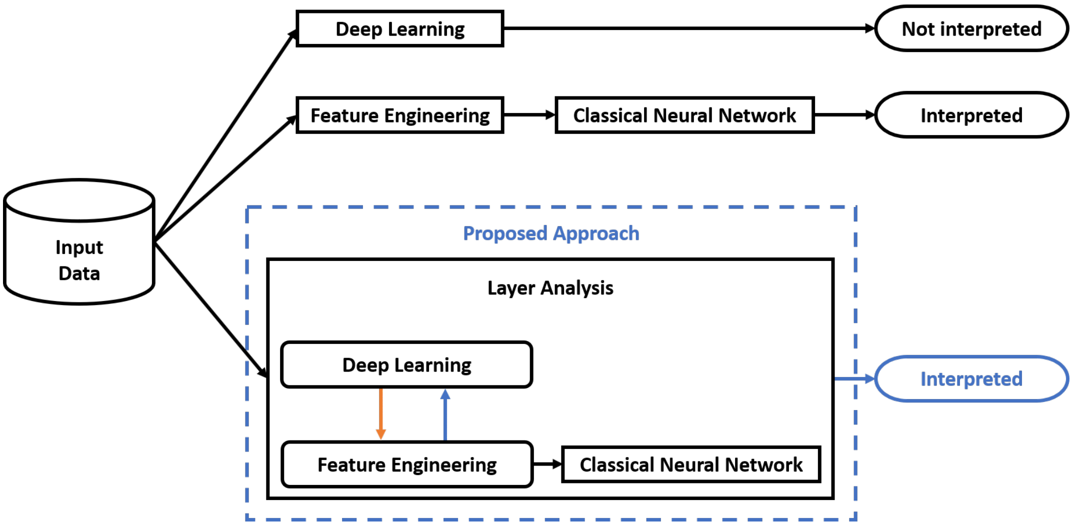


Figure 1.6: Proposed method for understanding deep learning

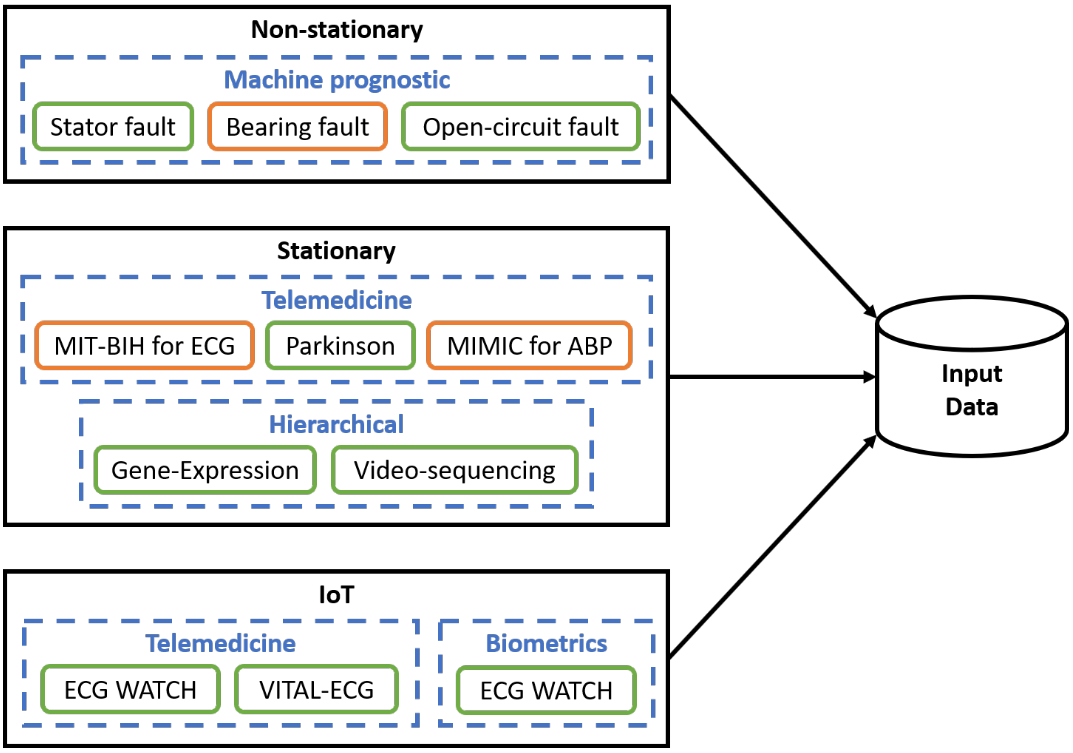


Figure 1.7: Experiment taxonomy: black solid boxes yield the input category; blue dashed boxes provide the application; green solid boxes represent experiments on datasets manually collected in our laboratory; orange solid boxes are experiments on public available databases.

the quality both of the proposed approach and of the new designed and built devices. The input data can be grouped in three main categories: non-stationary, stationary and IoT. The former focuses on input distributions that change over time, e.g. jump, and has been exploited for machine prognostic; in this case, novel unsupervised topological neural networks have been designed in order to track the full machine evolution towards faults and to detect pre-fault conditions. On the other side, stationary data experiments have been used to handle telemedicine and hierarchical applications. In this sense, the aim was to explore data internal structure and to discover new patterns. Finally, a real case scenario is presented where an application of vital parameters recording (telemedicine) has been studied: new wireless wearable devices, the ECG WATCH and the VITAL-ECG, have been developed to acquire and monitor vital signs, such as heart rate. The proposed approach has been exploited to diagnose possible heart diseases and to design a biometric identification system based on electrocardiogram.

Resuming, the purpose of this thesis is to tackle the lack of formalism and the black box approach by means of providing a scientific framework to analyse and understand data and their topology before performing classification. In this sense, neural networks are used both to explore data manifold and to determine which architecture is better tailored for a problem. The proposed approach has been validated also on a real-case application like vital parameter monitoring.

Chapter 2

Manifold Analysis for Intrinsic Dimensionality Estimation

The starting point for dealing with an input set is to estimate the intrinsic dimensionality of the manifold on which data lay. In this sense, topology is considered as the key to understand data regardless of the representation. Indeed, one of the difficulties about multivariate analysis is the visualization of data with many variables together with their relationships. Luckily, in such datasets, groups of variables are often correlated. For example, it is possible that more than one feature measures the same driving principle of the system at hand. Plenty of instruments allows to record, at the same time, dozens of system quantities; typically, only either a few are really significative or some are correlated; it means dataset intrinsic dimensionality is lower than the input number of variables. In such a scenario, it is possible to exploit the redundancy of information. A group of features can be replaced by a single new variable, i.e. the dataset can be projected, in a linear or non-linear way, to a smaller subspace. The notion of *intrinsic dimensionality* refers to the fact that any low-dimensional data space can trivially be turned into a higher-dimensional space by adding redundant or randomized dimensions, and in turn many high-dimensional data sets can be reduced to lower-dimensional data without significant information loss. Hence, this data preprocessing does not lose information but, on the contrary, let emerge it from them.

2.1 Linear analysis

At first, the manifold is assumed as linear; therefore, a classical analysis is performed by means of Principal Component Analysis (PCA) [3, 8, 9] and the associated Pareto chart. Indeed, PCA is the simplest of eigenvector-based multivariate techniques. In literature, it is frequently used for revealing data internal structure. It generates a new set of features, called principal components (PC), which are a linear combination of the original variables. PCs are orthogonal to each other, so there is no redundant information, and form an orthogonal basis for the input space. As a consequence, there are as many PCs as the original set of features. PCA can be interpreted as fitting an n-dimensional ellipsoid to the data, where each of its axes represents a principal component. The longer axes lay on the directions of maximum variance, where most of the information is contained. On the contrary, if some axis of the ellipsoid is small, then the corresponding variance is also small, and by omitting its associated PC, only an equally small amount of information is lost. Dealing with plots on a reduced number of features, may lead to develop a deeper understanding of the laws underlying the original data.

2.1.1 Pareto chart

As a possible representation, component variances are displayed as columns of a Pareto chart, where individual PC variances are plotted in descending order by bars, and the cumulative total is represented by a line. The goal is to highlight the most important PCs, i.e. those that contribute the most to the overall information; in this sense, it can be used to estimate the manifold intrinsic dimensionality, δ_{PCA}, as the x-value where the cumulative variance is greater than a predefined threshold T. Fig. 2.1 shows an example where δ_{PCA} is equal to 7 for $T = 90\%$.

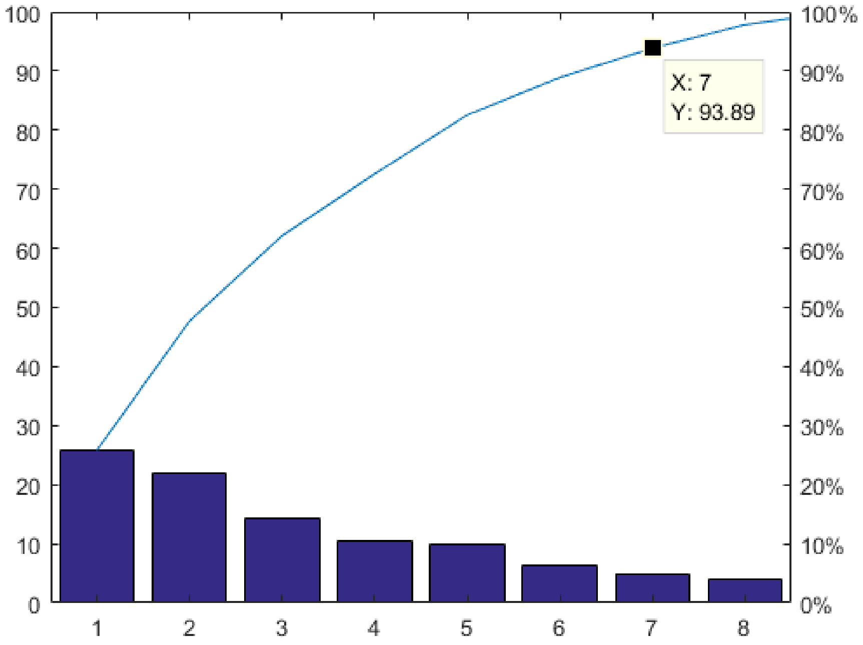


Figure 2.1: PCA Pareto chart example: intrinsic dimensionality δ_{PCA} is equal to 7

2.1.2 PCA geometrical analysis

In case of supervised learning, where each data belongs to a single class known a priori, it is possible to use PCA for performing both intracluster and intercluster analyses. At this purpose, two techniques can be used: principal angles [10] and biplots [11, 12]. According to [13], given a Euclidean space of arbitrary dimension, for any pair of flats (points, lines, planes etc.), it can be defined a set of mutual angles, called principal [10], which are invariant under isometric transformation of the Euclidean space. Geometrically, subspaces are flats that include the origin, thus any two subspaces intersect at least in the origin. Two two-dimensional subspaces $\mathcal{U}$ and $\mathcal{W}$ generate a set of two angles. In a three-dimensional Euclidean space, the subspaces $\mathcal{U}$ and $\mathcal{W}$ are either identical, or their intersection forms a line. In the former case, both $\theta_1 = \theta_2 = 0$. In the latter case, only $\theta_1 = 0$, where vectors u_1 and w_1 are on the line of the intersection $\mathcal{U} \cap \mathcal{W}$ and have the same direction. The angle $\theta_2 > 0$ will be the angle between the subspaces $\mathcal{U}$ and $\mathcal{W}$ in the orthogonal complement to $\mathcal{U} \cap \mathcal{W}$. Imagining the angle between two planes in 3D, one intuitively thinks of the largest angle, $\theta_2 > 0$.

Biplots are a generalization of a scatterplot, which display graphically, at the same time, both observations and variables of a dataset; the former are plotted as points while the latter either as vectors, linear axes or non-linear trajectories depending on the application at hand. In [14] biplots are presented as a useful and reliable method for the visualization of multivariate data when using PCA. In this sense, biplots can be considered as an exploratory tool that allows to visualize each variable contribution up to three PCs at the same time, and how each observation is represented in terms of those components.

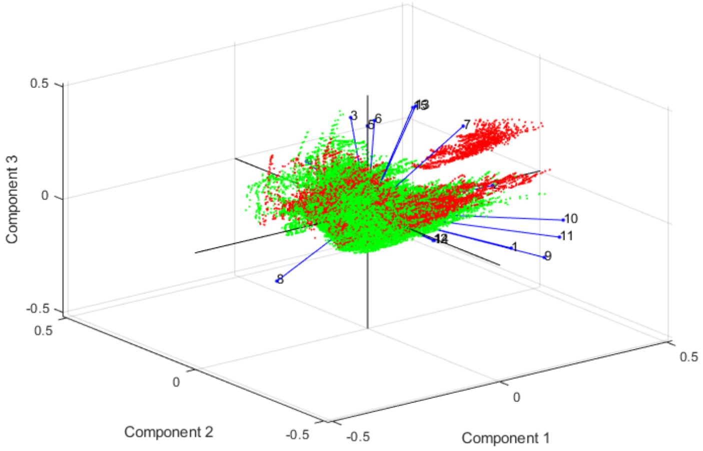


Figure 2.2: 3D PCA biplot example for cluster visualization: red and green points indicate data cluster, blue lines represent the original set of variables.

The geometrical analysis of data manifold can be used to perform both intracluster and intercluster analyses. In the former case, a single cluster can be visualized and inspected to check for outliers or data subgrouping. In the latter case, principal angles provide the relative orientation of clusters in order to detect the degree of data correlation, w.r.t. the selected subset of principal components. This can be further investigated using biplots as shown in Fig. 2.2; here, two clusters (red and green points) projected on the first three PCs are shown together with the original variables (blue lines). Of course, the same considerations hold for intraclass and interclass analyses.

2.2 Non-linear analysis

In the previous section, a simple linear projection by means of PCA has been presented; as a consequence, the inferences may result to be too approximated and the intrinsic dimensionality quite inaccurate. On the contrary, such an approach is not naive for two reasons: first, it is important to have a starting point; secondly, input data may also be linear as those of the silly database presented in the previous chapter (see Fig. 1.3). Therefore, δ_{PCA} is used mainly to have an idea of data intrinsic dimensionality; then, a more complex non-linear technique, i.e. Curvilinear Component Analysis (CCA) [5] [15], is used to refine this study.

CCA is a self-organizing neural network for data projection. It maintains the input topology by means of local distance preservation. In this sense, it can be used to reduce the amount of input variables without altering the original manifold. It is based on the Sammon mapping [16] and, in addition, is able of data extrapolation and unfolding.

Each neuron has associated two weight vectors, one in input space (say X) and another in the projected space (latent space, say Y). First, it quantizes the input space for finding the X-weight; then, it estimates the corresponding non-linear projection into the latent space. For each pair of different X-weights (i.e. data space), an in-between point distance D_{ij}, is computed as:

$$D_{ij} = \|x_i - x_j\| \tag{2.1}$$

The corresponding distance L_{ij} in the latent space, is calculated as:

$$L_{ij} = \|y_i - y_j\| \tag{2.2}$$

The goal is to have $L_{ij} = D_{ij}$, which, of course, is achievable only if the input manifold is linear w.r.t. the chosen dimension of projection, δ. In order to tackle the case of non-linear manifolds, CCA defines a metric function F_λ, which penalizes the long distances, but preserves local topology, by using a user-dependent parameter λ. The simplest implementation uses the following step function, which constrains only the L_{ij} smaller than the threshold λ:

$$F_\lambda(L_{ij}) = \begin{cases} 0 & \lambda < L_{ij} \\ 1 & \lambda \geq L_{ij} \end{cases} \tag{2.3}$$

The Y-weights are updated according to:

$$y_i(t+1) = y_i(t) + \alpha(t)F_\lambda(L_{ij})(D_{ij} - L_{ij})\frac{y_j(t) - y_i(t)}{L_{ij}} \tag{2.4}$$

where α is the hyperparameter for the learning rate.

2.3 The *dy-dx* diagram

A fundamental tool related to CCA is the *dy-dx* diagram, where the in-between neuron distances in the latent space (dy) are plotted against their corresponding ones in the input space (dx), as shown in Fig. 2.3.

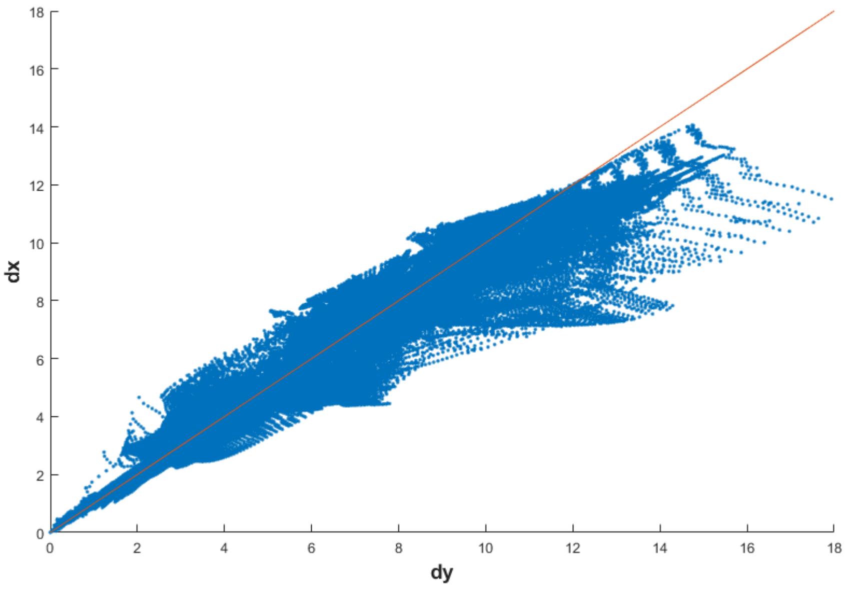


Figure 2.3: CCA *dy-dx* diagram example: blue points are the in-between neuron distances, red line indicates the bisector.

The proposed approach exploits the diagram as a tool for the detection and analysis of non-linearities. In this sense, the *dy-dx* diagram is used for measuring and analysing the quality of the CCA projection; here, the benchmark is the bisector line, which, of course, corresponds to the $L_{ij} = D_{ij}$ condition. Generally, the larger the divergence of the point cloud (blue points) from the bisector (red line), the more non-linear the manifold is. The thickness of distance pairs also depends on the level of noise in data. Fig. 2.3 illustrates an example; here, points clearly spread out of the bisector, which means the projection is not satisfactory because there is a significative difference between the original distances and their corresponding ones in the latent space. As a consequence, reducing the dimension implies a loss of information. On the contrary, a "good mapping" is when the data points are aligned along the bisector, i.e. the distances in the X-space are properly preserved in the Y-space, see Fig. 2.4. In this case, from the diagram it is possible to infer that input data can be projected to a lower dimensionality (δ) space (data belonging to a linear δ-dimensional manifold), i.e. the number of input features can be reduced, without losing much information about the data. It can also be deduced data are only slightly noised.

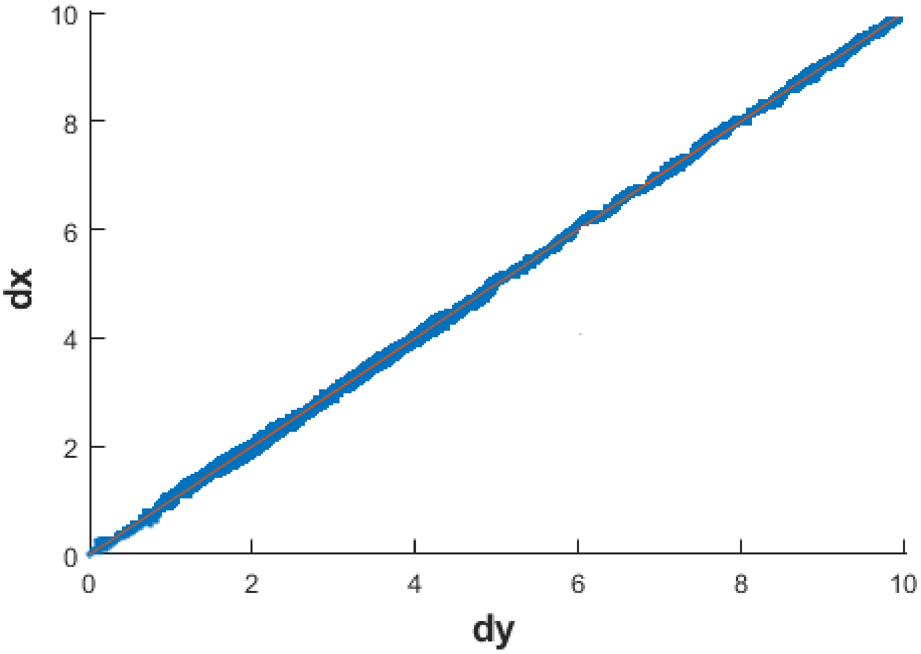
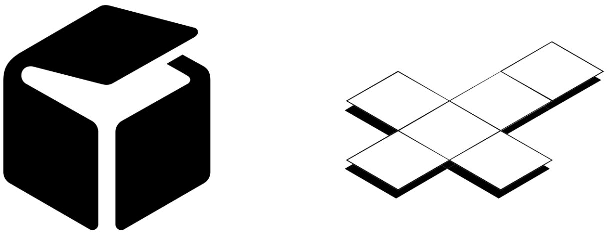


Figure 2.4: CCA *dy*-*dx* diagram example of noisy linear data: blue points are the in-between neuron distances, red line indicates the bisector.

More in general, a deeper analysis of the *dy*-*dx* diagram can provide further information about the projection at hand. The projection rule (2.3) only constrains the distances in the latent space lower than λ to be equal to the corresponding ones in the original space. This is a simple way to force a local linearity in the projection. As a consequence, it is important that, at least in the neighbourhood of the origin, points are concentrated in the bisector. The extent of this neighbourhood is inversely proportional to the level of data non-linearity.

In case of clusters around the bisector, the diagram suggests the presence of data clusters in the original space: the intracluster distances, i.e. the small ones, are represented by the neighbourhood of the origin, while the intercluster ones, i.e. longer distances, lay on the diagram clusters. For example, if distance pairs are along the bisector around the origin and on two other separate groups, it can be deduced the presence in the original space of at least three data manifolds. The first group of distance pairs, around the origin, is composed of the intercluster distances and its thickness represents the level of non-linearity. The other groups along the bisector correspond to the intracluster distances, which are related to the reciprocal position of the data manifolds in the original space. In this case, there are at least three manifolds with very different intercluster distances.



(a) Input space [17]   (b) Output space [18]

Figure 2.5: CCA unfolding property

Table 2.1: How to interpret a *dy-dx* diagram

***dy-dx* feature**	**Input data manifold**
Data along the Bisector	Linear
Extent around the bisector	Degree of non-linearity and/or noise
Neighborhood of the origin	Local linearity
Extent of the neighborhood of the origin	Level of data linearity
Bending below the bisector	Folded manifold
Bending above the bisector	Insufficient projection
One cluster far from the origin	Two manifolds
More Clusters around the bisector	More than two manifolds
Distance between clusters	Distance between manifold
Pairs very far from the bisector	High non-linearity or outliers

When data points bend under the bisector, Y-distances are greater than their corresponding ones in the X-space; in this sense, CCA performs a manifold unfolding as shown in Fig. 2.5. It can be deduced that the input manifold is folded.

On the contrary, points bending above the bisector signal that the projection is insufficient and the output manifold has been folded w.r.t. the input distribution; probably, the chosen dimension of projection δ is too small or either λ or α need further tuning. Table 2.1 summarizes how to interpret a *dy-dx* diagram.

Finally, by analysing different *dy-dx* diagrams varying δ and λ, it is possible to estimate the intrinsic dimensionality δ_{CCA} as the value where the corresponding CCA projection is satisfactory w.r.t. the *dy-dx* diagram, i.e. there is a clear interpretation on the input manifold. However, it must be taken into account that when δ approaches the number of original features, the diagram tends to the bisector (no projection).

Chapter 3

Data Projection

In the previous chapter, several techniques for understanding data intrinsic dimensionality (i.e. the smallest number of features needed to represent the input manifold) have been introduced and discussed. In this sense, it is possible to project data from the input space into another, whose dimensionality is, in general, much lower than the original one. Dimensionality reduction (DR) also attenuates the so called *curse of dimensionality.* DR is a fundamental tool because it simplifies the handling of high-dimensional datasets, i.e. big data, as those produced from internet and/or IoT. Indeed, as explained in [19], the impressive increase in the magnitude of available data is not only visible in the huge amount of samples, but also in the quantity of variables (features), that can be simultaneously recorded on a task. As a consequence, modern techniques have to work on high-dimensional data, whose variables are not independent one another. Moreover, high-dimensional spaces have few drawbacks [20]: typically, the higher the number of variables, the higher the acquisition noise and, consequently, the error; in addition, there is not a sufficient number of samples for getting good estimates. One of the key for training successfully a learning system is having enough samples so that they fill the space or the part of it where the model must be valid. Unfortunately, the first consequence of the curse of dimensionality is that the order of magnitude of samples needed for a significative training is related by means of an exponential law with the amount of features of the dataset: if 10 samples are enough to learn a smooth 1-D manifold, 100 are needed for a 2-D model with the same smoothness, 1000 for a 3-D model, an so on, (see Fig. 3.1). Moreover, exploring a space becomes much more difficult as its dimensionality increases. A simple but exhaustive example is presented in [21]: *«let's say someone has a straight line 100 yards long and he dropped a penny somewhere on it. It would not be too hard to find. He walks along the line and it takes two minutes. Now let us say he has a square 100 yards on each side and he dropped a penny somewhere on it. It would be pretty hard, like searching across two football fields stuck together. It could take days. Now a cube 100 yards across. That's like searching a 30-story building the size of a football stadium».*

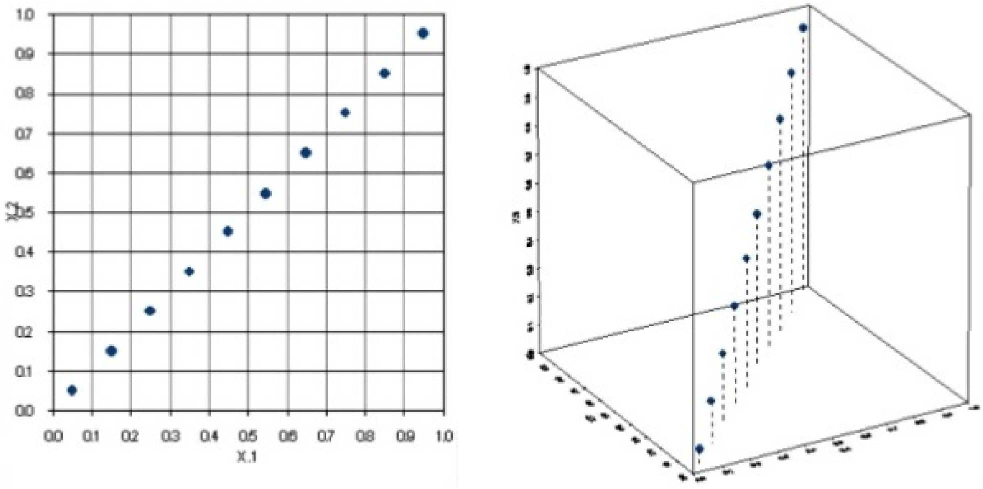


Figure 3.1: Representation of 10% sample probability space in 2-D (left) and 3-D (right)

In addition, high-dimensional spaces have surprising geometrical properties that are counter-intuitive. The volume of a unit-radius sphere grows when the dimensionality increases from one (i.e. a segment) to five (i.e. a 5-D hypersphere); on the contrary, it falls to approximately zero when the dimensionality becomes greater than twenty. In a 2-D space, the most part of the volume of a cube is contained in it, but it spreads towards the corners as the space dimensionality increases. In terms of data density, this implies that if samples are drawn randomly and uniformly in a cube, the probability that they fall near the corners is almost one [19]. Finally, consider a multi-variate Gaussian function whose integral is equal to one. The percentage of the volume inside a radius is equal to 90% in 1-D; this percentage quickly drops to almost 0 in dimension as low as ten: almost all the volume is in the function tails and not near its center, as intuitively expected [19]. More than geometrical properties, the above examples demonstrate that, when dimensionality increases, data migrate to unexpected portions of the space and that functions thought as local become not local anymore. As a consequence, such properties must be considered when designing a data analysis technique.

Traditionally, DR was tackled using linear methods, such as PCA. Recently, several non-linear techniques have been proposed in literature to handle tasks more accurately. A first example [22] uses PCA locally in restricted portion of the space; combining local linear projections yields a global non-linear model, which, unfortunately, is not continuous. Another technique is the kernel PCA (kPCA) [23], which first projects the input into a space higher enough to make the manifold linear, and, then, applies PCA on the transformed data. The advantage is the strong theoretical background of kernel methods; on the contrary, the algorithm is prone to selecting a proper initial space dimensionality and to all the drawbacks of augmenting the input space dimensionality. For a complete overview see [24].

Most dimensionality reduction methods work offline, i.e. they require a static database (batch) of data, whose dimensionality is reduced. This is the case of, for example, both PCA and CCA. However, having a real time DR tool is fundamental: it allows to project data after only the presentation of few samples (i.e. a very fast projection response), and also to track non-stationary input distributions (e.g. time-varying manifolds). This can be exploited, for instance, for real time pattern recognition applications [25]: novelty and outlier detection, fault diagnosis, computer vision and scene analysis, intrusion detection for alarm systems, and so on.

3.1 State of the art

Real time algorithms need to be fed with a data stream, i.e. a continuous input, which, in general, belongs to a stationary distribution. The fastest algorithms are linear projection methods, like the Generalized Hebbian Algorithm (GHA, [26]), the incremental PCA (candid covariance-free CCIPCA [27]) and the Adaptive Principal-component Extractor (APEX, [28]). Current non-linear DR algorithms cannot be used for online applications. Many efforts have been done to speed-up these techniques by means of designing incremental variants [29, 30, 31]. Unfortunately, they are computationally heavy; as a consequence, these methods are useless in a real time scenario.

Neural networks can also be employed for DR. They are usually trained offline and, then, used in real time (i.e. recall phase). In this context, they are effective only for stationary data and should be thought as implicit models of the embedding. The adaptivity of the multilayer perceptron (MLP) and radial basis functions is well suited for this goal by means of designing specific architectures and error functions [31]. An example is SAMANN [32], where an MLP is trained on a precalculated Sammon's mapping. MLP-based algorithms need the training set to be stationary. The same consideration holds for deep neural autoencoders [33] trained for modelling data projection. The main drawback is the slowness of training convergence, especially when the input and target are very high-dimensional. Furthermore, they can be biased by local minima in the objective function.

An important category of neural networks comprises the self-organizing feature maps (SOM [34]) and its incremental variants [35]. SOM is a feature mapper with fixed topology, which constitutes also its main drawback. The variants try to overcome this limit by implementing either no topology (neural gas, NG [36]) or a variable topology and an incremental approach like growing neural gas (GNG, [37]). This latter group exploits the Competitive Hebbian Rule (CHR, [37]) to map the manifold. The combination of NG and CHR is called Topology representing network (TRN, [38]). When the DR used method is a multidimensional scaling (MDS), the technique is called TRNMap [39]; RBF-NDR [40] models the DR with an RBF and an error function based on Euclidean and geodesic distances. The last two techniques perform data projection after estimating the graph, which prevents tracking changes in real time. When the graph is built using GNG, then the projection can be performed by OVI-NG [41], if Euclidean distances are used, and GNLG-NG [42] in case of geodesic distances. However, in a real time scenario only OVI-NG can be employed, because it performs, at the same time, the graph updating and its projection.

3.1.1 Non-stationary techniques

In case of a non-stationary input data stream, e.g. fault and pre-fault diagnosis systems, the above-cited methods cannot be used. For example, techniques based on geodesic distances always require a connected graph; if the input jumps, they cannot follow it. On the contrary, DSOM [43], a variant of SOM, can be employed. Instead of time-decreasing parameters, DSOM uses constant ones (learning rate and elasticity); therefore, the model can quickly adapt to non-stationary inputs. Because it is a forgetting network, only the last changes can be tracked; if the past samples carry important information, this becomes a dangerous limit. Approaches like SOINN and its variants [44] model the whole life of the input data stream (i.e. life-long learning), but do not perform data projection. Nevertheless, they can be used as a preprocessing step before the DR. As a consequence, DR tools can be selected according to the application at hand. When only the last data are meaningful, a forgetting network, such as DSOM, can be used; if the whole evolution is of interest, because, for example, samples can occur again in the future, then a SOINN-like method is better tailored.

The same considerations hold for stream clustering techniques [45], which can be grouped depending on their underlying clustering approach: incremental versions of GNG (e.g., G-Stream [46]), hierarchical stream methods [47, 48], partitioning stream tecniques, like CluStream [49], and density-based stream methods [50, 51]. None of the cited approaches consider the dimensionality reduction step, which is essential when dealing with high dimensional data streams.

3.2 The online Curvilinear Component Analysis

Recently, the online Curvilinear Component Analysis (onCCA [52]) has been proposed to tackle the problem described in the previous section. It is an online incremental neural network that performs data projection according to the CCA projection rule (2.4). As a consequence, it can be used for real-time applications such as prognostic. As in CCA, each neuron is equipped with two weight vectors, one in the input space X, which determines its quantization, and another in the output space Y, which is the corresponding projection. Neurons are connected by links according to the Competitive Hebbian Learning (CHL [53]); they define the network topology, which is designed to mimic the input distribution. Each link has associated an *age*; when its value exceeds the global parameter age_{max}, it is pruned. Similarly, neurons which remain without links are removed from the network.

3.2.1 The algorithm

At the presentation of a new data $x_0 \in X$, neurons are sorted w.r.t their distances d_i^X in the X-space: the closest unit is called first winner, w_1, the second one is the second winner, w_2, and so on. Then, the *novelty test* is performed: if the distance d_1^X from the first winner is greater than a global scalar threshold ρ, the novelty test is passed and a new neuron, w_{x_0}, is created; otherwise, the test is failed and the network will update its weights for adapting to the new data. In the former case, the current network configuration is considered unable to explain x_0 properly, and a new neuron is created on top of it (i.e. its X-weight is set equal to x_0). The new neuron and w_1 are linked (link age is set to zero), while the age of the other links emanating from w_1 is incremented by one.

Conversely, when $d_1^X \leq \rho$, x_0 is assigned to w_1; the soft competitive learning (SCL) [37] is used to update the X-weights of both the first winner, w_1^X, and the neurons connected by a link, i.e. its neighbours (N_{w_1}), according to (3.1). The first and second winners are connected with a link whose age is set to zero, while the age of the others links emanating from w_1 is incremented by one.

$$\Delta w_1^X = \alpha_1(x_0 - w_1^X) \tag{3.1a}$$

$$\Delta w_i^X = \alpha_n(x_0 - w_i^X) \tag{3.1b}$$

where $w_i \in N_{w_1}$.

Finally, the onCCA training starts from a CCA trained on a small dataset; indeed, for a small bunch of data, it does not make sense to make a projection with onCCA because the input topology is still undefined. The size of this input set, i.e. the number of samples used to train the CCA, is equal to the hyperparameter *online threshold*, whose magnitude is related to the complexity of the application at hand. The final structure of CCA, i.e. units equipped with an X-weight and a Y-weight, is the initial structure for the onCCA. Fig. 3.2 shows the complete algorithm.

Data projection

Each time a change occurs in the X-side of the network, the Y counterpart, i.e. the projection, needs to be updated accordingly. In order to have a fast performing algorithm, it is impossible to recompute the projection of all data, i.e. the whole Y-weight set, at each iteration of the algorithm. As a consequence, two resolution parameters, δ in the input space and λ in the output space, are used to limit the number of Y-weight updating.

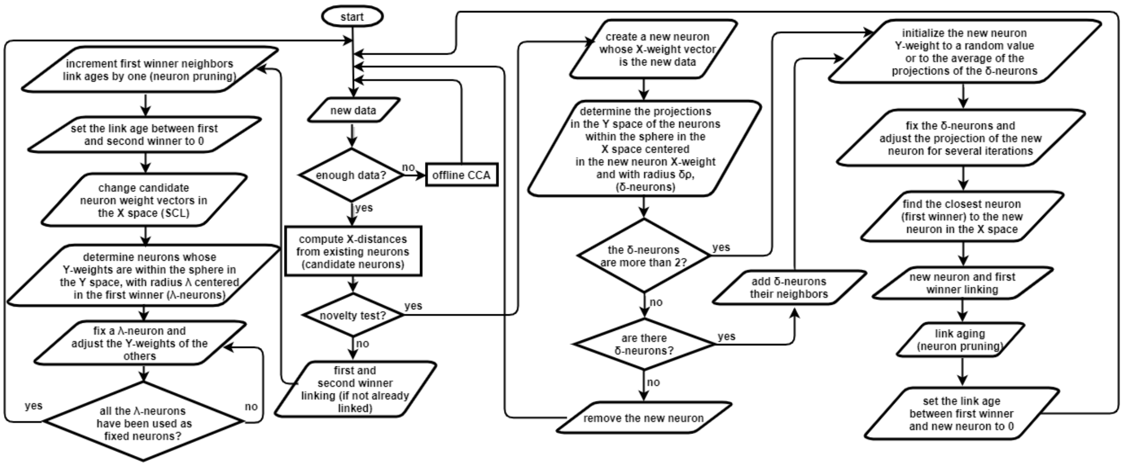
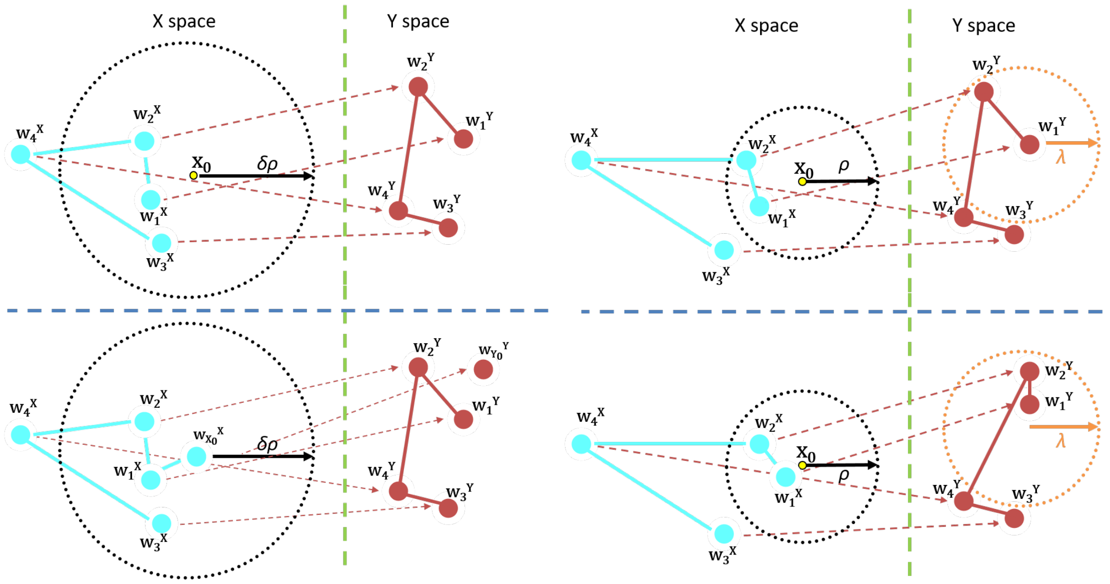

start
increment first winner neighbors link ages by one (neuron pruning)
set the link age between first and second winner to 0
change candidate neuron weight vectors in the X space (SCL)
determine neurons whose Y-weights are within the sphere in the Y space, with radius Λ centered in the first winner (Λ-neurons)
fix a Λ-neuron and adjust the Y-weights of the others
yes
all the Λ-neurons have been used as fixed neurons?
no
new data
enough data?
no
offline CCA
yes
compute X-distances from existing neurons (candidate neurons)
novelty test?
yes
no
first and second winner linking (if not already linked)
create a new neuron whose X-weight vector is the new data
determine the projections in the Y space of the neurons within the sphere in the X space centered in the new neuron X-weight and with radius δρ, (δ-neurons)
the δ-neurons are more than 2?
yes
no
add δ-neurons their neighbors
are there δ-neurons?
yes
no
remove the new neuron
initialize the new neuron Y-weight to a random value or to the average of the projections of the δ-neurons
fix the δ-neurons and adjust the projection of the new neuron for several iterations
find the closest neuron (first winner) to the new neuron in the X space
new neuron and first winner linking
link aging (neuron pruning)
set the link age between first winner and new neuron to 0

Figure 3.2: onCCA algorithm flowchart

Two scenarios may occur (see Fig. 3.3): either a new neuron, w_{x_0}, has been created (see Fig. 3.3a), or an SCL adaptation step has been performed on w_1 neighbours (see Fig. 3.3b). The former implies that $w^X_{x_0}$ is set, while the corresponding projection, $w^Y_{x_0}$ is still undefined. First, the δ-*neurons* are determined as the units within the sphere centred in $w^X_{x_0}$ and with radius equal to $\rho \cdot \delta$. The initial projection, $w^Y_{x_0}$, is set to a random value or to the average of the δ-*neuron* weights in the latent space; then, $w^Y_{x_0}$ is updated for several projection steps according to (2.4). Only the δ-*neurons* are used to estimate the new projection; indeed, they are considered as references, i.e. their Y-weights do not change in this phase. Then, the training procedure restarts with a new sample from the input distribution. If δ-*neuron* set is empty, w_{x_0} and its link are simply removed and the training procedure restarts with a new sample. If there is only one δ-*neuron*, its neighbours are added to the set and the projection is performed as described previously.

The second scenario occurs when x_0 is assigned to the first winner Voronoi set. In this case, due to the SCL update, the X-weights have changed; as a consequence, also the projections have to be updated to take into account of the new data. First, the λ-*neurons* are determined as the units within the sphere centred in w^Y_1 and with radius equal to λ; then, a λ-*neuron* is fixed and the Y-weights of the other λ-*neurons* are updated for several projection steps according to (2.4). The procedure is repeated until all the λ-*neurons* have been used as fixed neuron. Finally, the training procedure restarts with a new sample from the input distribution.



(a) Neuron creation

(b) SCL weight update

Figure 3.3: Novelty test and data projection: novelty test is passed and a new neuron is created in both spaces (left); novelty test is failed and the weights are updated in both spaces (right). Blue dotted lines represent the network evolution before (top) and after (bottom) x_0 presentation; green dotted lines separate the input and output spaces; blue and red points are the X-weights and the Y-weights, respectively; blue and red segments show the links between neurons.

Analysis of the hyperparameters

The onCCA training needs several hyperparameters to be tuned, which can be grouped into three subsets w.r.t. their scope: initial CCA, X-quantization and data projection. The former comprises all the CCA parameters ($\lambda_{offline}$, $\alpha_{offline}$, *epochs*) and the *online threshold*, which defines the size of CCA training set and is related to the complexity of the application. However, onCCA does not hardly rely on the quality of the initial projection; it is just a tool to generate an initial neural architecture.

The second group of hyperparameters deals with the quantization of the input space X. The novelty test threshold ρ is one of the crucial parameters, because it influences directly the input space quantization: a high value yields to a coarse quantization of the X space, but a faster network, because less neurons are created and maintained; therefore, it is absolutely goal dependent and must be tuned w.r.t. the specific application. The SCL hyperparameters, α_1 and α_n, are the same as GNG and represent the constant learning rates for the first winner and its neighbours, respectively. As a rule of thumb, α_n should be at least one order of magnitude smaller than α_1. The last parameter of this group is age_{max}, which is the global threshold above which links are deleted; of course, a smaller value means an higher pruning rate for both links and neurons.

The last hyperparameter subset regards the X-weight projection into the latent space. As explained in Sec. 3.2.1, δ and λ define the global resolution in the input and output spaces, respectively. The former is greater than one and is strictly related to the magnitude of ρ; therefore, its tuning can be less precise. On the contrary, the latter hyperparameter is used for topologically constraining distances. It influences heavily the quality of the projection; as a consequence, it should be tuned properly. Finally, the last hyperparameter is the number of onCCA projection steps. It is related to the previous parameter and affects directly the algorithm computational performance: a low value yields faster results but projection is less accurate.

Each time an Y-weight is updated, i.e. (2.4) is performed, an α parameter is implicitly employed. Experiments suggest a good trade-off value is 0.5, constant over time, so that the network will be always able to adapt to the non-stationarity.

The above considerations demonstrate the most critical hyperparameters are ρ and λ. Indeed, the former controls the neuron creation mechanism, i.e. the input space quantization, while the latter affects the projection quality. Obviously, also the output space dimensionality has to be defined in advance, but its value can be the derived using the techniques described in the previous chapter.

3.2.2 onCCA experiments

The onCCA neural network has been tested on both simulated and real datasets. The aim was to check if the induced quantization follows the input manifold and if the online projection technique keeps the CCA unfolding property.

In the next simulations 3-D datasets have been projected into a 2-D space such that both input and output can be visualized. In the first experiment, a spiral distribution made of around two hundred points (see Fig. 3.4 top left) has been fed to onCCA. The hyperparameters are: $\rho = 0.1$, $\alpha_1 = 0.5$, $\alpha_n = 0.05$, $age_{max} = 10$, $\delta = 5$, $\lambda = 0.2$, $\textit{projection steps} = 10$. Fig. 3.4 bottom shows the onCCA output, i.e. the Y-weights (red points), after 60 (left) and 190 (right) onCCA iterations (data presentation), respectively. Fig. 3.4 top left yields also the complete X-weight set after onCCA training. As shown, the onCCA quantization covers uniformly the input distribution. Finally, Fig. 3.4 top right, shows the corresponding CCA output ($epochs = 10$, $\lambda_{offline} = 0.2$) on the whole dataset as a benchmark for comparison. The onCCA projection proves to be already good after few samples; furthermore, it improves as more data are provided. The method correctly unfolds data as requested and its accuracy is comparable with the traditional offline CCA.

To better test the onCCA, a second simulation has been done by means of a more complex input manifold: 1400 data from two interlocked rings (see Fig. 3.5 left, blue points). Fig. 3.5 right shows the results, i.e. the Y-weights, of the CCA trained on the the whole dataset ($epochs = 10$, $\lambda_{offline} = 1$); as expected, the offline CCA breaks and separates the two rings (i.e. data unfolding).

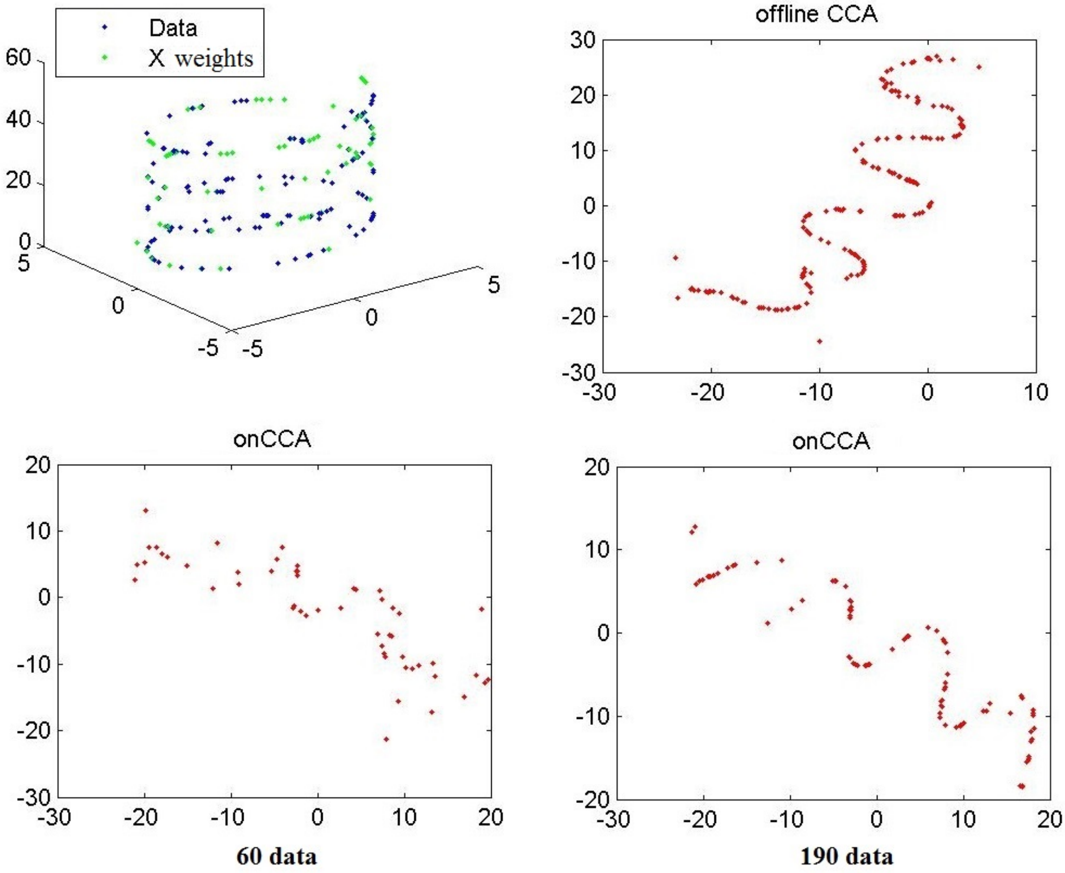


Figure 3.4: First simulation: the spiral dataset. Top left shows the input data (blue points) together with the result of the onCCA quantization (green points); the projection of a classical offline CCA on the whole dataset is displayed in top right; bottom figures show the onCCA projection after 60 data (left) and on the whole dataset (right).

Fig. 3.6 shows the onCCA outputs at different training intervals in two scenarios: good initial CCA (top row) and bad starting projection (bottom row). The parameters are the same for the two experiments: $\rho = 0.02$, $\alpha_1 = 0.4$, $\alpha_n = 0.05$, $age_{max} = 3$, $\delta = 1$, $\lambda = 1$, $projection\ steps = 5$. In the former case, the output of CCA is already well unfolded; onCCA just preserves this configuration over time. In the latter case, even if onCCA starts from a wrong unfolding, it is able to fix it after only 600 training samples and to preserve it until the end of the input set (1900 data).

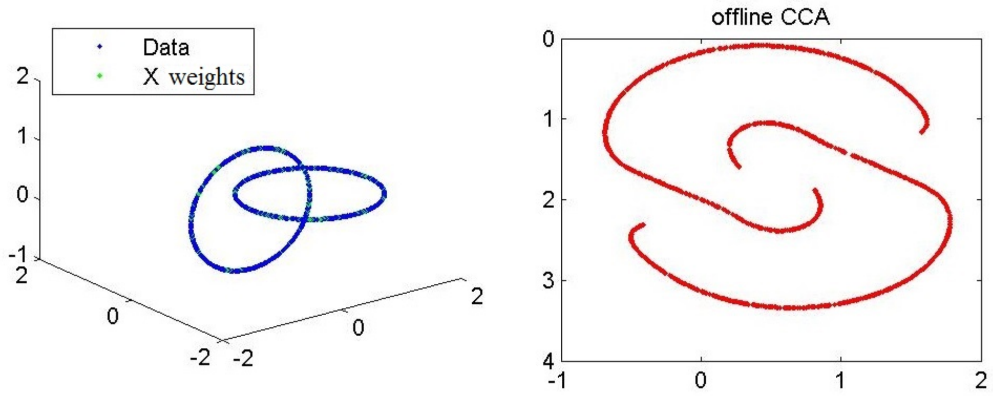
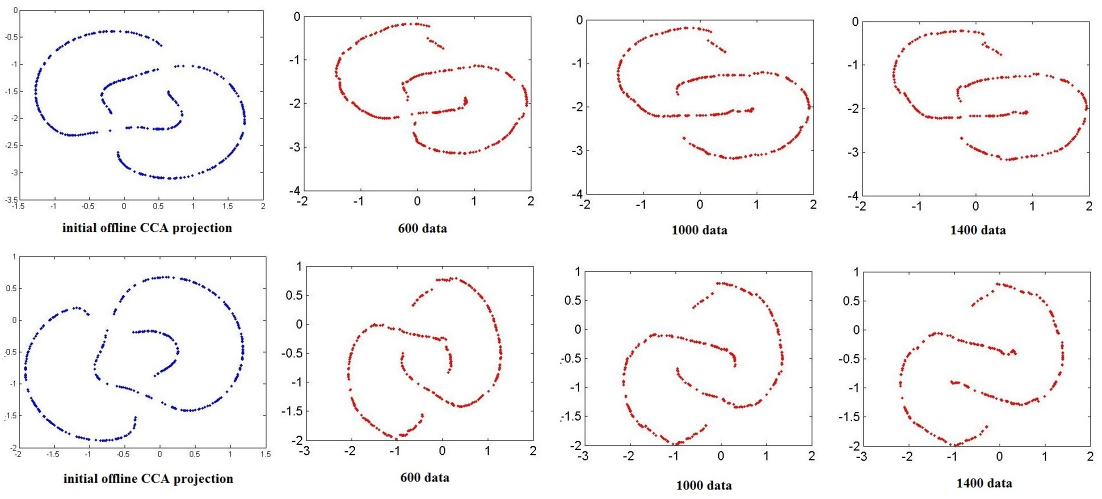


Figure 3.5: Second simulation: the interlocked rings dataset. Left figure shows the input data (blue points) together with the result of the onCCA quantization (green points). Right figure displays the output of a classical offline CCA on the whole dataset.



initial offline CCA projection

600 data

1000 data

1400 data

initial offline CCA projection

600 data

1000 data

1400 data

Figure 3.6: Second simulation: the interlocked rings dataset. The onCCA output over time starting from two different conditions: rings properly separated from the initial CCA (top row) and wrongly unfolded (bottom row).

The previous simulations have been performed on noiseless datasets. To test the network robustness to noise, a final simulation has been done adding a Gaussian noise ($\mu = 0, \sigma = 0.1$) to a subset (900 points) extracted from the intelocked rings, see Fig. 3.7 top left. The onCCA hyperparameters are as in the previous simulation. Fig. 3.7 top right yields the X-weight quantization of onCCA (green points) together with input data (blue points). Fig. 3.7 bottom shows the whole dataset projection by CCA (left) and onCCA (right), respectively. It can be observed the robustness to noise of onCCA and also the higher accuracy of its projection w.r.t. CCA.

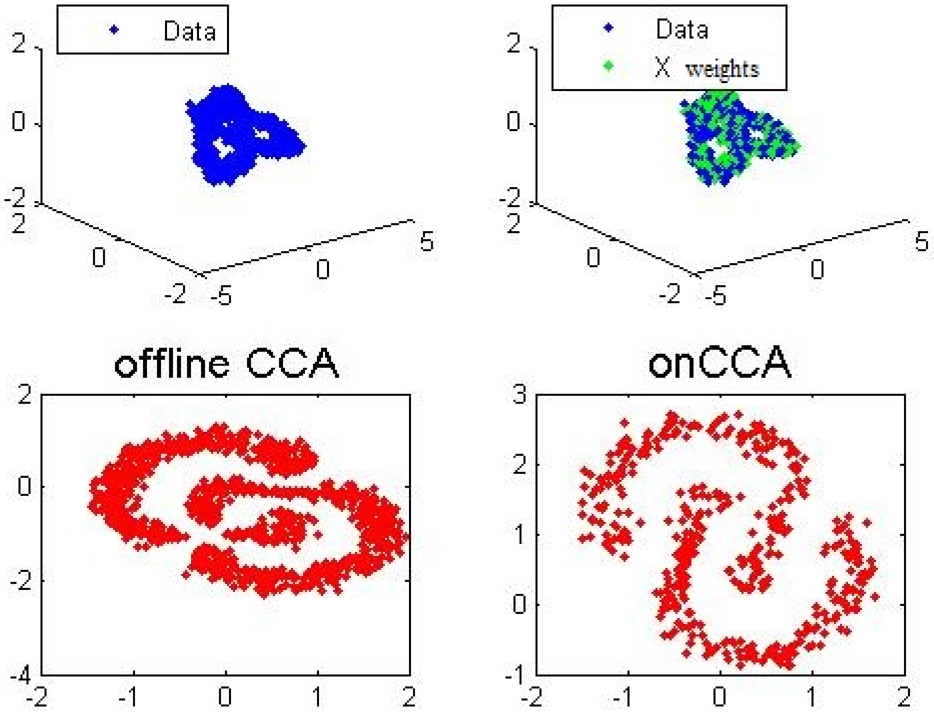


Figure 3.7: Third simulation: the interlocked rings dataset with Gaussian noise ($\mu = 0, \sigma = 0.1$). Top left shows the input data (blue points). Top right displays data (blue points) together with the result of the onCCA quantization (green points); bottom figures show the projection of the whole dataset (900 samples) yield by CCA (left) and onCCA (right), respectively.

The experimental results allow to make some general considerations about the onCCA. First, the initial CCA projection does not need to be much accurate, as demonstrated in the latter experiment (see Fig. 3.7). The network has the same characteristics (data unfolding and topological preservation) of the classical offline CCA, which are robust w.r.t. the input randomness. The onCCA is also robust to noise.

Prognostic application

To test the onCCA performance in a non-stationary context, the network has been fed with samples coming from the FEMTO real accelerated bearing degradation dataset [54]. Data have been gathered by a bearing failure diagnostic and prognostic platform [55], which evolves from an initial healthy condition towards a double fault state, i.e. the input distribution is non-stationary. The training set has 2155 samples and five features statistically extracted by the recordings of four vibration transducers installed on an electrical motor. The network follows the chain behaviour by updating in real time the data projection and by employing a simple variant for detecting the prefault and fault conditions. The latter means novelty detection, while the former is fault prognosis, which is much more interesting in a real scenario, because it allows to stop the machine before the fault occurs completely. Fig. 3.8 shows the CCA projection ($epochs = 30$, $\lambda_{offline} = 26$) of the whole dataset; of course, it is not possible to employ it for novelty detection, because it has been computed after the faults have already occurred. On the contrary, onCCA ($\rho = 0.02$, $\alpha_1 = 0.5$, $\alpha_n = 0.05$, $age_{max} = 3$, $\delta = 9$, $\lambda = 11$, *projection steps* $= 20$) can follow the machine evolution due to its real-time projection, see Fig. 3.9. These results demonstrate the quality of onCCA in non-stationary input tracking.

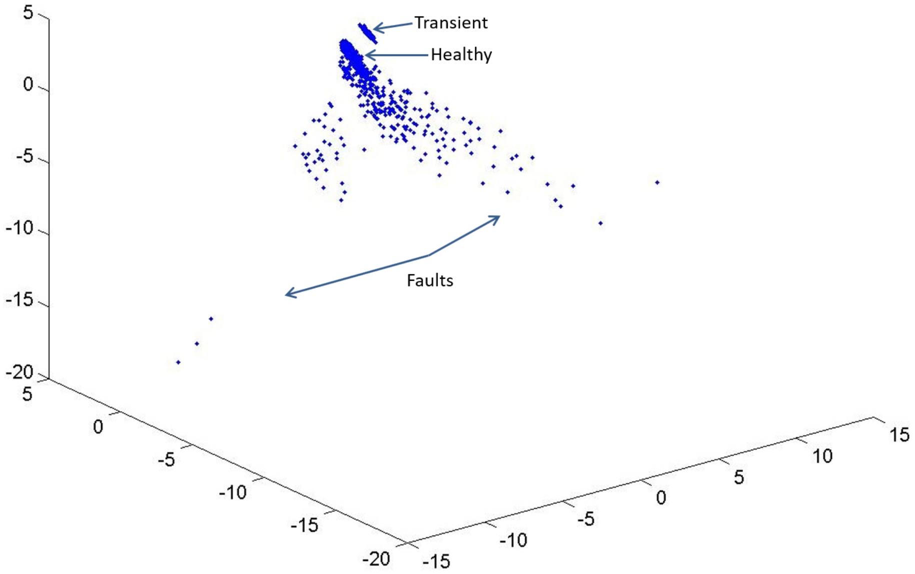


Figure 3.8: Prognostic experiment: CCA projection. Labels and arrows indicate the four bearing states.

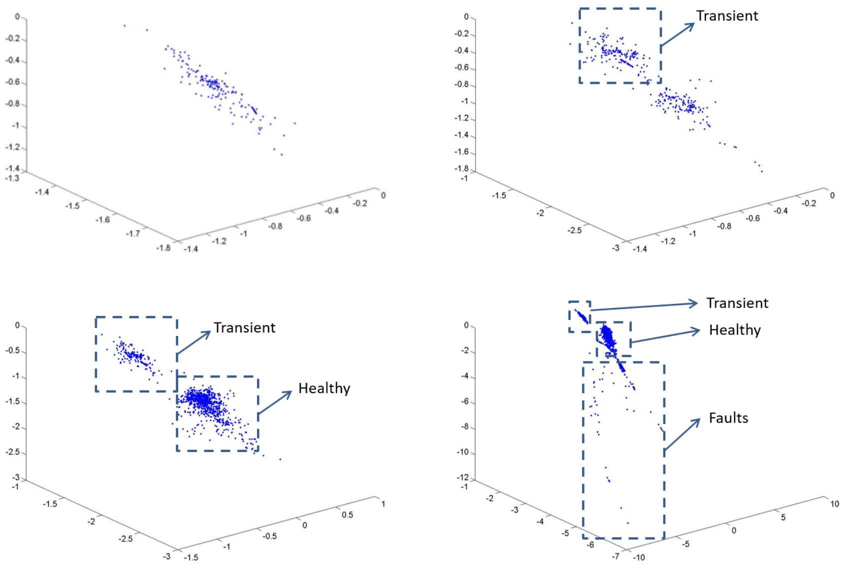


Figure 3.9: Prognostic experiment: onCCA projection over time (from top-left to bottom-right). Labels and arrows indicate the four bearing states.

A quite powerful but simple change in the algorithm can be made to handle outlier data. Instead of simply removing a novel born neuron when the associated *δ-neuron* set is empty, it would be wiser to record this information; indeed, if too many outliers appear consecutively in a short time period, it means the machine is changing its working state. As a consequence, a simple threshold-based mechanism can be employed to monitor when too many consecutive outliers appear in the network, i.e. to detect a possible fault. This idea is confirmed by Fig. 3.10, which illustrates how the number of consecutive outliers significantly increases as the fault progresses. For example, by setting a threshold equal to twelve, the fault onset can be determined after around 1800 samples.

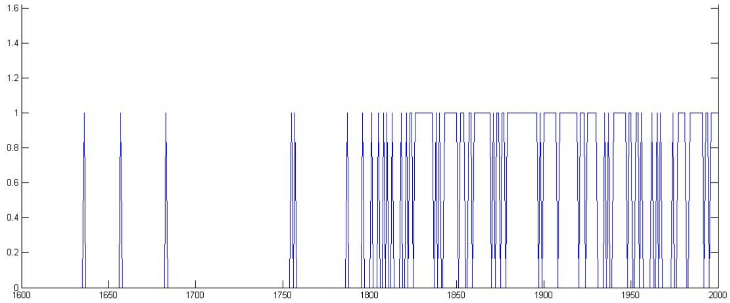


Figure 3.10: onCCA number of consecutive outlier (y-axis) over time (x-axis)

3.3 Growing Curvilinear Component Analysis

The growing CCA (GCCA, [56, 57, 58, 59]) neural network is an improved version of onCCA, which does not need an initial CCA architecture thanks to the *seed* colonization technique. It also uses a special kind of links, called *bridges*, to detect changes in the data stream, i.e. to track non-stationarity. As for its parents, i.e. onCCA and CCA, also the GCCA is a neural network for data projection and employs the same projection rules (2.3) and (2.4) and the concept of λ-neurons. Moreover, GCCA neurons have also two weight vectors, one in the input space, X, and another in the output space, Y; the X-weights determine the input space quantization, while the Y-weights provide the projection.

The receptive field in the X space of a generic neuron w_i is modelled by means of an automatically tailored novelty threshold T_{w_i}; in this sense, it can be seen as a local version of onCCA ρ hyperparameter. T_{w_i} is computed as the largest distance between w_i^X (the neuron X-weight vector), and its neighbours N_{w_i}. Neurons are connected by links which determine the neighbourhoods and, therefore, the induced manifold topology both in the input and output spaces.

GCCA uses two types of links: edges, which follow the CHL and define neighbourhoods, and bridges. The former are bidirectional and are employed to map stationarity and define the topology, while the *bridges* are directional connections to track non-stationarity in the input distribution.

GCCA is an incremental network; as onCCA, it creates a new neuron when the X-quantization is insufficient for explaining a new data, and prunes both links and neurons when they are not useful anymore for mapping the input manifold. The X-weight adaptation is performed using the SCL.

A *seed*, i.e. a couple of linked neurons, is employed to depict a change in the input data stream. The neuron pair is used for colonizing a novel region of the input space, which has not been already seen from the network; in this sense, the new data are coming from a portion of the input manifold too far from those already fed to the network, i.e. they represent a non-stationarity. A seed is created by the neuron doubling method, which creates a new neuron over an existing one, and adjusts its weight vectors by means of hard competitive learning (HCL, [37]) and neuron projection. GCCA begins with an initial seed, whose weights are, in general, random or set equal to the first two samples of the dataset.

3.3.1 The algorithm

Each time a new data, say x_0, is fed to the GCCA, its training algorithm is performed in order to adapt the network topology to the new input, as shown in Fig. 3.11. First, all neurons are ranked w.r.t. their Euclidean distances in the X space from the input x_0. The closest unit is the first winner w_1, the second closest neuron is the second winner w_2 and so on; similarly, their distances from x_0 are d_1,

d_2, etc. Then, the novelty test is performed: if d_1 is higher than T_{w_1} (the neuron receptive field radius), a new neuron is added to the network; otherwise, the data is associated to the first winner Voronoi and both the weights and the network topology, i.e. the links, are updated accordingly.

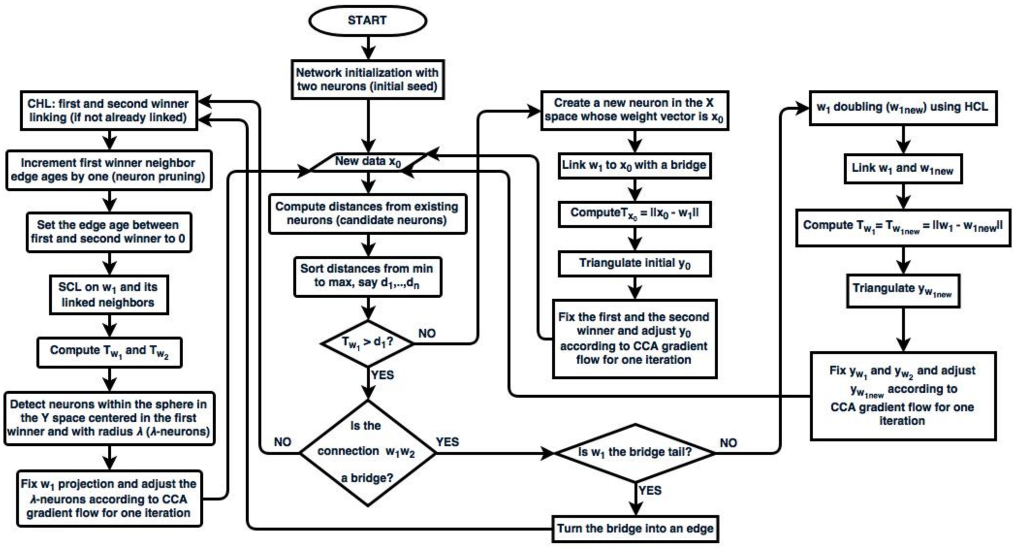

START
Network initialization with two neurons (initial seed)
CHL: first and second winner linking (if not already linked)
Increment first winner neighbor edge ages by one (neuron pruning)
Set the edge age between first and second winner to 0
SCL on w_1 and its linked neighbors
Compute T_{w_1} and T_{w_2}
Detect neurons within the sphere in the Y space centered in the first winner and with radius λ (λ-neurons)
Fix w_1 projection and adjust the λ-neurons according to CCA gradient flow for one iteration
New data x_0
Compute distances from existing neurons (candidate neurons)
Sort distances from min to max, say $d_1,..,d_n$
$T_{w_1} > d_1$?
NO
YES
Is the connection w_1w_2 a bridge?
NO
YES
Create a new neuron in the X space whose weight vector is x_0
Link w_1 to x_0 with a bridge
Compute$T_{x_0} = \|x_0 - w_1\|$
Triangulate initial y_0
Fix the first and the second winner and adjust y_0 according to CCA gradient flow for one iteration
Is w_1 the bridge tail?
NO
YES
Turn the bridge into an edge
w_1 doubling (w_{1new}) using HCL
Link w_1 and w_{1new}
Compute $T_{w_1} = T_{w_{1new}} = \|w_1 - w_{1new}\|$
Triangulate $y_{w_{1new}}$
Fix y_{w_1} and y_{w_2} and adjust $y_{w_{1new}}$ according to CCA gradient flow for one iteration

Figure 3.11: GCCA algorithm

Neuron creation

Each time the first winner novelty test is passed w.r.t an input x_0, a new neuron w_{x0} is created, whose X-weight w_{x0}^X is set equal to x_0. The first winner and the novel neuron are linked by a bridge from w_1 towards w_{x0}. $T_{w_{x0}}$ is set equal to d_1. The corresponding Y-weight is estimated in two steps. First, the initial projection w_{x0}^Y is computed: a triangulation technique (see [56], Appendix) inspired by [60] is used, where w_1^Y and w_2^Y are the centres of two circles, whose radii are the d_1 and d_2, respectively. The circles intersect in two points; the farthest from w_3^Y is chosen for providing the initial first two components of w_{x0}^Y. If the Y space dimensionality is higher than two, the other w_{x0}^Y components are chosen randomly. Then, the weight adaptation (2.4) is performed; here, w_1^Y and w_2^Y are considered as fixed in order to extrapolate w_{x0}^Y. Finally, the training procedure restarts with a new sample from the input distribution.

Adaptation, linking and doubling

If w_1 novelty test is failed, i.e. if $d_1 \leq T_{w_1}$, the network weight vectors in both spaces need to be updated to take into account of x_0.

If w_1 and w_2 are not connected with a bridge, they are linked by an edge. The link ageing procedure is the same as onCCA. Then, w_1^X and $\{w_i^X\}_{i \in N_{w_1}}$ are adapted using (3.1), i.e. SCL, and T_{w_1} and T_{w_2} are recomputed because of the updates. The Y-weights whose distance from w_1^Y is less than the hyperparameter λ are the λ-*neurons* (i.e. GCCA constraints short distances). Then, w_1^Y is fixed, and the new λ-neuron projections are interpolated by means of (2.4), and the training procedure restarts with a new sample from the input data stream.

On the contrary, if w_1 and w_2 are connected with a bridge, it is checked if w_1 is the bridge tail, i.e. the bridge departs from the first winner; in this case, the bridge is converted into an edge and the algorithm proceeds as in the previous case. Otherwise, a seed is created using the neuron doubling method: a new neuron $w_{1_{new}}$ is created on top of the first winner; $w_{1_{new}}^X$ is computed using HCL (i.e. only (3.1a) is used); w_1 and its double are linked with an edge, whose age is set to zero, and both their novelty thresholds are set equal to their Euclidean distance; $w_{1_{new}}$ is considered as a novel neuron and its projection is extrapolated as in the neuron creation scenario, where w_1^Y and w_2^Y are considered as fixed. Then, the training procedure restarts with a new sample from the input distribution.

Analysis of the hyperparameters

The GCCA technique needs very few hyperparameters to be tuned for the problem at hand. They can be grouped in two sets corresponding to the input and output weight vector updating. The latter group regards the data projection, which depends on α, i.e. the learning rate, and λ, i.e. the projection constraint. The learning rate influences the Y-projection through (2.4): the lower its value the less the network plasticity; on the contrary, when the input data stream is changing quickly, a higher value should be used, so that the network could promptly response, i.e. adapt, to the non-stationarity. This hyperparameter can be automatically increased when non-stationarity is detected, e.g. monitoring the density and length of bridges over time; then, when the input stream moves to a stationary condition, α can be lowered again to increase the projection accuracy. The second projection hyperparameter is λ, which is the most crucial GCCA parameter because it strictly related on the input manifold: when it is linear, λ can be set to ∞, because all the pairwise input distances can be preserved in the output space; conversely, the more the input manifold is non-linear w.r.t. the latent space dimensionality, the lower λ have to be set, which causes only small distances to be respected. A proper tuning of this hyperparameter is fundamental because a too low value could lead to a multitude of local projections without any global coordination. To handle this issue, the original CCA performs λ manual tuning by visual inspecting the *dy-dx* diagram. In alternative, as per the novelty thresholds T_{w_i}, each neuron can be equipped with a local λ_{w_i}, whose value can be automatically computed exploiting the distance pairs *dy-dx*, i.e. the projection accuracy. The

λ_{w_i} can also be thought as a way to represent the input manifold local curvature. Finally, in the original CCA, data are stationary and fully available before the network training starts; hence, the projection rule (2.4) can be applied for multiple iterations until the projection converges. On the contrary, GCCA is designed to be a fast responsive network working on real-time non-stationary data, which forbid to use (2.4) multiple times.

The second group of hyperparameters regards the SCL and the link and neuron pruning. The former comprises α_1 and α_n and the same considerations for onCCA hold, in particular, $\alpha_1 >> \alpha_n$; $\alpha_n = 0$ yields to HCL. To quickly adapt to non-stationarity, these two hyperparameters are set as constant for maintaining network plasticity. In alternative, α_1 can be made local to each unit, ruled by a decreasing exponential law. In this sense, this leads a higher quantization accuracy at the expense of a more rigid network; however, this rigidity should be removed in case of non-stationarity, which can be detected by the appearance of bridges. For each new neuron, the associated α_1 could be set to a user-dependent initial value or equal to the corresponding bridge length; indeed, in case of a severe non-stationarity in the input data stream, α_1 gets a larger value, which implies a more flexible architecture.

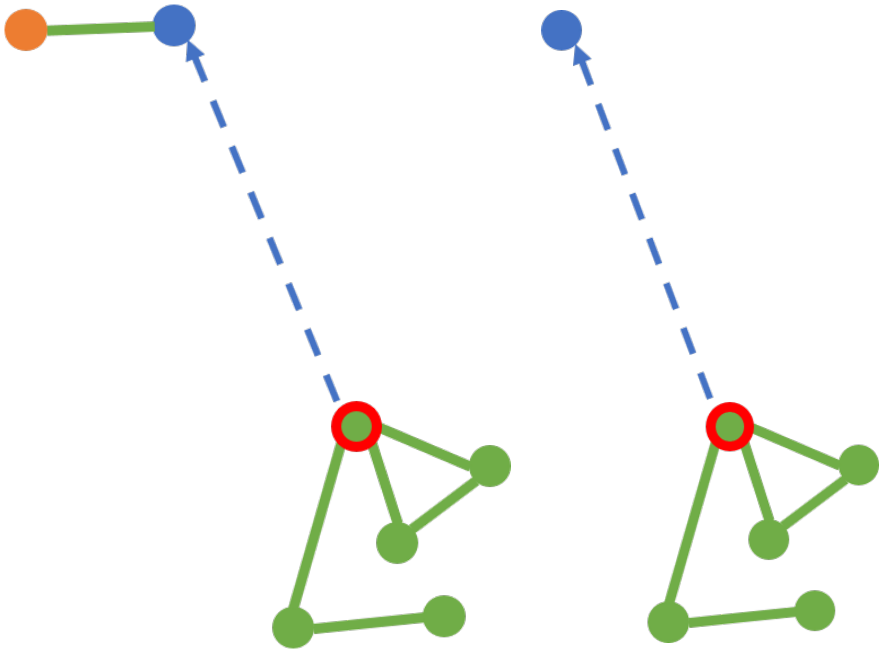
The latter hyperparameter is age_{max}, i.e. the global pruning threshold: a low value implies a shorter memory but also a more flexible network. It can be automatized by monitoring rate variance of bridges over time: if it increases, there is more novelty in the input samples and the age_{max} value could be decreased to better approximate the input distribution.

Resuming, GCCA requires only five hyperparameters (α, λ, α_1, α_n, age_{max}), which could also be completely automated.

Analysis of Bridges

Bridges are the GCCA tool to detect and follow non-stationarity. They are directional connections from an existing neuron towards the non-stationarity, i.e. the new neuron; in this sense, they point to the input change, e.g. a jump. Of course, they can also represent outliers, e.g. noise. If a bridge is long (see Fig. 3.12), two scenarios can occur: the bridge top neuron has doubled, i.e. there was an effective change in the input distribution (see Fig. 3.12a); otherwise, if the top neuron has no edges, it represents an outlier, as shown in Fig. 3.12b.

The bridge density provides an additional awareness of the time-varying distribution. When data change abruptly, i.e. a jump, there will be a limited quantity of long bridges. If the input distribution moves smoothly, the bridge density is related to the displacement velocity. In case of very slow displacement, only the border neurons (the input manifold frontier) become first winners and are moved on average in the direction of the displacement, while the remaining units do not move. Very slow displacement yields no bridges.



(a) Novelty detection

(b) Outlier

Figure 3.12: Bridge length analysis. The green neurons and their edges (green solid segments) represent the network before the new neuron (blue circle) is created. The red circled unit was the first winner when blue neuron was created. If the bridge (dashed arrow) length is long, there are two interpretations for the blue neuron: (a) novelty detection, i.e. it is the seed for the colonization of a novel input region (a non-stationarity), proved by the blue neuron doubling (orange unit); (b) outlier, which means the neuron remains sterile because it has not started a colonization of a novel input region.

3.3.2 GCCA testing

The GCCA performance has been compared with those of its ancestor, i.e. the onCCA, and with two other non-linear methods for non-stationary input distribution available in literature: the Dynamic Self-Organising Map (DSOM) [43] and the Online VIsualization Neural Gas (OVI-NG) [41]. First, as onCCA, GCCA has been tested on the 3-D spiral distribution to check its unfolding property. Then, the GCCA network ability to deal with non-stationary input manifolds has been assessed using a synthetic 3-times jumping square and the prognostic dataset seen before (see Sec. 3.2.2): the former simulation has been used to test the GCCA ability of discriminating between stationarity and non-stationarity conditions and its results have been compared with those of DSOM; in the latter experiment, both GCCA and OVI-NG have been trained on the prognostic dataset for assessing their performance in online data non-linear projection of non-stationary input distribution. Before showing the experimental results, a brief description of DSOM and OVI-NG neural networks is presented.

DSOM

DSOM is a variant of the Self-organizing maps (SOM) designed to remain plastic over time, by employing constant ϵ and η parameters instead of decreasing laws as SOM; in this sense, it can be used for tracking non-stationarity because the network flexibility allows to adapt to change in input data. As SOM and GCCA, neurons have two weight vectors: the first in the input space to quantize it, while the others in the Y-space are constrained to a fixed topology (i.e. a predefined grid). The quantization law is given by:

$$w_i^X(t+1) = w_i^X(t) + \epsilon \left\| x_0 - w_i^X(t) \right\| h_\eta(i, i_{w_1}, x_0)(x_0 - w_i^X(t)) \tag{3.2}$$

where

$$h_\eta(i, i_{w_1}, x_0) = e^{-\frac{1}{\eta^2} \frac{\left\| w_i^Y(t) - w_1^Y(t) \right\|^2}{\left\| x_0 - w_1^X(t) \right\|^2}} \tag{3.3}$$

and x_0 is the new data in X-space; i and i_{w_1} are the indices of the *i-th* neuron and the first winner, respectively; ϵ and η are the constant learning rate and elasticity (or plasticity) hyperparameters.

The learning rate is locally moduled by two factors: the input distance and the Gaussian (3.3) centred in the first winner projection and with variance equal to the distance between w_1 and the data in the input space. If an X-vector is close enough to data, there is no need for others to learn anything (the winner can represent the data), i.e. the $h_\eta(i, i_{w_1}, x_0)$ assumes a very low value; if there is no X-weight close enough to the data, any unit learns the data according to its own distance to it. The first fact prevents DSOM from fitting the magnification law: the quantization does not capture the data density, what is actually mapped by DSOM is the structure or support of the distribution rather than the density. The second one implies new data (from a changing environment) attract all X-weights. As a consequence, their positions (in input and output space) change and represent only the new information (memoryless network). Finally, if $x_0 \simeq w_1^X$ then $h_\eta(i, i_{w_1}, x_0) \simeq 0$ and DSOM becomes time-invariant.

DSOM is designed to track only the most recent representation of the input manifold. It maps each new configuration of the input data stream with a short transient error due to the input change.

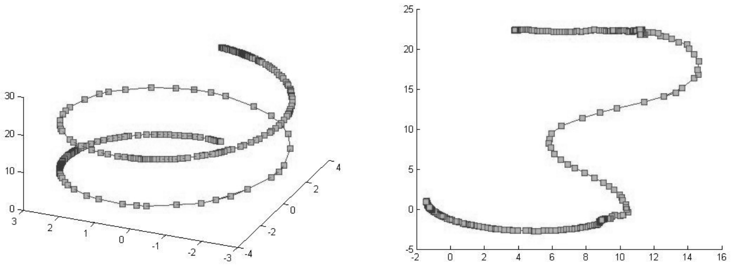
OVI-NG

The online visualization neural gas (OVI-NG) is a non-linear projection neural network based on the neural gas quantization with CHL linking, i.e. TRN, and

CCA. Quantization and projection are performed simultaneously. The quantization is not incremental: the amount of neurons and the corresponding initial conditions need to be defined in advance. The projection rule is the same as CCA, which is designed for offline problems: indeed, the network parameters decrease over time in an exponential way from an initial value to a final one, both set from the user. Therefore, twelve hyperparameters are required: initial and final values of the link lifespans, the NG learning rate, the CCA learning rate and the neighbourhood size in both spaces; the number of unit; the maximum time for the temporal schedules. OVI-NG is strictly dependent on time; thus, it is not well suited for non-stationary input. Moreover, having a predefined architecture forbid to track the underlying manifold. However, it is here employed for comparison because it uses the same projection rule as GCCA and has also some features in common with it. Because OVI-NG is not incremental, there is no neuron pruning; indeed, CHL is only used for visualization purposes.

Synthetic experiments

The first experiment deals with the same dataset used to test the onCCA unfolding property, i.e. 3-D spiral, as shown in Fig. 3.13. The GCCA hyperparameters are set equal to: $\alpha = 0.001$, $\lambda = 20$, $\alpha_1 = 0.4$, $\alpha_n = 0.1$, $age_{max} = 2$. The quantization (see Fig. 3.13a) covers the input manifold uniformly and the projection (see Fig. 3.13b) unfolds the spiral correctly. This result demonstrates that GCCA has the same behaviour as CCA and onCCA when the input is stationary.

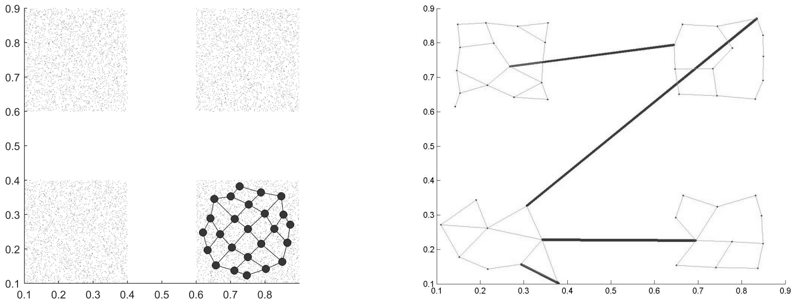


(a) Input quantization

(b) Data projection

Figure 3.13: First simulation: the 3-D spiral dataset

The second simulation deals with data drawn from a synthetic square whose domain jumps three times (from top left to top right, then from top right to bottom left and, finally, from bottom left to bottom right) during network training. In this case, both DSOM and GCCA have been trained and their results compared (see Fig. 3.14) to test their behaviour in handling non-stationarity. Fig. 3.14a yields the DSOM projection ($\epsilon = 1.5$, $\eta = 5.5$). The network, i.e. its neurons and edges, only maps the last domain of the input; the previous information is completely lost, i.e. DSOM is memoryless. The same behaviour can be found in other unsupervised neural networks, like SOM, when using constant hyperparameters. DSOM handles the input non-stationarity by means of the migration of all its units into the new domain, because their amount is fixed in advance. Fig. 3.14b illustrates the GCCA projection ($\alpha = 0.005$, $\lambda = 2$, $\alpha_1 = 0.01$, $\alpha_n = 0.0001$, $age_{max} = 8$) on the same dataset. Conversely to the previous case, the resulting quantization stores all the positions of the input distribution (the grid deformations derive from too few sample presentations before each jump). The GCCA incremental approach yields to representing the four different domains without choosing in advance the amount of network neurons. GCCA pruning only works in the current domain, because in the previous regions there are no more units who are selected as first winners; indeed, the age of the corresponding links remains static, i.e. the previous quantization is as frozen. The appearance of single long bridges signals a jump, i.e. an abrupt change, in the input data stream.



(a) DSOM output

(b) GCCA output

Figure 3.14: Second simulation: 3-times jumping square. DSOM output is on the left: neurons are circles, links are segments. GCCA projection is on the right: points are neurons, thin segments are edges, thicker segments are the bridges.

Real time experiments

Detecting faults, i.e. non-stationarity, in the input distribution in real time is an important feature. Detecting prefault conditions is even more important because allows to stop machines before a severe fault occurs, i.e. to avoid machines being damaged. To this purpose, in the following are shown two applications to real time prognostics: a stator fault in an induction machines, both in an open-circuit [58] and an increasing load [59] configurations, and the FEMTO bearing fault dataset [54, 55].

The GCCA is trained on a dataset made of three-phase current acquired from sensors on the stator windings of an induction motor (IM) while different inter-turn short circuit faults are induced over time. Several configurations have been studied: an open-circuit, with fault severity increasing from 0% to 30% and three load conditions (no load, 25% load and 40% load) where severity ranges from 0% to 10%.

In the former experiment, a 3-phase Squirrel cage IM of 1.1kW rating and connected to a 60Hz voltage supply is used. By preprocessing with a Tukey filter the input quality is improved by means of an increased signal to noise ratio (SNR).

Fig. 3.15 shows the space vector representation of the dataset, where the three-phase input is transformed into direct and quadrature currents. There is an initial transient depicted as decreasing spirals followed by a steady state of 1 s; then, every second, the percentage of fault rises by 5%, up to 30%. The fault evolution over time is clearly observable in the figure; indeed, the trajectories follow the same loci as in the healthy situation but with larger radii as the fault progresses.

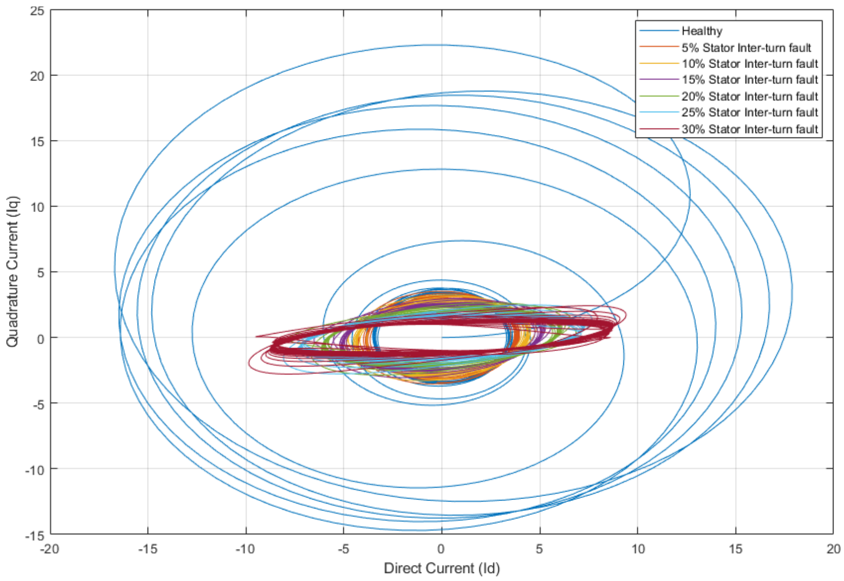


Figure 3.15: First experiment: stator inter-turn short circuit, open circuit evolving fault. Space Vector Loci. External light blue decreasing spirals represent the initial transient.

The parameters of GCCA are the following: $\alpha = 0.01$, $\lambda = 0.5$, $\alpha_1 = 0.2$, $\alpha_n = 0.04$, $age_{max} = 4$. GCCA is trained with the phase current information and performs a 2-D projection in real time. Fig. 3.16 shows the X-quantization. The trajectories have been modelled (tracked) accurately, spirals and circles are visible.

Fig. 3.17, instead, illustrates the projection together with the links. The first transient is depicted by small edges and bridges, which are also orthogonal to the true current projection. The transient is too fast to be learnt properly by the network (no pruning occurs). However, a compact network is yielded, which is typical of self-organization architectures. As the transient progresses, more and more bridges follow the current changes, which can be derived by their density. If the time change is abrupt, much more bridges appear by means of internal denser spirals. GCCA clearly detects the pre-fault condition using its anisotropic connections, i.e. the bridges; furthermore, it tracks the whole machine evolution over time.

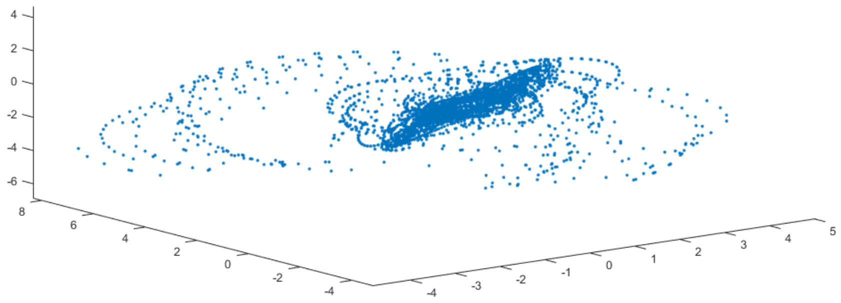


Figure 3.16: First experiment: stator inter-turn short circuit, open circuit evolving fault. GCCA quantization.

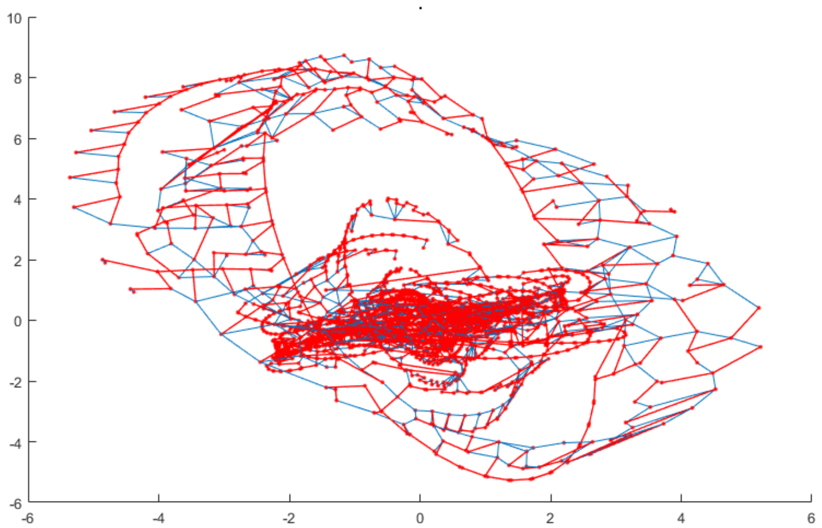
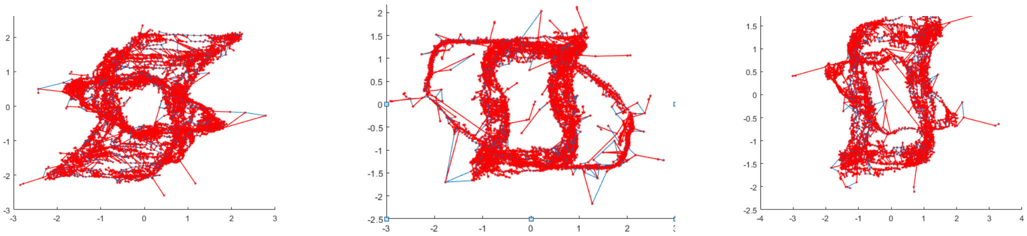


Figure 3.17: First experiment: stator inter-turn short circuit, open circuit evolving fault. GCCA projection: edges are blue, bridges are red.

A second test rig has been set up to further deepen the stator fault application. The test rig is made up of a 3-phase squirrel-cage IM of 1.1kW connected to a SEMIKRON IGBT Voltage Source Inverter (VSI) of 12 kVA. LEM (LA 55-P) current transducers are used to acquire 3-phase current signals via DS1104 card (dSPACE) at a sampling frequency of 10 kHz. The IM electrical drive is controlled by using a scalar control [61]. The experimental test rig was located in the PowerTech laboratory of the School of Engineering and Physics at the University of the South Pacific. Fig. 3.18 shows the GCCA ($\alpha = 0.05$, $\lambda = 0.3$, $\alpha_1 = 0.4$, $\alpha_n = 0.04$, $age_{max} = 4$) result together with its links (edges are blue, bridges are red), in case of machine with no load (3.18a), with 25% load (3.18b) and with 40% load (3.18c), respectively. The quantization is quite accurate; indeed, all different time intervals are well represented, despite the rapidity of the fault. Furthermore, the bridges track the current changes and are denser when the non-stationarity increases. Some outliers are also shown because, in this experiment, no outlier detection method have been employed. The GCCA output can be exploited in several ways: for example, an increasing amount of bridges signals the onset of a pre-fault condition in the machine. As the fault severity increases, more and more bridges appear to follow the changes in the input distribution. This is well depicted by the change of the hysteresis-like pattern, as the fault progresses. Fig. 3.18 also illustrates the differences between non-stationarity; for example, the shape of the healthy and faulty states are well separated, especially in the 25% and 40% load conditions, i.e. any fault can be easily detected. Finally, also in this case, GCCA does not only detect the pre-fault scenario, but also stores the complete machine evolution over time.



(a) No load

(b) 25 % load

(c) 40 % load

Figure 3.18: First experiment: stator inter-turn short circuit, variable load. GCCA output: edges are blue, bridges are red

In the latter experiment, both GCCA and OVI-NG have been trained on the prognostic dataset for assessing their performance in online data non-linear projection of non-stationary input distribution. The OVI-NG is trained with constant parameters because of the non-stationary input. The configuration yielding the best results have been obtained as follows: 150 neurons; both the CCA and NG learning rates are equal to 0.25; the neighbourhood widths in the X and Y spaces are set to 35 and 70, respectively. The corresponding projection is shown in Fig. 3.19.

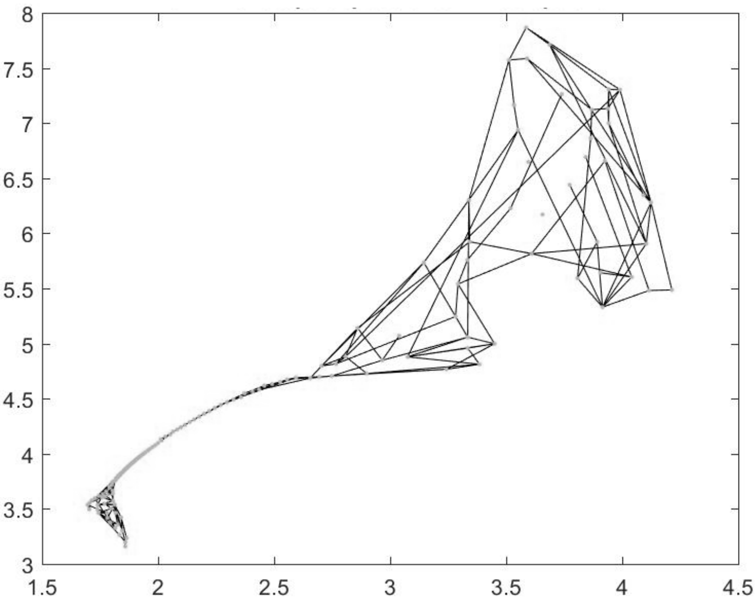


Figure 3.19: Second experiment: prognostic dataset. OVI-NG output

The best GCCA neural network, whose results are shown in Fig. 3.20, uses the following hyperparameters: $\alpha = 0.01$, $\lambda = 1.6$, $\alpha_1 = 0.05$, $\alpha_n = 0.005$, $age_{max} = 2$. The GCCA learns the chain behaviour and tracks it, by adapting in real time the data projection. Fig. 3.20 displays the full bearing lifecycle, from the initial transient phase, through the healthy state, towards, first, a pre-fault (characterized by an increasing bridge density), and, finally, the two faults which are clearly identified in the figure by the longer bridges.

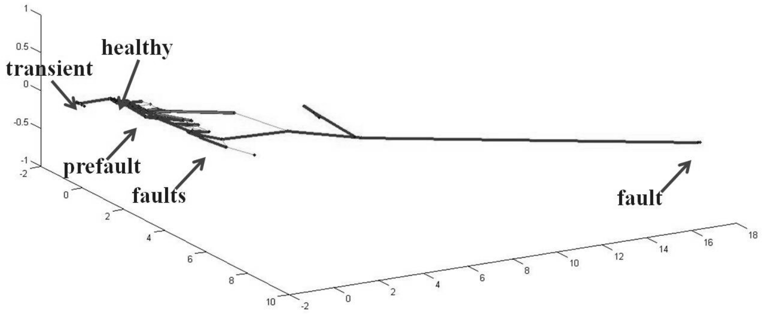


Figure 3.20: Second experiment: prognostic dataset. GCCA output

Both neural networks learns well the first two phases. However, they perform differently at the fault onset as shown in Fig. 3.21. OVI-NG begins to oscillate, while GCCA increases the number of bridges. This behaviour is reflected into the fault learning: OVI-NG only underlies oscillations, while GCCA shows two branches, one per each fault.

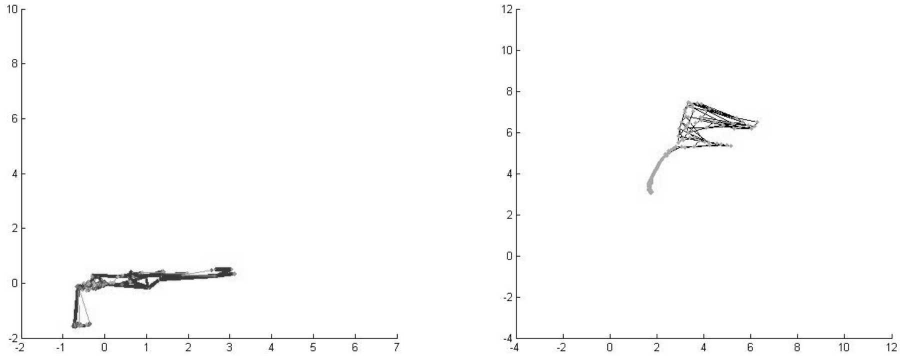


Figure 3.21: Second experiment: prognostic dataset. GCCA (left) and OVI-NG (right) comparison at the fault onset.

The advantage of GCCA versus OVI-NG does not only rely on its representation properties, but also on the accuracy of the quantization and projection. In this sense, the trustworthiness [62] and continuity [63] indices are used to evaluate the quality of the two networks.

A projection (map) onto a latent space is said to be *trustworthy* if the set of k nearest neighbours of a point in the map are also close in the original space. Let $U_k(i)$ be the set of data samples that are in the neighbourhood of the i-th point in the map but not in the original space. The measure of trustworthiness of the projection, M_1, is defined as:

$$M_1(k) = 1 - A(k) \sum_{i=1}^{N} \sum_{j \in U_k(i)} (r(x_i, x_j) - k) \tag{3.4}$$

where

$$A(k) = \frac{2}{Nk(2N - 3k - 1)} \tag{3.5}$$

N is the total number of neurons and $r(x_i, x_j)$ is the neuron ranking in input space.

A projection onto an output space is said to be *continuous* if the set of k closest neighbours of a point in the original space are also close by in the output space. Let $V_k(i)$ be the set of data samples that are in the neighbourhood of the i-th point in the original space but not in the map. The measure of continuity of the visualization, M_2, is defined as:

$$M_2(k) = 1 - A(k) \sum_{i=1}^{N} \sum_{j \in V_k(i)} (s(y_i, y_j) - k) \tag{3.6}$$

where $s(y_i, y_j)$ is the neuron ranking in output space and $A(k)$ is defined as (3.5).

To track these indices in time, because of the non-stationary nature of the problem, two plots for trustworthiness (Fig. 3.22) and continuity (Fig. 3.23) are given with regard not only to k, as usual, but also to time.

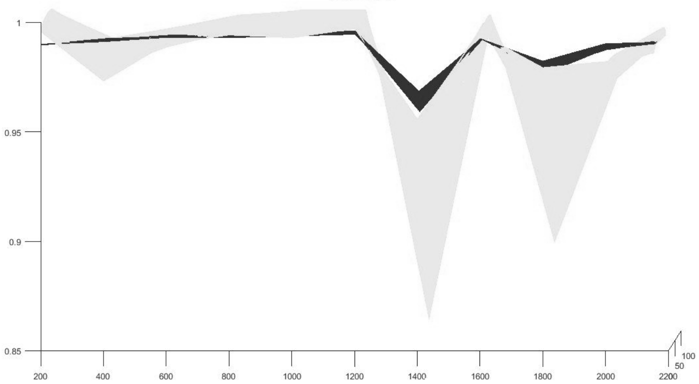


Figure 3.22: Second experiment: prognostic dataset. Trustworthiness as a function of time and k: OVI-NG (grey), GCCA (black).

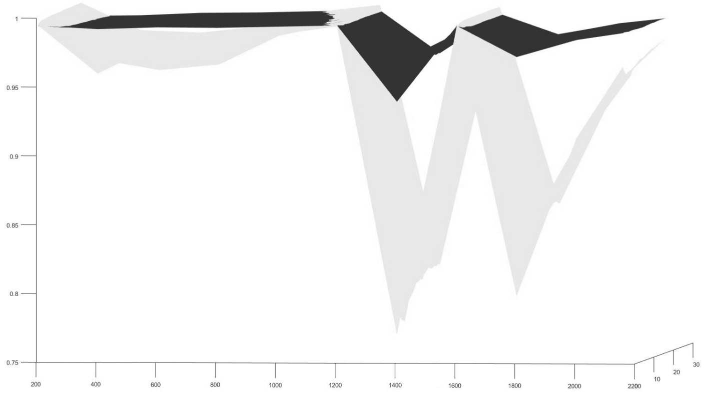


Figure 3.23: Second experiment: prognostic dataset. Continuity as a function of time and k: OVI-NG (grey), GCCA (black).

With regard to the former index, both techniques are very accurate and similar until the fault onset, where OVI-NG shows a big loss in the projection accuracy, that is restored only after a certain number of training iterations. This evidence corroborates the previous analysis about the advantage of using bridges to track non-stationarity, i.e. the fault, with regard to the OVI-NG oscillating behaviour. A similar study can be repeated for the continuity index, with the difference that in the healthy state GCCA performs better; maybe it is related to the incremental nature of GCCA.

Chapter 4

Unsupervised Learning

The fundamental assumption on which online neural networks, e.g. GCCA, are based is that the input topology is properly represented; in other words, the input space quantization plays a crucial role because it provides the solid base on top of which perform the data projection. The topological representation of the input distribution is one of the main objectives of unsupervised learning [64, 65, 66]. Neural networks build a graph of neurons to fill the input manifold. Such an approach typically requires a large number of neurons; as a consequence, it is prone to the curse of dimensionality. The neuron weight vectors represent the graph nodes, which are connected by edges. The weight adaptation is often performed as an error function minimization using CHL, which can be hard (*winner-take-all*, as LBG [67] and k-means [68]) or soft (*winner-take-most*, as neural gas and SOM). The graph edges are either found by using the CHL as in TRN or by back-projecting a fixed grid as in SOM.

As explained in the previous chapter, the ability of dealing in real-time with a non-stationary data stream is crucial in many applications, such as fault prognostic. Neural networks address this problem by means of different approaches w.r.t. their architecture and the application at hand. If the network uses a fixed number of neurons, e.g. DSOM, the only way of tracking non-stationarity is losing the past representation (embedded in the old weight vectors). Hence, they can be employed only if the focus is the latter input configuration. Conversely, incremental networks, e.g. GCCA, increase or decrease the number of neurons each time the input data stream changes, e.g. jumps. The precursor is the Growing Neural Gas (GNG [37]); because unit insertions are defined by the user, it cannot be applied on input whose dynamics is unknown. Its variant, GNG-U [69], is a forgetting network, which employs local utility parameters for estimating the data probability density; the aim is removing neurons in low density regions.

Life-long learning deals with the fundamental issue of how a learning system can adapt to new information without corrupting or forgetting previously learned information, the so-called *Stability-Plasticity Dilemma* [70]. Indeed, it should be able of repeatedly training a network using new data without losing the previous information, i.e. destroying the old nodes. Life-long neural networks must be incremental: the number of neurons grows over time to follow the input distribution; in this sense, the previous neurons become dead units, but represent past knowledge. At this aim, the idea of neuron threshold has been introduced to test if the network needs an additional neuron for explaining a new input sample. This scalar parameter represents the radius of a hypersphere in the input space centred at the weight vector of the winner neuron; in this sense, the test is local and isotropic and maps the region explained by the winner. The threshold can be a global hyperparameter set by the user (IGNG [71]) or estimated locally by the system. In the single-layer Enhanced Self-Organizing Incremental Neural Network (ESOINN [44]) the threshold of a given neuron is computed as the maximum distance to its current neighbours. In AING [72], it is defined as the sum of distances from the neuron to its data-points, plus the sum of weighted distances from its neighbouring units, averaged on the total number of the considered distances. However, this approaches use isotropic thresholds, i.e. the influence region of the neuron depends on the extension of its neighbourhood, but not on its shape. Both onCCA and GCCA employ the same threshold method as ESOINN.

4.1 The G-EXIN neural network

The G-EXIN neural network [73] is an unsupervised, incremental learning system based on the GCCA quantization layer (G-EXIN does not perform data projection); in this sense, it uses bridges and seeds, but improves GCCA by means of an anisotropic representation of the neuron influence region, i.e. its neighbourhood. At this purpose, it exploits the bounded convex polytope, which is the convex hull of the weight vectors of the neuron and its neighbours, i.e. the units connected through edges but not through bridges. Moreover, a novel technique for locally scaling the learning rate in SCL weight adaptation is proposed. Finally, conversely to GCCA, each neuron is equipped also with an *activation flag*, which signals if the neuron weight has changed since its creation, i.e. if the neuron has ever won the competition with the other units to be the first winner for some input data. This mechanism is exploited to better discriminate between stationary and non-stationary region of the input data stream. Indeed, each time the novelty test is passed by an input sample, either the input distribution manifold is expanding or it is moving to a different region of the input space. In the former case, data come *randomly* from a specific region of the input space, i.e. the input distribution is in a stationary condition; during the subsequent training iterations, it is highly probable that the novel born neuron will be either the first winner for some new data or member of the topological neighbour set of some other unit, i.e. because of SCL its weight vector will change. Given the stationary condition, most part of the bridges of the region do not point anymore to a non-stationarity; in this sense, they can be turned into an edge if both the neurons at their ends have the *activation flag* equal to true.

On the contrary, in the second scenario, the input distribution is moving to a different region of the input space, e.g. it jumped. In this case, given a bridge, one end will point to the previous region, which will not change anymore because of the input jump, and the other will be connected to the units in the novel domain of the input space. In such a scenario, the former unit will be as a dead unit and its *activation flag* will never swap from false to true; as a consequence, the link remains a bridge to signal the input non-stationarity.

4.1.1 The algorithm

The initial structure of G-EXIN is a seed, i.e. a pair of neurons connected by an egde, whose weight vectors are the first two samples of the input stream. Each time a new sample, $x_i \in X$, is fed to the network, the training algorithm shown in Fig. 4.1 is executed. All units are sorted according to the Euclidean distances d_i between x_i and their weight vectors. The neuron with the shortest distance (d_1) is the first winner, say w_1; then, the first winner novelty test w.r.t. the new data x_i is performed: if x_i is novel w.r.t. w_1, a new neuron is created; otherwise, the network topology is updated.

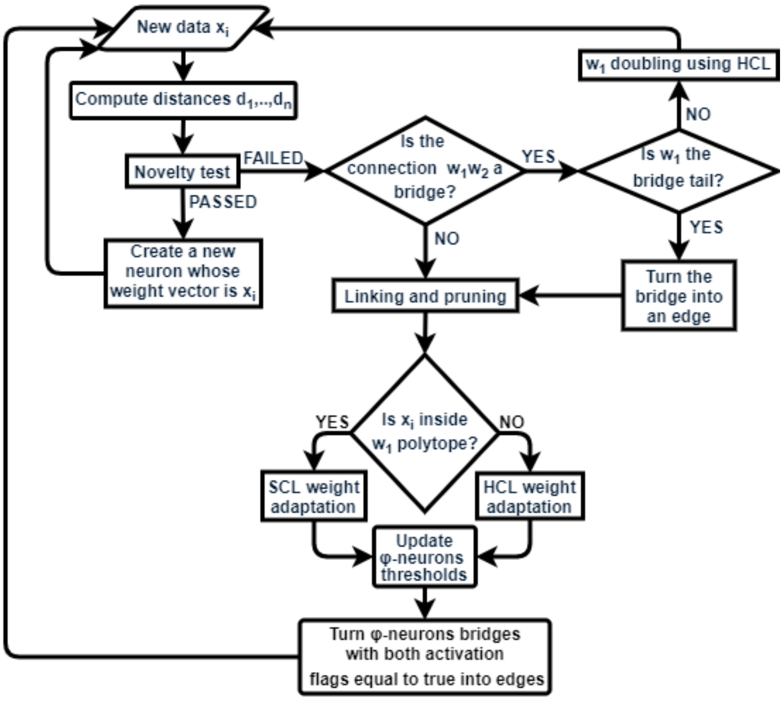

New data x_i
Compute distances d_1,..,d_n
Novelty test
FAILED
PASSED
Create a new neuron whose weight vector is x_i
Is the connection w_1w_2 a bridge?
YES
NO
Is w_1 the bridge tail?
NO
YES
w_1 doubling using HCL
Turn the bridge into an edge
Linking and pruning
Is x_i inside w_1 polytope?
YES
NO
SCL weight adaptation
HCL weight adaptation
Update φ-neurons thresholds
Turn φ-neurons bridges with both activation flags equal to true into edges

Figure 4.1: The G-EXIN algorithm

Novelty test

An input data x_i is considered *novel* w.r.t. neuron γ if the following two conditions are satisfied: their distance $d = \|x_i - \gamma\|$ is greater than the local threshold T_γ and x_i is outside the topological neighbourhood of γ, say N_γ; here, N_γ is modelled by the convex hull (bounded convex polytope) of the weight vectors of γ and its direct topological neighbours. The concept is illustrated in Fig. 4.2 for a 3-D space: the neighbourhood is defined using CHL, i.e. it is the set of neurons directly connected to γ by an edge (green dotted segments); therefore, neurons connected to γ with a bridge (red dotted segment) or neurons connected with an edge to a γ neighbour are excluded from the neighbourhood.

On the other side, T_γ is locally computed as the mean distance between γ and the N_γ neurons:

$$T_\gamma = \frac{1}{|N_\gamma|} \sum_{w_i \in N_\gamma} \|w_\gamma - w_i\| \tag{4.1}$$

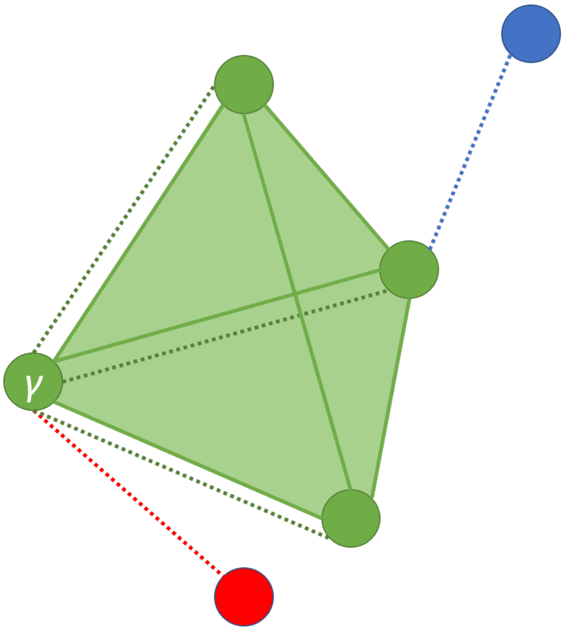


Figure 4.2: 3-D convex hull example: the shaded area delimited by solid lines is neuron γ convex hull; the dotted segments represent connections (edges in green and bridge in red); blue dotted segment points to a neuron (blue point) which is not a direct neighbour and does not belong to N_γ. Only the green neurons belong to the convex hull.

The novelty test proceeds as follows. If $d \leq T_\gamma$, the test is failed; in this sense, T_γ represents the minimal test resolution. Otherwise, the anisotropic convex-hull test is performed, when possible; indeed, if $|N_\gamma| < 2$, then, the convex hull cannot be built and the input data x_i is considered as outside the T_γ sphere, i.e. it is marked as novel.

On the contrary, if γ has at least two topological neighbours then, for the novelty detection, its is checked if x_i is inside the N_γ region, i.e. the convex hull, by means of the following simple and time-efficient anisotropic method (see Fig. 4.3). First, the β_j vectors are computed as follows:

$$\{\beta_j = \delta_j \cdot \Psi\}_{j \in N_\gamma \cup \gamma} \tag{4.2}$$

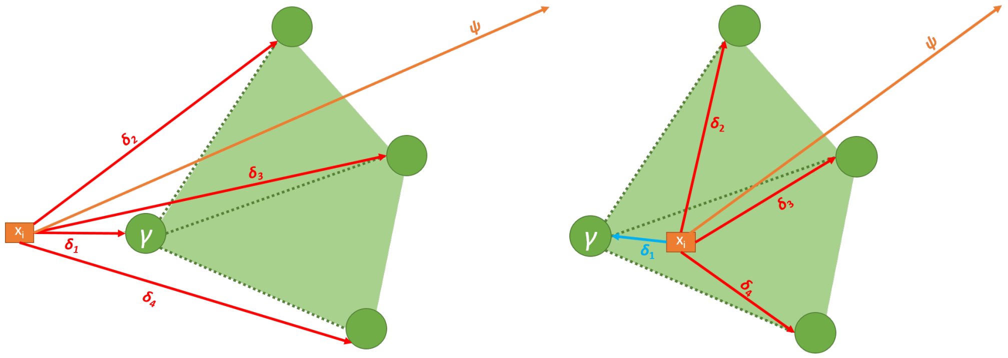
where

$$\Psi = \sum \delta_j \tag{4.3}$$

and

$$\{\delta_j = x_i - w_j\}_{j \in N_\gamma \cup \gamma} \tag{4.4}$$

If all the β_j have the same sign (null products are ignored), then x_i is outside the polytope, i.e. is novel, as shown in Fig. 4.3a. Otherwise, it is inside the polytope, $x_i \subset N_\gamma$, i.e. the novelty test is failed (see Fig. 4.3b).



(a) x_i outside N_γ polytope

(b) x_i inside N_γ polytope

Figure 4.3: 3-D convex hull novelty test: (a) x_i (orange rectangle) is outside γ neighbourhood (green dotted segments are edges from γ to its neighbours) because all δ_j vectors (red arrows) have the same orientation w.r.t. Ψ (orange arrow); (b) x_i is inside the convex hull because at least δ_1 (light blue arrow) have a different orientation w.r.t. Ψ and the other δ_j.

Neuron creation

When x_i passes the w_1 novelty test, a novel unit is added to the network, see Fig. 4.4. The weight vector, w_{x_i}, is set equal to the input data. The first winner and w_{x_i} are linked by a bridge, $w_1 \to w_{x_i}$, and their *activation flags* are set to false. Finally, when a novel neuron is created, its local novelty threshold, T_{x_i}, cannot be computed with (4.1). Indeed, the only neuron connected to w_{x_i} is, by construction, w_1, but $w_1 \notin N_{w_{x_i}}$ because they are connected with a bridge and not with an edge; as a consequence, $N_{w_{x_i}}$ is empty and, according to (4.1), T_{x_i} assumes an indeterminate form. Because of the use of CHL, each neuron needs to have a specific and determined reception field to be able to fair compete with the remaining units; therefore, when a novel neuron is added to the network, the radius of its influence region, i.e. T_{x_i}, is set equal to d_1 (i.e. the bridge is treated as an edge). Then, the training procedure restarts with a new sample from the input distribution.

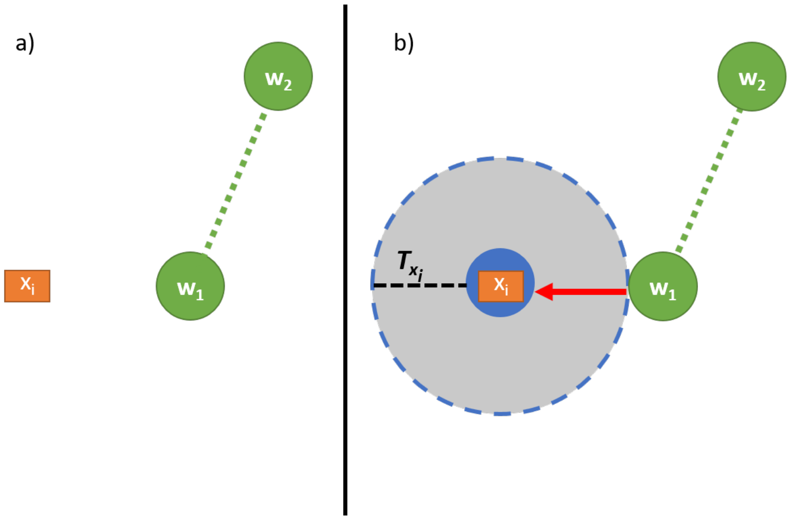


Figure 4.4: Neuron creation: a) before, b) after. The new sample x_i is represented with a rectangle and neurons with circles (existing ones are in green, the new one is in blue). Links are represented with dotted segments (existing edges) or solid arrow (new bridge). The shaded region represents the hypersphere whose radius is equal to T_{x_i} (dashed black segment).

Neuron linking and weight adaptation

The new input x_i has failed the first winner novelty test. If w_1 and w_2 are not connected by a bridge or if w_1 is the bridge tail, i.e. $w_1 \rightarrow w_2$, the first and second winner are linked by an edge, whose age is set to zero; in the former case, if the edge already exists, only its age is updated to zero, while in the latter case the bridge is substituted by an zero-age edge.

The same GCCA ageing procedure is used: the age of all the other links (both edges and departing bridges) of w_1 is incremented by one; if a link age is greater than the age_{max} scalar hyperparameter, it is pruned. If a neuron becomes lonely (i.e. no links), it is deleted.

If the input sample is within the first winner convex hull, i.e. $x_i \subset N_{w_1}$, the neighbours weight vectors are adapted according to SCL:

$$\Delta w_1 = \alpha_1 (w_1 - x_i) \tag{4.5a}$$

$$\Delta w_j = \alpha_n (w_j - x_0) \tag{4.5b}$$

$$\alpha_1 = \frac{\alpha}{n_{w_1}} \tag{4.5c}$$

$$\alpha_n = \alpha * \exp(-\frac{(w_j - x_i)^2}{2\sigma^2}) \tag{4.5d}$$

where $w_j \in N_{w_1}$, α and σ are the G-EXIN hyperparameters for weight adaptation (the former scales the learning rate, the latter modulates SCL) and n_{w_1} is the amount of times w_1 was the first winner since its creation. Conversely, if $x_i \notin N_{w_1}$, only (4.5a) is used (HCL).

For each unit whose weight vector has been updated, say *ϕ-neuron*, its novelty threshold is recomputed according to (4.1) and its *activation flag* is set to true. Then, for each *ϕ-neuron*, its bridges, both ingoing and outgoing, are checked and those whose end neurons have both activation flags equal to true become edges. Finally, the training procedure restarts with a new sample.

Neuron doubling

If x_i has failed the novelty test and the first and the second winner are linked by a bridge, where w_1 is the top, i.e. $w_2 \rightarrow w_1$, a seed is created by means of the neuron doubling technique. A novel neuron, $w_{1_{new}}$, is added to the network whose weight vector is computed using HCL (4.5a), where α is the G-EXIN hyperparameter for weight adaptation and N_{w_1} is the amount of times w_1 was the first winner since its creation.

The two neurons, w_1 and $w_{1_{new}}$ are linked with a zero-age edge and their novelty thresholds are set equal to their Euclidean distance. Then, the training procedure restarts with a new sample from the input distribution.

Hyperparameters analysis

G-EXIN only requires three user dependent parameters: α and σ for the weight updating and age_{max} for the link pruning. As its parent networks, G-EXIN employs both the soft and hard competitive learnings to update weights during training. On the other hand, G-EXIN makes both α_1 and α_n local and less related to user-dependent parameters due to (4.5c) and (4.5d). The former exploits the first winner inertia, i.e. the amount of CHL victories, to decrease locally the network plasticity, as in k-means [68]; in this sense, α is used only to slower this process by a factor directly proportioned to its magnitude; however, experiments suggest a suitable value is $\alpha = 1$. In alternative, it can be automatically tuned, i.e. increased or decreased, depending of the level of non-stationarity of the input data stream, e.g. analysing the amount of bridges and their lengths.

In order to perform a weight update strictly related to the input manifold topology, the parameter α_n used in (4.5d) is modelled as a multi-variate Gaussian centred in the input data with variance equal to the hyperparameter σ. In this sense, remote first winner neighbours w.r.t. the input sample are not influenced by its presentation because they will fall in the Gaussian tails, i.e. their $\Delta w_j \simeq 0$. The Gaussian is moduled by α, and the considerations done in the previous case can be repeated.

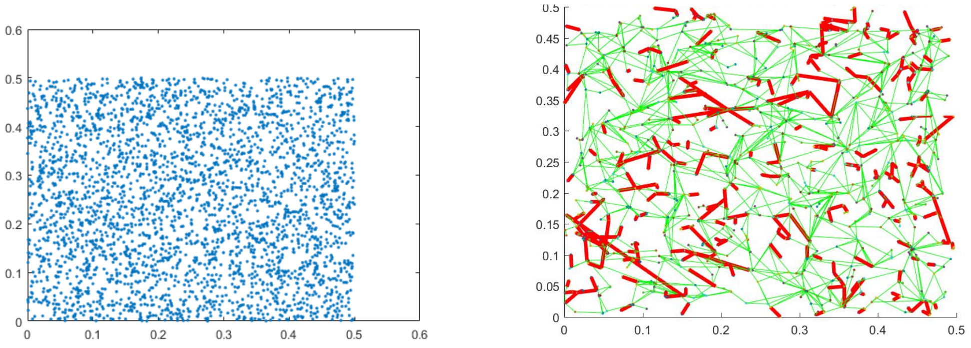
For age_{max} the analysis made for GCCA holds, i.e. a technique for automatizing it can be employed for correlating its value to the behaviour, i.e. the non-stationarity, of the input flow. For instance, it can be monitored the rate of bridges over time: if it increases, there is more novelty in the input samples and the age_{max} value could be decreased to better approximate the input distribution.

Analysis of bridges

Bridges are the directional connections (towards the change) used to detect and follow non-stationarity in the input distribution. They derive from GCCA and, as detailed in Sec. 3.3.1 and Fig. 3.12, their length and density over time provide additional awareness of the time-varying distribution. For example, as in the case of the three-times jumping square example, where the input distribution jumps abruptly from one region of the input space to another, there will be a limited quantity of long bridges.

G-EXIN employs an additional method for converting bridges into edges based on the neuron *activation flags*. In this sense, the aim is to better discriminate between abruptly changes in the input distribution and input embedded into the manifold. The former scenario has already been studied by means of the cited example. To deepen the latter case, the moving square manifold shown in Fig. 4.5a has been randomly fed to G-EXIN. First, the system learns the data in a stationary condition (see 4.5b): there are a lot of short bridges, which means the manifold is stationary, the amount is related to input randomness.

Once all the samples have been fed to G-EXIN, the distribution support starts to move vertically, i.e. the data stream becomes non-stationary. The experiment has been repeated several times at different displacement velocity: fast, medium, slow. The idea is to analyse the bridge characteristics, e.g. the density, at different displacement velocity of the distribution. In this sense, it is expected to have fewer and longer bridges as the velocity increases. Fig. 4.6 yields the G-EXIN quantization in the three cases; of course, the G-EXIN hyperparameters are equal to the stationary case. The amount of bridges (red segments) lowers as the velocity increases; on the contrary, as expected, their length exhibits an opposite trend.



(a) Initial dataset

(b) G-EXIN quantization

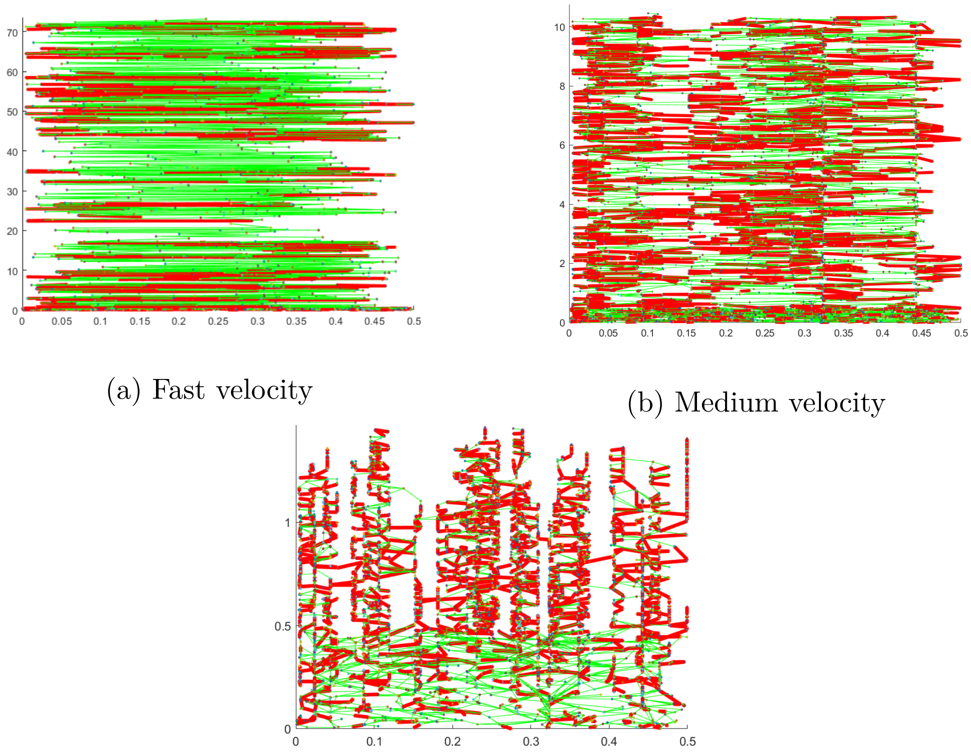
Figure 4.5: Vertically moving square, stationary scenario: (a) initial dataset, samples in blue; (b) G-EXIN quantization ($\alpha = 1$, $\sigma = 0.07$, $age_{max} = 5$): neurons (points), edges (green segments) and bridges (red segments).

The faster the displacement, the higher the change, which means the network can recognize non-stationarity easier. Fig. 4.6a exhibits very few long bridges w.r.t. the remaining cases. It can be argued that, when the input distribution is in the stationary condition, it is harder for the network to discriminate novelties in terms of non-stationarity and input coming from a different part of the input manifold. Remember that G-EXIN is an online technique, which means it sees each sample only once; therefore, if data come randomly from different sectors of the input manifold, it is almost impossible that bridges are converted in edges, i.e. it is harder to understand the kind of non-stationarity at hand. A lot of short bridges are visible in the slow velocity case as shown in Fig. 4.6c. Data do not change significatively w.r.t. the stationary case; neurons are very close, i.e. their domains are very small. Small influence regions mean novelty tests are often succeeded from input samples, i.e. a lot of neurons are created.

The previous considerations are confirmed by Fig. 4.7, which displays the histograms of the amount of bridges (top), edges (middle) and neurons (bottom) at the end of each training. All the subfigures follows the same naming convention for the bars: $\_s$ for the slow velocity case, $\_m$ for the medium and $\_f$ for the fast one. For sake of comparison, results are grouped w.r.t. the values of σ and age_{max}.

Fig. 4.7a validates what stated before, the number of bridges lowers when the level of non-stationarity increases regardless of the hyperparameters.

A more interesting consideration can be done about the edge diagram (see Fig. 4.7b). Conversely to the other histograms, in the medium velocity case, the number of edges increases. In this sense, it can be argued that neurons win, i.e. CHL linking is done, in both regions and fewer neurons are created w.r.t. the previous case. Moreover, the number of edges is much greater than the amount of bridges for all the tested configurations. This behaviour is due to both the neuron creation



(a) Fast velocity

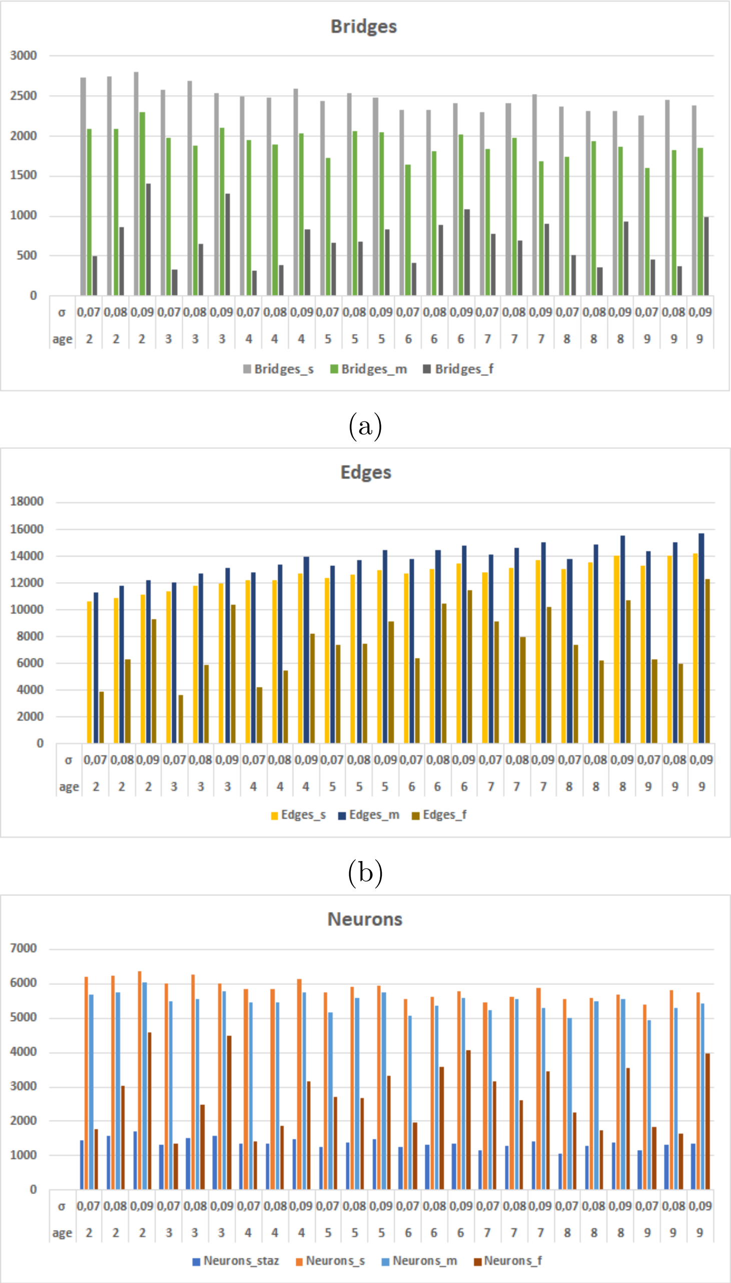
(b) Medium velocity

(c) Slow velocity

Figure 4.6: Vertically moving square, non-stationary scenario. G-EXIN quantization ($\alpha = 1$, $\sigma = 0.07$, $age_{max} = 5$): neurons (points), edges (green segments) and bridges (red segments).

and doubling techniques; indeed, when a new input space region is explored, the same first and second winners will be chosen, i.e. a lot of edges are created between neurons. At the same time, the links age quickly; for low values of age_{max} they are often pruned as shown from the corresponding lower columns in Fig. 4.7b.

Fig. 4.7c shows the number of neurons at the end of the training. The legenda is as before; for easy of comparison, an additional column, *neurons_staz*, has been added, which reports the number of neurons at the end of the stationary quantization. The figure strengthens the previous analysis. A lot of neurons are created in slow and medium cases. In the fast scenario, especially in case of low σ, the amount of neurons drops. Remember that, from (4.1), the novelty threshold is computed as the distance from a neuron farthest neighbour; if σ is low, in (4.5d) the Gaussian narrows and less neighbour weight vectors are updated, i.e. the average T_{w_i} tend to increase. Therefore, in a scenario where the input support is moving quickly, the new neurons will have big receptive fields and will result as first winners for a lot of subsequent samples, which means much less units will be created.



(a)

(b)

(c)

Figure 4.7: Vertically moving square, non-stationary scenario: network performance analysis. Number of bridges (top), edges (medium) and neurons (bottom) at different σ (0.07 - 0.09) and age_{max} (2 - 9) for different velocity: slow (*_s*), medium, (*_m*), fast (*_f*). *Neurons_staz* indicates the amount of neurons before displacement starts.

4.1.2 G-EXIN experiments

G-EXIN derives from the first layer of GCCA equipped with three new features: the new anisotropic threshold, the activation flags to convert more bridges into edges and a localized SCL. To test these improvements in mapping the input topological structure even in case of a non-stationarity environment, while, at the same time, maintaining the previously learned information (lifelong learning), three experiments have been conducted to compare G-EXIN and ESOINN performances. The first synthetic dataset is the same 3-times jumping dataset proposed in Sec. 3.3.2 for GCCA testing; in this sense, it can be used as a benchmark with the GCCA quantization. Moreover, it has also been fed to the ESOINN neural network for further comparison on non-stationarity handling. The second simulation analyses the network response to non-stationarity when the input domain changes smoothly instead of abruptly as in the previous case. To this purpose, the 2-D moving square manifold is used. Finally, the two architectures are compared on the real-time prognostic dataset (see 3.2.2). Before presenting the results, a brief description of ESOINN follows.

ESOINN

The Enhanced Self-Organizing Incremental Neural Network (ESOINN) has been designed to deal with online unsupervised learning tasks. It is a single-layer network, which performs non-stationary input clustering into a suitable number of classes, builds a graph of neurons to map input topological structure, and also separates clusters with high-density overlap. It requires only four hyperparameters.

Each node w_i is equipped with a local similarity threshold, T_{w_i}, which is continuously updated to adapt to the input data distribution samples fed during training iterations. If the node is linked to some other units, i.e. it has neighbours, the threshold is computed as:

$$T_{w_i} = \max_{j \in N_{w_i}} \|w_i - w_j\| \tag{4.6}$$

where N_{w_i} is the node neighbour set and w are the weight vectors.

On the contrary if N_{w_i} is empty, i.e. the node does not have any neighbours, the similarity threshold is defined as:

$$T_{w_i} = \min_{j \in N} \|w_i - w_j\| \tag{4.7}$$

where N is the set of all nodes without the first winner.

To start, the graph is made of two nodes which represent two randomly chosen inputs. Each time a new sample, ξ, is fed, ESOINN applies the CHL for determining the first (w_1) and second (w_2) winners w.r.t. the input vector; then, it applies a similarity criterion based on a local threshold to determine if ξ belongs to the same cluster of w_1 or w_2: if the distance between ξ and the first or the second winners is greater than, at least, one of their similarity thresholds, a new node is added to the graph, whose weight is equal to ξ. It must be underlined that using also the second winner threshold to determine ξ level of novelty does not always provide the best result: a sample very close to the first winner will not be assigned to it, if it is farer than T_{w_2} to from the second winner.

When a new node is inserted, the network training ends and the algorithm moves to the next input sample; otherwise, it performs few more steps. First, the age of all winner edges are increased. Then, if w_1 and w_2 are already linked, the age link is reset to zero; otherwise, the two nodes are linked according to CHR. They may not be linked in case w_1 and w_2 belong to different not overlapped classes; conversely, if the two classes have a high overlap, nodes are connected and their classes are combined.

The class overlap is computed using the idea of node density; here, it is defined considering the number of times each node wins, M_i, and the mean distance between the neurons and its neighbours. In order to understand whether two subclasses of neurons overlap or not, each λ_E iterations (a user-dependent parameter), ESOINN finds all apexes, which are the nodes with local-maximum density. Each apex is assigned one class label and the same is also given to all its neighbor nodes. After having decided whether to connect first and second winner, the algorithm updates winner local accumulated number of signals.

SCL is performed to update the weight vectors of the first winner w_1 and its neighbours $w_j \in N_{w_1}$:

$$\Delta w_1 = \varepsilon_1(\xi - w_1) \tag{4.8a}$$

$$\Delta w_j = \frac{\varepsilon_j}{100}(\xi - w_j) \tag{4.8b}$$

where

$$\varepsilon_i = \frac{1}{M_i} \tag{4.8c}$$

and M_i is the amount of times neuron w_i was selected as first winner, i.e. the cardinality of its associated sample set. The learning rate (4.8c) decreases according to the local accumulated number of signals of the neuron. It may be observed that an SCL where (4.8b) learning rate is reduced of a factor of one hundred, almost results in an HCL. Successively, all edges, whose age is greater than the age_{max} hyperparameter, are pruned.

Every λ_E iterations, a pruning of noise and overlapped nodes is performed. The former are defined as nodes with no neighbours, while the latter are nodes with only one or two neighbours whose densities satisfy the following condition:

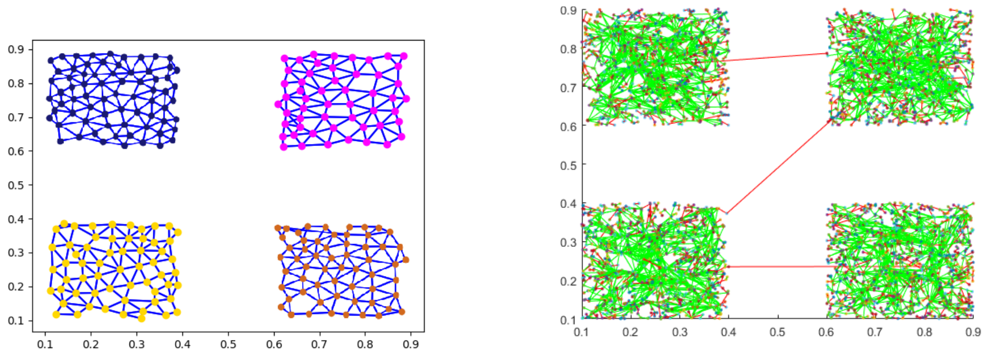
$$h(n) < c_{2,1} \frac{\sum_{i=0, i \neq n}^{N} h(i)}{N} \tag{4.9}$$

where $h(n)$ is the density of the n-node; c_2, if the node has only one neighbour, or c_1, when it has two neighbours, are user-dependent parameters and the last term is the mean densities of all nodes.

When the learning process ends, ESOINN classifies all nodes according to their labels and reports the estimated number of classes found; otherwise, it processes the next input signal.

Synthetic experiments

The first synthetic experiment deals with the 3-times jumping dataset (from NW to NE, then from NE to SW and, finally, from SW to SE) proposed in Sec. 3.3.2 for GCCA. The aim is testing G-EXIN and ESOINN on abrupt changes in the input distribution. Fig. 4.8 shows the results of ESOINN ($age_{max} = 100$, $\lambda_E = 200$, $c_1 = 0.001$, $c_2 = 1$) and G-EXIN ($age_{max} = 18$, $\alpha = 1$, $\sigma = 0.06$) on such dataset. In Fig. 4.8a, ESOINN neuron weights and edges are shown; the algorithm identifies four separated classes (different neuron colours) in the input distribution. Conversely, G-EXIN considers the data as belonging to a single cluster, as shown in Fig. 4.8b. It must also be underlined how well G-EXIN performs in the border quantization thanks to its anisotropic threshold. This is not achievable with an isotropic technique, such as the one employed by ESOINN or GCCA.



(a) ESOINN

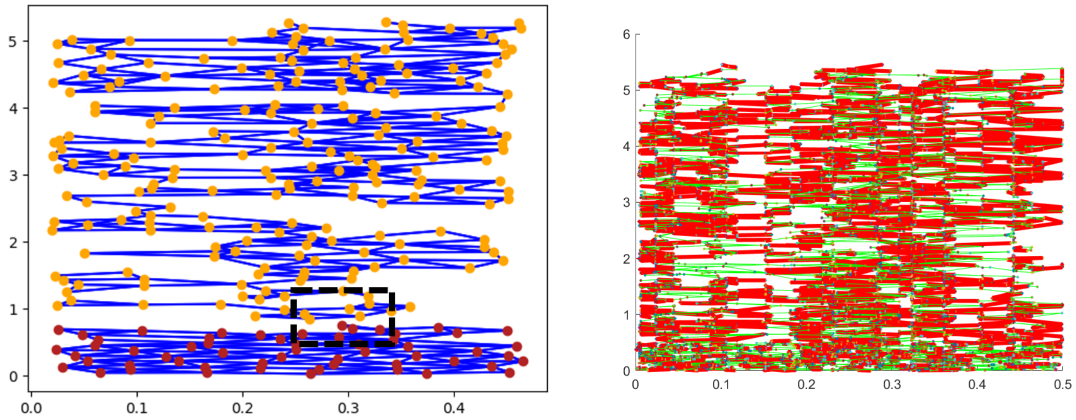
(b) G-EXIN

Figure 4.8: First simulation: 3-times jumping square. ESOINN output is on the left: neurons are circles (the color indicates the class), links are blue segments. G-EXIN quantization is on the right: points are neurons, green segments are edges and red segments are the bridges.

The two architectures learn the input manifold topology while keeping the whole history; G-EXIN recognizes the abrupt changes in the distribution through single, long, bridges whose length is related to the level of non-stationarity. It could be argued that the four classes yielded by ESOINN can be considered as one class in four different times. However, this consideration holds only in case of a discontinuity in the input data stream. If the change is smooth, ESOINN will interpret it as a single cluster, without any hints about the non-stationarity.

The second simulation employs the 2-D moving square distribution for deepening the architecture handling of smooth non-stationarity. As explained before, initially both ESOINN ($age_{max} = 300$, $\lambda_E = 200$, $c_1 = 0.001$, $c_2 = 1$) and G-EXIN ($age_{max} = 2$, $\alpha = 1$, $\sigma = 0.08$) are trained on the whole input distribution ($\sim$ five thousand samples); data are fed randomly and only once to each network; then, the distribution starts to move vertically for a number of iterations equal to his size.

Fig. 4.9a yields the ESOINN output. The input evolution is, on average, well tracked. As in the previous simulation, it misclassifies the non-stationary input as made of two different separated classes (red and yellow) because of a missing link (as highlighted by the black dotted rectangle). Fig. 4.9b shows G-EXIN follows and interprets distribution evolution after the displacement onset. By using bridges, G-EXIN correctly considers the data stream as made of a single cluster, which further proves the importance of having a different, anisotropic link for handling non-stationarity.



(a) ESOINN

(b) G-EXIN

Figure 4.9: Second simulation: vertical moving distribution. ESOINN output is on the left: neurons are circles (the colour indicates the class), links are blue segments. G-EXIN quantization is on the right: points are neurons, green segments are edges and red segments are the bridges.

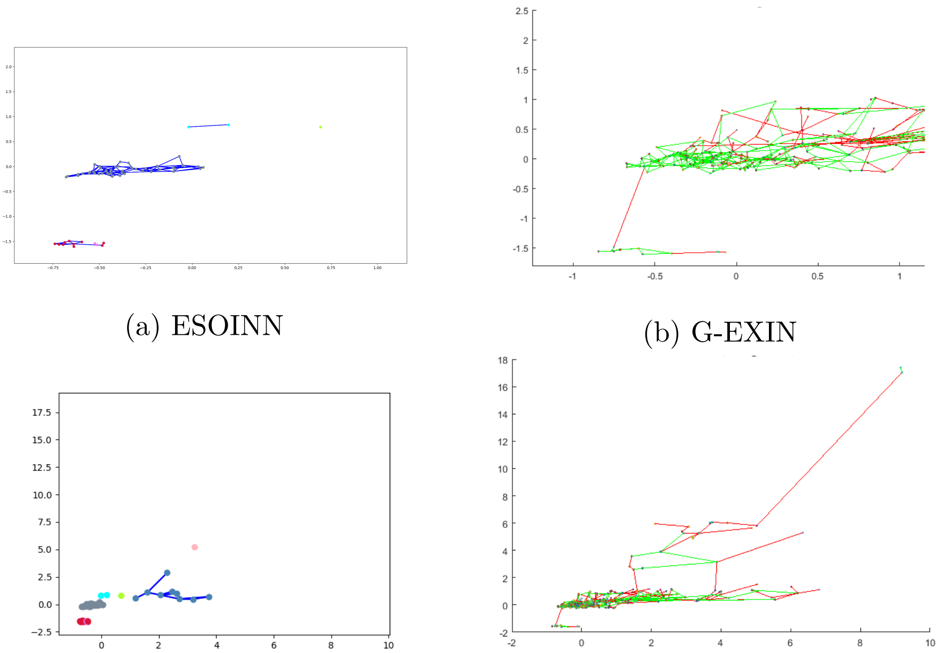
Prognostic application

As a latter experiment, the two architectures have been compared on the real-time prognostic dataset (see Sec. 3.2.2) to test how they perform in tracking the machine evolution up to the double fault.

Fig. 4.10 shows the comparison between ESOINN ($age_{max} = 100$, $\lambda_E = 100$, $c_1 = 0.001$, $c_2 = 1$) and G-EXIN ($age_{max} = 15$, $\alpha = 0.2$, $\sigma = 0.04$). Figs. 4.10a and

4.10b show ESOINN and G-EXIN quantizations during the initial transient, the healthy state and initial deterioration. The former correctly classifies the first two phases as different classes but it cannot represent the transition from the transient phase to the healthy state because of the lack of links between clusters. For the same reasons, it cannot depict the pre-fault condition; indeed, it just finds three nodes, which look like noise. Fig. 4.10b shows G-EXIN learns the three phases correctly, i.e. the quantization is accurate; moreover, from the appearance of long vertical bridges, it is possible to infer the onset, first of the healthy state, which is then represented by green edges, and then of the pre-fault.

Figs. 4.10c and 4.10d show the whole bearing evolution over time. ESOINN (Fig. 4.10c) groups data into separate classes w.r.t. the machine phases; the most severe fault is represented only by one neuron, which is absolutely insufficient to represent the faulty condition. On the contrary, G-EXIN (Fig. 4.10d) is able to discriminate the four phases and to relate them by means of bridges, whose length is proportional to the severity of the faults.



(a) ESOINN

(b) G-EXIN

(c) ESOINN

(d) G-EXIN

Figure 4.10: Third experiment: prognostic dataset. ESOINN (left): neurons are circles (the colours indicate the class), links are blue segments. G-EXIN (right): points are neurons, green segments are edges and red segments are the bridges. Top figures show the comparison at the fault onset, while the bottom ones yield the full quantization.

Chapter 5

Hierarchical Clustering

A special kind of unsupervised learning is the hierarchical clustering. Such an approach assumes the input information is stratified, i.e. several layers of data interpretation are possible; in this sense, the network builds a hierarchy (a tree), where the root corresponds to a coarse resolution, while each subsequent layer refines its ancestor [74]. In data mining, for example, this approach can extract a deeper information than plain clustering. Depending on the tree building strategy, bottom-up or top-down, the hierarchical techniques can be grouped into two categories: Hierarchical Agglomerative Clustering (HAC) and Hierarchical Divisive Clustering (HDC). The former starts with one cluster per each sample and then pairs of nodes are connected up to the tree root, which contains all data. Conversely, the latter approach follows an opposite strategy: it starts from a root node with all data and then creates the tree by recursively splitting clusters until all nodes are singletons.

The divisive approach gathers better results in representing data because it starts with a single cluster with all samples. Conversely, the HAC approach is, in this sense, more arbitrary in the initial steps, thus affecting the final tree quality. Also, it is unmanageable on big data. However, the HDC splitting technique is an open problem. A promising strategy employs neural network algorithms; in this sense, they can be grouped according to the training strategy and the *basic neural unit* (BNU), i.e. the neural architecture employed for each node. As a first taxonomy, two main categories exist: synchronous training (ST), where the training is done on the whole tree, and asynchronous training (AT), where it is carried out node by node. An example of ST is the Dynamic Neural Tree Network (DNTN, [75]), which adapts an evolving hierarchy to samples: all growing nodes are fed by the same input and are trained simultaneously. It employs a tolerance for estimating the new neurons and a threshold for the child growth. It cannot correctly handle the outliers. The Competitive Evolutionary Neural Tree (CENT, [76]), derived from DNTN, is presented as a hyperparameter-free network. Actually, there are internal parameters tuned w.r.t. samples, but are empirical and not justified. It is based on the unit activity, which is decreased over time to address a wrong initialization, and tackles outlier detection. Another ST technique is TreeSOM [77, 78]. The tree sensitivity to the SOM topology and the initialization are faced employing the consensus tree, which is a fictitious structure that averages the trees yielded from different initial conditions. The best tree is determined as the most similar to the consensus one.

Asynchronous training is the most common strategy used by neural techniques, which can be grouped w.r.t. the node clustering method. The k-means technique is employed in [79], which splits samples (HDC) in a predefined amount of clusters, but applies an extra HAC for refinement. The HCAKC algorithm [80] employs an improved Silhouette to determine the value of k. In [81] and [82], a preprocessing stage derived from PCA and divide-and-conquer, respectively, is performed to handle high-dimensional input spaces. An elementary SOM made of three linked neurons yields the basic structure (triangle) for the Growing Hierarchical Tree SOM (GHTSOM) [83]. Two types of links are used: the training connections to build the neural triangles and the class links for clustering each tree layer. The choice of adopting triangles as basic modules derives from the Delaunay triangulation technique, which is proven to yield the best possible quantization; unfortunately, it is not demonstrated that GHTSOM will induce it. On the contrary, employing triangles reduces significantly the performance. When Growing Cell Structure (GCS, [84]) is used as the basic neural unit, its hierarchical version is HiGCS [85]. Growing Hierarchical SOM (GH-SOM, [86]) exploits a Growing Grid (GG, [87]), with decreasing learning rate and neighborhood range. Both the vertical and horizontal growths are ruled by the average quantization error, by means of two parameters whose tuning is not straightforward, as explained in [88], where the Growing Hierarchical Neural Gas (GHNG) is presented. It is built on the Growing Neural Gas (GNG, [89]) and achieves a better performance than GH-SOM. A promising technique is the Dynamically Growing Self Organizing Tree (DGSOT, [90]), which is the enhanced dynamic variant of the Self Organizing Tree Algorithm (SOTA, [91]), which creates a binary hierarchy.

5.1 The GH-EXIN neural network

The GH-EXIN neural network [92, 93] is a variant of G-EXIN for hierarchical clustering; as a consequence, it is self-organizing (data driven), i.e. the final tree is automatically determined. The number of neurons and levels are not specified in advance; in this sense, it is an incremental technique, which also employs pruning to remove noise. The output tree is not balanced, due to its dependence on samples.

Because it is an HDC method, the algorithm starts with a single root node and then, using vertical and horizontal growths, subsequent splits are performed. The former implies the conversion of a leaf into a node, called *father neuron*, whose sons represent its leaves. Then, for each father neuron, an additional neural network (i.e. a basic neural unit) is trained on its related sample set, i.e. the inputs associated to the father unit. The obtained neurons are its sons and induce a subsplitting (horizontal growth) of its Voronoi set. For each leaf, the algorithm is repeated.

Vertical growth

Vertical growth (VG) is the step in which a leaf is converted into a node and an additional deeper level is added to the tree. It begins with a *seed*, i.e. a pair of neurons, which is the initial structure of a new BNU. VG is performed until the required resolution is reached. To this purpose, it is tested if the BNU quantization error is under a user-defined value. GH-EXIN tests both sample heterogeneity by means of a task-dependent measure, called H_{max}, and, at the same time, the leaf cluster size (min_{card}), i.e. the amount of data of its Voronoi.

The H index is related to the quantization error and estimates the clustering quality. Different implementations are possible, which depend on the application at hand. GH-EXIN employs the same H_{cc} as [94]. Its description is given in Sec. 5.2.6.

Horizontal growth

Horizontal growth (HG) is the process of adding additional units (*siblings*) to the initial seed by training the corresponding BNU. It is employed for expanding a tree layer and create more advanced hierarchical architectures rather than binary.

5.1.1 sG-EXIN

GH-EXIN employs sG-EXIN as BNU for clustering the input data. The sG-EXIN is the stationary version (no bridges) of G-EXIN; it does not have any predefined topology, because it is induced in the linking phase.

Training relies on the concept of *epoch*, which is the presentation of the randomized complete input dataset. After each epoch the hyperparameter H_{perc} is used to monitor the HG. However, this has not to be confused with *batch learning*, which implies weights to be updated or created after the whole batch presentation rather than at each iteration (data presentation). Such an approach exploits the whole batch, i.e. all the available information is used for the tree construction. It must be underlined that building a hierarchical tree is basically a stationary issue; therefore, the complete dataset is needed for building an accurate representation.

Neuron creation

The sG-EXIN, and, therefore, GH-EXIN are incremental techniques. Indeed, the amount of units is driven by data and changes over time by means of neuron creation and pruning. The former is performed using the *novelty test*: if the current units are not sufficient to represent the new sample, say x_i, a novel neuron is added. It exploits the local manifold topology; in this sense, the unit influence region is used to drive automatically the quantization. To this purpose, state of the art techniques employ, in a way or another, a threshold to model the radius of an hypersphere, which represents isotropically this region; therefore, it does not take into account the topology of the input manifold. An exhaustive description can be found in [95].

GH-EXIN exploits the same novelty test of G-EXIN based on the neuron neighbourhood, whose shape is represented by the convex hull and the extent by the isotropic hypersphere. In case x_i is within the convex polytope, it is assigned to the first winner (w_1) and the weights are updated accordingly. Otherwise, the isotropic test is used to assess if the sample is actually novel w.r.t. the current graph. The neuron isotropic threshold, say T_{w_1}, is determined as in G-EXIN (4.1): when x_i is farther than T_{w_1}, an additional unit is created on the datum; otherwise, it is assigned to w_1, and its neighbours are moved to take into account x_i. The anisotropic technique is employed to better model the input manifold, which is not assured by performing only the isotropic test. Fig. 5.1 explains the idea. THe G-EXIN approach better represents the input domain borders.

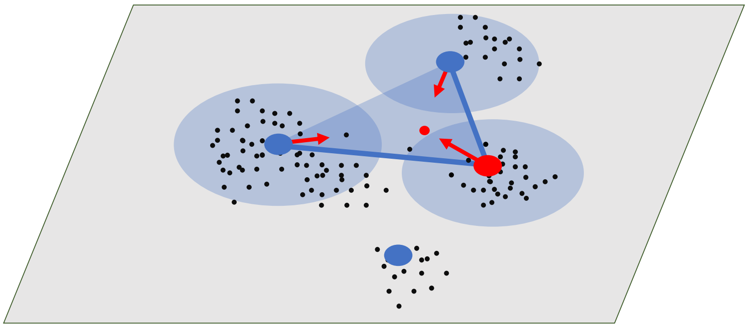


Figure 5.1: Novelty test: T_{w_1} (isotropic criterion) and convex-hull (anisotropic criterion) are employed. The input (small red circle) is external to the hypershphere centred on w_1 (big red circle), but within its convex hull (blue connected nodes are the neighbours), and it is assigned to w_1. Then, SCL moves both w_1 and its neighbours towards the input as illustrated by the arrows.

Lonely neuron

An unit without any edges is called *lonely neuron*. Fig. 5.1 presents an example at the bottom of the figure. A unit becomes lonely when all its edges are removed. New neurons are also created lonely. GH-EXIN uses the lonely neurons for leaf deletion; indeed, at the end of each epoch, they are removed from the graph and the data reallocation algorithm is performed.

Soft-competitive learning

As per its parent techniques, the SCL is employed for the weight adaptation. Only units associated to the same BNU compete to be the first winner. The first winner, w_1, and its direct topological neighbours N_{w_1} are moved towards the input sample by fractions of the vector between the weights and the data according to:

$$\Delta w = \alpha_w (w - x_i) \tag{5.1a}$$

where

$$\alpha_w = \frac{\alpha_{0i}}{t_w} \tag{5.1b}$$

and α_{0i} is an hyperparameter, higher for the winner (α_{0_1}) and smaller for the neighbours (α_{0_n}), and t_w is the amount of times neuron w has been selected as first winner in the past.

Edge creation and network topology

An edge is a connection between two neurons, which is used to define the topology (neighborhood). At this aim, as its parent networks, GH-EXIN employs CHL: when a unit is selected as first winner, it is connected with an zero-age edge to the second winner. If the edge already exists, its age is set to zero. Then, the same aging technique of G-EXIN is used: the age of all links towards its neighbours is incremented by one. If some link age exceeds the age_{max} scalar hyperparameter, it is deleted. If all edges are removed, the neuron becomes lonely.

5.1.2 Data reallocation

When a neuron is removed, its data become orphans and may be reassigned to other units. This approach is a novelty of GH-EXIN and it is derived from DGSOT. The aim is correcting clustering errors occurred in the previous levels.

In GH-EXIN, at the end of each epoch, all orphans are considered as potential outliers. For each of these, say x_p, a new winner w_r is sought among all leaves. If w_r belongs to the same neural unit (NU_{w_r}) of the pruned neuron, but x_p lies outside its hypersphere, it is definitely labelled as outlier and is not reallocated (see Fig. 5.2c). Conversely, if x_p is inside the hypersphere (or the convex-hull) of another neuron $w_r \in NU_{w_r}$ (see Fig. 5.2a) or if $w_r \notin NU_{w_r}$ (see Fig. 5.2b), then x_p is reallocated to w_r. This outlier identification can be exploited in many fields.

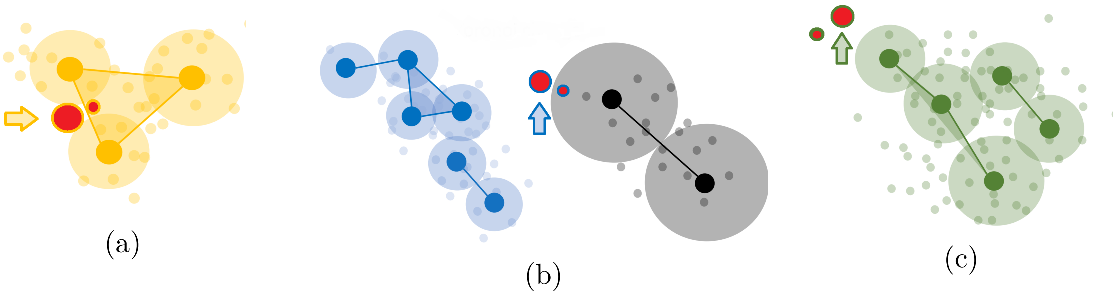


Figure 5.2: GH-EXIN data reallocation. Points are samples while big circles are the units: big red circles (indicated by arrows) are the pruned units, while small red points are their data before their pruning. Colours indicate different neural units.

5.1.3 Connected graph test

A significative innovation of GH-EXIN is the *double vertical growth*. At the end of each horizontal growth the resulting graph of the BNU is inspected looking for connected components (CCs), i.e. connected subgraphs. If at least two CCs are found, GH-EXIN attempts to derive an abstract data representation. Therefore, each CC, which maps a cluster of samples, is assigned to a novel (abstract) unit, whose weight vector is the centroid of the corresponding Voronoi. The tree is updated by adding an intermediate layer among the leaves and the father neuron, yielding a double simultaneous vertical growth (see 5.3).

This approach derives from the exploitation of the GH-EXIN topology graph. CCs are converted into the hierarchical tree through this supplementary VG. Indeed, GH-EXIN does not only cluster the samples into nested Voronoi sets, but also uses its induced (CHL) Delaunay triangulation.

5.1.4 The GH-EXIN algorithm

The complete GH-EXIN algorithm is summarized in Fig. 5.4. For each node, an sG-EXIN neural network is trained on its related Voronoi set, i.e. the samples assigned to the father neuron. For each leaf, a vertical growth, i.e. a father splitting into a seed, is performed if its H index is greater than H_{max} and its cardinality is higher than min_{card} (both are hyperparameters). Then, the seed (i.e. two linked neurons) is expanded by means of the horizontal growth, which corresponds to the training of the related BNU for several epochs as shown in Fig. 5.5.

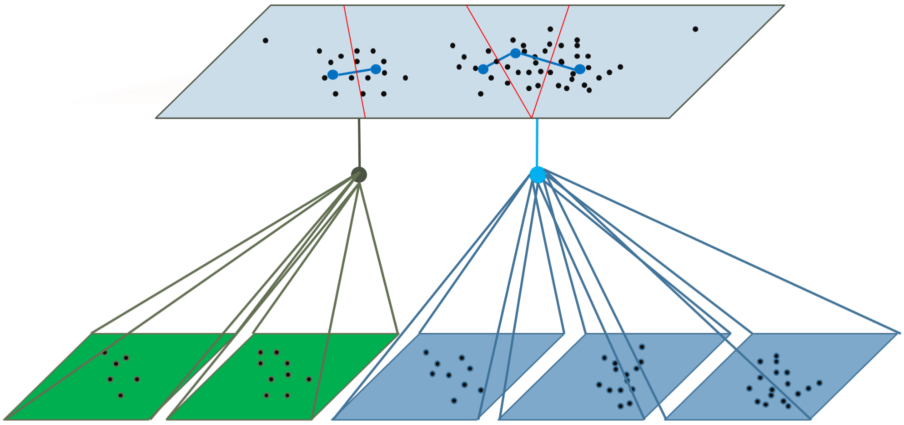


Figure 5.3: Connected Graph Test: the Voronoi regions of each neuron are defined by solid red segments; chains in the father sG-EXIN become sons.

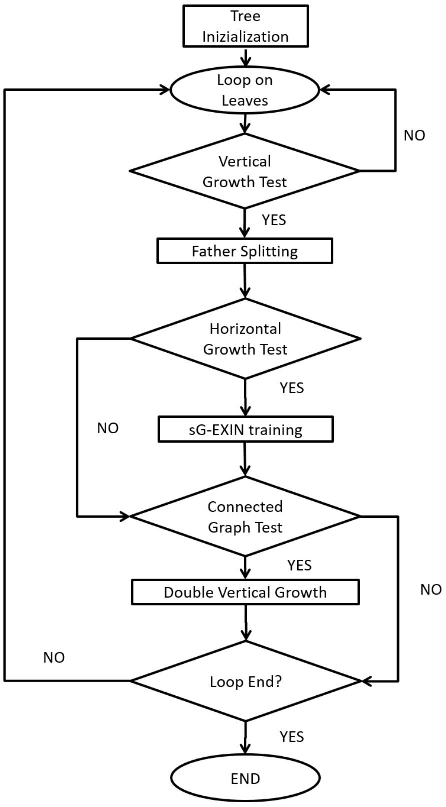


Figure 5.4: GH-EXIN flowchart

For each epoch, the basic iteration begins at the presentation of a new input, say x_i. All weight vectors are sorted w.r.t. their Euclidean distances from the input. When the sample succeeds the first winner novelty test, i.e. it is, at the same time, outside w_1 convex polytope and the hypersphere of radius T_{w_1}, a new neuron x_{new}

is added, see the left branch of Fig. 5.5. The weight vector and the threshold $T_{w_{new}}$ are given by heuristics: the new neuron has its weight x_{new} equal to x_i, and its threshold is set equal to the first winner one. No edge is created; therefore, x_{new} is a *lonely neuron*.

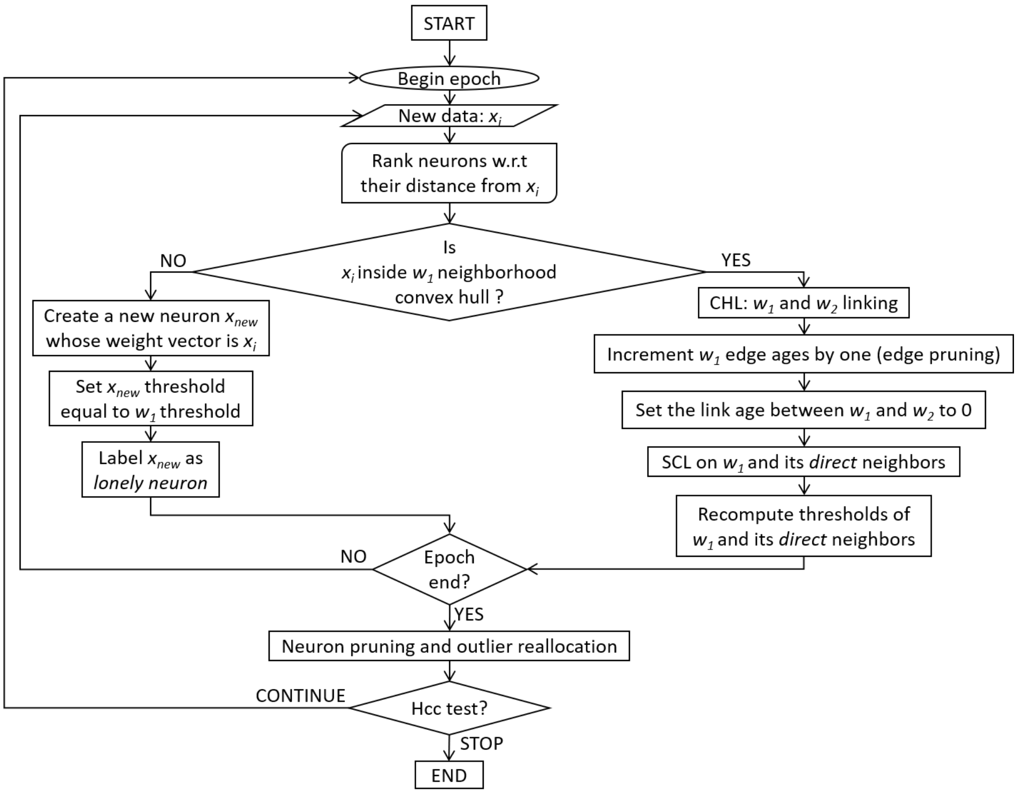

START
Begin epoch
New data: x_i
Rank neurons w.r.t their distance from x_i
Is x_i inside w_1 neighborhood convex hull ?
NO
YES
Create a new neuron x_{new} whose weight vector is x_i
Set x_{new} threshold equal to w_1 threshold
Label x_{new} as *lonely neuron*
CHL: w_1 and w_2 linking
Increment w_1 edge ages by one (edge pruning)
Set the link age between w_1 and w_2 to 0
SCL on w_1 and its *direct* neighbors
Recompute thresholds of w_1 and its *direct* neighbors
NO
Epoch end?
YES
Neuron pruning and outlier reallocation
CONTINUE
Hcc test?
STOP
END

Figure 5.5: GH-EXIN horizontal growth: sG-EXIN training flowchart

Otherwise, when x_i fails w_1 novelty test (see the right branch of Fig. 5.5), the first and second winners are linked by an edge (CHL), if it does not exist yet, whose age is set to zero. Then, the ageing and pruning procedures are performed by means of the hyperparameter age_{max}. The SCL weight adaptation (5.1a) is performed on w_1 and its neighbours N_{w_1} w.r.t. the α_{0_1} and α_{0_n} hyperparameters (5.1b), and their novelty thresholds are recomputed, because their reference vectors have changed. The sG-EXIN training algorithm is iterated for all the samples assigned to the father neuron, i.e. an epoch. Then, the data reallocation technique is exploited for checking if orphan data can be reassigned to other units.

The horizontal growth is performed until the H average value, computed at the end of a training epoch, falls below a percentage (H_{perc}, an hyperparameter) of the H value of the father neuron. Then, the *connected graph test* is performed.

The main algorithm (see Fig. 5.4) yields a VG, while the HG is induced by the units of each BNU; nevertheless, an additional VG can occur (see Sec. 5.1.3).

5.1.5 Hyperparameters analysis

The GH-EXIN employs six hyperparameters: α_{0_1}, α_{0_n} (SCL) and age_{max} (link pruning) for sG-EXIN training, H_{perc} and H_{max} for controlling the horizontal and vertical growths, respectively, and min_{card} to constraint the maximum depth. The first three hyperparameters derive from G-EXIN and the same considerations about their tuning hold. Indeed, α_{0_1} and α_{0_n} represent two constant learning rates, decreased over time as much as a neuron wins (5.1b). They control the network Stability-Plasticity: higher values yield a more flexible architecture. The scalar age_{max} rules the pruning: the lower its value, the higher the amount of deleted edges. Because of the lonely neuron handling approach, it indirectly influences the leaf cardinality. The hierarchy hyperparameters, H_{perc} and H_{max}, define the growth stop criteria; they are task-related and need to be tuned w.r.t. the application at hand. Finally, min_{card} is a design hyperparameter used to avoid too small clusters, i.e. it helps to decide in advance a preferred tree depth. In this sense, it is less significative for the user-dependent setting.

Given the last consideration about min_{card}, the meaningful hyperparameters to be tuned to train GH-EXIN are only five.

5.1.6 Analysis of the GH-EXIN algorithmic complexity

Define N as the number of samples in the complete training set (TS), d as the input space dimensionality, J as the mean number of epochs for the sG-EXIN training, and k as the mean amount of neighbours for each unit. Let be b as the mean tree branching factor. The hierarchy depth is given by $h = log_b M$, where M is the total number of leaves. For a complete tree (each leaf node assigned to one sample), M is *O(N)*, i.e. it has a linear cost.

The GH-EXIN computational cost can be estimated by analysing the training process step-by-step (see Figs. 5.4 and 5.5).

sG-EXIN iteration

At each iteration in an epoch, see Fig. 5.5, a sample x_i is fed to the BNU and the two closest units, w_1 and w_2, are determined w.r.t. their Euclidean distances from x_i. Let m_i the sG-EXIN network size, i.e. the amout of units, at the *i-th* iteration. Then, the distance estimation costs $O(m_i d)$. If an ordinary *min* technique is used, the second step (i.e. first and second minimum search) is done in $O(m_i)$. Considering an average value for all the Voronoi set cardinalities in an HG, m_i can be substituted by the mean branching factor b. Therefore, both steps have a complexity of $O(bd) + O(b) = O(b)$, because d is a constant w.r.t. the evaluation of the algorithm complexity.

Once w_1 has been chosen, the anisotropic test is done. It implies the building of the w_1 convex polytope. For generalizing the topology of all units in the BNU, the

k mean value is employed. As described in Sec. 4.1.1, for determining if $x_i \subset N_{w_1}$, the vector $\Psi = \sum \delta_j$ needs to be computed according to (4.2), which costs $O(kd)$. Then, the inner products between Ψ and all the δ_j's require, in the worst case, $O(kd)$ (all comparisons are performed). Resuming, the novelty test costs $O(kd) = O(k)$, according to the previous considerations.

Two scenarios may occur: a novel unit is added to the BNU or the first winner and its neighbours move towards the input (SCL). The former can be performed in $O(1)$ (it is a series of atomic transactions). The second scenario, instead, requires a deeper analysis: the CHL linking is $O(1)$; the ageing procedure is performed in exactly $O(k)$ operations; the SCL adaptation costs $O(kd)$ because it executes k times (the cardinality of N_{w_1}) a vector adjustment, i.e. the sum of a scaled difference vector to the weight, whose computation requires $O(d)$. Recomputing the T_{w_i} means to calculate, for each of the N_{w_1}, the distances from its neighbours; therefore, this step costs $O(k^2d)$, plus $O(k)$ for finding the farthest neighbour, i.e. $O(k^2)$, or $O(1)$ in case of neuron creation.

Resuming, a single sG-EXIN training iteration requires $O(b) + O(k^2)$ transactions. Nevertheless, it must be considered that b corresponds to the number of neurons of a BNU, and that GH-EXIN yields, by construction, a tree and not a fully connected graph. As a consequence, $b >> k$ and the total cost becomes $O(b)$.

GH-EXIN Horizontal Growth

The GH-EXIN horizontal growth is equivalent to an sG-EXIN training (see Fig. 5.5), i.e. performing an sG-EXIN iteration for each one of the father node Voronoi samples (one epoch) and then repeating it for the necessary amout of epochs. To take into account all the HGs of a single layer, the average amount of epochs, J, is introduced. In the worst scenario, all leaves are converted into father nodes and the BNUs are trained on input sets whose size sums exactly to N. Assuming both neuron pruning and outlier reallocation have a negligible cost, the whole layer HG requires $O(NJb)$ operations at worse.

GH-EXIN Cost

The worst scenario requires to perform a full HG (i.e. expansion of all the leaves) for each of the layers, that is, the height h of the tree. Remember that $h = log_bM$ and $M = O(N)$, then the overall training is $O(NJb \cdot log_bN)$. In general, both J and b are smaller than N; as a consequence, they can be ignored and the global GH-EXIN complexity is $O(Nlog_bN)$.

5.2 GH-EXIN experiments

The GH-EXIN performance has been compared with those of GHNG [88] and DGSOT [90] both on artificial datasets and on real world applications. The former comprises the same two planar datasets used in [88]: a uniform X-shape manifold and a square-shaped distribution. The aim was to inspect the partitioning characteristics and to have a simple direct visual inspection. Then, a third synthetic hierarchical database of two Gaussian mixture models has been built to test the output tree quality.

The first real world application is on video sequence hierarchical clustering. It is the same used in [88], which has been designed for testing the hierarchical clustering ability in discriminating between different faces. The latter real world application deals with two-way clustering for gene expression analysis. For each experiment, the hyperparameter value set yielding the best results for GH-EXIN, DGSOT and GHNG are shown in Tabs. 5.1, 5.2 and 5.3, respectively.

In order to compare the chosen architectures in a fair and objective way, a quantitative evaluation has been carried out by means of the peak-signal to noise ratio ($PSNR$) index [88], the Davies–Bouldin index (DB) [96] and the global Silhouette value (S) [97]. The former is defined as follows (in dBs, the higher the better):

$$PSNR = 10\log_{10}\left(\frac{MAX_l^2}{MSE}\right) \tag{5.2}$$

where MAX_l is the greatest Euclidean distance between two samples in the input dataset and MSE is the Mean Squared Error computed as the sum of the Euclidean distances between the weight of each leaf and its associated data. $PSNR$ measures only the intra-cluster compactness, while discards the inter-cluster separation; as a consequence, it is not a very accurate index of the clustering quality. Nevertheless, it has been employed because of its usage in [88].

On the contrary, the second and third indices take into account both aspects. The former is given by:

$$DB = \frac{1}{N}\sum_{i=0}^{N}\max_{j\neq i}\frac{RMSE_i + RMSE_j}{D_{i,j}} \tag{5.3}$$

where $RMSE_i$ is the Root Mean Squared Error for the *i-th* cluster, $D_{i,j}$ is the Euclidean distance between the *i-th* and *j-th* cluster centroids and N is the total amount of clusters. The lower the DB value, the better the clustering.

Finally, the Silhouette index is given by:

$$S = \frac{1}{C}\sum_{i=1}^{C}\frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{5.4}$$

where i is *i-th* input, $a(i)$ is its mean distance from the samples in the same cluster, $b(i)$ is the minimum among its mean distances from the samples in the other clusters and C is the size of the input dataset. The Silhouette index is computed for each sample in the dataset; here, the average value is considered for sake of comparison. While DB is designed for recognizing groups of clusters that are compact and well separated, S can be exploited to determine if, on average, inputs are correctly assigned to the nearest cluster.

Results have been cross-validated by running all the neural techniques 10 times for each experimental dataset and the related bar plots (see Fig. 5.15) show the average and the standard error mean (s.e.m.) for the three indices together with the number of neurons and the required training time. Before showing the experimental results, a brief description of the two benchmark networks, GHNG and DGSOT, follows.

5.2.1 GHNG

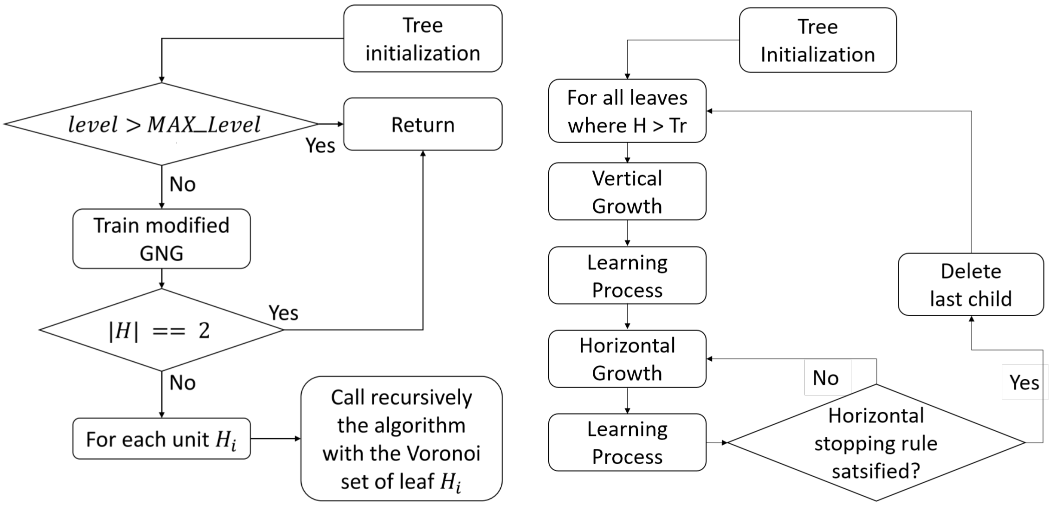
The GHNG is an AT hierarchical self-organizing neural network, based on the Growing Neural Gas (GNG, [89]). It recursively builds a tree of GNGs, where each node is trained on the Voronoi set of its father. The VG in a branch of the recursion (i.e. the tree) is performed until the maximum depth MAX_Level is reached or the deepest BNU starts the convergence phase with only two units ($|H| == 2$); in this case, the small BNU is pruned because is considered not suitable for describing any distribution. The complete algorithm is shown in Fig. 5.6a.

5.2.2 DGSOT

DGSOT, see Fig. 5.6b, derives from the Self Organizing Tree Algorithm (SOTA) [91]. Each time a SOTA would perform a VG, DGSOT does also a corresponding horizontal growth, which means adding a node to the current set of sons; in this sense, the cluster partitioning at each level is better performed. Then, a subsequent learning process is done in order to automatically estimate the right number of units for the quantization at hand. As in its ancestor, the training proceeds until the relative heterogeneity of all sons, w.r.t. the previous epoch, is lower than the hyperparameter T_R. The above steps are iterated until the Cluster Separation index decreases to a value lower than the hyperparameter T_E (horizontal stopping rule).

5.2.3 Planar synthetic experiments

The first two simulations deal with planar datasets: a uniform X-shape and a square-shape with higher density at the borders. Fig. 5.7 yields the three method hierarchical clusterings for the first dataset. At the first level (see 5.7 top row), the three techniques perform in a similar, satisfactory, way. In the second level (see 5.7



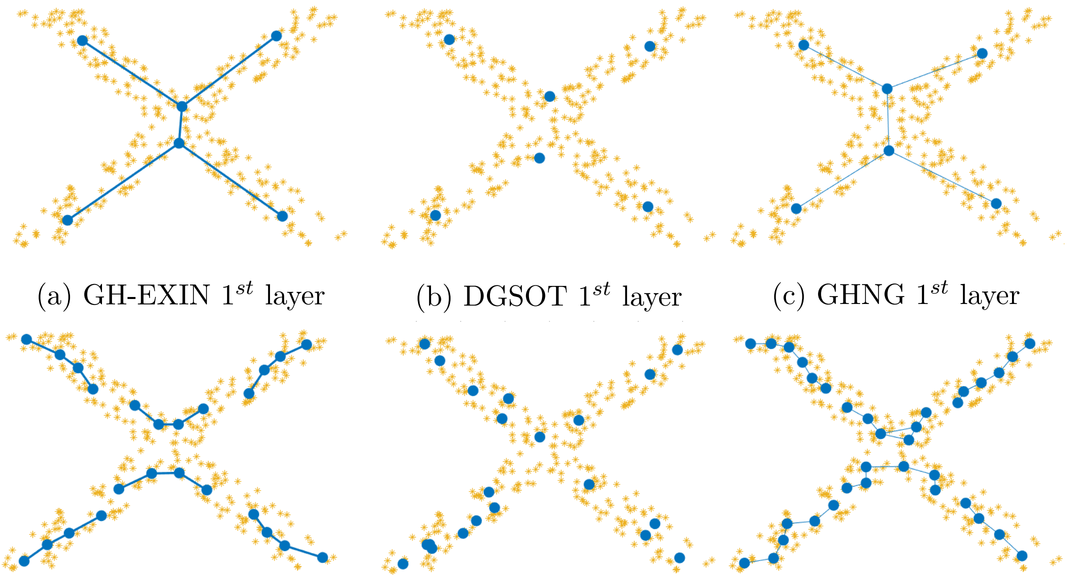
(a) GHNG (b) DGSOT

Figure 5.6: Benchmark network flowcharts

bottom row), the performances differ. DGSOT output is the less symmetric due to the lack of links; for example, the bottom-left and the top-right branches are composed of eight and three units, respectively, and in the bottom-left branch three neurons are superimposed. Similarly, GHNG induced topology is not symmetric in the number of units per branch and, therefore, their density. On the contrary, GH-EXIN employs fewer units than GHNG, which are uniformly distributed over the input manifold. The associated quantitative analysis (see. Fig. 5.15) shows that despite GH-EXIN requires slightly more time, it creates fewer neurons on average, and it is more stable than the other two on this metric. Both GH-EXIN and GHNG clusterings perform better than DGSOT for all clustering indices. DB is slightly higher for GH-EXIN than GHNG, while DGSOT shows large variance.

The second simulation is performed on the square dataset. The results are shown in Fig. 5.8 for the three architectures. In the first level (see 5.8 top row), GH-EXIN and DGSOT cover more uniformly than GHNG the input manifold. In the second level (see 5.8 top row), GH-EXIN employs fewer units than the other techniques; it better gathers the input symmetry by means of placing its units along the borders, proportionally to the sample distribution, whose central part is sparse. Conversely, both GHNG and DGSOT have several units also in the central area and do not follow the input symmetry. This visual inspection is confirmed also from the quantitative analysis; indeed, GH-EXIN is the best algorithm w.r.t. DB and S indexes, while using much less neurons. Even in this experiment, GH-EXIN training time is higher than DGSOT and, above all, GHNG. Finally, on the square distribution, all the techniques are quite stable on average.

The GH-EXIN performance over the two above simulations have been studied



(a) GH-EXIN 1^{st} layer

(b) DGSOT 1^{st} layer

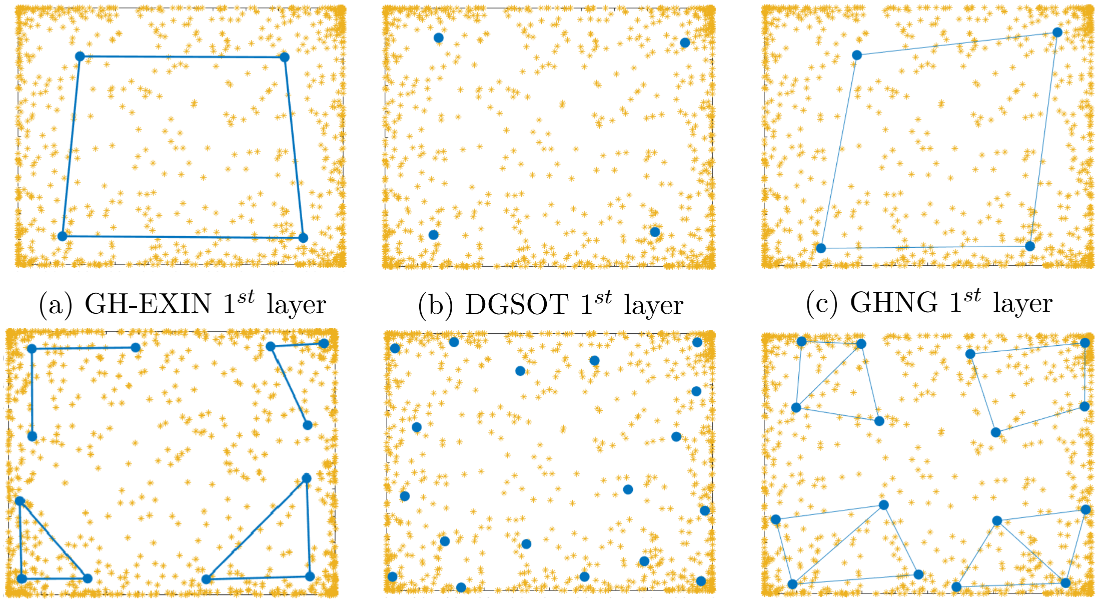
(c) GHNG 1^{st} layer

(d) GH-EXIN 2^{nd} layer

(e) DGSOT 2^{nd} layer

(f) GHNG 2^{nd} layer

Figure 5.7: First simulation: X-shape distribution. First (top row) and second (bottom row) levels of GH-EXIN (left), DGSOT (middle) and GHNG (right).



(a) GH-EXIN 1^{st} layer

(b) DGSOT 1^{st} layer

(c) GHNG 1^{st} layer

(d) GH-EXIN 2^{nd} layer

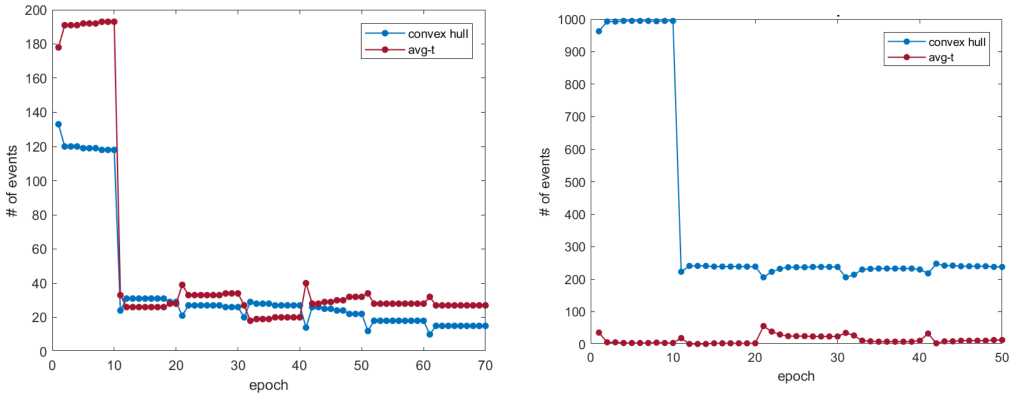
(e) DGSOT 2^{nd} layer

(f) GHNG 2^{nd} layer

Figure 5.8: Second simulation: square distribution. First (top row) and second (bottom row) levels of GH-EXIN (left), DGSOT (middle) and GHNG (right).

w.r.t. its innovative novelty test. Figs. 5.9a and 5.9b illustrate the amount of times the anisotropic test (convex-hull) has been used w.r.t. the isotropic method. As shown in Fig. 5.9b, the anisotropic test is extensively used for the square manifold due to the relevance of the borders.

Finally, in both simulations the GH-EXIN performs the best clustering, as confirmed by the visual inspection, and both the S and DB indexes. Despite it is more time-consuming, it creates fewer nodes.



(a) X-shape distribution

(b) Square distribution

Figure 5.9: Synthetic experiments: GH-EXIN novelty test analysis. Frequency of usage of the anisotropic (blue) and isotropic (red) criteria. Each neuron training lasts ten epochs; the first ten regard the first level units, while the subsequent epochs refer to the second level neurons.

5.2.4 Hierarchical synthetic experiment

The previous simulations emphasize the clustering level by level. However, GH-EXIN, DGSOT and GHNG have been designed not for plain clustering, but for the hierarchical one. Therefore, a third synthetic dataset, whose hierarchy is well-defined a priori, has been exploited as a benchmark for measuring the yielded tree quality. The chosen dataset, see Fig. 5.10, is made of two Gaussian mixture models of three and four Gaussians, respectively.

Fig. 5.11 yields the three technique outcomes: in the first level (top row), both GH-EXIN and DGSOT place units correctly w.r.t. the Gaussian means, while GHNG does not recognize the hierarchy in the input. The corresponding trees are shown in Fig. 5.12, which confirms the previous considerations: GH-EXIN and DGSOT place two neurons in the first level, which represents the two Gaussian models, and as many leaves as the dataset Gaussians in the second layer, which yields the mixtures.

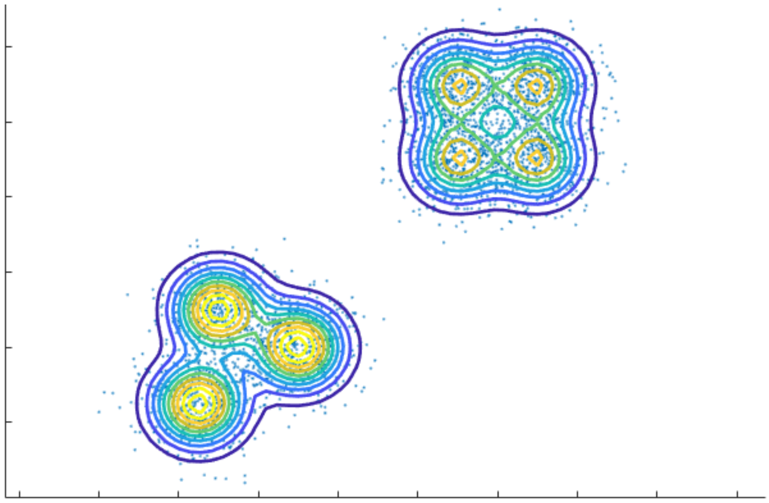
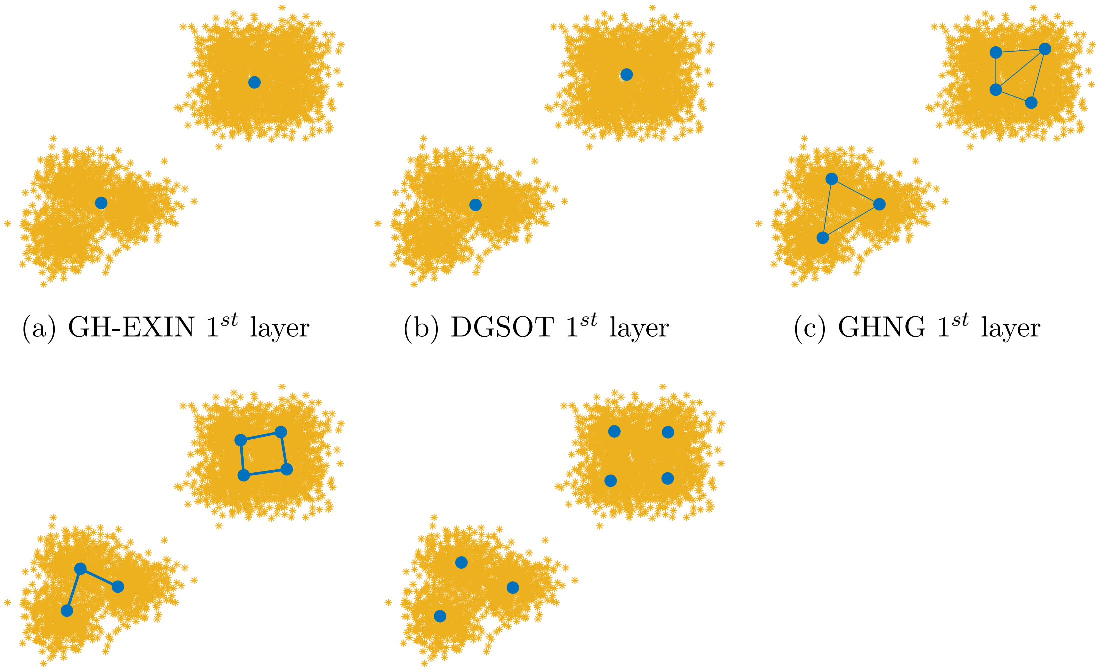


Figure 5.10: Third simulation: two Gaussian mixture models. Blue points are the data while lines represent the distribution contours.



(a) GH-EXIN 1^{st} layer

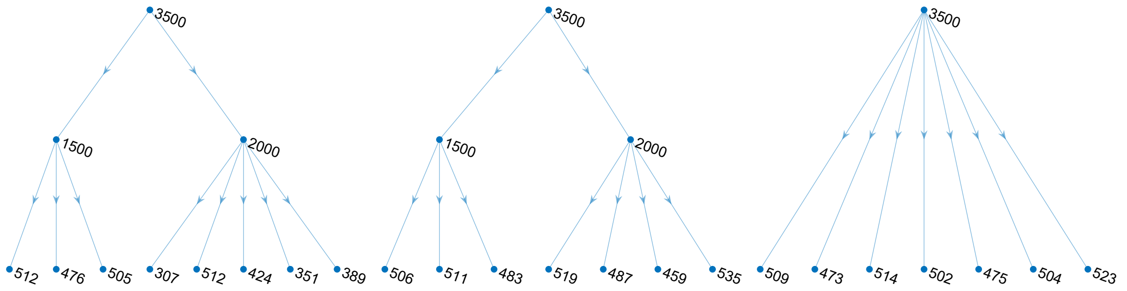
(b) DGSOT 1^{st} layer

(c) GHNG 1^{st} layer

(d) GH-EXIN 2^{nd} layer

(e) DGSOT 2^{nd} layer

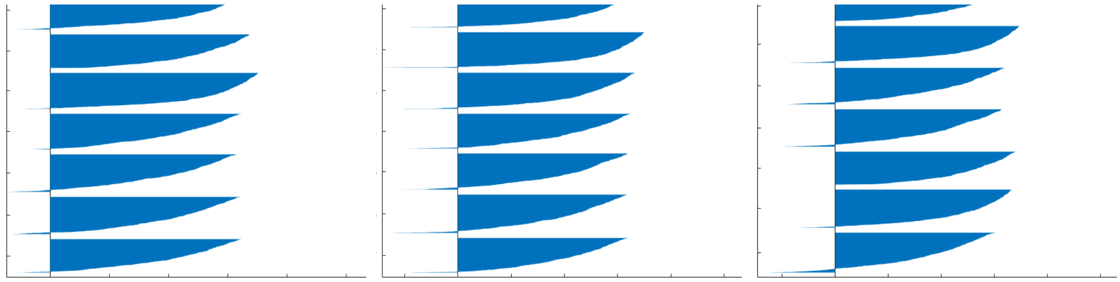
Figure 5.11: Third simulation: two Gaussian mixture models. First (top row) and second (bottom row) levels of GH-EXIN (left), DGSOT (middle) and GHNG (right). GHNG second layer is not shown because it covers all the input manifold at the first level.



(a) GH-EXIN (b) DGSOT (c) GHNG

Figure 5.12: Third simulation: two Gaussian mixture models. Final trees of GH-EXIN (left), DGSOT (middle) and GHNG (right). Nodes are labelled by the corresponding cluster cardinality.

The quality indexes, see Fig. 5.15, are computed on the complete clustering. *PSNR* and *S* are moderately better for GH-EXIN, while *DB* is the same for the three methods. This statement is confirmed by Fig. 5.13, which displays the *S* values for each neuron of the second layer: the GH-EXIN scores are mostly positive and in few cases only moderately negative, conversely to the other two algorithms. GH-EXIN, DGSOT and GHNG yield a similar amount of units, but the latter is quite faster.



(a) GH-EXIN (b) DGSOT (c) GHNG

Figure 5.13: Third simulation: two Gaussian mixture models. Silhouettes for GH-EXIN (left), DGSOT (middle) and GHNG (right).

Resuming, GHNG is a very fast technique but it is better suited for plain clustering rather than hierarchical. It opens the question if its rapidity just derives from its modified GNG module.

5.2.5 Hierarchical clustering for video sequences

The first real experiment has been designed to assess the hierarchical clustering quality. The dataset [98] is made of five video sequences, each portraying only one out of four subjects (two men, classes 2 and 4, and two women, classes 1 and 5) or one container (class 3). The aim is clustering together all the frames of the same class. There are around 1.5K samples, whose dimensionality is 25K (176×144 pixels), per three RGB channels. As an initial preprocessing, a greyscale transformation is applied on the input images and only the luminance information is kept; in this sense, hue and saturation are not considered informative. Due to the high dimensionality of input space, the PCA has been employed to project samples to an 8-D subspace by means of the eigenface technique [99] (83% of the original data variance has been preserved). The dataset has been already exploited in [88], where GHNG performance overcomes those of GNG and, especially, GHSOM and SOM.

GH-EXIN, DGSOT and GHNG have been trained on the dataset, Fig. 5.14-left displays the output trees, whose units have been labelled from a progressive unique identifier using a breadth-first search [100]. In Fig. 5.14-right, for each leaf (the values in the x-axis correspond to the tree labels) is reported the highest class efficiency, which is the greatest percentage of samples belonging to a single class in a cluster (here, the leaf Voronoi set). In this sense, it can be interpreted as an external measure of the clustering quality. The best class w.r.t. efficiency index is shown on the top of each bar. Moreover, experiments show GH-EXIN and DGSOT leaves reach a 100% purity, while a few of GHNG ones do not share this property; here, the purity index is defined as the percentage of samples in a cluster (here, the leaf Voronoi set) belonging to the most common class.

GH-EXIN units 2 and 3 have been created by the double vertical growth (red edges in Fig. 5.14a) explained in Sec. 5.1.3 and derive from the presence of two connected components in the input dataset. Indeed, the former node contains only data from classes 1, 2 and 5, while the latter one from classes 3 and 4. Therefore, GH-EXIN first layer perfectly splits samples in two subsets.

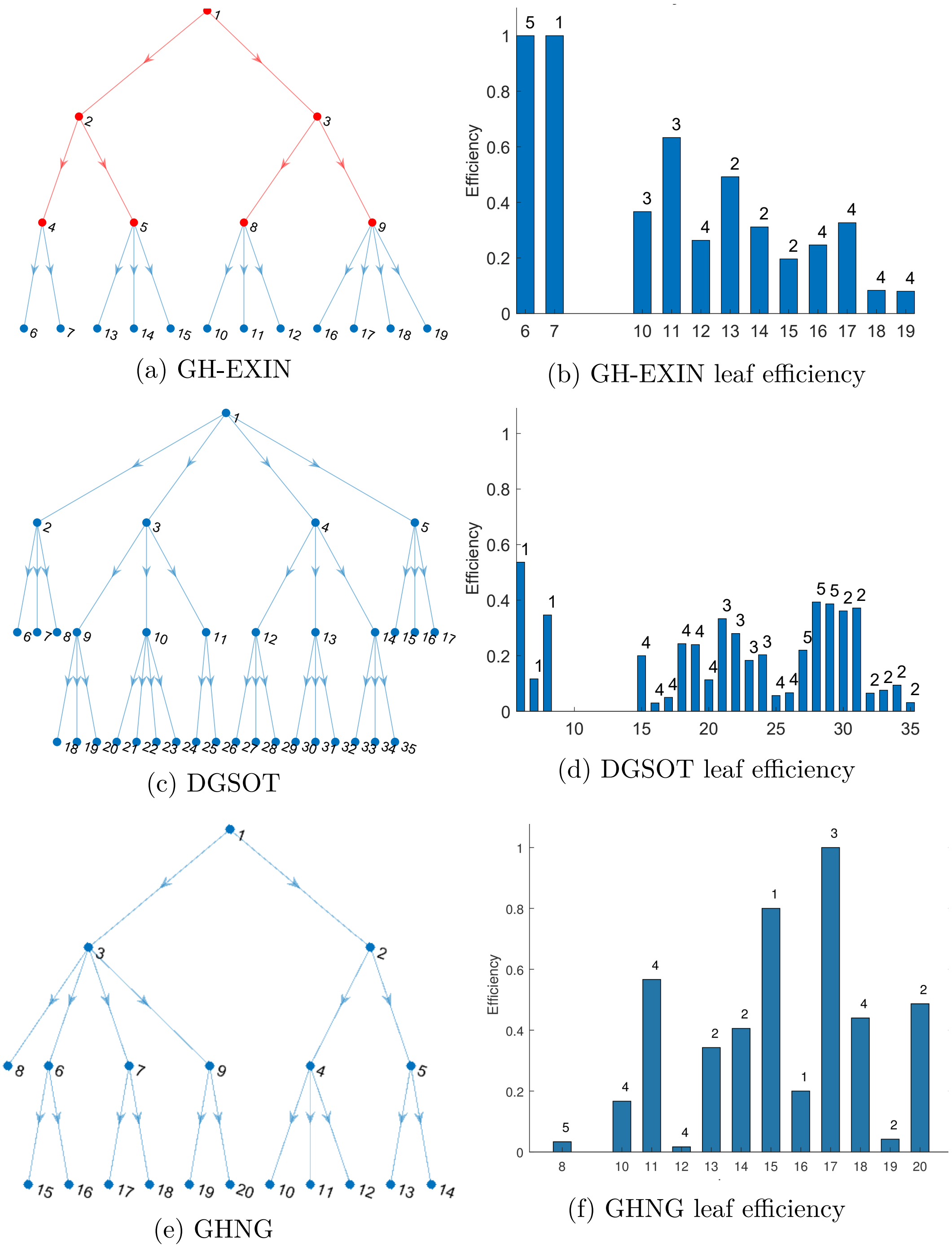
DGSOT first layer is similar to GH-EXIN second level:

- node 2: all class 1 samples;
- node 3: all class 3 inputs and 65% of class 4 data;
- node 4: all class 2 and 5 samples;
- node 5: 35% of class 4 data.

Node 2 reaches 100% for both purity and efficiency indices, while node 5 is 100% pure and only 35% efficient.

The GH-EXIN second layer is as follows:

- node 4: all class 1 and 5 samples;
- node 5: all class 2 data;



(a) GH-EXIN

(b) GH-EXIN leaf efficiency

(c) DGSOT

(d) DGSOT leaf efficiency

(e) GHNG

(f) GHNG leaf efficiency

Figure 5.14: First experiment: video sequences. Final trees (left) of GH-EXIN (top), DGSOT (middle) and GHNG (bottom): for easy of comparison, nodes are labelled as x-axis of the leaf efficiency bar plots (right). The class with the greatest efficiency within the leaf is reported on top of each bar.

- node 8: all class 3 inputs and 25% of class 4 samples;
- node 9: 75% of class 4 points.

Node 9 reaches 100% purity but 75% efficiency for the same class of DGSOT node 5. In addition, node 4 correctly splits the two women classes as two 100% efficient and pure sons, i.e. nodes 6 and 7.

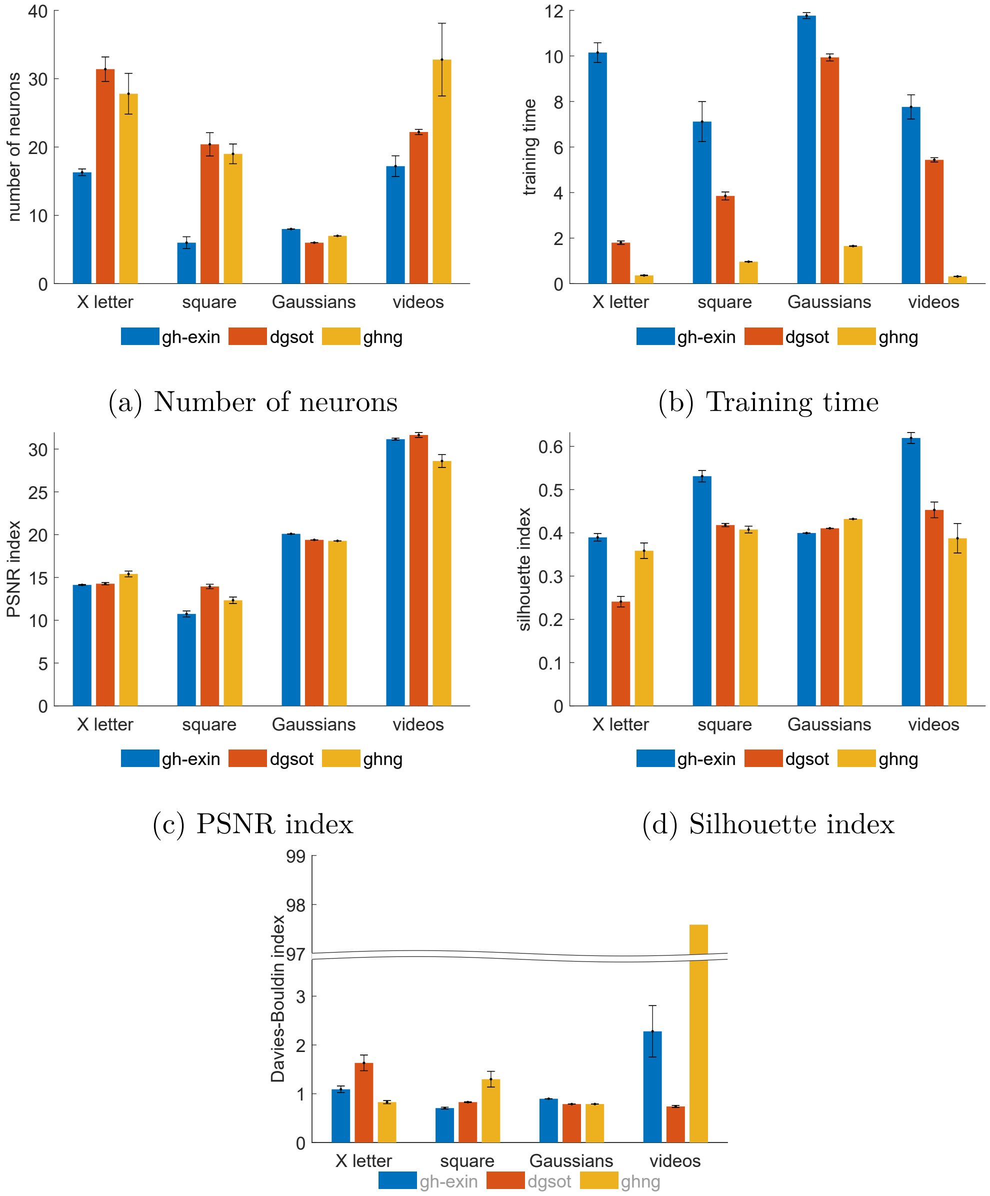
Both the GH-EXIN third layer and the DGSOT second level are composed of units whose Voronoi is made of a single class. Nevertheless, 65% class 4 samples for DGSOT and 25% class 4 inputs for GH-EXIN are grouped together with class 3.

The GHNG first layer does not perform a proper partitioning: node 2 holds a slice of class 2 and 4 samples; therefore, these classes will be grouped in the subsequent layers by clusters in different branches. Indeed, the lack of a reallocation technique prevents from solving this issue. The GHNG second layer has an higher amount of units w.r.t. the same level of the other two methods. Class 2 is shared, in equal parts, by nodes 5 and 9, which are in different branches. The same happens for class 4 and nodes 4 and 7. Class 1 is fully clustered from node 6, while class 3 is just found in the third layer. Node 8 yields the worst result because it only collects less than 0.1% class 5 samples; therefore, a proper class 5 clustering is inhibited. Even if node 8 Voronoi set is empty (this is allowed by the GNG algorithm[1]), its efficiency is higher than zero; it derives from the recall phase, in which it is possible that an *empty* neuron wins because its weight vector is in another unit Voronoi set. The same considerations hold for node 12. Conversely, this drawback is fixed in GH-EXIN by means of the data reallocation method performed at the end of each sG-EXIN training epoch.

Fig. 5.15 shows that GH-EXIN uses less neurons than the others. As before, GHNG is by far the most rapid to execute. *PSNR* has similar values, *S* is quite higher for GH-EXIN, while *DB* is lower for DGSOT. It has been proved experimentally that the extremely high GHNG *DB* value is due to the empty neurons. Remember that these indexes do not measure the hierarchy quality, but only the final partitioning.

Finally, both GH-EXIN and DGSOT yield an optimal hierarchical clustering. In addition, the former can discriminate between men and women. Conversely, GHNG creates a very bad hierarchy. Probably, this explains why, in [88], the GHNG authors do not show the whole tree, but only the results of some units, with the corresponding leaves.

[1] GNG places a novel neuron in the middle between father and mother units, i.e. its position is not related to the presence of a sample. Because it is connected to its parents, even if it never wins, it can move because of SCL (i.e. a neighbour has won) and get close to another neuron samples, which will not be fed again to the network; therefore, even if it never wins during training, i.e. its Voronoi remains empty, it may win in the recall phase.



(a) Number of neurons

(b) Training time

(c) PSNR index

(d) Silhouette index

(e) Davies-Bouldin index. On the video dataset is not shown the GHNG s.e.m. bar because o its range.

Figure 5.15: Bar plots (values are averaged over 10 trainings) with s.e.m. intervals of experiment statistics for each neural network.

Table 5.1: GH-EXIN hyperparameters

	H_{max}	H_{perc}	α_{0_1}	α_{0_n}	age_{max}	min_{card}
X-shape	0.00002	0.9	0.1	0.01	5	10
Square	0.00002	0.9	0.35	0.001	10	30
Gaussians	0.001	0.9	0.5	0.05	5	300
Videos	0.8	0.9	0.8	0.1	20	10

Table 5.2: DGSOT hyperparameters

	α	σ_0	T_R	T_E	ϵ_{AD}	ϵ_{ET}	K
X-shape	0.2	1	0.3	10	0.046	0.03	1
Square	0.1	1	0.001	10	0.09	0.03	0
Gaussians	0.2	1	200	2	0.2	0.05	1
Videos	0.2	1	250	2	0.2	0.05	1

Table 5.3: GHNG hyperparameters

	MAX_{LEVEL}	τ	λ	ϵ_B	ϵ_N	α	A_{max}	D
X-shape	2	0.25	100	0.1	0.01	0.5	50	0.995
Square	2	0.3	100	0.35	0.01	0.5	50	0.995
Gaussians	2	0.1	100	0.4	0.01	0.5	14	0.995
Videos	3	0.2	100	0.001	0.001	0.5	50	0.995

5.2.6 Gene expression analysis experiment

The second real experiment deals with a hierarchical clustering of DNA microarrays extracted from cancerous tissues. The cancer phenomenon seems to origin from a sequence of genetic alterations. In this complex scenario, clinical treatments adds a level of external complexity to the tumour behaviour. In recent years, Patient-Derived Xenografts (PDXs) have proved to be reliable tools for biomarker discovery and drug development in oncology [101, 102, 103]. PDXs have been exploited for doing large-scale preclinical researches for determining correlations between genetic or functional traits and sensitivity to anti-cancer drugs. In such a scenario, along the last 10 years, the Cancer Institute of Candiolo (IRCC, Italy) has put together the biggest academic database of PDXs from metastatic colorectal cancer (mCRC) available worldwide. Samples have been largely characterized at the molecular level through the Illumina bead array technology [104], and each record has been labelled w.r.t. the response to therapies, including cetuximab, an anti-EGFR antibody approved for clinical use [105, 106, 107]. The resulting dataset is made of DNA microarrays, with the expression of 20.023 genes in 403 CRC murine tissues. Each cancerous tissue is labelled by a Boolean variable, which represents the tumour response to cetuximab (responsive or not responsive), as described in previous works

[108, 109, 110, 111, 112]. The resulting dataset lies in a very high-dimensional space, where data manifolds are embedded in subspaces. This is a harder problem compared to normal clustering, because also the more significative features, i.e. genes, must be detected. Even if this strategy is called subspace clustering, there is not a unique definition of the related techniques. In a certain sense, it relies on the implementation: *two-way clustering*, if it is applied first in the row and then in the column space of the input matrix [113]; *biclustering*, if these operations are performed simultaneously [94].

Two-way clustering

Two-way clustering seeks biclusters with constant values, either on rows or columns, and with coherent values. It can be demonstrated that the rank of the corresponding submatrices is lower than four in absence of noise; therefore, it can be exploited to measure the bicluster quality. To this purpose, the H_{cc} index has been used because it also considers the noise in training set. It is designed for biclustering problems, but, here, it is extended to two-way clustering. Considering the characteristics of the biclusters that can be found, it can also be employed for plain clustering. It is defined as:

$$H_{cc} = \frac{\sum_i^{N_r} \sum_j^{N_c} r_{ij}^2}{N_r N_c} \tag{5.5}$$

where N_c represents the total number of columns of the matrix, N_r represents the total number of rows, and $r_{i,j}$ is the residue, which is calculated as:

$$r_{ij} = a_{ij} - \frac{\sum_k^C a_{ik}}{C} - \frac{\sum_h^R a_{hj}}{R} + \frac{\sum_i^R \frac{\sum_j^C a_{ik}}{C}}{R} \tag{5.6}$$

The components a_{ij} are the elements of the matrix representing the dataset. C and R are the number of columns and rows of the bicluster at hand, respectively. The second and third terms are the mean value of the i-th row and j-th column, respectively, while the last one is the mean value of the whole bicluster. H_{cc} value lowers as the values in the bicluster tend to be constant, differing for a constant on the rows or on the columns. It results zero for the trivial *1*x*1* bicluster. Indeed, an additional check on the bicluster cardinalities ($min_{card} > 1$) is adopted to avoid this case.

GH-EXIN, driven by H_{cc}, is trained considering first the rows as features and then the columns, as shown in Alg. 1. It is not mandatory to use the same value for the minimum cardinality for rows and columns. The column clustering can be thought as a feature selection step, i.e. an orthogonal projection in the column space; therefore, this approach can be also called *projected clustering*, because it does not allow overlapped biclusters.

Algorithm 1 Two-way (projected) clustering pseudo-code

```
two-way clustering:
clustering on rows
for all leaves do
    if $leaf.cardinality \leq min_{card_1}$ then
        skip leaf
    else
        clustering on the columns of the leaf (projection)
        for all leaves do
            if $projectedLeaf.cardinality \leq min_{card_2}$ then
                skip leaf
            else
                save projected leaf
                goto two-way clustering
            end if
        end for
    end if
end for
return
```

Experimental results

The GH-EXIN two-way clustering has been trained with the gene-expression dataset: rows are the genes and columns the murine tissues, while the records are the gene expressions. To better analyse common genetic expressions for the various patients, the training set has been split into three classes, which are related to the tissue response to anti-cancer treatment:

- start to recover after three weeks;
- stable situation;
- drugs have no effect and the tumour keeps growing.

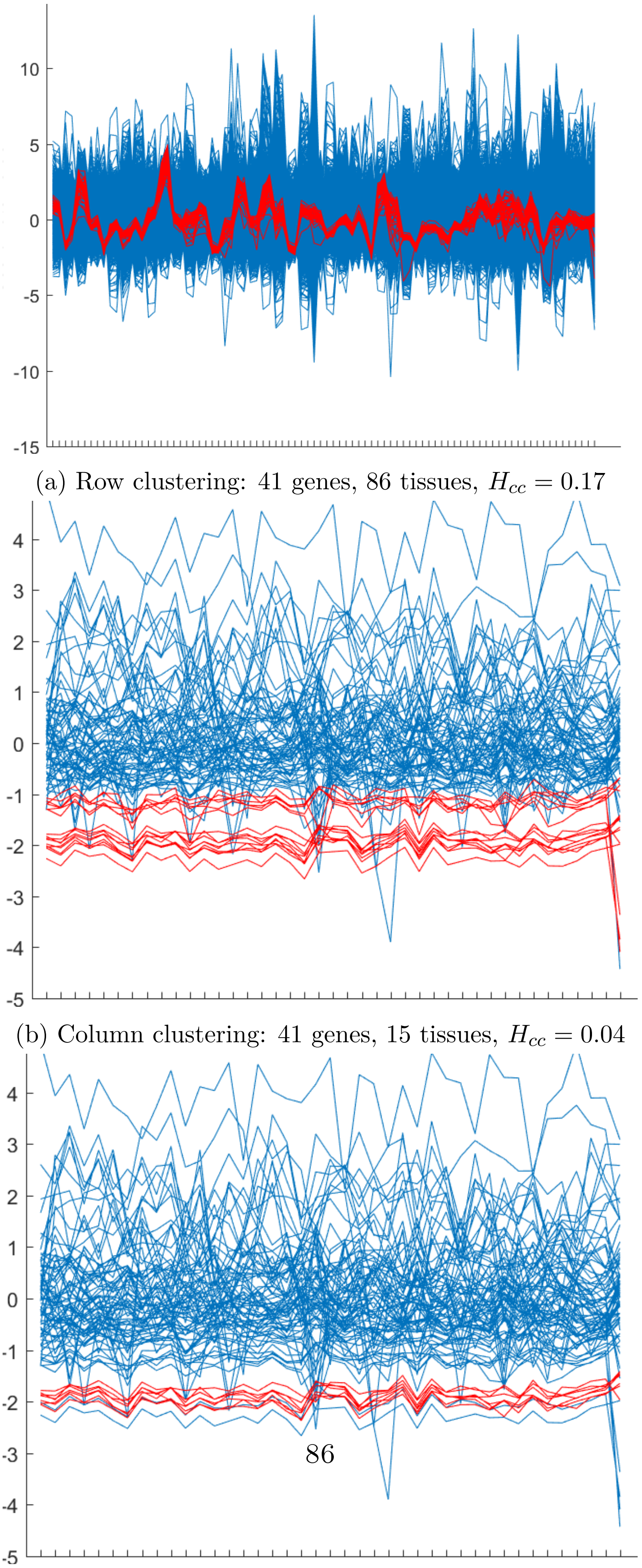
Given the huge amount of genes ($\sim 20K$), the hierarchical clustering is performed first in the row space ($H_{max} = 0.1$, $H_{perc} = 0.9$, $\alpha_{0_1} = 0.8$, $\alpha_{0_n} = 0.08$, $age_{max} = 20$, $min_{card} = 20$) and, then, on columns ($H_{max} = 0.001$, $H_{perc} = 0.5$, $\alpha_{0_1} = 0.5$, $\alpha_{0_n} = 0.05$, $age_{max} = 3$, $min_{card} = 20$). The results are studied by means of the parallel coordinates plot [114], which is a technique to visualize high-dimensional data, to compare variable behaviours and to detect their relationships. Each variable has its own axis and all the axes are placed parallel to each other, typically vertical and equally spaced. A point in a multi-dimensional space is represented as a polyline whose vertices are on the parallel axes; vertex position on the *i-th* axis corresponds to the *i-th* coordinate of the point. Fig. 5.16a shows an example of this plot on a leaf yielded by the first clustering, i.e. genes are samples (colored polylines) and tissues are features (parallel vertical axes). Blue polylines are all dataset available genes, while red polylines stand for genes collected in the gene cluster. The red grouping of polylines is thin and exhibits coherency w.r.t. features, i.e. the leaf has properly grouped similar genes; therefore, the first clustering quality is satisfactory. A similar validation analysis (see Figs. 5.16b and 5.16c) is used after the second clustering, i.e. projecting the leaf Voronoi set into the tissue space; also in this case, red polylines have a clear, common pattern.

From a biological point of view, the scientific importance of the output clustered genes has been analysed. Among all the yielded biclusters, the one with the lowest H_{cc} index value has also grouped the most significative genes in the cancer domain. Indeed, the seven genes found in the bicluster are the following:

- *CSAG1*, *CSAG3*, *CSAG3A*, which belong to the same CSAG family. These genes are associated with chondrosarcomas, but they are also present in healthy tissues. Furthermore, CSAG3 and CSAG3A are genes coding the Chondrosarcoma-associated gene 2/3 protein, which, according to [115], is a drug-resistance related protein, whose expression is connected to the chemotherapy resistant and neoplastic phenotype. May also be linked to the malignant phenotype.

- *MAGEA2*, *MAGEA3*, *MAGEA12*, *MAGEA6*, belonging to the same MAGEA family. These genes are melanoma antigens, which reduce p53/TP53 transactivation function and also repress p73/TP73 activity, as explained in [116]. Both p53 and p73 are tumour suppressor proteins, which regulate cell cycle and induce apoptosis.

The above analysis suggests that, at least in the observed conditions, these gene families are not only significative by themselves, but may also co-regulate each other. It is also important to notice that this bicluster phenomenon has been observed within the tissues belonging to the third class, to which drug unresponsive tissues belong.



(a) Row clustering: 41 genes, 86 tissues, $H_{cc} = 0.17$

(b) Column clustering: 41 genes, 15 tissues, $H_{cc} = 0.04$

(c) Column clustering: 41 genes, 7 tissues, $H_{cc} = 0.005$

Figure 5.16: Second experiment: Gene-expression analysis. Parallel coordinates for a leaf: cluster of genes (Fig. 5.16a) and tissues (Figs. 5.16b and 5.16c).

Chapter 6

Supervised Learning

Unsupervised learning is an amazing tool to deal with data whose structure is unknown a priori and to discover their underlying patterns. A perfect example is the gene-expression analysis experiment presented in the previous chapter; among a multitude of input features, novel, interesting, gene co-regulation were found (see Sec. 5.2.6). On the contrary, when the problem is well known a priori, a supervised approach can be a better way to tackle it. Indeed, embedding an external knowledge in a neural system enriches the learning process and, above all, yields a more powerful tool. Of course, such an approach can be exploited only for stationary distribution, whose classes have to be known before the training begins. This is the case, for instance, of medical applications, where physician expertise can be conveyed to the neural architecture by means of sample labelling. In this sense, doctors become the teachers of a new virtual assistant, which will help them to perform a faster and more accurate diagnosis.

To this purpose, in the following two biomedical clinical applications are studied by means of the proposed neural approach. Given the stationariness of the problems at hand, it is possible to perform an initial analysis for determining the intrinsic dimensionality, which can be exploited for sizing the neural system input layer, i.e. the amount of features to be fed. At the same time, different feature extraction approaches are employed to assess the validity of the selected features. The simpler technique considers raw data as meaningful in themselves, i.e. feature extraction is not performed; in this sense, it can be seen as a benchmark. Moreover, in some applications, like telemedicine wearable devices, both the computational power (algorithms are often embedded into smartphone apps) and a short acquisition time are a key constraint for making the device user-friendly and, therefore, effective in preventing diseases; in such a scenario, feature extraction cannot be performed and a lighter algorithm is preferable. On the contrary, there are applications, e.g. offline medical analysis, where these requirements do not need to be satisfied, and feature extraction is employed because it provides meaningful information on the signal, e.g. its statistical time-evolution.

6.1 Arterial blood pressure estimation

High blood pressure (aka hypertension) is a global health disease, which constitutes one of the main causes of premature deaths, killing around 8 million people per year [117]. Hypertension is a pathology where the blood vessel pressure is constantly high; it causes an intensified heart pumping and an increased arterial stiffness. A *normal* blood pressure (BP) is when systolic (SBP) and diastolic (DBP) blood pressures are less than 120 mmHg and 80 mmHg, respectively; conversely, hypertension occurs when either SBP or DBP (or both of them) exceed these thresholds. SBP corresponds to the arterial pressure as heart contracts and, thus, is the highest value reached by the pressure; the DBP represents the arterial pressure during heart resting between two consecutive beats.

According to the American Heart Association (AHA), it is possible to categorize the hypertension in five level of severity w.r.t. the pressure values (in mmHg) [118]:

- Healthy: $SBP < 120$ and $DBP < 80$
- Pre-hypertensive: $120 \leq SBP < 130$ and $DBP < 80$
- Hypertension stage 1: $130 \leq SBP < 140$ or $80 \leq DBP < 90$
- Hypertension stage 2: $140 \leq SBP$ or $90 \leq DBP$
- Hypertensive crisis: $180 \leq SBP$ or $120 \leq DBP$

Hypertension is hard to detect because it does not present any symptoms; in this sense, it is part of the diseases called *silent killers*. The best way to prevent the onset of irreversible problems, such as coronary heart disease or stroke, is continuously monitoring blood pressure.

The gold standard technique for BP measurement employs a sphygmomanometer and the Korotkoff sound technique [119]: the cuff is first inflated with a pressure highly above the SBP and then gradually lowered; when the cuff pressure equals the ABP, the Korotkoff sounds become audible through the stethoscope. The first sound is the SBP; by further lowering the cuff pressure, the noises become more intense, and then start to disappear: the complete absence of sounds is the DBP [120]. This approach is non-invasive but it is prone to big inaccuracies, which can affect the classification process and prevent the timely identification of subjects at risk [121]; for example, a wrong cuff positioning may imply a misclassification of a subject, and there is the tendency for BP to increase in the presence of a physician (aka the *white coat effect*).

An alternative measurement approach is the intra-arterial blood pressure (IBP), an invasive technique mainly employed in the Intensive Care Unit (ICU) and in the operating theatre. This method performs a direct measurement of arterial blood pressure (ABP) by inserting a cannula needle in a suitable vessel. The great advantage is the possibility of performing a continuous patient ABP monitoring together with its visualization on a screen [122]. Albeit the method is accurate, it can be employed only in medical facilities due to its invasiveness and the possibility of infections [123].

In order to overcome the limits of both the invasive and cuff-based methods, several researches have proposed cuff-less non-invasive techniques to measure ABP. Among these, Pulse Wave Velocity (PWV) propagation estimates BP by employing the mathematical description of Moens and Korteweg [124]. The inverse proportionality between the BP and the PWV is shown in [125]; however, the proposed mechanical-mathematical model is based on physiological parameters, which are unsuitable to be measures, like the artery diameter or the distance from heart to fingertip. In alternative, [126] proposes the use of the Pulse Transit Time (PTT), which is the time required for the pulse wave to travel between two arterial sites within the same cardiac cycle. The model proposed by [127] solves the issue of the physiological parameters acquisition but, the mathematical relation between PTT and BP is prone to approximations, which make the model not very robust.

Resuming, state-of-the-art techniques are generally based on estimated physiological parameters averaged on a quite different population; therefore, these approaches cannot generalize effectively. Moreover, measuring these parameters, e.g. artery diameter, is problematic.

To overcome these limits, few neural network models have been proposed in literature. An overview and comparison is presented in [128], which shows how these models still have a low prediction reliability for both SBP and DBP.

6.1.1 The proposed neural approach

The proposed strategy tackles the ABP generalization problem by means of artificial neural networks (ANN) [129]. To predict ABP values, the photoplethysmographic (PPG) signal is fed as input of a neural system. In literature, blood pressure has been proven to be strictly related to PPG [130], which is an optical measurement technique for identifying blood volume variations in the microvascular bed of tissues [131]. Because BP is, also, strictly related to cardiac activity, the electrocardiographic signal (ECG) as been also employed as neural input and the results have been compared with the PPG ones. In this sense, understanding the ABP behaviour is addressed by means of a regression approach, where the supervised architecture is trained to learn the physiological relation between the inputs (ECG and PPG), and the output (ABP).

As proven in the following experiments, this strategy overcomes both the invasive approach and the non-invasive mathematical models; indeed, despite it is still a non-invasive method, the predicted values resemble the invasive ones, but it does not need a cuff to be inflated, which is quite uncomfortable for the users.

To tackle the regression problem, three different supervised neural networks have been trained for assessing both the best input set and the best architecture [132, 133]:

- Multi-Layer Perceptron (MLP)
- Output-Error Neural Networks (NNOE)
- Long Short Term Memory (LSTM)

Each network has been trained using IBP as target and both ECG and PPG as inputs. The predicted signal is compared with the target one, i.e. IBP, w.r.t. the root-mean-square error (RMSE) [134], which yields the prediction error ($RMSE = 0$ is the perfect prediction). Then, during recall, systole (SBP) and foot (DBP) points are extracted (see Fig. 6.1) from both the target and output signals, and the relative error (in mmHg) is computed. Indeed, for each patient, both the IBP target signal and the network output are continuous ABP signals, which need, first, to be discretized to obtain SBP and DBP values, and, then, averaged to determine two single pressure values, i.e. systolic and diastolic pressures, with regard to the whole acquisition.

Finally, the network SBP and DBP were compared with those yielded by a certified sphygmomanometer; in this sense, also the non-invasive blood pressure (NIBP) measuring performance were tested. The idea is to exploit the IBP as a benchmark for the comparison of the two non-invasive techniques: the neural network and the sphygmomanometer.

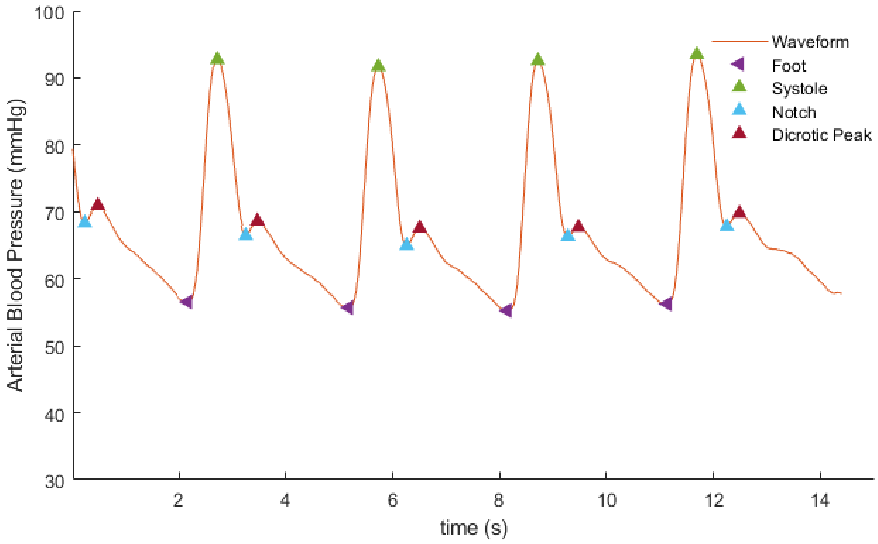


Figure 6.1: ABP signal characterization (after filtering): foot point (violet), systole point (green), notch point (blue), dicrotic peak (red).

The trained model can be embedded in e-health wearables, such as the VITAL-ECG, which will be presented in detail in Chap. 8. The aim is to provide this kind of devices with an anytime, everywhere, unobtrusive BP measurement feature. In this sense, IBP is exploited to make the neural model able to correctly relate the acquired ECG and PPG signals with its corresponding ABP values (systolic and diastolic) even when used in a non-invasive approach.

The MIMIC dataset

The proposed approach requires a training set where ECG, PPG and ABP (both as IBP and NIBP) signals were acquired simultaneously and, above all, synchronously.

The dataset used for the neural network training derives from the MIMIC (Multiparameter Intelligent Monitoring in Intensive Care) database [135, 136], a multi-parameter collection where clinical data are obtained from the patient's medical record; in particular, it includes, synchronous PPG, ECG and IBP signals as shown in Fig. 6.2.

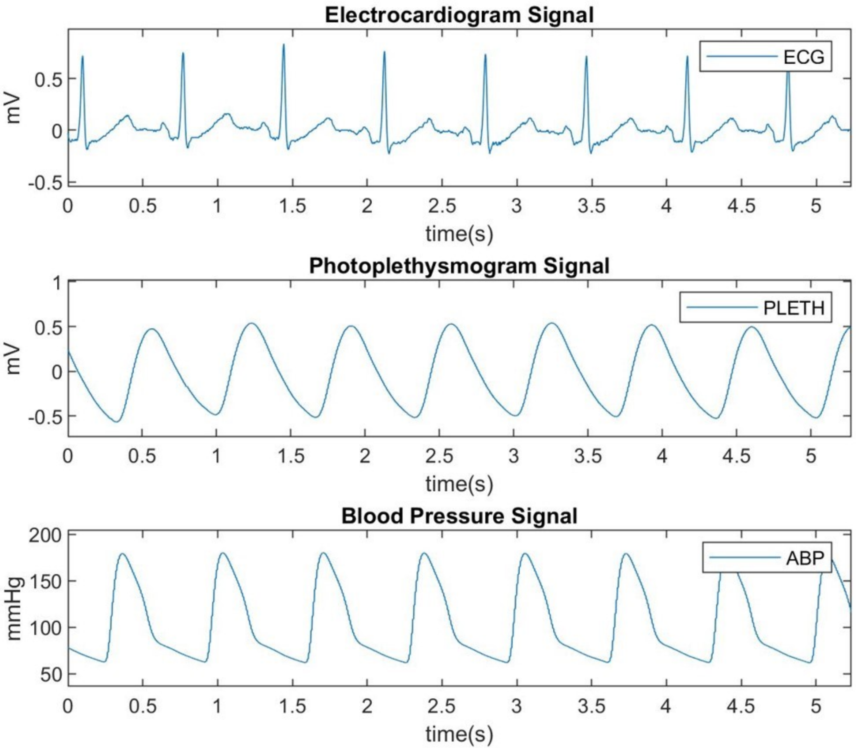


Figure 6.2: Synchronized signals representation: ECG (top), PLETH/PPG (center), ABP (bottom).

For each of the three signals, thirty-seven ten-minute patient recordings have been extracted from the MIMIC database.

For network training, both the input and target have been randomly split into three subsets: 70% for training, 15% for validation, and the remaining 15% for testing. The k-fold cross validation technique [137] was employed to assess the validity of the results.

6.1.2 MLP

As a first benchmark, a simple MLP [138] has been trained. The input layer, whose size is equal to the number of inputs fed to the MLP, is made of passive units, which do not alter the input, but only transmit the information to the following layer. Different amount of hidden layers have been tested for assessing the related network regression performance; each hidden layer has an arbitrary amount of units, which alter its input by means of the weights and propagate the information using the activation function.

The optimal architecture has 15 and 8 neurons in the first and second hidden layers, respectively, equipped with the hyperbolic tangent as transfer function; the output regression layer, instead, employs the linear function. The training algorithm is based on the Levenberg–Marquardt technique [139], which is usually used in curve-fitting problems, because is quite performing in local minimum seeking; in this sense, it can be considered as a trade-off between the gradient descent and the Gauss–Newton algorithms. The network is trained for 50 epochs by means of the backpropagation technique [140]. The raw signals have been filtered with a moving mean, whose window length is equal to three. Both ECG and PPG are used as input.

Fig. 6.3 shows the network predicted ABP w.r.t. the IBP target. Their superposition proves the network predicted ABP values, i.e. its outputs, closely match the invasive values, i.e. the targets. This consideration holds for almost each patient except subjects 32 and 5 for the diastolic and systolic cases, respectively; thus, the MLP has proven to be a valid tool to predict ABP given the invasive blood pressure measurements.

With regard to the NIBP, i.e. the sphygmomanometer, the MLP outperforms the gold standard method because it shows a higher reliability. Indeed, as shown in Table 6.1, the MLP RMSE is smaller for both DBP and SBP, i.e. 3.2 and 3.6, respectively. The proposed method, and consequently each device embedding it, would comply with the ANSI/AAMI/ ISO 81060- 2:2013 (the sphygmomanometer certification regulation) because the yielded ABP values are within +/- 5 mmHg w.r.t. the IBP.

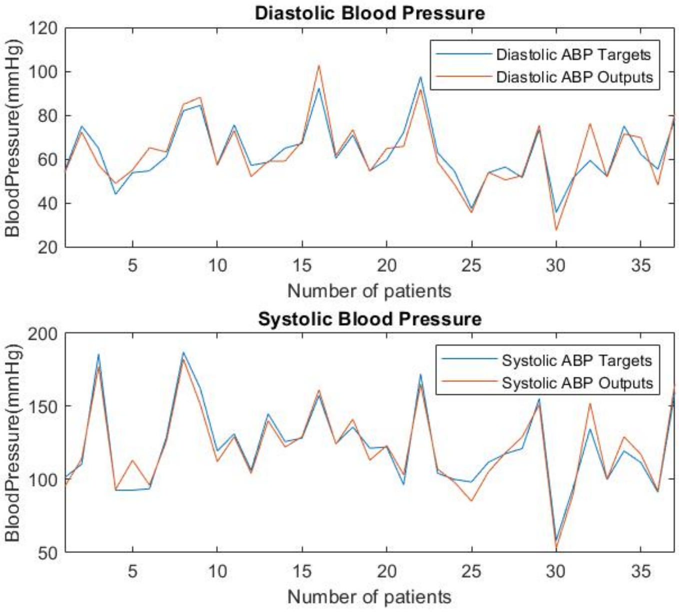


Figure 6.3: MLP: DBP (top) and SBP (bottom) prediction performance (red) w.r.t the IBP target (blue).

Table 6.1: MLP RMSE (in mmHg) comparison

	DBP	**SBP**
Sphygmomanometer	4.1	4.7
Multi-layer perceptron	3.2	3.6

6.1.3 NNOE

Neural Network Output-Error (NNOE) aims to identify non-linear dynamic systems in stochastic environments [141]. The model structure is shown in Fig. 6.4.

The Z^N defines the whole system:

$$Z^N = \{[u(t), y(t)]\}_{t=1,...,N} \tag{6.1}$$

where $u(t)$ and $y(t)$ are the control and output signals, respectively, and t is the sampling instant.

The set of regressors must be tuned accordingly to the problem at hand. For NNOE, the regression vector is given by:

$$\varphi(t) = [\hat{y}(t-1|\theta) \ldots \hat{y}(t-n|\theta)\ u(t-d) \ldots u(t-d-m)]^T \tag{6.2}$$

where θ is a vector containing the weights, n is the y-predicted lag, m is the input lag, and d is the delay for obtaining the prediction (aka skip). The prediction vector is defined as:

$$\hat{y}(t|\theta) = g(\varphi(t), \theta) \tag{6.3}$$

where g is the function mapping yielded by the neural network.

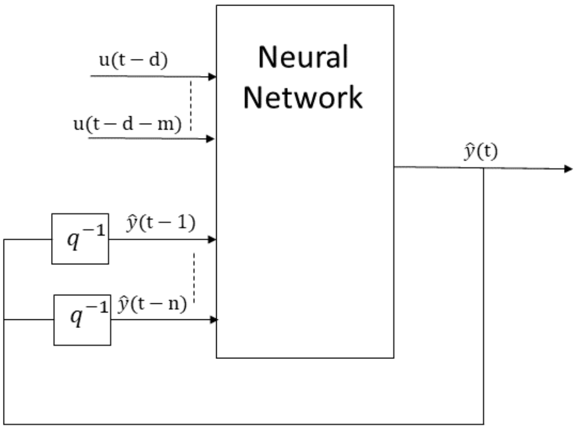
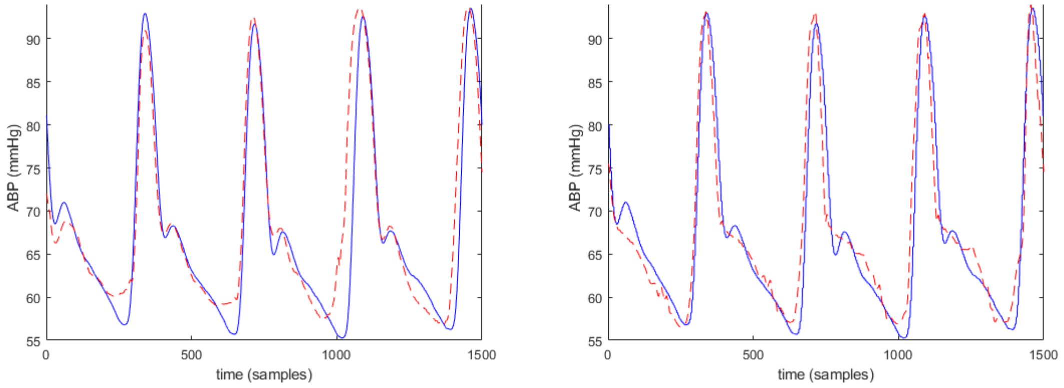


Figure 6.4: The NNOE model structure

In the application at hand, the NNOE architecture was implemented by a multilayer perceptron (MLP), because of its ability to learn non-linear relationship from an input set, whose hidden layer has 35 and 40 neurons when the input signal is the PPG or the ECG, respectively. The hyperbolic tangent is employed as activation function. As before, the Levenberg-Marquardt method is used to train the network. The chosen error function is the sum of squared errors.

Several network configurations have been tested w.r.t. the amount of regressor units; the best one has resulted to be the one equipped with six regressor units. The network is trained twice: first with PPG signal as input and then with ECG as input. Before comparing the network output with the target, a moving mean filter (window length equal to 25 and 10, respectively) is applied to the output signal for removing noise artifacts.

Fig. 6.5 shows the comparison between target (blue solid line) and output (red dashed line) signals with PPG (see Fig. 6.5a) and ECG (see Fig. 6.5b) as input, respectively. In both cases, the prediction is accurate. The model has been evaluated in terms of RMSE, which shows better performance for NNOE with ECG input (RMSE = 2.42) than PPG (RMSE = 5.80). This consideration is confirmed also w.r.t. the absolute errors for SBP and DBP, which confirms ECG input is better suited than PPG for the proposed NNOE. Table 6.2 summarizes the prediction error (in mmHg).



(a) PPG

(b) ECG

Figure 6.5: NNOE prediction: target (blue solid line) vs output (red dashed line). Two inputs are compared: PPG (left) and ECG (right).

Table 6.2: NNOE prediction error (in mmHg)

	RMSE	**DBP**	**SBP**
PPG	5.80	2.36	0.69
ECG	2.42	1.26	0.7

6.1.4 LSTM

Recurrent Neural Networks (RNN) are a generalization of feedforward neural network with embedded an internal memory. The most popular RNN's are the Long Short-Term Memory (LSTM) networks [142], which resolve the RNN vanishing gradient issue. LSTMs are built on special units called *memory blocks*. Each memory block contains an *input gate*, which drives the input activation flow, and an *output gate*, which regulates the cell activation output flow into the rest of the network. The *forget gate* is used to scale the cell internal state before it is added to the cell input through the self-recurrent connection, therefore forgetting or resetting the cell's memory [143].

The working memory is called the *hidden state* h_t, which stores the information on past inputs and it is also exploited for predictions:

$$h_t = o_t * tanh(c_t) \tag{6.4}$$

being c_t is the current state of the cell, defined as:

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{6.5}$$

where the forget gate f_t determines which information has to be forgotten by multiplying 0 to a position in the matrix and it is given by:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{6.6}$$

The input gate i_t decides which information can be stored into the cell state:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{6.7}$$

The modulation input gate $\tilde{c}_t$ allows the cell to forget memory:

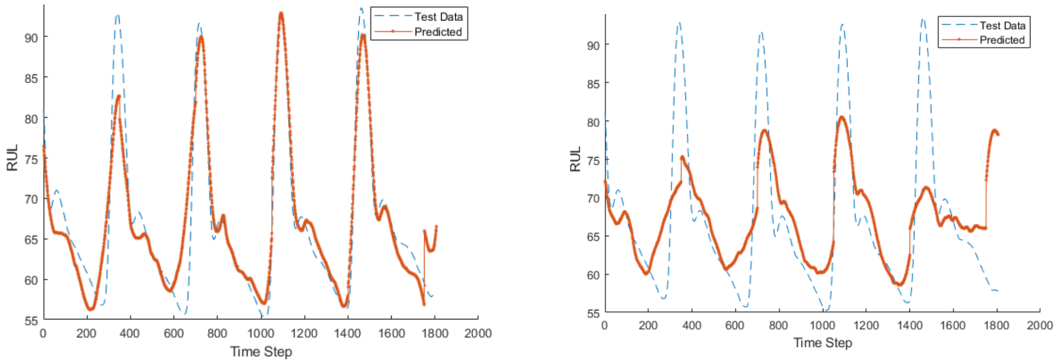
$$\tilde{c}_t = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{6.8}$$

The output gate o_t yields the next hidden state:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{6.9}$$

here, W are the weight vectors, x_t the input vector, b is the bias and σ is the sigmoid function.

In the application at hand, the LSTM architecture is made of one input layer (350 units), one recurrent hidden layer (equipped with 500 units) and one regression output layer. To minimize the training error and avoid minimal points, the Adam optimizer is employed, which is an adaptive optimization technique very well suited for training deep neural networks (DNNs) [144]. The network has been trained as NNOE: first on PPG, then on ECG. Fig. 6.6 yields the comparison between target (blue dashed line) and output (orange solid line) with PPG (see Fig. 6.6a) and ECG (see Fig. 6.6b) as input, respectively. The ABP prediction is very accurate in the PPG case; on the contrary, when the ECG is used as LSTM input, the prediction is quite bad. This is also confirmed by the RMSE, which is equal to 5.35 and 7.43, respectively, and by means of the absolute errors for SBP and DBP, as shown in Table 6.3. In this sense, LSTM exhibits an opposite behaviour w.r.t. NNOE.



(a) PPG

(b) ECG

Figure 6.6: LSTM prediction: target (blue dashed line) vs output (orange solid line). Two inputs are compared: PPG (left) and ECG (right).

Table 6.3: LSTM prediction error (in mmHg)

	RMSE	**DBP**	**SBP**
PPG	5.35	1.51	5.26
ECG	7.43	5.54	12.72

6.1.5 Network evaluation

ABP is an important physiological parameter, which must be monitored to prevent and detect cardiovascular diseases. To this purpose, the MLP has been trained on PPG and ECG, while NNOE and LSTM either on PPG or ECG. The three architectures have been compared with IBP and NIBP gold standards.

The MLP predictive performances are quite promising; indeed, it outperforms the sphygmomanometer and it is compliant with the ANSI/AAMI/ ISO 81060-2:2013 because predictions are within +/- 5 mmHg w.r.t. IBP.

The two recurrent neural networks do not show coherency on which is the best input, PPG or ECG. Actually, by looking at RMSE and absolute errors, they show an opposite behaviour, NNOE behaves better with ECG, while LSTM with PPG. In addition, ECG-NNOE yields the best configuration in terms of both the proposed metrics; the predicted ABP respects the normative ANSI/AAMI/ ISO 81060- 2:2013 for sphygmomanometer certification.

NNOE is tailored on the input signal by means of the regressors and associated lag choice, which is related on the application at hand. In this sense, the method can be seen as a feature selection, unlike the deep learning approach, which automatically extracts its own attributes. Albeit NNOE needs far less inputs than LSTM (6 vs 350), the time-sequence is not well understood, at least in the ECG case, by the deep approach.

The proposed NIBP neural approach can be embedded in a wearable, unobtrusive devices, such as the VITAL-ECG, and used to fight cardiovascular diseases and prevent their dangerous effects. In this sense, the MLP requires two synchronous input signals, while the NNOE only the ECG. Conversely, the MLP architecture is simpler to be implemented than NNOE because of the absence of feedback connections. In this sense, the choice of the best architecture to be embedded in a wearable device is related to the hardware at hand; if the computation power is not an issue, NNOE can be preferred because it uses only ECG, while if a simpler microcontroller is used with both ECG and PPG as inputs, then the MLP is better suited for ABP estimation.

6.2 Parkinson's handwriting feature analysis

Neurodegenerative diseases (NDDs) [145] affect the central nervous system (CNS) by means of neuronal necrosis, which leads to an unavoidable and permanent brain damage. The reasons behind the onset are still unknown [146]. Certainly, different factors, e.g. genetic or the environment, concur in the pathology onset [147]. NDDs are characterized by a gradual brain damage that is phenotypically evident only when it reaches an advanced stage: on average, when the NDD is diagnosed, the subject has already lost up to 70% of the neurons, thus reducing the chances of an effective treatment [148]. It is crucial to design reliable early detection techniques for intervening with a tailored therapy when the neuronal destruction mechanism is in the early stages.

Parkinson's disease (PD) [149, 150] is a NDD of the CNS that affects muscle control, and therefore can alter movement, speech and posture. It is often characterized by muscle stiffness, tremor, deficit of physical movement, and in the most severe cases, a complete loss of physical motion. From a pathological perspective, it does not exist a reliable technique for an objective and quantitative diagnosis of Parkinson's disease.

Calligraphy and speech are motor control tasks accomplished by human brain; thus, their degradation derives from a neurological deterioration. In this sense, handwriting analysis can be a useful tool both for diagnostic and disease progress monitoring. Several tests [151], e.g. house drawing, can be exploited for investigating the NDD progress. One of the most famous studies for PD diagnosis deals with the analysis of patient calligraphy [152]. Indeed, it is often accompanied by the arising of micrographia, which is a contraction of the writing size, and other deficits w.r.t. geometry, kinematics, pressure patterns and air movement [153, 154].

Feature transformation techniques, such as PCA and independent component analysis [155], build a new set of features by converting the original ones. On the contrary, feature selection techniques lower the input space dimensionality by ignoring the irrelevant features and retaining only the most meaningful ones. Both approaches have been employed for dealing with handwriting recordings. A popular handwriting-based technique for PD detection extracts kinematic features, which can be either a single value or a time series [156]. A comparative analysis of these techniques and their application to PD handwriting is presented in [157]. In [158] it is proposed an experimental analysis of ANOVA [159], which is a method for determining if differences in two or more datasets are statistically significant. An alternative approach [160] proposes feature selection, based on Support Vector Machine with Radial Basis Functions as kernel [161]; it is employed for classifying input data into two groups (PD and healthy). In classification tasks, the selected features are fed to the clustering techniques. According to [162], high-level attributes, i.e. those that discriminate better among classes, are more relevant than the others w.r.t. performance. In the ReliefF algorithm [163], features are chosen based on their suitability with target function; the idea resembles the k-NN basic rules. The Sequential Forward Selection (SFS) [164] is a simple and fast feature selection method, built on a greedy search algorithm, which assembles the attribute subset by maximising its efficiency.

6.2.1 The proposed neural approach

The proposed strategy aims to analyse a PD handwriting database using neural networks. As a consequence, the purpose is not to build the perfect classifier, but to assess the quality of the corresponding attributes. The underlying assumption is the best phenomenon description is represented by the best performing classifier. In this sense, neural networks are not employed in a traditional way, i.e. for classifying data, but, instead, they are exploited as a tool for exploring the data manifold.

The database under study is made of handwriting gathered from 36 Parkinsonian patients (18m and 18f, aged between 33 and 83 years old) and 10 healthy subjects (6m and 4f, aged between 49 and 67 years old) enrolled at the Matarò Hospital in Barcelona. Subjects were all right-handed: 22 of these had attended primary school (21 PD/1 Healthy), 17 secondary school (9 PD/8 Healthy), 6 University (5 PD/1 Healthy) and one had not attended any academic studies.

PD patient handwriting was recorded before and after the daily drug (L-dopa COMT catecolo–metal transferasi) administration. Testing was performed individually in an audiovisual noiseless facility. At the beginning, the task was explained to the attendees, which consisted in writing the sentence "*La casa de Barcelona es preciosa*" (in Spanish, their native language). Handwriting has been gathered by means of a digitizing tablet with an ink pen. Such an approach overwhelms the classic method, based on posterior paper scanning, because it can measure the pen pressure on the tablet even if the pen does not touch its surface. Data were acquired using the Intuos Wacom digitizer, whose sample frequency is 100 Hz; the total amount of samples is around 244K. The recorded features are the same of [165, 166]: X and Y pen positions (the spatial coordinates), altitude (the angle between the pen and the tablet surface along the vertical), azimuth (the horizontal angle between the pen and the tablet surface) and the pen pressure on the tablet surface.

6.2.2 Dataset linear analysis

The techniques explained in Chap. 2 have been employed to determine the dataset intrinsic dimensionality and to understand the manifold topology; the former has been analysed using Pareto charts, the latter with biplots. The whole dataset, say *H-Pre-Post*, is made of the healthy participants (H) together with the PD patients before (Pre) and after ($Post$) the medical treatment. It has been fed to PCA and the relative Pareto chart is shown in Fig. 6.7. The bars represent in decreasing order the singular values w.r.t. the associated principal components. The first four PCs explain 88.47% of the variance, which suggests the manifold intrinsic dimensionality is around five.

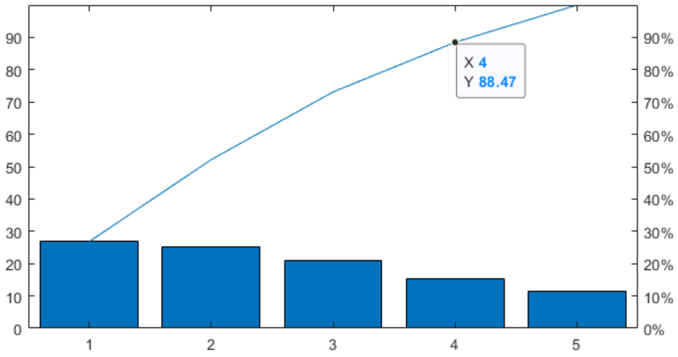


Figure 6.7: Parkinson's handwriting. Pareto chart on the whole dataset *H-Pre-Post*.

Further information on the PCA linear analysis can be retrieved from a biplot, which, as explained in Sec. 2.1.2, is a generalization of a scatterplot for visualizing, at the same time, both samples and features of a matrix; in this sense, it allows to display both the samples projected into the PCA space together with the input feature directions. Fig. 6.8 shows the biplot of the PCA projection of the *H-Pre-Post* dataset. Albeit data cluster along PC3, it is not possible to assert which features discriminate, i.e. explain, the three data clusters (healthy, pre-treatment, post-treatment).

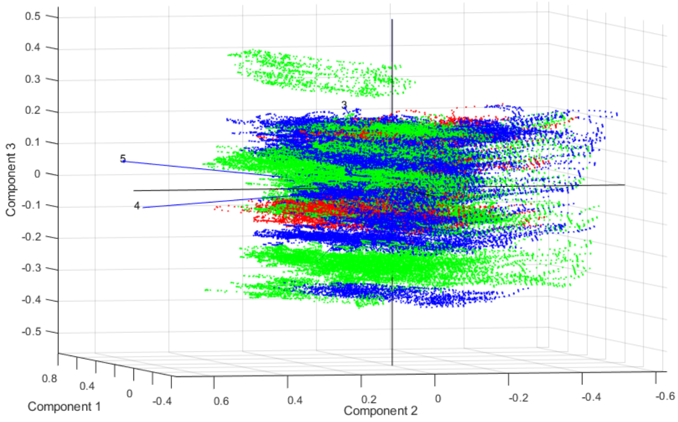
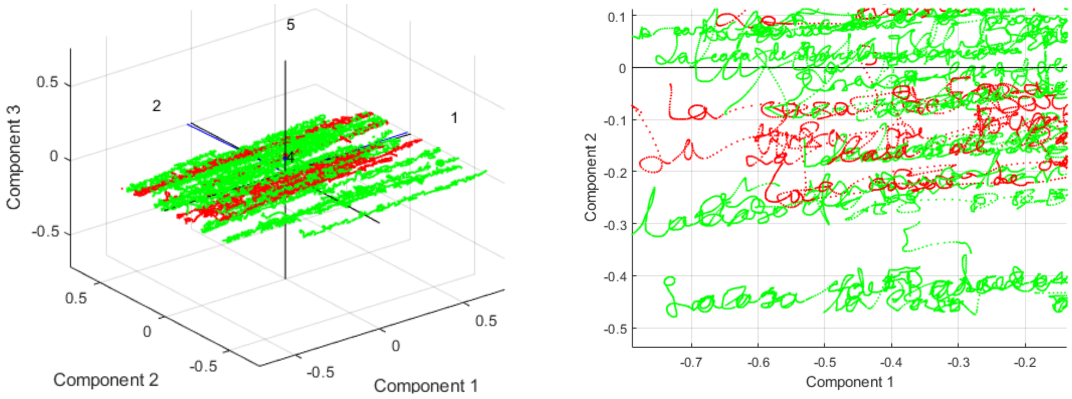


Figure 6.8: Parkinson's handwriting. Biplot on *H-Pre-Post* dataset: healthy (red), pre-treatment (green), post-treatment (blue). Blue lines are the input directions, whose corresponding feature is given by the numerical label.

Because the analysis based on Fig. 6.8 is not conclusive, a deeper study of the database subset has been performed. To this purpose, three new subsets have been created:

1. *H-Pre*: healthy and pre-treatment PD patients.
2. *H-Post*: healthy and post-treatment PD patients.
3. *Pre-Post*: pre-treatment and post-treatment PD patients.

The former, *H-Pre*, has been analysed in Fig. 6.9a. It can be argued that the first two input attributes (represented as blue directions 1 and 2 in the figure) are almost parallel to the first two axes, PC1 and PC2, while the rest is explained by the last principal component, PC3. This behaviour becomes obvious by zooming near the origin (see Fig. 6.9b). Here, it is tangible that the PC1 and PC2 correspond to the X and Y pen positions, i.e. the first two attributes; indeed, PC1 and PC2 draw exactly the original handwriting "*La casa de Barcelona es preciosa*". Albeit the maximum variance direction, i.e. PC1, maps the X component (writing from left to right), the most meaningful attribute is the Y pen position because, as visible in Fig. 6.9, the healthy and the pre-treatment clusters are clearly separated along this direction.

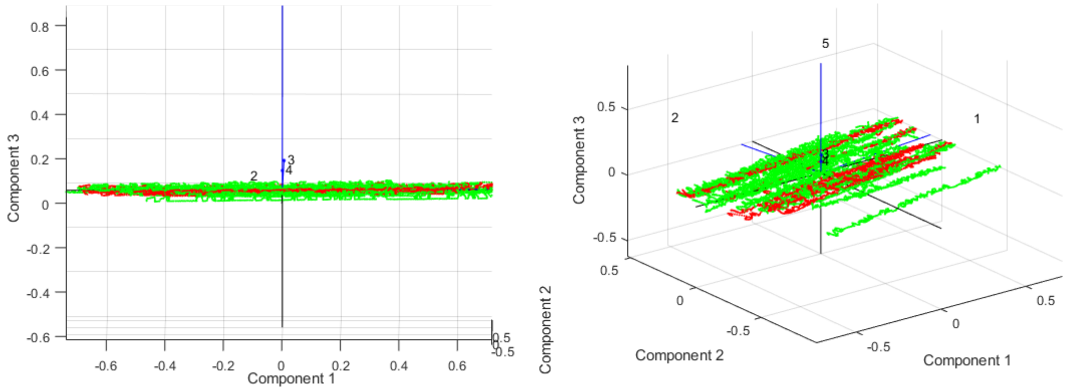


(a) whole biplot

(b) zoom near the origin

Figure 6.9: Parkinson's handwriting. Biplot on *H-Pre* dataset: healthy (red), pre-treatment (green). Blue lines are the input directions, whose corresponding feature is given by the numerical label. The whole biplot is on the left, a zoom near the origin is on the right.

The *H-Post* biplot is displayed in Fig. 6.10a. The same considerations made for the first two input attributes can be repeated; conversely, in this case, the remaining three attributes can be exploited for discriminating between the clusters. Indeed, the Z-view (see Fig. 6.10b) proves subsets are linearly separated along the third principal component.



(a) whole biplot

(b) Z-view

Figure 6.10: Parkinson's handwriting. Biplot on *H-Post* dataset: healthy (red), post-treatment (green). Blue lines are the input directions, whose corresponding feature is given by the numerical label. The whole biplot is on the left, the Z-view is on the right.

The biplot related to the last subset, *Pre-Post*, is shown in Fig. 6.11. As in the prior cases, the first attribute (the X pen position), fully explains the clusters, while the Y pen position distinguishes between the subclusters. The difference with the former biplots is that their directions are moderately rotated w.r.t. PC1 and PC2; it may result from the absence of the healthy cluster. The remaining attributes are pointless because the manifold is almost a hyperplane.

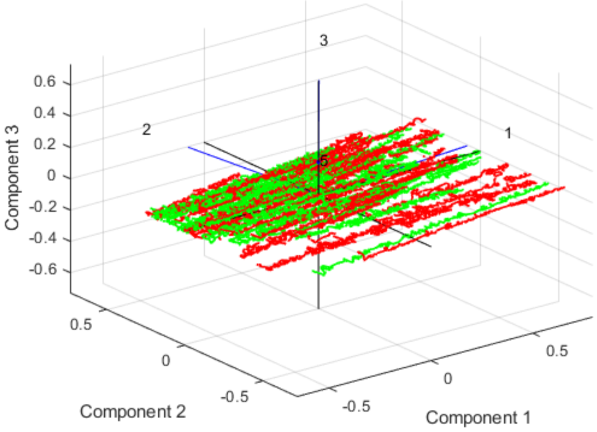


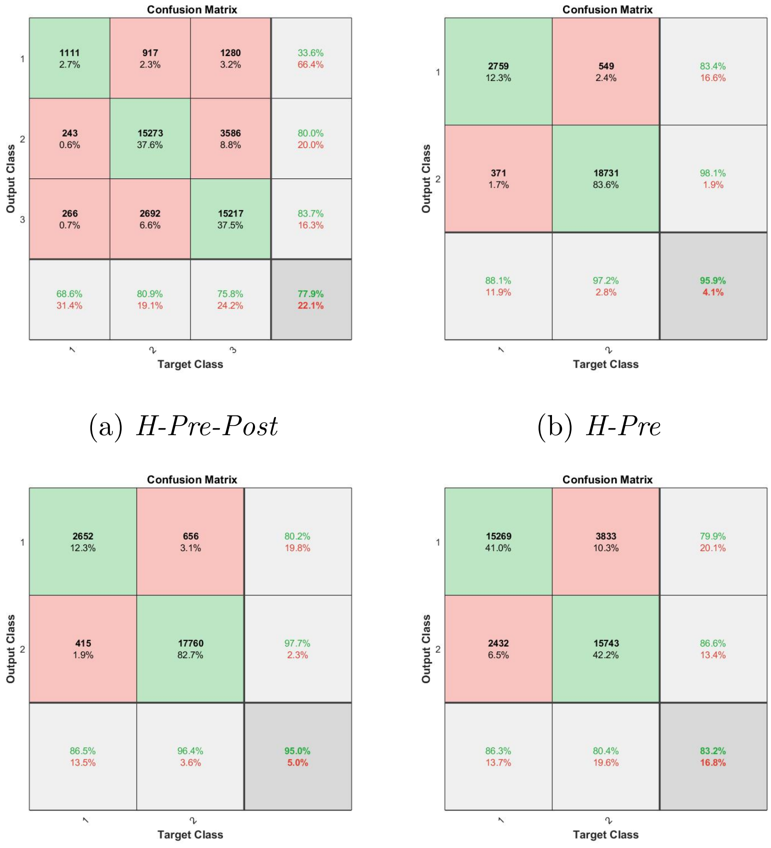
Figure 6.11: Parkinson's handwriting. Biplot on *Pre-Post* dataset: pre-treatment (red), post-treatment (green). Blue lines are the input directions, whose corresponding feature is given by the numerical label.

Finally, it can be argued that the selected attributes only approximate the input manifold. PC1 and PC2 barely coincide with the X and Y pen positions, which is natural because most variance in writing is embedded in these two directions. Therefore, the most significative information should stem from the other three PCs, but, as seen in Figs 6.8 and 6.10, they do not sufficiently separate clusters. The Y pen position ability to separate clusters may be related to vertical micrographia and to the interphalangeal and metacarpophalangeal joints activation. However, this statement needs much more data than the available dataset for being assessed with certainty.

6.2.3 Neural classification

A comparative study on the classification performance of a shallow neural network has been carried out to assess the discriminative capabilities of the input attributes. The shallow neural network has been chosen because it is tailored for pattern recognition [4]. The hidden layer is made of twenty neurons, and the output units are associated with soft-max activation functions [4]. Due to the cross-entropy error function, the network outputs the membership probability for the same input classes: healthy, pre-treatment, post-treatment. The input layer is mapped one-to-one to the input features; thus, it is made of five units. The shallow network has been trained, by means of the Scaled Conjugated Gradient algorithm [4], both on the whole database (three output units) and on the three subsets (two output units) previously defined; then, from each one of these training sets (TSs), fifteen statistical features, based on the recording temporal behaviour, have been extracted and fed to other shallow networks for assessing their classification performances. Due to the lack of clinical information, in all the simulations, the labels (healthy, pre-treatment, post-treatment) were exploited for splitting the input dataset into balanced training, validation and test subsets w.r.t. the labels. In all the experiments, 70% of the TS has been used for training and the rest was divided in equal parts, i.e. 15% each, between test and validation sets.

In the first simulation data are drawn directly from the *H-Pre-Post* dataset. Each sample is labelled w.r.t its class: healthy, pre-treatment, post-treatment. The resulting TS is a five column matrix with as many rows as the whole number of samples ($\approx$ 244K). The confusion matrix of the testing phase is shown in Fig. 6.12a; the overall accuracy is 77.9%. The second simulation deals with the *H-Pre* subset; therefore, the TS has only two labels (healthy and pre-treatment) and around 134K samples. Fig. 6.12b yields the results; the overall accuracy is 95.9%. This classification is particularly accurate, which is evident because healthy and sick people have significantly different motor control and, thus, handwriting. The experimental setup for the *H-Post* dataset ($\approx$ 129K examples) is the same as before: two output classes (healthy and post). Compared to the previous experiment, the overall test performance decrease to 95.0% (see Fig. 6.12c). This is not a bad result; actually, it suggests that, after drug administration, some participants have recovered enough, w.r.t. their motor control, to be misclassified with the healthy ones. The last experiment regards the *Pre-Post* subset, composed of around 224K samples, which, of course, are labelled only with two classes: pre-treatment and post-treatment. The classification (see Fig. 6.12d) is worse (only 83.2%) than the previous simulations. All the patients are affected by Parkinson's; therefore, their handwritings have similar characteristics, and these two classes are the tougher to be separated. Unfortunately, PD treatments are still not very effective; thus, motor control improvements are quite limited, even after drug administration, especially when the disease is, already, in an advanced stage. Another explanation is that, maybe, participants are still in early stages of Parkinson's, where levodopa effect on handwriting is negligible. Resuming, the healthy state is the easiest to discriminate, because it is related on very peculiar feature values; in this sense, it can be exploited as a baseline for determining if the post-treatment condition exhibits an improvement, i.e. post-treatment features get closer to the healthy ones.



(a) *H-Pre-Post*

(b) *H-Pre*

(c) *H-Post*

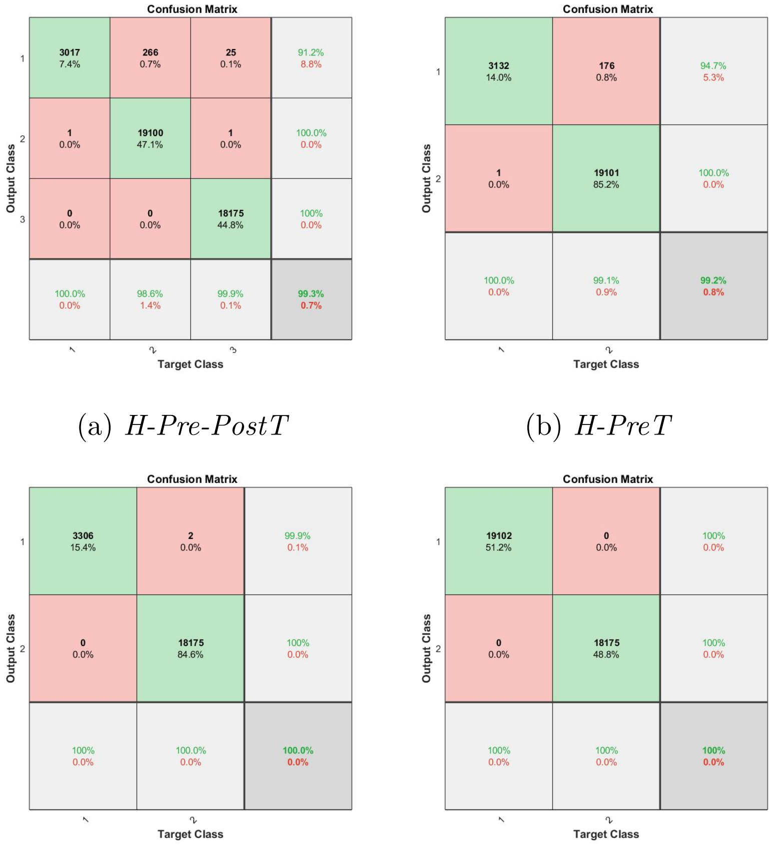
(d) *Pre-Post*

Figure 6.12: Parkinson's handwriting. Shallow neural network: raw data test set confusion matrix.

The data manifold analysis presented in Sec. 6.2.2 and the previous study have demonstrated that the raw attributes were not sufficient to separate the three subsets. As consequence, a better discriminating group of features has been proposed. The purpose is the exploitation of their temporal content; in this sense, from each record of the four datasets (*H-Pre-Post*, *H-Pre*, *H-Post*, *Pre-Post*) fifteen features have been extracted: mean, max value, root mean square (RMS), square root mean (SRM), standard deviation, variance, shape factor (with RMS), shape factor (with SRM), crest factor, latitude factor, impulse factor, skewness, kurtosis, normalized 5th central moment, normalized 6th central moment. For each of the four new datasets (*H-Pre-PostT*, *H-PreT*, *H-PostT* and *Pre-PostT*), the classification assessment w.r.t. the input features has been performed again. For all the simulations the shallow networks are equipped with a forty neuron hidden layer and a fifteen unit input layer. The remaining setup is the same as the raw experiments.

The first simulation deals with the *H-Pre-PostT* dataset. Each sample is labelled according to its class: healthy, pre-treatment, post-treatment. The resulting set is a fifteen column matrix with as many rows as the number of samples ($\approx$ 244K), which, of course, is the same as the corresponding raw case. The test set confusion matrix is shown in Fig. 6.13a; the overall accuracy is 99.3%, i.e. an 27% increase compared to *H-Pre-Post*. In the second simulation samples are extracted from the *H-PreT* subset; as usual, the TS has only two labels (healthy and pre-treatment). Despite this classification (see Fig. 6.13b) is more accurate (99.2%) than its corresponding raw case, the overall accuracy is not significantly improved (3%); the considerations of the raw case hold. In the third simulation, the network has been trained using the *H-PostT* subset. The overall test performance, shown in Fig. 6.13c, reaches its maximum (100%) with an increase of 5.3%. It must be stressed that, in this case, the network does not misclassify between recovered patients and the healthy subjects; this may suggest that even when the handwritings are closer to normality, the temporal evolution differs in such a way that the network can discriminate the healthy case. The last experiment regards the *Pre-PostT* subset. Fig. 6.13d yields the classification results for the test set. The *Pre-Post* dataset, which was difficult to cluster (83.2% of accuracy), is now perfectly understood (100% of accuracy) by the classifier by means of the two different classes, *Pre* and *Post*. The performance increase is more than the 20%.

Some final considerations can be deduced from Table 6.4. The input layer employs fewer neurons in the raw case; because the neural network is fully connected, the use of temporal features implies more training epochs. However, the final training error is several orders of magnitude smaller than in the raw case. This consideration is strengthened by the classification rates and demonstrates the temporal model better represents the manifold, i.e. the PD handwriting. This justifies the medical consideration about the handwriting temporal evolution relevance.



(a) *H-Pre-PostT*

(b) *H-PreT*

(c) *H-PostT*

(d) *Pre-PostT*

Figure 6.13: Parkinson's handwriting. Shallow neural network: temporal feature test set confusion matrix.

Table 6.4: Shallow classification perfomances

	# Epochs	**Final Error**	**% Training**	**% Test**
H-Pre-Post	990	0.18	77.8	77.9
H-Pre-PostT	1000	0.01	99.3	99.3
H-Pre	629	0.57	96.0	95.9
H-PreT	831	0.013	99.3	99.2
H-Post	497	0.07	94.8	95
H-PostT	1000	0.0008	100	100
Pre-Post	972	0.175	83.5	83.2
Pre-PostT	1000	0.0004	100	100

Chapter 7

Interpreting Deep Learning

Deep learning is able to automatically extract features from data and provide good classification outcomes, but it has to be treated as a black box and the results cannot be interpreted in a theoretical framework. On the other side, classical neural networks, such as shallow ones, need a human-based feature engineering phase prior to their training; due to the network simplicity (compared to deep models), it is possible to interpret its outcomes and to relate them with the input features. The proposed approach, see Fig. 7.1, combines these two techniques for exploiting their advantages. First, deep learning is trained to reach a good classification performance P; then, P is used as a benchmark to evaluate and guide classical neural network training and feature selection (orange arrow). Once the model reaches a satisfactory performance, the features F extracted in the engineering phase are sought in the deep learning model by means of a correlation analysis between F and the deep network layers (blue arrow). In this sense, by understanding which are the features automatically extracted by the deep technique, it would be possible to give an interpretation, i.e. an explanation, of its results.

In order to assess the validity of the proposed approach, an application to electrocardiogram (ECG) analysis is presented in the following.

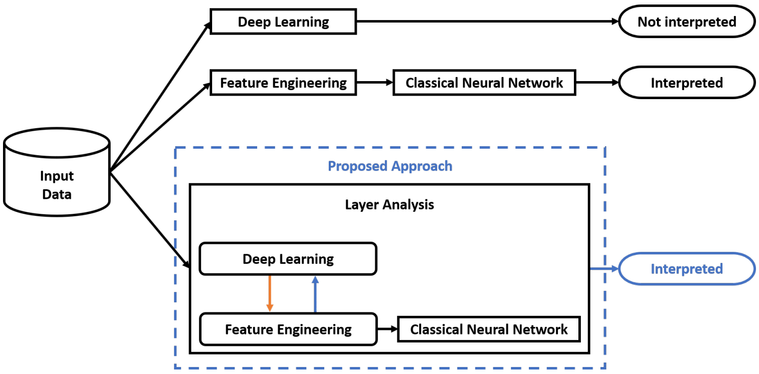


Figure 7.1: Proposed method for understanding deep learning

7.0.1 The ecg case study

One of the most prominent effects of the technological progress in the last decade is the pervasive diffusion of IoT and wearable devices; their ubiquity make them the perfect candidates for medical applications such as disease monitoring and prevention. Nevertheless, the tremendous amount of medical data yielded by such devices is completely useless if it is not inspected by a medical expert. In this sense, there is still the need of systems capable of automatically analysing their recordings.

The standard medical approach for heart monitoring is measuring its electrical activity using an electrocardiograph device, which records the various phases of the heart muscular activation. A healthy ECG [167], shown in Fig. 7.2, presents six fiducial points (P, Q, R, S, T, U), which yields the PR, QRS, ST, U segments, relatives to the four principal stages of a cardiac cycle: isovolumic relaxation, inflow, isovolumic contraction, ejection. Irregularities in the ECG rhythm are called *arrhythmias* and signal an anomalous muscular activity, i.e. a possible disease.

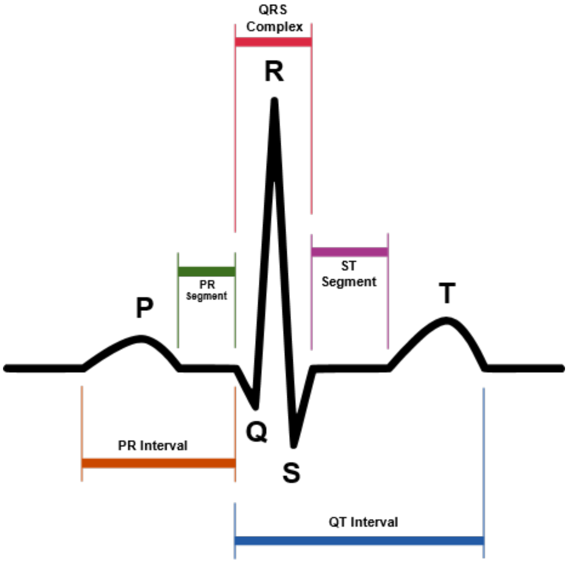


Figure 7.2: Example of an healthy ECG

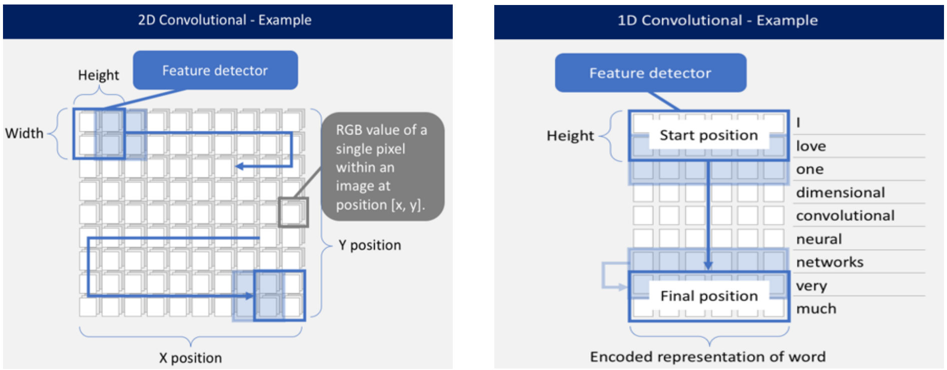
ECG classification has been tackled by means of many different strategies mostly based on feature extraction w.r.t. temporal or morphological properties; then, these features are exploited for the ECG classification [168]. The most famous technique for QRS-complex recognition is the Pam-Tompkins [169], where both morphological and temporal attributes are considered for detecting R-peaks. In [170], Support Vector Machines are used for the same goal. In [171], ventricular fibrillation and tachycardia are detected with a temporal analysis. Hidden Markov models are employed in [172], while fuzzy and artificial neural networks are exploited in [173] and [174, 175, 176], respectively. Noise removal and arrhythmia detection by means of adaptive filtering is proposed in [177], while [178] employs wavelet transformation and artificial neural network. A fuzzy K-NN is introduced in [179]. Atrial fibrillation (AF) recognition is addressed in [180, 181]. An extensive review can be found in [182].

Despite feature engineering yields interesting results, it assumes the chosen attributes are the most significative for efficiently classifying the input signal. Moreover, feature selection and engineering, i.e. choosing the best feature set, implies a deep knowledge both of the signal and the recording conditions, e.g. the environment or the device. In this sense, the approach has a limited applicability.

Convolutional Neural Networks

In the last decade, more and more techniques based on Convolutional Neural Networks (CNN) have been proposed in literature. CNNs are able to autonomously build a data representation (feature extraction) and to discover new hidden patterns in the input dataset. In other words, they perform automated feature engineering on data, i.e. they are antagonists to feature engineering techniques. Inspired by the human visual cortex, CNNs are composed of multiple layers, each of which is activated from specific patterns in the input data. These subsets are tiled to introduce region overlap, and the process is iterated layer by layer to obtain a high level abstraction of the original dataset as shown in Fig. 7.3a.

The main drawback of this architecture is the lack of control on the algorithm; indeed, due to the terrific amount of samples required during training and the exponential number of extracted features, most, if not all, deep neural networks are considered black boxes.



(a) 2-D CNN

(b) 1-D CNN

Figure 7.3: Convolution Neural Network: 2-D (left) vs 1-D (right)

A promising kind of convolutional neural networks is the 1-D CNN, which takes as input a single stream (a signal), e.g. ECG, and slides a kernel along it in search of particular patterns (see Fig. 7.3b). In literature there are already some heartbeat classification studies based on 1D-CNN [183]; for instance, in [184] and [185] CNNs are designed for the classification of 4 and 5 different arrhythmia, respectively, while in [186] 2-D CNNs, which are typically employed for image processing, and 1-D CNNs are compared for the classification of 5 heartbeat classes; in [187] a very deep 1-D CNN is proposed to distinguish heartbeats amongst 14 different classes. Finally, [188] proposes a biometric identification system based on CNNs.

The dataset

The MIT-BIH arrhythmia database [189, 190], yielded by PhysioNet [136], is one of the key references in ECG analysis. Each heartbeat, within each acquisition, is labelled; therefore, a supervised approach is quite straightforward. The database is well documented and it covers a great range of heart diseases. It is composed of 30 minutes ECG recordings from 48 subjects for the L2 and V1 leads [191], for a total amount of heartbeats equal to approximately $110K$, which have been manually labelled by two professional cardiologists into 16 different classes (see Table 7.1).

As a preprocessing, the over 31 million MIT-BIH samples were splitted in smaller segments for building the training set. To ensure at least one heartbeat is present in each TS sample, segments must range between 1 and 2 second length. The database sample frequency is 360 $samples/s$; thus, a segment size of 500 time-instants was selected. In addition, to augment the TS, an overlapping factor of 10% was employed. The TS was statistically normalized and each sample labelled accordingly. Finally, the TS was randomly splitted in training and validation datasets with a ratio of 90%/10%, respectively.

Table 7.1: MIT-BIH heartbeat label meaning

Label	Meaning
/	Paced beat
A	Atrial premature beat
E	Ventricular escape beat
F	Fusion of ventricular and normal beat
J	Nodal premature beat
L	Left bundle branch block beat
N	Normal beat
Q	Unclassifiable beat
R	Right bundle branch block beat
S	Supraventricular premature beat
V	Premature ventricular contraction
!	Ventricular flutter wave
a	Aberrated atrial premature beat
e	Atrial escape beat
f	Fusion of paced and normal beat
j	Nodal escape beat

7.1 1-D CNN for ECG classification

To test the arrhythmia classification performance of a 1-D CNN on the MIT-BIH dataset, several CNN configurations have been evaluated, where the amount of convolutional layers, the filter size (Ks) and quantity, and the dropout rate, have been varied to determine the best network [192]. Among the all tested possibilities configurations, only the four most significative ones (w.r.t. classification) are presented. Their confusion matrices have been analysed for studying the network behaviour on the different arrhythmia classes. In the following, the whole arrhythmia class set is employed whereas in other similar researches [185, 193], only the most meaningful ones are used.

The first network ($\approx 65K$ parameters), say Net1, is made of a first convolutional layer of 16 filters ($Ks = 32$), followed by a max pooling layer and a softmax classifier. The training and testing accuracies are equal to 92% and 91%, respectively. The second experiment uses a more structured network ($\approx 257K$ parameters), say Net2: it has a first convolutional layer of 64 filters ($Ks = 8$), followed by a max pooling layer and a softmax classifier. Having a higher amount of convolutional layer has increased the accuracy to 96% and 94% for training and testing, respectively. The third network (Net3) is deeper. It is composed of three convolutional layers with a growing number of filters - 64, 128, 256 - whose kernel size decreases (32, 16, 8). A pooling layer follows each convolutional layer. Finally, there is a 128 neuron fully-connected (FC) layer and a softmax classifier. Albeit the parameters doubled (533K), the performance remained the same.

To increase the classification performance, the last experiment deals with a much more complex architecture ($\approx 1200K$ parameters), called Net4, shown in Fig. 7.4: it has five convolutional layers, 2 FC layers and 1 FC softmax classifier. Indeed, it reaches the best performance w.r.t. arrhythmia classification: 98% for training and 95% for testing.

Table 7.2 summarizes the accuracies of the four configurations.

Table 7.2: Classification accuracy for the best 1-D CNN architectures

	Training Accuracy	Test Accuracy	Total Parameters
Net 1	92 %	91 %	65,056
Net 2	96 %	94 %	257,104
Net 3	96 %	94 %	533,072
Net 4	98 %	95 %	1,266,768

7.1.1 Results Analysis

Table 7.2 proves Net4 is the best performing architecture. Although several attempts were performed, the accuracy did not improve further. As a consequence, the Net4 confusion matrix (see Fig. 7.5) has been examined to analyse the network behaviour w.r.t. the different kind of heartbeats.

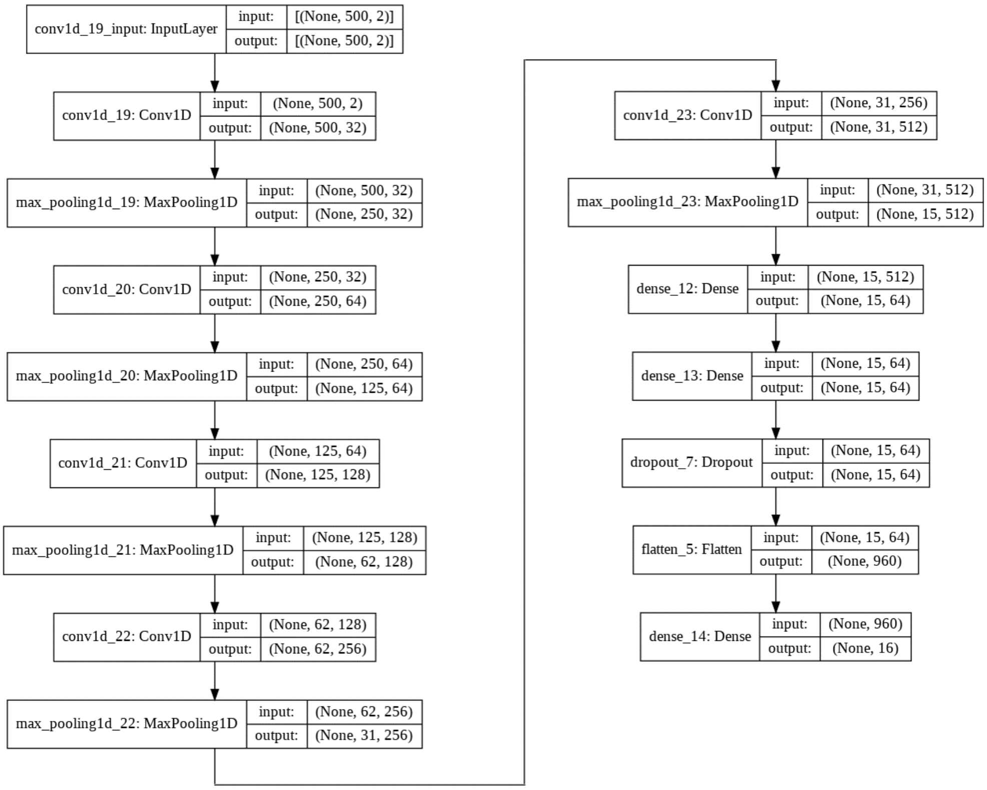


Figure 7.4: 1-D CNN: Net4 architecture

Class F is occasionally confused with class N (Normal beat) or class V (Premature ventricular contraction); because F is the fusion of ventricular and normal beat classes, if the segment window is not aligned exactly with the whole series of heartbeats, these classes are quite undistinguishable. Expanding the segmentation window would fix this issue, but it will also affect the recognition of the other classes. Looking at class e (Atrial escape beat), it is possible to note that it is very badly performing. Indeed, it is the least represented class in the whole dataset; thus, the network does not have sufficient information for learning it. However, because it is very similar to class A (Atrial premature beat), the system partially classifies as it. Class S (Supraventricular premature beat) is completely misinterpreted as class V (Premature ventricular contraction). The last consideration regards class Q (Unclassifiable beat), which is a special case because, by definition, does not have a specific pattern to be recognized. It is a heartbeat that even professional cardiologists were not able to classify. It is curious that the network mostly classified (49%) those heartbeats as N. Although, in principle, it is a wrong classification, it cannot be excluded, in advance, that those samples were wrongly classified as Q; maybe, the network has found some new specific pattern for class N.

Finally, the factor that affected the most the classification performance was the high class unbalance. Indeed, class N amounts to 40% of samples, while the remaining classes just have a very small amount of heartbeats; as an order of magnitude, class e only counted less than 2% of the dataset. Of course, this under-representation of most part of classes deeply affects the results.

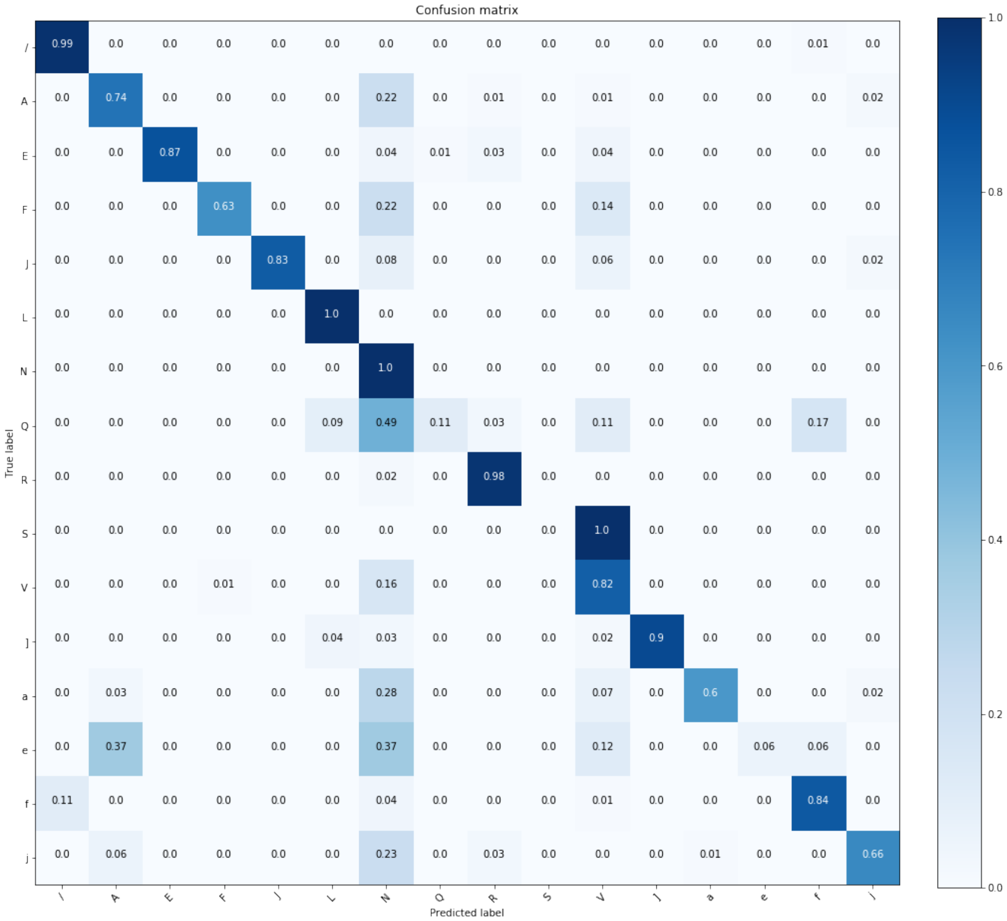


Figure 7.5: 1-D CNN: Net4 confusion matrix

5 class 1-D CNN

The previous considerations on 1-D CNN performance led to the conclusions that it could be interesting to rebalance the dataset. To this purpose, the four most represented classes are kept, i.e. N, R, V and A, while the remaining twelve are grouped into a fictitious class *others* (O). Because of this approximation, it has been trained a novel simpler deep model [194], say *Net5*, which is made of only four CNN layers followed by a softmax classifier, as shown in Fig. 7.6. The corresponding test set confusion matrix (see Fig. 7.7) shows the model benefits from the label sub-grouping; indeed, its accuracy increases up to 99.6%.

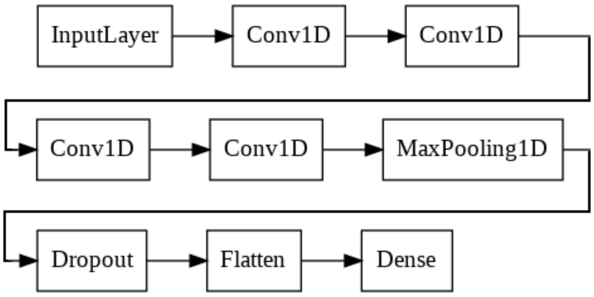
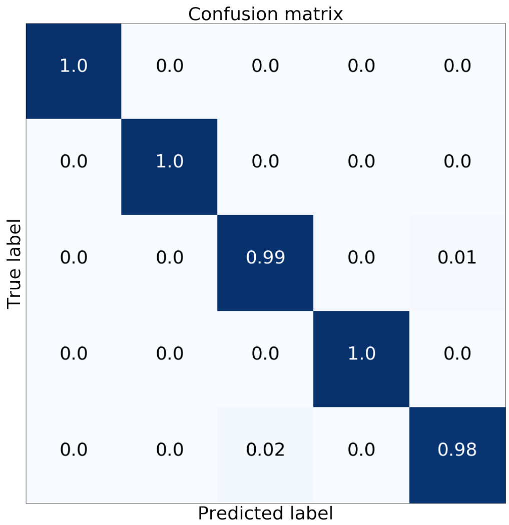


Figure 7.6: 1-D CNN: Net5 architecture



1.0	0.0	0.0	0.0	0.0
0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.99	0.0	0.01
0.0	0.0	0.0	1.0	0.0
0.0	0.0	0.02	0.0	0.98

Figure 7.7: 1-D CNN: Net5 confusion matrix

7.2 Feature extraction and analysis

An alternative approach for ECG automatic classification is based on feature engineering [195]. As explained in [196], this approach requires cardiologists to extrapolate the relevant information from the recordings and, then, to use it for training the neural system; for instance, in [197], several attributes were abstracted from surgeon hand motion recording to assess his ability during training. Actually, misclassifications are often related to a wrong feature engineering phase [198, 199, 200]. With regard to the extraction approach, features can be grouped into several categories such as temporal-based and eigenvectors-based. In the former, the temporal evolution of the ECG signal (e.g. R-R variance) is considered as significative [201, 202], while eigenvector techniques estimate the signal frequencies from noise-corrupted recordings by means of an eigen-decomposition of the related correlation matrix; an application to ECG classification of Pisarenko [6] and MUSIC [7] - the two most relevant eigenvector algorithms - can be found in [203, 204, 205].

Supervised learning arrhythmia classification has been deeper analysed by studying the performance as input features change. As explained before, deep learning automatically extracts its features from the raw data; therefore, it does not make sense to feed it with human-engineered attributes. On the contrary, a simpler MLP can be exploited to this purpose [206]; six different training sets have been employed: ECG raw data, temporal attributes, eigenvector features and the corresponding CCA projections. The input layer size is always equal to the number of features of the relative TS; as a rule of thumb, the hidden layer dimension is at least the double of the input one. In order to balance the less powerful architecture w.r.t. deep learning, and given the considerations made in Sec. 7.1.1, only the four most represented classes are used, i.e. N, R, V and A, while the remaining twelve are grouped into the fictitious class *others* (O); the MLP output layer has five units equipped with the soft-max activation function, which yield the class membership probability due to the usage of the cross-entropy error function. The first step is the intrinsic dimensionality analysis of each dataset, which, together with the study of the confusion matrices, can be used for determining the most meaningful subset of features for classification. Two-thirds of samples are used for training, while the remaining third for testing.

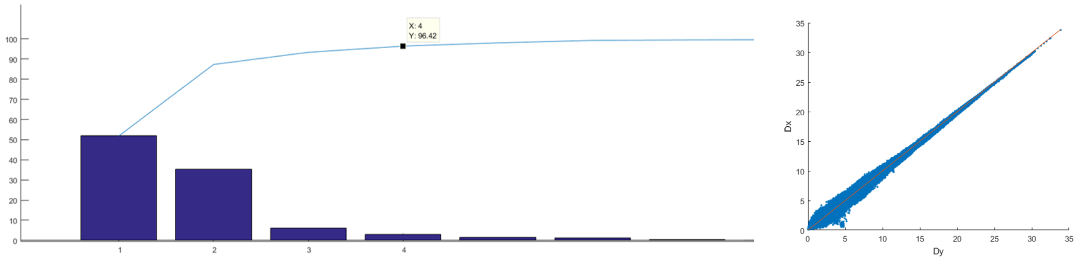
7.2.1 MLP - ECG raw data

In the first experiment raw data are fed to the MLP; in this scenario, samples are considered as meaningful in themselves (no feature extraction is applied). For each sample of the training set, i.e. a QRS complex, a symmetric 41-time instants window w.r.t. the R-peak has been used to parse data. In addition, the R-R time, i.e. the time among two consecutive beats, has been appended as last attribute of this initial set. Consequently, the input set is made of forty-two variables and as many rows as the amount of QRS complexes.

The PCA manifold analysis suggests the intrinsic dimensionality is probably four (96.42% explained); the corresponding Pareto chart is shown in Fig. 7.8a. To take into account the data non-linearity, CCA is performed ($\lambda = 70$, $epochs = 10$) for projecting data to a 4-D subspace. The corresponding *dy-dx* diagram, see Fig. 7.8b, is concentrated around the bisector, which demonstrates that in 4-D the manifold becomes nearly a hyperplane.

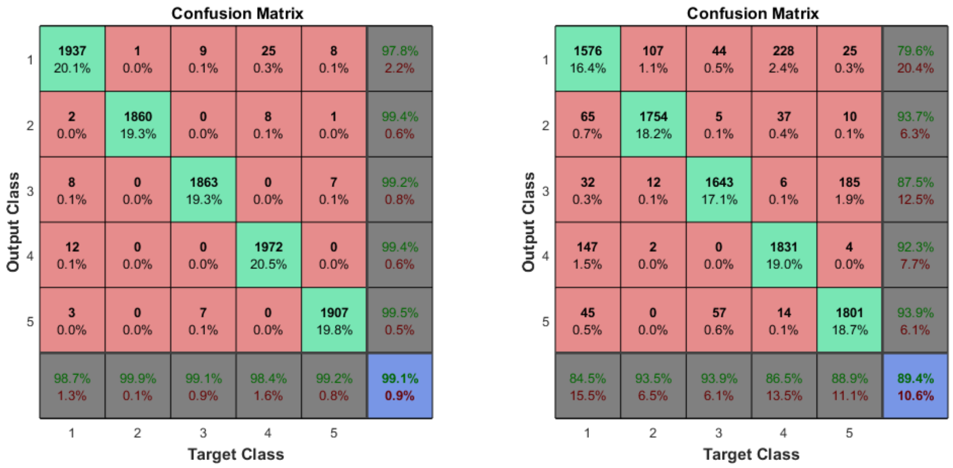
The MLP used for classifying the raw data has 42 input units and 100 neurons in the hidden layer; the associated test set confusion matrix is shown in Fig. 7.9a.

This classification is very accurate (99.1%) although it needs 42 features; indeed, because there is no feature creation, the algorithm becomes time-consuming. If data resulting from the above CCA are fed as input to an MLP with four input neurons and a single twenty-unit hidden layer, the overall test performance lowers to 89.4% (see Fig. 7.9b), i.e. a 9.78% accuracy loss.



(a) Pareto chart (b) *dy-dx* diagram

Figure 7.8: MLP on raw data: intrinsic dimensionality estimation.



	1	2	3	4	5	
1	1937 20.1%	1 0.0%	9 0.1%	25 0.3%	8 0.1%	97.8% 2.2%
2	2 0.0%	1860 19.3%	0 0.0%	8 0.1%	1 0.0%	99.4% 0.6%
3	8 0.1%	0 0.0%	1863 19.3%	0 0.0%	7 0.1%	99.2% 0.8%
4	12 0.1%	0 0.0%	0 0.0%	1972 20.5%	0 0.0%	99.4% 0.6%
5	3 0.0%	0 0.0%	7 0.1%	0 0.0%	1907 19.8%	99.5% 0.5%
	98.7% 1.3%	99.9% 0.1%	99.1% 0.9%	98.4% 1.6%	99.2% 0.8%	99.1% 0.9%

	1	2	3	4	5	
1	1576 16.4%	107 1.1%	44 0.5%	228 2.4%	25 0.3%	79.6% 20.4%
2	65 0.7%	1754 18.2%	5 0.1%	37 0.4%	10 0.1%	93.7% 6.3%
3	32 0.3%	12 0.1%	1643 17.1%	6 0.1%	185 1.9%	87.5% 12.5%
4	147 1.5%	2 0.0%	0 0.0%	1831 19.0%	4 0.0%	92.3% 7.7%
5	45 0.5%	0 0.0%	57 0.6%	14 0.1%	1801 18.7%	93.9% 6.1%
	84.5% 15.5%	93.5% 6.5%	93.9% 6.1%	86.5% 13.5%	88.9% 11.1%	89.4% 10.6%

(a) Original space (b) CCA projected space

Figure 7.9: MLP on raw data: test set confusion matrix

7.2.2 MLP - Temporal attributes

In the second experiment fifteen statistical attributes are extracted from each raw record (see Table 7.3); the resulting TS is z-scored to avoid that some attributes mask the information embedded in small range features.

Fig. 7.10a presents the linear analysis: the intrinsic dimensionality is probably six (96.68% explained). To study the dataset non-linearity, the attributes are

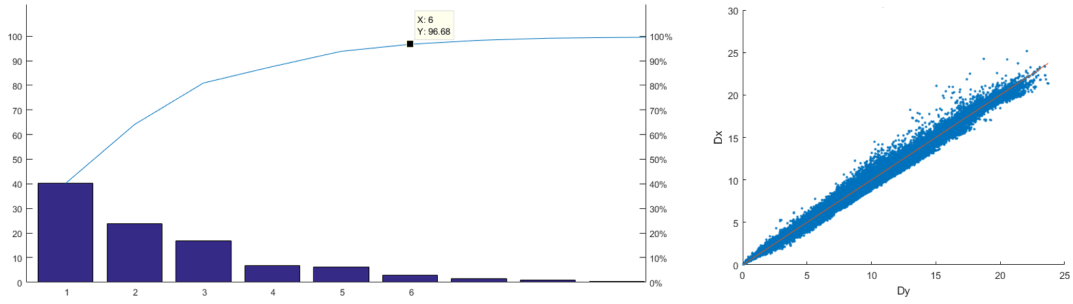
Table 7.3: Temporal attribute extracted from the ECG raw data

#	Attribute	
F1	Mean	$\bar{x} = \sum \frac{x_i}{N}$
F2	Max value	$max\,(x)$
F3	Root Mean Square (RMS)	$\sqrt{\sum \frac{x_i^2}{N}}$
F4	Square Mean Root (SMR)	$\left(\sum \frac{\sqrt{\lvert x_i \rvert}}{N}\right)^2$
F5	Standard deviation	$\sqrt{\frac{1}{N-1}\sum (x_i - \bar{x})^2}$
F6	Variance	$F5^2$
F7	Shape factor (using RMS)	$\frac{F3}{\sum \frac{\lvert x_i \rvert}{N}}$
F8	Shape factor (using SMR)	$\frac{F4}{\sum \frac{\lvert x_i \rvert}{N}}$
F9	Crest factor	$\frac{F2}{F3}$
F10	Latitude factor	$\frac{F2}{F4}$
F11	Impulse factor	$\frac{F3}{\sum \frac{\lvert x_i \rvert}{N}}$
F12	Skewness	$\frac{\frac{1}{N}\sum (x_i - \bar{x})^3}{\left[\frac{1}{N-1}\sum (x_i - \bar{x})^2\right]^{\frac{3}{2}}}$
F13	Kurtosis	$\frac{\frac{1}{N}\sum (x_i - \bar{x})^4}{\left[\frac{1}{N-1}\sum (x_i - \bar{x})^2\right]^{2}}$
F14	Normalized 5th central moment	$\frac{\frac{1}{N}\sum (x_i - \bar{x})^5}{\left[\frac{1}{N-1}\sum (x_i - \bar{x})^2\right]^{\frac{5}{2}}}$
F15	Normalized 6th central moment	$\frac{\frac{1}{N}\sum (x_i - \bar{x})^6}{\left[\frac{1}{N-1}\sum (x_i - \bar{x})^2\right]^{3}}$

Note: N is the number of the elements of the vector x, whilst x_i is the i^{th} element.

projected to a 6-D subspace by means of CCA ($\lambda = 70$, $epochs = 10$). The corresponding *dy-dx* diagram, see Fig. 7.10b, is wider w.r.t. the bisector than in the raw case. Because it is thicker for greater distances, the manifold is only locally linear, i.e. short distances are preserved in the subspace.

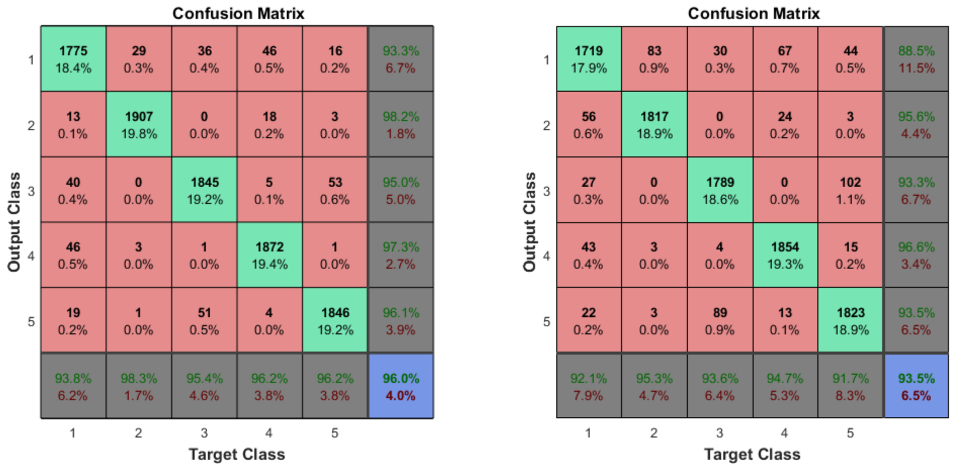
As with the previous experiment, the R-R time, i.e. the time between two consecutive R-peaks, has been appended as last attribute, yielding a 16-unit input layer; here, the hidden layer is made of 40 neurons. The corresponding test set confusion matrix is displayed in Fig. 7.11a. The overall accuracy is 96.0%. Albeit the classification is worsened w.r.t. corresponding raw data case, this approach requires around one-third of input variables (42 → 16). If the CCA six projected features are fed to an MLP with six input units and one twenty-neuron hidden layer, the overall test performance lowers to 93.5% (see Fig. 7.11b), i.e. 2.6% loss.



(a) Pareto chart

(b) *dy*-*dx* diagram

Figure 7.10: MLP on temporal attributes: intrinsic dimensionality estimation.



	1	2	3	4	5	
1	1775 18.4%	29 0.3%	36 0.4%	46 0.5%	16 0.2%	93.3% 6.7%
2	13 0.1%	1907 19.8%	0 0.0%	18 0.2%	3 0.0%	98.2% 1.8%
3	40 0.4%	0 0.0%	1845 19.2%	5 0.1%	53 0.6%	95.0% 5.0%
4	46 0.5%	3 0.0%	1 0.0%	1872 19.4%	1 0.0%	97.3% 2.7%
5	19 0.2%	1 0.0%	51 0.5%	4 0.0%	1846 19.2%	96.1% 3.9%
	93.8% 6.2%	98.3% 1.7%	95.4% 4.6%	96.2% 3.8%	96.2% 3.8%	96.0% 4.0%

	1	2	3	4	5	
1	1719 17.9%	83 0.9%	30 0.3%	67 0.7%	44 0.5%	88.5% 11.5%
2	56 0.6%	1817 18.9%	0 0.0%	24 0.2%	3 0.0%	95.6% 4.4%
3	27 0.3%	0 0.0%	1789 18.6%	0 0.0%	102 1.1%	93.3% 6.7%
4	43 0.4%	3 0.0%	4 0.0%	1854 19.3%	15 0.2%	96.6% 3.4%
5	22 0.2%	3 0.0%	89 0.9%	13 0.1%	1823 18.9%	93.5% 6.5%
	92.1% 7.9%	95.3% 4.7%	93.6% 6.4%	94.7% 5.3%	91.7% 8.3%	93.5% 6.5%

(a) Original space

(b) CCA projected space

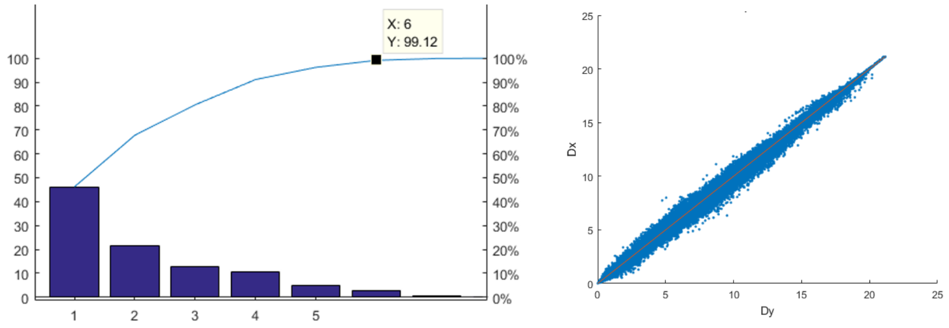
Figure 7.11: MLP on temporal attributes: test set confusion matrix

7.2.3 MLP - Eigenvector features

The last dataset, normalized with z-score, is composed of eight features extracted from each raw record using the MUSIC algorithm. Many MUSIC sub-space dimensions have been compared to see how the classification is affected by this parameter. The best performance has been reached when the subspace dimensionality is equal to five. As usual, the R-R time has been appended.

The linear analysis (see Fig. 7.12a) suggests the intrinsic dimensionality is probably six (99.12% explained). The *dy-dx* diagram for a 6-D projection ($\lambda = 30$, $epochs = 10$) is presented in Fig. 7.12b; it resembles the temporal feature case, but it is thick also for smaller distances, which means the manifold is locally less linear.

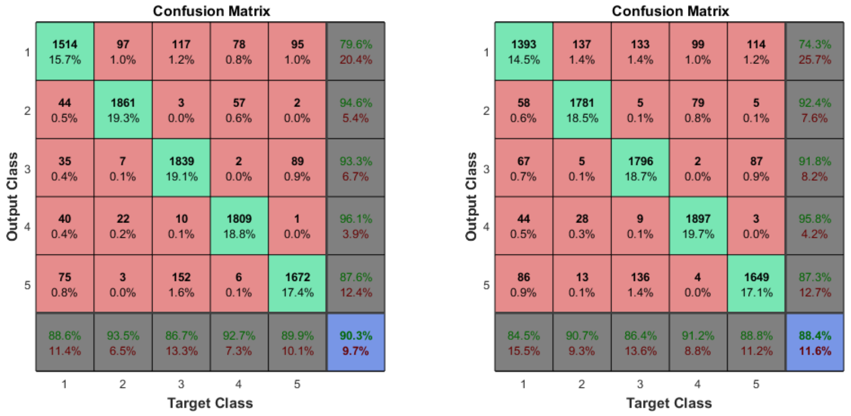
The selected MLP has nine input units and 40 neurons in the hidden layer. The corresponding test set confusion matrix is shown in Fig. 7.13a with an overall accuracy of 90.3%. This classification is the worst, but still accurate; however, this technique uses the smallest amount of attributes: from 42 to 9, that is almost 79% reduction. If the CCA six projected features are fed to an MLP with six input units and one twenty-neuron hidden layer, the overall performance lowers to 88.4% (see Fig. 7.13b), i.e. 2.1% accuracy loss.



(a) Pareto chart

(b) *dy-dx* diagram

Figure 7.12: MLP on eigenvector features: intrinsic dimensionality estimation.



Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	
1	1514 15.7%	97 1.0%	117 1.2%	78 0.8%	95 1.0%	79.6% 20.4%
2	44 0.5%	1861 19.3%	3 0.0%	57 0.6%	2 0.0%	94.6% 5.4%
3	35 0.4%	7 0.1%	1839 19.1%	2 0.0%	89 0.9%	93.3% 6.7%
4	40 0.4%	22 0.2%	10 0.1%	1809 18.8%	1 0.0%	96.1% 3.9%
5	75 0.8%	3 0.0%	152 1.6%	6 0.1%	1672 17.4%	87.6% 12.4%
	88.6% 11.4%	93.5% 6.5%	86.7% 13.3%	92.7% 7.3%	89.9% 10.1%	90.3% 9.7%

Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	
1	1393 14.5%	137 1.4%	133 1.4%	99 1.0%	114 1.2%	74.3% 25.7%
2	58 0.6%	1781 18.5%	5 0.1%	79 0.8%	5 0.1%	92.4% 7.6%
3	67 0.7%	5 0.1%	1796 18.7%	2 0.0%	87 0.9%	91.8% 8.2%
4	44 0.5%	28 0.3%	9 0.1%	1897 19.7%	3 0.0%	95.8% 4.2%
5	86 0.9%	13 0.1%	136 1.4%	4 0.0%	1649 17.1%	87.3% 12.7%
	84.5% 15.5%	90.7% 9.3%	86.4% 13.6%	91.2% 8.8%	88.8% 11.2%	88.4% 11.6%

(a) Original space

(b) CCA projected space

Figure 7.13: MLP on eigenvector features: test set confusion matrix

7.2.4 MLP classification analysis

All the experiments exhibit a trade-off between the smallest amount of attribute and data linearity, which is even more evident in case of data projection. Table 7.4 summarizes the classification performance of the six MLPs.

Table 7.4: MLP test set accuracy

	Original Space (# Features)	**Reduced Space (# Features)**
ECG Raw Data	99.1 (42)	89.4 (4)
Temporal Features	**96.0 (16)**	93.5 (6)
Eigenvector Features	90.3 (9)	88.4 (6)

The raw data belong to a quasi-linear manifold in a 4-D space. Despite this simple topology, the greatest amount of attributes are needed (42). This analysis is confirmed after the CCA non-linear reduction; despite the subspace has size equal to the intrinsic dimensionality, the worst decrease in accuracy (9.78%) is observed.

The feature extraction approach, either temporal or eigenvector based, yields a significative reduction in the number of variables at the expense of a loss of linearity. The temporal attributes manifold is only locally linear. The eigenvector features lie in a similar manifold, but the linearity persists only for smaller neighbourhoods.

It is worth to be noticed that the temporal technique accuracy is similar to the raw case but uses only sixteen features, i.e. a 61.9% reduction, for a loss of only 2.9% of the overall accuracy; this consideration holds also for the reduced space, where the technique needs the minimum amount of features (6) w.r.t. the accuracy (loss of 5.6%). The same observations hold for the eigenvector technique but its accuracy is worse.

Finally, it can be concluded that the best trade-off in terms of accuracy and network simplicity is given by the temporal approach. On the other end, the higher amount of features of the raw case reaches the best accuracy, but the manifold simplicity prevents any meaningful dimensionality reduction.

7.3 CNN and MLP cross correlation analysis

In this chapter, the ECG classification problem, w.r.t. different kind of arrhythmia, has been tackled first with a deep 1-D CNN architecture, trained on the whole set of labels (see Sec. 7.1), and, then, with a simpler MLP, where lot of labels have been collapsed into a single one. The former is a very powerful tool and, so, it is able to deal with the whole label set even if some limits have been already shown (see Sec. 7.1.1); its accuracy can be thought as a benchmark for arrhythmia classification. However, it is impossible to deeply analyse its behaviour because of the complex architecture and the huge amount of parameters. On the other side, using the simpler MLP (see Sec. 7.2) allows to perform a deeper analysis on how the classification accuracy is influenced by the different input features; however, this characteristic requires a simpler input structure, i.e. few labels. Actually, the input space topological analysis done with the MLP can help to expand the 1-D CCN performance analysis of Sec. 7.1.1.

The proposed approach provides an explanation of the deep model behaviour by comparing it with the more traditional feature engineering technique. Indeed, the MLP analysis has shown how the temporal attributes (see Table 7.3) are quite promising for ECG classification; here, the same attributes are sought within the features automatically extracted by the deep model, by means of a cross-correlation analysis for similarity assessing. By understanding if the 1-D CNN inner representation exploits somehow the same handmade temporal features, it will be possible to determine if these attributes are really useful for ECG arrhythmia classification.

7.3.1 Similarity metric

Given a pair of random variables X and Y, whose probability density functions (PDFs) are, respectively, f_X and f_Y, the PDF of their difference $d = X - Y$ is known as *cross-correlation* [207] and it is given by:

$$f_d = f_X \star f_Y \tag{7.1}$$

The f_d can be seen as a metric for determining how much X and Y are similar. In a multivariate space, it becomes:

$$R_{\mathbf{XY}} := \mathbb{E}[\mathbf{XY}^T] \tag{7.2}$$

where $\mathbf{X} = (X_1, \ldots, X_m)^T$ and $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$, and $R_{\mathbf{XY}}$ is the cross-correlation matrix. For simplifying the similarity interpretation, (7.2) is replaced by its normalized version:

$$\rho_{\mathbf{XY}} = \frac{\operatorname{cov}(\mathbf{X}, \mathbf{Y})}{\sigma_{\mathbf{X}} \sigma_{\mathbf{Y}}} \tag{7.3}$$

which ranges in $[-1,1]$: 1 is the perfect correlation, 0 no correlation, and -1 the perfect anti-correlation.

Feature similarity

The 1-D CNN internal layers have been analysed for digging into the feature extraction phase performed by deep learning; because it is an automatic process, it is often treated as a black box. As a consequence, the focus was on characterizing the internal neurons rather than understanding which of the input variables influence the network output such as in Grad-Cam [208] and its variants [209, 210, 211]. Due to the high number of convolutional filters, it is hard to determine what each filter does. The temporal evolution is definitely important for characterizing the ECG signal; for example, R-peak frequency over time can discriminate between an healthy subject and an atrial fibrillating one. In this sense, the cross-correlation function (7.3) has been exploited for testing if some of the MLP temporal attributes can be found in the 1-D CNN feature maps. For each sample (x_i), for each convolutional filter (j), and for each temporal feature (F), it has been computed the cross-correlation between the feature map $x_i^j = (x_{i1}^j, \dots, x_{im}^j)^T$ and the temporal representation of the sample $x_i^F = (x_{i1}^F, \dots, x_{in}^F)^T$:

$$\rho_{x_i^j x_i^F} = \frac{\operatorname{cov}(x_i^j, x_i^F)}{\sigma_{x_i^j} \sigma_{x_i^F}} \tag{7.4}$$

For each j, the obtained score, averaged across all samples, yields the similarity between the feature map and the temporal attributes (remember that the higher the module of the score, the higher the similarity):

$$\rho_{j,F} = \frac{1}{N} \sum_i^N \rho_{x_i^j x_i^F} \tag{7.5}$$

The final score (7.5) is exploited for quantifying the abstraction level of the temporal attributes; in other words, it is checked *if* and *how* the deep model automatically abstracts the temporal attributes from the raw data. Table 7.5 yields the similarity results: for each temporal attribute, the minimum and maximum cross-correlations have been reported together with the best matching feature map and the corresponding filter position in the deep architecture.

Table 7.5 clearly proves that the CNN automatically extracts, in the first layer, *temporal-like* features very close to the human-engineered attributes. Probably, in the subsequent layers the network further abstracts the *temporal-like* features to improve the classification accuracy. In particular, the most correlated attributes are:

- the mean value (F1, $\rho = -0.882$)
- the max value (F2, $\rho = -0.808$)
- the root mean square (F3, $\rho = 0.875$)
- the square mean root (F4, $\rho = 0.882$)
- the crest factor (F9, $\rho = -0.838$)

Table 7.5: Maximum and minimum $\rho_{j,F}$ between temporal attributes and CNN feature maps. The highest similarity values are highlighted in bold.

Temporal feature		Layer	Filter	ρ
F1	max	1	2	0.497
	MIN	**1**	**1**	**-0.882**
F2	max	1	2	0.685
	MIN	**1**	**1**	**-0.808**
F3	**MAX**	**1**	**2**	**0.875**
	min	3	7	-0.236
F4	**MAX**	**1**	**2**	**0.882**
	min	6	4	-0.252
F5	max	1	2	0.748
	min	3	7	-0.236
F6	max	1	2	0.699
	min	3	7	-0.246
F7	max	1	2	0.553
	min	7	3	-0.180
F8	max	2	6	0.658
	min	7	6	-0.138
F9	max	1	2	0.725
	MIN	**1**	**1**	**-0.838**
F10	max	1	2	0.701
	min	1	1	-0.700
F11	max	1	2	0.711
	min	1	1	-0.780
F12	max	1	2	0.500
	min	1	1	-0.553
F13	max	1	4	0.655
	min	2	3	-0.048
F14	max	2	8	0.532
	min	1	1	-0.300
F15	max	1	4	0.681
	min	3	3	-0.014

They have been all extracted in the first layer of the network and can be grouped w.r.t. the most similar feature map. F1, F2 and F9 are strongly anti-correlated with the feature map of the first filter; their grouping can be seen as a measure of how steep the peaks are w.r.t. the average. On the other hand, because F3 and

F4 are mostly correlated with the second filter, this cluster measures the peak-to-peak amplitude. As a consequence, these feature maps can be seen as a compact representation of core waveform characteristics.

7.3.2 Final considerations

This chapter has presented an original approach, which is suited both for the impact evaluation of feature engineering in a classification problem, and, on the other side, as a tool for interpreting the feature maps automatically extracted in deep convolutional layers; in the former, the deep model is exploited as a kind of non-linear performance evaluation for the classical approach, while in the latter, human-designed features are employed as clues for understanding deep learning.

The comparison between *Net5* and the MLP on the arrhythmia classification task has confirmed the quality of the deep approach; indeed, it achieves better results by extracting in the first layer the same temporal attributes used for the MLP, i.e. these variables are fundamental for the problem at hand, and improves the classification (4%) with the others automatic extracted features of the subsequent layers.

Resuming, the two approaches have been compared on a subset of the original MIT-BIH database with a reduced number of classes; the goal is not to find the best classifier, but to determine the influence of the more abstract features extracted by the CNN w.r.t. the human-based temporal attributes. In this sense, the proposed approach has paved the way of interpreting a convolutional layer by using certain choices of features; if applied to analyse all the deep layers, will maybe provide a theoretical framework for motivating transfer learning.

Chapter 8

Telemedicine and wearable devices

Pervasive dissemination of smart, low-cost, simple-to-use devices has deeply affected modern society. Each day more and more sensors become available in the market; due to their very low price, several algorithms have been developed to handle their data and to extract the relative information. Different kind of sensors have also been combined, by both scientists and companies, to build new devices for improving everyday life. For instance, think of smartphones, which eliminate distances between people worldwide and led information to be shared easily and fastly, and wearable devices for health monitoring. A broad range of applications [212, 213] uses smartphones and their embedded sensors, such as GNSS or gyroscopes. ECG recording with the help of smartphones is shown in [214, 215], while [216] extends the approach to multichannel vital signal monitoring. The same paradigm is applied in [217] to monitor driver conditions and assess his level of stress, while [218, 219] proposes wearable devices with ad-hoc sensors; conversely, a vision system is presented in [220]. Sleep is monitored and analysed by means of different approaches: infrared cameras and motion sensors are employed in [221], while [222] uses support vector machines. Individual real time monitoring using smart clothes is proposed in [223].

Medicine and, even more, telemedicine is probably the field where the employment of personal wearable devices can have the most disruptive impact. Indeed, continuous monitoring would allow both physicians and subjects at risk to perform a medical checkup without being physically in a hospital; in this sense, it opens new terrific perspectives for the healthcare future development. In the previous chapters the importance of exploiting neural networks for medical applications has been illustrated. Indeed, because of their advanced pattern recognition and inference capabilities, such tools can help physicians to better understand diseases, e.g. Parkinson's, and, thus, to perform more accurate and earlier diagnoses. Two scenarios can happen: either these algorithms are embedded in professional medical instrumentation, e.g. electrocardiograph, or they are embedded in portable, user-friendly, low cost, wearable devices. In the former, this technology is exploited to detect and enhance significant part of an acquisition, e.g. a tumour or an arrhythmia. The latter scenario regards e-health devices, where smartphones and tablets are combined with specialized hardware to record several vital parameters (e.g. ECG or PPG), which can be stored, analysed in real time, and uplinked to the network at any time by the smartphones [224]. The neural tools can also be embedded for data fusion and, above all, to continuously monitor subjects at risk, such as hypertensive ones. In this context, two new wireless wearable devices, the ECG WATCH and the VITAL-ECG, have been developed at the Neuronica and and $Polito^{BIO}Med$ laboratories of Politecnico di Torino to acquire and monitor vital signs, such as the heart rate. The purpose is providing people with simple and effective tools for anytime, everywhere, unobtrusive checkups without the need of any medical expertise; in this sense, the final goal is exploiting continuous monitoring for detecting asymptomatic heart problems, i.e. to fight *silent* cardiovascular diseases like atrial fibrillation, and prevent their dangerous effects such as stroke, ictus and death.

8.1 The ECG WATCH

The natural human ageing may lead to alterations of the heart pace called *cardiovascular diseases* (CVDs). Several studies [225, 226, 227] account CVDs as the leading cause of death worldwide, with approximately one third of all deaths, i.e. the double than cancer, as well as more than all communicable, maternal, neonatal and nutritional disorders combined. Statistics forecast the amount of elders worldwide is expected to increase significantly in the next years; thus, also CVDs will follow the same trend. In such context, instrumentation and measurement become a key asset for cardiologists to understand patient conditions and perform diagnoses [228].

The gold standard to assess heart state of health is recording its electrical activity by means of an electrocardiograph, which employs ten wet electrodes on the human body, to analyse, simultaneously, twelve leads, both peripherals (I, II, III, aVR, aVL, aVF) and precordials (V1, V2, V4, V5, V6). The recordings are visualized into a time graph, called electrocardiogram (ECG) [229], which has to be visually inspected by an expert, e.g. a cardiologist, seeking for anomalies, i.e. diseases. These machines typically perform high-resolution recordings and, therefore, are quite expensive. ECGs have been proven to be the most effective tools for detecting CVDs [230, 231]. Multi-leads recording system yields a collection of signals, which represent different perspectives of the heart muscle electrical field; in this sense, by inspecting the twelve recordings, doctors gather a comprehensive view of the heart: an ECG anomaly signals a disease. However, patients and clinicians are required to be in the same room of the acquisition system, which implies that pathologies characterized by sporadic symptoms (*silent*), such as atrial fibrillation, are hard to diagnose because their symptoms need to occur exactly during ECG recording. Unfortunately, in a more realistic scenario, these pathologies may be latent for a very long period and, in the worst case, kill people without any prior evident symptom. To address these limits, several techniques have been proposed [232]: the common medical approach requires continuous monitoring of suspected CVD patients through Holter device [233, 234, 235], which records and stores ECGs for one or two days. The advantage is having, at the end of the monitoring period, such an amount of samples that even sporadic anomalies will be recorded, and, thus, therapies can be tailored accordingly. On the other side, these devices are non-wireless and expensive, which limits their availability in medical facilities and, consequently, the amount of subjects they could be applied on; furthermore, two days of acquisitions may not be sufficient to discover sporadic but very serious pathologies. Finally, they cannot be employed for real-time diagnoses because they need first to record ECGs and, only after their removal, the acquired data can be sought for anomalies by a practitioner.

In literature, there have been proposed many proofs of concept for controlling heart activity while being non-invasive, reliable and user-friendly [236, 237]. A diagnostic device, which uses web-services to share acquisition, is presented in [238]. Disposable electrodes and a built-in alarm routine are exploited in [239]. CVD remote monitoring based on multiple vital sign acquisition device is presented in [240]. Conductive fabric is used in [241, 242, 243, 244, 245] by means of sensors embedded in clothes. A design constraint analysis w.r.t energy efficiency for long-term monitoring can be found in [246]. For an overview of wearable and wireless ECG monitoring systems see [247].

8.1.1 Market available devices

Portable devices for ECG recording are already available in the market, but only in a small subset the ECG acquisition quality complies with the medical requirements and even fewer can share the recordings, e.g. via email. In general, they are not wearable, and require more than twenty seconds for recording. For instance, [248] proposed a device, shown in Fig. 8.1a, for acquiring, one at a time, the three peripheral leads (I, II and III); the recording cannot be printed neither shared, and, above all, its quality did not satisfy the medical standard, i.e. it cannot be exploited by cardiologists to make a diagnosis.

From Series 4 on, the Apple Watch [249, 250] (see Fig. 8.1b) is equipped with specific functionalities for heart rate computation and thirty-second ECG recording by means of sensors on the clock ring and on its back. Results can be stored and shared through Apple smartphones. Because is a smartwatch, it is inherently wearable and wireless but can record only lead I; in addition, it does not require any medical expertise to be correctly used.

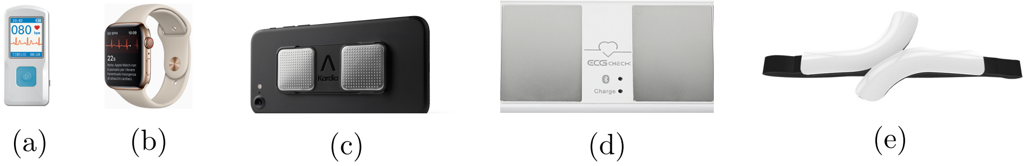
AliveCor Kardia Mobile [251, 252] is a two-plated bar, shown in Fig. 8.1c.

for acquiring a thirty-second lead I ECG, which is then shared with the corresponding smartphone or tablet app through an audio signal; then, the app removes acquisition noise, seeks for anomalies in the recording, i.e. a possible disease, and yields a pdf that can be freely shared. The device is thought to be attached to the smartphone backside as a phone cover. Because of the chosen transmission channel (audio), it needs to be very close to the smartphone to send data with low noise. Albeit heart rate computation is accurate, the EN 60601-2-27 regulation forbids to use the recorded ECG for medical diagnosis because data are excessively filtered (output signal is too flat); in other words, it cannot be exploited for heart monitoring because it misses relevant information w.r.t. heart activity.

A device similar to the previous one is the ECG Check [253] (see Fig. 8.1d), which exploits Bluetooth protocol to send data instead of audio. It is FDA cleared and Americans can subscribe to a cardiological service that will inspect the thirty-second ECG recordings from remote.

Instead of a two-plated bar to be used between two hands, QardioCore [254] (see Fig. 8.1e) has developed a chest belt to be worn under clothes. The main advantage is the possibility of a continuous ECG; it is also able to monitor physical activity and to acquire the perspiration rate. Albeit it has been thought for endless recording, its design is better tailored for usage during sport; indeed, wearing it under clothes during everyday activities may result troublesome [255].

Resuming, at the state of the art, all the available solutions need a large acquisition time ($> 20s$) and none yields a result, which can be numerical analysed while, at the same time, being comfortable for the user.



(a) (b) (c) (d) (e)

Figure 8.1: ECG recording market available devices (from left to right): GIMA palmar ECG, Apple Watch, Kardia Mobile, ECG Check, QardioCore.

8.1.2 The device

The ECG WATCH [256, 257], shown in Fig. 8.2, is a wearable and unobtrusive device, which records, in only ten-seconds, a single-lead ECG and, then, shows it into a smartphone or desktop app; recordings are stored in the smartphone/tablet in an open format, i.e. they can be used for numerical analysis, and can also be sent for inspection to practitioner (see Fig. 8.3), who determines if a deeper examination is needed. Because of its compactness, it is slightly larger (5 cm x 3 cm x 1.5 cm) than an everyday watch, the ECG WATCH can be constantly worn at wrist without any discomfort for the user; it is low cost (≈ 30€) and wireless, i.e. it does not need cables or disposable electrodes. The algorithm embedded in the app detects silent atrial fibrillation. In this sense, it has been designed to provide a full-heart-monitoring device. Currently, the proposed architecture is patent pending (WO2018073847A1: *Wearable device for acquiring electrocardiographic signals (ECG) signals*).

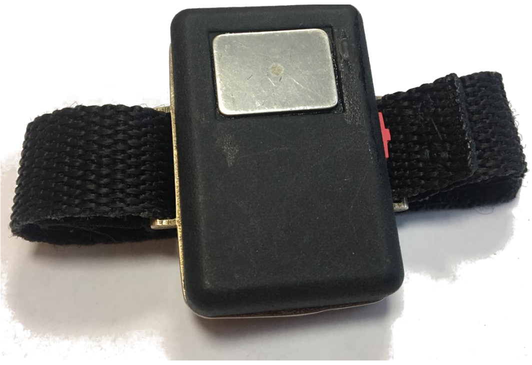


Figure 8.2: The ECG WATCH

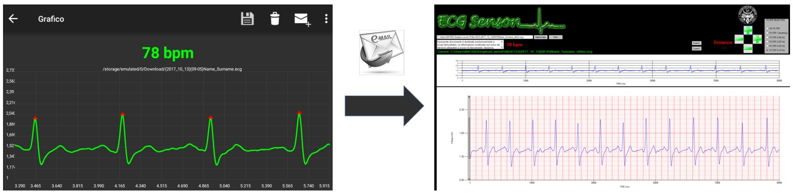


Figure 8.3: ECG recordings are visualized on the app (left) and, then, sent via email to the doctor desktop software (right).

The ECG WATCH uses two dry electrodes (one on top and the other on the back) to measure the user electrical potential difference along one of the three peripheral leads (I, II , III) of the Einthoven's triangle [258] shown in Fig. 8.4: when it is placed between user wrists, it acquires the lead I; when signal is recorded among the left leg and the right arm, the device measures the lead II; finally, if it is used between the left leg and arm, it gathers the lead III.

An acquisition lasts only ten seconds; then, noise filtering algorithm is applied and the result is transmitted via Bluetooth to the smartphone app, which:

- memorizes the ECG in a persistent way on the mobile device, which is exploited as a data logger;
- performs more advanced filtering for removing baseline wandering and the remaining noise;

- plots the filtered ECG;
- inspects the ECG to assess if an atrial fibrillation event has occurred during acquisition;
- fires an alarm when atrial fibrillation is detected;
- shares the ECG via e-mail upon user request.

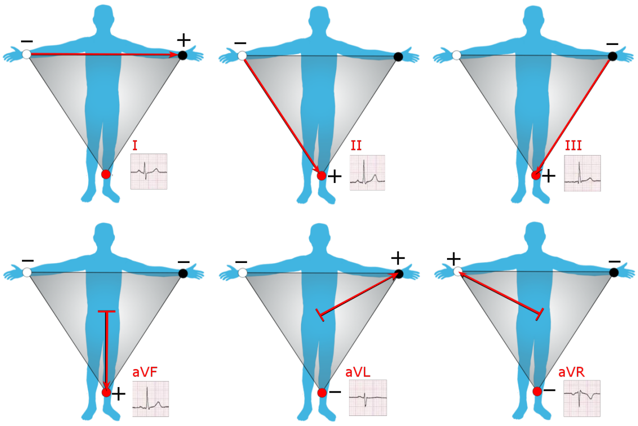


Figure 8.4: Graphical representation of Einthoven's triangle [259].

Analog circuit design

The analog chain, shown in Fig. 8.5, is analogous to [260], with a specific focus on low power consumption and space constraints. The CMRR of the IC instrumentation amplifier of the front end, the isolated battery supply and the filters provide enough rejection of RF, 50 Hz line, and muscular noises to remove the commonly used right leg drive amplifier [261], avoiding de facto the usage of a third electrode.

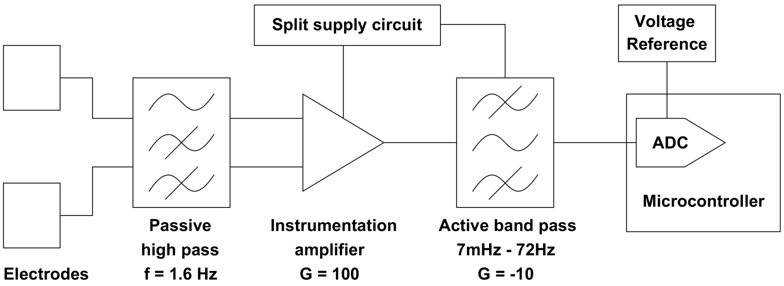

Split supply circuit
Voltage Reference
ADC
Microcontroller
Electrodes
Passive high pass f = 1.6 Hz
Instrumentation amplifier G = 100
Active band pass 7mHz - 72Hz G = -10

Figure 8.5: ECG WATCH: analog chain

The analog front end, shown in Fig. 8.6, is the Texas Instruments INA333, a single ended IC instrumentation amplifier with a passive high pass input filter and an active band pass filter, which yields another 20 dB of gain, for an overall amount of 60 dB. Because the most part of the ECG power spectrum lies beneath 70 Hz [262], the high gain analog chain can be built using low GBP operational amplifiers, which several manufacturers produce as extremely low power models; indeed, this is a crucial characteristic in a portable battery powered device like the ECG WATCH. It has been examined if including an analog 50 Hz notch filter to further lower the common mode coupling with the European main line; actually, the filtering outcome is already satisfactory without it.

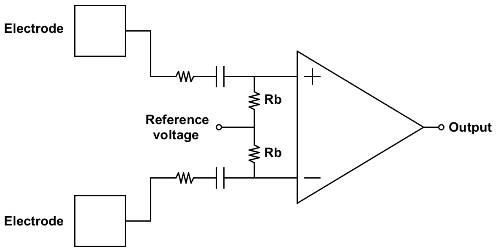


Figure 8.6: ECG WATCH: front end instrumentation amplifier schematic with the high pass filters and the biasing resistors Rb.

Fig. 8.7 shows the active band pass filter schematic, whose transfer function is:

$$H(s) = -\frac{R2}{R1}\frac{R1C1s}{R1C1s+1}\frac{1}{R2C2s} \tag{8.1}$$

which has a zero in the origin, two poles at $-\frac{1}{R1C1}$ and $-\frac{1}{R2C2}$, respectively, and a DC gain of $-\frac{R2}{R1}$. The circuit is powered from a 3.7 V battery regulated at 3.3 V by a buck/boost switching regulator, which works at 1.5 MHz. The regulator's switching harmonics are far beyond the bandwidth of the amplifiers PSRR+, but they are filtered by the relatively low GBP of the operational amplifiers, combined with the active band pass filter.

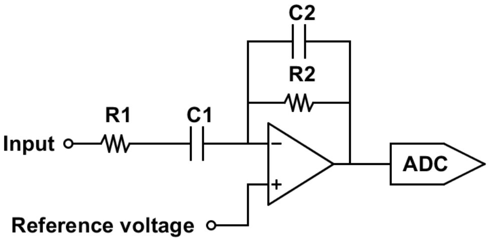


Figure 8.7: ECG WATCH: active band pass filter schematic

Fig. 8.8 yields the active circuit for splitting the voltage supply and providing the reference voltage (Vref) for the amplifiers. The slow-varying DC offsets of Vref and the amplifiers of the analog chain are irrelevant in this scenario, because the microcontroller ADC acquires the ECG signal superimposed to Vref. The absolute value of the DC voltage of the virtual ground acquired by the ADC has no interest w.r.t. the ECG signal.

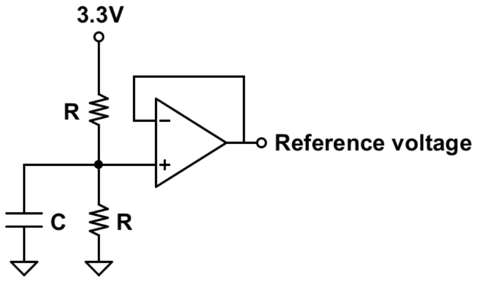


Figure 8.8: ECG WATCH: active split supply circuit schematic

The chosen electrodes are two small stainless steel plates (2 cm by 1 cm by 1 mm). The analog front end input impedance is sufficiently large for handling even electrodes made by oxidized materials such as heavily oxidized silver (exposed for a long amount of time to the ambient air), which have a quite greater impedance, without any appreciable alteration on the acquired ECG.

Digital circuit design

The ECG signal is acquired using the TI MSP430 low power microcontroller (μC), which has a 10 b 200 kbps SAR ADC on board. Signals are acquired at 1kbps, which is sufficient to reach a satisfying temporal resolution. An external reference gives a precise DC reference voltage to the ADC. For identifying the occurrence of an atrial fibrillation episode, ten-second ECG acquisition are sufficient for the designed algorithm; however, the microcontroller flash memory has enough space to memorize, on board, many seconds of acquisitions at 1 kbps, thus removing the need of an external memory module and, so, keeping the circuit compact. Furthermore, because the application is not time critical and, above all, to save PCB space, the μC exploits its internal oscillator for running at 16 MHz. The μC computational power is far beyond the actual application requirements; hence, some digital signal processing [263] could be performed directly on board rather than in the smartphone app; of course, this would reduce the battery life. Fig. 8.9 shows the printed circuit board (PCB): on the top (Fig. 8.9 left) are visible the two electrode connectors, the USB and the battery connectors; on the bottom, (Fig. 8.9 right), the μC and the remaining components.

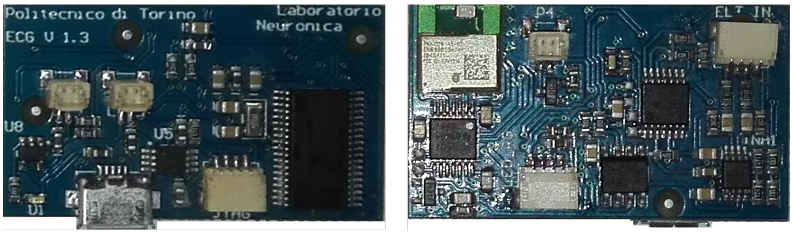


Figure 8.9: ECG WATCH: the PCB, top (left) and bottom (right)

Power consumption

Power consumption is mainly related to the digital and power sections, which grossly absorb 30 mW for acquiring the ECG and 150 mW during the short Bluetooth data transmission. Using a standard 190 mAh single cell LiPo battery the device has an estimated working period of around 10 days.

8.1.3 Testing

To assess the ECG WATCH quality, its acquisition have been compared with those of a standard three-lead monitor, the General Electric (GE) B105 [264], and a patient simulator, the FLUKE ProSim4[265]. The former has been chosen because is one of the most common CE medical apparatus used by clinicians in medical facilities, like hospitals; the latter is the gold standard for certifying medical instrumentation measurement, even the GE B105.

The test has been conducted on 30 volunteers (15 males, 15 females), aged 25—35 years old, with no pre-existing cardiological problems. Three-channel four-electrode ECGs were recorded using pre-gelled Silver-Silver Chloride (Ag/AgCl) electrodes as standard for ECG comparison. ECG WATCH acquisitions were taken among wrists, except in 5 cases (2 males, 3 females), where lead I signal was too weak and acquisitions were taken between the left leg and the right arm (lead II). Then, both the GE and ECG WATCH recordings, were post-processed using three different digital filters in cascade:

- a 50 Hz notch for removing powerline noise;
- a baseline wander removal;
- a low pass moving average for smoothing the results.

Fig. 8.10 shows the comparison on lead I on a single subject using both ECG WATCH (blue) and GE B105 (green). Qualitatively speaking, it is evident the ECG WATCH recording sensibly resembles the GE one. In this sense, it can be argued that the ECG WATCH can be employed as a medical device.

In order to asses its quality also quantitatively, a deeper analysis follows. First, the heart rate estimation has been taken into account for defining the quality of the ECG WATCH w.r.t. the GE B105; actually, it is one of the main parameters monitored by cardiologist for determining cardiac state of health. Then, given ECG recording is the main objective of ECG WATCH, the second quantitative analysis deals with ECG quality; indeed, as explained in [256], despite the vastness of instrumentation and knowledge of ECG, defining its quality is not trivial, especially from an analytical perspective. In the following, an attempt of ECG signal evaluation has been performed both in the frequency domain, with Power Spectral Density (PSD) and Signal to Noise Ratio (SNR), and in the time domain, through direct signal differences.

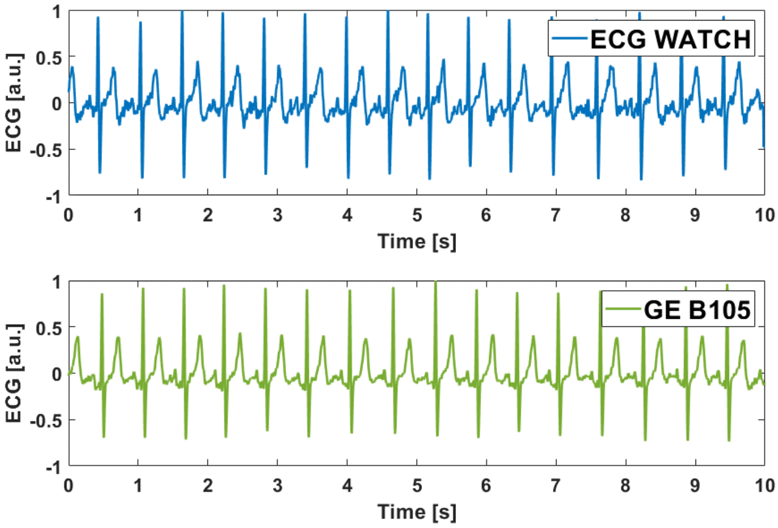


Figure 8.10: ECG WATCH: GE B105 comparison. Lead I example.

Bland-Altman plot

The performances, w.r.t. heart rate, were assessed with a Bland-Altman plot (BA plot), which is a technique to compute discrepancies in two measurement devices [266]. Differences between couples of measurements are reported in the y-axis, while the x-axis yields their means.

Fig. 8.11 shows the BA plot for the ECG WATCH and the GE B105: each blue point represents the difference between the measurement systems for a couple of heart rates. In this case, it is evident that ECG WATCH overestimates, on average (yellow line), the heartbeat by 0.6 bpm; here, data are consistent because only spread in a range of around 5% of the maximum. The measurement consistency has also been confirmed by the cross correlation between the two heart rate estimations, which resulted around 98.7% with a mean standard deviation for each subject of 2 bpm. In this sense, the proposed device has proven to be a valid instrument to estimate heart rate and follow its variation along time.

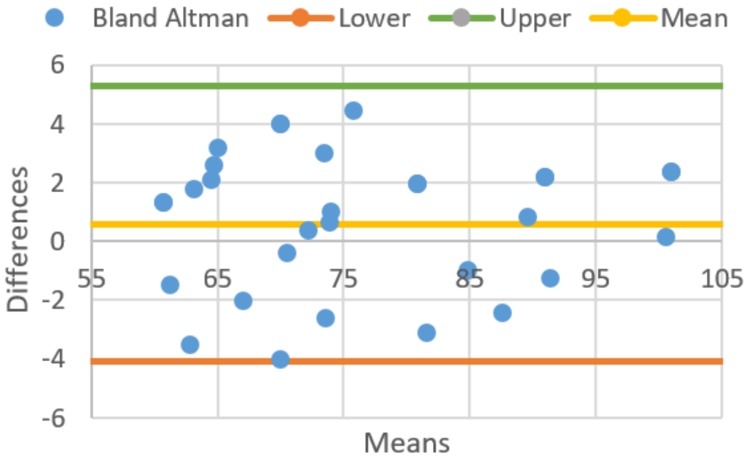


Figure 8.11: ECG WATCH: GE B105 comparison. Bland-Altman plot.

Power Spectral Density

The Power Spectral Density (PSD) yields information on the power distribution of the signal among the spectrum; here, it is exploited to evaluate the information content of each frequency. There exist several techniques for estimating PSD; for sake of simplicity, the squared discrete Fast Fourier Transform (FFT) module has been employed:

$$PSD(f) = \frac{(\Delta t)^2}{T} \left| \sum_{n=1}^{N} x_n e^{-iwn\Delta t} \right|^2 \tag{8.2}$$

Fig. 8.12 illustrates the PSD for ECG WATCH (in red) superimposed to the GE B105 one (in blue). Albeit, by visual inspection, there is no relevant discrepancy between the two curves, an additional analytical study was conducted by means of Cumulative Spectral Power (CSP). CPS is derived from PSD by means of a cumulative sum normalized with the total power. The resulting curve, $CSP(f)$, is a monotone function of the energy percentage contained by the frequencies under a certain frequency of interest f:

$$CSP(f) = \sum_{n=1}^{f} PSD(n) \tag{8.3}$$

Analysing the function argument f, it can be asserted at which frequency the input signal arrives to a specific fraction of the total power, and, therefore, of the whole information content. Consequently, it can be defined the median PSD, i.e. the f that splits the power in half, and a specific bandwidth around the median; here, f has been chosen equal to 60%. Table 8.1 reports the frequencies at which 20%, 50%, and 80% of the total power are distributed, w.r.t. CSP: the values confirm the information of ECG WATCH and GE B105 is distributed in a very similar matter, in according with Fig. 8.12. ECG WATCH has a spectrum concentrated on slightly lower frequencies than GE B105; because the great part of ECG information is found on low frequencies [267], Table 8.1 proves ECG WATCH exhibits a better behaviour in this bandwidth.

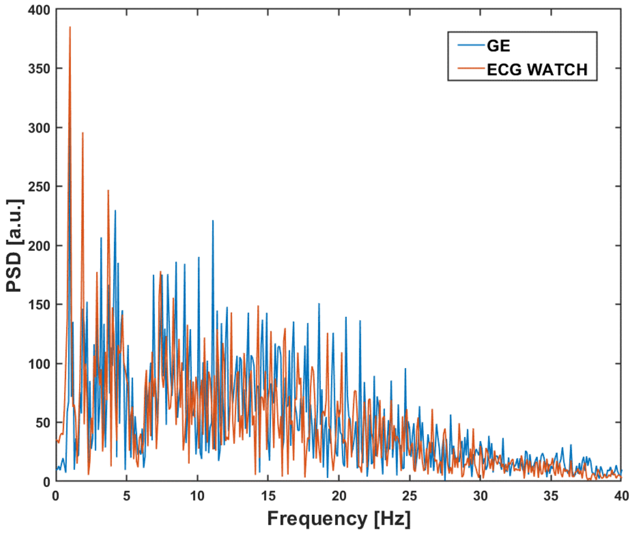


Figure 8.12: ECG WATCH: GE B105 comparison. Power spectral density.

Table 8.1: ECG WATCH: GE B105 comparison. CSP frequencies

	f 20% [Hz]	f 50% [Hz]	f 80% [Hz]
GE B105	3.9	8.7	15.3
ECG WATCH	3.6	8.6	15.3

Signal to Noise Ratio

Another frequency- based metric is the Signal to Noise Ratio (SNR), which is defined as the ratio between the signal power and the noise power and it is usually expressed in decibel (dB). The former is the meaningful content, while the latter is the unmeaningful information, and are defined w.r.t the application at hand. In this case, it has been defined as signal, i.e. meaningful information, everything in the bandwidth of 0.67 – 40 Hz, as stated in IEC 60601-2-27 regarding electrocardiographic monitoring instruments, and noise everything lying outside that band. The results are summarized in Table 8.2 as mean and standard deviation: ECG WATCH has a slightly lower SNR than GE B105 but it has also less variability, i.e. the information content of its acquisitions is more consistent in the considered bandwidth. Finally, a difference of 17 dB on the average is not very meaningful when the values are way above 100 dB.

Table 8.2: ECG WATCH: GE B105 comparison. SNR

	Mean [dB]	Standard deviation [dB]
GE B105	145.7	27
ECG WATCH	128.14	10

Time domain differences

The final comparison between the ECG-WATCH and the GE B105 is in the time domain. A dataset composed of different single heartbeats extracted from random subjects has been built; the aim was evaluating point-to-point differences between two contemporary ECG acquisitions. Signals were normalized; then, matching heartbeats were isolated and compared in pairs. Fig. 8.13 displays an example pair: the first is recorded with the ECG-WATCH (orange), while the second with the GE B105 (light blue).

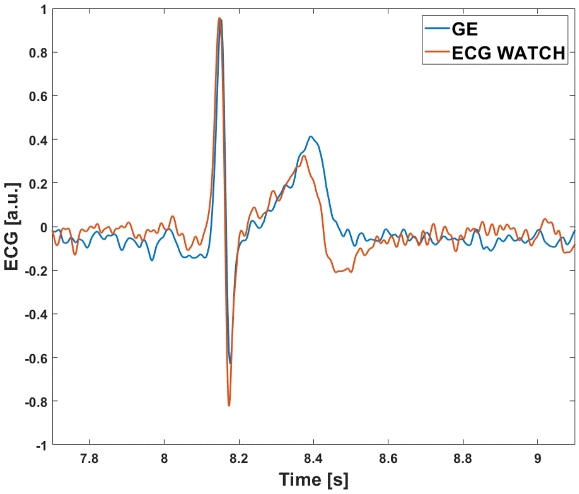


Figure 8.13: ECG WATCH: GE B105 comparison. Single heartbeat example.

Table 8.3 reports the average, the standard deviation, and the maximum value of the difference between each point of the two signals normalized to 1. Table 8.3 further confirms there are not significative discrepancies, with a average difference below 3%, and a standard deviation slightly above 9%.

Table 8.3: ECG WATCH: GE B105 comparison. Time domain differences

	Mean	Standard deviation	Max
Differences	-0.027	0.0931	0.1508

8.1.4 Atrial fibrillation detection

One of the most frequent, dangerous and hard to detect cardiac pathologies is atrial fibrillation (AF or A-fib). According to [268], A-fib is an abnormal heart rhythm where atrial chambers beat with a rapid and irregular pace. It can remain silent, i.e. without any symptom, for years and undetected even by professional tools [269]. It frequently begins as few abnormal beatings which become more frequent over time [270]. Sometimes there may be symptoms such as heart palpitations, fainting, lightheadedness, shortness of breath, or chest pain [271]. Of course, a heart beating in such an irregular way increases the risk of heart failure, dementia, and stroke.

The ECG WATCH is small as a wrist watch and needs just a tap on a phone app for recording a ten seconds ECG, that is, to check heart health. It does not demand any particular expertise, e.g. medical, to be used. Therefore, ECG WATCH is perfectly suitable to perform heart check anytime, anywhere; in this sense, ECG WATCH is the perfect candidate for atrial fibrillation prevention and, in effect, it embeds an algorithm for automatically detecting A-fib as shown in Fig. 8.14. At first, the algorithm extracts the R-peaks, i.e. the heartbeats, from the ten-second recorded ECG. Then, the beat by beat rhythm is analysed; if its variations over time exceeds a predefined threshold, the recording is classified as A-fib. On the contrary, if the rhythm is labelled “normal”, a final check on the P wave is performed. As well known in literature, in case of atrial fibrillation, P waves will be absent. Anyhow, some people with A-fib will have fibrillatory waves (a wavy baseline), on their ECG, which signal atria pulse irregularly. They may resemble P waves, and this can make an A-fib rhythm looking like an healthy sinus one. Indeed, the final algorithm block seeks for P waves and, when it found something that resembles it, tests that it is a true P wave. The algorithm is part of a patent pending for approval. As a consequence, it cannot be further detailed.

The A-fib algorithm has been tested both on real and simulated recordings. Ten fibrillating ECGs were collected from real subjects; for all patients, the software correctly detected and signaled the disease. Fig. 8.15 shows an example.

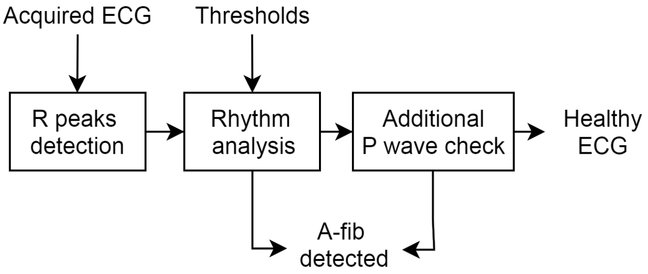


Figure 8.14: ECG WATCH: A-fib detection algorithm.

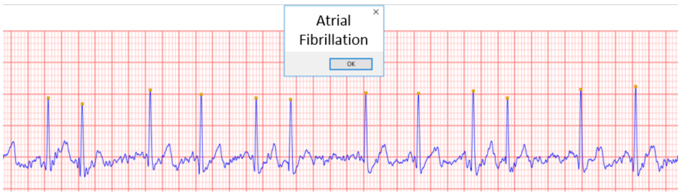


Figure 8.15: ECG WATCH: A-fib detection algorithm. Real case example.

Finally, in order to determine the algorithm quality, a stress test has been performed with the use of a certified, standard, simulator, the FLUKE ProSim 4, which is able to produce, among the others, both healthy and atrial fibrillation ECG signals. Either coarse or fine AFs have been tested. The algorithm has correctly labelled all the pathological signals as dangerous and, so, it has generated a corresponding alert for the users. Fig. 8.16 shows a comparison between ECG WATCH (in red) and GE B105 (in blue) for a simulated atrial fibrillation. As the previous case, the recording is almost identical to the gold standard.

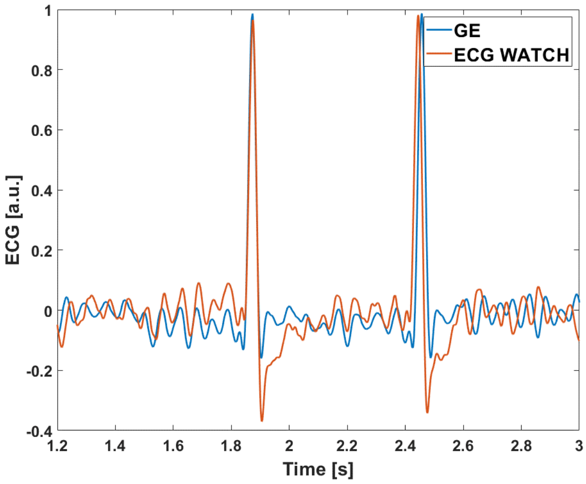


Figure 8.16: ECG WATCH: A-fib detection algorithm. Fluke simulation example.

8.2 The VITAL-ECG

Today, there not exists a device that make doctors able to remotely check patients' health or to carry out medical analysis. For instance, consider the after-surgery hospital procedures: even for basic surgeries, people are required to be hospitalized, i.e. to be under medical control, for decreasing the chance of incurring in any medical complication, which may imply severe consequences [272, 273]. Such an approach requires that, even for simple surgeries, a bed is occupied; given the limited amount of beds in a medical facility, it would be much better to use all of them only for more severe operations. Actually, people are required to stay in hospital because, in this way, it is possible for physicians to monitor their vital parameters [274], such as:

- Heart activity
- Blood oxygen saturation
- Blood pressure
- Temperature
- Fatigue
- Perspiration

With regard to patient disease, additional exams, e.g. blood tests or urine culture, may also be required to acquire a deeper knowledge about patient state of health [275, 276]. In case of day surgery, monitoring the vital parameters listed above is sufficient to assess patient conditions and determine if he can be discharged from hospitals without incurring in any complication.

Resuming, the current approach requires to have together and simultaneously:

- several medical instruments, one for each vital sign to be recorded (e.g. sphygmomanometer, electrocardiograph and saturimeter):
- specialized personnel able to correctly perform the various analysis and to interpret its outcome (e.g. the ECG) for an accurate diagnosis;
- a free bed.

Such an approach implies, of course, that the hospitalization cost per day per patient grows considerably, even in case of basic surgery [277, 278, 279].

8.2.1 State of the art

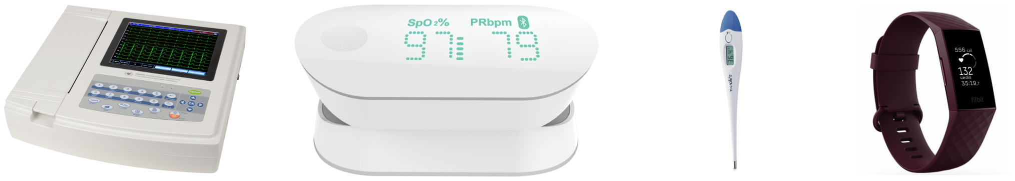
The current medical standard procedure needs ECG to be recorded with an electrocardiograph and ten wired electrodes for recording twelve leads at the same time. Portable devices for acquiring a 12-lead ECG, filtering it, and detecting alterations in the recorded signal are already available in the market [280, 281, 282]. They still employ wired electrodes that must be applied on the patient body by trained personnel (see Fig. 8.17a).

Pulse oximeters are either a component of a broader multi-parameter station - which monitors heart rate, body temperature and blood oxygen saturation (SpO2) level - or stand-alone wireless finger devices (see Fig. 8.17b), which record only heart rate and SpO_2. The latter category yields the perfusion index and is used both in medical facilities and at home; recently, they can also share data to mobile phone apps. Albeit tools like [283] can be exploited for medical diagnosis, they are not suited for continuous monitoring because their continuous wearing on fingertips would be absolutely unbearable for users.

Body temperature can be acquired either using bulb thermometers and manual readings or, in a more sophisticated way, with digital readout thermometers [284], as the one shown in Fig. 8.17c. Then, values need to be registered on a paper or, in a digital clinical folder; in this sense, such an approach avoids an automatic diagnosis.

Many non-medical devices, e.g. the Fitbit Charge 4 [285] shown in Fig. 8.17d, can be employed for monitoring physical activity. Their result is not sufficiently accurate for estimating patient level of fatigue. Values must be read and transcribed manually in the clinic folder; because such an approach can be quite user-demanding, the risk that values are not registered as often as needed is sensibly high. Therefore, they cannot be considered as medical tools and their acquisitions cannot be exploited for medical purposes.

Resuming, it does not exist a device for monitoring all the listed vital signs, which is, at the same time, wearable, user-friendly, wireless, and is able to perform acquisitions anywhere, anytime, without medical expertise.



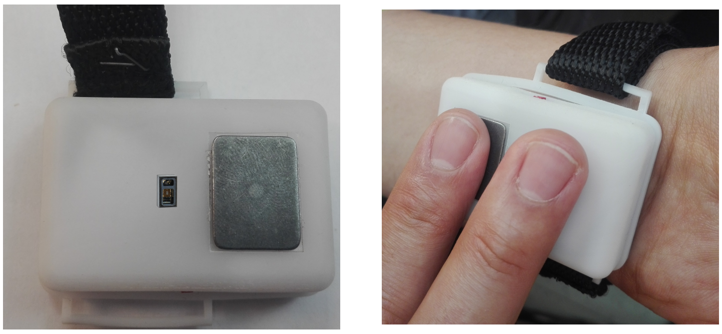
(a) (b) (c) (d)

Figure 8.17: State of the art monitoring devices (from left to right): ECG CONTEC 1200G, iHEALTH pulse oximeter, Microlife digital thermometer, Fitbit charge 4.

8.2.2 The device

The VITAL-ECG [286, 287], shown in Fig. 8.18, was designed on a precise request of two Italian hospitals, because none of the existing devices is able to satisfy all the telemedicine requirements. It is a smart wristband developed by the Neuronica Lab of the Politecnico di Torino based on the ECG-WATCH; in this sense, it can be argued that VITAL-ECG extends its ancestor by adding further sensors for measuring the most important vital parameters:

- ECG and heart rate (as the ECG WATCH);
- SpO2;
- temperature and humidity of the skin;
- physical activity level.



(a) Top view

(b) Acquisition in progress

Figure 8.18: The VITAL-ECG

The design has followed the same guidelines as the ECG WATCH: it is low-cost, wearable (size of a wristwatch) and employs a mobile app to store, visualize (see Fig 8.19), and analyse the recordings. The main focus has been its ease of use; everyone can monitor his state of health without any specific medical expertise: there is no need of precise positioning or calibrations; the only required knowledge is how to open a mobile Android app and tap a button to start the acquisition. When the algorithm detects a hazardous situation, e.g. an atrial fibrillation, it is sufficient to tap another button on the app to send patient coordinates and the recording to a predefined medical facility. The remote assistance center is provided with a complete software for asserting the patient condition; for instance, the tool allows to view the acquisition, analyse data, store notes, filtering records, and compare acquisitions. In other words, VITAL-ECG is an instrument for real-time patient telemonitoring, i.e. vital signs can be correctly acquired even if no trained specialized personnel is physically near the patient. Such an approach offers physicians a method for keeping patient health under control, and, only when really required, ask for his return to the medical facility for additional examinations; at the same time, a bed in a hospital is available for more severe cases, i.e. it is used only when really needed.

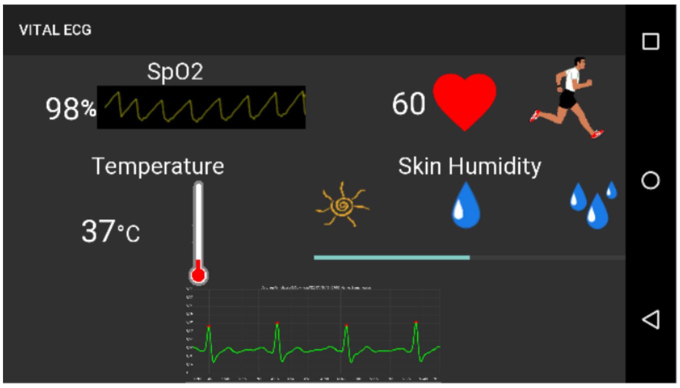


Figure 8.19: VITAL-ECG: mobile app

8.2.3 System specifications

The full design specifications, which derive from the ECG WATCH, can be resumed in the following points:

- Two-plate electrocardiograph. Any of the three peripheral leads (I, II, and III) can be recorded individually.
- Automatic atrial fibrillation detecting algorithm.
- Sensing of temperature and relative humidity of the skin.
- Gesture recognition for counting steps and waking up the device.
- Pulse oximeter to estimate SpO_2 and improve heart rate computation.
- Biocompatible: all the selected materials (the electrodes and the polycarbonate for the wristband and the case) are medical certified as skin biocompatible for preventing any harms over long-term usage.
- Highly user-friendly and intuitive.
- Bluetooth 4.0+ for connecting the device to the app in low-energy mode.
- Rechargeable lithium battery with standard USB Micro-B connector.

- Very-low power consumption; a full charge should last a week in normal conditions.
- Comfortable to be worn, which means compact and light.

The VITAL-ECG printed circuit board (PCB) and its associated block diagram are shown in Fig. 8.20 and Fig. 8.21, respectively. The ECG is the only signal directly sampled by the μC, while the other sensor modules autonomously acquire their data, which are sent via SPI to the μC only upon request. The μC memorizes both data and the acquired ECG in its internal FLASH memory until a paired device, e.g. a smartphone, is ready to accept them. After Bluetooth transmission, the system switches to a low-power mode, until the following acquisition request is received from the smartphone. The last two design specifications force the PCB to be as compact and light as possible and to reduce the power consumption at its minimum. At this aim, every component has been selected in its smallest and flattest package.

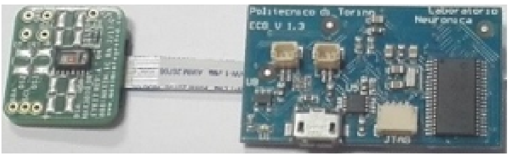


Figure 8.20: VITAL-ECG: PCB

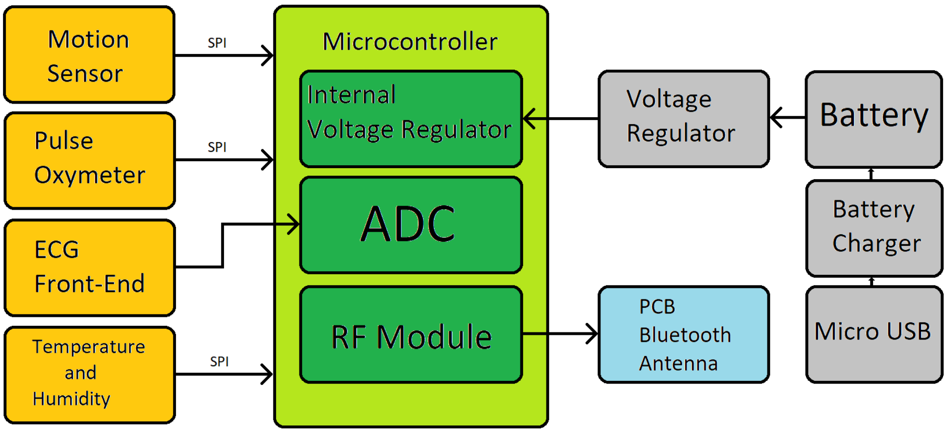


Figure 8.21: VITAL-ECG: block diagram

ECG Front-End

The ECG front-end schematic is presented in Fig. 8.22. It is made of an instrumentation amplifier, directly connected to the electrodes, and two op-amps used as high-pass, and low-pass filters, respectively.

The instrumentation amplifier is the Texas Instruments INA333 [288], which has been chosen due to its very low power consumption ($150 \mu W$), a CMRR higher than $100 dB$, and a built-in RFI filter; moreover, its relatively high resistance to ESD (4 kV HBM and 1 kV CDM) avoids the usage of an ESD suppressor.

An OPA4330 op-amp [289] has been selected due to its low power consumption and high price-to-performance ratio.

The high-pass filter is made with a single pole LP filter closed in loop to the INA333 reference. Since the system works with a monopolar power supply, a reference voltage of 1.65 V (half V_{cc}, obtained via a decoupled voltage divider) was used as a reference. The last stage of the front-end is the low-pass filter, made of a Sallen-Key topology circuit by means of Butterworth polynomial. The cut-off frequencies of the high-pass and low-pass filters were set to 0.5 Hz and 40 Hz, respectively. Additional ECG noise filtering is then performed digitally.

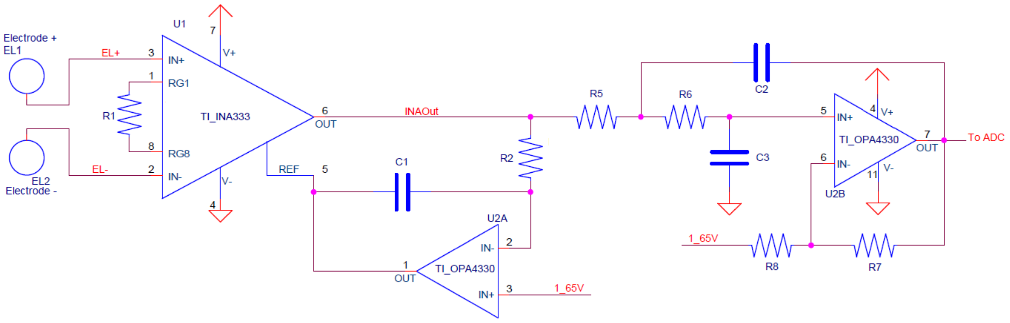


Figure 8.22: VITAL-ECG: the ECG front-end

The front-end is not meant for an usage during Bluetooth transmissions because the system only transfer data after acquisition is over. However, the 8 MHz corner frequency of the INA333's built-in RFI filters is sufficient to significatively reduce any interference with the Bluetooth RF front-end, or any analogous frequency signal (e.g. Wi-Fi). Remaining interferences are removed by the low-pass filter with over -200 dB attenuation for frequencies over 4 MHz. As a matter of fact, all the recordings ever done with the VITAL-ECG device have been performed in presence of multiple mobile phones — and any kind of connected devices nearby — with no evidence of signal degradation.

Electrodes

One of the main objectives of VITAL-ECG is to be simple to be used; in this sense, dry electrodes were the best choice. Wet electrodes require gels or disposable elements, which made the ECG recording considerably more uncomfortable for the user. Conversely, dry electrodes show poor performance with regard to wet ones.

In general, they are done with expensive materials, e.g. silver, which are prone to oxidization. The former category certainly offers better performance over dry ones; however, the additional overhead and the user discomfort w.r.t. the usage of gel is not counterbalanced by a meaningful signal improvement, which is already satisfactory for the application at hand (with dry electrodes).

To determine the best material, w.r.t. cost and performance, a study on several possible options have been conducted: stainless steel resulted as the best compromise. It is biocompatible over long-term usage on the wrist; in fact, it is one of the standard materials for building watches.

Pulse Oximeter

Pulse oximetry is a non-invasive measurement of the peripheral oxygen saturation (SpO_2) [290]. The standard way of measurement employs two LEDs, which emit red and infrared lights, respectively, and a photodiode for quantifying the light reflected from the person blood.

In the VITAL-ECG, pulse oximetry is realized using the MAX30102 [291] because it embeds in a single chip all the required electronics (the LEDs, the photodiode and the related optics). This approach yields a significative shrinking in the PCB space and, above all, reduces the final board manufacturing complexity. In addition, acquisition, post-processing, and SpO_2 computation are all completed on-chip, i.e. the microcontroller must only receive and store the SpO_2 values without any further operation.

Temperature and Humidity

Skin temperature and humidity are measured using the HTS221 [292], which is a factory calibrated, low power, ultra-compact sensor with an embedded 16-bit ADC, and communicates with the μC via SPI.

Motion Sensor

The motion sensing is performed using a MPU-9250 [293], a nine-axis tracking module that embeds in a small QFN package:

- a three-axis accelerometer;
- a three-axis magnetometer;
- a three-axis gyroscope;
- a digital motion processor (DMP).

The MPU-9250 has also nine 16-bit ADCs (one for each axis), and programmable digital filters. The embedded processor performs basic gesture recognition, e.g. tilts, and triggers interrupts through a dedicated pin.

This sensor is actually used only for step counting and waking up the device. In any case, in a future release, it could be exploited for advanced recognition of hazardous events, e.g. falling.

Power

A single 3.7 V LiPo rechargeable battery with 190 mAh capacity powers the VITAL-ECG. The power supply is then regulated at 3.3 V with the MAX1759 [294], a low noise buck-boost voltage converter that does not need an external inductor, thus saving space.

The power circuitry section embodies the MAX1555 [295], which is an integrated circuit for safely charging the battery avoiding considering the needed protection from events such as overcharging, short-circuit, overheat, polarity exchange, etc.

The device is charged with a standard micro-USB type B connector [296], i.e. the same recharging cable of modern cellphones can be used.

The overall power consumption ranges from an mean measured value of $\approx 160\mu W$, in standby, to $30mW$, during transmission.

Microcontroller

The chosen microcontroller for VITAL-ECG is the CC2640R2F, from Texas Instruments [297], which is expressly designed for low power wireless sensing applications: from 9 mW in full speed operation, to 2 μW in standby.

To further reduce its energy demand, it comprises an additional ultra-low power sensor controller, which let the integrated 12 bit ADC to work independently of the main processor; in this sense, it can be selectively shut down during recordings for energy harvesting.

In addition to what listed above, the CC2640R2F has been selected because of its integrated Bluetooth transceiver, which needs only a small patch antenna, i.e. a very small PCB area and an easier implementation.

Finally, for improving battery performance, the μC is shut down whenever possible; its awakening is performed tapping three times on the top electrode (the event is recognized by the motion sensor).

Bluetooth

The integrated CC2640R2F RF transceiver is compliant with Bluetooth low energy (BLE) 4.2 and 5.0 specifications [298]. It is unrealistic that VITAL-ECG will require long range Bluetooth transmissions given that is watch-shaped. In this sense, the antenna was designed as a low gain patch antenna, i.e. a PCB trace, which yields space-saving and also limits the overall power consumption to less than 13 mW during transmission.

To further lower power consumptions, Bluetooth communication is restricted to a single transmission burst at the end of each acquisition, except for the necessary starting signal from the mobile app.

8.2.4 Testing

Among all the vital signs acquired by the VITAL-ECG, only the ECG is measured with an ad-hoc circuit, i.e. the front end detailed in Sec. 8.2.3; indeed, the others, e.g. SpO_2, exploit sensor modules designed and calibrated from their corresponding manufacturers at this specific purpose: i.e. measuring pulse oximetry, skin temperature and humidity and subject motion. In this sense, the only signal whose quality needs to be assessed is the ECG, while the others have already been optimized by the producers.

The ECG quality has been analysed on 36 healthy volunteers w.r.t. the GE MAC2000 [299], which is a professional electrocardiograph widely used in medical facilities and known for its reliability. As per the ECG WATCH testing (see Sec. 8.1.3), all the recordings are obtained from healthy resting subjects to avoid all superfluous motion artefacts. For each volunteer, five ECGs were recorded for the two instruments simultaneously, and the resulting acquisitions were compared for discrepancies; recordings were alternated with one minute of rest, for data saving and setting up the following acquisition.

The MAC2000 was equipped with four stainless steel electrodes — the same of VITAL-ECG — placed on the body according to the Einthoven's triangle: right hand, left hand, and left leg [300], while the fourth was placed on the right leg for disturbances reduction. Despite this configuration yields the three peripheral leads, only lead I has been considered because the majority of VITAL-ECG recordings are acquired between the right wrist and the left thumb.

Eventually, some volunteers were demanded to wear the VITAL-ECG a whole day and to perform as many recordings as possible; the aim was testing the device comfort over long period and to check if ECGs were robust over time or if, on the contrary, they were affected by some noise; results did not show any significative fluctuation except in case of battery discharge. At this aim, a firmware module has been implemented to monitor battery voltage and to inhibit recordings when this scenario occurs; therefore, also the mobile app has been extended to yield this piece of information.

Bland-Alman plot

The performances, w.r.t. heart rate, were evaluated with a Bland-Altman plot (BA plot) [266]. The y-axis reports differences between couples of measurements, while the x-axis yields their means. Fig. 8.23 shows the BA plot for the VITAL-ECG and the GE MAC2000. It exhibits a zero-mean value, with less than 5% of variation around the maximum, i.e. there is no meaningful difference (on average) between the heart rate estimated by the two devices. The cross-correlation between the two heart rate detections is 90.5%.

Resuming, measurements are consistent and it can be argued that the VITAL-ECG correctly estimates the heartbeat.

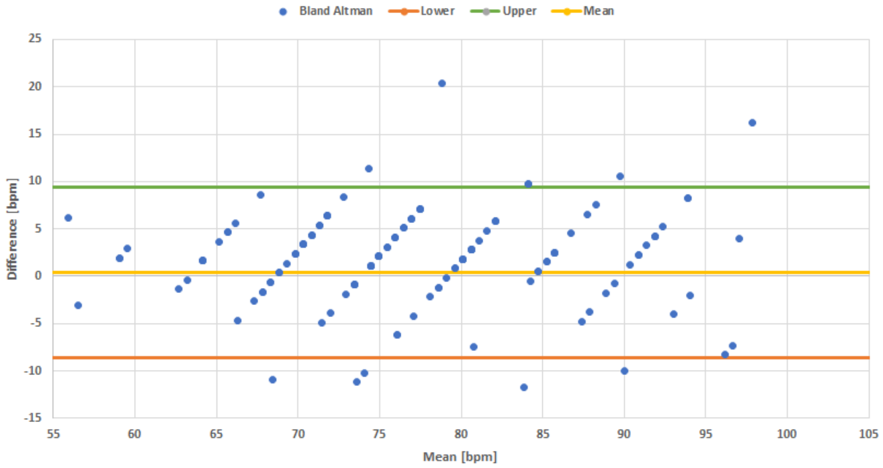


Figure 8.23: VITAL-ECG: GE MAC2000 comparison. Bland-Altman plot.

Power Spectral Density

ECG acquisition have been deeper analysed by means of Power spectral density (PSD), which is an indirect way to determine how the information content is spread among the spectrum. As per the ECG WATCH, the squared discrete FFT (8.2) has been employed.

Fig. 8.24 yields the results. Despite by visual inspection there is no significative difference between the two PSDs, am additional analytical study was performed by means of CSP (8.3). Table 8.4 reports the frequencies at which 20%, 50%, and 80% of the total power is distributed w.r.t. CSP: values reconfirm the information of the VITAL-ECG and MAC2000 is distributed in a quite similar way, in accordance with Fig. 8.24. Half of the VITAL-ECG information content is in the bandwidth 0–11.4 Hz, i.e. its spectrum is concentrated on slightly lower frequencies than MAC2000 (0–13.6 Hz). Since most of ECG information is located on low frequencies [267], Table 8.4 proves VITAL-ECG exhibits a better behaviour in that bandwidth.

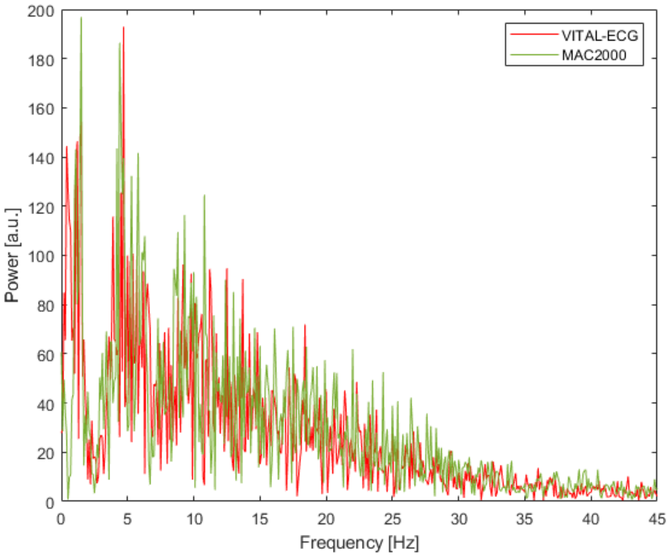


Figure 8.24: VITAL-ECG: GE MAC2000 comparison. Power spectral density.

Table 8.4: VITAL-ECG: GE MAC2000 comparison. CSP frequencies

	f 20% [Hz]	f 50% [Hz]	f 80% [Hz]
GE MAC2000	5.5	13.6	31.9
VITAL-ECG	4.3	11.4	25.6

Time domain differences

The final comparison between the VITAL-ECG and the GE MAC2000 is in the time domain. A set composed of different single heartbeats extracted from random volunteers has been built; in this sense, the aim was evaluating point-to-point differences between two contemporary ECG acquisitions. Signals were normalized; then, matching heartbeats were isolated and compared in pairs. Fig. 8.25 displays an example pair: the first is recorded with the VITAL-ECG (light blue), while the second with the GE MAC2000 (orange).

Table 8.5 reports the average and the standard deviation of the difference between each point of the two signals. It confirms the previous analysis: there are not significative discrepancies, with an average difference below around 1%, and a standard deviation slightly above 12%.

Table 8.5: VITAL-ECG: GE MAC2000 comparison. Time domain differences

	Mean	Standard deviation
Differences	-0.011	0.1213

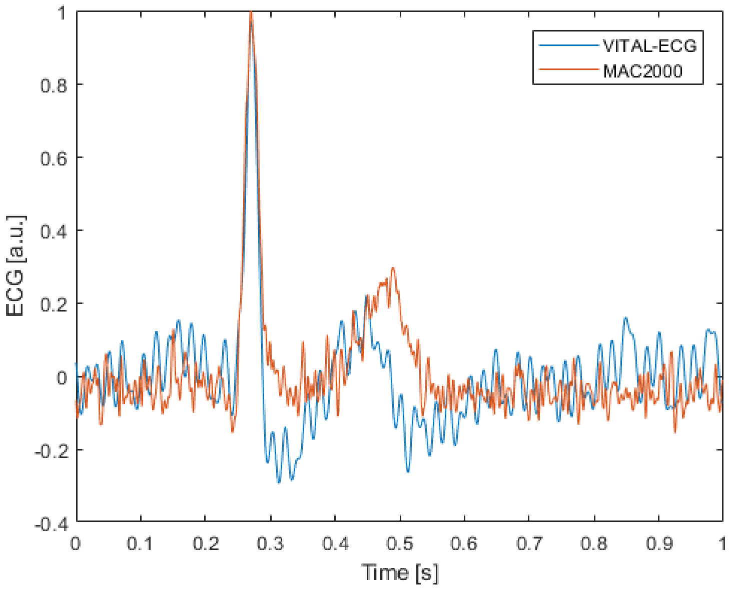


Figure 8.25: VITAL-ECG: GE MAC2000 comparison. Single heartbeat example.

8.3 Final considerations

Cardiovascular diseases characterized by sporadic ECG anomalies, such as atrial fibrillation, are hard to be detected. Current solutions do not solve this problem; although they detect some episodes, they are neither wearable nor wireless and their usage over a long-term period is unrealistic. On the other side, no portable or wearable solution exists to allow patients being monitored remotely; to this purpose, several vital parameters need to be kept under physician control. Currently, it means to stay in hospital wired connected to various medical instrumentations, whose results need to be interpreted by specialized personnel, which needs, also, to be physically near the machinery at hand.

To tackle all the above problems a telemedicine approach based on wearable, low-cost, user-friendly devices has been proposed. Two novel tools, the ECG WATCH and the VITAL-ECG have been designed, built and tested w.r.t. the corresponding gold standard. Few considerations can be done about both the devices:

- ECG quality has proven to be as good as gold standards both in terms of spectral content and time domain differences;
- they can be comfortable wrist-worn all day;
- no medical expertise is required for placing or usage;
- unobtrusiveness implies anytime, anywhere, recordings without the need to physically go to hospitals or cardiologists;

- a single touch on the relative mobile app yields the acquisition in only ten seconds;
- despite recordings last only ten seconds, heartbeat estimation is consistent with regard to the gold standards;
- atrial fibrillation algorithm has proven to be valid in detecting Afib episodes;
- having a numerical open-format output file allows to apply any kind of subsequent post-processing.

Due to the above considerations, both the ECG WATCH and the VITAL-ECG have proven to be interesting and promising devices for health monitoring and pathology recognition, such as silent atrial fibrillation, without any user medical expertise or going to a doctor; as a consequence, they can be also employed for continuous monitoring of subjects at risk.

The proposed telemedicine wearables can be expanded for detecting more advanced medical information. For instance, the neural networks for ABP estimation and ECG classification proposed in the previous chapters can be embedded in the mobile app; in this sense, the devices will exploit the additional level of intelligence and their *general* sensors for performing much complex diagnoses such as arrhythmia or hypertension, which, as explained, affect (and kill) a huge amount of people worldwide.

Chapter 9

Biometric ECG

In the last years, security applications are gaining more and more attention; the growing amount of available information technologies, smartphones and wearables, yields to an exponential increase in the data rate shared on internet. Information is always travelling around some sort of connection, e.g. Wi-Fi or Bluetooth. In such a scenario, one fundamental aspect to be considered is access control [301], which means each piece of information is provided only to authorized users. In applications where sensitive data are at stake - such as surveillance, banking and healthcare - data confidentiality and integrity are strictly related to accurate human recognition, i.e. access is inhibited until the user has been authenticated [302]. In this sense, the key step for performing an effective access control is *authentication*, where the user identity is unequivocally determined.

According to [303], during 2018, just in USA, around 445K identity thefts have been reported to the consumer sentinel network; indeed, this kind of fraud is the third more frequent (15% of all frauds) just after imposter scams (ID falsification) and debt collection. The latest data (March 2020) reported by US Federal Trade Commission show an increasing trend (see Fig. 9.1) over the last years for all the accounted types; among this credit card fraud, i.e. people who said their information was misused on an existing account or to open a new one, tops the list since mid 2017.

The most common strategy for automatic identity recognition exploits the use of a *secret* piece of information for encrypting and decrypting some authentication data. Such an approach is widespread because of its ease of implementation; it works very well until the *secret* is kept safe, i.e. it is highly prone to the risk of exposure, forgetting, loss, or theft. In general, depending on the kind of information it relies on, *authentication* can be grouped into three categories [305]:

- knowledge-based, e.g. passwords, PIN numbers and questions;
- physical authentication device, e.g. tokens or ID cards;
- biometrics, e.g. fingerprints, iris and face.

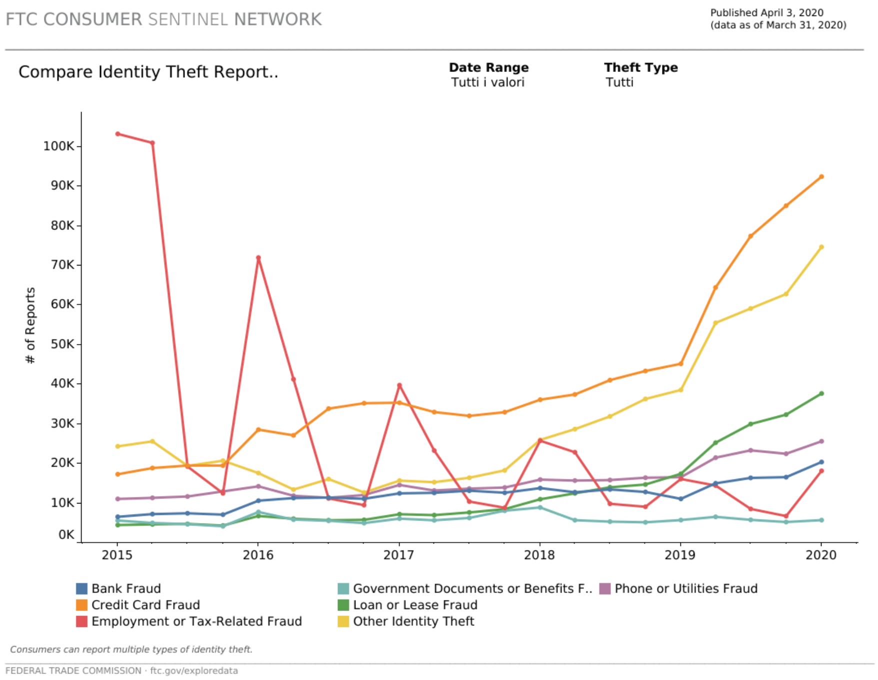


Figure 9.1: US identity theft report types over time [304].

Biometric-based systems exploits the intrinsic properties of an individual, which can be further subgrouped into physiological and behavioural features. The former category is related to the shape of the human body like fingerprints, faces, DNA, hand and palm geometry, and iris; the latter group relies, instead, on the subject behaviour, e.g. typing rhythm, gait, and voice. Biometric systems are, in general, harder to be falsified than the other two categories; indeed, biometric security is now mainstream thanks to smartphone applications, such as fingerprint, face and speech recognition. However, these methods are not 100% safe because they still can be falsified, sniffed and counterfeited: face recognition can be tricked by a picture, fingerprints can be artificially recreated and voice can be imitated or pre-recorded [306, 307].

Biometric features

In order to find the best human-related feature to be exploited for user authentication, with regard also to its forgery, it is essential to define which biological measurements can be used for biometrics. In this sense, any physiological and/or behavioural trait can be employed if it fulfils the following [305]:

- Universality: everyone must possess the characteristic; it may seem trivial but it is not, e.g. consider subjects without hands, deaf or voiceless people.

- Distinctiveness: any pair of subjects should differ significantly w.r.t. the selected characteristic, which means it should be meaningful with respect of the classifier.
- Permanence: it should be sufficiently invariant w.r.t the matching criterion over a period of time and also on the acquisition conditions; e.g. fingerprints do not change over a person life and need to be recognized for different finger sweating.
- Collectability: it must be quantifiable w.r.t. a measurement technique.

In addition, a real biometric authentication method should also take into account the following aspects:

- Performance: what are the maximum and average accuracy and speed of the technique? What are the resources needed to have a satisfying level of performance within a *reasonable* amount of time? For example, is a smartphone sufficient? Or a supercomputer and a fast connectivity are required?
- Acceptability: how many people are inclined to use that specific biometric identifier for daily operations, such as mobile phone unlocking or electronic payments?
- Circumvention: how hard is fooling the system? In other words, how robust is the authentication algorithm?

Actually, in order to be successful, a real biometric authentication method should meet the specified recognition accuracy, speed, and resource requirements, be harmless to the users, be accepted by the intended population, and be sufficiently robust to various fraudulent methods and attacks to the system [305].

ECG biometrics

A huge effort has been made for the development of new biometric techniques inherently robust to circumvention, obfuscation and replay attacks [308]. In this sense, a novel family of authentication methods has been explored extensively during the last decades; it is based on biosignals typically used for medical diagnoses such as electrocardiogram [309, 310], electroencephalogram [311, 308] and PPG [312]. In particular, the former has recently gathered much attention of the research community because both the physiological and geometrical differences of each human heart correspond to uniqueness in the ECG morphology [313]. ECG exhibits various significant properties such as uniqueness, permanence, and ease of collection [309]; compared with traditional techniques, ECG-based methods can yield a more reliable and safer way for user authentication [314]:

- ECG is an internal signal and no latent signatures are naturally left behind; it is tougher to be sniffed without user knowledge.
- The inherent inter-variability of each recording implies ECG is hard to be fabricated; as side effect, this feature yields also liveness detection, which can be extremely useful for system security.
- ECG is typically less influenced by the ambient environment than other biometric techniques, such as voice or face recognition, where ambient noise or lighting conditions can deeply affect the recognition process.
- The ECG signal can be acquired via various conductive materials and simple electronics, which can also be easily embedded in fabric or wearables (see Chap. 8).
- ECG can be exploited for continuous authentication and *beyond authentication*, i.e. not only for identifying the user but also for providing a real-time insight into his wellness state and/or level of stress.

With respect to the nature of the considered features [315], ECG-based biometric systems can be grouped into three sets: fiducial, non-fiducial and hybrid. The former approach is based on fiducial extraction, which are specific points on the ECG heartbeat related to the characteristic P-QRS–T waves, and their employment as input features, which may also involve their amplitude, angle, or duration. For instance, in [316] is proven that emotional and mental state variations do not affect ECG based authentication process. Using the same database, in [317] fifteen fiducial features are extracted with respect to the R peaks; this technique reaches 82% and 79% heartbeat identification rates using two different ECG sites (neck and chest) and average accuracies of 80.1% and 64.5% w.r.t. different anxiety states. On the other hand, [318] included also the fiducial amplitude and duration, as well as QRS and PR intervals; it achieves 79% and 85.3% of accuracy w.r.t. different lead configurations. Further examples can be found in [319, 320, 321, 322, 323].

Non-fiducial methods are based on signal statistical characteristics rather than specific points on the ECG curve; the extracted features can be either in the time or frequency domain. Autocorrelation and linear dimension reduction together with kernel principle component analysis and SVM is proposed in [324]. K-nearest neighborhood classifier and Hadamard transform are used in [325]. A 1-D CNN is presented in [326]. The discrete cosine transform and autocorrelation coefficients are employed in [327, 328, 329]. Spectro-temporal signal features based on a 2-D CNN are exploited in [330]. Statistical features and random forest are suggested in [331]. Wavelet and autoencoders are recommended in [332]. Genetic algorithm and particle swarm optimization are studied in [333], while [334] implements fuzzy logic.

Finally, the hybrid approach combines both fiducial and non-fiducial features [335, 336]. As an example, in [313] the P, Q, R, S and T positions and amplitudes are used as fiducial features, while autocorrelation coefficients and discrete cosine transform as non-fiducial ones.

The ECG WATCH biometric system

Data breaches could be avoided by the use of biometric authentication systems for restricting the access to critical software and sites, e.g. airport security areas, hospital neonatal wards, and public buildings. With physical security and safety firmly at the top newspaper pages, the relative identity and access management market is quickly growing and it already accounts for more than $4bn, with biometric hardware credentials being a key growth trend [337]; in particular, an increasing amount of companies is developing the technology for deploying ECG biometrics in both consumer and enterprise applications, such as smart clothing, access control cards and wrist wearables [314].

In this contest, a perfect tool for ECG biometric authentication is the ECG-WATCH (see Sec. 8.1) [257]; indeed, as proven in Sec. 8.1.3, the device yields high-quality acquisitions, which can be exploited for discriminating among people by means of the neural approach described in Chap. 7. In this sense, instead of classifying heart pathologies, the neural system is exploited for discriminating among different individuals. Because of the usage of wearable devices and mobile apps, and the need of a fast recognition algorithm, the MLP approach (see Sec. 7.2) is preferred with respect to the 1-D CNN shown in Sec. 7.1.

9.1 The experimental dataset

ECGs have been collected in the Neuronica Lab of Politecnico di Torino on six male volunteers: five healthy subjects and one cardiopathic (*Subject3*). All acquisitions were taken among wrists; because of ECG WATCH, data are sampled at 1 KHz; autocorrelation and discrete cosine transform have been applied to each ten-second ECG recording for extracting heartbeats (HBs), whose length was empirically fixed to twenty time-instants. For each subject, the number of acquired ECGs, together with their corresponding total amount of HBs, are detailed in Table 9.1. The final dataset has 2331 rows, equal to the cumulative sum of HBs, and 20 columns, i.e. the chosen heartbeat size. As depicted in Fig. 9.2, there is not a common pattern for all the subjects; R-peaks are somehow distinguishable but the plot is very noisy. A deeper level of analysis is displayed in Fig. 9.3, where each volunteer HBs are plotted into a separate subfigure. Subject2 and Subject5 are very well concentrated around their mean; indeed, the heartbeat is clearly visible. Subject6 is thicker than the previous cases but the overall shape is still appreciable. On the contrary, Subject1 and Subject4 exhibit a much noisier behaviour, while Subject3 heartbeats are absolutely indistinguishable. It can be argued that the morphology loss of the latter case is related to the cardiovascular disease.

Table 9.1: Dataset taxonomy

	Age	Sex	No. of ECGs	No. of heartbeats
Subject1	26	M	47	429
Subject2	27	M	22	185
Subject3*	60	M	63	748
Subject4	24	M	56	531
Subject5	27	M	20	190
Subject6	23	M	31	248

**Cardiopathic*

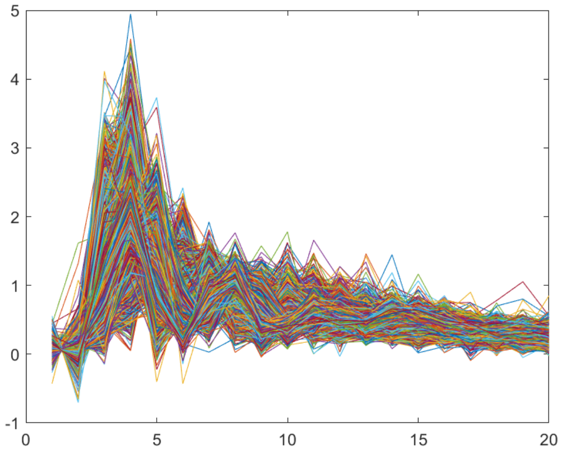
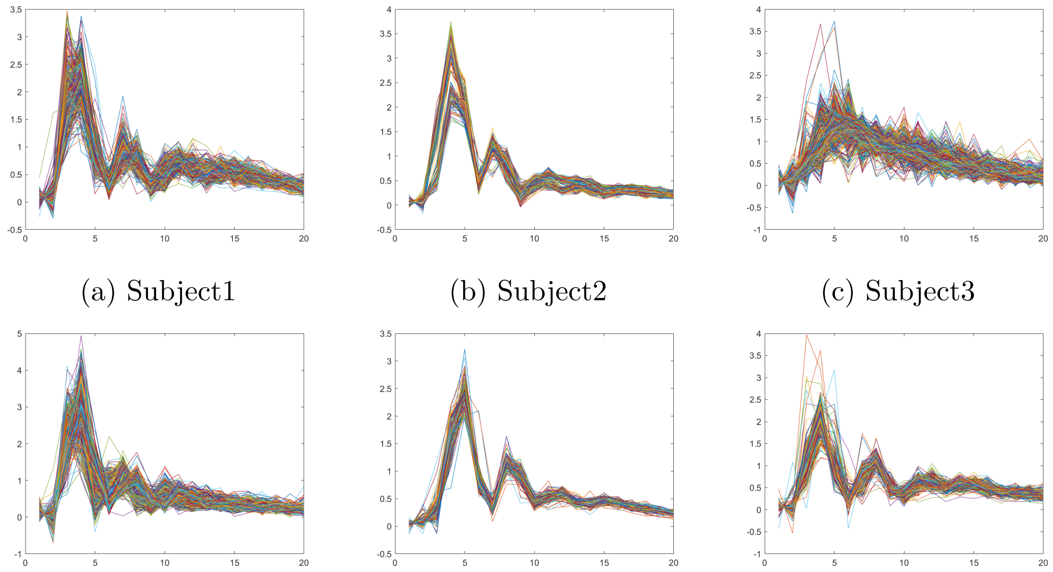


Figure 9.2: Heartbeat visualization: whole dataset.

9.2 Manifold analysis

In order to have a first insight on the database and, particularly, on its intrinsic dimensionality, a preliminary PCA analysis has been performed. The relative Pareto chart computed on the whole dataset is shown in Fig. 9.4; the intrinsic dimensionality is around 12 (more than 90% of variance explained). Because of the differences emerged in Sec. 9.1, each subject subset has been studied separately w.r.t. its intrinsic dimensionality; Table 9.2 summarizes the results: despite the intrinsic dimensionality of the whole dataset is equal to 12, it varies a lot w.r.t the subsets, from a minimum of 8 up to 15. Interestingly, Subject3, whose plot is the less HB shaped, has also the higher intrinsic dimensionality w.r.t. the PCA linear analysis.



(a) Subject1 (b) Subject2 (c) Subject3

(d) Subject4 (e) Subject5 (f) Subject6

Figure 9.3: Heartbeat visualization: single subject.

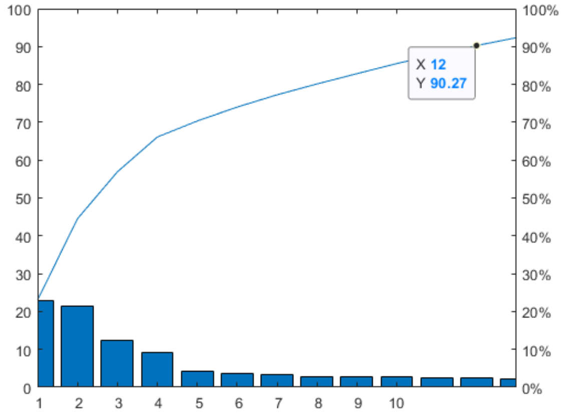


Figure 9.4: Pareto chart: whole dataset.

9.2.1 CCA

The PCA analysis has shown a large range of variation w.r.t. the manifold intrinsic dimensionalities; in this sense, it was not conclusive. As a consequence, a more advanced non-linear study has been conducted by means of the CCA. As explained in Chap. 2, the CCA *dy-dx* diagram can be exploited to assess the projection quality, i.e. to determine the intrinsic dimensionality. As a first step, the CCA ($\lambda = 5$, $epochs = 50$) is used for projecting the whole dataset into a

10-D subspace; Fig. 9.5 shows the corresponding *dy-dx* diagram, which proves the intrinsic dimensionality is 10, i.e. lower than PCA. Fig. 9.6 yields the *dy-dx* diagrams for all the subsets; in all the projections λ and *epochs* were set equal to 5 and 50, respectively, while the corresponding subspace dimensionalities can be found in Table 9.2. The non-linear analysis shows that, in all cases, PCA overestimates the intrinsic dimensionality except for Subject3; it can be argued that because of the higher explained variance ($> 92\%$), the input manifold and, therefore, the intrinsic dimensionality, is better approximated. As before, the CVD patient lies in a manifold quite higher (1.5 times) than the healthy subjects, which suggests the CCA is able to catch the higher level of irregularity (non-linearity) of the input signals; in a certain sense, this confirms what seen in Fig. 9.3c, where the heartbeat shape was lost. Finally, it must be underlined the higher coherency of the non-linear analysis w.r.t. the PCA one; indeed, the intrinsic dimensionality estimated for the whole dataset, i.e. 10, is also the value derived for the most part of subjects, while in the linear analysis, it changes depending on the subset at hand. In this sense, the local topology preservation property of CCA proves to be a valid tool for input space approximation; therefore, it yields more meaningful results on manifold analysis, such as the intrinsic dimensionality estimation.

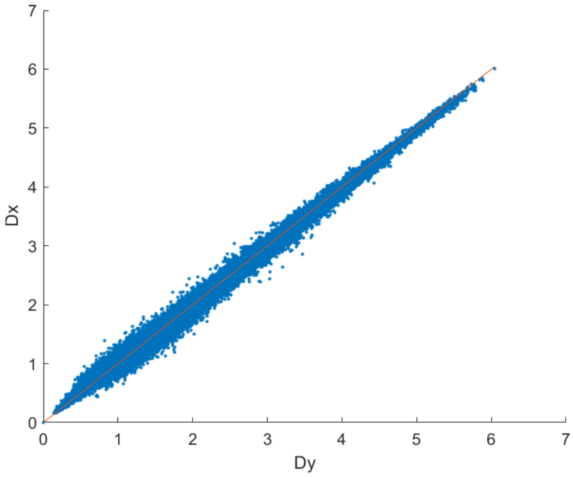
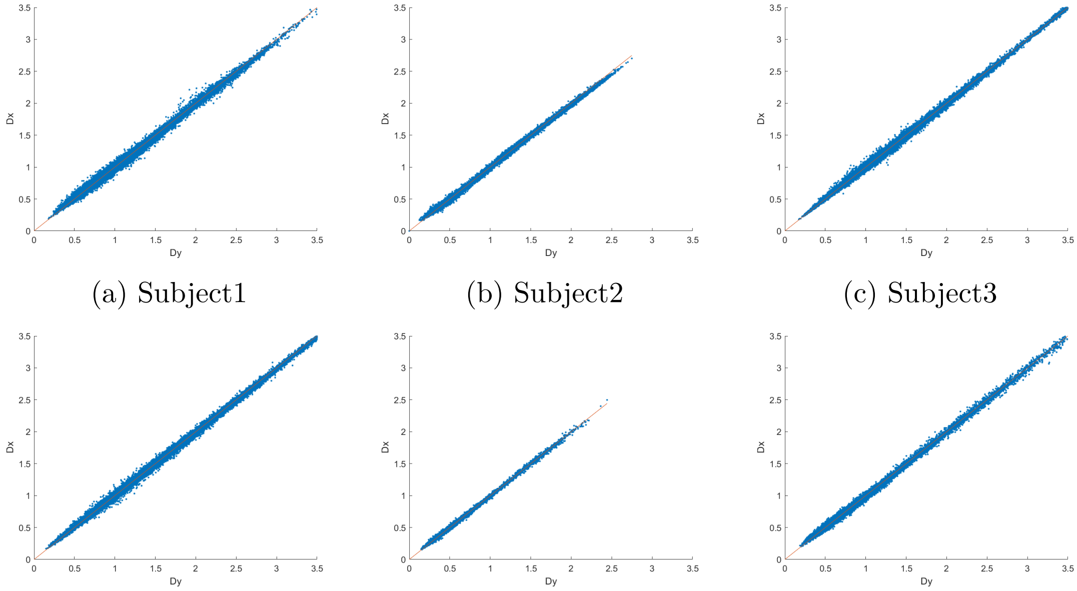


Figure 9.5: *dy-dx* diagram: whole dataset.

Table 9.2: Intrinsic dimensionality

	Whole DB	**Subject1**	**Subject2**	**Subject3**	**Subject4**	**Subject5**	**Subject6**
PCA*	12 (90.27)	14 (91.04)	8 (90.12)	15 (92.23)	11 (90.45)	14 (91.44)	13 (91.92)
CCA	10	10	6	15	10	10	10

*in brackets the percentage of explained variance.



(a) Subject1 (b) Subject2 (c) Subject3

(d) Subject4 (e) Subject5 (f) Subject6

Figure 9.6: *dy-dx* diagram: single subject.

9.3 MLP authentication

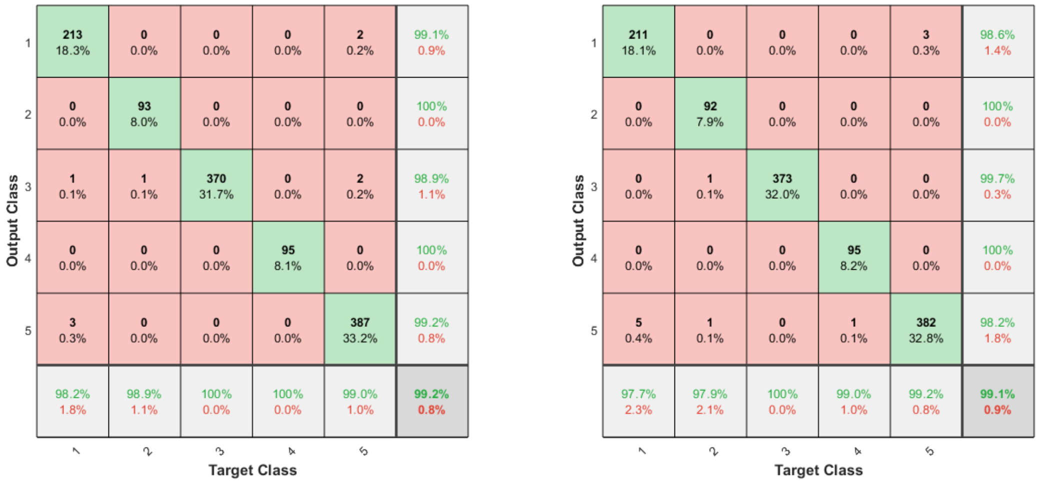
The previous analyses have shown that the input dataset is quite simple to be clustered in terms of healthy and sick patients; unfortunately, the same cannot be already stated about discriminating among individuals. As previously explained, the wearable paradigm requires to have the simplest possible algorithm with regard to both the computational complexity and the time needed for providing a result, i.e. the authorization token. On the other hand, the algorithm accuracy cannot be ignored; on the contrary, it is the most relevant constraint to be taken into account.

At this purpose, a simple shallow neural network, similar to the one employed in Sec. 7.2, has been trained. The input layer is mapped one-to-one to the input features; thus, it is made of twenty units. The hidden layer is made of fifty neurons, and the output units are associated with soft-max activation functions [4]. Due to the cross-entropy error function, the network outputs the membership probability for the input classes. In order to counterbalance the overrepresentation of Subject3 ($\approx$ 750 samples), the two youngest attendees (Subject4 and Subject6), were merged into a novel fifth class, say *other*, which accounts to around 780 heartbeats; in addition, this class is employed for modelling all the individuals *external* to the authentication system. The shallow network has been trained by means of the Scaled Conjugated Gradient algorithm [4]. In all the simulations, the five labels were exploited to split the input dataset into balanced training, validation and test subsets; in this sense, the input label distribution is preserved. Seventy percent of the training set was used for training, while and the rest was divided in equal parts for test and validation sets, respectively.

The training and testing confusion matrices are shown in Fig. 9.7; few considerations can be done:

- in both cases the overall accuracy exceeds 99%;
- the precision for classes 2 and 4 (Subject2 and Subject5) reaches the maximum value for both training and testing;
- class 3 precision is higher in testing than in training;
- class 3 recall reaches 100% in testing and training;
- class 5 recall is higher in testing than in training.

Class 3, which corresponds to the cardiopathic attendee, has confirmed to be the easiest to be discriminated with regard to the others; however, both the overall and the single class performances are quite impressive. In this sense, the proposed approach has proven to be suitable for the application at hand.



	1	2	3	4	5	
1	213 18.3%	0 0.0%	0 0.0%	0 0.0%	2 0.2%	99.1% 0.9%
2	0 0.0%	93 8.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	1 0.1%	1 0.1%	370 31.7%	0 0.0%	2 0.2%	98.9% 1.1%
4	0 0.0%	0 0.0%	0 0.0%	95 8.1%	0 0.0%	100% 0.0%
5	3 0.3%	0 0.0%	0 0.0%	0 0.0%	387 33.2%	99.2% 0.8%
	98.2% 1.8%	98.9% 1.1%	100% 0.0%	100% 0.0%	99.0% 1.0%	99.2% 0.8%

(a) Training set

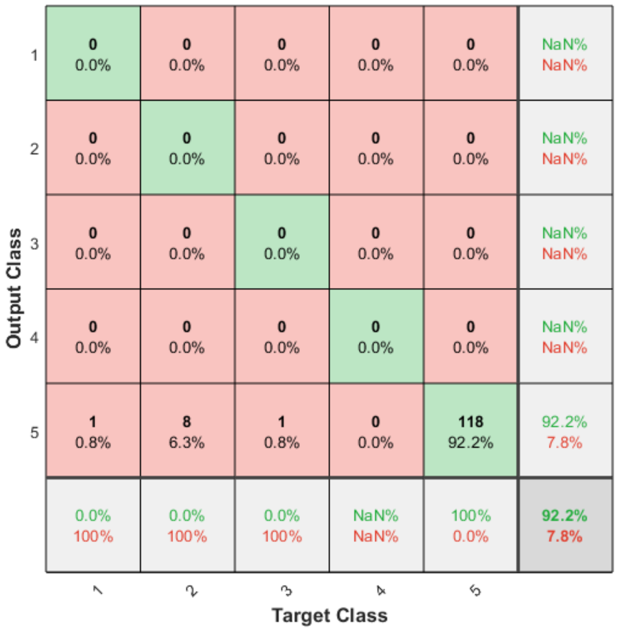
	1	2	3	4	5	
1	211 18.1%	0 0.0%	0 0.0%	0 0.0%	3 0.3%	98.6% 1.4%
2	0 0.0%	92 7.9%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	1 0.1%	373 32.0%	0 0.0%	0 0.0%	99.7% 0.3%
4	0 0.0%	0 0.0%	0 0.0%	95 8.2%	0 0.0%	100% 0.0%
5	5 0.4%	1 0.1%	0 0.0%	1 0.1%	382 32.8%	98.2% 1.8%
	97.7% 2.3%	97.9% 2.1%	100% 0.0%	99.0% 1.0%	99.2% 0.8%	99.1% 0.9%

(b) Test set

Figure 9.7: Shallow neural network confusion matrices: training (left) and testing (right).

9.3.1 Unknown subject

As a final test, the authentication robustness of the above network has been measured by feeding a novel, additional subject never seen from the network, neither in training nor in test. In this sense, the scope was simulating a real case scenario, where an intruder tries to fool the system by using a fake ID; here, this situation is modelled by means of the fifth class, which represent the rejected tokens. The *intruder* is a ten-years old child, who kindly provided 128 heartbeats. Fig. 9.8 yields the confusion matrix of the recall phase: the intruder is never misclassified, which means the biometric model is robust and can be exploited for authorization purposes.


Output Class
Target Class

	1	2	3	4	5	
1	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
3	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
5	1 0.8%	8 6.3%	1 0.8%	0 0.0%	118 92.2%	92.2% 7.8%
	0.0% 100%	0.0% 100%	0.0% 100%	NaN% NaN%	100% 0.0%	92.2% 7.8%

Figure 9.8: Shallow neural network confusion matrix: intruder simulation

9.4 Final considerations

ECG-based authentication yields greater security and safety in a world of risk; if combined with other kind of biometrics, it can result in the most powerful digital security strategy currently available. If this approach will gather sufficient attention, it will completely change the security paradigm, from external-based biometric to internal physiological data, almost impossible to forge. Moreover, R&D in this field have the potential to extrapolate human insights, which could have even more useful and interesting application than authentication. Indeed, extracting novel unique physiological and psychological parameters can have disruptive effect on current industries. For instance, ongoing researches [338] are deepening the employment of wearable devices for monitoring modifications in a person nervous system w.r.t.

external inputs: it was possible to relate pre-defined emotional states with physiological data gathered with a wristband. In this sense, it can be considered as an advancement towards the description of the physiology underlying emotions.

Chapter 10

Conclusions

One of the most widespread approaches to neural networks is their employment as black boxes; in this sense, the focus is more on achieving amazing performance on specific tasks, e.g. classification, rather than understanding the reasons behind them. It is like if the responsibility of understanding the problem at hand is shifted from humans to machines. Actually, it is true that neural systems map better the data manifold than human-designed models, especially if the input distribution is non-linear or is embedded in a high dimensional space; of course, it requires to find the architecture that is better suited at the purpose. Unfortunately, in the recent years, where IoT has become pervasive and tons and tons of data are produced every instant, it is easy to think the best strategy is gathering as much data as possible and, then, handle them using approaches like deep learning. The underlying assumption is, of course, that collecting and aggregating a huge amount of data, sometimes from a plethora of sources, would yield the sought piece of information, e.g. diagnose a disease. It must be noted that a deep approach requires, first, to collect a huge amount of data, which is not, at all, a trivial task, and, then, to perform several training iterations; indeed, deep learning is, definitely, one of the most time-consuming strategies. Is it really required to use such a great computational power, time and efforts just because the dataset is huge? Moreover, why does the deep learning works? What are its automatically extracted features? In other words, what kind of representation and abstraction is induced by using a multitude of layers? Why should the feature maps be shared to address different tasks (transfer learning)?

In the end, the real question to be answered is: when is it useful to add a new layer and when does it add only noise? Today, no building science exists to create and optimize a neural architecture for one task; indeed, it is a process of trials and errors, which may never converge to a proper solution.

10.1 Achieved results

This thesis tried to answer the above questions, which meant to tackle the lack of formalism and the black box approach of neural system design, by providing a scientific framework to analyse data and understand their topology before performing classification. In this sense, neural networks are used both to explore data manifolds and to determine which architecture is better tailored for a problem.

In a nutshell, the main theoretical achievements are:

- an innovative framework for intrinsic dimensionality estimation based on interpreting the CCA *dy-dx* diagram;
- an online neural network, the GCCA, for real-time projection, even for non-stationary dataset;
- the anisotropic G-EXIN novelty test for a more reliable manifold quantization;
- the GH-EXIN double vertical growth for optimizing the hierarchical clustering;
- the usage of neural networks as a tool for exploring the data manifold;
- an initial framework for interpreting deep learning layers;
- a novel paradigm for telemedicine based on low-cost, unobtrusive, user-friendly, wearables for continuous, remote, vital-sign monitoring.

The importance of the listed novelties has been confirmed by the experiments, whose most meaningful results are:

- non-stationary tracking and bearing pre-fault detection;
- machine lifelong learning;
- discover of co-regulation between CSAG and MAGEA gene families;
- proof that ECG can be used for blood pressure estimation (compliant with European regulation);
- relevance of handwriting temporal features for Parkinson's disease evolution;
- finding *temporal-like* attributes within the first convolutional layer of a deep model;
- the ECG-WATCH and VITAL-ECG devices, whose ECG quality and associated heartbeat estimation are compatible with certified medical gold standards;

- a novel algorithm for atrial fibrillation automatic detection, tailored for wearables;
- the use of the ECG-WATCH as a valid biometric tool for user authorization.

10.1.1 Manifold analysis

Before choosing an architecture, it would be better to understand data. Chap. 2 has shown how neural networks for data projection, such as PCA and CCA, can be successfully employed also for exploring the data manifold. PCA explained variance and its associated Pareto chart have been exploited to have an initial linear approximation of the intrinsic dimensionality; then, an innovative instruction manual for interpreting the CCA *dy-dx* diagram has been developed to measure the non-linear projection quality and estimate the *input set intrinsic dimensionality*.

10.1.2 Unsupervised online learning

Since CCA requires the computation of all the pairwise distances between input samples, it cannot be used on very huge datasets. At this aim, the onCCA and GCCA neural networks have been designed and described in Chap. 3. They represent two online incremental versions of CCA. Simulations have proven they possess the same unfolding property of CCA, even in presence of noise. Because of the use of seeds and bridges for handling and tracking novelties in the input distribution, GCCA is also able to deal with non-stationary data streams and to track its whole evolution over time; the latter feature has been proven in two prognostic applications, where GCCA was able to follow the whole machine lifecycle and, above all, detect the fault onsets.

Topological quantization

The input space quantization plays a crucial role because it provides the solid base on top of which perform data projection. At this purpose, the G-EXIN neural network is introduced in Chap. 4. It derives from the GCCA first layer of weights, but improves the input quantization by employing an additional anisotropic criterium for the novelty test, and the *activation flags* for assessing the dynamics of the input data stream; these two novelties together with the use of seeds and bridges make G-EXIN able to properly represent the input manifold, even along its borders (see Secs. 4.1.1 and 4.1.2). The experiment on the prognostic dataset has demonstrated G-EXIN is a valid tool for lifelong learning. Eventually, it employs only three hyperparameters because the others are automatically tuned by the network w.r.t. input data.

10.1.3 Hierarchical clustering

A special kind of unsupervised learning is the hierarchical clustering. Such an approach assumes the input information is stratified, i.e. several layers of data interpretation are possible; in this sense, the network builds a hierarchy (a tree), where the root corresponds to a coarse resolution, while each subsequent layer refines its ancestor. The GH-EXIN neural network, presented in Chap. 5, is a hierarchical divisive clustering technique, which introduces two innovative techniques for refining the hierarchy: data reallocation and connected graph test. The former is used for handling potential outliers at the end of each sG-EXIN epoch and reallocate samples which were wrongly assigned. The latter exploits the topology graph for finding connected components yielded by the horizontal growth, and improve the hierarchy by adding fictitious neurons for any CC; this mechanism enriches the hierarchical clustering as shown in the video sequences experiment of Sec. 5.2.5. Despite its complex structure, a full training costs only $O(Nlog_bN)$, where N is the number of samples in the whole training set and b is the mean tree branching factor. Several simulations show the quality of both the plain internal clustering and the overall hierarchy. Finally, the most significative application of GH-EXIN is on two-way clustering for gene expression analysis; here, by first clustering the genes and then the tissues, some relevant co-regulation between CSAG and MAGEA gene families emerged from the bicluster analysis presented in Sec. 5.2.6. In this sense, unsupervised learning proves to be an amazing tool to deal with data whose structure is unknown a priori and to discover their underlying patterns.

10.1.4 Supervised learning

When the problem at hand is well known a priori, a supervised approach is, typically, a better way to tackle it. Indeed, embedding an external knowledge in a neural system enriches the learning process and, above all, yields a more powerful tool. This is the case of medical applications, where physician expertise can be conveyed to the neural architecture by means of sample labelling. To this purpose, in Chap. 6, two biomedical clinical applications have been studied: the arterial blood pressure estimation (Sec. 6.1) and the Parkinson's handwriting feature analysis (Sec. 6.2).

ABP estimation

ABP is an important physiological parameter, which must be monitored to prevent and detect cardiovascular diseases. The relation between the ABP and the ECG and/or PPG has been studied using MLP, NNOE and LSTM neural networks, whose performances have been compared with IBP and NIBP gold standards. The proposed strategy overcomes both the invasive approach and the non-invasive mathematical models (see Sec. 6.1.5); indeed, despite it is still a non-invasive method, the predicted values resemble the invasive ones, but does not need a cuff to be inflated, which is quite uncomfortable for the users. In particular, the MLP predictive performance is quite promising, because outperforms the sphygmomanometer and is compliant with the ANSI/AAMI/ ISO 81060- 2:2013. The two recurrent neural networks do not show coherency on which is the best input; NNOE behaves better with ECG, while LSTM with PPG. In addition, also ECG-NNOE fulfils the ANSI/AAMI/ ISO 81060- 2:2013.

Parkinson's handwriting

Supervised learning has been also applied for analysing a Parkinson's handwriting database; the purpose was not to build the perfect classifier, but to assess the quality of the corresponding attributes. Neural networks have not been employed in a traditional way, i.e. to classify data, but, instead, as a tool for exploring the data manifold. The dataset biplot analysis (6.2.2) yields that the selected attributes only approximate the input manifold: the first two principal components barely coincide with the X and Y pen positions, which is natural, because most variance in writing is embedded in these two directions.Because the linear analysis was not conclusive, it was deepened using MLPs (see Sec. 6.2.3); first, raw data were used, then, because they were still not sufficient to separate the three subsets, a better discriminating group of features, based on temporal content, has been proposed. Even if the use of temporal features implies more training epochs than the raw case, its final training error is several orders of magnitude smaller. This consideration is strengthened by the classification rates and demonstrates that the temporal model better represents the manifold, i.e. the PD handwriting. This justifies the medical consideration about the handwriting temporal evolution relevance.

10.1.5 Deep learning analysis

In Chap. 7, supervised learning performance has been deeply analysed by studying the ECG classification results as input features change. Deep learning automatically extract features and provide good classification outcomes, but it is a black box and its results cannot be interpreted in a theoretical framework; in this sense, the best classification performance is assessed in Sec. 7.1.1 using a 1-D CNN. On the other side, shallow neural networks need a human-based feature engineering phase prior to their training but it is possible to interpret their outcomes w.r.t. the input features. To this purpose, in Sec. 7.2, six different training sets have been employed to test the MLP: ECG raw data, temporal attributes, eigenvector features and the corresponding CCA projections. The intrinsic dimensionality analysis of each dataset, together with the study of the corresponding confusion matrices, was used to determine the most meaningful subset of features for ECG arrhythmia classification. All the experiments showed a trade-off between the smallest amount of attributes and data linearity, which was even more evident in case of data projection. Because the best compromise in terms of accuracy and network simplicity was given by the temporal approach, these MLP attributes were sought within the features automatically extracted by the deep model. To this aim, a cross-correlation analysis for similarity assessing was performed. The study clearly proved that the CNN has automatically extracted, in the first layer, *temporal-like* features very close to the human-engineered attributes. Probably, in the subsequent layers the network further abstracted the features to improve the classification accuracy. In this sense, the proposed approach has paved the way of interpreting a convolutional layer by using certain choices of features; if applied to analyse all the deep layers, will, maybe, provide a theoretical framework for motivating transfer learning.

10.1.6 Wearable devices

Chap. 8 presented an application to real case medical scenarios: silent cardiovascular disease monitoring and hospital early discharge. There not exist CVD detection devices which are, at the same time, wearable, wireless, and can be used over a long-term period. In the latter scenario, no portable or wearable solution exists to allow patients being monitored remotely; currently, it means to stay in hospital wired connected to various medical instrumentations, whose results need to be interpreted by specialized personnel, which need to be physically near the machinery at hand. To tackle this issues a telemedicine approach based on wearable, low-cost, user-friendly devices has been proposed. Two novel tools, the ECG WATCH (see Sec. 8.1) and the VITAL-ECG (see Sec. 8.2) have been designed, built and tested w.r.t. the corresponding gold standard: to validate the device quality, data have been collected at the *Neuronica* and $Polito^{BIO}Med$ laboratories of Politecnico di Torino. Despite recordings last only ten seconds, the ECG quality and the heartbeat estimation have been demonstrated to be as good as gold standards; also, the embedded atrial fibrillation detection algorithm has proven to be valid in detecting Afib episodes. They can be comfortable wrist-worn all day; unobtrusiveness implies anytime, anywhere, recordings without the need to physically go to hospitals or cardiologists; no medical expertise is required for placing or usage, because a single touch on the relative mobile apps yields the acquisition in only ten seconds. In this sense, both devices can be employed for health monitoring and pathology recognition, such as silent atrial fibrillation, without any user medical expertise or going to a doctor; as a consequence, they can be also used for continuous monitoring of subjects at risk.

ECG WATCH for biometrics

In Chap. 9, a biometric authentication system based on the ECG WATCH has been presented. The key step for performing an effective access control is *authentication*, where the user identity is unequivocally determined. ECG biometrics is gathering more and more attention because it has been proven robust to circumvention, obfuscation and replay attacks. Due to its high-quality acquisitions, a perfect tool for this kind of authentication is the ECG-WATCH: the MLP has been trained for discriminating among different individuals. The dataset collected at the Politecnico di Torino (see Sec. 9.1) has been studied by means of PCA and CCA techniques to determine the manifold intrinsic dimensionality of each subset of recordings.The confusion matrix analysis confirmed the CVD attendee is the easiest to be discriminated; however, both the overall and the single class performances are worth of notice. As a final test, the authentication robustness of the above network has been measured by feeding a novel, additional subject never seen from the network, neither in training nor in test. The scope was simulating a real case scenario, where an intruder tries to fool the system by using a fake ID. The associated confusion matrix of the recall phase showed the intruder was never misclassified, which means the biometric model is robust and can be exploited for authorization purposes.

10.2 Future pathways

This thesis has analysed neural networks from different perspectives, which have been used both to learn data and to explore their manifold. In a certain sense, it can be argued that this work dealt with unorthodox use of existing neural networks. Innovative strategies have been proposed for intrinsic dimensionality estimation, online learning, data projection, hierarchical analysis and medical analysis. Both stationary and non-stationary input distributions have been examined. A set of novel architectures have been designed. Deep learning and shallow neural networks have been combined to deepen the induced representation of data and to explore how the machines learn. Finally, the proposed approach was exploited in a real case biometric application, where neural networks were embedded in novel designed wearables for telemedicine.

Despite the obtained results are interesting, meaningful and promising, all the above innovations were just the fist steps for a deeper understanding of how machines learn and to define a theoretical solid framework. Indeed, much more has still to be done; few examples are:

- the intrinsic dimensionality estimation is still performed by qualitative approach, it would better to define a quantitative measure for the projection quality; in this sense, the first step may be the automatization of the CCA hyperparameters;

- novel initial projection techniques have to be studied in order to improve online projection;
- G-EXIN hyperparameters must be completely automatized w.r.t. the input distribution and novel anisotropic criterion must be designed, especially for high-dimensionality input spaces;
- non-stationary detection using bridges needs to be deepened by means of an ad-hoc algorithm;
- subspace clustering must be refined to derive more advanced inferences, e.g. gene co-regulations, on input datasets;
- deep learning interpretation is just in its infancy;
- Grad-Cam and its variants could be used to determine if the ECG fiducial points are relevant also for machine learning or just from a biological perspective;
- clinical trials must be performed for the proposed wearable devices, with particular focus on biometrics;
- novel wearables can be designed to tackle other pathologies, such as diabetes and neurodegenerative diseases.

Finally, all the experiments, especially in the medical field, have to be expanded on a broader population to better assess the quality of all the derived conclusions.

Bibliography

[1] L Rundo. "Computer-Assisted Analysis of Biomedical Images". In: (2019).

[2] Renato Ferrero et al. "Ubiquitous fridge with natural language interaction". In: *2019 IEEE International Conference on RFID Technology and Applications (RFID-TA)*. IEEE. 2019, pp. 404–409.

[3] Harold Hotelling. "Analysis of a complex of statistical variables into principal components." In: *Journal of educational psychology* 24.6 (1933), p. 417.

[4] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[5] Pierre Demartines and Jeanny Hérault. "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets". In: *IEEE Transactions on neural networks* 8.1 (1997), pp. 148–154.

[6] Vladilen F Pisarenko. "The retrieval of harmonics from a covariance function". In: *Geophysical Journal International* 33.3 (1973), pp. 347–366.

[7] Ralph Schmidt. "Multiple emitter location and signal parameter estimation". In: *IEEE transactions on antennas and propagation* 34.3 (1986), pp. 276–280.

[8] I T Jolliffe. *Principal Component Analysis*. Springer, 2002.

[9] Wojtek Krzanowski. *Principles of multivariate analysis*. Vol. 23. OUP Oxford, 2000.

[10] Åke Björck and Gene H Golub. "Numerical methods for computing angles between linear subspaces". In: *Mathematics of computation* 27.123 (1973), pp. 579–594.

[11] Karl Ruben Gabriel. "The biplot graphic display of matrices with application to principal component analysis". In: *Biometrika* 58.3 (1971), pp. 453–467.

[12] Michael J Greenacre. *Biplots in practice*. Fundacion BBVA, 2010.

[13] Camille Jordan. "Essai sur la géométrie à n dimensions". In: *Bulletin de la Société mathématique de France* 3 (1875), pp. 103–174.

[14] John C Gower, Sugnet Gardner Lubbe, and Niel J Le Roux. *Understanding biplots*. John Wiley & Sons, 2011.

[15] Jigang Sun, Colin Fyfe, and Malcolm K Crowe. “Curvilinear component analysis and Bregman divergences.” In: *ESANN*. 2010.

[16] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. “Dimensionality reduction: a comparative”. In: *J Mach Learn Res* 10.66-71 (2009), p. 13.

[17] M. S. Rohith. *Cube*. The noun project. URL: https://thenounproject.com/term/cube/72558/ (visited on 04/17/2020).

[18] Robbe de Clerck. *Unfolded box*. The noun project. URL: https://thenounproject.com/term/unfolded-box/448675/ (visited on 04/17/2020).

[19] Michel Verleysen and Damien François. “The curse of dimensionality in data mining and time series prediction”. In: *International Work-Conference on Artificial Neural Networks*. Springer. 2005, pp. 758–770.

[20] Nikhil Sharma. *Curse of dimensionality*. Slideshare.net. URL: http://www.slideshare.net/NikhilSharma6/curse-of-dimensionality (visited on 04/21/2020).

[21] Kevin Lacker. *What is the curse of dimensionality?* Quora. URL: https://www.quora.com/What-is-the-curse-of-dimensionality (visited on 04/21/2020).

[22] Nandakishore Kambhatla and Todd K Leen. “Dimension reduction by local principal component analysis”. In: *Neural computation* 9.7 (1997), pp. 1493–1516.

[23] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. “Nonlinear component analysis as a kernel eigenvalue problem”. In: *Neural computation* 10.5 (1998), pp. 1299–1319.

[24] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. “Dimensionality reduction: a comparative”. In: *J Mach Learn Res* 10.66-71 (2009), p. 13.

[25] Giansalvo Cirrincione et al. “Power switch open-circuit fault detection in an interleaved DC/DC buck converter for electrolyzer applications by using curvilinear component analysis”. In: *2018 21st International Conference on Electrical Machines and Systems (ICEMS)*. IEEE. 2018, pp. 2221–2225.

[26] Terence D Sanger. “Optimal unsupervised learning in a single-layer linear feedforward neural network”. In: *Neural networks* 2.6 (1989), pp. 459–473.

[27] Juyang Weng, Yilu Zhang, and Wey-Shiuan Hwang. “Candid covariance-free incremental principal component analysis”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.8 (2003), pp. 1034–1040.

[28] Konstantinos I Diamantaras and Sun Yuan Kung. *Principal component neural networks: theory and applications*. John Wiley & Sons, Inc., 1996.

[29] Olga Kouropteva, Oleg Okun, and Matti Pietikäinen. “Incremental locally linear embedding”. In: *Pattern recognition* 38.10 (2005), pp. 1764–1767.

[30] Peng Jia et al. “Incremental Laplacian eigenmaps by preserving adjacent information between data points”. In: *Pattern Recognition Letters* 30.16 (2009), pp. 1457–1463.

[31] Housen Li et al. “Incremental manifold learning by spectral embedding methods”. In: *Pattern Recognition Letters* 32.10 (2011), pp. 1447–1455.

[32] Jianchang Mao and Anil K Jain. “Artificial neural networks for feature extraction and multivariate data projection”. In: *IEEE transactions on neural networks* 6.2 (1995), pp. 296–317.

[33] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *science* 313.5786 (2006), pp. 504–507.

[34] T. Kohonen, M. R. Schroeder, and T. S. Huang. *Self-Organizing Maps.* Springer, 2001.

[35] Xinjian Qiang, Guojian Cheng, and Zhen Li. “A survey of some classic self-organizing maps with incremental learning”. In: *2010 2nd International Conference on Signal Processing Systems.* Vol. 1. IEEE. 2010, pp. V1–804.

[36] Thomas Martinetz, Klaus Schulten, et al. “A" neural-gas" network learns topologies”. In: (1991).

[37] Bernd Fritzke. “A growing neural gas network learns topologies”. In: *Advances in neural information processing systems.* 1995, pp. 625–632.

[38] Thomas Martinetz and Klaus Schulten. “Topology representing networks”. In: *Neural Networks* 7.3 (1994), pp. 507–522.

[39] Agnes Vathy-Fogarassy, Attila Kiss, and Janos Abonyi. “Topology representing network map–a new tool for visualization of high-dimensional data”. In: *Transactions on computational science I.* Springer, 2008, pp. 61–84.

[40] Vladimir Tomenko. “Online dimensionality reduction using competitive learning and Radial Basis Function network”. In: *Neural networks* 24.5 (2011), pp. 501–511.

[41] Pablo A Estévez and Cristián J Figueroa. “Online data visualization using the neural gas network”. In: *Neural Networks* 19.6-7 (2006), pp. 923–934.

[42] Pablo A Estévez et al. “Nonlinear projection using geodesic distances and the neural gas network”. In: *International Conference on Artificial Neural Networks.* Springer. 2006, pp. 464–473.

[43] Nicolas Rougier and Yann Boniface. “Dynamic self-organising map”. In: *Neurocomputing* 74.11 (2011), pp. 1840–1847.

[44] Shen Furao, Tomotaka Ogura, and Osamu Hasegawa. "An enhanced self-organizing incremental neural network for online unsupervised learning". In: *Neural Networks* 20.8 (2007), pp. 893–903.

[45] Mohammed Ghesmoune, Mustapha Lebbah, and Hanene Azzag. "State-of-the-art on clustering data streams". In: *Big Data Analytics* 1.1 (2016), p. 13.

[46] Mohammed Ghesmoune, Hanene Azzag, and Mustapha Lebbah. "G-stream: Growing neural gas over data stream". In: *International Conference on Neural Information Processing*. Springer. 2014, pp. 207–214.

[47] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. "BIRCH: an efficient data clustering method for very large databases". In: *ACM Sigmod Record* 25.2 (1996), pp. 103–114.

[48] Philipp Kranen et al. "The ClusTree: indexing micro-clusters for anytime stream mining". In: *Knowledge and information systems* 29.2 (2011), pp. 249–272.

[49] Charu C Aggarwal et al. "A framework for clustering evolving data streams". In: *Proceedings 2003 VLDB conference*. Elsevier. 2003, pp. 81–92.

[50] Feng Cao et al. "Density-based clustering over an evolving data stream with noise". In: *Proceedings of the 2006 SIAM international conference on data mining*. SIAM. 2006, pp. 328–339.

[51] Charlie Isaksson, Margaret H Dunham, and Michael Hahsler. "SOStream: Self organizing density-based clustering over data stream". In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer. 2012, pp. 264–278.

[52] Giansalvo Cirrincione, Jeanny Hérault, and V Randazzo. "The on-line curvilinear component analysis (onCCA) for real-time data reduction". In: *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2015, pp. 1–8.

[53] Ray H White. "Competitive hebbian learning: Algorithm and demonstrations". In: *Neural Networks* 5.2 (1992), pp. 261–275.

[54] NASA Ames Prognostics Data Repository. *FEMTO Bearing Data Set*. NASA Ames Research Center, Moffett Field, CA. URL: https://ti.arc.nasa.gov/c/18/ (visited on 04/24/2020).

[55] Patrick Nectoux et al. "PRONOSTIA: An experimental platform for bearings accelerated degradation tests." In: 2012.

[56] Giansalvo Cirrincione, Vincenzo Randazzo, and Eros Pasero. "The Growing Curvilinear Component Analysis (GCCA) neural network". In: *Neural Networks* 103 (2018), pp. 108–117.

[57] Giansalvo Cirrincione, Vincenzo Randazzo, and Eros Pasero. “Growing Curvilinear Component Analysis (GCCA) for dimensionality reduction of non-stationary data”. In: *Multidisciplinary Approaches to Neural Computing*. Springer, 2018, pp. 151–160.

[58] Giansalvo Cirrincione et al. “Growing Curvilinear Component Analysis (GCCA) for stator fault detection in induction machines”. In: *Neural Approaches to Dynamics of Signal Exchanges*. Ed. by Anna Esposito et al. Springer, 2020, pp. 235–244.

[59] RR Kumar et al. “Analysis of stator faults in induction machines using growing curvilinear component analysis”. In: *2017 20th International Conference on Electrical Machines and Systems (ICEMS)*. IEEE. 2017, pp. 1–6.

[60] Rasa Karbauskaitė and Gintautas Dzemyda. “Multidimensional data projection algorithms saving calculations of distances”. In: *Information Technology And Control* 35.1 (2006).

[61] Maurizio Cirrincione, Marcello Pucci, and Gianpaolo Vitale. *Power converters and AC electrical drives with linear neural networks*. CRC Press, 2012.

[62] Samuel Kaski et al. “Trustworthiness and metrics in visualizing similarity of gene expression”. In: *BMC bioinformatics* 4.1 (2003), p. 48.

[63] Jarkko Venna and Samuel Kaski. “Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity”. In: *Proceedings of WSOM*. Vol. 5. Citeseer. 2005, pp. 695–702.

[64] Vincenzo Randazzo, Giansalvo Cirrincione, and Eros Pasero. “A New Unsupervised Neural Approach to Stationary and Non-stationary Data”. In: *Advances in Data Science: Methodologies and Applications*. Ed. by Gloria Phillips-Wren, Anna Esposito, and Lakhmi C. Jain. Cham: Springer International Publishing, 2021, pp. 125–145. ISBN: 978-3-030-51870-7. DOI: 10.1007/978-3-030-51870-7_7. URL: https://doi.org/10.1007/978-3-030-51870-7_7.

[65] Pietro Barbiero et al. “Topological Gradient-based Competitive Learning”. In: *arXiv preprint arXiv:2008.09477* (2020).

[66] Giansalvo Cirrincione et al. “Gradient-based Competitive Learning: Theory”. In: *arXiv preprint arXiv:2009.02799* (2020).

[67] Yoseph Linde, Andres Buzo, and Robert Gray. “An algorithm for vector quantizer design”. In: *IEEE Transactions on communications* 28.1 (1980), pp. 84–95.

[68] James MacQueen et al. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[69] Bernd Fritzke. “A self-organizing network that can follow non-stationary distributions”. In: *International conference on artificial neural networks*. Springer. 1997, pp. 613–618.

[70] Gail A. Carpenter and Stephen Grossberg. “The ART of adaptive pattern recognition by a self-organizing neural network”. In: *Computer* 21.3 (1988), pp. 77–88.

[71] Yann Prudent and Abdellatif Ennaji. “An incremental growing neural gas learns topologies”. In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* Vol. 2. IEEE. 2005, pp. 1211–1216.

[72] Mohamed-Rafik Bouguelia, Yolande Belaid, and Abdel Belaid. “An adaptive incremental clustering method based on the growing neural gas algorithm”. In: 2013.

[73] Vincenzo Randazzo et al. “Nonstationary topological learning with bridges and convex polytopes: the G-EXIN neural network”. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pp. 1–6.

[74] B.S. Everitt et al. *Cluster Analysis*. Wiley Series in Probability and Statistics. Wiley, 2011. ISBN: 9780470978443. URL: https://books.google.it/books?id=w3bE1kqd-48C.

[75] T Li et al. “A Structurally Adaptive Neural Tree for Recognition of Large Character Set”. In: *Proceedings of the 11th IAPR International Joint Conference on Pattern Recognition* 2 (Jan. 1992), pp. 187–190. DOI: 10.1109/ICPR.1992.201751.

[76] R. G. Adams, K. Butchart, and N. Davey. “Hierarchical Classification with a Competitive Evolutionary Neural Tree”. In: *Neural Networks* 12.3 (1999), pp. 541–551. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(99)00010-6. URL: http://www.sciencedirect.com/science/article/pii/S0893608099000106.

[77] Elena Samsonova, Joost Kok, and Ad Ijzerman. “TreeSOM: Cluster Analysis in the Self-Organizing Map”. In: *Neural networks : the official journal of the International Neural Network Society* 19 (July 2006), pp. 935–49. DOI: 10.1016/j.neunet.2006.05.003.

[78] Johan Himberg. “A SOM Based Cluster Visualization and its Application for False Coloring”. In: *IEEE Int. Joint Conf. on Neural Networks* 3 (Feb. 2000), 587–592 vol.3. DOI: 10.1109/IJCNN.2000.861379.

[79] M Venkat Reddy, Makara Vivekananda, and R U V N Satish. “Divisive Hierarchical Clustering with K-means and Agglomerative Hierarchical Clustering”. In: *International Journal of Computer Science Trends and Technology (IJCST)* 5 (5 Oct. 2017).

[80] GuoYan Hang et al. “A Hierarchical Clustering Algorithm Based on K-Means with Constraints”. In: *Innovative Computing ,Information and Control, International Conference on* 0 (Dec. 2009), pp. 1479–1482. DOI: 10.1109/ICICIC.2009.18.

[81] George Aloysius. “Efficient High Dimension Data Clustering using Constraint-Partitioning K-Means Algorithm”. In: *International Arab Journal of Information Technology* 10 (Nov. 2013).

[82] Madjid Khalilian et al. “A Novel K-Means Based Clustering Algorithm for High Dimensional Data Sets”. In: *International MultiConference of Engineers and Computer Scientists*. 2010, pp. 17–19. ISBN: 9789881701282.

[83] Alberto Forti and Gian Luca Foresti. “Growing Hierarchical Tree SOM: An Unsupervised Neural Network with Dynamic Topology”. In: *Neural networks* 19.10 (2006), pp. 1568–1580.

[84] Bernd Fritzke. “Growing cell structures–A self-organizing network for unsupervised and supervised learning”. In: *Neural Networks* 7 (1994), pp. 1441–1460.

[85] Vanco Burzevski and Chilukuri K. Mohan. “Hierarchical Growing Cell Structures”. In: *IEEE int. conference on neural networks*. 1996, pp. 207–218.

[86] A. Rauber, D. Merkl, and M. Dittenbach. “The Growing Hierarchical Self-Organizing Map: Exploratory Analysis of High-Dimensional Data”. In: *IEEE Transactions on Neural Networks* 13.6 (2002), pp. 1331–1341. ISSN: 1045-9227. DOI: 10.1109/TNN.2002.804221.

[87] Bernd Fritzke. “Growing Grid - A Self-Organizing Network with Constant Neighborhood Range and Adaptation Strength”. In: *Neural Processing Letters* 2.5 (1995), pp. 9–13. ISSN: 1573-773X. DOI: 10.1007/BF02332159. URL: https://doi.org/10.1007/BF02332159.

[88] Esteban J Palomo and Ezequiel López-rubio. “The Growing Hierarchical Neural Gas Self-Organizing Neural Network”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2016), pp. 1–10.

[89] Bernd Fritzke. “A growing neural gas network learns topologies”. In: *Advances in neural information processing systems*. 1995, pp. 625–632.

[90] Latifur Khan and Feng Luo. “Hierarchical Clustering for Complex Data”. In: *International Journal on Artificial Intelligence Tools* 14 (2005), pp. 791–810.

[91] Joaquin Dopazo and José Maria Carazo. “Phylogenetic Reconstruction Using an Unsupervised Growing Neural Network that Adopts the Topology of a Phylogenetic Tree”. In: *Journal of Molecular Evolution* 44.2 (1997), pp. 226–233.

[92] Giansalvo Cirrincione et al. "The GH-EXIN neural network for hierarchical clustering". In: *Neural Networks* 121 (2020), pp. 57–73.

[93] Gabriele Ciravegna et al. "Assessing discriminating capability of geometrical descriptors for 3D face recognition by using the GH-EXIN neural network". In: *Neural Approaches to Dynamics of Signal Exchanges*. Springer, 2020, pp. 223–233.

[94] Yizong Cheng and George M. Church. "Biclustering of Expression Data". In: *Proceedings. International Conference on Intelligent Systems for Molecular Biology* 8 (2000), pp. 93–103.

[95] Mohamed-Rafik Bouguelia, Yolande Belaid, and Abdel Belaid. "Online Unsupervised Neural-Gas Learning Method for Infinite Data Streams". In: *Pattern Recognition Applications and Methods*. Springer, 2015, pp. 57–70.

[96] David L. Davies and Donald W. Bouldin. "A Cluster Separation Measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1979). ISSN: 01628828. DOI: 10.1109/TPAMI.1979.4766909.

[97] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* (1987). ISSN: 03770427. DOI: 10.1016/0377-0427(87)90125-7.

[98] M. Reisslein et al. *YUV Video Sequences*. The noun project. URL: http://trace.eas.asu.edu/yuv/index.html (visited on 04/17/2020).

[99] Matthew A Turk and Alex P Pentland. "Face recognition using eigenfaces". In: *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 1991, pp. 586–591.

[100] Camil Demetrescu, Irene Finocchi, and Giuseppe F Italiano. *Algoritmi e strutture dati*. McGraw-Hill, 2004.

[101] Manuel Hidalgo et al. "Patient-Derived Xenograft Models: An Emerging Platform for Translational Cancer Research". In: *Cancer discovery* 4 (Sept. 2014), pp. 998–1013. DOI: 10.1158/2159-8290.CD-14-0001.

[102] John Tentler et al. "Patient-Derived Tumour Xenografts as Models for Oncology Drug Development". In: *Nature reviews. Clinical oncology* 9 (Apr. 2012), pp. 338–50. DOI: 10.1038/nrclinonc.2012.61.

[103] Annette Byrne Phd et al. "Interrogating Open Issues in Cancer Medicine with Patient-Derived Xenografts". In: *Nature reviews. Cancer* 17 (Sept. 2017). DOI: 10.1038/nrc.2017.85.

[104] "Illumina. Array-Based Gene Expression Analysis". 2011. URL: https://www.illumina.com/documents/products/datasheets/datasheet_gene_exp_analysis.pdf.

[105] Carla Boccaccio et al. “A Molecularly Annotated Model of Patient-Derived Colon Cancer Stem-Like Cells to Assess Genetic and Nongenetic Mechanisms of Resistance to Anti-EGFR Therapy”. In: *Clinical Cancer Research* 24 (Oct. 2017), clincanres.2151.2017. DOI: 10.1158/1078-0432.CCR-17-2151.

[106] Eugenia R Zanella et al. “IGF2 Is an Actionable Target that Identifies a Distinct Subpopulation of Colorectal Cancer Patients with Marginal Response to Anti-EGFR Therapies”. In: *Science translational medicine* 7 (Jan. 2015), 272ra12. DOI: 10.1126/scitranslmed.3010445.

[107] Andrea Bertotti et al. “The Genomic Landscape of Response to EGFR Blockade in Colorectal Cancer”. In: *Nature* 526 (Sept. 2015), pp. 263–267. DOI: 10.1038/nature14969.

[108] P. Barbiero et al. “Unsupervised Gene Identification in Colorectal Cancer”. In: *Quantifying and Processing Biomedical and Behavioral Signals*. Ed. by Anna Esposito et al. Springer International Publishing, 2019, pp. 219–227. ISBN: 978-3-319-95095-2. DOI: 10.1007/978-3-319-95095-2_21. URL: https://doi.org/10.1007/978-3-319-95095-2_21.

[109] P. Barbiero et al. “Supervised Gene Identification in Colorectal Cancer”. In: *Quantifying and Processing Biomedical and Behavioral Signals*. Ed. by Anna Esposito et al. Springer International Publishing, 2019, pp. 243–251. ISBN: 978-3-319-95095-2. DOI: 10.1007/978-3-319-95095-2_23. URL: https://doi.org/10.1007/978-3-319-95095-2_23.

[110] Pietro Barbiero et al. “DNA Microarray Classification: Evolutionary Optimization of Neural Network Hyper-parameters”. In: *Neural Approaches to Dynamics of Signal Exchanges*. Springer, 2020, pp. 305–311.

[111] Claudio Isella et al. “Selective Analysis of Cancer-Cell Intrinsic Transcriptional Traits Defines Novel Clinically Relevant Subtypes of Colorectal Cancer”. In: *Nature Communications* 8 (May 2017). DOI: 10.1038/ncomms15107.

[112] Barbiero Pietro et al. “Neural Biclustering in Gene Expression Analysis”. In: *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. 2017, pp. 1238–1243.

[113] Gad Getz, Erel Levine, and Eytan Domany. “Coupled Two-Way Clustering Analysis of Gene Microarray Data”. In: *Proceedings of the National Academy of Sciences* 97.22 (2000), pp. 12079–12084. DOI: 10.1073/pnas.210134797. eprint: https://www.pnas.org/content/97/22/12079.full.pdf.

[114] Edward J. Wegman. “Hyperdimensional data analysis using parallel coordinates”. In: *Journal of the American Statistical Association* (1990). DOI: 10.1080/01621459.1990.10474926.

[115] *CSAG3 gene (Protein Coding)*. GeneCards - The human gene database. URL: https://www.genecards.org/cgi-bin/carddisp.pl?gene=CSAG3 (visited on 04/17/2020).

[116] *MAGEA2 gene (Protein Coding)*. GeneCards - The human gene database. URL: https://www.genecards.org/cgi-bin/carddisp.pl?gene=MAGEA2 (visited on 04/17/2020).

[117] *High blood pressure: a public health problem*. World Health Organization (WHO). URL: http://www.emro.who.int/media/world-health-day/public-health-problem-factsheet-2013.html (visited on 04/17/2020).

[118] *Understanding Blood Pressure Readings*. American Heart Association (AHA). URL: https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings (visited on 04/17/2020).

[119] *Korotkoff Sounds - Taking Blood Pressure*. Practical Clinical Skills. URL: https://www.practicalclinicalskills.com/korotkoff-sounds (visited on 04/17/2020).

[120] MYM Wong, XY Zhang, and YT Zhang. “The cuffless arterial blood pressure estimation based on the timing-characteristics of second heart sound”. In: *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2006, pp. 1487–1488.

[121] Thomas G Pickering et al. “Recommendations for blood pressure measurement in humans and experimental animals: part 1: blood pressure measurement in humans: a statement for professionals from the Subcommittee of Professional and Public Education of the American Heart Association Council on High Blood Pressure Research”. In: *Hypertension* 45.1 (2005), pp. 142–161.

[122] Cyril Pellaton et al. “Accuracy testing of a new optical device for noninvasive estimation of systolic and diastolic blood pressure compared to intra-arterial measurements”. In: *Blood Pressure Monitoring* 25.2 (2020), pp. 105–109.

[123] Christoph Ilies et al. “Comparison of a continuous noninvasive arterial pressure device with invasive measurements in cardiovascular postsurgical intensive care patients: a prospective observational study”. In: *European Journal of Anaesthesiology (EJA)* 32.1 (2015), pp. 20–28.

[124] FJ Callaghan et al. “The relationship between arterial pulse-wave velocity and pulse frequency at different pressures”. In: *Journal of medical engineering & technology* 8.1 (1984), pp. 15–18.

[125] Ruiping Wang et al. “Cuff-free blood pressure estimation using pulse transit time and heart rate”. In: *2014 12th international conference on signal processing (ICSP)*. IEEE. 2014, pp. 115–118.

[126] A Hennig and A Patzak. “Continuous blood pressure measurement using pulse transit time”. In: *Somnologie-Schlafforschung und Schlafmedizin* 17.2 (2013), pp. 104–110.

[127] Parry Fung et al. “Continuous noninvasive blood pressure measurement by pulse transit time”. In: *The 26th annual international conference of the IEEE engineering in medicine and biology society*. Vol. 1. IEEE. 2004, pp. 738–741.

[128] Ümit Şentürk, Ibrahim Yücedağ, and Kemal Polat. “Repetitive neural network (RNN) based blood pressure estimation using PPG and ECG signals”. In: *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. IEEE. 2018, pp. 1–4.

[129] Zhongheng Zhang. “A gentle introduction to artificial neural networks”. In: *Annals of translational medicine* 4.19 (2016).

[130] Eric Chern-Pin Chua et al. “Towards using photo-plethysmogram amplitude to measure blood pressure during sleep”. In: *Annals of biomedical engineering* 38.3 (2010), pp. 945–954.

[131] John Allen. “Photoplethysmography and its application in clinical physiological measurement”. In: *Physiological measurement* 28.3 (2007), R1.

[132] Annunziata Paviglianiti et al. “Noninvasive Arterial Blood Pressure Estimation using ABPNet and VITAL-ECG”. In: *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE. 2020, pp. 1–5.

[133] Annunziata Paviglianiti et al. “Neural Recurrent Approches to Noninvasive Blood Pressure Estimation”. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020, pp. 1–7.

[134] Anthony G Barnston. “Correspondence among the correlation, RMSE, and Heidke forecast verification measures; refinement of the Heidke score”. In: *Weather and Forecasting* 7.4 (1992), pp. 699–709.

[135] George B Moody and Roger G Mark. “A database to support development and evaluation of intelligent intensive care monitoring”. In: *Computers in Cardiology 1996*. IEEE. 1996, pp. 657–660.

[136] Ary L Goldberger et al. “PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals”. In: *circulation* 101.23 (2000), e215–e220.

[137] Seymour Geisser. *Predictive inference*. Vol. 55. CRC press, 1993.

[138] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[139] Kenneth Levenberg. "A method for the solution of certain non-linear problems in least squares". In: *Quarterly of applied mathematics* 2.2 (1944), pp. 164–168.

[140] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

[141] Peter Magnus Nørgård et al. "Neural networks for modelling and control of dynamic systems-A practitioner's handbook". In: (2000).

[142] Hasim Sak, Andrew W Senior, and Françoise Beaufays. "Long short-term memory recurrent neural network architectures for large scale acoustic modeling". In: (2014).

[143] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. "Learning precise timing with LSTM recurrent networks". In: *Journal of machine learning research* 3.Aug (2002), pp. 115–143.

[144] Zijun Zhang. "Improved Adam optimizer for deep neural networks". In: *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS).* IEEE. 2018, pp. 1–2.

[145] Marie-Therese Heemels. "Neurodegenerative diseases". In: *Nature* 539.7628 (2016), pp. 179–180.

[146] Aaron D Gitler, Paraminder Dhillon, and James Shorter. "Neurodegenerative disease: models, mechanisms, and a new hope". In: *Disease Models and Mechanisms* 10.5 (2017), pp. 499–502.

[147] Maurice Masson, Henri Dehen, Jean Cambier, et al. *Neurologia.* Edra Masson, 2013.

[148] Beatice Rizzi et al. *La malattia di Parkinson Guida per pazienti e familiari.* Fondazioni Don Gnocchi, 2013.

[149] James Parkinson. "An essay on the shaking palsy". In: *The Journal of neuropsychiatry and clinical neurosciences* 14.2 (2002), pp. 223–236.

[150] Rajesh Pahwa, Kelly E Lyons, and William Koller. *Therapy of Parkinson's disease.* CRC Press, 2004.

[151] Andrew J Larner. "Addenbrooke's Cognitive Examination-Revised (ACE-R) in day-to-day clinical practice". In: *Age and ageing* 36.6 (2007), pp. 685–686.

[152] Marie-Charlotte Lepelley et al. "Age-related differences in sensorimotor representation of space in drawing by hand". In: *Clinical neurophysiology* 121.11 (2010), pp. 1890–1897.

[153] Serge Pinto and jean-luc Velay. "Handwriting as a marker for PD progression: a shift in paradigm". In: *Neurodegenerative disease management* 5.11 (2015), pp. 367–369.

[154] Sara Rosenblum et al. “Handwriting as an objective tool for Parkinson’s disease diagnosis”. In: *Journal of neurology* 260.9 (2013), pp. 2357–2361.

[155] Wei Lu and Jagath C Rajapakse. “Approach and applications of constrained ICA”. In: *IEEE transactions on neural networks* 16.1 (2005), pp. 203–212.

[156] Peter Drotár et al. “A new modality for quantitative evaluation of Parkinson’s disease: In-air movement”. In: *13th IEEE international conference on bioinformatics and bioengineering*. IEEE. 2013, pp. 1–4.

[157] Peter Drotár et al. “Contribution of different handwriting modalities to differential diagnosis of Parkinson’s disease”. In: *2015 IEEE international symposium on medical measurements and applications (MeMeA) proceedings*. IEEE. 2015, pp. 344–348.

[158] Zahari Abu Bakar et al. “Classification of Parkinson’s disease based on Multilayer Perceptrons (MLPs) Neural Network and ANOVA as a feature extraction”. In: *2012 IEEE 8th International Colloquium on Signal Processing and its Applications*. IEEE. 2012, pp. 63–67.

[159] Lukas Zoubek. “Introduction to educational data mining using MATLAB”. In: *Proc. Int. Conf. Tech. Comput. Prague*. 2009, pp. 1–7.

[160] Catherine Taleb et al. “Feature selection for an improved Parkinson’s disease identification based on handwriting”. In: *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*. IEEE. 2017, pp. 52–56.

[161] Vladimir Naumovich Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[162] Musa Peker et al. “A novel hybrid method for determining the depth of anesthesia level: Combining ReliefF feature selection and random forest algorithm (ReliefF+ RF)”. In: *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*. IEEE. 2015, pp. 1–8.

[163] Igor Kononenko. “Estimating attributes: analysis and extensions of RELIEF”. In: *European conference on machine learning*. Springer. 1994, pp. 171–182.

[164] Tatjana Liogienė and Gintautas Tamulevičius. “SFS feature selection technique for multistage emotion recognition”. In: *2015 IEEE 3rd Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*. IEEE. 2015, pp. 1–4.

[165] Marcos Faundez-Zanuy et al. “Biometric applications related to human beings: there is life beyond security”. In: *Cognitive Computation* 5.1 (2013), pp. 136–151.

[166] Vincenzo Randazzo et al. “Neural Feature Extraction for the Analysis of Parkinsonian Patient Handwriting”. In: *Progresses in Artificial Intelligence and Neural Systems*. Ed. by Anna Esposito et al. Singapore: Springer Singapore, 2021, pp. 243–253.

[167] Anthony Atkielski. *Electrocardiography*. Wikipiedia Foundation. URL: https://en.wikipedia.org/wiki/Electrocardiography (visited on 04/17/2020).

[168] Tanis Mar et al. “Optimization of ECG classification by means of feature selection”. In: *IEEE transactions on Biomedical Engineering* 58.8 (2011), pp. 2168–2177.

[169] J. Pan and W. J. Tompkins. “A Real-Time QRS Detection Algorithm”. In: *IEEE Transactions on Biomedical Engineering* BME-32.3 (1985), pp. 230–236.

[170] SS Mehta and NS Lingayat. “SVM-based algorithm for recognition of QRS complexes in electrocardiogram”. In: *IRBM* 29.5 (2008), pp. 310–317.

[171] Muhammad Abdullah Arafat, Abdul Wadud Chowdhury, and Md. Kamrul Hasan. “A simple time domain algorithm for the detection of ventricular fibrillation in electrocardiogram”. In: *Signal, Image and Video Processing* 5.1 (2011), pp. 1–10.

[172] Douglas A Coast et al. “An approach to cardiac arrhythmia analysis using hidden Markov models”. In: *IEEE Transactions on biomedical Engineering* 37.9 (1990), pp. 826–836.

[173] Liang-Yu Shyu, Ying-Hsuan Wu, and Weichih Hu. “Using wavelet transform and fuzzy neural network for VPC detection from the Holter ECG”. In: *IEEE Transactions on Biomedical Engineering* 51.7 (2004), pp. 1269–1273.

[174] Tanoy Debnath, Md Mehedi Hasan, and Tanwi Biswas. “Analysis of ECG signal and classification of heart abnormalities using Artificial Neural Network”. In: *2016 9th International Conference on Electrical and Computer Engineering (ICECE)*. IEEE. 2016, pp. 353–356.

[175] Sahar H El-Khafif and Mohamed A El-Brawany. “Artificial neural network-based automated ECG signal classifier”. In: *ISRN Biomedical Engineering* 2013 (2013).

[176] G Vijaya, Vinod Kumar, and HK Verma. “ANN-based QRS-complex analysis of ECG”. In: *Journal of medical engineering & technology* 22.4 (1998), pp. 160–167.

[177] Nitish V Thakor and Y-S Zhu. “Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection”. In: *IEEE transactions on biomedical engineering* 38.8 (1991), pp. 785–794.

[178] Preeti N Ranaware and Rohini A Deshpande. “Detection of arrhythmia based on discrete wavelet transform using artificial neural network and support vector machine”. In: *2016 International Conference on Communication and Signal Processing (ICCSP)*. IEEE. 2016, pp. 1767–1770.

[179] Muhammad Arif, Muhammad Usman Akram, et al. “Pruned fuzzy K-nearest neighbor classifier for beat classification”. In: *Journal of Biomedical Science and Engineering* 3.04 (2010), p. 380.

[180] S. G. Artis, R. G. Mark, and G. B. Moody. “Detection of atrial fibrillation using artificial neural networks”. In: *[1991] Proceedings Computers in Cardiology*. 1991, pp. 173–176.

[181] Gari D Clifford et al. “AF Classification from a short single lead ECG recording: the PhysioNet/Computing in Cardiology Challenge 2017”. In: *2017 Computing in Cardiology (CinC)*. IEEE. 2017, pp. 1–4.

[182] Nicolò Gambarotta et al. “A review of methods for the signal quality assessment to improve reliability of heart rate and blood pressures derived parameters”. In: *Medical & biological engineering & computing* 54.7 (2016), pp. 1025–1035.

[183] RR Karhe and B Bhagyashri. “Arrhythmia detection using one dimensional convolutional neural network”. In: *Int. J. Eng. Technol* 5.8 (2018).

[184] Nikhil Gawande and Alka Barhatte. “Heart diseases classification using convolutional neural network”. In: *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*. IEEE. 2017, pp. 17–20.

[185] Dan Li et al. “Classification of ECG signals based on 1D convolution neural network”. In: *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*. IEEE. 2017, pp. 1–6.

[186] Yunan Wu et al. “A comparison of 1-D and 2-D deep convolutional neural networks in ECG classification”. In: *arXiv preprint arXiv:1810.07088* (2018).

[187] Awni Y Hannun et al. “Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network”. In: *Nature medicine* 25.1 (2019), p. 65.

[188] RR Karhe and B. Badhe. “Heart Disease Classification Using One Dimensional Convolutional Neural Network”. In: *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering* 6.6 (2018).

[189] *MIT-BIH Arrhythmia Database*. PhysioNet. URL: https://www.physionet.org/content/mitdb/1.0.0/ (visited on 04/17/2020).

[190] George B Moody and Roger G Mark. “The impact of the MIT-BIH arrhythmia database”. In: *IEEE Engineering in Medicine and Biology Magazine* 20.3 (2001), pp. 45–50.

[191] *The Standard 12 Lead ECG*. Eccles Health Sciences Library University of Utah. URL: https://ecg.utah.edu/lesson/1 (visited on 04/17/2020).

[192] Jacopo Ferretti et al. “1-D Convolutional Neural Network for ECG Arrhythmia Classification”. In: *Progresses in Artificial Intelligence and Neural Systems*. Ed. by Anna Esposito et al. Singapore: Springer Singapore, 2021, pp. 269–279.

[193] Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. “Real-time patient-specific ECG classification by 1-D convolutional neural networks”. In: *IEEE Transactions on Biomedical Engineering* 63.3 (2015), pp. 664–675.

[194] Jacopo Ferretti et al. “Towards Uncovering Feature Extraction From Temporal Signals in Deep CNN: the ECG Case Study”. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020, pp. 1–6.

[195] S Karpagachelvi, M Arthanari, and M Sivakumar. “ECG feature extraction techniques-a survey approach”. In: *arXiv preprint arXiv:1005.0957* (2010).

[196] Hassan Ismail Fawaz et al. “Deep learning for time series classification: a review”. In: *Data Mining and Knowledge Discovery* 33.4 (2019), pp. 917–963.

[197] Munenori Uemura et al. “Feasibility of an AI-based measure of the hand motions of expert and novice surgeons”. In: *Computational and mathematical methods in medicine* 2018 (2018).

[198] Vitoantonio Bevilacqua et al. “On the comparison of NN-based architectures for diabetic damage detection in retinal images”. In: *Journal of Circuits, Systems, and Computers* 18.08 (2009), pp. 1369–1380.

[199] Antonio Brunetti et al. “Computer vision and deep learning techniques for pedestrian detection and tracking: A survey”. In: *Neurocomputing* 300 (2018), pp. 17–33.

[200] Vitoantonio Bevilacqua et al. “An innovative neural network framework to classify blood vessels and tubules based on Haralick features evaluated in histological images of kidney biopsy”. In: *Neurocomputing* 228 (2017), pp. 143–153.

[201] Valber Cesar Cavalcanti Roza, Ana Maria de Almeida, and Octavian Adrian Postolache. “Design of an artificial neural network and feature extraction to identify arrhythmias from ECG”. In: *2017 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE. 2017, pp. 391–396.

[202] David Cuesta-Frau, Juan C Perez-Cortes, and Gabriela Andreu-Garcia. "Clustering of electrocardiograph signals in computer-aided Holter analysis". In: *Computer methods and programs in Biomedicine* 72.3 (2003), pp. 179–196.

[203] Elif Derya Ubeyli. "Combining recurrent neural networks with eigenvector methods for classification of ECG beats". In: *Digital Signal Processing* 19.2 (2009), pp. 320–329.

[204] Elif Derya Ubeyli, Dean Cvetkovic, and Irena Cosic. "Analysis of human PPG, ECG and EEG signals by eigenvector methods". In: *Digital Signal Processing* 20.3 (2010), pp. 956–963.

[205] Walid Zgallai et al. "Music-based bispectrum detector: a novel non-invasive detection method for overlapping fetal and mother ECG signals". In: *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.'Magnificent Milestones and Emerging Opportunities in Medical Engineering'(Cat. No. 97CH36136)*. Vol. 1. IEEE. 1997, pp. 72–75.

[206] Giansalvo Cirrincione, Vincenzo Randazzo, and Eros Pasero. "A Neural Based Comparative Analysis for Feature Extraction from ECG Signals". In: *Neural Approaches to Dynamics of Signal Exchanges*. Springer, 2020, pp. 247–256.

[207] Jae-Chern Yoo and Tae Hee Han. "Fast normalized cross-correlation". In: *Circuits, systems and signal processing* 28.6 (2009), p. 819.

[208] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* 128.2 (2019), pp. 336–359. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://dx.doi.org/10.1007/s11263-019-01228-7.

[209] Aditya Chattopadhay et al. "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 839–847.

[210] Daniel Omeiza et al. *Smooth Grad-CAM++: An Enhanced Inference Level Visualization Technique for Deep Convolutional Neural Network Models*. 2019. arXiv: 1908.01224 [cs.CV].

[211] Haofan Wang et al. "Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 24–25.

[212] CM De Dominicis et al. "Evaluation of Bluetooth Hands-Free profile for sensors applications in smartphone platforms". In: *2012 IEEE Sensors Applications Symposium Proceedings*. IEEE. 2012, pp. 1–6.

[213] Wazir Zada Khan et al. "Mobile phone sensing systems: A survey". In: *IEEE Communications Surveys & Tutorials* 15.1 (2012), pp. 402–427.

[214] Alessandro Depari et al. "A wearable smartphone-based system for electrocardiogram acquisition". In: *2014 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE. 2014, pp. 1–6.

[215] CM De Dominicis et al. "Acquisition and elaboration of cardiac signal in android Smartphone devices". In: *2014 IEEE Sensors Applications Symposium (SAS)*. IEEE. 2014, pp. 83–88.

[216] Claudio Crema et al. "The WearPhone: changing smartphones into multichannel vital signs monitors". In: *2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE. 2016, pp. 1–6.

[217] C Crema et al. "Smartphone-based system for the monitoring of vital parameters and stress conditions of amatorial racecar drivers". In: *2015 IEEE SENSORS*. IEEE. 2015, pp. 1–4.

[218] Stephan Mühlbacher-Karrer et al. "A driver state detection system—Combining a capacitive hand detection sensor with physiological sensors". In: *IEEE Transactions on Instrumentation and Measurement* 66.4 (2017), pp. 624–636.

[219] Minho Choi et al. "Wearable device-based system to monitor a driver's stress, fatigue, and drowsiness". In: *IEEE Transactions on Instrumentation and Measurement* 67.3 (2017), pp. 634–645.

[220] Fei Deng et al. "Design and implementation of a noncontact sleep monitoring system using infrared cameras and motion sensor". In: *IEEE Transactions on Instrumentation and Measurement* 67.7 (2018), pp. 1555–1563.

[221] Bappaditya Mandal et al. "Towards detection of bus driver fatigue based on robust visual analysis of eye state". In: *IEEE Transactions on Intelligent Transportation Systems* 18.3 (2016), pp. 545–557.

[222] Emina Alickovic and Abdulhamit Subasi. "Ensemble SVM method for automatic sleep stage classification". In: *IEEE Transactions on Instrumentation and Measurement* 67.6 (2018), pp. 1258–1265.

[223] F Axisa, A Dittmar, and G Delhomme. "Smart clothes for the monitoring in real time and conditions of physiological, emotional and sensorial reactions of human". In: *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439)*. Vol. 4. IEEE. 2003, pp. 3744–3747.

[224] Maged N Kamel Boulos et al. "How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from eCAALYX". In: *Biomedical engineering online* 10.1 (2011), p. 24.

[225] *Cardiovascular diseases (CVDs)*. World Health Organization. URL: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds) (visited on 04/17/2020).

[226] *The top 10 causes of death*. World Health Organization. URL: https://www.who.int/en/news-room/fact-sheets/detail/the-top-10-causes-of-death (visited on 04/17/2020).

[227] Nick Townsend et al. "Cardiovascular disease in Europe: epidemiological update 2016". In: *European heart journal* 37.42 (2016), pp. 3232–3245.

[228] Shervin Shirmohammadi et al. "Instrumentation and measurement in medical, biomedical, and healthcare systems". In: *IEEE Instrumentation & Measurement Magazine* 19.5 (2016), pp. 6–12.

[229] Colin C. Schamroth. *An introduction to electrocardiography 7th ed.* Blackwell Scientific Pub, Oxford, 1990.

[230] S Yusuf et al. "The entry ECG in the early diagnosis and prognostic stratification of patients with suspected acute myocardial infarction". In: *European Heart Journal* 5.9 (1984), pp. 690–696.

[231] David W Smith, Douglas Nowacki, and John KJ Li. "ECG T-wave monitor for potential early detection and diagnosis of cardiac arrhythmias". In: *Cardiovascular Engineering* 10.4 (2010), pp. 201–206.

[232] Pravin Pawar et al. "A framework for the comparison of mobile patient monitoring systems". In: *Journal of biomedical informatics* 45.3 (2012), pp. 544–556.

[233] Laszlo Szilagyi et al. "Quick ECG analysis for on-line Holter monitoring systems". In: *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2006, pp. 1678–1681.

[234] Keisuke Kasahara et al. "Sudden cardiac arrest risk stratification based on 24-hour Holter ECG statistics". In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2015, pp. 5817–5820.

[235] SM Szilágyi et al. "Holter Telemetry in the Study of Heart Rate Variability". In: *Romanian Heart Journal* 2.6 (1996), p. 143.

[236] Marjan Gusev and Ana Guseva. "State-of-the-art of cloud solutions based on ECG sensors". In: *IEEE EUROCON 2017-17th International Conference on Smart Technologies*. IEEE. 2017, pp. 501–506.

[237] Jocelyne Fayn and Paul Rubel. "Toward a personal health society in cardiology". In: *IEEE Transactions on Information technology in Biomedicine* 14.2 (2009), pp. 401–409.

[238] Claudio De Capua, Antonella Meduri, and Rosario Morello. "A smart ECG measurement system based on web-service-oriented architecture for telemedicine applications". In: *IEEE Transactions on Instrumentation and Measurement* 59.10 (2010), pp. 2530–2538.

[239] Chin-Teng Lin et al. "An intelligent telecardiology system using a wearable and wireless ECG to detect atrial fibrillation". In: *IEEE Transactions on Information Technology in Biomedicine* 14.3 (2010), pp. 726–733.

[240] Luca Fanucci et al. "Sensing devices and sensor signal processing for remote monitoring of vital signs in CHF patients". In: *IEEE Transactions on Instrumentation and Measurement* 62.3 (2012), pp. 553–569.

[241] Bahareh Taji et al. "An ECG monitoring system using conductive fabric". In: *2013 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE. 2013, pp. 309–314.

[242] Ebrahim Nemati, M Jamal Deen, and Tapas Mondal. "A wireless wearable ECG sensor for long-term applications". In: *IEEE Communications Magazine* 50.1 (2012), pp. 36–43.

[243] Chulsung Park et al. "An ultra-wearable, wireless, low power ECG monitoring system". In: *2006 IEEE biomedical circuits and systems conference*. IEEE. 2006, pp. 241–244.

[244] Isabel G Trindade et al. "Design and evaluation of novel textile wearable systems for the surveillance of vital signals". In: *Sensors* 16.10 (2016), p. 1573.

[245] Murat A Yokus and Jesse S Jur. "Fabric-based wearable dry electrodes for body surface biopotential recording". In: *IEEE Transactions on Biomedical Engineering* 63.2 (2015), pp. 423–430.

[246] Arsalan Mohsen Nia et al. "Energy-efficient long-term continuous personal health monitoring". In: *IEEE Transactions on Multi-Scale Computing Systems* 1.2 (2015), pp. 85–98.

[247] Mirza Mansoor Baig, Hamid Gholamhosseini, and Martin J Connolly. "A comprehensive survey of wearable and wireless ECG monitoring systems for older adults". In: *Medical & biological engineering & computing* 51.5 (2013), pp. 485–495.

[248] *ECG PALMARE PM-10 Bluetooth*. GIMA. URL: https://www.gimaitaly.com/prodotti.asp?sku=33246&dept_selected=5801&dept_id=5801 (visited on 04/17/2020).

[249] *Apple Watch*. Apple. URL: https://www.apple.com/it/shop/buy-watch/apple-watch (visited on 04/17/2020).

[250] *Tracciare un ECG con l'app ECG su Apple Watch Series 4 o modelli successivi*. Apple. URL: https://support.apple.com/it-it/HT208955 (visited on 04/17/2020).

[251] *KardiaMobile*. AliveCor. URL: https://www.alivecor.it/ (visited on 04/17/2020).

[252] *Kardia Mobile*. Quiver. URL: https://quiver.store/prodotto/kardia/ (visited on 04/17/2020).

[253] *ECG check*. Cardiac Design. URL: https://www.cardiacdesigns.com/ (visited on 04/17/2020).

[254] *QardioCore*. Qardio. URL: https://www.getqardio.com/it/qardiocore-wearable-ecg-ekg-monitor-iphone/ (visited on 04/17/2020).

[255] Annunziata Paviglianiti and Eros Pasero. "VITAL-ECG: a de-bias algorithm embedded in a gender-immune device". In: *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT*. IEEE. 2020, pp. 314–318.

[256] Vincenzo Randazzo, Jacopo Ferretti, and Eros Pasero. "ECG WATCH: a real time wireless wearable ECG". In: *2019 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE. 2019, pp. 1–6.

[257] Eros Pasero, Eugenio Balzanelli, and Federico Caffarelli. "Intruder recognition using ECG signal". In: *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2015, pp. 1–8.

[258] *Einthoven's triangle*. Miller-Keane Encyclopedia, Dictionary of Medicine Nursing, and Allied Health Seventh Edition. 2003. URL: https://medical-dictionary.thefreedictionary.com/Einthoven%5C%27s+triangle (visited on 04/17/2020).

[259] *Graphical representation of Einthoven's triangle*. Npatchett. URL: https://commons.wikimedia.org/w/index.php?curid=39235282 (visited on 04/17/2020).

[260] Nitish V Thakor and John G Webster. "Ground-free ECG recording with two electrodes". In: *IEEE Transactions on Biomedical Engineering* 12 (1980), pp. 699–704.

[261] Bruce B Winter and John G Webster. "Driven-right-leg circuit design". In: *IEEE Transactions on Biomedical Engineering* 1 (1983), pp. 62–66.

[262] Jozef Surda et al. "Spectral properties of ECG signal". In: *2007 17th International Conference Radioelektronika*. IEEE. 2007, pp. 1–5.

[263] Dong Jingwei and Jiang Wenwen. "Design of digital filter on ecg signal processing". In: *2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*. IEEE. 2015, pp. 1272–1275.

[264] *B105 and B125 Patient Monitors*. GE Healthcare. URL: https://www.gehealthcare.com/products/patient-monitoring/patient-monitors/b105-and-b125-patient-monitors (visited on 04/17/2020).

[265] *ProSim 4 Vital Sign and ECG Simulator*. Fluke Biomedical. URL: https://www.flukebiomedical.com/products/biomedical-test-equipment/patient-monitor-simulators/prosim-4-vital-signs-patient-simulator (visited on 04/17/2020).

[266] Douglas G Altman and J Martin Bland. "Measurement in medicine: the analysis of method comparison studies". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 32.3 (1983), pp. 307–317.

[267] ALLEN M SCHER and ALLAN C YOUNG. "Frequency analysis of the electrocardiogram". In: *Circulation Research* 8.2 (1960), pp. 344–346.

[268] *Other Conditions Related to Heart Disease*. Centers for Disease Control and Prevention. URL: https://www.cdc.gov/heartdisease/other_conditions.htm (visited on 04/17/2020).

[269] Thomas M Munger, Li-Qun Wu, and Win K Shen. "Atrial fibrillation". In: *Journal of biomedical research* 28.1 (2014), pp. 1–17.

[270] Massimo Zoni-Berisso et al. "Epidemiology of atrial fibrillation: European perspective". In: *Clinical epidemiology* 6 (2014), p. 213.

[271] Andrew R Houghton and David Gray. *Chamberlain's Symptoms and Signs in Clinical Medicine, An Introduction to Medical Diagnosis*. CRC Press, 2010.

[272] R Verma et al. "Day case and short stay surgery: 2". In: *Anaesthesia* 66.5 (2011), pp. 417–434.

[273] WHO Patient Safety, World Health Organization, et al. *WHO guidelines for safe surgery: 2009: safe surgery saves lives*. WHO/IER/PSP/2008.08-1E. World Health Organization, 2009.

[274] Daniel J Quemby and Mary E Stocker. "Day surgery development and practice: key factors for a successful pathway". In: *Continuing Education in Anaesthesia, Critical Care & Pain* 14.6 (2014), pp. 256–261.

[275] C Hamer, K Holmes, and M Stocker. "A Generic Process for Transferring Procedures to the Day Case Setting-The Torbay Hospital Proposal". In: *Journal of One Day Surgery* 18.1 (2008), p. 09.

[276] Douglas W Wilmore and Henrik Kehlet. "Management of patients in fast track surgery". In: *Bmj* 322.7284 (2001), pp. 473–476.

[277] Kaiser Family Foundation. *Hospital adjusted expenses per inpatient day.* 2015.

[278] F Epelde, ML Iglesias-Lepine, and L Anarte. "Economic crisis: Cost and effectiveness of short stay hospital units". In: *Anales del sistema sanitario de Navarra*. Vol. 35. 3. 2012, p. 469.

[279] Zahra Tolou-Ghamari, Vahid Shaygannejad, and Fariborz Khorvash. "Preliminary investigation of economics issues in hospitalized patients with stroke". In: *International journal of preventive medicine* 4.Suppl 2 (2013), S338.

[280] *ECG CONTEC 1200G - 12 canali con display con WI-FI.* GIMA. URL: https://www.gimaitaly.com/prodotti.asp?sku=33223&dept_selected=5802&dept_id=5802 (visited on 04/17/2020).

[281] *ECG CARDIO 7 - 12 canali con touch screen.* GIMA. URL: https://www.gimaitaly.com/prodotti.asp?sku=33357&dept_selected=5802&dept_id=5802 (visited on 04/17/2020).

[282] *ECG CARDIOGIMA 12: 3-6-12 canali.* GIMA. URL: https://www.gimaitaly.com/prodotti.asp?sku=33354&dept_selected=5802&dept_id=5802 (visited on 04/17/2020).

[283] *PULSOXIMETRO iHEALTH.* GIMA. URL: https://www.gimaitaly.com/prodotti.asp?sku=23525&dept_selected=82&dept_id=82 (visited on 04/17/2020).

[284] *Family termometro digitale.* microlife. URL: https://www.microlife.it/prodotti-domiciliari/febbre/termometri-digitali/mt-16f1 (visited on 04/17/2020).

[285] *Charge 4.* Fitbit. URL: https://www.fitbit.com/it/charge4 (visited on 04/17/2020).

[286] Vincenzo Randazzo, Jacopo Ferretti, and Eros Pasero. "A Wearable Smart Device to Monitor Multiple Vital Parameters—VITAL ECG". In: *Electronics* 9.2 (2020), p. 300.

[287] Vincenzo Randazzo, Eros Pasero, and Silvio Navaretti. "VITAL-ECG: a portable wearable hospital". In: *2018 IEEE Sensors Applications Symposium (SAS)*. IEEE. 2018, pp. 1–6.

[288] *INA333 - Low-Power, Zero-Drift, Precision Instrumentation Amplifier.* Texas Instrument. URL: https://www.ti.com/product/INA333 (visited on 04/17/2020).

[289] *OPA4330 - 1.8V, 35µA, micro Power, Precision, Zero Drift CMOS Op Amp.* Texas Instrument. URL: http://www.ti.com/product/OPA4330 (visited on 04/17/2020).

[290] S Nakajima, K Ikeda, H Nishioka, et al. "Clinical application of a new (fingertip type) pulse wave oximeter". In: *Jpn J Surg* 41 (1979), pp. 57–61.

[291] *MAX30102: High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health*. Maxim Integrated. URL: https://www.maximintegrated.com/en/products/interface/sensor-interface/MAX30102.html (visited on 04/17/2020).

[292] *HTS221: Capacitive Digital Sensor for Relative Humidity and Temperature*. ST Microelectronics. URL: https://www.st.com/en/mems-and-sensors/hts221.html (visited on 04/17/2020).

[293] *MPU-9250 Nine-Axis (Gyro + Accelerometer + Compass) MEMS MotionTracking™ Device*. TDK InvenSense. URL: https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/ (visited on 04/17/2020).

[294] *MAX1759: Buck/Boost Regulating Charge Pump in µMAX*. Maxim Integrated. URL: https://www.maximintegrated.com/en/products/MAX1759 (visited on 04/17/2020).

[295] *MAX1555: SOT23, Dual-Input, USB/AC Adapter, 1-Cell Li+ Battery Chargers*. Maxim Integrated. URL: https://www.maximintegrated.com/en/products/power/battery-management/MAX1555.html (visited on 04/17/2020).

[296] *Micro-USB*. Wikipiedia Foundation. URL: https://it.wikipedia.org/wiki/Micro-USB (visited on 04/17/2020).

[297] *CC2640R2F: SimpleLink Bluetooth® low energy Wireless MCU*. Texas Instrument. URL: http://www.ti.com/product/CC2640R2F (visited on 04/17/2020).

[298] *Bluetooth Specifications*. Bluetooth SIG. URL: https://www.bluetooth.com/specifications/ (visited on 04/17/2020).

[299] *MAC 2000 Resting ECG*. GE Healthcare. URL: https://www.gehealthcare.com/products/mac-2000 (visited on 04/17/2020).

[300] J SCOTT BUTTERWORTH and JOHN J THORPE. "On evaluating the Einthoven triangle theory". In: *Circulation* 3.6 (1951), pp. 923–925.

[301] Ravi S Sandhu and Pierangela Samarati. "Access control: principle and practice". In: *IEEE communications magazine* 32.9 (1994), pp. 40–48.

[302] Stephen Krawczyk and Anil K Jain. "Securing electronic medical records using biometric authentication". In: *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer. 2005, pp. 1110–1119.

[303] Federal Trade Commission et al. "Consumer sentinel network data book 2018". In: *Retrieved from* (2019).

[304] *Identity theft reports - theft types over time*. Federal Trade Commission. URL: https://public.tableau.com/profile/federal.trade.commission#!/vizhome/IdentityTheftReports/TheftTypesOverTime (visited on 05/27/2020).

[305] Anil K Jain, Arun Ross, and Salil Prabhakar. "An introduction to biometric recognition". In: *IEEE Transactions on circuits and systems for video technology* 14.1 (2004), pp. 4–20.

[306] John Chirillo and Scott Blaul. *Implementing biometric security*. Hungry Minds, Incorporated, 2003.

[307] RM Bolle et al. "Guide to Biometrics, Springer". In: (2004).

[308] Alejandro Riera et al. "STARFAST: A wireless wearable EEG/ECG biometric system based on the ENOBIO sensor". In: *Proceedings of the international workshop on wearable micro and nanosystems for personalised health*. 2008.

[309] Lena Biel et al. "ECG analysis: a new approach in human identification". In: *IEEE Transactions on Instrumentation and Measurement* 50.3 (2001), pp. 808–812.

[310] Foteini Agrafioti et al. "Heart biometrics: Theory, methods and applications". In: *Biometrics*. InTech Shanghai, China, 2011, pp. 199–216.

[311] Jianfeng Hu and Zhendong Mu. "EEG authentication system based on autoregression coefficients". In: *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. IEEE. 2016, pp. 1–5.

[312] Anthony Lee and Younghyun Kim. "Photoplethysmography as a form of biometric authentication". In: *2015 IEEE SENSORS*. IEEE. 2015, pp. 1–2.

[313] Yongjin Wang et al. "Analysis of human electrocardiogram for biometric recognition". In: *EURASIP journal on Advances in Signal Processing* 2008.1 (2007), p. 148658.

[314] Adrian Condon and Grace Willatt. "ECG biometrics: the heart of data-driven disruption?" In: *Biometric Technology Today* 2018.1 (2018), pp. 7–9.

[315] Ikenna Odinaka et al. "ECG biometric recognition: A comparative analysis". In: *IEEE Transactions on Information Forensics and Security* 7.6 (2012), pp. 1812–1824.

[316] John M Irvine et al. "A new biometric: human identification from circulatory function". In: *Joint Statistical Meetings of the American Statistical Association, San Francisco*. 2003.

[317] Steven A Israel et al. "ECG to identify individuals". In: *Pattern recognition* 38.1 (2005), pp. 133–142.

[318] Zhaomin Zhang and Daming Wei. “A new ECG identification method using bayes’ teorem”. In: *TENCON 2006-2006 IEEE Region 10 Conference*. IEEE. 2006, pp. 1–4.

[319] Sunil Kumar Singla and Ankit Sharma. “ECG as biometric in the automated world”. In: *International Journal of Computer Science & Communication* 1.2 (2010), pp. 281–283.

[320] Masaki Kyoso and Akihiko Uchiyama. “Development of an ECG identification system”. In: *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 4. IEEE. 2001, pp. 3721–3723.

[321] JM Irvine et al. “Heart rate variability: a new biometric for human identification”. In: *Proceedings of the International Conference on Artificial Intelligence (IC-AI’01)*. 2001, pp. 1106–1111.

[322] John M Irvine and Steven A Israel. “A sequential procedure for individual identity verification using ECG”. In: *EURASIP Journal on Advances in Signal Processing* 2009.1 (2009), p. 243215.

[323] Yogendra Narain Singh and Phalguni Gupta. “Correlation-based classification of heartbeats for individual identification”. In: *Soft Computing* 15.3 (2011), pp. 449–460.

[324] Maryamsadat Hejazi et al. “ECG biometric authentication based on non-fiducial approach using kernel methods”. In: *Digital Signal Processing* 52 (2016), pp. 72–86.

[325] Carmen Camara, Pedro Peris-Lopez, and Juan E Tapiador. “Human identification using compressed ECG signals”. In: *Journal of medical systems* 39.11 (2015), p. 148.

[326] Qingxue Zhang, Dian Zhou, and Xuan Zeng. “HeartID: A multiresolution convolutional neural network for ECG-based biometric human identification in smart health applications”. In: *Ieee Access* 5 (2017), pp. 11805–11816.

[327] M Bassiouni et al. “A machine learning technique for person identification using ECG signals”. In: *Int. J. Appl. Phys* 1 (2016), pp. 37–41.

[328] Maryamsadat Hejazi et al. “Non-fiducial based ECG biometric authentication using one-class support vector machine”. In: *2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. IEEE. 2017, pp. 190–194.

[329] Robin Tan and Marek Perkowski. “Toward improving electrocardiogram (ECG) biometric verification using mobile sensors: A two-stage classifier approach”. In: *Sensors* 17.2 (2017), p. 410.

[330] Sara S Abdeldayem and Thirimachos Bourlai. "ECG-based human authentication using high-level spectro-temporal signal features". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 4984–4993.

[331] Turky N Alotaiby et al. "ECG-Based Subject Identification Using Statistical Features and Random Forest". In: *Journal of Sensors* 2019 (2019).

[332] Di Wang et al. "A Novel Electrocardiogram Biometric Identification Method Based on Temporal-Frequency Autoencoding". In: *Electronics* 8.6 (2019), p. 667.

[333] Felipe Gustavo Silva Teodoro, Sarajane M Peres, and Clodoaldo AM Lima. "Feature selection for biometric recognition based on electrocardiogram signals". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, pp. 2911–2920.

[334] Ho J Kim and Joon S Lim. "Study on a biometric authentication model based on ECG using a fuzzy neural network". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 317. 1. IOP Publishing. 2018, p. 012030.

[335] Hugo Silva, Hugo Gamboa, and Ana Fred. "One lead ECG based personal identification with feature subspace ensembles". In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer. 2007, pp. 770–783.

[336] Janani C Sriram et al. "Activity-aware ECG-based patient authentication for remote health monitoring". In: *Proceedings of the 2009 international conference on Multimodal interfaces*. 2009, pp. 297–304.

[337] Jim Dearing. *IHS Markit Predictions for 2017 -Electronic Access Control*. 2017. URL: `https://technology.informa.com/588015/electronic-access-control-ihs-markit-pre%ADdictions-for-2017` (visited on 05/27/2020).

[338] Annamalai Natarajan, Kevin S Xu, and Brian Eriksson. "Detecting divisions of the autonomic nervous system using wearables". In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2016, pp. 5761–5764.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was pdfLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.