Interpretable predictions with Convolutional Neural Networks for complex data

Interpretable predictions with Convolutional Neural Networks for complex data

Interpreteerbare voorspellingen met Convolutionele Neurale Netwerken voor complexe gegevens

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor ann de
Universiteit Utrecht
op gezag van de
rector magnificus, prof. dr. H.R.B.M. Kummeling
ingevolge het besluit van het college voor promoties
in het openbaar te verdedigen op
woensdag 14 juni 2023
des middags te 2.15 uur

door

Giacomo Lancia

geboren 14 september 1993
te Velletri, Italië

Promotor :
Prof. dr. J. E. Frank

Copromotor :
Dr. C. Spitoni

Beoordelingscommissie:
Dr. A. Barra
Prof. dr. J. Beyersmann
Prof. dr. M.E.E. Kretzschmar
Prof. dr. ir. C.W. Oosterlee
Prof. dr. H. Putter

To Bienne

Sai ched'è la statistica? È na' cosa
che serve pe fà un conto in generale
de la gente che nasce, che sta male,
che more, che va in carcere e che spósa.

Ma pè me la statistica curiosa
è dove c'entra la percentuale,
pè via che, lì, la media è sempre eguale
puro co' la persona bisognosa.

Me spiego: da li conti che se fanno
seconno le statistiche d'adesso
risurta che te tocca un pollo all'anno:

e, se nun entra nelle spese tue,
t'entra ne la statistica lo stesso
perch'è c'è un antro che ne magna due.

Trilussa

Contents

List of Acronyms

Introduction

Quod fuit huius pretium cursus?

Seneca. *Medea*

Machine Learning (ML), Deep Learning (DL), and Artificial Intelligence (AI) represent nowadays the most used tool to explore, analyze, and cluster massive data sets (Janiesch et al., 2021; Sarker, 2021). The continuous demand for increasingly sophisticated and precise algorithms is often supported by the availability of massive datasets containing increasingly complex information (Najafabadi et al., 2015). Unlike ML techniques, DL techniques (e.g., the feed-forward Artificial Neural Networks (ANN)) are the most promising to capture and recognize salient insight into data (Zhang et al., 2017; Pratt, 2015; Buchanan, 2015). The superior skill of ANN in solving complex classification problems is a natural consequence of the fact that ANN mirror the ability of biological neural networks to gain much of their power in their deep structure (Hassabis et al., 2017; Aggarwal et al., 2018). Several ANN models have shown outstanding performance in the past decades; they do not consist of random connections between the units but are, instead, arranged in well-organized structures (Dean, 2022; Hornik et al., 1989; He et al., 2016). Among DL techniques, ANN are thus inspired by biological networks and, in particular, are designed to mimic the synaptic mechanism of neuronal cells.

The first attempt to mimic the behavior of a biological neural system with a mathematical model began in 1943; when the neurophysiologist Warren McCulloch and the mathematician Walter Pitts designed a simple electrical circuit named McCulloch-Pitts model(McCulloch and Pitts, 1943). Their model was characterized by a set of binary variables called *neurons* or *connectionisms*. Each of these variables can attain only two values, 0 or 1. When a neuron attains the value 1, that neuron is in its *active state* (i.e., this situation represents the post-synaptic activation of a neural cell). Conversely, when a neuron attains the value 0, that neuron is in its *rest state* (i.e., the neuron remains unactivated after the transmission of a nervous impulse); see figure 1. To simulate the behavior of the i-th neural cell at later times, Pitts proposed a simple linear rule based on the synaptic activity of all neural cells, namely

$$h_i(t) = \sum_{j=0}^{N} w_{ij} n_j, \tag{1}$$

with n_j denoting the j-th neuron, N the total number of neural cells considered, w_{ij} the *synaptic efficieny* (i.e. the coupling strength) between the i-th and j-th neuron; $h_i(t)$ represents the total *synaptic activity* of the i-th neuron at time t. By construction, $w_{ii} = 0 \ \forall i \in \{0, 1, \dots, N\}$, namely, one neuron cannot be synaptically connected with itself. In addition, the behavior of a neuron at time t is conditioned to its *synaptic activity* at the last time $t - 1$. The activation of a single neuron

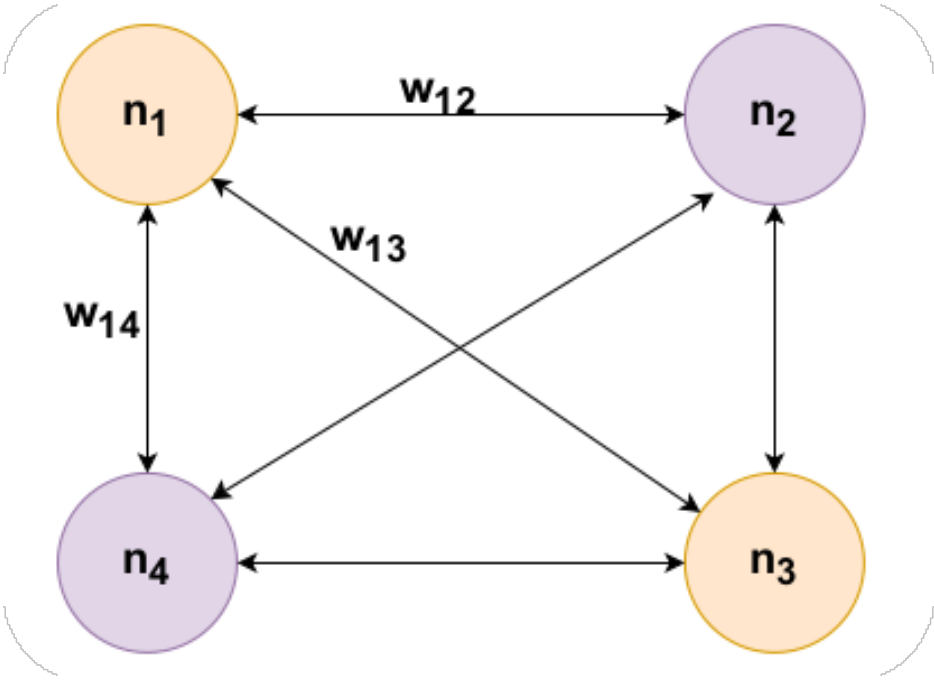


Figure 1: Scheme of the McCulloch-Pitts model.

is based on a *threshold process*, namely

$$n_i(t) = \theta(h_i(t-1) - \theta_0); \tag{2}$$

with θ_0 a threshold and $\theta(\cdot)$ the Heaviside function. However, the McCulloch-Pitts model does not provide any criteria for choosing the synaptic efficiency w_{ij}, i.e., the way the neurons can solve a specific classification task.

Some years later, in 1949, the work developed by McCulloch and Pitts was utilized by Donald Hebb to explain the *synaptic plasticity* of neuronal cells, that is, the adaption of neural cells during the learning phase (Hebb, 1949). Donald Hebb postulated that a pre-synaptic cell could improve its synaptic efficacy when affected by the persistent simulation arising from a post-synaptic cell. This behavior is usually called *Hebbian Learning*, and it is summarized with the sentence: *"cells that fire together, fire together"* (Hebb, 1949). The theory introduced by Hebb represents one of the first attempts to explain associative learning; it is often considered the basic rule of unsupervised learning of neural networks.

In 1961 Eduardo Caianiello incorporated the findings of Hebb into the McCulloch-Pitts model. Caianello used Hebbian learning to give the McCulloch-Pitts model a learning rule, the so-called *Mnemonic rule* (Caianiello, 1961).

Rosenblatt made a giant step forward in 1961. The *perceptron* (Rosenblatt, 1960, 1961) proposed by Rosenblatt represented the first example of ANN. Unlike the McCulloch-Pitts model, where the connections of the neural cells are modeled as a complete graph, Rosenblatt introduced a layer-like disposition of neuronal cells equipped with forward-oriented connections. This innovation made the learning phase of the perceptron simple and convergent. The perceptron model was trained through an iterative algorithm, and the synaptic efficiency could be adjusted to reproduce

the desired target output.

The perceptron represented the fundamental archetype of all layered-structured ANN (e.g., Multi-Layer Perceptron (MLP), Convolutional Neural Networks (CNN), and the Recurrent Neural Networks (RNN)) that will have been developed in the 80s and 90s. In these models, the flux of information is forwardly propagated from the input units toward the output units.

The development of the Backpropagation (BP) algorithm by Werbos (Werbos, 1974) represented another crucial turning point in improving the efficiency of the learning phase of ANN. The idea behind this algorithm is simple: the output values of a ANN model are iteratively backpropagated through all the hidden layers. The goal is the evaluation of the *error terms*, that is, some quantities directly related to the change of the loss function depending on the weights of the hidden layers.

In this thesis, we present the application of one type of ANN models (i.e., CNN), but in three different contexts:

1. **Binary Classification:** This work has been conducted within the framework *Diagnostic po-*

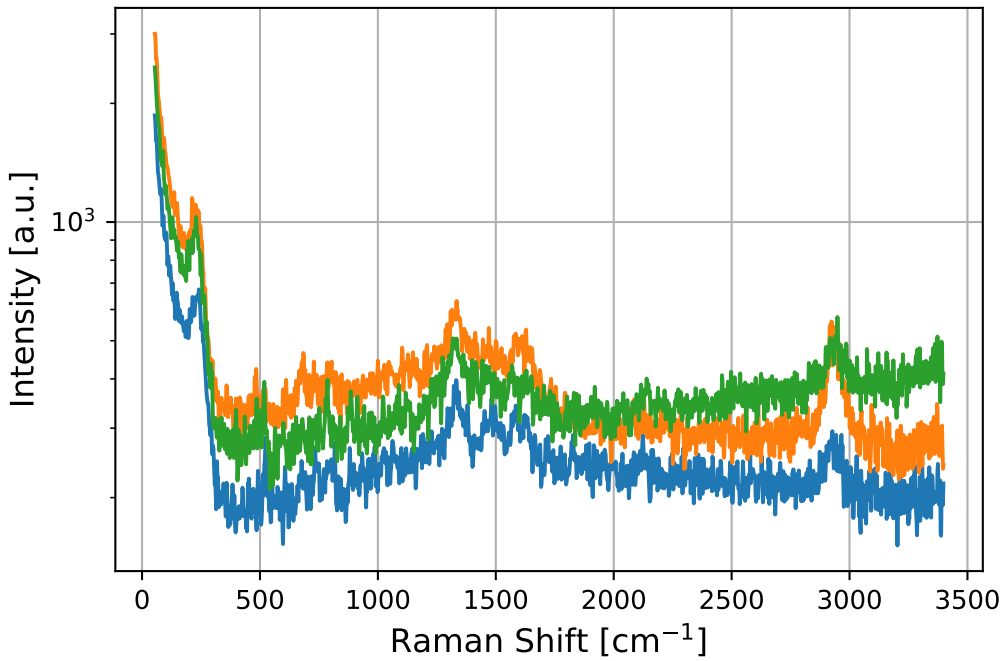


Figure 2: An example of three Raman Spectra.

tential of disorder: development of a Nanostructured innovative platform for rapid, label-free, and low-cost analysis of genomic DNA (DIANA). In this project, we attempt to improve the diagnosis of malignant neoplasms; we implemented several 1-D Convolutional Neural Networks (1-D CNN) models to classify the Raman spectra of genomic DNA of either malignant or healthy cells. Raman spectroscopy is a technique aiming to detect molecules' vibrational and rotational modes; this technique represents a fundamental approach to studying the elements composing any material, either solid or fluid.

Briefly, Raman spectroscopy is a scattering process involving photons: a monochromatic laser beam incises a sample, and the scattered photons deviate towards a mirror system that conveys them into a detector. Depending on the sample's response to the monochromatic incident light, one will record different energy values of the scattered photons. The values attained by this energy are usually represented in a *Raman spectrum*, namely, a plot illustrating the response energy of a sample as a function of the *Raman Shift* (i.e., the difference between the excitation and the Raman wavenumber); see figure 2.

In this work, we used a 1-D CNN model to study in-depth and improve the results reported by Durastanti et al. (2022). In particular, the data have been acquired using the same experimental setting of Durastanti et al. (2022). Thus, we observed that a 1-D CNN model could accurately discriminate different classes of tumor cells. Moreover, we compared DL and traditional ML (e.g., Logistic Regression) models to determine which one provides the most accurate prediction.

The data features that support the high predictive performance of 1-D CNN are investigated through *saliency maps*. Saliency maps represent the key to making explainable the activity of a broad class of ANN types (Simonyan et al., 2013; Montavon et al., 2019; Mundhenk et al., 2019); these maps denote those regions of input data where a ANN model captures the most predictive patterns and relevant data features. Hence, we exploited the potentiality of saliency maps to visualize how the Raman spectra of both healthy and cancer cells are pre-processed when propagated through the hidden layers of the 1-D CNN model. More specifically, we visualized the most relevant patterns of Raman spectra and tried to retrieve cancer cells' most distinctive physicochemical properties. Through this approach, we aimed to validate Raman Spectroscopy as a predictive and reliable tool for cancer diagnosis.

2. **Time-Series Classification:** We used a 1-D CNN model to make Time-Series Classification. We focused on classifying events such as *El Niño* and *La Niña* utilizing Time-Series data. These events represent a class of climate physics phenomena usually referred to as El Niño Southern Oscillation (ENSO). ENSO usually occurs on average once per four years in the southeast Pacific Ocean. Irregular variations in the water surface temperature, the water level rise, and the wind strength characterize ENSO. Usually, ENSO perturbs the climate of Peruvian and Chilean coasts; such a climatic upheaval is often propagated over many regions on Earth, e.g., the southwest Pacific Ocean. A representation of the ENSO events is shown in Fig 3; in these plots, different observations of the Oceanic Niño Index (ONI) are illustrated (the ONI is a metric to evaluate the variation (in Celsius degrees) of the Ocean surface regarding the average temperature value in the last 30 years). Although numerous studies revealed that ANN are excellent tools to forecast ENSO with significant lead times (Ham et al., 2019a; Yan et al., 2020), we decided to focus on the possibility of making interpretable what a 1-D CNN model can learn of the dynamics governing the ENSO events. This represents the novelty introduced by this work. However, the data involved in this process are not observations of physical observables in the southwest Pacific Ocean. Conversely, we used the Zebiak-Cane (ZC) model to simulate the dynamics occurring in a particular region of the southwest Pacific Ocean, i.e., *Niño 3.4*. Generated data offer the possibility of simulating different dynamics by configuring the parameters governing the velocity of the equatorial waves and *upwelling feedback*. We shall refer to a variation of these parameters from their reference value as *distorted physics*.

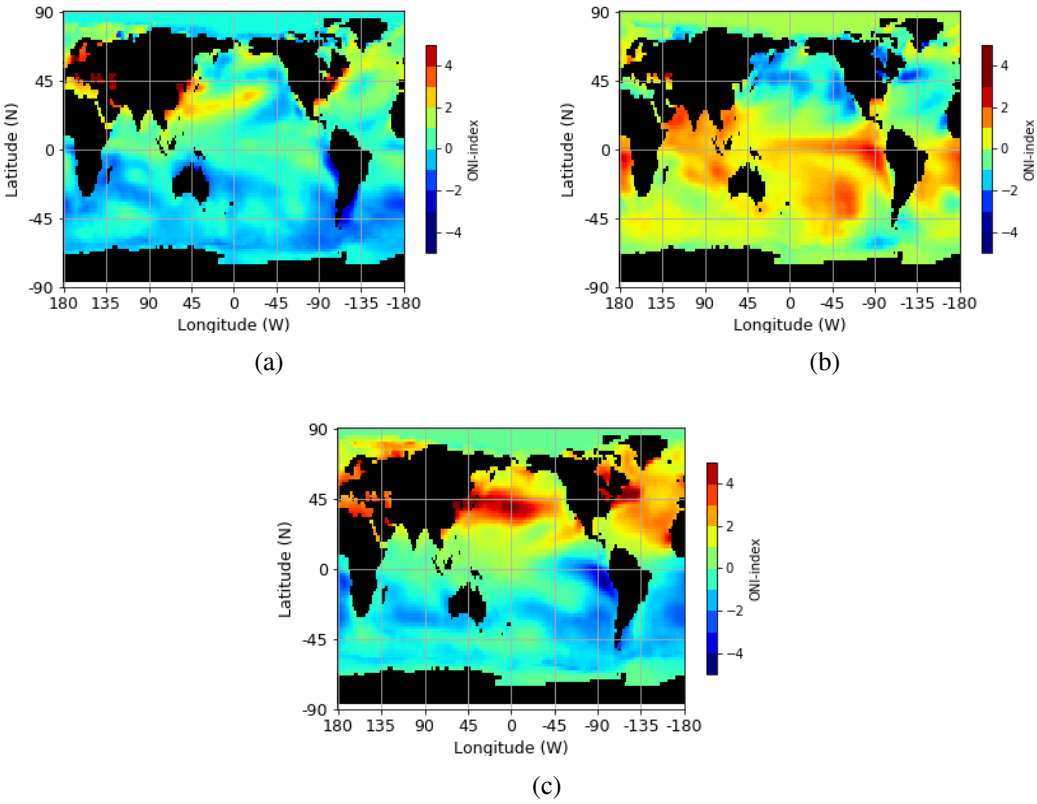


Figure 3: Representation of the ONI-index at different moments. (a) Example of *neutral ENSO*, ordinary measurement of the surface temperature of the southeast Pacific Ocean; (b) Example of *El Niño*, an extraordinary increase of the surface temperature of the southeast Pacific Ocean; (c) Example of *La Niña*, a notable decrease in the surface temperature of the southeast Pacific Ocean.

Once again, we exploited the potentiality of saliency maps-based representations to explain the activity recognition of the 1-D CNN model. Saliency maps represent the primary tool we used to understand which dynamic characteristics of the ZC model are captured by the 1-D CNN model. Thus, we completed our analysis by considering the *distorted physics data*. Through the frequency analysis of the most salient patterns, we isolated the contribution in the 1-D CNN-based predictions of both macroscopic (i.e., low-frequency signals) and microscopic components (i.e., high-frequency signals) of the input Time-Series. Hence, the construction of saliency maps is here meant as a tool to have a deep understanding of the physical processes associated with temporal patterns of one particular class of outputs.

In addition, we compared the activity 1-D CNN with another class of DL methods broadly used in ENSO forecasting, i.e. Gaussian Density Networks (GDN) (Petersik and Dijkstra, 2020). The framework we worked on is the same, i.e., we used the same ZC simulated data. However, these two approaches are conceptually different; 1-D CNN is designed as a classifier, whereas GDN is employed to solve regression problems. Hence, we focused on determining which of the two approaches (classification or regression) is the most promising;

we observed that a 1-D CNN-based classification could make the difference when dealing with simulated ZC data.

3. **Healthcare Records Classification:** We implemented here 1-D CNN and 2-D Convolutional Neural Networks (2-D CNN) models to improve the prediction of the onset of Intensive Care Unit Acquired Infections (ICUAI) occurring within the Intensive Care Unit (ICU) department. This project represents the most challenging application of CNN methods. Although ML techniques have already shown many promising results in the forecast of ICUAI (Chang et al., 2011; Roimi et al., 2020; Feretzakis et al., 2020; Anand et al., 2018), we considered the same problem from a broader point of view. We used the CNN models to analyze a vast data collection with a short sampling frequency (i.e., 1 minute), i.e., the Electronic Health Record (EHR) of ICU patients. We shall refer to the EHR as the high-frequency data. In addition, we considered an overall population size of over 5000 ICU-admitted patients. The detailed information in the EHR represents a source of high-detailed data often not included in formal studies. The novelty introduced by this work is represented by the attempt to exploit the recognition activity of the 1-D CNN model to improve the predictive performance of a Competing Risks (CR) model. In other words, we used the potentiality of 1-D CNN to individualize the patterns of 1-minute EHR (i.e., Hearth Rate, Arterial Blood Pressure, Pulse Pressure, Saturation level, and Respiratory Rate) that help improve the early detection of the onset of ICUAI. More specifically, the implementation of a 1-D CNN model aims to evaluate *risk score of infection* based on the observations recorded by the ICU monitors. Next, we included the 1-D CNN risk score into a broad collection of traditional predictors. We used the solid mathematical background of CR methods (e.g., the CR Cox model) to make dynamic predictions. Hereafter, we refer to this approach based on the combination of CNN and CR as the *two-step approach*. A schematic representation of the two-step approach is shown in figure 4. The research questions we addressed are the followings:

   - Can DL be regarded as a methodology to provide novel explanatory variables (e.g., a risk score) that could help anesthesiologists predict acquired infections in advance?
   - Can DL improve the traditional way of extracting information in the ICU setting?

   Similarly to the previous application, the saliency maps have also been used to visualize the most relevant patterns of the input vital signals that support the predictions provided by the 1-D CNN model. Hence, the study of the statistical property from the most relevant patterns represents the methodology we developed to improve the traditional way of including explanatory variables extracted from continuously monitored data in the ICU.

The content of this thesis can be summarized as follows: Chapter 1 is dedicated to reviewing some background material on ANN. Chapter 2 presents the results obtained within the framework of DIANA, i.e., the resolution and the interpretation of a 1-D CNN-based classification problem to distinguish tumor and sane cells. Chapter 3 focuses the attention on the interpretation of the 1-D CNN-based classification of synthetic ZC data. Chapter 4 is entirely dedicated to the attempt to dynamically predict the onset of ICUAI through the two-step approach. In the appendixes, the reader will find a deep insight into the mathematical details of the models employed. In particular, Appendix A and Appendix B contain the formulation of the backpropagation algorithm for both the 1-D CNN and the Long Short-Term Memory (LSTM) model, respectively. Appendix C is

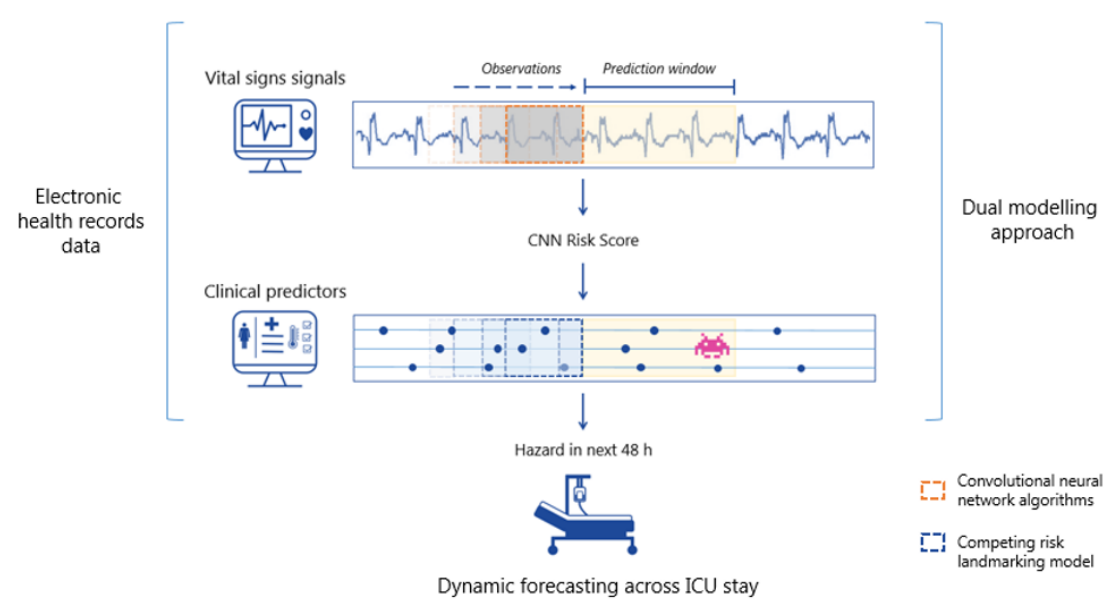


Figure 4: Scheme of the two-step approach modeling to detect ICUAI.

dedicated to the ZC model. In Appendix D, more details about Survival Analysis (SA) and CR models are provided.

Chapter 1

Feed-forward Artificial Neural Networks

> E la matematica, il linguaggio odierno, note grida scomposte; essa è il coro dei sopravissuti, il latino con cui l'uomo d'oggi celebra la liturgia dell'estinzione senza capirci granché: I numeri non si possono amare
>
> M. Sgalambro, *Corpi in movimento*

In this chapter, we shall review some basic knowledge about ANN with the scope of making the reader familiar with the terminology of ANN. Thus, we shall introduce here the main characteristics of the ANN models that the reader will encounter in this thesis, such as MLP (section 1.1.2), CNN (section 1.1.3), Mixture Density Networks (MDN) (section 1.1.5), and RNN (section 1.1.4). The reader will find in an application 1-D CNN in Chapter 2, 1-D CNN and MDN have been applied in Chapter 3, and both CNN and MLP and RNN have been used in Chapter 4. For completeness, a review of the techniques used to train a ANN model in supervised problems is presented in section 1.2. The techniques for Explainable Artificial Intelligence (XAI) will also be introduced; those algorithms have been used to interpret the activity of pattern recognition after that one ANN model was trained. In particular, we shall focus on two algorithms, i.e., the Vanilla Gradient (VG) algorithm and the Saliency Map Order Equivalent (SMOE) scale (section 1.3).

1.1 Types of Artificial Neural Networks

1.1.1 Perceptron

The perceptron (Rosenblatt, 1960) represents the first basic idea of feed-forward neural networks and the most simple and primitive one. Such a network's components are the input and output units; see Fig 1.1. The input units are directly connected with the output units. Let $\mathbf{X}$ be the input units (i.e., the data that one propagates through the perceptron; we often shall refer to it as input data or input features), $\mathbf{o}$ the output units (i.e., the output values returned by the perceptron and arranged

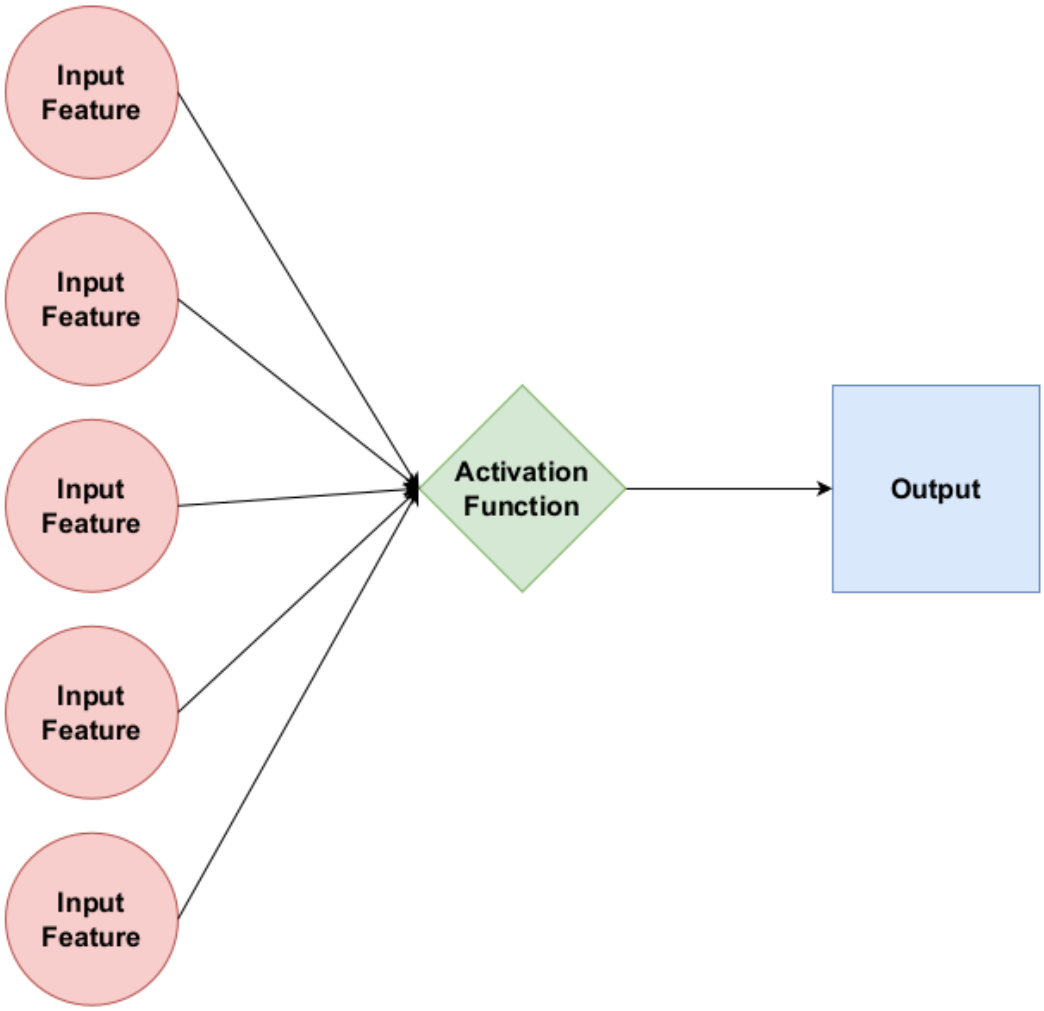


Figure 1.1: Schematic illustration of the Perceptron model.

like an array), $\varphi(\cdot)$ the activation function, and $\mathbf{w}$ the weight matrix related to the input units; the dependence between the outputs and inputs are given by

$$o_j = \varphi\left(\sum_{l=0}^{N} w_{jl} X_l\right); \tag{1.1}$$

where o_j is the j-th component of the output values. Usually, the scalar product in (1.1) involves the presence of a bias term; such a term is often embedded in w_{jl}. In other words, one assumes w_{j0} to be the actual bias term of the j-th output; consequently, $X_0 = 1$ by construction. The activation function represents the critical point of the perceptron. Its presence is necessary to introduce that non-linearity characteristic that represents the distinctive feature of this model. When dealing with classification problems, $\varphi(\cdot)$ takes the form of a step function with outputs -1 and 1.

As with the ML techniques, the training phase of a Perceptron has the scope of minimizing a loss function; the choice of such a functional depends on the nature of the problem to solve. For instance, the regression problems usually involve the minimization of the *mean squared error* between the outputs $\mathbf{o}$ (i.e., the values estimated by (1.1)) and the target values $\mathbf{t}$ (the true values one aims to estimate), namely

$$\text{MSE}(\mathbf{o}, \mathbf{t}) = \sum_{k=0}^{N} (\mathbf{o}^{(k)} - \mathbf{t}^{(k)})^2; \tag{1.2}$$

with N the number of data points or instances, $\mathbf{o}^{(k)}$ and $\mathbf{t}^{(k)}$ the output and the target values of the k-th instance, respectively. Binary classification problems, instead, are usually solved by minimizing the *binary cross-entropy*, namely

$$\text{BCE}(\mathbf{o}, \mathbf{t}) = -\frac{1}{N} \sum_{k=0}^{N} \mathbf{t}^{(k)} \log(\mathbf{o}^{(k)}) + (1 - \mathbf{t}^{(k)})(1 - \log(\mathbf{o}^{(k)})) \tag{1.3}$$

with $\mathbf{o}^{(k)}$ the output values (as estimated by (1.1)) and $\mathbf{t}^{(k)}$ as the true class of the k-th sample. In this case, $\mathbf{t}^{(k)}$ is usually a binary variable denoting whether the k-th instance belongs to either class 0 or 1.

With one suitable choice of both the activation function and the loss function, one can see that the perceptron model is equivalent to some popular machine learning techniques such as the *Support Vector Machine* and the *Logistic Regression*. Specifically, perceptron works as a Supported Vector Machine (SVM) model when the activation function is continuous and linear (i.e., $\varphi(\sum_{l=0}^{N} w_{jl} X_l) = \sum_{l=0}^{N} w_{jl} X_l$), and the loss function is the *hinge loss function*, namely

$$\max\left[0, 1 - \left(\sum_{j=0}^{J} \sum_{l=0}^{N} t_j w_{jl} X_l\right)\right];$$

note that, in this case, we denoted with J the number of output nodes of the perception and $\mathbf{t}$ the true class associated with the instance $\mathbf{X}$. Alternatively, the Logistic Regression (LR) can be represented by choosing the cross-entropy as the loss function and choosing a sigmoid activation function, namely

$$\varphi\left(\sum_{l=0}^{N} w_{jl} X_l\right) = \frac{1}{1 + \exp\left(\sum_{l=0}^{N} w_{jl} X_l\right)}.$$

Hence, DL can mimic many conventional ML techniques. This indicates how DL is much more versatile with respect to ML.

The perceptron model, however, might not lead to successfully solving a classification task if a large amount of sparse data is involved in the learning phase. In this case, the decision surface proposed by the perceptron model (i.e., that surface dividing at best the input data **X** by class label) might not learn all crucial details into input features. We recall that the decision surface of a generic binary classifier is defined as follows: let $\mathcal{X} = \{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$ a set of observations involved in the training phase of a binary classifier, where each observation has dimension n equal to the number of explanatory variables; let $f : \mathbb{R}^n \to [0, 1]$ be the target function of a binary classifier, i.e., the function returning the output values of the binary classifier (by construction, $f(x^{(i)})$ usually estimates $\mathbf{P}(Y = 0|X = x^{(i)})$, with Y denoting the target class). Thus, we define the decision surface $\mathcal{S}$ these with all points (instances) $\chi \in \mathbb{R}^n$ such that both the estimated probabilities $\mathbf{P}(Y = 1|X = \chi)$ and $\mathbf{P}(Y = 1|X = \chi)$ are equal; so in formulae

$$\mathcal{S} = \{\chi \in \mathbb{R}^n : f(\chi) = 1 - f(\chi)\}.$$

The decision surface, therefore, includes all boundary points that a binary classifier cannot label as either class 1 or class 0.

1.1.2 Multi-Layer Perceptron

The MLP is a class of ANN models that consist of a sequential arrangement of several perceptrons. More specifically, the connection between the inputs **X** and the outputs **o** pass through several in-

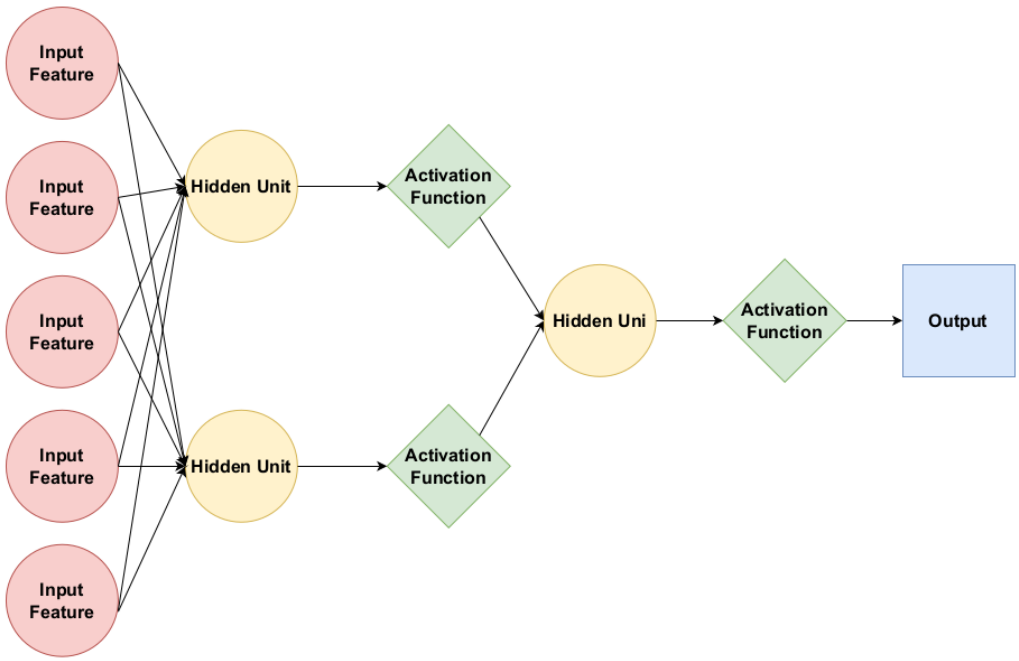


Figure 1.2: Schematic illustration of a Multi-Layer Perceptron model.

termediate operative units that are displaced on the computational layers, i.e., the so-called *hidden layers*; see figure 1.2. As with the perceptron, the MLP is a *feed-forward* neural network. The forward propagation rule is equivalent to a nested version of (1.1). In other words, for a Multi-Layer Perceptron with N hidden layers, we have

$$\begin{cases} \theta_j^{(1)} = \varphi^{(1)}\left(\sum_{l=0}^{M} w_{jl}^{(1)} X_l\right), \\ \theta_j^{(K)} = \varphi^{(2)}\left(\sum_{l=0}^{M} w_{jl}^{(2)} \theta_j^{(K-1)}\right),\ K = 2, 3, \dots, N \quad ; \\ o_j = \varphi^{(N)}\left(\sum_{l=0}^{M} w_{jl}^{N} \theta_j^{(N)}\right); \end{cases} \tag{1.4}$$

with $\theta_j^{(K)}$ the *activated-value* of the j-th unit on the K-th layer, $w^{(K)}$ the weights connecting the $(K-1)$-th and the K-th layer, and $\varphi^{(K)}(\cdot)$ the activation function on the K-th layer. As one can see, the connection between the input features and the output units takes the form of a composite function, and this represents the advantage of applying DL methods to large and sparse datasets. Indeed, the use of nonlinear functions is a fundamental aspect of making linear separable the input features; that is, the sequence of non-linear mappings leads the input features to be transformed so that they are linearly separable at most (Hornik et al., 1989; Aggarwal et al., 2018).

In support of this, the absence of non-linear mappings in MLP would degenerate such a model into a Perceptron. In fact, if one admits the activation functions $\varphi^{(1)}(\cdot), \varphi^{(2)}(\cdot), \dots, \varphi^{(N-1)}(\cdot)$ of (1.4) to be linear (i.e. $\varphi(X) = X$), then the forward propagation rule takes the form

$$\begin{cases} \theta_j^{(1)} = \sum_{l=0}^{M} w_{jl}^{(1)} X_l \\ \theta_j^{(K)} = \sum_{l=0}^{M} w_{jl}^{(K)} \theta_l^{(K-1)},\ K = 2, 3, \dots, N-1 \quad ; \\ o_j = \varphi^{(N)}\left(\sum_{l=0}^{M} w_{jl}^{N} \theta_j^{(N)}\right) \end{cases}$$

that is, the activated values of $\theta^{(N)}$ are exactly the result of the application of a linear kernel on the input features X_l, and the latter equation is equal to (1.1). Hence, the choice of a linear activation function leads the hidden layers of a MLP model to process the data by means of a linear transformation. In contrast, the construction of the decision function follows the same strategy as a Perceptron. Consequently, such a model is expected to work as a Perceptron.

Therefore, the repetition of non-linear transformations represents the essence of DL techniques. The composition of several hidden layers in Multi-Layer Perceptron can often reduce the number of units per layer. It's reasonable to think that the total number of units in a MLP is associated with the *power* of this type of ANN. In other words, this parameter represents the amount of insight the network could capture into the input features. However, the arrangement of several units represents another parameter to which MLP could be sensitive. Indeed, ANN with a large number of hidden layers but with a few units usually better generalizes the description of the decisional surface (Aggarwal et al., 2018). Deeper architectures with a few units aim to capture the repeated regularities in the patterns of the input data and generalize the learning in even those domains of the data where observations are missing or have sparse correlations.

1.1.3 Convolutional Neural Networks

CNN are the core of this thesis. This section will review some basic knowledge about this class of ANN. CNN represents, therefore, a specific class of ANN that is designed to work with grid-structured data, e.g., Time-Series and images. Due to this intrinsic characteristic of working with

multi-level data, CNN found a lot of applications in image recognition (Liu, 2018; Zheng et al., 2017; Lou and Shi, 2020; Kagaya et al., 2014), anomaly detection (Kwon et al., 2018; Naseer et al., 2018; Staar et al., 2019), and Time-Series forecasting (Borovykh et al., 2017; Selvin et al., 2017; Livieris et al., 2020; Guo-yan et al., 2019; Sezer et al., 2020). More specifically, convolutional and max-pooling operators are combined to encode the sequentiality of the patterns contained in the input data. As a result, the optimization of the weights of the convolutional filters of the convolutional layers aims to give the most linearized latent representation of the input Time-Series. An example of 2-D CNN is sketched out in figure 1.3.

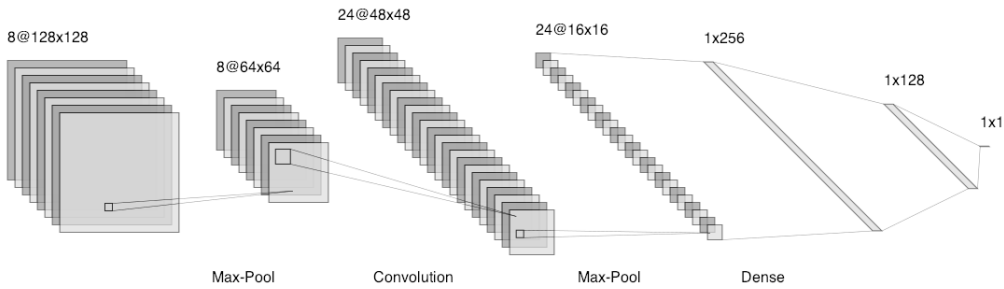


Figure 1.3: Schematic illustration of the 2-D Convolutional Neural Network model. Starting from the left, the input signal (2-D structured data) is composed of 8 features of size 128×128; then the Max-pooling operator reduces the size of the features; we now have 8 features of size 64×64. Next, convolutional layers generate 24 hidden features of size 48×48, which are newly resized by a Max-pooling layer; we now have a *feature map* of 24 features of size 16×16. Thus, the feature map is flattened and propagated through three dense layers with 256, 128, and 1 output units. The dense layer with one output unit provides the output of the 2-D CNN model.

The idea behind CNN is inspired by the experiments on the cats' visual cortex (Hubel and Wiesel, 1959). The visual cortex is the area of the brain where the stimuli from moving or static objects in the visual field are processed; this area stands out in image recognition. The mechanism that rules the activations of neurons in the visual cortex is similar to that of any other biological neural system, i.e., the cells of small neuronal areas of the visual cortex are sensitive to some specific electric impulse sent by some other particular regions of the visual field. In the presence of a stimulus (e.g., the detection of one specific object), some areas of the visual field can get excited; this information is then propagated toward the visual cortex, where some other cells get activated consequently. The activation of one area of the visual cortex depends on the shape and the orientation of the objects that one detects. For example, vertical edges cause some neuronal cells to be excited, whereas horizontal edges cause the excitation of some other neuronal cells. Hubel and Wiesel found that the cells of the visual cortex are connected by means of a layered structure. This result suggested that mammals' neural systems could be arranged in sequential layers to extract different levels of abstraction from external stimuli (Aggarwal et al., 2018). Such a conjecture has some analogies with pattern recognition in DL, where similar aspects are related to the *hierarchical feature extraction*.

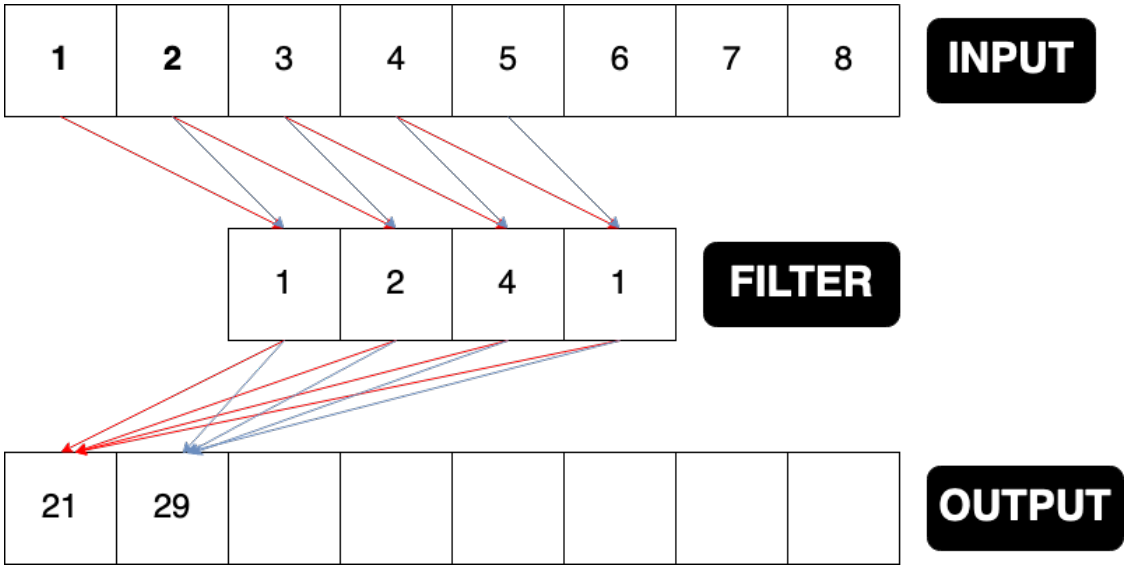


Figure 1.4: Example of the 1-D convolutional operator.

The core of the CNN is undoubtedly the convolutional layers. These computational layers are designed to process grid-structured data by means of convolution operators. Similarly to MLP, the CNN belongs to the family of the feed-forward networks, and the operation that takes place in the convolutional layers is still a sort of linear weighting, like in (1.4). In convolutional layers, however, the forward propagation does not take place by a simple weighting but rather by utilizing a discrete convolution operator. Thus, we consider 1-D-type data (e.g., Time-Series) and denote it with Z. Note that Z is a matrix with dimension $T \times N$ whose columns are Time-Series with amplitude T; we shall refer to them as *Time-Series features*. Likewise, we denote with w a three-order tensor of weights containing the *convolutional masks* (often called *convolutional filters*, or filters). The term convolutional mask is often used in image processing to denote a matrix or array containing weights that are applied to pre-process an image or a Time-Series with the general purpose of de-blurring, sharpening, embossing, edge-detecting, etc. The three dimensions of w describe the number of new features constructed by the convolutional layer, the size S of the convolutional masks, and the Time-Series features of the input data. Note that one chooses $S \leq T$. The output of a convolutional layer is denoted with ζ. It has the same matrix shape as Z, a matrix whose columns are the new Time-Series features constructed by the convolutional layer. The convolutional operations to apply on the input data Z to obtain ζ are defined as follows:

$$\zeta_{ij} = \sum_{k=0}^{N} \sum_{s=0}^{S} w_{j,s,k} Z_{i-s,k}. \tag{1.5}$$

The convolution operator is often utilized for processing visual images and Time-Series. For example, in Adaptive Optics, some deformable mirrors are embedded in large telescopes to provide the best correction of some noised wavefront; that is, those mirrors act like a convolution with the goal of de-noising the raw images acquired by a large telescope (Roddier, 1999; Riser and Cassarly, 2001). Likewise, the human eye could be considered an optic system that applies a sequence of convolution operations to extract and compress the relevant details in object recognition hierarchi-

cally (Tuli et al., 2021). Filter banks represent a field of *Digital Signal Processing* that make use of convolution operators to provide a multi-scale representation of 1-D sequential data (Rioul and Vetterli, 1991; Sainath et al., 2013; Cimpoi et al., 2014).

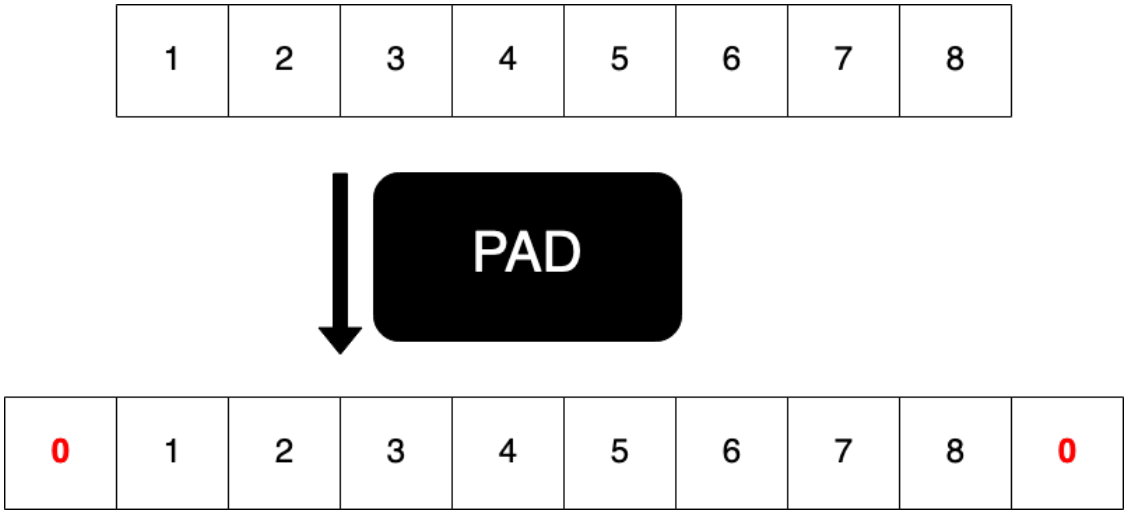


Figure 1.5: Example of the zero-valued padding.

However, when dealing with discrete convolution, one has always bear in mind that the finiteness of the data represents a limit for the computation of convolutions. Depending on the problem under consideration, one has to pay attention to the *padding* of the convolution, whether assuming a specific behavior of the data outside their domain or not. For example, when assuming the data has no behavior outside their domain, one speaks of *valid padding*. Instead, when padding the edge of the data with a zero-valued array, one speaks of *zero-valued padding*; see figure 1.5. The latter is implemented when one needs the convolutional layers to return some time-series features with the same length as the input features; otherwise, *valid padding* remains the most recommended. However, when using the zero-valued padding, one has always to bear in mind that the convolutions that occur in the proximity of the edges of time-series features will always involve some zero-valued patterns that might be non-informative in some cases. Another essential characteristic of the convolutional layers is the *stride*. In a standard approach, one would compute all the convolutions at all spatial locations. In some cases, however, it might not be necessary to require such a large amount of operations; one can reduce the granularity of convolutions by applying the convolution operators to a smaller ensemble of spatial locations. Thus, we say that a stride of σ_q is applied on the convolution of the K-th layer if these convolutions are computed at locations $1, \sigma_q + 1, 2\sigma_q + 1$ and so on. The standard approach of computing convolutions corresponds to setting $\sigma_q = 1$. A schematic example for $\sigma_q = 2$ is shown in figure 1.6. However, larger values of the stride parameter are not recommended (Aggarwal et al., 2018). The choice $\sigma_q = 1$ is recommended, even though in some cases $\sigma_q = 2$ can lead to some substantial improvements.

Pooling operations represent the other central aspect of CNN; they often follow convolution operations. Similarly to convolution operators, pooling operators are layer-like arranged. The pooling operations' purpose is to encode convolution layers' results effectively. A popular choice is represented by the *Maxpooling layer*; that is, the input representation of activated feature maps (i.e., an activation function applied to the results of convolutional layers) is downsampled by taking the

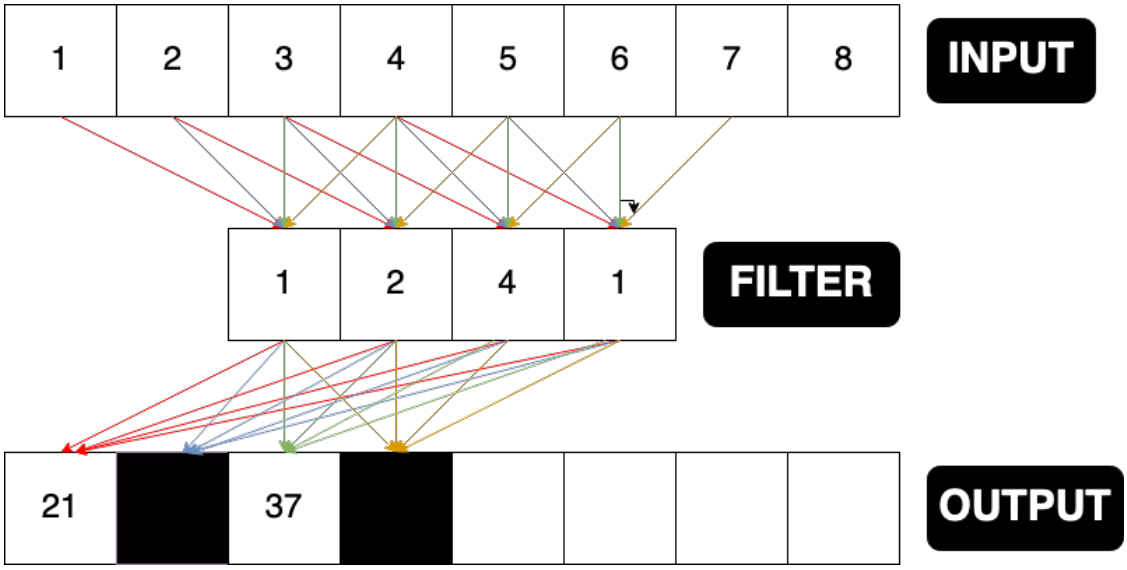


Figure 1.6: An example of stride in convolution operators for a stride of $\sigma_q = 2$.

maximum value over a spatial window; see figure 1.7. The size of each spatial window is often called *pooling size*. Usually, the recommended pooling size of 2 is often sufficient to train well-performing networks (Aggarwal et al., 2018). On the contrary, larger values would correspond to focusing the receptive field of the network on some macro-details of the input data. Consequently, the network could lose sight of relevant details at lower time scales.

Pooling layers, in particular *Max-pooling* layers, are fundamental to making a CNN *translational invariant*. Suppose that after training a 1-D CNN, we propagated one modified instance of the training set; e.g., we applied a slight translation or rotation to that instance. As a result, we can reasonably expect that the convolutions in the convolutional layers could not be affected by such a linear transformation acting in the features of our modified instance; likewise, the output of the max-pooling layer would not be affected. Thus, CNN would encode similar input features similarly; this represents their property of being *translational invariance*. CNN mirrors the main property of visual cortices in recognizing static objects when they are slightly moved from their usual position.

The encoding proposed by the convolutional and pooling layers must finally be processed to make the final predictions. For this scope, one can use the *Fully-Connceted Layers* (a.k.a., *Dense Layer*), which is the typical structure of a MLP. In other words, one Fully-Connected layer is equivalent to one hidden layer of MLP. The encoding provided by the convolutional and pooling layers is flattened to be propagated through one MLP whose outputs nodes represent the outputs of a CNN model.

1.1.4 Recurrent Neural Networks

RNN is a class of ANN devoted to predicting future observations of Time-Series. They found many applications in speech recognition and Time-Series forecasting (Sagheer and Kotb, 2019; Chimmula and Zhang, 2020; Siami-Namini et al., 2018; Irie et al., 2016; Li et al., 2015). These networks are based on estimating the probability of future observations given the immediate history

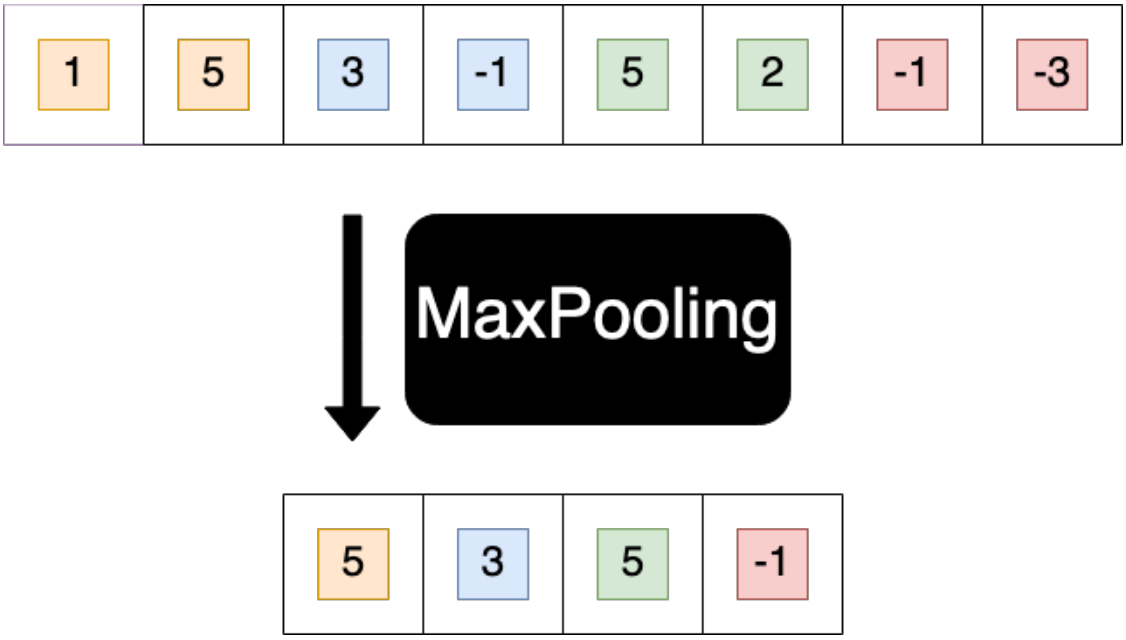


Figure 1.7: An example of Maxpooling operator with pooling size of 2.

of the previous ones. Similarly to the MLP, the RNN is composed of hidden layers often denoted as *hidden states*. The hidden states' main characteristic is that they are conceived as variables depending on time; their scope is to capture the aspects of the sequential evolution of several input data.

Let x_t be the observations at time t, and h_t the hidden state at time t, the basic equation ruling the update of the hidden state h_t at later times can be written as

$$h_t = \phi(Vx_t + Uh_{t-1}); \tag{1.6}$$

with h_{t-1} the hidden state at time $t-1$ (i.e., the latent representation of the observations at the last time $t-1$), $\phi(\cdot)$ an activation function, V and U some time-fixed matrix of weights. Note that we assume x_t and h_t to be column arrays; the products involved in (1.6) are standard matrix multiplications. Let y_t the output state at time t (usually the predicted probability that a specific observation will occur at time t). We can directly evaluate it from the latent description of x_t (i.e., h_t). Hence, we have

$$y_t = \theta(h_t); \tag{1.7}$$

with $\theta(\cdot)$ another activation function. RNN described by means of both (1.6) and (1.7) are usually denoted as Vanilla RNN; a schematic representation is shown in figure 1.8. *Vanilla RNN* suffers from the fact that vanishing and exploding gradients can dramatically alter the quality of the learning phase as well as the predictions; see Appendix B. During the employment of the BP algorithm (we shall discuss it in section 1.2), a sequence of multiplications involving the same weights can lead the gradient of the loss function to vanish or to grow unexpectedly (Hochreiter et al., 2001).

To address this instability, one needs to modify the recursive rules and avoid the recursive multiplications of weights like in (1.6) and (1.7). LSTM networks represent a solution to this problem because an additive form describes the updating of the hidden state. LSTM layers, however, require

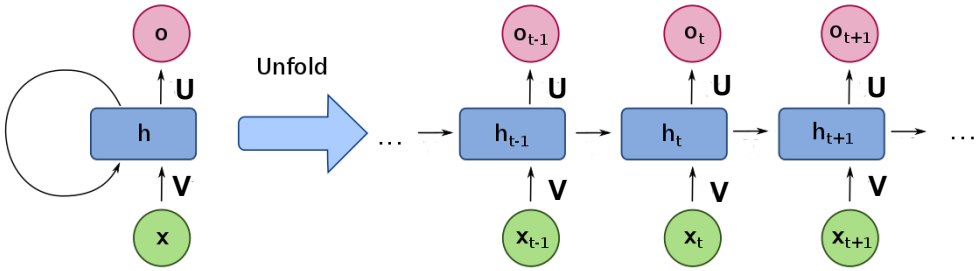


Figure 1.8: Scheme of s recurrent Vanilla RNN; compressed (left) and time-layered (right) representation. This image was taken from Wikipedia (2022b)

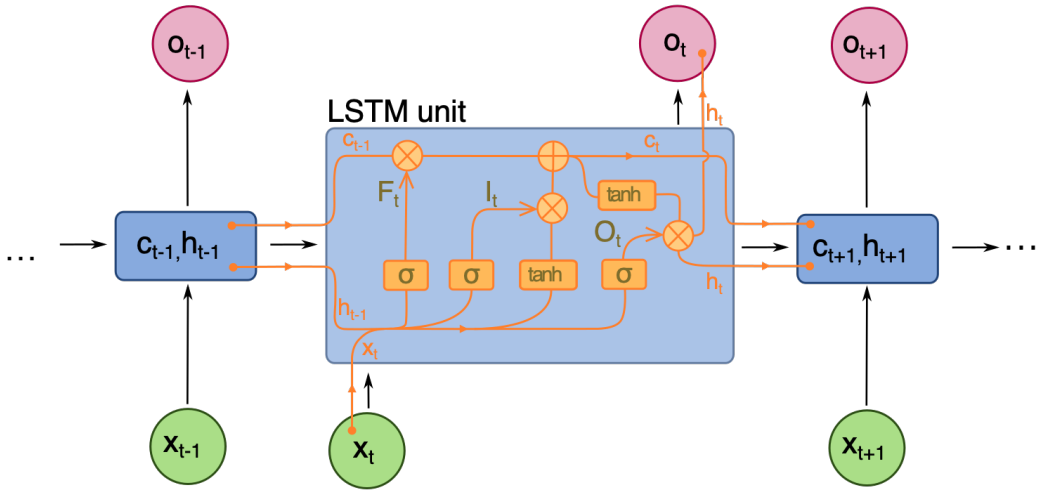


Figure 1.9: Scheme of an LSTM layer. This image was taken from Wikipedia (2022b)

the evaluation of a larger ensemble of parameters, usually referred to as *gates*. The first gate that one considers is the input state at time t (denoted as i_t), namely

$$I_t = \sigma(V_I x_t + U_I h_{t-1}); \tag{1.8}$$

next one considers the so-called *forget state* at time t, denoted as f_t, so one has

$$F_t = \sigma(V_F x_t + U_F h_{t-1}); \tag{1.9}$$

then the output state, namely

$$O_t = \sigma(V_O x_t + U_O h_{t-1}); \tag{1.10}$$

and the *new cell-state* at time t, denoted with S_t, is given by

$$S_t = \tau(V_S x_t + U_S h_{t-1}). \tag{1.11}$$

In (1.8)-(1.11) the matrices V and U contain, respectively, the weights of the input and recurrent connections, and the subscripts specify in which gate they are involved. Both $\sigma(\cdot)$ and $\tau(\cdot)$ are, respectively, the sigmoid and hyperbolic tangent functions. Furthermore, two additional rules are also employed to update the cell-state and the hidden states. So, one updates the cell-states via

$$c_t = F_t \odot c_{t-1} + I_t \odot S_t, \tag{1.12}$$

while the hidden states are updated via

$$h_t = O_t \odot \sigma(c_t). \tag{1.13}$$

The symbol $\odot$ denotes the Hadamard product; by definition, given two matrices A and B with the same dimension $m \times n$, the Hadamard product is defined as $(A \odot B)_{ij} = (A)_{ij}(B)_{ij}$. A representation of LSTM is shown in figure 1.9.

To show the advantages of using a LSTM, we can consider a straightforward case where the forward rules (1.8)-(1.13) involves one-dimensional quantities. From (1.12), one can see that $\frac{\partial c_t}{\partial c_{t-1}}$ is exactly the output of the forget gate, this represents a peculiarity of the forward rules of LSTM that avoid the evaluation of the gradient of being either exploring or vanishing; for more details see Appendix B. Because all updating rules involve element-wise operations, we can generalize such a result even to the multi-dimensional case.

1.1.5 Mixture Density Networks

MDN (Bishop, 1994) represent a class of ANN models designed to estimate the posterior probability $\mathbf{P}(\mathbf{t}|\mathbf{X})$; with $\mathbf{t}$ the targets and $\mathbf{X}$ the input features. Given a set of sample data $\mathbf{X}$ and observations $\mathbf{t}$, ML and DL techniques might often be designed to estimate the probability $\mathbf{P}(\mathbf{t}|\mathbf{X})$ to make inference for a new set of sample data $\bar{X}$. The approach that aims to model the posterior probabilities is usually termed *discriminative model*; the knowledge of the posterior probability enables one to determine a function to discriminate two or more classes of data. Thus, MDN represents a way to rearrange the known operational layers (e.g., dense, convolutional, and recurrent layers) in order to estimate the desired posterior probability. This class of models is applied to solve *inverse problems* (Bishop, 2009),; that is, those problems where one aims to estimate from a set of observations the causal variables that generated them.

To clarify, we can consider this toy example: suppose we have a set of uniform random variables $X \overset{i.i.d}{\sim} U(-0.1, 1)$, representing our set of data. The corresponding target values (i.e., the true observations) are denoted by t_n and are given as $t_n = 0.5X_n + 0.5\cos(2\pi X_n) + \varepsilon_n$, with $\varepsilon_n \overset{i.i.d}{\sim} U(-.1, 1)$; see figure 1.10. In our case, the inverse problem consists of obtaining the values of the data points X_n, given the targets t_n. When dealing with these types of problems, the least squares approach can lead to a poor solution. For example, if we try to solve our toy example as a regression problem (i.e., minimization of the mean squared error function), then we will see that a MLP will not find an adequately accurate solution; we indeed obtained an *explained variance score* equal to 0.08; see Figure 1.11a. Briefly, we recall that the explained variance score (here denoted with Υ) for regression problems is given by

$$\Upsilon = 1 - \frac{\mathbf{Var}(y - \hat{y})}{\mathbf{Var}(y)};$$

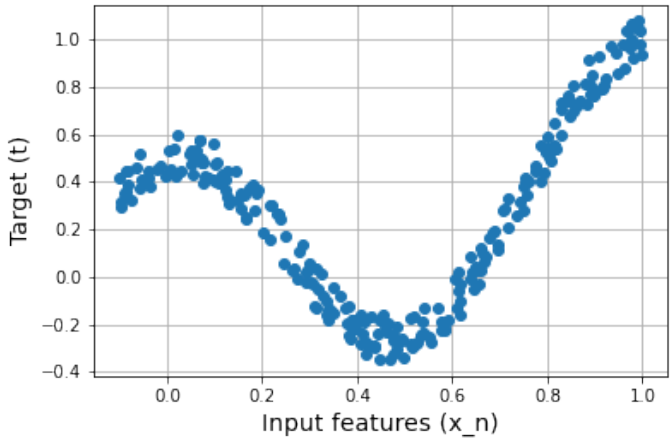


Figure 1.10: Illustarion of some target variables t_n generated via $t_n = x_n + 0.3\cos(2\pi x_n) + \varepsilon_n$.

with $\mathbf{Var}(\cdot)$ denoting the variance, y the true observations, and $\hat{y}$ the estimated observations.

The idea behind MDN is the assumption that $\mathbf{P}(\mathbf{t}|\mathbf{X})$ can be expressed by the mixture of probability distribution functions, namely

$$\mathbf{P}(\mathbf{t}|\mathbf{X}) = \sum_{k=0}^{K} \pi_k(\mathbf{X})\phi_k(\mathbf{t}|\Theta(\mathbf{X})); \tag{1.14}$$

with K the number of distributions considered, $\pi_k(\mathbf{X})$ some coefficients, $\phi_k(\mathbf{t}|\Theta(\mathbf{X}))$ a probability distribution function with parameters Θ. The mixing coefficients $\pi_k(\mathbf{X})$ meet the constraint

$$\sum_{k=0}^{K} \pi_k(\mathbf{X}) = 1 \quad 0 \leq \pi_k(\mathbf{X}) \leq 1; \tag{1.15}$$

while the parameters $\Theta(\mathbf{X})$ can be estimated by means of the maximum likelihood principle; this is equivalent to minimizing the following loss function

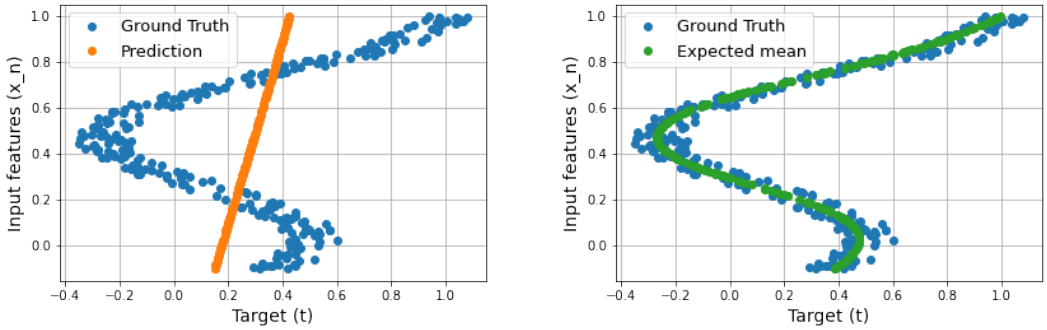
$$\tilde{\Lambda} = -\sum_{n=0}^{N} \log\left[\sum_{k} \pi_k(\mathbf{X_n})\phi_k(\mathbf{t_n}|\Theta(\mathbf{X_n}))\right]; \tag{1.16}$$

with N the number of sample data considered. The probability distribution function $\phi(\cdot)$ is often assumed normal; in this case, we speak of GDN. Accordingly, the maximization of the likelihood function is equivalent to minimizing the following loss function:

$$\tilde{\Lambda} = -\sum_{n} \log\left[\sum_{k} \pi_k(\mathbf{X_n}) \exp\left(\left[-\frac{\mathbf{X_n} - \mu(\mathbf{X_n})}{2\sigma(\mathbf{X_n})}\right]^2\right) \frac{1}{\sqrt{2\pi\sigma(\mathbf{X_n})}}\right]; \tag{1.17}$$

with $\pi(\mathbf{X_n})$, $\mu(\mathbf{X_n})$, and $\sigma(\mathbf{X_n})$ the parameters to estimate. The estimation of all parameters Θ enables us to generate new targets given the input values X. Alternatively, the search for the best Θ enables to estimate some crucial quantities as the *conditonal mean value* of the targets, namely

$$\mathbf{E}(\mathbf{t}|\mathbf{X}) = \int \mathbf{t}\mathbf{P}(\mathbf{t}|\mathbf{X})d\mathbf{t}; \tag{1.18}$$



(a) (b)

Figure 1.11: Solution to the inverse problem proposed for our toy example. We used (a) an MLP (5 layers with 128 units, hyperbolic tangent as activation function) approach and (b) the MDN (Mixture of 5 normal distributions, 5 layers of 8 units with hyperbolic tangent as activation function). For the MLP approach, the predictions (orange dots) are based on the resolution of a least-square problem. In contrast, for the MDN approach, the predictions (green dots) are the estimated expected mean values of the targets.

by plugging (1.14) into (1.19) one obtains

$$\mathbf{E}(\mathbf{t}|\mathbf{X}) = \sum_{k=0}^{K} \pi_k(\mathbf{X}) \mathbf{E}[\phi_k(\mathbf{t}|\Theta(\mathbf{X}))]. \tag{1.19}$$

Thus, the MDN does not represent a particular architecture of ANN (e.g. MLP, CNN, or LSTM) but is just an approach of using ANN to estimate the posterior probability of the targets from a set of observations. Theoretically, any architecture can be used to design MLP; according to the typology of the data involved, one might prefer to arrange CNN, LSTM, or MLP to estimate the best parameters Θ.

However, when choosing the layers and the activation functions, one has to pay attention to how to design the output nodes because these nodes represent the parameters ?. In particular, one should be sure that the proper constraints of these quantities are always satisfied. For example, in GDN we know that the mixture coefficients are subjected to (1.15); in this case, we can apply a *softmax* activation function on the activations $\mathbf{a}_k^{\pi}$ that determine the mixing coefficients, i.e.

$$\pi_k = \frac{\exp\left(a_k^{\pi}\right)}{\sum_l \exp\left(a_l^{\pi}\right)}.$$

The parameter $\mu(\mathbf{X_n})$ represents the mean value of the normal distribution and can therefore take any values in $[-\infty, \infty]$; a linear activation function is suitable for this purpose, and can be applied on the activations $\mathbf{a}_k^{\mu}$. Conversely, the parameter $\sigma(\mathbf{X_n})$ represents a standard deviation and must be a positive value. In this case, one can still use a linear activation function on the activations $\mathbf{a}_k^{\sigma}$; but the parameters estimated by the model are intended as $\log\left(\mathbf{a}_k^{\sigma}\right)$. Returning to our toy example, the fit of a GDN leads to a successful resolution of the inverse problem; see figure 1.11b (explained variance score 0.97).

1.2 Supervised training of Artificial Neural Networks

This section will review the main characteristics of supervised training of ANN. Supervised training of ANN is a fundamental topic of DL; it consists of using iterative learning algorithms to make a ANN model capable of solving a classification task. The main scope of the supervised training is, therefore, enabling the ANN to learn a function that maps input features (e.g., observations) to output values (e.g., true labels) based on example input-output pairs. Thus, we shall recall some basic knowledge on the learning phase of ANN and the BP algorithm. At last, we shall introduce the problem of overfitting and which methods can help contrast it.

1.2.1 The learning phase

When dealing with a simple architecture (e.g., the perceptron whose architecture has no hidden layers), the connection between the output node and the input data can usually be expressed in a closed form. As a result, minimizing the loss function does not encounter any limitations; thus, one can also find the best weights of a perceptron through a direct gradient computation (Bishop, 2009; Hastie et al., 2009). In more complicated cases (e.g., MLP and CNN), the evaluation of the gradient of the loss function cannot be directly computed as in the previous case because the dependence between the loss function and the weights can no longer be written in a closed form. To address this problem, we can focus, for one moment, on the two main steps of the learning phase of MLP, i.e., the *forward phase* and the *backward phase*; see figure 1.12.

Before introducing the forward and backward phases of any feed-forward ANN, we need to specify a few more basic concepts about the loss function. For example, in section 1.1.1, we introduced two standard loss functions, such as the mean squared error (see (1.2)) and the binary cross entropy (see (1.3)). The minimization of these loss functions is aimed at estimating the best parameters of an ANN; the resolution of such a minimization problem has the scope of establishing the best weight of each layer (e.g., the best convolutional filters of a CNN or the weights on each node of a MLP). More specifically, when choosing a loss function, one often opts for a metric to evaluate the distance between the estimated and true values for a given instance of data (denoted, respectively, with $\mathbf{o}$ and $\mathbf{t}$ in (1.2) and (1.3)). Intuitively, when implementing an ANN, minimizing the distance between estimated and true values has theoretically the scope of fitting the ANN model in question to output the target values $\mathbf{t}$ associated with the input data.

The forward phase represents the propagation of the input data through the hidden layers of an ANN. As in a cascading mechanism, each layer node elaborates the response from the upper layer, computes its response, and finally sends it to the layers underneath. Therefore, the activity of each node depends on two characteristics: the value that its weights take and the functional form of the activation function. The latter is a fixed characteristic and cannot be changed once the neural network's architecture has been decided. On the other hand, the weights represent the trainable set of variables that must be fitted to solve regression or classification problems.

The *backward phase* comes after the forward phase; the former is the phase where the loss function is minimized. For ANN, the backward phase is represented by the BP algorithm, as we shall explain in section 1.2.2. For both ML and DL techniques, the loss function is often minimized by means of methods based on a gradient descent algorithm. We shall denote a generic loss function with $\Lambda(\mathbf{w})$, where we made explicit the dependence of the loss function with respect to the weights $\mathbf{w}$ of the model. The idea behind the gradient descent algorithm is to find a minimum of $\Lambda(\mathbf{w})$ while

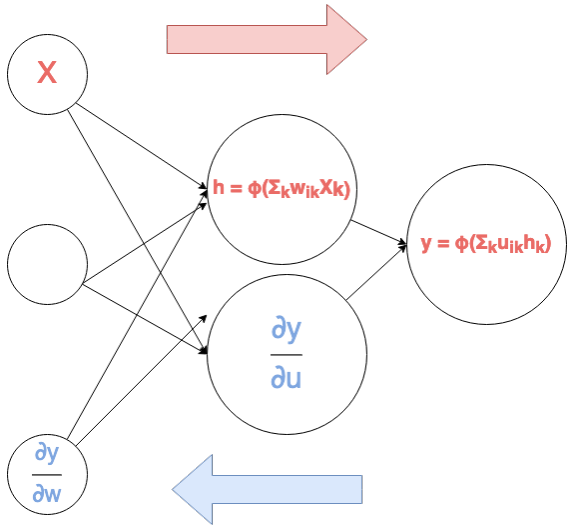


Figure 1.12: Scheme of the *forward* and *backward* phase of ANN.

continuously updating the weights $\mathbf{w}$ in the following way:

$$\mathbf{w}^n \longleftarrow \mathbf{w}^{n-1} - \gamma \nabla \Lambda(\mathbf{w}^{n-1}); \tag{1.20}$$

where $\mathbf{w}^n$ represents the weights at the n-th epoch, and γ the so-called *learning rate*, which is a magnification factor along the direction proposed by the gradient. Note that $\mathbf{w}^n$ indicate the weights that the model assumes during the forward phase of the n-th epoch; accordingly, the evaluation of $\Lambda(\mathbf{w}^n)$ represents the evaluation of the loss function during the forward phase of the n-th epoch. We recall that the term *epoch* is used in ML and DL to indicate one complete pass of the training dataset through a ML or DL algorithm. For convenience, large datasets are usually grouped into batches or mini-batches during the training phase of any ML or DL model. When a dataset is grouped into N mini-batches, then (1.20) takes form

$$\mathbf{w}^n \longleftarrow \mathbf{w}^{n-1} - \gamma \sum_{n=0}^{N} \nabla \Lambda_i(\mathbf{w}^{n-1}); \tag{1.21}$$

with Λ_i computed for the i-th mini-batch of data. Hence, the goal of both (1.20) and (1.21) is to search recursively the best set of weights $\mathbf{w}^*$ such that $\nabla \Lambda(\mathbf{w}^*) = 0$. However, correct evaluation of quantities like $\frac{\partial \Lambda}{\partial \mathbf{w}}$ remains a fundamental step to apply rules like (1.20) and (1.21). For feed-forward ANN, the gradient of the loss function is evaluated through the BP algorithm, as we shall discuss in the next section.

1.2.2 Backpropagation Algorithm

The *backpropagation algorithm* (BP) algorithm (Rumelhart et al., 1986) enables one to evaluate the desired gradient by exploiting the dependence between subsequent hidden layers. Such an algorithm is therefore based on the mechanism that rules the activity of each node. In this section, we shall discuss the BP algorithm for MLP.

Let's consider a generic MLP with n hidden layers, each one of those with I units. We denote with $a_j^{(k)}$ the output of the j-th node on the k-th layer, and its activated form with $\eta_j^{(k)}$ (i.e., the evaluation of the activation function whose argument is nothing than $a_j^{(k)}$). As we introduced in section 1.1.2, we can express the activity of one generic node by means of the following formulae

$$a_j^{(k)} = \sum_{i=0}^{I} w_{ji}^{(k)} \eta_i^{(k-1)}, \tag{1.22}$$

$$\eta_j^{(k)} = \phi(a_j^{(k)}); \tag{1.23}$$

where $\phi(\cdot)$ is the activation function, and $w_{ji}^{(k)}$ is the weight connecting the j-th node on the k-th layer and the i-th node on the $(k-1)$-th layer. Both (1.22) and (1.23) represent how the input data information is propagated through the hidden layers during the forward phase.

We focus for one moment on the last layer of our network, i.e., the n-th layer. The network's output is exactly the array $\eta^{(k-1)}$ by construction. We compute now the gradient of the loss function with respect to the weight $w_{ji}^{(n)}$ by using the chain rule, that is

$$\frac{\partial \Lambda}{\partial w_{ji}^{(n)}} = \sum_{q=0}^{I} \frac{\partial \Lambda}{\partial \eta_q^{(n)}} \frac{\partial \eta_q^{(n)}}{\partial a_q^{(n)}} \frac{\partial a_q^{(n)}}{\partial w_{ji}^{(n)}}. \tag{1.24}$$

By means of (1.22), we can see that the term

$$\frac{\partial a_q^{(n)}}{\partial w_{ji}^{(n)}} = \eta_i^{(n-1)} \xi_{qj}; \tag{1.25}$$

with

$$\xi_{qj} = \begin{cases} 1 & \text{if } q = j \\ 0 & \text{if } q \neq j \end{cases}$$

While the product $\frac{\partial \Lambda}{\partial \eta_q^{(n)}} \frac{\partial \eta_q^{(n)}}{\partial a_q^{(n)}} = \frac{\partial \Lambda}{\partial a_q^{(n)}}$, i.e., the so-called *error term*. Note that if Λ were the *mean squared error function* (as it often occurs with regression problems; see (1.2)), then the quantity $\frac{\partial \Lambda}{\partial a_i^{(n)}}$ would be proportional to the difference between the true observations and the i-th output of the MLP; this justifies the reason of its name. For clarity, we shall denote with δ_i^n the error term with respect to the output value due to the i-th node on the n-th layer, in formulae

$$\delta_i^k = \frac{\partial \Lambda}{\partial a_i^{(k)}}. \tag{1.26}$$

Hence we can write (1.24) as

$$\frac{\partial \Lambda}{\partial w_{ji}^{(n)}} = \delta_j^n \eta_i^{(n-1)}. \tag{1.27}$$

The same approach can be adopted but slightly modified when dealing with a generic hidden layer. Indeed, for the generic k-th hidden layer (with $k < n$), we have

$$\frac{\partial \Lambda}{\partial w_{ji}^{(k)}} = \frac{\partial \Lambda}{\partial \eta_j^{(k)}} \phi'(a_i^{(k)}) \eta_i^{(k-1)}; \tag{1.28}$$

where we are still exploiting the dependence relations (1.22) and (1.23) for two subsequent hidden layers. When considering the term $\frac{\partial \Lambda}{\partial \eta_j^{(k)}}$, we can apply once again the chain rule, i.e. we can exploit the relation established between the k-th and the $(k-1)$-th. In other words, the derivative $\frac{\partial \Lambda}{\partial \eta_i^{(k)}}$ is meant as the total derivative with respect to all the outputs $\mathbf{a}^{(k-1)}$ that are sent to the i-th node of the k-th layer. Thus, we have

$$\frac{\partial \Lambda}{\partial \eta_j^{(k)}} = \sum_{l=0}^{I} \frac{\partial \Lambda}{\partial \eta_l^{(k+1)}} \frac{\partial \eta_l^{(k+1)}}{\partial \eta_j^{(k)}} = \sum_{l=0}^{I} \frac{\partial \Lambda}{\partial \eta_l^{(k+1)}} \frac{\partial \eta_l^{(k+1)}}{\partial a_l^{(k+1)}} w_{lj}^{(k+1)}.$$

From the latter, we obtain

$$\frac{\partial \Lambda}{\partial w_{ji}^{(k)}} = \phi'(a_j^{(k)}) \eta_i^{(k-1)} \sum_{l=0}^{I} \delta_l^{(k+1)} w_{lj}^{(k+1)}; \tag{1.29}$$

note that

$$\delta_j^k = \phi'(a_j^{(k)}) \sum_{l=0}^{I} \delta_l^{(k+1)} w_{lj}^{(k+1)}. \tag{1.30}$$

As shown above, the combination of all equations represents the core of the BP algorithm. We can summarize the main step of the training phase (and BP algorithm) as follows:

1. We propagate the input data $\mathbf{X}$ through the hidden layers of an ANN. The evaluation of the output node is given by means of the sequential application of rules (1.22) and (1.23).

2. We consider the output units and evaluate its error term utilizing (1.26). Thus, we evaluate the desired derivative via (1.27).

3. We backpropagate the error terms towards the upper layers; that is, we recursively use (1.30).

4. We finally evaluate the desired derivative via (1.29).

Thus, the BP algorithm is the crucial point of any ANN: the way to go back and forward along the connections of the neural networks. Equations of type (1.22) and (1.23) enable one to go forward, i.e., to evaluate the output node from the input data. Conversely, the equation of type (1.29) and (1.30) exploit the connection between neighbor layers to evaluate the gradient of the loss function backwardly.

We have presented the derivation of generic feed-forward neural networks, such as MLP. Until the outputs of the nodes in the hidden layers are linear in the weights, then both (1.22) and (1.23) are still valid. Accordingly, the BP algorithm can still be implemented to train the neural network, e.g., CNN or RNN; for example, the reader will find a version of the BP algorithm for 1-D CNN in Appendix A. The BP algorithm, in addition, offers a way to understand the dependence between the inputs and the outputs. Theoretically, the idea behind this algorithm allows the evaluation of the change of the output nodes with respect to the input data. In other words, it allows the visualization of the most relevant input features, i.e., those whose small changes can lead to an abrupt change in the output values.

1.2.3 Regularization Techniques

During the learning phase of a neural network, one often may require the model to be as general as possible. The term *generality* refers to the ability of one model not to be detailed and specific for one reference dataset, but for all datasets that could be generated by the same dynamical rules that produced the reference ones (Hastie et al., 2009; Bishop, 2009; Aggarwal et al., 2018). One can often encounter the loss of generality in ML, and it is usually referred to as *overfitting*.

Regularization is a practical method to contrast overfitting, and it usually consists of adding a penalty to the loss function. Such a term is designed to be a function of the weights only. For example, if we denote with $\Lambda(\mathbf{w})$ the loss function as a function of the weights of the hidden layers, then the regularization term could consist of a term dependent on the weights (and it often occurs to be equal to the norm of the weights), that is we have

$$\Lambda(\mathbf{w}) \rightarrow \Lambda(\mathbf{w}) + \lambda|\mathbf{w}|^p,$$

with λ the shrinkage parameter. Depending on the value of p, one has a different type of regularization, e.g. $p = 1$ is the so-called *LASSO*, while the case $p = 2$ is usually referred to as *Ridge*.

Despite this approach working with a lot of ML techniques, it is inefficient with ANN. Indeed, Bishop and Nasrabadi (2006) proved that two MLP models with the same parameter shrinkage are not equivalent when trained with the same input features that could differ, at least in a linear transformation. They referred to it as a violation of the consistency property. As a result, such a couple of networks would not train the same way and would not find the same decision function. This problem can be fixed if one looks for a regularization term that remains invariant under the linear transformations on the input features. Consequently, one requires the regularization to be invariant with respect to any rescaling or shifts of both weights and biases terms, namely

$$\Lambda(\mathbf{w}) \rightarrow \Lambda(\mathbf{w}) + \sum_{l=0}^{L} \lambda_l |\mathbf{w}_l|^p;$$

with l denoting the level of one layer. In other words, one applies a regularization term to each hidden layer; this approach should leave unchanged the values of the weights norm when using a rescaling factor $\alpha \in \mathbf{R}^+$ on the weights (since such a factor is being absorbed by the shrinkage factor λ_l). When considering the special case, $p = 2$, this regularization per layer is equivalent to assuming that the weights are distributed according to the following prior distribution, namely,

$$\mathbf{P}(\mathbf{w}|\lambda) \propto \exp\left(-\sum_l \frac{\lambda_l}{2}||\mathbf{w}_l||^2\right).$$

It is important to stress that such a prior form represents an *improper prior* (i.e., a form of prior distribution that cannot be normalized). The bias parameters are unconstrained, and the corresponding *evidence* term turns out to be vanishing (Bishop, 2009).

Ensemble methods represent one of the most used alternatives to give generality to neural network models. *Dropout* method (Srivastava et al., 2014) is the most emblematic as the easiest to implement. The idea behind this method is to train and update a ANN by using a different and lower-size subset of weights at each learning step. This is equivalent to affirming that updating the overall neural network model is based on updating different (sub-)neural networks (because one chooses a different configuration of weights at each learning step). Such an approach should give

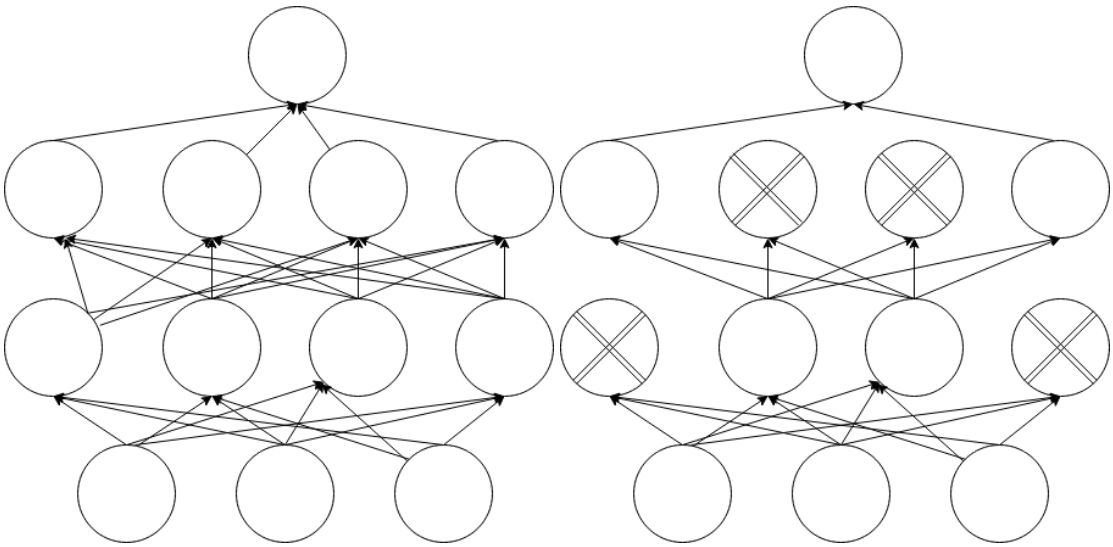


Figure 1.13: Scheme of Dropout method. (a) one standard ANN. (b) sub-sampling via Dropout.

the ANN model the desired level of generality (Aggarwal et al., 2018). The steps of the Dropout algorithm can be schematically described as follows

1. One samples a (sub-)neural network from a reference network. The fraction δ_d of units (or filters in CNN) per each hidden layer is sampled. The leftovers are temporarily unenabled, and all their connections are removed; see figure 1.13. The sampling of units is assumed to follow a uniform distribution.

2. One trains this sampled configuration by propagating a mini-batch of training instances.

3. The BP algorithm provides the updating of the weights involved in the propagation of step 2.

4. Before propagating a new mini-batch of training instances, one returns to steps 1 and 2.

It is a usual choice to set *dropout rates* δ_d in the range [0.2, 0.5]; larger values might lead to a stiff learning phase with the possibility for the ANN to be unable to learn essential details. The Dropout algorithm is usually implemented in a layered fashion to be embeddable with the most popular neural network architectures.

The application of a Dropout layer after each hidden layer turns out to be equivalent to injecting noise into the input data indirectly. Theoretically, one can always perturb the input data with a small amount of noise and then let the neural network model learn the optimal weights on the perturbed data. Alternatively, one always injects some noise into the outputs of one hidden layer. Likewise, the random dropping of connections proposed by the Dropout method represents the counterpart of injecting noise to the outputs of the hidden layers. Indeed, dropping a node could be regarded as the injection of a quantity of noise that makes that node unactivated. This can be made possible if one applies, for example, some multiplicative noise normally distributed with unitary mean and standard deviation $\tilde{\sigma}$. Such an approach is often referred to as *Gaussian Dropout* (Srivastava et al., 2014; Shen et al., 2017). The values attained by $\tilde{\sigma}$ aim to mirror the meaning of the dropout rate in

the Dropout layer. A broad common relation to connecting these two parameters is the following (Srivastava et al., 2014),

$$\widetilde{\sigma} = \sqrt{\frac{\delta_d}{1-\delta_d}}.$$

As one can see, no dropout rate corresponds to injecting no noise, whereas $\delta_d = 1$ is precisely represented by the injection of noise with infinite variance. In the latter case, the neural network model is brought to learn just from meaningless noisy signals.

The *early stopping method* represents another method to contrast overfitting. It consists of searching the iteration where the gradient-descendent optimizer proposes the optimal solution for both the training and the validation dataset. When applied during the learning phase of ANN, the *early stopping method* is described by the following steps

- Firstly, the dataset is divided into a training set and a test set; next, one puts apart a portion of the training set, e.g., 25%. We shall refer to the latter as the *validation set*.

- The ANN method is trained via BP. At this stage, one only involves the training set.

- After one epoch is completed, one computes the loss function for both the training set and the validation set. We shall refer to this quantity as the *error model* and denote it with Δ_{model}. During the first epochs, the Δ_{model} usually descends because the ANN model captures the patterns that are common to both the training and the validation datasets.

- Although the learning phase keeps learning some deeper details from the training set, Δ_{model} might begin to rise at some point. That point represents the moment when the model is going to get overfitted. Therefore, stopping the learning phase at that point represents a solution to confer the desired generality to the ANN model.

- Anyway, it is not always possible to determine when one should stop the iterations. to find the exact optimal solution. One possible criterium consists of setting a *patience* π_{err} and stopping the learning phase if π_{err} iterations one still have $\Delta_{model} > \delta_{model}$; with δ_{model} a threshold that is usually set to 0.

Practically, the algorithm can be summarized as follow

1. One monitors the values of Δ_{model} at each iteration. If $\Delta_{model} \leq \delta_{model}$, then the learning phase can continue uninterrupted.

2. As $\Delta_{model} > \delta_{model}$, one monitors the model error at the next π_{err} iterations.

3. If during all π_{err} iterations, the condition $\Delta_{model} > \delta_{model}$ is maintained, then the learning phase is stopped; otherwise, the learning phase is not interrupted and keeps being monitored by means of both steps 1 and 2.

One advantage of using early stopping is that it can be implemented quickly without significantly altering the architecture of the neural network models. Likewise, it can be regarded as a direct control of the learning phase.

1.3 Methods for Explainable Artificial Intelligence

Explainable Artificial Intelligence (XAI) represents a class of methods designed to understand the decisions and the predictions formulated by AI and DL techniques (Phillips et al., 2020; Vilone and Longo, 2021; Castelvecchi, 2016). The scope of XAI is to contrast the broad "black box" opinion that many users have when employing DL techniques, especially when one cannot explain how a DL technique arrived to formulate some specific decisions. We here present two algorithms that have been primarily used in all works presented in this thesis, i.e., the *Vanilla Gradient* (Simonyan et al., 2013) and the *SMOE scale* (Mundhenk et al., 2019). The reader will find an application of the VG in Chapter 2, while the SMOE scale in both Chapter 3 and 4

1.3.1 Vanilla Gradient

Gradient-Based algorithms represent a class of methods to determine the sensitivity of the output (or even an activated feature map) with respect to a small change in the input features in ANN models. Therefore, the main idea of these methods is to use the BP algorithm to compute the gradient of the output value with respect to the magnitude of the input data. Thus, the BP algorithm turns out to help reveal the relevancy of the input features.

Let's consider an ANN with input $\tilde{I}$ and output o. The *sensitivity* of the ANN can be defined as the quantity $\nabla_{\tilde{I}} o$, i.e. how changes the output o with respect to a change in the input $\tilde{I}$. This enables one to get a precise correspondence between the output and the input domain. We can leverage such a correspondence to construct a *saliency map* (Simonyan et al., 2013), i.e., a map denoting the sensitivity or the relevancy of the input features in being predictive according to the ANN model. We know that the detailed evaluation of $\nabla_{\tilde{I}} o$ is based on the implicit relation given by the hidden layers of the neural network model, i.e., the same kind of operation executed by the BP algorithm.

VG method is an algorithm based on the work of Simonyan et al. (2013); it represents one of the simplest gradient-based algorithms. Due to its intuitiveness and practicality, this method can be implemented for every kind of ANN where the up-dating of hidden weights takes place through the BP algorithm. This method is simple and is designed to be independent of the architecture of the neural networks. Let's consider an input data J belonging to class c and the class score function $S_c(J)$, i.e., the c-th element of the output o that usually represents the probability the instance J truly belongs to class c. Due to the extreme complexity and non-linearity of the hidden layers, the quantity $S_c(\tilde{I})$ cannot take a linear form; and accordingly, the decision functions cannot be expressed through $c-1$ hyper-planes taking the form

$$S_c(J) = \eta_0 + \langle w_c; J \rangle \,;$$

with $\langle ; \rangle$ denoting the scalar product, and w_c some weights dedicated to estimating the class score function of the c-th class. However, we can always make a linearization via Taylor-expansion around some reference input data J_0, namely

$$S_c(J) \simeq S_c(J_0) + \langle \nabla_w S_c(J); J - J_0 \rangle \,, \tag{1.31}$$

next, we can match the quantities

$$\begin{cases} \eta_0 = S_c(J_0) - \langle \nabla_w S_c(J); J_0 \rangle \,, \\ w_c = \nabla_w S_c(J_0); \end{cases}$$

with $\nabla_w S_c(J)$ the gradient of the c-th class score function at the input J with respect to all the weights $\mathbf{w}$ of the connections intercurring between J and the output node evaluating $S_c(J)$. Hence, the evaluation of the class score-derivative in (1.31) represents the change needed at the pixel J to affect the class score the most (Simonyan et al., 2013).

After training a neural network, we can therefore construct a saliency map. Firstly, we compute the class score derivative. Next, we take the maximum absolute value across all channels (Time-Series features in 1-D case); the maximum absolute value can be immediately connected to the highest sensitivity of the output o with respect to the input feature J. However, the visualization of saliency maps can suffer from scale problems, i.e., we need to find an interpretable scale of values to highlight the domain with the highest saliency values. To solve this inconvenience, we can apply any strictly monotone function as the cumulative distribution function of any distribution law. In doing so, we do not alter the explanatory feedback returned by the gradients' evaluation; therefore, map the saliency values into the domain $[0, 1]$. At this stage, we do not possess any constraint that can help us to determine the best cumulative distribution function to apply. Anyway, we can construct a saliency map that is specific to a fixed population of instances. That is, we can feed a ANN with some instances (e.g., a validation set), then backpropagate via the VG method, and finally apply the empirical distribution function obtained for all class score-derivative values. Technically, this approach is executed by estimating all percentiles via the *closest observation* approach (Hyndman and Fan, 1996). The cumulative distribution function is finally estimated via linear spline, ensuring the function is monotonically increasing and evaluable at any point. This strategy can be meant as a non-parametric attempt to describe the sensitivity of input features which is limited to the validation of the ANN model. In general, we opt for a simple criterion to detect saliency regions according to the values attained by the saliency maps; that is, when the saliency map attains values larger than 0.6, we mark the underlying region as salient (or relevant); if the saliency map attains values lower than 0.4 than we consider the corresponding region of the input domain as non-relevant. The neutral case is represented by the interval of values 0.4-0.6; when a saliency map attains those values, then we can consider the corresponding input domain as non-informative.

1.3.2 Saliency Map Order Equivalent scale

SMOE scale (Mundhenk et al., 2019) represents a valid alternative method for computing saliency maps. Unlike popular methods, such as VG(Simonyan et al., 2013) and *CamGrad*(Selvaraju et al., 2017), this method offers a different perspective of conceiving the estimation of saliency maps from CNN-activated feature maps. Indeed, this method does not involve the computation of the gradient of the activated features maps to understand how a change in the magnitude of the pixels in the input space can affect the evaluation of the decision function (prediction score). Instead, it focuses on how the activated feature maps tend to get activated. In practice, the SMOE scale is designed to be relatively efficient and return a brief analysis at both the testing time and the learning phase by means of the computation of simple statistics; in many cases, this approach might be faster than computing the nested gradients in deep CNN model via BP algorithm.

The SMOE scale aims to give a reasonably correct representation of the information contained in the input data. For instance, when the CNN drops the "bad" input features, the flow of information should be localized in pertinent areas that are truly salient in the input space. As a result, one expects that the informativeness of some areas of the input domain should be tied with the overall activation level of the activation layers as well as their variance. In other words, the more the magnitude of

the overall activation of the feature maps, the more the input features are likely to be informative. Specifically, the SMOE scale is focused on the activated feature maps made of a sparse pattern of sharp peaks. If the feature maps turn out to be highly activated, then the input data have not been highly vanished by the activation function; thus, the data propagated throughout the hidden layers are likely to contain relevant information for the final prediction. However, when the data are propagated throughout the deeper hidden layers, the flow of information is squeezed into a lower dimensional space; therefore, it is reasonable to realize that most of the information has been filtered by the activation function, and the leftover is represented by a few sparse pixels.

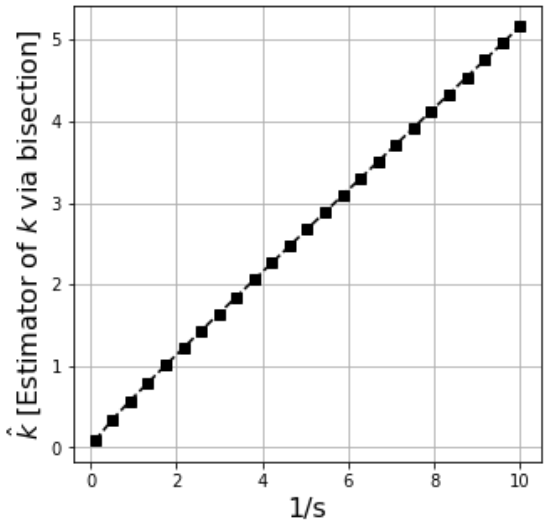


Figure 1.14: Estimated $\hat{k}$ via Bisection method as a function of the value $1/s$.

The goal of the SMOE scale is the localization of the areas where the highly informative patterns lay. To do so, this algorithm estimates some statistics of the activation values arising at different input locations. Let's consider a 1-D CNN model; we denote with $\chi_{ij} \in \mathbb{R}^+$ the values of an activated feature map with ReLU activation function (i.e., $\text{ReLU}(x) = \max(0, x)$) and denote with i and j, respectively, the spatial domain and the depth (i.e., number of time-series features). A function $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$ is applied at each point of the spatial domain χ_i all over the depth dimension. Thus, we obtain the saliency map via the relation $S = \varphi(\chi)$. The derivation of φ is obtained in the following way: one supposes the activated feature map χ to be Gamma distributed with shape parameter k and scale parameter θ. The reason for such a choice is motivated by the fact that ReLU-activated feature maps attain only real positive values; one also wants all these activation values to follow a probability distribution whose moments are all well-defined. In addition, one also considers the property of Gamma distribution of being the *maximum entropy probability distribution* for a random variable whose mean and entropy are fixed; that is, one assumes that each activation map has both a fixed mean value (i.e., the scale of the activation map) and fixed entropy (i.e., the information captured in the feature map). We search for the best parameters by means of the Maximum Likelihood Principle (Choi and Wette, 1969), namely

$$\hat{\theta}_i = \frac{\sum_{j=0}^{D} \chi_{ij}}{Nk}, \tag{1.32}$$

and

$$\log(\hat{k}_i) - \psi(\hat{k}_i) = \log\left(\frac{\sum_{j=0}^{D}\chi_{ij}}{N}\right) - \frac{\sum_{j=0}^{D}\log\chi_{ij}}{N};$$

with the sums running over the depth domain (i.e. the domain of the input features), with D the number of input features, and $\psi(x)$ the digamma function (Silverman et al., 1972). We recall that the digamma function is defined as

$$\psi(x) = \frac{d\log\Gamma(x)}{dx};$$

with $\Gamma(x)$ the well-known Euler's Gamma function (Silverman et al., 1972). Note, the estimation of the best parameters θ_i and $\hat{k}_i$ is restricted to the i-th element along the spatial domain of the activated feature map χ; that is we are therefore extracting a piece of information about the sparseness of activation along the depth domain, and not along the spatial domain. This way, we can measure the sparseness of patterns of each activated feature map along their spatial domain. However, we cannot find an estimation of $\hat{k}_i$ in a closed form, but we can let

$$s = \log\left(\frac{\sum_{j=0}^{D}\chi_{ij}}{N}\right) - \frac{\sum_{j=0}^{D}\log\chi_{ij}}{N};$$

next, we make use of the asymptotic expansion of the digamma function (Abramowitz and Stegun, 1964) and obtain the following approximation

$$\log(\hat{k}) - \psi(\hat{k}) \simeq \frac{1}{2k}\left(1 + \frac{1}{6k}\right);$$

and thus, we give a first approximation of $\hat{k}$ as

$$\hat{k} \simeq \frac{\frac{1}{4} + \frac{1}{2}\sqrt{1+3s}}{s}. \tag{1.33}$$

We can, however, refine this first estimation of $\hat{k}$ by using the Newton-Ralphson method (Ypma, 1995) (or any other method, e.g., the bisection method (Solanki et al., 2014)) and use (1.33) as an initial value. As a result, $\hat{k}$ appears to be related to $\frac{1}{s}$; as shown in figure 1.14. That figure is the result of the calculation of $\hat{k}$ by means of the Bisection method, next the fit of a linear least squares is used to ascertain the linearity between $\hat{k}$ and $1/s$. Estimating the coefficient of determination R^2 (whose value is 0.9998) reveals the linear dependency. After substituting $\frac{1}{s}$ with k in (1.32) we obtain

$$\hat{\theta}_i = \left(\frac{\sum_{j=1}^{D}\chi_{ij}}{D}\right)\left[\log\left(\frac{\sum_{j=1}^{D}\chi_{ij}}{D}\right) - \sum_{j=1}^{D}\frac{\log\chi_{ij}}{D}\right], \tag{1.34}$$

this can be finally arranged as

$$\hat{\theta}_i = \frac{1}{D}\sum_{j=1}^{D}\langle\chi\rangle\frac{\log\langle\chi\rangle}{\chi_{ij}}, \tag{1.35}$$

where

$$\langle\chi\rangle = \frac{1}{D}\sum_{j=1}^{D}\chi_{ij}.$$

Hence, (1.35) represents the *SMOE scale*, i.e., the statistics involved in the computation of the saliency maps. As one can see from (1.34), the saliency map is proportional to the activated mean value (along depth) via the term $\langle\chi\rangle$ while the second moment (also along depth) is relevant in the term in the square brackets. Indeed, if we perform a Taylor expansion of $\log\chi_{ij}$ around $\langle\chi\rangle$ we obtain,

$$\log\chi_{ij} = \log\langle\chi\rangle + \frac{\chi_{ij} - \langle\chi\rangle}{\langle\chi\rangle} - \frac{(\chi_{ij} - \langle\chi\rangle)^2}{2\langle\chi\rangle^2} + o(\chi_{ij}^2).$$

Next, applying the mean operator along the depth, we obtain

$$\frac{1}{D}\sum_{j=1}^{D}\log\chi_{ij} = \log\langle\chi\rangle - \frac{1}{D}\sum_{j=1}^{D}\frac{(\chi_{ij} - \langle\chi\rangle)^2}{2\langle\chi\rangle^2} + o(\chi_{ij}^2);$$

namely,

$$\log\langle\chi\rangle - \langle\log\chi\rangle \simeq \frac{\langle(\chi - \langle\chi\rangle)^2\rangle}{2\langle\chi^2\rangle}.$$

The simplification used in the estimation of the Gamma scale parameters (Choi and Wette, 1969) is the SMOE to the full iterative scale parameter estimation (Mundhenk et al., 2019). SMOE can be defined this way: given two saliency maps, if we sort the pixels by value, both saliency maps will be sorted the same way. The most salient locations will be the same, even if we create a binary mask with the $\nu\%$ of the most salient pixels.

By construction, we can apply the SMOE scale only on one single activated feature map; we can only estimate the informative sparseness of each activated feature map independently. Howsoever, we can combine each of these contributions to obtain an overall saliency measurement at each spatial/temporal location. This step usually involves the application of an increasing monotone function (i.e., the cumulative distribution function of a normal distribution) on each statistic θ_i. Thus we finally compute the saliency map as the weighted average of all θ_i but give the estimations arising from the deepest hidden layers higher importance. That is, we obtain the final saliency map Θ^* as

$$\Theta_i^* = \frac{\sum_{l=0}^{D^*} w_l \Phi(\theta_i^l|\mu^*,\sigma^*)}{\sum_{l=0}^{D^*} w_l}, \tag{1.36}$$

with Φ the cumulative distribution function of a normal distribution with parameters μ_l^* and σ_l^*; both these two parameters equal to the empirical mean value and standard deviation of the l-th activation feature map; the l index run over the deepness of the neural network model (D^* denotes the actual number of activated feature maps).

Before concluding this section, we can present a brief example of the XAI methods we introduced above. Let's consider the following toy data set for binary classification on Time-Series: the class 0 is generated as a rule:

$$X_0^{(n)}(t) = \sin\left(2\pi[t+\phi_n]\right),\ t\in[0,1];$$

with $X_0^{(n)}(t)$ denoting the n-th instance of the class 0, and $\phi_n \overset{i.i.d}{\sim} U(-0.125, 0.125)$; likewise, for class 1, one has

$$X_1^{(n)}(t) = -\sin\left(2\pi[t+\phi_n]\right),\ t\in[0,1].$$

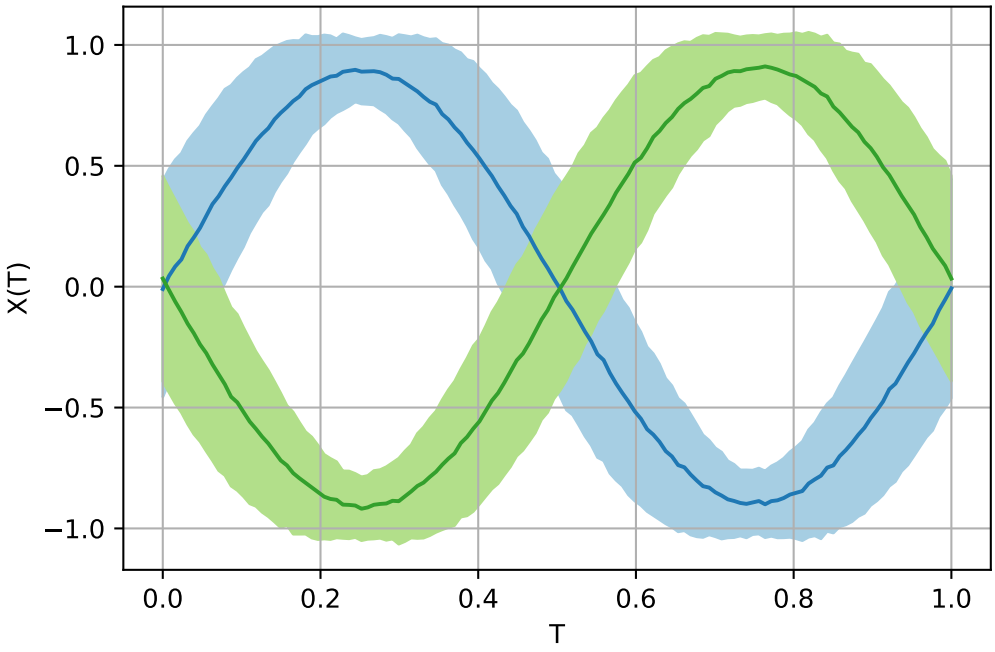


Figure 1.15: Example of the toy sine dataset. The mean samples are plotted in blue and green, respectively, for classes 0 and 1. The light-colored areas represent the standard deviation values of the mean samples.

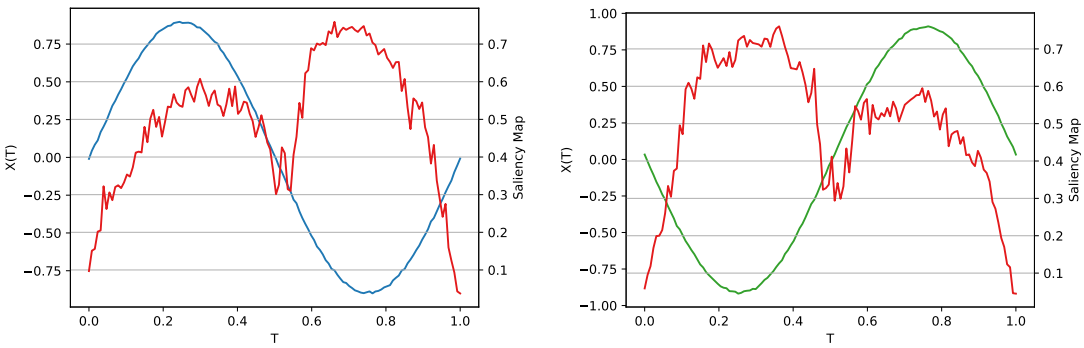
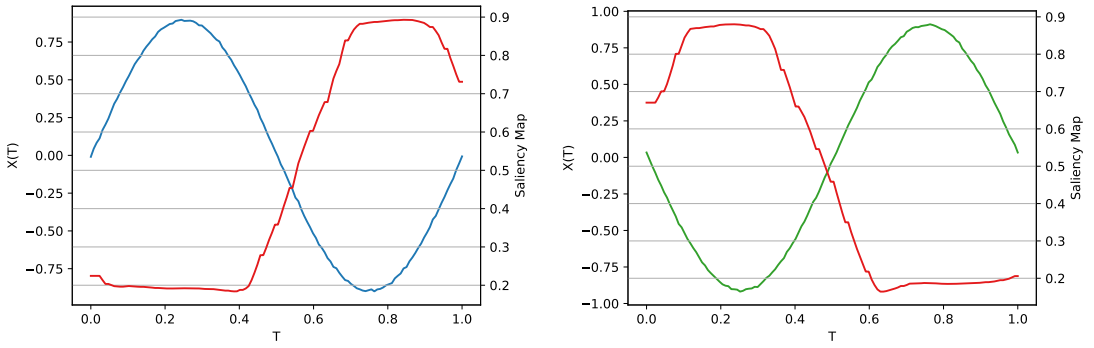


Figure 1.16: Saliency maps obtained via VG method for the sine toy dataset. (a) Mean sample (blue) and mean saliency map (red) for class 0. (b) Mean sample (blue) and mean saliency map (red) for class 1.



(a) (b)

Figure 1.17: Saliency maps obtained via SMOE Scale method for the sine toy dataset. (a) Mean sample (blue) and mean saliency map (red) for class 0. (b) Mean sample (blue) and mean saliency map (red) for class 1.

We shall refer to this dataset as *toy sine dataset*. A representation of the toy sine dataset is shown in Fig 1.15. We employ a 1-D CNN model to distinguish these two classes of signals; we use a one-held-out approach (i.e., we only evaluate the model's accuracy after making just one split into train and test set) with train size 75% (i.e., we use 75% of the dataset to train the 1-D CNN model); as a result, the Area Under the Receiver Operating Characteristic (AUROC) of the model is equal to 0.99. After testing the high accuracy of the network, we use both the two introduced to visualize the saliency maps; figure 1.16 and figure 1.17 show the saliency maps for the VG and the SMOE Scale, respectively. Considering the VG method, one can see that the saliency maps detect a salient area in correspondence with the trough of the sinusoidal oscillation, i.e., those areas of the input domain where the saliency maps achieve the highest values. As expected, the localization of the trough in two distinct areas of the input domain $T \in [0, 1]$ represents the critical feature that the 1-D CNN captures to distinguish the two classes of signals. Similarly, the SMOE scale method highlights the same salient features, i.e., the trough of the sinusoidal oscillation.

Chapter 2

Cancer diagnosis via Raman Spectroscopy

Άρμονίη άφανής φανερής κρείσσων

Ήράκλειτος

In this chapter, we present the first application of 1-D CNN (see section 1.1.3) models, i.e., the classification of Raman spectra of both healthy and tumor cell lines (e.g., melanoma and colorectal cancer). The work presented here is divided into two parts. We first deal with the melanoma case only. We want to compare here the predictive power between 1-D CNN-based classification tasks and traditional ML approaches, that is, pre-processing via either Principal Component Analysis (PCA) or Average Pooling, and classification via LR. We thus aim to show that 1-D CNN-based classification can turn out to be a promising approach to accurately distinguish Raman Spectra acquired from healthy and human skin tumor cells. The interpretation of the 1-D CNN activity is performed by means of the VG (see section 1.3.1) algorithm.

After analyzing how both 1-D CNN and ML techniques can classify melanoma data, we pass to colon data. Thus, we re-adapt these classification techniques to classify again the Raman Spectra of healthy cells tumor cell lines. Still, we consider either human skin or colon-rectal tumor lines in this case. For this purpose, the data have newly been acquired from another cell culture and have been analyzed after improving the technical instrumentation of the Raman spectrometer; the reader will find more details in section 2.2.1 and 2.2.2. The analysis is specialized on three particular sub-domains; we opted for this strategy because we want either ML techniques or 1-D CNN to be focused on recognizing a few precise physicochemical characteristics of interest contained in the Raman Spectra. More specifically, we want to validate Raman Spectroscopy as a robust and reliable methodology for a cancer diagnosis; we wish that the prediction performance of the 1-D CNN is actually based on capturing precise physicochemical properties of tumor cells. In other words, we shall show that 1-D CNN models can accurately distinguish tumor and healthy cells and base their high predictive performance on precise physicochemical characteristics of the cell samples that are contained in the Raman Spectra. Once again, we shall use the VG (see section 1.3.1) algorithm to explain the prediction provided by the 1-D CNN (section 1.1.3).

The reader will find a review of Raman spectroscopy in section 2.1. The description of the

datasets and their visualization is shown in section 2.2; Methods in section 2.4, results and comments on them are reported in section 2.5 and 2.6, respectively.

2.1 Raman Spectroscopy

Raman spectroscopy has emerged in the last decade as a powerful tool in medical diagnosis (Kong et al., 2015). With special attention to the study of cancer diseases, Liu et al. (2021) has recently shown vibrational spectroscopy is suited to decode and understand the relationship and interactions between cancer and the microbiota at the molecular level. These label-free techniques enable us to directly analyze biological samples and extract unaltered information about their physicochemical properties (Mussi et al., 2021).

Raman spectroscopy is one of the most popular techniques to study the vibrational spectroscopy of molecules; like all spectroscopy techniques, it is based on the scattering process; a scheme of a Raman spectrometer is shown in figure 2.1. *Raman scattering* process is based on the inelastic scattering of photons on a molecular sample, i.e. a laser beam interacts with a sample; the interaction between molecular vibrations, phonons, or other excitations gives rise to shifts (either positive or negative) in the energy of the laser photons (Movasaghi et al., 2007; Talari et al., 2015). These shifts are usually denoted as *Raman shifts* and mirror the structure of the atomic vibrational modes of the sample. It is important to stress that the laser beam is monochromatic with a wavelength laying in the visible spectrum, but either infrared or ultraviolet wavelengths can be selected sometimes. Although, it is not rare to use X-rays for some special cases of study (Braun et al., 2015).

The Raman shifts are usually reported in wavenumbers because this value is a direct connection to the energy shifts, i.e. the Plank law

$$E = \frac{\hbar}{2\pi\lambda}; \tag{2.1}$$

with E and λ, respectively, the energy of one photon, and λ its wavelength; $\hbar$ is the famous *reduced Planck constant*. Equation (2.1) offers a direct way to connect the Raman shifts (and their intensity) with the wavelengths involved in one Raman scattering process, namely

$$\Delta\bar{\nu} = \left(\frac{1}{\lambda_0} - \frac{1}{\lambda_1}\right); \tag{2.2}$$

where $\Delta\bar{\nu}$ denotes the Raman shift, λ_0 is the excitation wavelength (i.e., the wavelength of the incident photon), and λ_1 is the Raman spectrum wavelength. Most commonly, the unit chosen for expressing wavenumber is the inverse centimeter (cm^{-1}). Depending on the frequency of the monochromatic line, the system's response changes according to (2.2), that is, the number of scattered photons one expects to sample. The measurements of scattered photons with respect to a broad region of frequency values are usually denoted as the *Raman Spectrum* of one sample.

The inelastic scattering occurring in this kind of process has small cross sections, and consequently, the intensity of pure Raman shifts is typically very low (Petry et al., 2003). To contrast this difficulty of collecting signals with low intensity and avoid the loss of information, it is necessary to augment the intensity of Raman signals; this is possible by means of the Surface Enhanced Raman Scattering (SERS) (Stiles et al., 2008). In particular, the work of Fleischmann et al. (1974) has shown that because of electromagnetic and chemical effects, SERS can increase by many orders of magnitude the Raman signal coming from molecules adsorbed on a metal nanostructure.

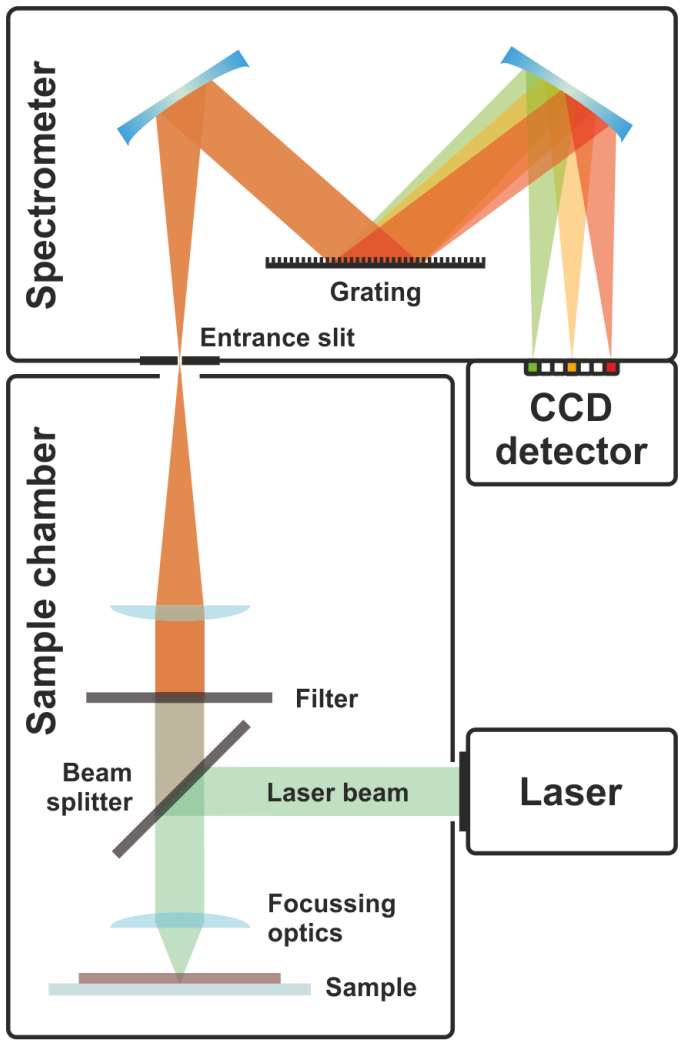


Figure 2.1: Schematic representation of one Raman Spectrometer (Schmid and Dariz, 2019; Wikipedia, 2022a).

More in specific, after the incident light interacts with the sample inside the spectrometer (a scheme of a possible spectrometer is shown in figure 2.1), most of the scattered light is usually at the same wavelength as the laser source; this elastic scattering is referred to as the Rayleigh scattering (Young, 1981) and is non-informative and concurrent with the Raman Scattering (see figure 2.2). In order to avoid collecting the signals coming with a Rayleigh scattering, it is necessary to apply a filter (e.g., a low-pass or a band-pass filter) on the scattered emission. After selecting the Raman signals only (i.e., the scattered radiation generated by Raman scattering), these signals are conveyed on a grater. A diffraction grating is a device used to separate polychromatic light (i.e., a monochromator); the groovy surface of this device enables the separation of the collected Raman signals into constituent wavelengths. The grater also controls the dispersion of light by means of the *groove density*. The higher the groove density, the more defined the spectral resolution. At last, the diffracted Raman signals are convoyed in a highly sensitive photon detector, i.e. the Charge

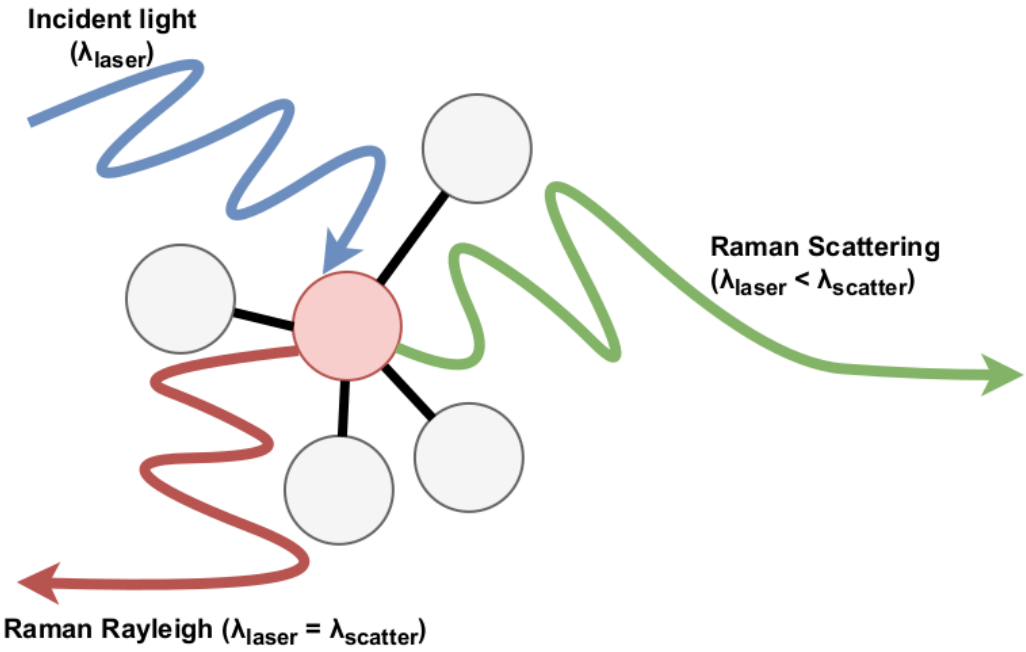


Figure 2.2: Schematic representation of the interaction between the laser beam and the sample in a spectrometer.

Coupled Device (CCD). CCD comprises plenty of array-like arranged light-sensitive lenslet, i.e. the CCD array. When the light diffracted by the grater is sent on the CCD array, all the lenslet contribute to defining the image of the Raman spectrum; each lenslet is operative on one single spectral sub-domain. For example, the first element will detect light from the lowest edge of the spectrum, whereas the last one on the upper edge of the spectrum.

In this work, Raman spectroscopy has been leveraged to investigate samples of aqueous droplets with genomic DNA. These droplets were deposited onto silver-coated silicon nanowires (Ag/SiNWs). The experimental procedure for collecting the measures of the Raman shifts followed the strategy proposed by Mussi et al. (2021). Thus, Raman spectroscopy has been used to collect several sample spectra analyzed with different statistical methods to discern whether a sample was extracted from a tumor or a healthy cell. In this context, SERS has been exploited to augment the Raman signals of a DNA aqueous solution dripped on a substrate made of a disordered array of silver-coated silicon nano-wires; this made it possible to collect several Raman spectra of the deposited drops.

The core of this work is, therefore, the resolution of binary classification problems; we compared the accuracy and the predictive performance by two classes of statistical learning methods, i.e., ML and CNN. We first use a traditional ML technique such as the LR, but we combine it with different specific pre-processing techniques (i.e., Average-Pooling, and PCA); this choice has been made in order to reduce and refine the initial amount of predictors of the input data. Concurrently, we also employ more sophisticated methods such as 1-D CNN. Unlike the former techniques, we know that CNN methods do not often need to reduce *a priori* the number of explanatory variables because this task is fully performed by the hidden layers; they encode the information of the input data propagated through the hidden layers. This fact represents a piece of the great potentiality of CNN

techniques. A fundamental and complementary aspect of this comparison is also the interpretation of the predictive performances of both classes of methods. That is, we paid special attention to the visualization and interpretation of those patterns (i.e., which peculiarly of the spectral bands in the Raman spectra) that are mainly captured by the LR and the 1-D CNN model. Thus, the predictions formulated by 1-D CNN were interpreted by means of one of the most popular gradient methods, i.e., the VG. Likewise, the prediction formulated by the LR was interpreted through the *permutation feature importance* technique (Breiman, 2001).

In this attempt to classify healthy and cancer DNA, we focus on either human skin or colon-rectal cancer. Weber et al. (2016) have proposed a classification study to distinguish melanoma one cell line (SK-MEL-28) and human immortalized keratinocyte (HaCaT); where the latter has been used as a control health skin model. In our analysis, instead, we will come up with a more general analysis that will try to include one extra melanoma cell line, i.e., the A-375 cell line. The earlier identification of colon cancer via the classification of Raman spectra has already been highlighted during the last one and a half decades (Widjaja et al., 2008; Lin et al., 2011). In our analysis, however, we made use of a larger set of Raman Spectra; where we included two types of *adenocarcinoma* cell lines, i.e., HT29 and CaCO. As for the melanoma cell lines, HaCaT cell samples have been used as a control health colon-rectal model.

We implemented single binary problems; so we adopted a *one-versus-one* strategy, i.e., we solved 6 distinct binary classification problems for each case of interest (either melanoma or colon). This approach allowed us to consider either classification problems of type healthy vs. tumor or classification problems of type tumor vs. tumor. Thus, we were also able to investigate which physicochemical features could represent the major distinguishing factor between two different cancer cell lines of the same tumor, e.g., A375 vs. SkMel, or HT29 vs. CaCO. This represents the most important contribution added by our analysis, i.e., the possibility of not only distinguishing different cell lines of the same class of tumors but also giving an interpretation of the prediction formulated in terms of the physicochemical properties of human cells.

2.2 Datasets of Raman spectra

We here present the two datasets used in all the analyses. The dataset includes samples acquired from the direct sampling of the spectral lines on the central area of the droplet of genomic DNA. These kinds of samples are usually considered the most reliable samples since the interaction between the single DNA molecules, and the nanostructured substrate is direct and tighter; the biological layer formed after dehydration is thinner and well adheres to the substrate As a result, the SERS can enhance the intensity of the Raman signals and increase the quality of the Raman spectra. The sampling of spectral lines is performed at different points of the droplet whose distance is equal to or larger than 4 μm. DNA molecules usually have a size that lies in the nano-meter order, allowing us to assume our samples are independent.

2.2.1 Experimental Setting 1

The first dataset we shall present has been used and proposed in (Durastanti et al., 2022). In that work, the substrate has been prepared as follows:

1. Plasma-enhanced chemical vapor deposition (PECVD) was used to grow Au-catalyzed SiNWs on Si wafers, kept at 350K, using SiH_4 and H_2 as precursors.

2. The coating was realized after evaporating an Ag film onto SiNWs arrays with a nominal thickness of 100 nm.

The preparation of the call samples, instead, followed the following procedure:

1. After standard culture, harvesting, and centrifugation, cell pellets were obtained and used to extract the genomic DNA, which, in turn, was re-suspended in DNase-free water to obtain a 20 $\frac{ng}{\mu L}$ solution

2. The final samples are prepared by depositing one drop of the DNA solution on Ag/SiNWs substrates coming from the very same batch.

The spectrum of each droplet is mapped after drying by means of a DXR2xi Thermo Fisher Scientific Raman Imaging Microscope equipped with a 532 nm excitation laser and a 50 objective. For each droplet, Raman spectra are collected at points on a square grid with spacing 4 μm, at 1 mW laser power, and performing four accumulations lasting 5 ms each. Each Raman spectrum acquired within this experimental setting is composed of 1680 Raman intensity values (shifts) lying in the interval between 52 and 3400 cm^{-1}. Henceforth, we refer to this experimental setting as *Experimental Setting 1*.

2.2.2 Experimental Setting 2

We shall present here two datasets. The first one has been generated within the same experimental conditions, and experimental setting of Durastanti et al. (2022), i.e., the same excitation laser (both frequency and power), the accumulations on the CCD array have the same duration, the same square grid with spacing 4 μm. However, in the dataset we used, the entire experimental devices for the detection and acquisition of Raman signals have been renewed completely, e.g. monochromator, mirrors, CCD, and Peltier. This new experimental instrumentation confers more sensibility to the Raman spectrometer; that is, it is less susceptible to heating and optical loss. As a result, the Raman spectra show a higher definition, are less blurry, and are likely to contain some patterns that could not be visible within Experimental Setting 1. Complementary to this, the DNA has been newly extracted from other cell cultures. When dealing with biological data, high variability is an essential characteristic of data. In order to keep control of this variability, we considered two different cell lines, i.e., melanoma and colorectal data.

Each Raman spectrum is thus composed of 1680 Raman intensity (shifts) values lying in the interval between 125.25 and 3399.83 cm^{-1}. We precise here that when analyzing data with this experimental setting, we divide the spectral domain into three sub-domain:

1. The *low wavenumbers* (LW) region consisting of 221 spectral points with wavenumber ranging from 125.25 cm^{-1} to 549.27 cm^{-1}

2. The *intermediate wavenumbers* (IW) region consisting of 754 spectral points with wavenumber ranging from 551.20 cm^{-1} to 2002.50 cm^{-1}

3. The *high wavenumbers* (HW) region consisting of 570 spectral points with wavenumber ranging from 2303.16 cm^{-1} to 3399.83 cm^{-1}

Henceforth, we refer to this experimental setting as *Experimental Setting 2*.

2.2.3 Pre-processing of Raman Spectra

Filtering methods have been applied to smooth each instance, i.e. we used the *Savitzky-Golay* method (Savitzky and Golay, 1964) with a filtering window of 90 pixels, treated as convolution coefficients. In figure 2.3, the average spectrum of melanoma data (experimental setting 1) is shown; while in figure 2.4 the average spectrum of both melanoma and colon data (experimental setting 2) are shown. To remove outliers and anomalous samples we have adopted the following criterion

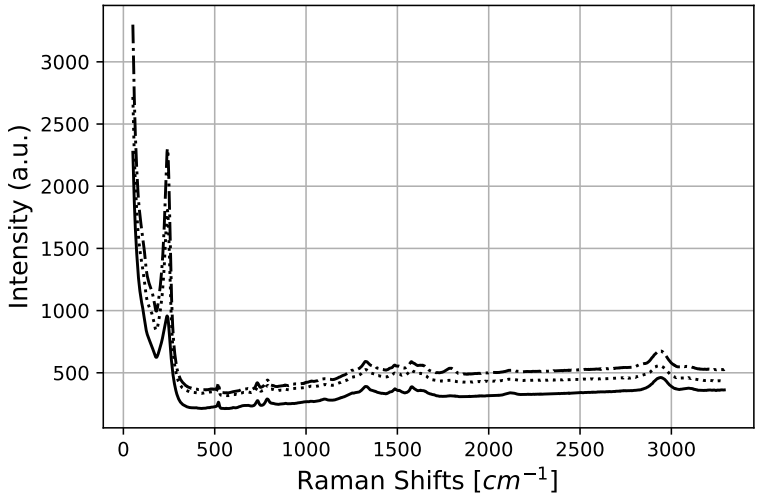


Figure 2.3: Experimental setting 1. Average Raman spectrum for the HaCaT (solid), A-375 (dotted), SK-MEL-28 (dashed) samples.

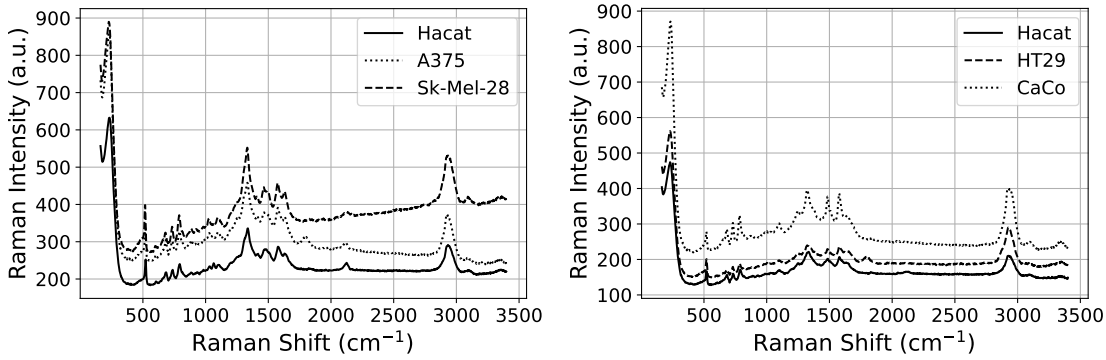


Figure 2.4: Experimental setting 2. Left: average Raman spectrum for the HaCaT (solid), A-375 (dotted), SK-MEL-28 (dashed) samples. Right: average Raman spectrum for the HaCaT (solid), CaCo (dotted), HT29 (dashed) samples.

based on the values attained by local fluctuations of Raman Spectra, that is, for both healthy and cancer spectra, we eliminate those instances that show at least one spectral line whose magnitude does not lay within 3 times the empirical (point-wise) standard deviation. In words, we compute

Cell line	Sample size	Mean value (u.a.)	Standard Deviation (u.a.)	min-max (u.a.)
HaCaT	6076	358	203	89-6769
A-375	6061	490	295	126-7827
SKM-28	6078	567	390	120-8644

Table 2.1: Experimental setting 1. Descriptive statistics of the Raman Shifts.

the average spectrum for both classes and we build a decision surface corresponding at all points, laying 3 times the empirical (point-wise) standard deviation of the mean value; if one spectrum has one Raman shift over this boundary, then the entire spectrum is discarded. For the experimental setting 1, the decision surface is reported in figure 2.5, while for the experimental setting 2 see figure 2.6. A statistical description, after the pre-processing procedures, of all the Raman Spectra of the Experimental Setting 1 is reported in Table 2.1, while for the experimental setting 2 see Table2.2. Note that Raman Spectra represent energy measurements which are always reported in Atomic Units (a.u.).

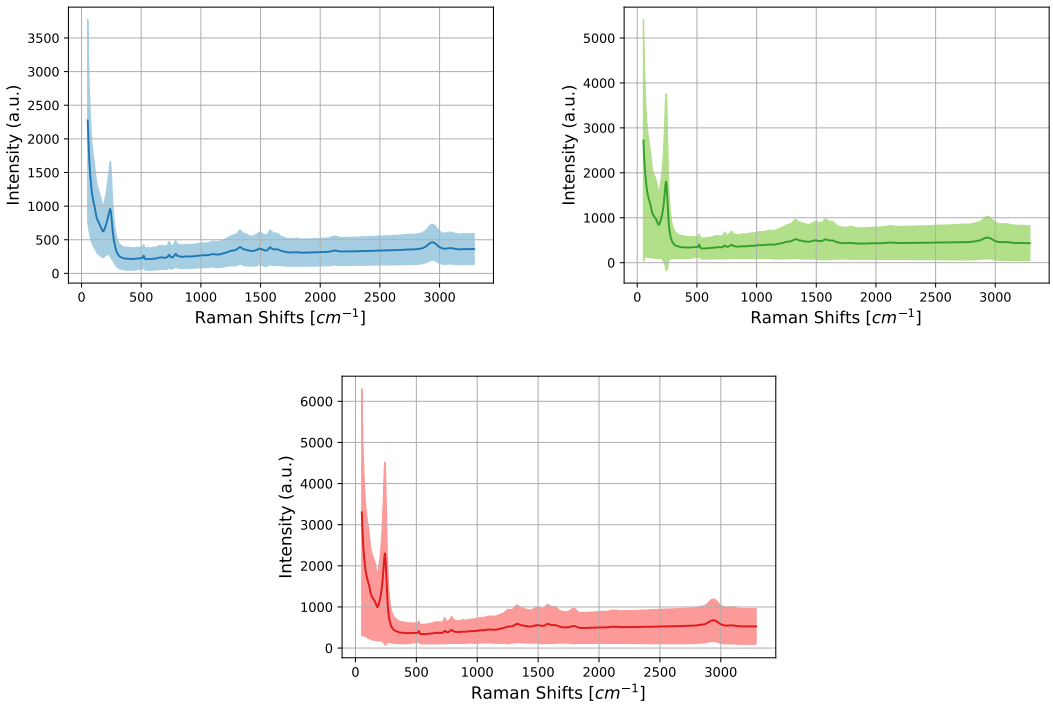


Figure 2.5: Experimental setting 1. Decision surface of the Raman spectra (light colored) and the average instance of melanoma data (solid line). HaCaT (blue), A-375 (green), and SK-MEL-28 (red). The light-colored area represents the decision surface.

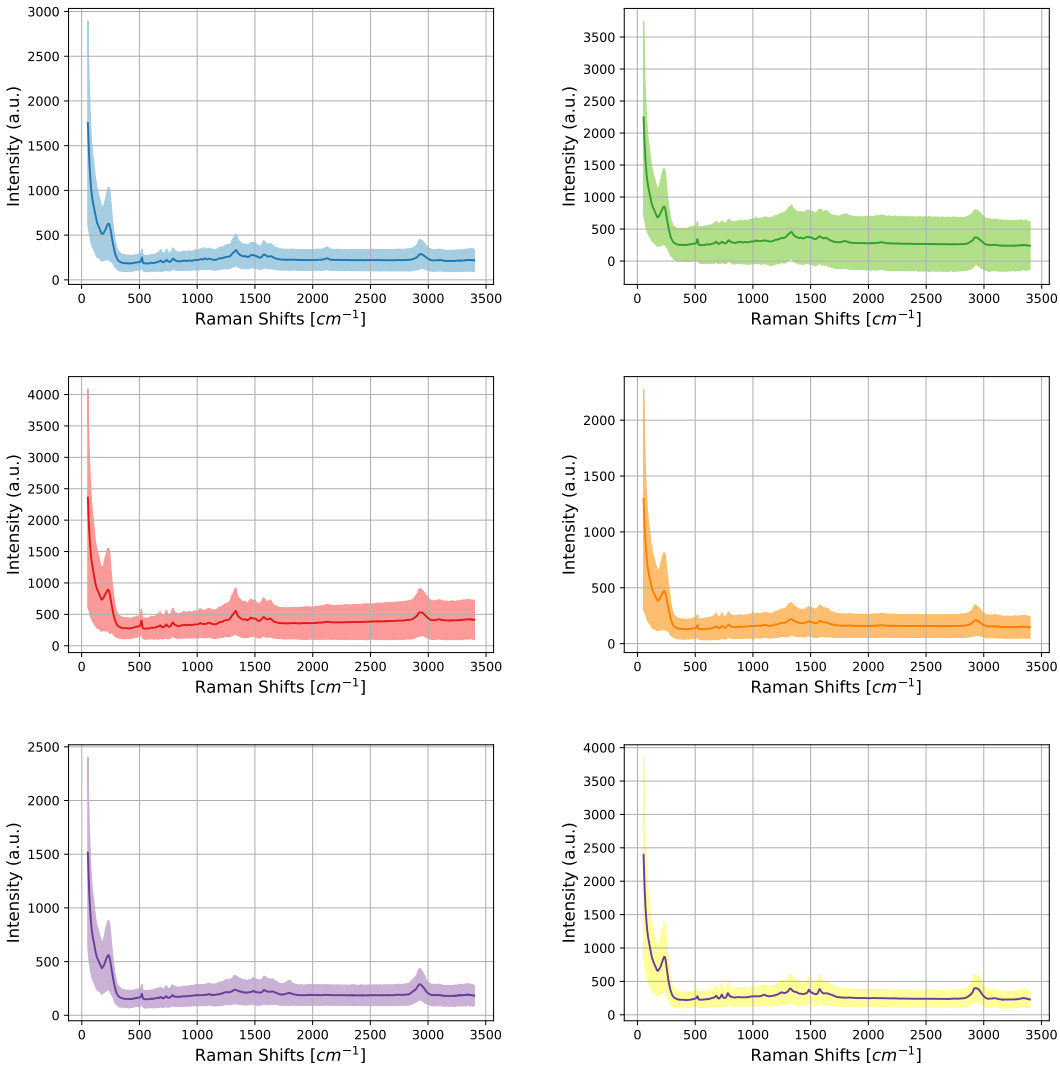


Figure 2.6: Experimental setting 2. Decision surface with the average instance of both melanoma and colon-rectal data. HaCaT melanoma (blue), A-375 (green), SK-MEL-28 (red), HaCaT colon (orange), HT29 (purple), and CaCo (brown). The light-colored area represents the decision surface.

Cell line	Sample size	Mean value (u.a.)	Standard Deviation (u.a.)	min-max (u.a.)
HaCaT (melanoma)	1645	259	154	89-3730
A-375	1672	335	232	98-5750
SKM-28	1667	419	216	107-6277
HaCaT (colon)	1619	186	118	49-3732
HT29	1619	219	131	29-2915
CaCO	1651	312	208	100-6219

Table 2.2: Experimental setting 2. Descriptive statistics of the Raman Shifts.

2.3 Physicochemical properties of biological samples

As introduced above, Raman spectroscopy represents a powerful technique for investigating the composition and the chemical bonds of either solid or fluid samples; for example, Raman spectroscopy finds an application with the determination of the structural and electronic properties of carbon nanowires (Milani et al., 2015). We report here a description (with the location in the Raman spectral domain) of the main physicochemical properties that one might detect when Raman spectroscopy is applied to biological samples; we used the works of Talari et al. (2015) and Mussi et al. (2021) as a reference. Thus, starting with the lowest spectral domain, we have:

- **230 cm^{-1}** Ag —— N stretching vibration mode
- **512-514 cm^{-1}** Si
- **600-800 cm^{-1}** Nucleotide conformation
- **614 cm^{-1}** Cholesterol ester
- **678 cm^{-1}** Ring breathing mode in the DNA bases
- **735 cm^{-1}** C —— S stretch (thiocynnate peak)
- **788 cm^{-1}** O —— P —— O stretching DNA
- **800-1200 cm^{-1}** BackBone geometry and phosphate ion interactions
- **920 cm^{-1}** C —— C stretch of proline ring/glucose/lactic acid
- **1177-1185 cm^{-1}** Cytosine, Guanine, Adenine
- **1200 cm^{-1}** Nucleic acids and phosphates; aromatic C —— O and C —— N
- **1320 cm^{-1}** Cathedral Peak
- **1335 cm^{-1}** Guanine
- **1421 cm^{-1}** Ring breathing modes of the DNA/RNA bases (Adenine, Guanine)
- **1520 cm^{-1}** Cathedral Peak
- **1575 cm^{-1}** Ring breathing modes of the DNA/RNA bases

- **1608-1610 cm^{-1}** Cytosine (NH_2)
- **1634 cm^{-1}** Amide I
- **1650 cm^{-1}** C$=$C Amide I
- **1652-1653 cm^{-1}** C$=$C stretch (Lipid)
- **1674 cm^{-1}** C$=$C stretch vibration
- **1716-41 cm^{-1}** C$=$O
- **1750 cm^{-1}** C$=$O stretch (Lipid)
- **2700-3100 cm^{-1}** CH_2, and CH_3 symmetric and anti-symmetric stretching
- **2928 cm^{-1}** CH_3 stretch
- **2929-40 cm^{-1}** CH_2 anti-symmetric stretch
- **2956 cm^{-1}** CH_3 symmetric stretch
- $>$ **3000 cm^{-1}** CH stretching

2.4 Methods for the classification of Raman spectra

2.4.1 Logistic regression on the global average (LRA)

The first model we propose is based on a very simple idea, i.e. one uses the global mean as the unique predictor of a logistic regression model. In other words, one uses the mean value to represent the Raman Spectra. The idea behind such a simplistic approach lay in the attempt to provide a tool to classify Raman Spectra without involving highly detailed data. We aim to see if a piece of super basic information, such as the mean value, can be sufficient to distinguish healthy and tumor cells when the classification task is performed over particular spectral subdomains of the Raman Spectra. The LR model is penalized by means of a ridge regularization with the shrinkage parameter equal to 1. For the validation of the model, we use the 10-fold cross-validation; we assess the model's goodness with the average AUROC (often denoted as AUC, i.e., Area Under the Curve) over the 10 cross-validation folds. We do not need to inspect the predictors here; we don't need to utilize the permutation feature importance technique to interpret the prediction provided by the analysis of a single predictor.

2.4.2 Logistic regression on average pooling (LRP)

Another method we propose is still based on a LR model, but the input features are obtained by applying the average pooling operator on the input spectra. Thus, one divides the spectral domain of each spectrum into non-overlapped and equispaced sub-domains and computes the mean value per each one of those. Such an approach has the scope of pre-processing and representing the profile of each spectrum with a lower number of explanatory variables. Depending on the binary task to solve, one can opt for a finer or coarser partitioning of the spectral domain; in our case, we opted

for a 16-feature partitioning. The choice of 16 features is motivated by the fact that 16 features are sufficient for the PCA method to explain at least 99% of the total empirical variance of data see section 2.4.3. Thus, to compare this model with the PCA method, we used the LRP model with a 16-feature description.

The LR model is here penalized by ridge regularization with the shrinkage parameter equal to 1. To validate the model, we use the 10-fold cross-validation. As for LRA, we assess the model's goodness employing the average AUROC over the 10 cross-validation folds.

To examine which input features mainly support the predictions of the LR model, we use the *permutation importance* technique Breiman (2001); see section 2.4.5. This technique is often employed to inspect if a random shuffling of one column feature can cause a drastic decrease in the model's accuracy. Random permutations of a column feature might break the correlations between that specific column feature (i.e., the data of an explanatory variable) and the target variables; when this occurs, it causes a drop in the predictive performance of the model. To quantify the degradation of the model's accuracy caused by the permutations on one column feature, we evaluate the difference between the AUROC of the model and the average AUROC after applying a finite number of permutations. We denote such a quantity as the *importance score* of one column feature.

2.4.3 Logistic regression on PCA components (PCA)

The last method LR-based method that we present consists of a pre-processing made via PCA; the classification task is still solved via LR. PCA represents an adaptive linear transformation used to reduce the number of explanatory variables of a dataset while keeping the highest level of information (Jolliffe and Cadima, 2016). It is also known as *Karhunen-Loève transform*. We recall that PCA can be formulated as a *Lagrangian Multipliers* problem, where one aims to find the best orthonormal vectors $\upsilon^{(1)}, \upsilon^{(2)}, \dots, \upsilon^{(N)}$, maximizing the Lagrangian

$$\begin{aligned}\mathcal{L}(\upsilon^{(1)}, \upsilon^{(2)}, \dots, \upsilon^{(N)}, \lambda^{(1)}, \lambda^{(2)}, \dots \lambda^{(N)}) = \sum_{k=0}^{N}\sum_{i=0}^{M}\sum_{j=0}^{M} \upsilon_i^{(n)} S_{ij} \upsilon_j^{(n)} \\ + \sum_{k=0}^{N}\sum_{l=0}^{N} \lambda^{(k)} \left(\delta_{k,l} - \sum_{j=0}^{M} \upsilon_j^{(k)} \upsilon_j^{(k)} \right);\end{aligned} \tag{2.3}$$

with S the empirical covariance matrix of a multivariate dataset of M explanatory variables, $\lambda^{(k)}$ the k-th lagrangian multiplier, and δ denotes the Kronecker delta. We assume that the multivariate dataset can be of e represented as a maxi size (N_0, M) (where N_0 is the number of samples, and M the number of explanatory variables); we denote with $\mathbf{X}_{ij}$ the observation of the i-th sample and the j-th explanatory variable. So, the empirical covariance matrix is obtained as

$$S_{ij} = \frac{1}{N_0} \sum_{k=0}^{N_0} (X_{ki} - \tilde{X}_i)(X_{kj} - \tilde{X}_j);$$

with

$$\tilde{X}_l = \frac{1}{N_0} \sum_{k=0}^{N_0} X_{kl}, \forall l \in \{1, 2, \dots, M\}.$$

The calculation of the gradient of the lagrangian, with respect to both vectors υ and multipliers λ, leads to

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \upsilon_n^m} = \sum_{j=0}^{M} S_{nj} \upsilon_n^{(m)} - \lambda^{(m)} \upsilon_n^{(m)} \\ \frac{\partial \mathcal{L}}{\partial \lambda^{(m)}} = \sum_{l=0}^{M} \left[\delta_{ml} - \sum_{j=0} \upsilon_j^{(m)} \upsilon_j^{(l)} \right]. \end{cases} \tag{2.4}$$

At this point one can imposes both equation of (2.4) to be zero; and then next multiply at left the first equation of (2.4) for all υ vectors, namely

$$\sum_{q=0}^{N} \left[\sum_{i=0}^{M} \sum_{j=0}^{M} \upsilon_i^{(q)} S_{ij} \upsilon_j^{(m)} - \lambda^{(m)} \sum_{j=0}^{M} \upsilon_j^{(q)} \upsilon_j^{(m)} \right] = 0. \tag{2.5}$$

when plugging the second equation of (2.4) into (2.6), one obtains

$$\sum_{q=0}^{N} \left[\sum_{i=0}^{M} \sum_{j=0}^{M} \upsilon_i^{(q)} S_{ij} \upsilon_j^{(m)} - \lambda^{(m)} \sum_{j=0}^{M} \delta_{m,q} \right] = 0. \tag{2.6}$$

Equation (2.6) vanishes if, for any value of q, yields

$$\sum_{i=0}^{M} \sum_{j=0}^{M} \upsilon_i^{(q)} S_{ij} \upsilon_j^{(m)} - \lambda^{(m)} \sum_{j=0}^{M} \delta_{m,q} = 0. \tag{2.7}$$

Equation (2.7) is equivalent to say that the set of vectors $\upsilon^{(1)}, \upsilon^{(2)}, \dots, \upsilon^{(N)}$ diagonalize the empirical covariance matrix. PCA method is equivalent to applying a linear transformation that spans the input data into the space of the orthogonal eigenvectors of the empirical covariance matrix. Note that, by construction, any observation can be described as a superposition of the vectors $\upsilon^{(1)}, \upsilon^{(2)}, \dots, \upsilon^{(N)}$; in addition, the eigenvalues of the empirical correlation matrix are equivalent to the empirical variance of the explanatory variables. Thus, we denote with $\lambda^{(m)}$ the empirical variance of the m-th input feature. The explained variance of the m-th input feature is therefore obtained as

$$\tilde{\lambda}^{(m)} = \frac{\tilde{\lambda}^{(m)}}{\sum_{k=0}^{N} \tilde{\lambda}^{(k)}}.$$

To select the number of components υ to determine with the PCA method, we can pick up any set of components explaining at least 99% of the total empirical variance; for more details, see section 2.5. After applying the PCA algorithm to the input data, the LR model is fitted on these data; like in the other models, we use a ridge regularization with the shrinkage parameter equal to 1. The model is validated after using 10-fold cross-validation. As for LRP, we assess the model's goodness by means of the average AUROC over the 10 cross-validation folds.

2.4.4 1-D CNN for classification of Raman Spectra

1-D CNN (see section 1.1.3) represents the last model used to analyze peculiarities of Raman Spectra of healthy and tumor cancer cells. The natural advantage of using this model class is avoiding selecting a desirable set of explanatory variables, as we did for the LR model. Indeed, this operation is naturally executed during the learning phase of the 1-D CNN model, when input data are propagated through the hidden layers. However, implementing such a class of methods requires optimizing several hyper-parameters. In addition, since we desire the 1-D CNN model to be compared

Binary Problem	Filters	Kernel size	Dropout rate	Deepness	Activation	Learning Rate
HaCaT-A375	4	17	0.25	2	Rational	$5 \cdot 10^{-4}$
HaCaT-SK-MEL-28	4	17	0.25	2	Rational	$5 \cdot 10^{-4}$
A375-SK-MEL-28	4	17	0.25	2	Rational	$5 \cdot 10^{-4}$

Table 2.3: Exeperimental setting 1. Table with the best hyperparameters configuration for the 1-D CNN models.

Binary Problem	Filters	Kernel size	Dropout rate	Deepness	Activation	Learning Rate
HaCaT-A375 (LW)	16	3	0.25	3	Softplus	$5 \cdot 10^{-4}$
HaCaT-SK-MEL-28 (LW)	16	3	0.25	3	Softplus	10^{-3}
A375-SK-MEL-28 (LW)	16	3	0.25	3	Softplus	10^{-3}
HaCaT-HT29 (LW)	16	3	0.25	3	Softplus	10^{-3}
HaCaT-CaCO (LW)	16	3	0.25	3	Softplus	10^{-3}
HT29-CaCO (LW)	16	3	0.25	3	Softplus	10^{-3}
HaCaT-A375 (IW)	16	3	0.25	3	Softplus	10^{-3}
HaCaT-SK-MEL-28 (IW)	16	3	0.25	3	Softplus	10^{-3}
A375-SK-MEL-28 (IW)	16	3	0.25	3	Softplus	10^{-3}
HaCaT-HT29 (IW)	16	3	0.25	3	Softplus	10^{-3}
HaCaT-CaCO (IW)	16	3	0.25	3	Softplus	10^{-3}
HT29-CaCO (IW)	16	3	0.25	3	Softplus	10^{-3}
HaCaT-A375 (HW)	16	3	0.25	3	Softplus	10^{-3}
HaCaT-SK-MEL-28 (HW)	16	3	0.25	3	Softplus	10^{-3}
A375-SK-MEL-28 (HW)	16	3	0.25	3	Softplus	10^{-3}
HaCaT-HT29 (HW)	16	3	0.25	3	Softplus	10^{-3}
HaCaT-CaCO (HW)	16	3	0.25	3	Softplus	10^{-3}
HT29-CaCO (HW)	16	3	0.25	3	Softplus	10^{-3}

Table 2.4: Exeperimental setting 2. Table with the best hyperparameters configuration for the 1-D CNN models.

with the LR models described above, we need to impose our 1-D CNN model to be qualitatively comparable with the LR introduced above, that is we impose the final prediction layer (i.e., the last fully-connected layer) to be fed by a flattened latent representation with dimension 16. This sort of additional constraint aims to figure out whether the pre-processing taking place in the hidden layers of the 1-D CNN model can effectively extract some detailed information that cannot be captured by the ML techniques introduced in the previous models.

The tuning of the hyper-parameters consists of controlling a more significant number of parameters, i.e., the dropout rate, deepness, number of filters (per hidden convolutional layer), kernel size, and activation function. We selected the best parameters by running the 1-D CNN model over a fine grid of configurations; the final selection was made by choosing that configuration whose overall AUROC (we used the 10-fold cross-validation) attained the highest value. Depending on the binary problem to solve and the experimental setting used, we draw a specific set of optimized hyperparameters; for the Experimental Setting 1, see Table 2.3, while for the experimental setting 2 Table 2.4. For convenience, the stride of convolutions is set equal to one, and the max-pooling size is equal to 2 (the most recommended choice (Aggarwal et al., 2018)); we adopted *valid* padding to avoid the convolution being miscalculated at the edges.

Thus, we design our 1-D CNN as follows:

- Input data are propagated through a convolutional layer; next follows the activation function $\phi(\cdot)$, and finally, the max-pooling layers with max-pooling size equal to 2. Bias terms are here involved during convolutions.

- The choice of the activation function depends on the dataset analyzed. When dealing with data from the experimental setting 1, we implemented a Rational Activation function (Boullé et al., 2020), i.e. $\phi(x) = \frac{P(x)}{Q(x)}$, with

$$P(x) = \sum_{k=0}^{2} \alpha_k x^k;$$
$$Q(x) = \sum_{k=0}^{3} \beta_k x^k.$$

  In implementing a rational activation function, one treats all the coefficients $\alpha_0, \alpha_1, \alpha_2, \beta_0, \beta_1, \beta_2, \beta_3$ as additional weights to be fitted on the input data. For the data from experimental setting 2, we used the *softplus* activation function, i.e., $\phi(x) = \log(1 + \exp x)$.

- Such a scheme is repeated a number of times equal to the deepness of the model.

- Flatten layer is used to make the activate feature maps array-like. In this stage, an extra propagation through a fully-connected layer with 16 units is performed. We repeat that such an operation is needed to make the 1-D CNN model compared with respect to the LR-based approaches.

- The final prediction is made via a dense layer with one single output with sigmoidal activation function $\sigma(\cdot)$, i.e., $\sigma(x) = \frac{1}{1+\exp -x}$.

- Dropout layers with a rate of 0.25 are positioned after each max-pooling layer to prevent overfitting.

At last, we use the VG (see section 1.3.1) method to finally interpret the pattern activity recognition of the hidden layers of 1-D CNN. We repeat that the VG (see section 1.3.1) method exploits the properties of the backpropagation algorithm to evaluate the sensitivity of the output value with respect to a change in each single input feature domain.

2.4.5 Permutation feature importance

The idea behind the permutation feature importance technique is based on the decreasing of the model accuracy as one single explanatory feature is randomly shuffled (Breiman, 2001). With this procedure, one aims to break the correlations between one specific explanatory variable and the classes of events. Accordingly, the drop in the model accuracy is informative about the dependency of the model with respect to that variable. Let's introduce this method with an example tailored to our case study. Let's consider the features $\bar{\nu}$, and the set $\bar{N} = \{\bar{\nu_1}, \bar{\nu_2}, \dots, \bar{\nu_L}\}$, with L= 16. To be closer to the models described in section 2.4.2 and 2.4.3, $\bar{\nu}$ is meant as one feature extracted either via average pooling or involving the PCA method. The features $\bar{\nu}$ feed, therefore, the LR model. We can say that $\bar{\nu}_n$ is a relevant feature for the fitted LR model if a random shuffling of $\bar{\nu}_n$ causes an increase in the model error; this is equivalent to say that the classification task made

by the LR model is mainly based on $\bar{\nu}_n$. Note that "shuffling one feature" means gathering apart that feature for all instances available, making a random shuffle, and then randomly reassigning these values to the instances. Conversely, the feature $\bar{\nu}_n$ is irrelevant if applying a random shuffling on $\bar{\nu}_n$ does not increase the model error. In order to give an estimate of how much a random shuffling of $\bar{\nu}_n$ can deteriorate the model accuracy, we can evaluate the AUROC after applying the random shuffling. The difference between the AUROC evaluated when no random shuffling and when one random shuffling is applied represents our metric to understand if $\bar{\nu}_n$ is a relevant feature. Indeed, higher values of such a discrepancy mean that if we replace one feature with another one that has no correlations, then the fitted model cannot any longer solve the classification task correctly. Note that such a method investigates the ability of the model to retrieve the correct patterns after permuting one feature; however, anything true can be argued *a priori* about the intrinsic power of some features to be more predictive than others. Instead of applying one single random shuffling, we repeat the same approach for the same feature for a fixed number K of times (but with K different and independent random shufflings).

This allows us to express the estimation of the relevancy (or *mean importance*) of $\bar{\nu}_n$ as follows

$$\mathrm{R}(\bar{\nu}_n) = \mathrm{AUCROC} - \frac{1}{K}\sum_{k=1}^{K} \mathrm{AUCROC}_k^{\bar{\nu}_n}, \tag{2.8}$$

with $\mathrm{R}(\bar{\nu}_n)$ denoting the relevancy of the input feature $\bar{\nu}_n$, AUROC is the AUROC value when no shuffling is applied to $\bar{\nu}_n$, and $\mathrm{AUROC}_k^{\bar{\nu}_n}$ the AUROC of LR when the k-th random shuffling is applied on the feature $\bar{\nu}_n$. Accordingly, we can estimate the error of R via Standard Error Mean (SEM), e.g. one could use (2.8) on each fold of a M-fold cross-validation. Instead of fitting a few distinct models and then studying the relevancy of one single feature on each trained model, the advantage that the permutation feature importance algorithm offers is that one can inspect the relevancy of one single feature by validating the accuracy of one fitted model after applying a sequence of different permutations on one desired feature. The advantage of using such a technique is that we can reuse it many times without fitting the same model a few times; we only need to apply several different permutations of one feature and then validate an accuracy matrix.

2.5 Results

To validate the approaches introduced above (LRP, LR on PCA, and 1-D CNN), we estimate the AUROC score. Thus, we compute the mean AUROC over all 10 folds of the 10-fold cross-validation; the errors are computed via SEM. The results for the models developed by using the data of the experimental setting 1 are shown in Table 2.5; while for the experimental setting 2 see Table2.6.

2.5.1 Results for the Experimental Setting 1

When employing the LR model on the data of the experimental setting 1, the distinction of tumor and healthy samples is performed with high accuracy. Indeed, the PCA-based approach shows outstanding performances with AUROC 0.99 ± 0.01, but the LRP achieves slightly lower accuracy, i.e., AUROC 0.97 ± 0.01 in HaCaT-A375 and 0.98 ± 0.01 in HaCaT-SK-MEL-28. When attempting to distinguish the two skin cancer cell lines (i.e., A375-SK-MEL-28), the AUROC values decrease

Case	LRA	LRP	PCA	1D-CNN
HaCaT-A375	0.91 ± 0.01	0.97 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
HaCaT-SK-MEL-28	0.83 ± 0.01	0.98 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
A375-SK-MEL-28	0.76 ± 0.01	0.84 ± 0.01	0.91 ± 0.01	0.96 ± 0.01

Table 2.5: Experimental Setting 1. AUROC $\pm$ SEM for each model developed.

Case	LRA	LRP	PCA	1D-CNN
HaCaT-A375 (LW)	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
HaCaT-SK-MEL-28 (LW)	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
A375-SK-MEL-28 (LW	0.61 ± 0.01	0.88 ± 0.01	0.92 ± 0.01	0.92 ± 0.01
HaCaT-HT29 (LW)	0.75 ± 0.01	0.93 ± 0.01	0.99 ± 0.01	0.95 ± 0.01
HaCaT-CaCO (LW)	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
HT29-CaCO (LW)	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
HaCaT-A375 (IW)	0.99 ± 0.01	0.99 ± 0.01	0.94 ± 0.01	0.98 ± 0.01
HaCaT-SK-MEL-28 (IW)	0.99 ± 0.01	0.96 ± 0.01	0.98 ± 0.01	0.97 ± 0.01
A375-SK-MEL-28 (IW)	0.74 ± 0.01	0.87 ± 0.01	0.94 ± 0.01	0.99 ± 0.01
HaCaT-HT29 (IW)	0.76 ± 0.01	0.93 ± 0.01	0.95 ± 0.01	0.99 ± 0.01
HaCaT-CaCO (IW)	0.99 ± 0.01	0.97 ± 0.01	0.98 ± 0.01	0.99 ± 0.01
HT29-CaCO (IW)	0.97 ± 0.01	0.99 ± 0.01	0.95 ± 0.01	0.99 ± 0.01
HaCaT-A375 (HW)	0.96 ± 0.01	0.99 ± 0.01	0.95 ± 0.01	0.96 ± 0.01
HaCaT-SK-MEL-28 (HW)	0.96 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
A375-SK-MEL-28 (HW)	0.96 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.98 ± 0.01
HaCaT-HT29 (HW)	0.96 ± 0.01	0.99 ± 0.01	0.95 ± 0.01	0.99 ± 0.01
HaCaT-CaCO (HW)	0.96 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01
HT29-CaCO (HW)	0.96 ± 0.01	0.93 ± 0.01	0.99 ± 0.01	0.99 ± 0.01

Table 2.6: Experimental Setting 2. AUROC $\pm$ SEM for each model developed.

for both the approaches, that is, PCA accuracy decrease to 0.91 ± 0.01, while LRP to 0.84 ± 0.01. A LR model fed with only the mean value of Raman spectra shows the lowest results, HaCaT-A375 AUROC 0.91 ± 0.01, HaCaT-SK-MEL-28 AUROC 0.83 ± 0.01, and A375-SK-MEL-28 AUROC 0.76 ± 0.01. When fitting the 1-D CNN to these three cases of study, form Table2.5, one can see that the model can distinguish with the maximal accuracy (AUROC 0.99 ± 0.01) both the cases HaCaT-A375 and HaCat-SK-MEL-28; unlike the LR approach, for the case, A375-SK-MEL-28, the 1-D CNN model can still achieve high accuracy (AUROC 0.96 ± 0.01), higher than the PCA-based approach.

2.5.2 Results for the Experimental Setting 2

When considering the data acquired with Experimental Setting 2, one can see that both LR and 1-D CNN models can still distinguish healthy and tumor samples with high accuracy.

When applied in the LW region, a basic method as the LRA can perfectly classify almost all binary problems involving healthy and tumor cells; the unique exception is represented by the case A375-SK-MEL-28, where the AUROC drops to 0.61 ± 0.01. In opposition to this, the case HT29-

CaCO is perfectly solved by the LRA method (AUROC 0.99 ± 0.01). Similarly to the LRA, both the LRP and PCA can achieve the same results, but being more precise with the critical case A375-SK-MEL-28; where the latter shows higher accuracy with respect to the former (i.e., LRP AUROC 0.88 ± 0.01, and PCA AUROC 0.92 ± 0.01). Another critical case as HaCat-HT29 is perfectly solved by the PCA; LRP shows, howsoever, outstanding performance with AUROC 0.93 ± 0.01. 1-D CNN also shows highly accurate predictive performances when dealing with health and tumor cells; almost all cases are perfectly solved with AUROC 0.99 ± 0.01. The case HaCat-HT29 is solved with a little more uncertainty (i.e., AUROC 0.95 ± 0.01), but with the case, A375-SK-MEL-28 the 1-D CNN model is as accurate as the PCA.

In analyzing the IW region, one can see from Table2.6 that all methods describe a scenario similar to the LW region; the classification of healthy and tumor cells are solved with high accuracy by almost all methods. In particular, the case A375-SK-MEL-28 is partially solved by the LRA (AUROC 0.74 ± 0.01), LRP and PCA can improve the predictions with AUROC, respectively, 0.87 ± 0.01 and 0.94 ± 0.01, but 1-D CNN can perfectly solve this problem with AUROC 0.99 ± 0.01. Likewise, the case as HaCat-HT29 is partially solved by the LRA (AUROC 0.76 ± 0.01), LRP and PCA can achieve more accurate results with AUROC 0.93 ± 0.01 and 0.95 ± 0.01, respectively; but perfect discrimination is performed by the 1-D CNN.

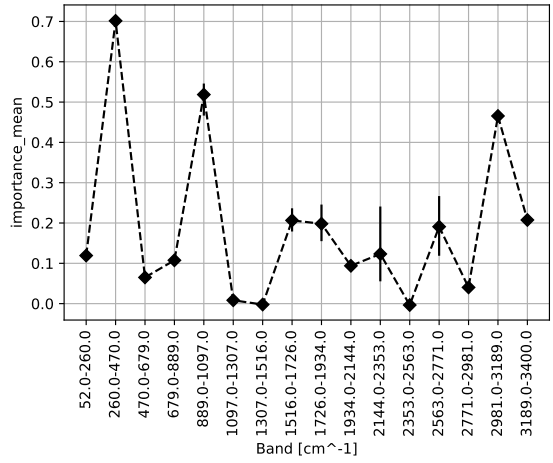
The resolution of the binary with the data extracted from the HW region lead to straight results, i.e. the classification of Raman spectra of either healthy vs. tumor cells or tumor vs. tumor cells is always highly accurate (AUROC≥ 0.95) with all methods. Such a result is regardless of the employment of human skin tumor or colorectal data. In all cases analyzed, the errors of the AUROC are all equal, i.e., 0.01.

2.6 Interpretation and discussion

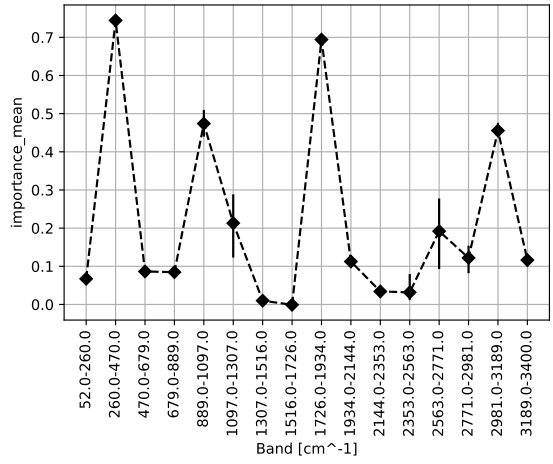
To interpret the prediction performance of all the methods used, we apply the Permutation feature importance algorithm on the fitted LRP models, saliency maps via the VG (see section 1.3.1) method on the 1-D CNN models. For the PCA we visualize and comment on the normalized eigenvectors of the empirical covariance matrix (here denoted as components or loadings). We start with the data acquired with Experimental Setting 1.

2.6.1 Interpretation for Experimental Setting 1

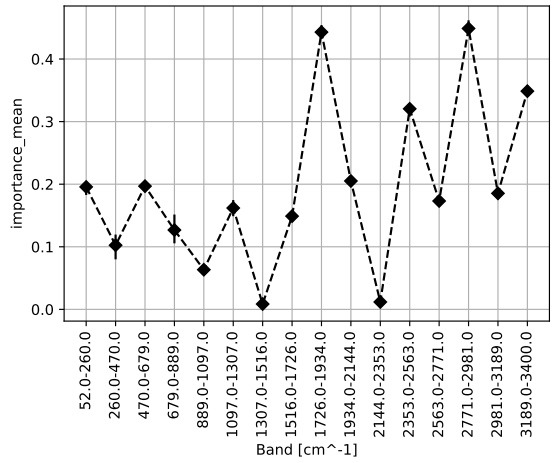
The *mean importance* of the LRP models is shown in figure 2.7. For the case, HaCaT-A375 the highest values of the mean importances (where the relevancy is higher than 0.4) are at bands 260-470, 869-1097, and 2981-3189 cm^{-1}; while for the case HaCaT-SK-MEL-28 the LR base its recognition activity on the features located at bands 260-470, 869-1097, 1726-1934 and 2981-3189 cm^{-1}; see figure 2.7a and 2.7b. Thus for both cases including healthy and tumor cells of experimental setting 1, the LRP method can focus its attention on almost the exact characteristics of input spectra. In particular, the band 2981-3189 cm^{-1} contains one important characteristic of the samples, i.e. the stretch of the vibrational modes of the group CH; thus, a proper biochemical property of DNA. Another important band is 869-1097, where one finds the stretch of C——C bounds of proline; the metabolism of proline is related to ATP production in tumor cells (Geng et al., 2021).In opposition to this, the inspection of the features for case A375-SK-MEL-28 does not reveal any important characteristic in the IW region, except for the band 1726-1932 cm^{-1}



(a)



(b)



(c)

Figure 2.7: Experimental Setting 1. Relevance Importance of LRP for (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, and (c) A375-SK-MEL-28.

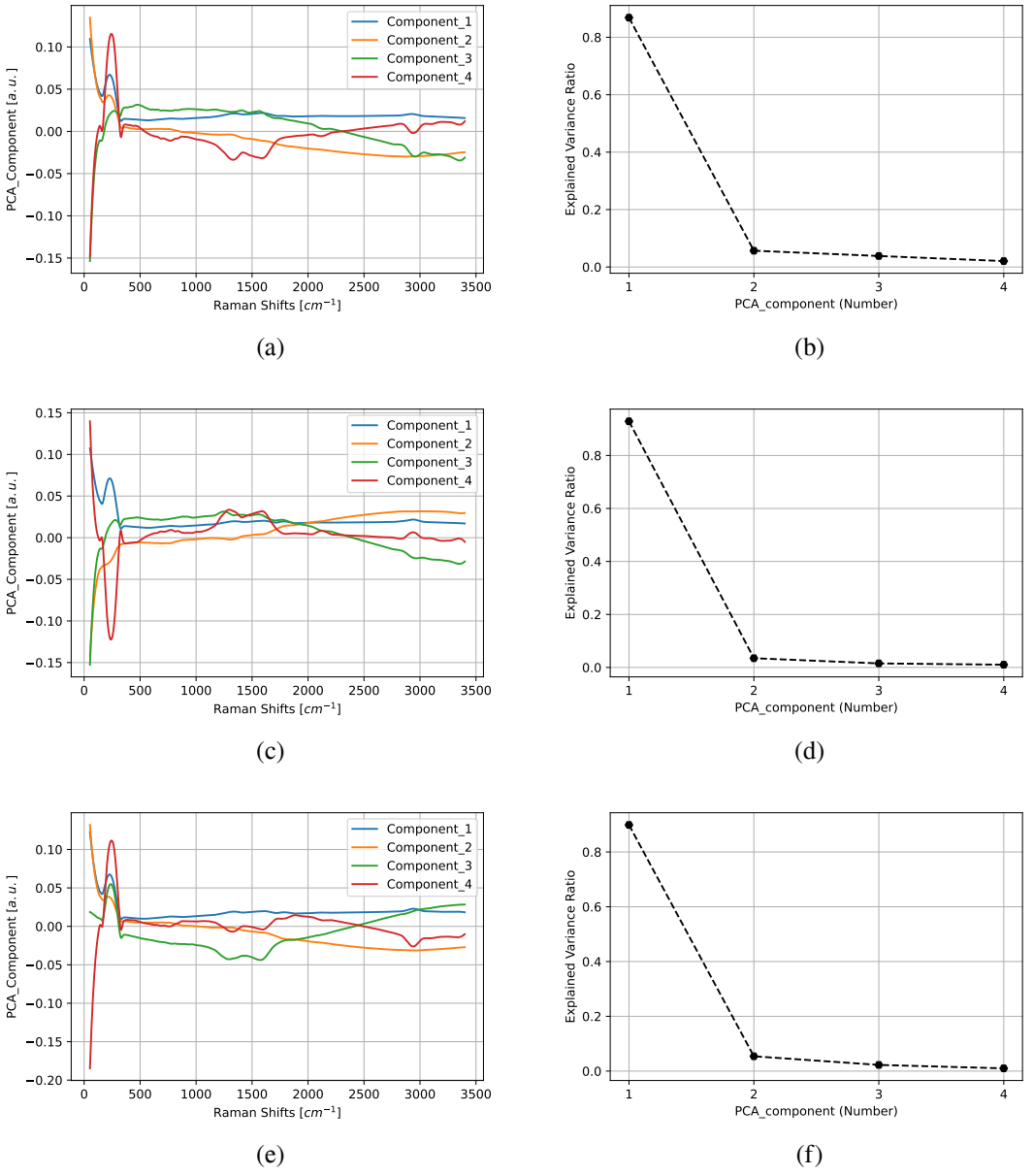
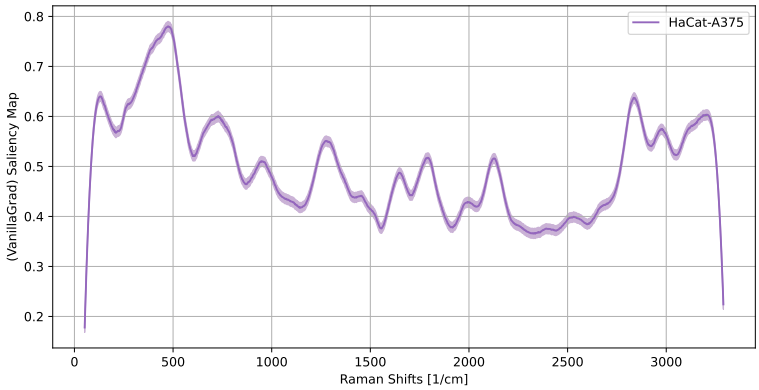
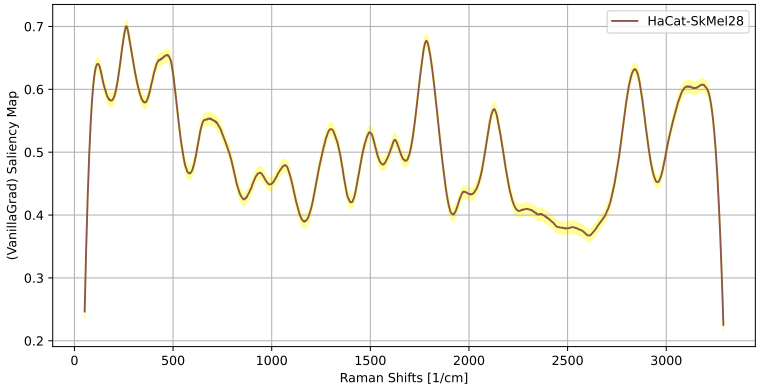


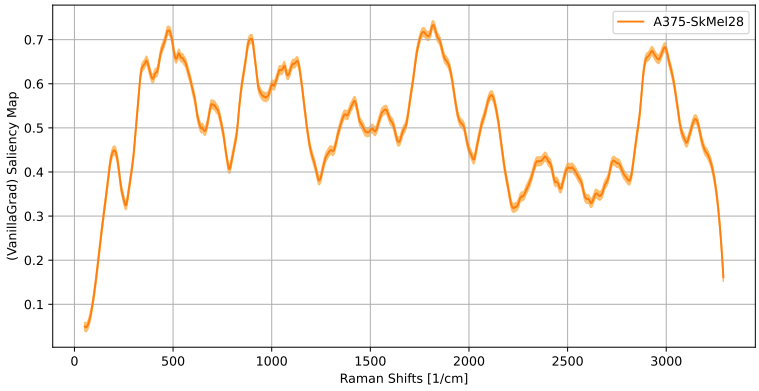
Figure 2.8: Experimental Setting 1. The first four components of the PCA algorithm with their explained variance ratio for the cases (a-b) HaCaT-A375, (c-d) HaCaT-SK-MEL-28, and (e-f) A375-SK-MEL-28. The first component of PCA is depicted in blue, the second in orange, the third in green, and the fourth in red.



(a)



(b)



(c)

Figure 2.9: Experimental Setting 1. Saliency Map via VG algorithm for cases (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, and (c) A375-SK-MEL-28.

where one obtains an important mean of 0.43; see figure 2.7c We recall that band 1726-1932 cm $^{-1}$ indicate a change in the C$=$O of the acyl chains of membrane lipids; abnormal changes in colon rectal and human skin tumor cells have already been reported by Rigas et al. (1990) and Kyriakidou et al. (2017). That band mirrors the interaction between the sample and the nanowires substrate, but it usually has no direct physical interpretation. Bands in the HR region, like 2981-3189 cm^{-1}, are also relevant for case A375-SK-MEL-28 with mean importance larger than 0.4.

The visualization of the PCA components (with the explained variance ratio of each component) is shown in figure 2.8. In all three cases considered, one can see from figure 2.8b, figure 2.8d, and figure 2.8c that almost 90% of the variance is explained by the first component, and barely 10% by the second component. The PCA components (see figure 2.8b, figure 2.8d, and figure 2.8c) reveal that the PCA algorithm mainly captures patterns composed of a peak in the region 200-250 cm^{-1} and constant spectrum elsewhere (i.e., the first component), other second-order" patterns still include a broader peak in the region 200-250 cm^{-1} and a flat pattern at 1700-3390 cm^{-1}, the rest of the spectrum attaints values close zero. Thus, the combination of the interaction of the biological sample with the nanowires and the fact that spectra of different classes can differ from a constant factor is the main feature that enables one to classify the Raman Spectra of either healthy and skin tumor cells or two different types of skin tumors cells. We recall that the region 200-250 cm^{-1} contains a peak (usually centered at 230 cm^{-1}) indicating the presence of biological molecules with nitrogen adsorbed on metallic nanostructures (Mussi et al., 2021)

The visualization of the saliency maps (see figure 2.9) indicates, for case HaCaT-A375, some saliency regions (i.e., we look at all domains where the Saliency Map attains values larger than 0.6) in the proximity of 500 cm^{-1} and 2900 cm^{-1}; see 2.9 a. Both these characteristics are related to two physicochemical properties of the samples, i.e., the group CHs at 2930 cm^{-1}, and the interaction between the samples and the silicon in the nanowires of the substrate (the latter is usually detected with a peak at 500 cm^{-1} in the Raman spectra). Likewise, for case HaCaT-SK-MEL-28 (see 2.9 b) the most salient regions are also located in the proximity of 500 cm^{-1} and 2900 cm^{-1}; in this case we need to consider the characteristics of the spectra at 1780-1820 cm^{-1}, i.e., some interactions between samples and nanowires substrate that do not have a direct physicochemical interpretation. The visualization of the saliency map of case A375-SK-MEL-28 (see 2.9 c) shows plenty of salient characteristics with respect to the previous cases. In addition to the characteristics of both the regions near 500 cm^{-1} and 2900 cm^{-1}, the 1-D CNN can distinguish these two human skin tumors by capturing relevant patterns occurring at 900-950 cm^{-1}, and 1700-1800 cm^{-1}. The first region (900-950 cm^{-1}) contains the C$-$C stretching of amino acids proline and valine, while the band 1700-1800 cm^{-1} contains information on the structure of lipidic molecules (when this kind of molecules are examined via Rama n Spetrocsooy, one usually detect the peaks at 1754 cm^{-1}; these peaks are related to the bounds C$=$O) (Talari et al., 2015).

The study of the pattern recognition activity of all the three models developed brought to light that the predictions of LRP and 1-D CNN are based on two characteristics of the Raman Spectra of healthy and malignant cells, i.e., the interaction of the samples with both the substrate of nanowires and the group CH. The first reflects how the droplets of genomic DNA tend to get displaced on the surface of silver nanowires, i.e., droplets of malignant and healthy cells tend to assume a different shape when displayed on the nanowires. The latter instead reflects a proper chemical property of the biological samples, i.e. malignant cells might interact differently when analyzed via Raman Spectroscopy. Interestingly, these two features can be appreciated when even attempting to distinguish two different types of the same tumor; so different malignant cells might displace

differently on the substrate and show different biological characteristics with respect to the group CH. Different from the other two methods, the analysis made via PCA revealed that the interaction between the samples and the silver contained in the substrate is a relevant feature. However, this characteristic must be associated with the fact that spectra belonging to different classes must differ from a constant factor. Hence, these results motivate a deeper study to classify healthy and malignant cells by using the information arising from either LW or HW region only. Howsoever, the IW region remains still a valid field for exploring how the information arising from the methylation of DNA can be exploited to distinguish different types of malignant cells belonging to the same class of tumors. Indeed, when dealing with the binary problem A375-SK-MEL-28, the employment of saliency maps revealed as relevant some information based on the proper characteristics of skin malignant cells, e.g., the C——C stretching of amino acids proline and valine, and the bounds C═══O of lipidic structures. Based on these results, the analysis of the Experimental Setting 2 shall aim to provide a method to make a cancer diagnosis by using the Raman Spectra in either LW or HR regions only. At the same time, we shall focus on the IW region to see which characteristics related to the methylation of DNA can be exploited to accurately discern different types of the same class of malignant cells.

2.6.2 Interpretation for Experimental Setting 2: LW region

The results showed with the data acquired within Experimental Setting 1, revealed that the interaction in the LW region between the samples and the nanowires substrate can support the predictive performance of all three methods considered. The values of the mean importance for the LRP methods are reported in figure 2.10. For both cases HaCat-HT29 and HaCat-CaCO the band 180-270 cm^{-1} are relevant with mean importance larger than 0.5; see figure d and e. We recall that this spectral band includes the interaction between the samples and the metallic surface of the substrate. Anyway, all the cases considered show that the relevancy value of the spectral band 480-548 cm^{-1} is higher than 0.5; this spectral region includes the interaction between the sample and the silicon contained in the nanowires substrate. Thus, we conclude that the approach based on the average of some specific sub-spectral domain can not only distinguish healthy and tumor cells with a high level of accuracy, but the predictions are mainly supported by a piece of information extracted from specific sub-domains related to the interaction that samples have with the nanowires.

In opposition to what showed the permutation feature importance with the LRP, the visualization of the PCA components (see figure 2.11) highlights that the peak occurring at 230 cm^{-1} in the Raman Spectra (i.e. the interaction between the samples and the metallic structures of the nanowires) is the most salient feature. For all cases considered, the first PCA component turns out to be the most salient one, containing more than 90& of the explained variance (see figure 2.12). This component shows a pattern with a broad peak around 230 cm^{-1}; at the right of this region, the pattern continues with a flat line constant line close to 0. Similarly to Experimental Setting 1, the PCA method captures either one peculiar property of samples due to their interaction with the nanowires or a scale factor intercurring between spectra belonging to different classes of data.

The visualization of the Saliency Maps of the 1-D CNN models (see figure 2.13) shows that the interaction between the samples and the silicon contained in the substrate is a relevant feature for all binary problems considered. More specifically, for all four problems involving the classification of healthy and tumor cells, one can see that the saliency maps show a broad peak in a region centered at 500 cm $^{-1}$. An exception to this is represented by the case HaCaT-HT29 (see figure 2.13d)

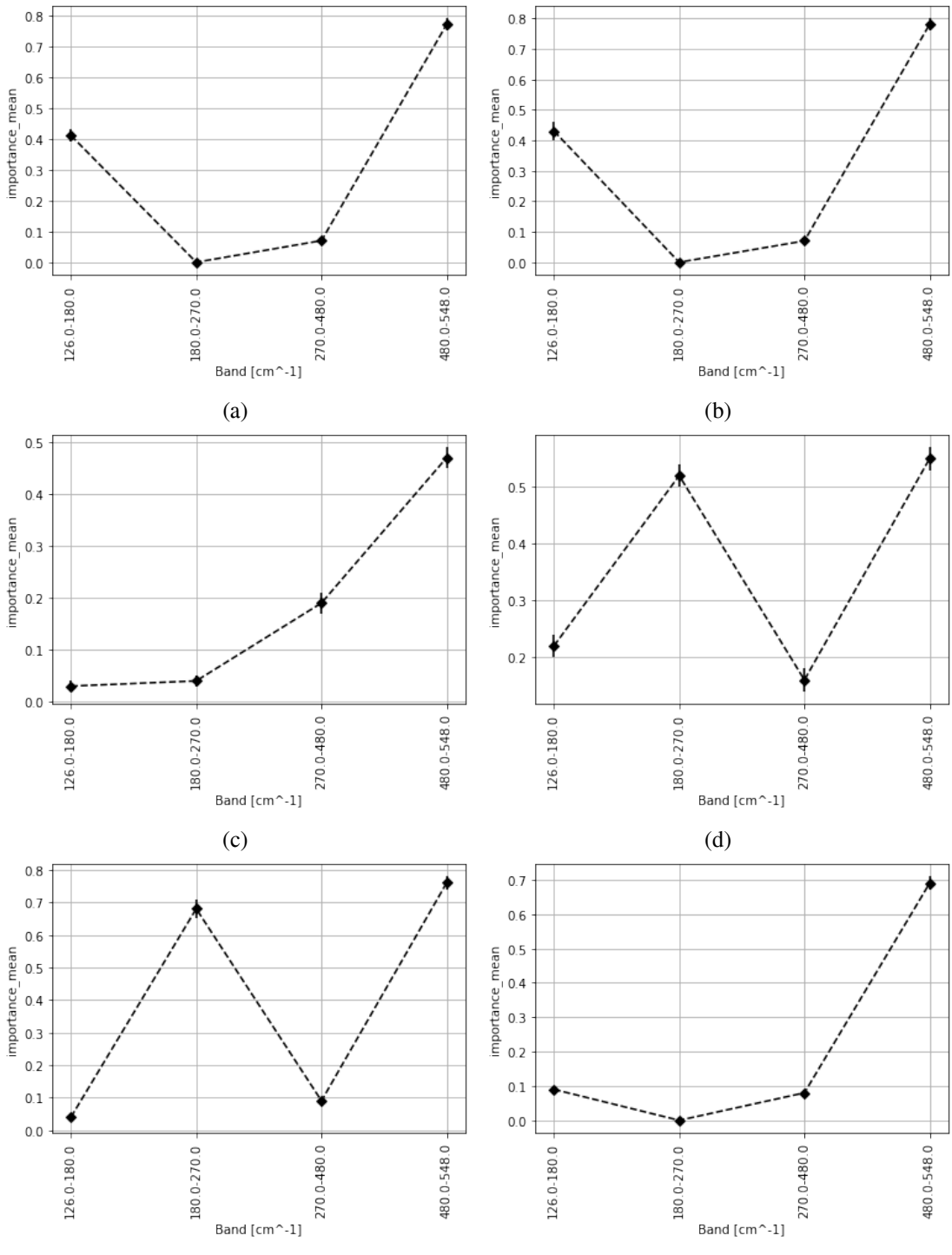


Figure 2.10: Experimental Setting 2; LW region. Relevance Importance of LRP for (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, (c) A375-SK-MEL-28, (d) HaCaT-HT29, (e) HaCaT-CaCO, and (f) HT-29-CaCO.

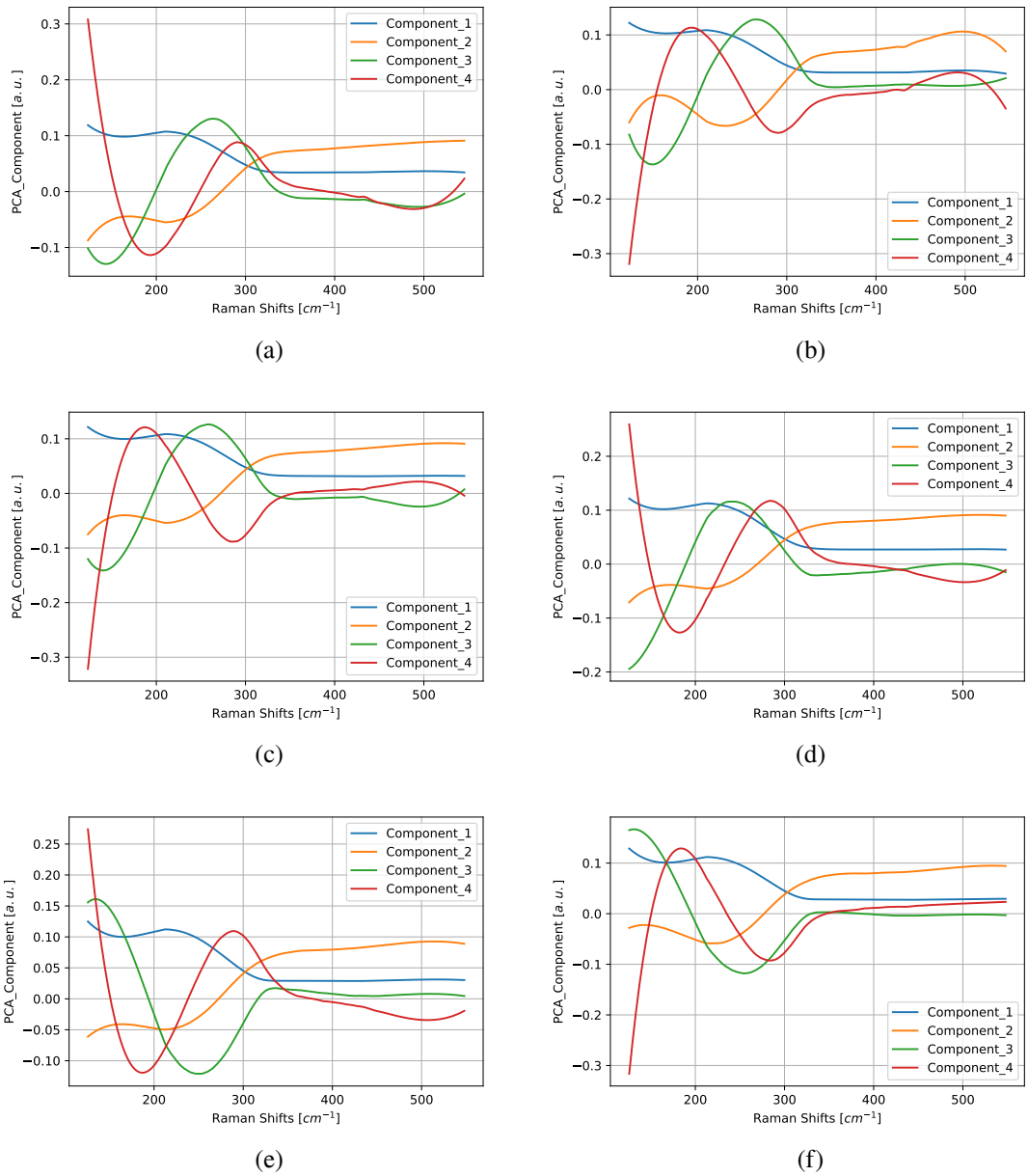


Figure 2.11: Experimental Setting 2; LW region. The first four components of the PCA algorithm for cases (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, (c) A375-SK-MEL-28, (d) HaCaT-HT29, (e) HaCaT-CaCO, and (f) HT-29-CaCO. The first component of PCA is depicted in blue, the second one in orange, the third one in green, and the fourth one in red.

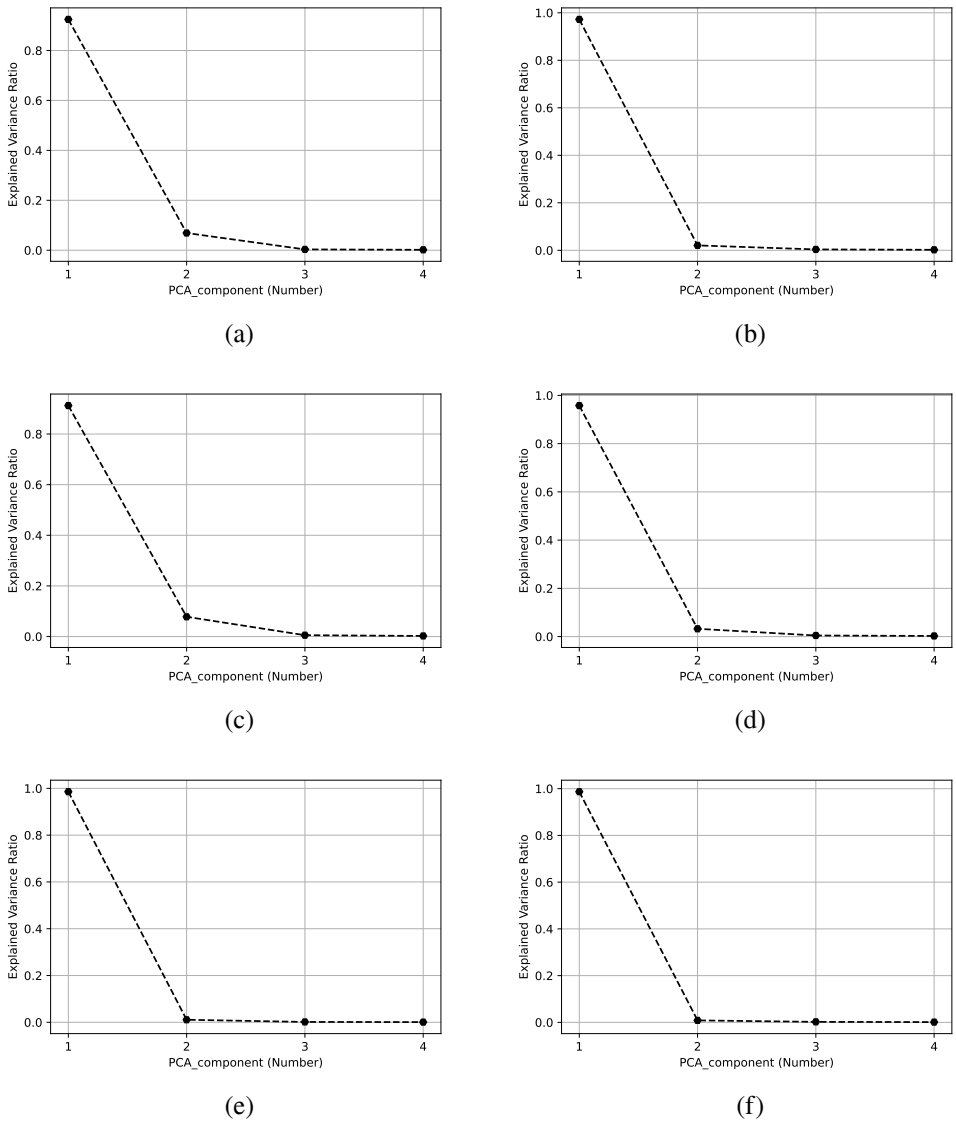
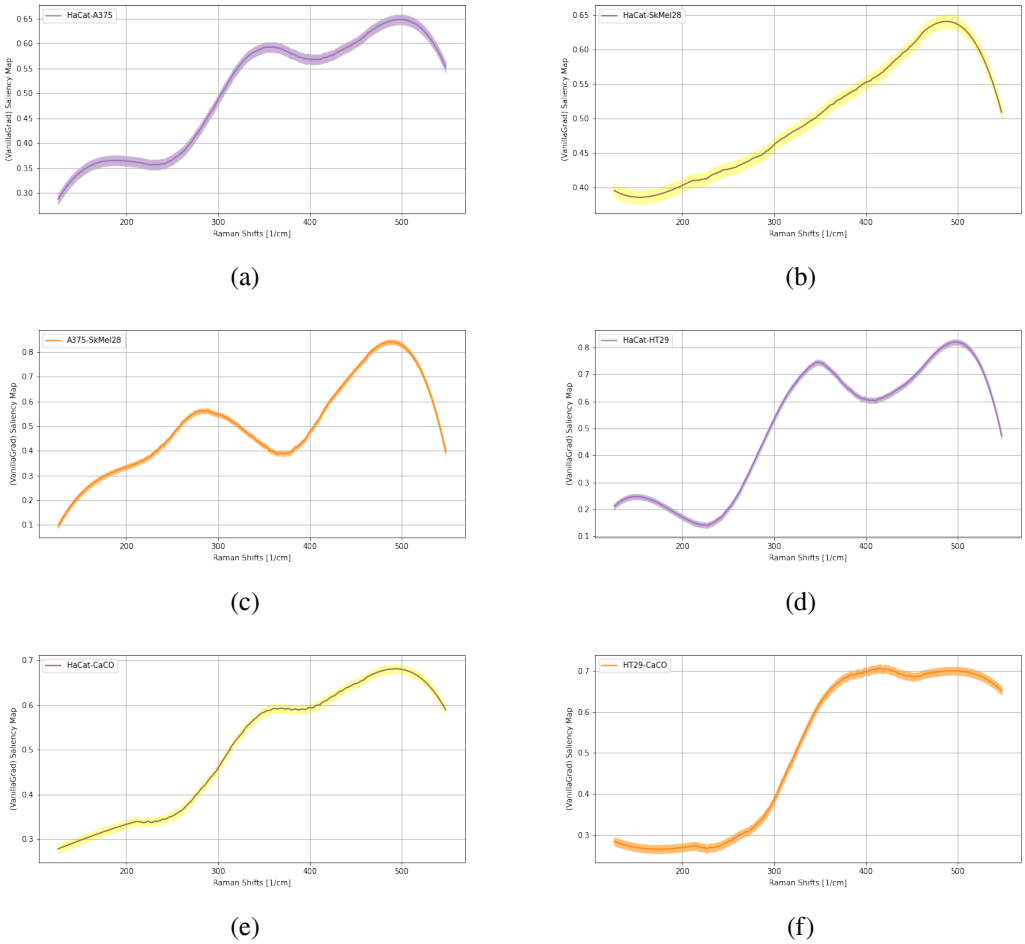


Figure 2.12: Experimental Setting 2; LW region. The explained variance ratio of the first four components of the PCA algorithm for cases (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, (c) A375-SK-MEL-28, (d) HaCaT-HT29, (e) HaCaT-CaCO, and (f) HT-29-CaCO.



(a) (b)

(c) (d)

(e) (f)

Figure 2.13: Experimental Setting 2; LW region. Saliency Map via VG algorithm for cases (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, (c) A375-SK-MEL-28, (d) HaCaT-HT29, (e) HaCaT-CaCO, and (f) HT-29-CaCO.

where is also shown a salient peak centered at 350 cm $^{-1}$. We recall that the region 300-400 cm $^{-1}$ is a flat and silent region for all Raman Spectra considered, i.e. neither peaks nor other similar patterns can be detected and led to specific physicochemical properties of samples. In this region, the saliency maps, in this case, are evincing a difference in the magnitude of the Raman Spectra. When considering the case A375-SK-MEL-28, the saliency map highlights the same feature shown above, i.e. a peak centered a 500 cm $^{-1}$; see figure 2.13c The visualization of the saliency map of case HT29-CaCO shows, instead, a flat saliency region at the right of 350 cm^{-1}; similarly to the case HaCaT-HT29, the 1-D CNN model is not only capturing the characteristics related to the interaction between the samples and the silicon of the substrate, but also a difference in the magnitude of the spectra.

2.6.3 Interpretation for Experimental Setting 2: IW region

The values of the mean importance for the LRP methods are reported in figure 2.14. We first comment on the cases of study involving melanoma data. In each one of the three problems, one of the most relevant characteristics is represented by the spectral band 1120-1450 cm^{-1}. This spectral sub-band includes several physicochemical features such as conformal changes in guanine-cytosine and adenine-thymine oligonucleotides induced by aminooxy (both these molecules should show different spectral features at 1335 cm^{-1}) (Talari et al., 2015), and the first (of two) so-called "cathedral peaks". Those peaks may be originated from the formation of a carbonaceous layer due to the photodecomposition of the DNA at the silver surface upon laser irradiation (Furtak, 1983). Case HaCaT-A375 (see figure 2.14 a)also shows an important feature at band 860-1120 cm^{-1}; where one finds the O-P-O stretching mode of phosphodiester at 920 cm^{-1}. Note that such a feature emerged in the comments on the saliency maps of data acquired within Experimental Setting 1. Interestingly, both cases HaCaT-SK-MEL-28 and A375-SK-MEL-28 (see figure 2.14 b and figure 2.14 c, respectively) are characterized by a feature located in an influent region as 1704-2001 cm^{-1}. In that region, the samples interact with the substrate, but the spectrum is not detecting there any physicochemical properties.

The employment of the permutation feature importance technique on the LRP model fed with colon data shows that all binary problems considered show different relevant features, The prediction formulated for the case HaCaT-HT29 (see figure 2.14 d) turns out to be supported by the information contained in both the spectral sub bans 860-1120 cm^{-1} and 1600-1704 cm^{-1}; the first is the related to the O-P-O stretching mode of phosphodiester, while the latter specifies the protein contents of biological samples (amide I) (Talari et al., 2015). Case HaCaT-CaCO (see figure 2.14 e) is mainly solved by the interaction of three specific features; i.e., the spectral band at 700-760 cm^{-1}, containing both the ring breathing modes of DNA bases and the stretch of bounds C —— N; the spectral sub-band 760-860 cm^{-1} containing the O-P-O stretching in DNA; and the sub-band 1450-1650 cm^{-1}, containing the second "cathedral peak" and specifications of protein contents (CH_2 bend at 1450 cm^{-1}) and lipidic contents (structures C ═ C at 1656 cm^{-1}) of biological samples (Talari et al., 2015). Case HT29-CaCO (see figure 2.14 f) shows the same relevant characteristics as case HaCaT-CaCO, but with a feature more which is the sub-band 640-700 cm^{-1}, i.e. the ring breathing modes in the DNA bases (Talari et al., 2015).

The visualization of the PCA components in the IW region (see figure 2.15) shows that, for all cases, the first component is a flat signal, while the other PCA components are more structured and more pattern detailed. The visualization of the explained variance score (see figure 2.16) shows

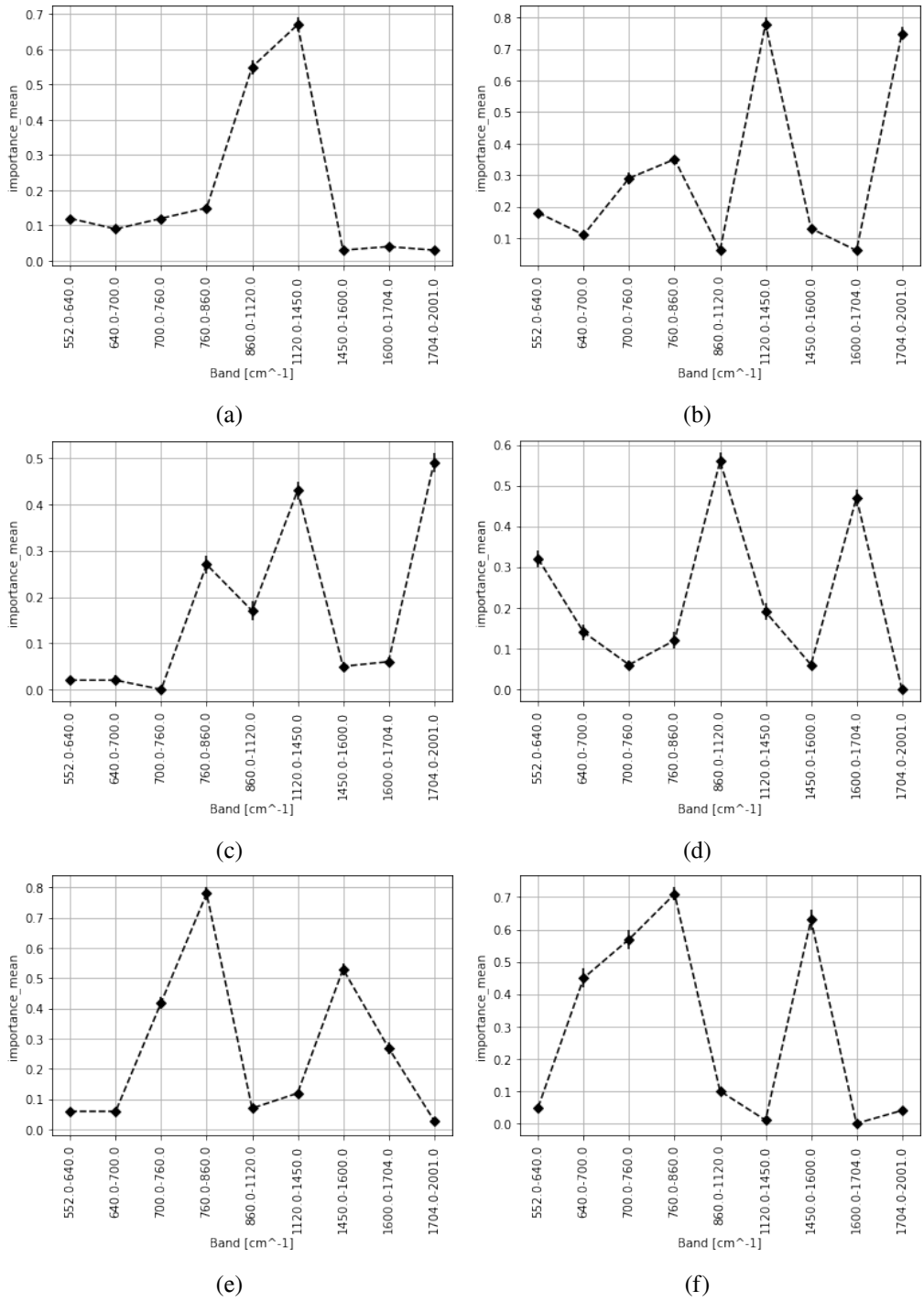


Figure 2.14: Experimental Setting 2; IW region. Relevance Importance of LRP for (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, (c) A375-SK-MEL-28, (d) HaCaT-HT29 (e) HaCaT-CaCO (f) HT-29-CaCO.

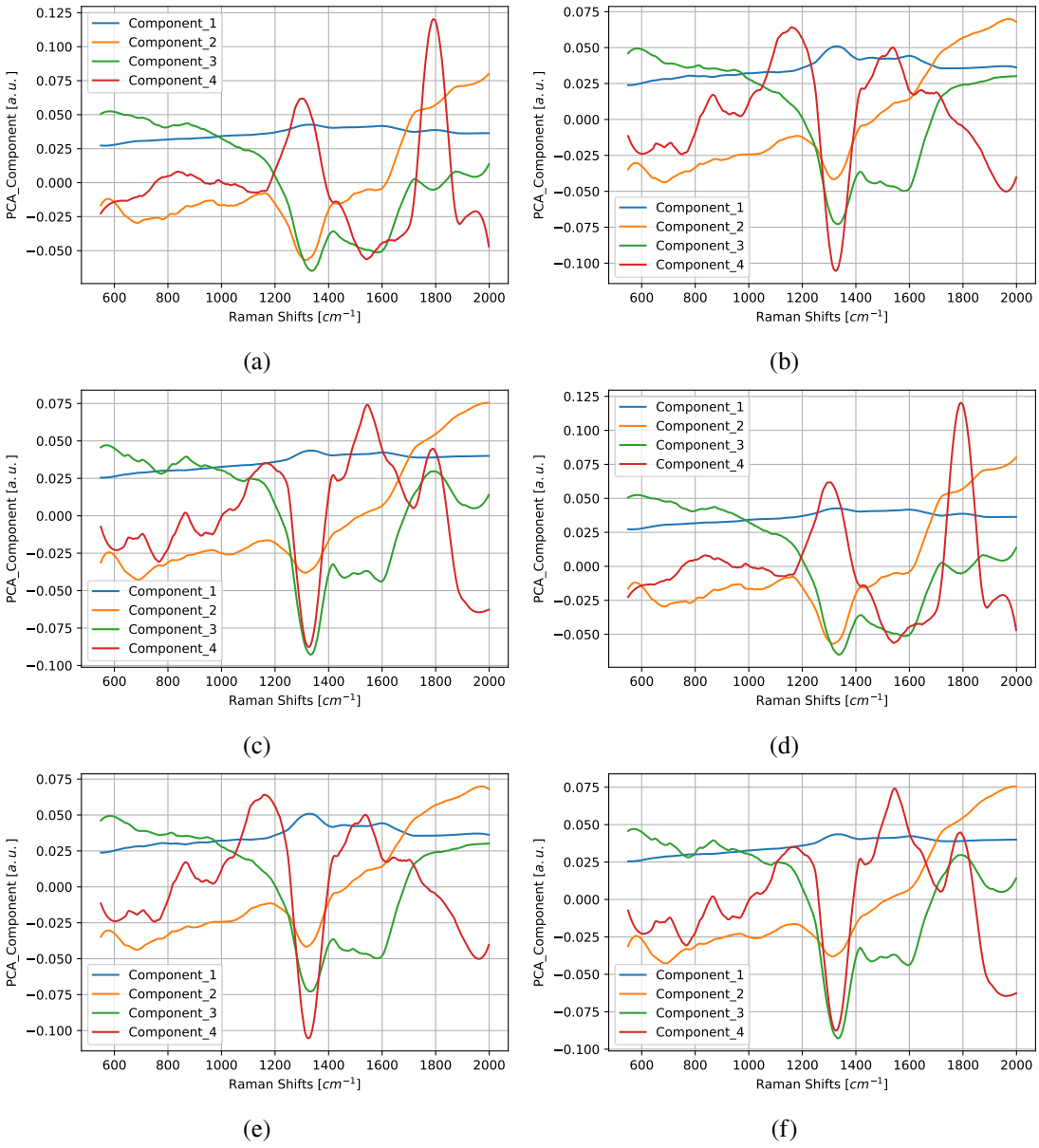


Figure 2.15: Experimental Setting 2; IW region. The first four components of the PCA algorithm for cases (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, (c) A375-SK-MEL-28, (d) HaCaT-HT29, (e) HaCaT-CaCO, and (f) HT-29-CaCO. The first component of PCA is depicted in blue, the second one in orange, the third one in green, and the fourth one in red.

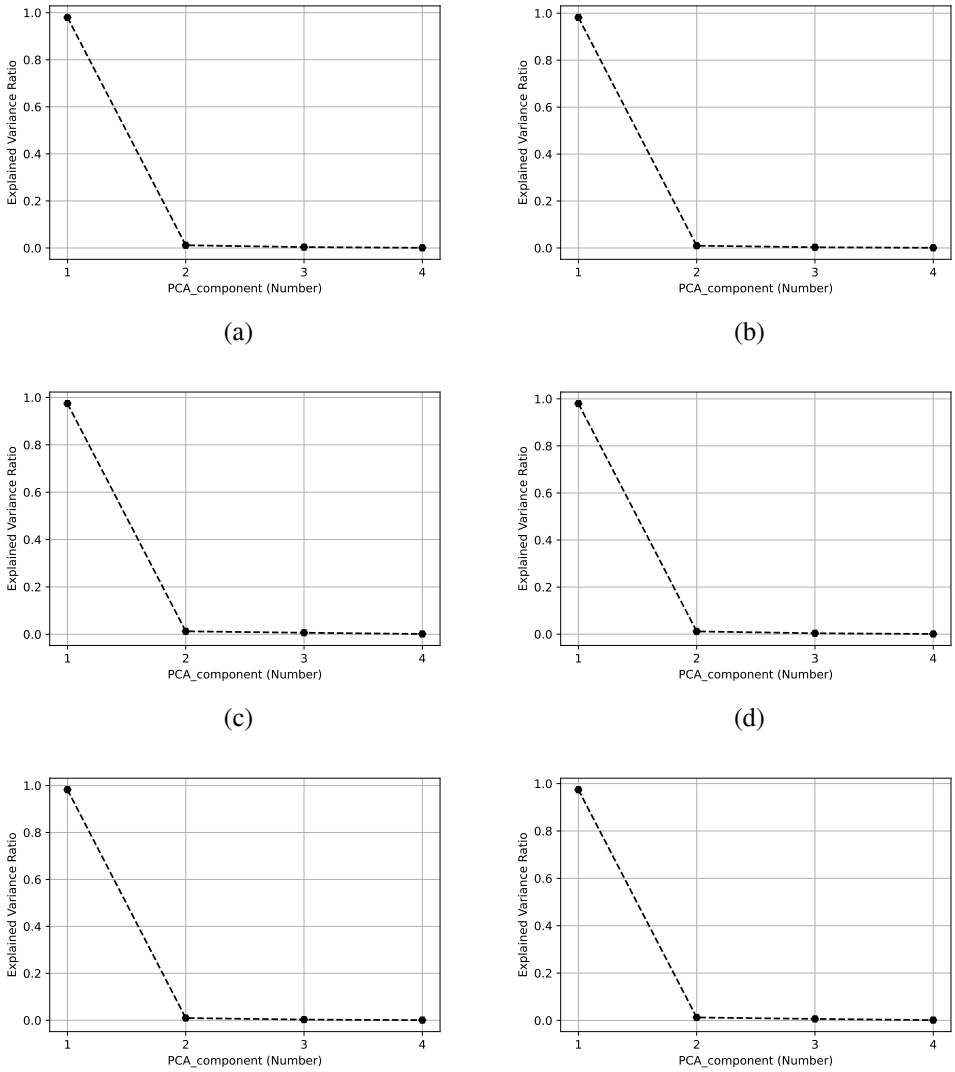


Figure 2.16: Experimental Setting 2; IW region. The explained variance ratio of the first four components of the PCA algorithm for cases (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, (c) A375-SK-MEL-28, (d) HaCaT-HT29, (e) HaCaT-CaCO, and (f) HT-29-CaCO.

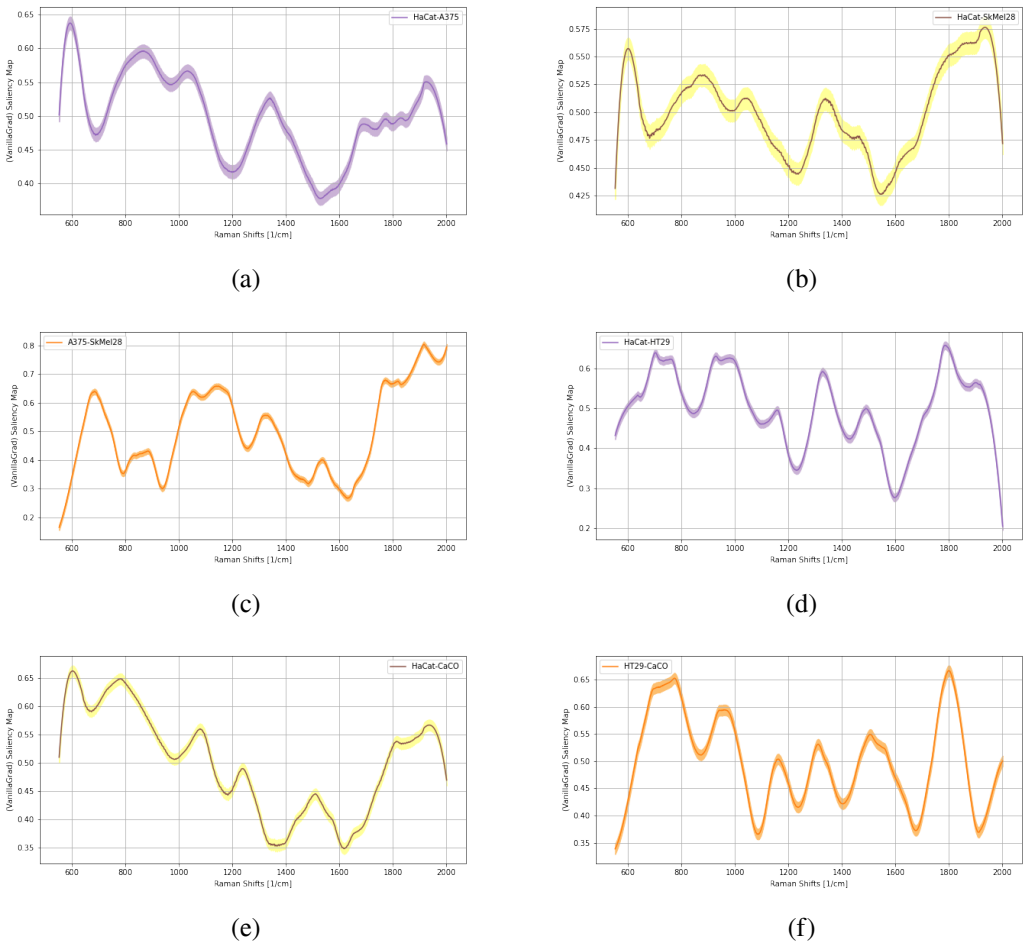


Figure 2.17: Experimental Setting 2; IW region. Saliency Map via VG algorithm for cases (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, (c) A375-SK-MEL-28, (d) HaCaT-HT29, (e) HaCaT-CaCO, and (f) HT-29-CaCO.

that the first PCA component explains more than 95% of the empirical variance. Similarly to Experimental Setting 1, the first PCA component is the most salient, and that one highlights a difference in the magnitude of Raman Spectra.

The visualization of the saliency maps in the IW region (see 2.17) shows that each binary problem is solved by looking at different saliency regions. The saliency map of case HaCaT-A375 (see 2.17 a) offers only one salient region, i.e. a narrow sub-band centered at 600 cm^{-1}. This spectral sub-region contains the twisting of C —— C twisting of proteins (613 cm^{-1}). The saliency map of case A375-SK-MEL-28 (see 2.17 c), instead, shows saliency regions in a narrow band centered at 700 cm^{-1} (probably connected with the ring breathing of DNA bases), another flat salient region at 1000-1200 cm^{-1}, and at the right of 1700 cm^{-1}. When considering case HaCaT-HT29 (see 2.17 d), the saliency maps highlight saliency regions at 700-800cm $^{-1}$, 1000-1100 cm^{-1}, and a peak region centered at 1800 cm^{-1}. When considering case HaCaT-CaCO (see 2.17 d), the saliency map is composed of two saliency regions located in narrow areas centered at 600 cm^{-1} and 800 cm^{-1}. At

last, the saliency map of case HT29-CaCO (see 2.17 f) shows only two saliency regions such as a flat area at 750-800 cm^{-1} and a narrow peak at 1800 cm^{-1}. All regions mentioned here are connected to some precise physicochemical features that we have already described with the interpretation of the LRP activity recognition. However, it is important to stress, that both the 1-D CNN and the LRP method might be highly accurate in solving a binary problem, but capturing different properties of the spectrum. This might arise from the fact that LRP only loos at differences in the mean value of different fixed subdomains, while the 1-D CNN have access to the complete information contained in the spectrum.

2.6.4 Interpretation for Experimental Setting 2: HW region

The values of the mean importance for the LRP methods in the HW region are reported in figure 2.18. The majority of cases report only one relevant spectral sub-band, i.e., 2701-3200 cm^{-1}. This band contains one aforementioned physicochemical property of biological samples, i.e. the vibrational modes of the group CH. Both cases HaCaT-SK-MEL-28 and A375-SK-MEL-28 (see figure 2.18 b and figure 2.18 c, respectively) show a unique salient sub-band located at 3200-3900 cm^{-1}. In this case, the band 3200-3900 cm^{-1} does not represent a known physicochemical property of biological samples; the LRP model is basing its predictive performance on an overt difference in the magnitude of the Raman Spectra.

Similarly to the cases analyzed above, the visualization of the PCA components in the HR region reveals that the first component explains almost 99% of the empirical variance (see figure 2.20) and is represented by a flat signal (see figure 2.19). Thus, the classification proposed by the PCA method is mainly based on an overt difference in the magnitude of the Raman Spectra. Note that the patterns related to the presence of the group CH are represented by the third and fourth PCA components, i.e. patterns possessing less than 1% of the empirical variance.

Last, the saliency maps of the HR region are shown in figure 2.21. Both cases HaCaT-A375 and HT-29-CaCO (see figure 2.21 a and figure 2.21 f, respectively) show a narrow saliency region centered at 2930 cm^{-1}; other saliency regions are also located at right the right 2930 cm^{-1}. Similarly to the LRP, in both cases HaCaT-SK-MEL-28 and A375-SK-MEL-28 (see figure 2.21 b and figure 2.21 c, respectively) do not show any saliency region at 2930 cm^{-1}, but at the right of this wavenumber. At last, case CaCO-HaCaT (see figure 2.21 e) shows a drop in correspondence of 2930 cm^{-1}, while a high-valued saliency map covers a large subdomain at 3000-3200 cm^{-1}.

2.7 Summary of the findings and future developments

The analysis provided by the LRA method within Experimental Setting 1 showed that the 1-feature representation via the mean of the Raman spectra is sufficient per se to distinguish human skin cancer and healthy cells. This fact highlights fundamental characteristics of data we dealt with, that is the Raman spectrum either healthy or a tumor cell sample shows a difference of scale; cancer cells tend to release more energy along the whole spectral domain when involved in light scattering processes. In opposition to this, when considering two different types of melanoma, the global mean value cannot be used to solve the classification task. Likewise, the LRA method revealed the same results within Experimental Setting 2; both melanoma and colon-rectal Raman spectra lead to the same results.

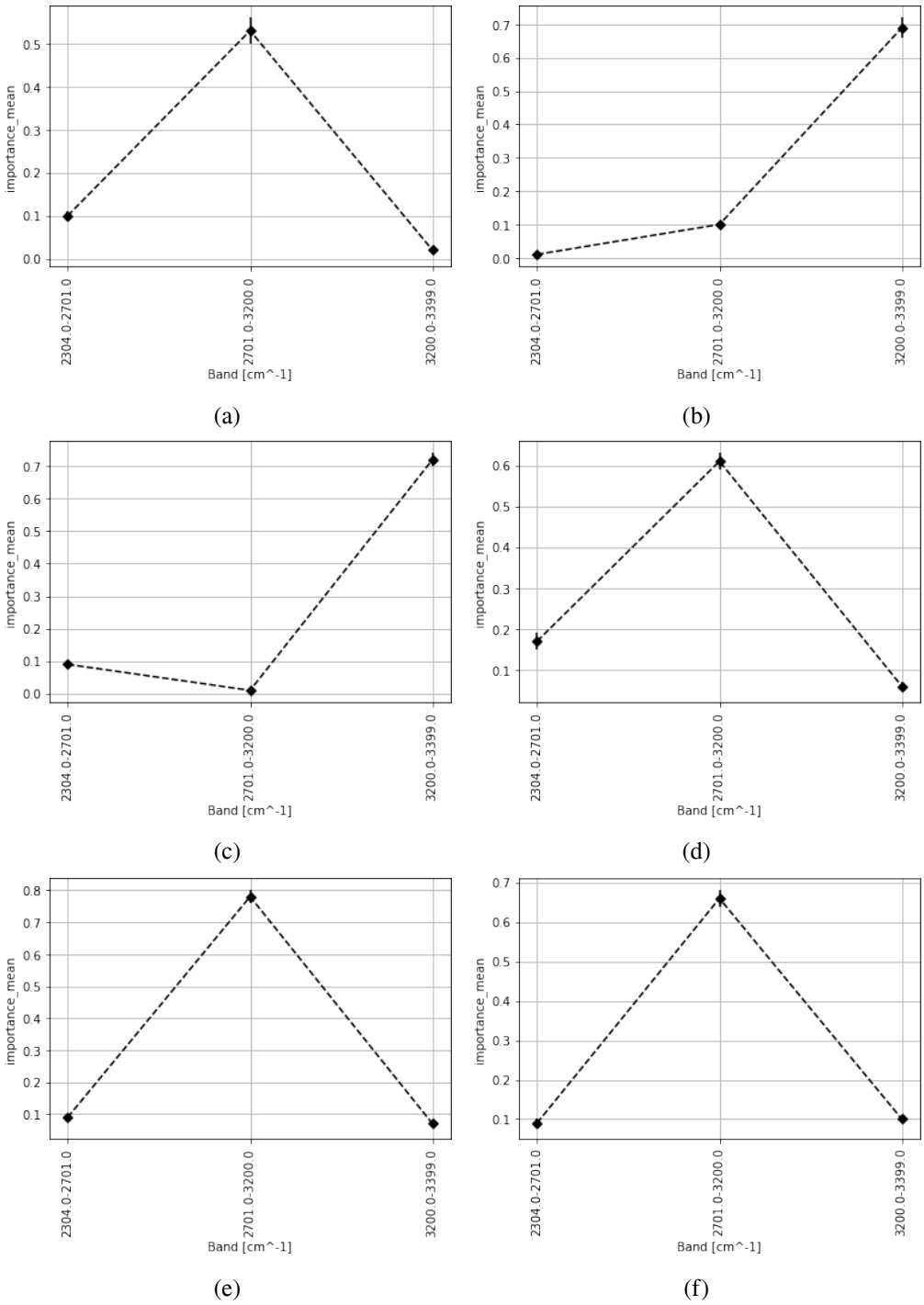


Figure 2.18: Experimental Setting 3; HW region. Relevance Importance of LRP for (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, (c) A375-SK-MEL-28, (d) HaCaT-HT29 (e) HaCaT-CaCO (f) HT-29-CaCO.

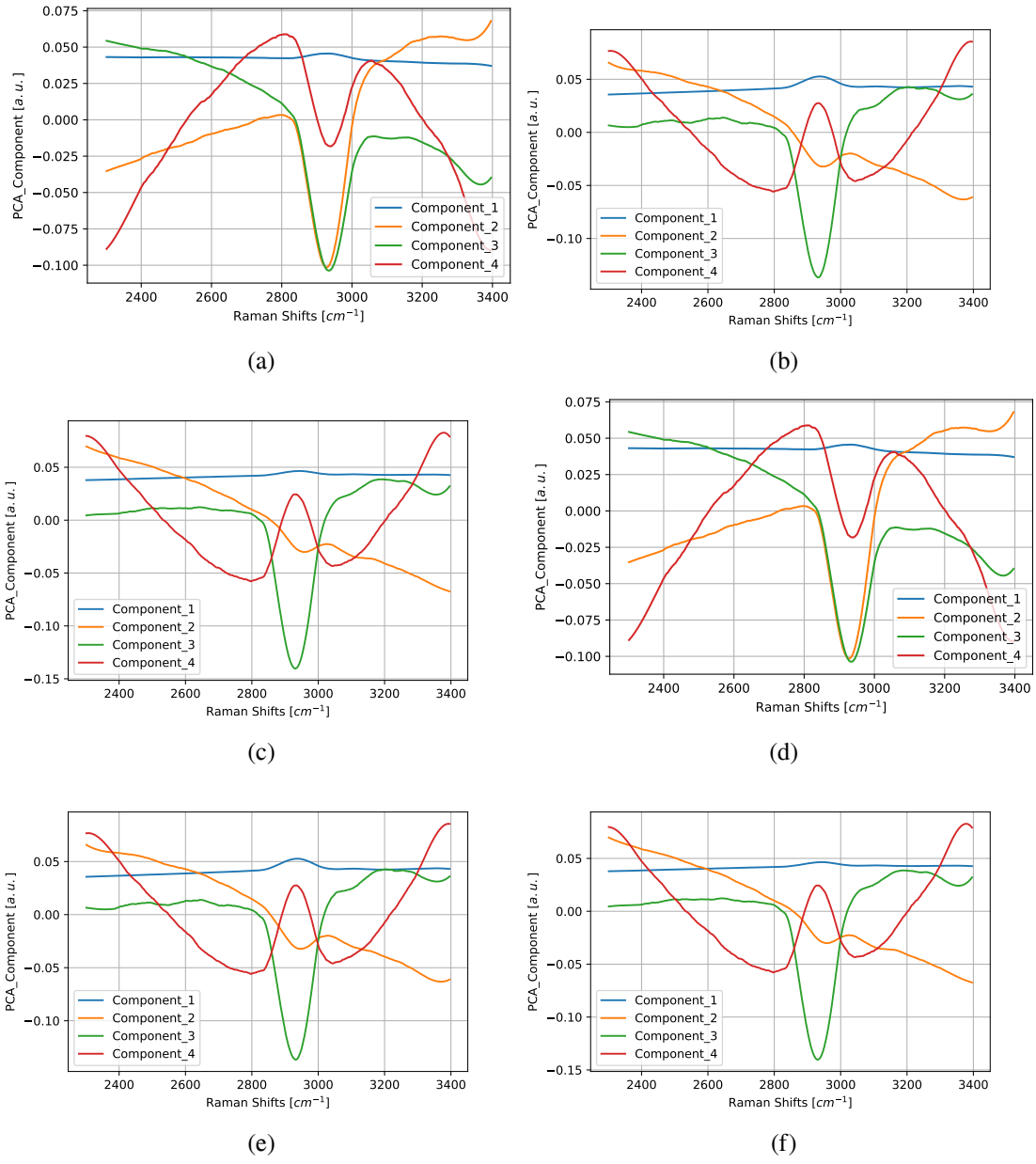


Figure 2.19: Experimental Setting 2; HW region. The first four components of the PCA algorithm for cases (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, (c) A375-SK-MEL-28, (d) HaCaT-HT29, (e) HaCaT-CaCO, and (f) HT-29-CaCO. The first component of PCA is depicted in blue, the second one in orange, the third one in green, and the fourth one in red.

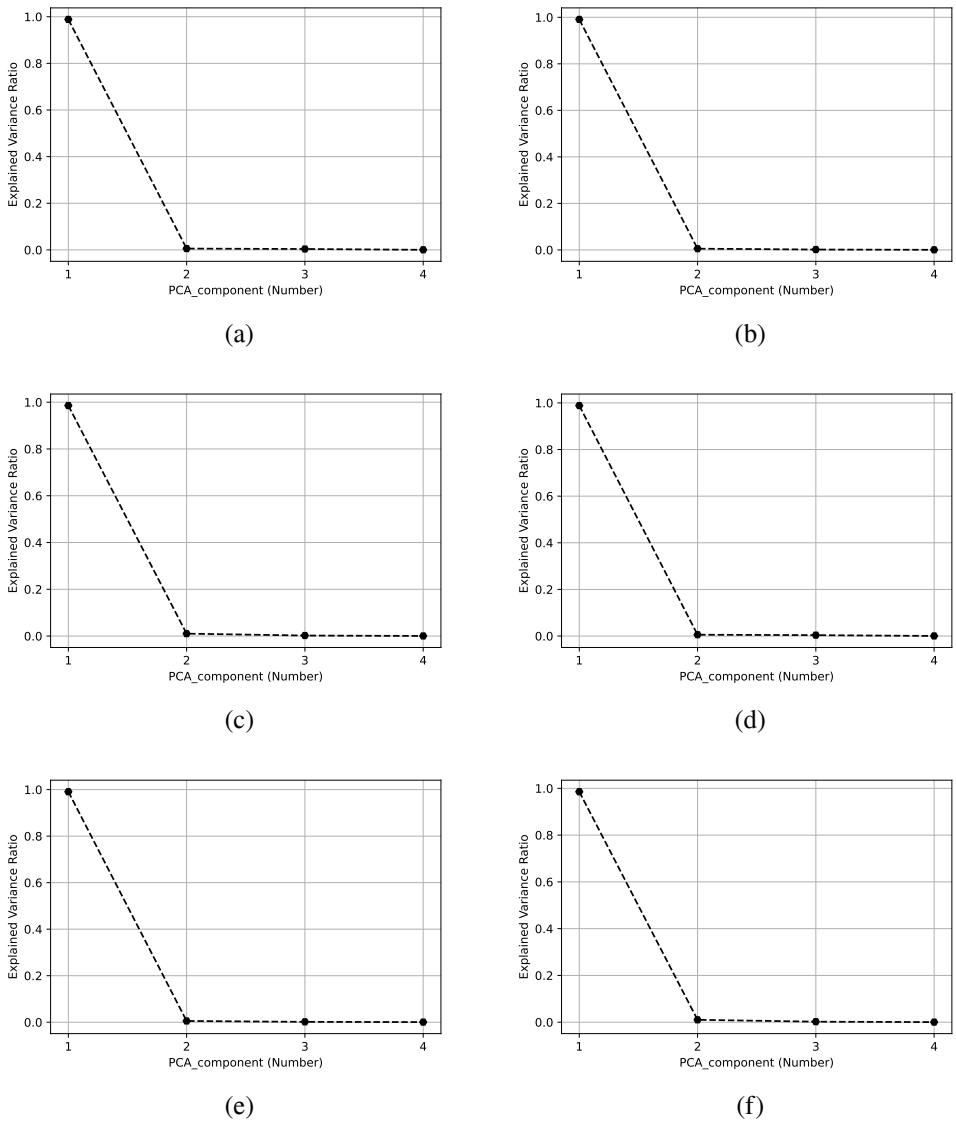


Figure 2.20: Experimental Setting 2; LW region. The explained variance ratio of the first four components of the PCA algorithm for cases (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, (c) A375-SK-MEL-28, (d) HaCaT-HT29, (e) HaCaT-CaCO, and (f) HT-29-CaCO.

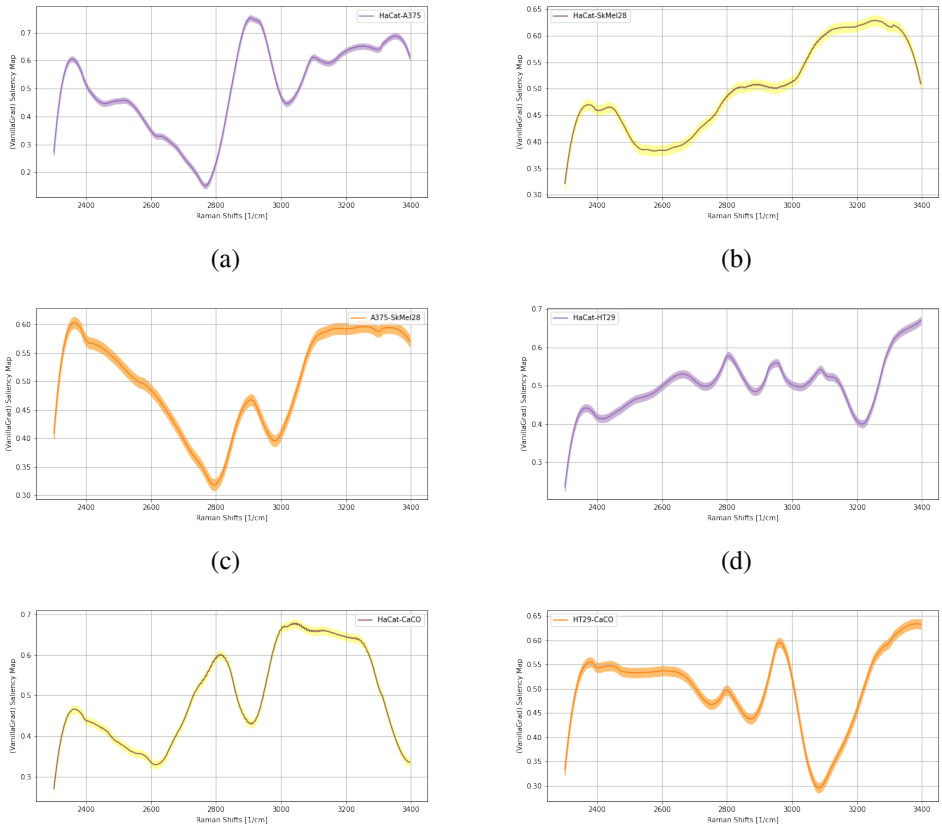


Figure 2.21: Experimental Setting 2; HW region. Saliency Map via VG algorithm for cases (a) HaCaT-A375, (b) HaCaT-SK-MEL-28, (c) A375-SK-MEL-28, (d) HaCaT-HT29, (e) HaCaT-CaCO, and (f) HT-29-CaCO.

The difference of scale in both healthy and tumor Raman spectra is the main characteristic captured by the PCA model as well. For data within both Experimental Setting 1 and Experimental Setting 2, we showed that more than 95% of the explained variance of Raman Spectra was captured by the first component which is usually represented by a flat pattern. Although the PCA model presents higher predictive performances than the LRA model, we showed that the further PCA components that are connected to the physicochemical properties of DNA all represent just 5% of the explained variance. So, basically, a possible alteration of the DNA of tumor cells can be visualized by the PCA model only if a difference in the scale of the Raman Spectra is present.

When considering the LRP model, the interpretation of the input features that support the high predictive power of this method highlighted a few distinctive physicochemical properties of cell samples. Similar results have also been shown when applying the 1-D CNN; these results are summarized in Table2.7 (Experimental Setting 1) and Table2.8 (Experimental Setting 2). Considering both Experimental Setting 1 and Experimental Setting 2, one can observe that four main physicochemical properties are captured by both models, that is the metabolism of proline in cancer cells, an abnormal change in the membrane lipids of tumor cells, an alternation of the CH group in the DNA and a different tendency of tumor cells to displace differently on the nanowires substrate. In particular, the last one is highlighted by the presence of a peak around 512 cm^{-1} in the Raman spectrum; this peak is a characteristic of silicon atoms. When acquiring the Raman Spectrum of a tumor cell, the presence of some exclusive property of the nanowires substrate indicates that the tumor sample might not fully adhere to it; this points out a salient difference in the structure of both melanoma and colon-rectal cells with respect to healthy cells.

It is important to stress that the predictions of both the LRP and 1-D CNN model are supported by different characteristics of Raman Spectra; that is LRP model classifies the difference of scale of the Raman spectra in a fixed number of non-overlapping subdomains, while the 1-D CNN model makes its predictions after providing a particular pre-processing of data that is based on the capture of specific patterns located at some precise sub-domains of the Raman spectra. This difference represents the advantage of using in this context a 1-D CNN model with respect to traditional ML models. 1-D CNN revealed to be accurate and highly performative when data have either a high definition (Experimental setting 2) or a poorer quality (Experimental setting 1). Especially in the latter case, the traditional ML models could not show the prediction skill as accurately as the 1-D CNN model. In this work, we showed that 1-D CNN model took advantage of its pattern recognition activity to distinguish the structures of Raman Spectra that are directly related to some cancer alteration of DNA; this was always possible when dealing with different data showing different resolutions and definitions. This fact together with the possibility of interpreting the prediction skill of CNN models via Saliency map, offers the possibility of boosting up the analysis of Nanomedicine data. The constantly evolving field of Nanomedicine provides the acquisition of an increasing amount of different kinds of informative healthcare data (Serov and Vinogradov, 2022; Dimitri and Talamo, 2018); and ANN, in particular CNN, found here an application to improve a nanomaterial-based approach of investigation for cancer diagnosis.

Raman Shift	HaCat-A375	HaCat-SK-MEL-28	A375-SK-MEL-28
$230\ cm^{-1}$	NYN	NYN	NYN
$512 - 514\ cm^{-1}$	NNY	NNY	NNY
$600 - 800\ cm^{-1}$	NNN	NNN	NNN
$614\ cm^{-1}$	NNN	NNN	NNN
$678\ cm^{-1}$	NNN	NNN	NNN
$735\ cm^{-1}$	NNN	NNN	NNN
$788\ cm^{-1}$	NNN	NNN	NNN
$800 - 1200\ cm^{-1}$	YNN	YNN	YNY
$920\ cm^{-1}$	YNN	YNN	YNY
$1177 - 1185\ cm^{-1}$	YNY	YNN	YNN
$1200\ cm^{-1}$	YNY	YNN	YNN
$1320\ cm^{-1}$	YNY	YNN	YNN
$1335\ cm^{-1}$	YNY	YNN	YNN
$1421\ cm^{-1}$	NNY	NNN	NNN
$1520\ cm^{-1}$	NNY	NNN	NNN
$1575\ cm^{-1}$	NNY	NNN	NNN
$1608 - 1610\ cm^{-1}$	NNY	NNN	NNN
$1634\ cm^{-1}$	NNN	NNN	NNN
$1650\ cm^{-1}$	NNN	NNN	NNN
$1652 - 1653\ cm^{-1}$	NNY	NNN	NNN
$1674\ cm^{-1}$	NNN	NNN	NNN
$1716 - 41\ cm^{-1}$	NNN	YNN	NNY
$1750\ cm^{-1}$	NNN	YNN	NNY
$2700 - 3100\ cm^{-1}$	YNY	YYN	NNY
$2928\ cm^{-1}$	YNY	YNY	NNY
$2929 - 2940\ cm^{-1}$	YNY	YNY	NNY
$2956\ cm^{-1}$	YNN	YNY	NNY
$> 3000\ cm^{-1}$	YNN	YNN	NNN

Table 2.7: Summary of the salients Raman shifts that support the prediction of the predicting models (Experimental Setting 1). The connection between Raman shifts and the physicochemical properties of samples is reported in section 2.3. Blue: LRP; Red: PCA; Green: 1D-CNN. In each cell, the flag Y (Y= yes) indicates that the model base its activity on a precise Raman shift; the opposite situation is described by the flag N (N= No).

Raman Shift	HaCat-A375	HaCat-SK-MEL-28	A375-SK-MEL-28	HaCat-HT29	HaCat-CaCo	HT29-CaCo
230 cm^{-1}	NYN	NYN	NYN	YYY	YYN	NYN
$512 - 514$ cm^{-1}	YNY	YNY	YNY	YNN	YNY	YNY
$600 - 800$ cm^{-1}	NNN	NNN	NNN	NNN	NNY	NNY
614 cm^{-1}	NNN	NNN	NNN	NNN	NNY	NNY
678 cm^{-1}	NNN	NNN	NNN	NNY	NNN	NNN
735 cm^{-1}	NNN	NNN	NNN	YNY	YNN	YNN
788 cm^{-1}	NNN	NNN	NNN	YNY	YNN	YNN
$800 - 1200$ cm^{-1}	YNN	YNN	YNN	YNN	YNN	YNN
920 cm^{-1}	YNN	YNN	YNN	YNN	YNN	YNN
$1177 - 1185$ cm^{-1}	YNY	YNN	YNY	YNY	YNN	YNN
1200 cm^{-1}	YNY	YNN	NNY	YNY	NNN	NNN
1320 cm^{-1}	YNY	YNN	YNY	NNY	NNN	NNN
1335 cm^{-1}	YNY	YNN	YNY	NNY	NNN	NNN
1421 cm^{-1}	YNY	YNN	YNY	NNY	YNN	YNN
1520 cm^{-1}	NNY	YNN	NNY	NNY	YNN	YNN
1575 cm^{-1}	NNY	NNN	NNY	NNY	YNN	YNN
$1608 - 1610$ cm^{-1}	NNY	YNN	NNY	NNY	YNN	YNN
1634 cm^{-1}	NNY	YNN	NNY	NNY	YNN	YNN
1650 cm^{-1}	NNY	YNN	NNY	NNY	TNN	YNN
$1652 - 1653$ cm^{-1}	NNY	YNN	NNY	NNY	YNN	NNN
1674 cm^{-1}	NNY	YNN	NNY	NNY	NNN	NNN
$1716 - 41$ cm^{-1}	NNY	YNN	NNY	NNY	NNN	NNN
1750 cm^{-1}	NNY	YNN	NNY	NNY	YNN	NNN
$2700 - 3100$ cm^{-1}	YNY	NNN	NNN	YNN	YNY	YNY
2928 cm^{-1}	YNY	NNN	NNN	YNN	YNN	YNY
$2929 - 2940$ cm^{-1}	YNY	NNN	NNN	YNN	YNN	YNN
2956 cm^{-1}	YNY	NNN	NNN	YNN	YNY	YNY
> 3000 cm^{-1}	NNN	YNY	YNY	NNN	NNN	NNN

Table 2.8: Summary of the salients Raman shifts supporting the predictions (Experimental Setting 2). The connections between Raman shifts and the physicochemical properties are in section 2.3. Blue: LRP; Red: PCA; Green: 1D-CNN. In each cell, the flag Y (Y= yes) indicates that the Raman shift supports the predictions; the flag N (N= No) describes the opposite situation.

Chapter 3

Physics captured by 1-D CNN in El Niño prediction

三人行必有我师

孔子

Tropical Pacific can periodically be subjected to irregular variations in sea surface temperature and wind strength. Such an event is called El Niño and occurs on average once per four years, affecting the climate sensitivity over many regions on Earth. In the last decade, DL techniques, and in specific CNN (see section 1.1.3), have shown to be peculiarly accurate in ENSO predictions, even at long lead times (see section 3.1). In order to give a deeper understanding and an interpretation of this high skill of CNN, we used the ZC model (see section 3.2), which is also routinely used for ENSO prediction. Thus, we collected data from *distorted physics experiments* to determine what aspects of El Niño physics can be captured and recognized in a specific CNN-based classification method (see section 3.3), that is, we generated physical processes for different choices of the parameters ruling the equatorial wave dynamics and ocean-atmosphere feedbacks in the ZC model. In particular, saliency maps have been applied to visualize which features of the ZC simulated data contribute most in determining the prediction skill of a 1-D CNN model; we shall discuss it in section 3.5. We aimed to reveal what feature of the ENSO dynamic in ZC data is mainly captured by the 1-D CNN model. For completeness, in section 3.6, the prediction skill of 1-D CNN models has been compared with the GDN (see section 1.1.5); the latter is another DL technique that has revealed a high predictive performance in ENSO forecast (Petersik and Dijkstra, 2020).

3.1 The role of DL in ENSO forecast

ENSO is the most dominant factor in the interannual climate change in the Tropical Pacific regions. This event is characterized by an increase of a few degrees in the surface temperatures compared to the seasonal averaged values; it is usually called *El Niño*. As a counterpart, the decrease of a few degrees compared to the seasonal averaged values is the so-called *La Niña*. The anomaly in the sea surface temperature is usually represented by the *NIÑO3.4 index*, i.e., the deviation from the mean seasonal cycle over the region 170°W-120°W $\times$ 5°S- 5°N. It represents the actual state of

ENSO. December is the period of the year where El Niño events usually culminate, typically each two to seven years, with varying intensities that can consistently vary over decadal time series. The temporal patterns in ENSO can be detected via principal component analysis (Preisendorfer, 1988); at least two different types of El Niño events exist (Zhang et al., 2019), with the most significant temperature anomalies either in the eastern Pacific (Eastern Pacific or EP El Niño's) or near the dateline (Central Pacific or CP El Niño's). ENSO events can significantly affect both the climate in the Tropic Pacific and the whole climate of the globe through well-known teleconnections; skillful forecasting of up to a one-year lead time is demanded to be able to mitigate these effects (Balmaseda et al., 1995). For ENSO predictions, often the ONI is used, which is defined as the three-month running mean of the NINO3.4 index. Both statistical models and dynamic models are used for El Niño prediction (Latif et al., 1998; Chen and Cane, 2008; Barnston et al., 2012b; Saha et al., 2014; Timmermann et al., 2018; Tang et al., 2018). El Niño events, however, represent a difficult outcome to forecast because of the irregular occurrence and the slight changes occurring in the development of the vent at each time (McPhaden et al., 2015; Timmermann et al., 2018). In several ENSO prediction evaluation studies (Barnston et al., 2012a; L'Heureux et al., 2017) is shown that dynamical models can lead to more accurate results than traditional statistical models; when initialized before the boreal spring, most models even perform much worse than when initialized in summer. The latter notion has been indicated by the spring predictability barrier problem (McPhaden, 2003).

ENSO theory (Neelin et al., 1998) provides a robust framework to understand the existence of such predictability barriers. Indeed, the ENSO phenomenon can be modeled as an internal mode of the coupled equatorial ocean-atmosphere system, which can be self-sustained or excited by small-scale processes, often considered noise (Fedorov et al., 2003). Bjerknes' feedbacks are central in amplifying Sea Surface Temperature (SST) anomalies, whereas equatorial ocean wave processes provide delayed negative feedback and are responsible for the time scale of ENSO. Interactions between the internal mode and the external seasonal forcing can lead to chaotic behavior through nonlinear resonances (Tziperman et al., 1994; Jin et al., 1994). On the other hand, the dynamical behavior can be strongly influenced by noise, in particular, westerly-wind bursts (Lian et al., 2014). In addition, during boreal spring and summer, the Pacific climate system is most susceptible to perturbations leading to predictability barriers (Latif and Barnett, 1994). The growth of perturbations from a particular initial state has been investigated in detail from a much-used intermediate-complexity model, the ZC model (Zebiak and Cane, 1987a). DL methods are powerful statistical models which have now been used in a wide range of applications such as speech recognition and image reconstruction (Goodfellow et al., 2016). These methods also include the ANN; over quite some time now, DL have been applied to El Niño prediction Dijkstra et al. (2019). For example, in Ham et al. (2019b) CNN were trained on model data from the Climate Model Intercomparison Project, phase 5 (CMIP5), using transfer learning and subsequently trained on reanalysis data. The CNN-based scheme shows a better forecasting skill than most dynamical models, and this forecast skill remains high up to lead times of about 17 months. Second, in Petersik and Dijkstra (2020), Deep Ensemble Methods (Lakshminarayanan et al., 2017), in particular, and GDN and other MDN models, were used in ENSO prediction. These methods also give a skillful model for the long-lead time prediction of the ONI (and its uncertainty) using a relatively small predictor set. At the moment, there is an enormous effort to understand the performance of DL models generally referred to as explainable AI (Arrieta et al., 2020). The research described above shows that DL models are a very promising tool in ENSO prediction that can provide valuable skills for El Niño forecasts beyond the predictability barriers. In this framework, CNN can represent a powerful method

for forecasting ENSO events with lead times of up to one and a half years (Ham et al., 2019a) or for solving a binary classification problem in hybrid models with high complexity, multi-resolution input data (Yan et al., 2020). The predictions formulated by the CNN can, however, be made explainable through saliency maps (see section 1.3). Therefore, CNN represents the perfect choice for classifying the occurrence of ENSO events in ZC simulations and investigating in detail which features contained in data can lead to highly accurate predictions.

3.2 Zebiack-Cane model

The ZC model (Zebiak and Cane, 1987a) represents the coupled ocean-atmosphere system on an equatorial β-plane in the equatorial Pacific. In this model, a shallow-water ocean component is coupled to a steady shallow-water (Gill, 1980) atmosphere component (Fig. 3.1). The atmosphere is driven by heat fluxes from the ocean, depending linearly on the anomaly of the sea surface temperature T with respect to a radiative equilibrium temperature T_0. The reader will find more details about the ZC model in Appendix C. We use the numerically implicit fully-coupled version of this model, developed in van der Vaart et al. (2000), and slightly extended in Feng and Dijkstra (2017). In this version, the zonal wind stress τ^x is written as

$$\begin{aligned} \tau^x &= \tau^x_{ext} + \tau^x_c, \\ \tau^x_{ext} &= -\tau_0 e^{-\frac{1}{2}\left(\frac{y}{L_a}\right)^2}. \end{aligned} \tag{3.1}$$

Here the external part τ^x_{ext} represents a weak ($\tau_0 \sim 0.01\ Pa$) easterly wind stress due to the Hadley circulation, L_a is the atmospheric Rossby deformation radius, and y is the meridional coordinate. The zonal wind stress τ^x_c is proportional to the zonal wind from the atmospheric model which, in turn, depends on sea surface temperature. As shown in van der Vaart et al. (2000), the parameter measuring the strength of all ocean-atmosphere coupled feedbacks is the coupling strength μ. When $\mu < \mu_c$, where μ_c indicates a critical value, the Tropical Pacific climatology (a stationary state of the model) is stable. However, if the coupling strength exceeds the critical value μ_c, a supercritical Hopf bifurcation occurs, and sustained oscillations occur with a periodicity of approximately four years. A seasonal cycle is included in the model by varying μ over time with a specific amplitude $\Delta\mu$ and with an annual period.

Apart from the coupled ocean-atmosphere processes, ENSO is also affected by fast processes in the atmosphere, such as westerly wind bursts. These processes are considered as noise in the ZC model. The representation of atmospheric noise in the model is similar to that in Feng and Dijkstra (2017), where the westerly wind bursts are represented by one Empirical Orthogonal Function pattern (with the associated principle component fitted to an AR(1) process) in the zonal wind stress. The observation-based data set in Feng and Dijkstra (2017) contains weekly patterns of this wind-stress noise. In the ZC model, we randomly add one of such patterns at each time step (of a week) to the zonal wind stress. The effect of the noise on the model behavior depends on whether the model is in the super- or sub-critical regime (i.e, whether μ is above or below μ_c). If $\mu < \mu_c$, the noise excites the ENSO mode, causing irregular oscillations. In the supercritical regime, the cycle of approximately four years is still present, but the noise causes an irregular amplitude of ENSO variability.

While the ZC model is used for ENSO predictions, it also has its limitations as it cannot capture either tropical basin interactions (e.g. Atlantic and Indian Oceans) or tropical-extratropical interac-

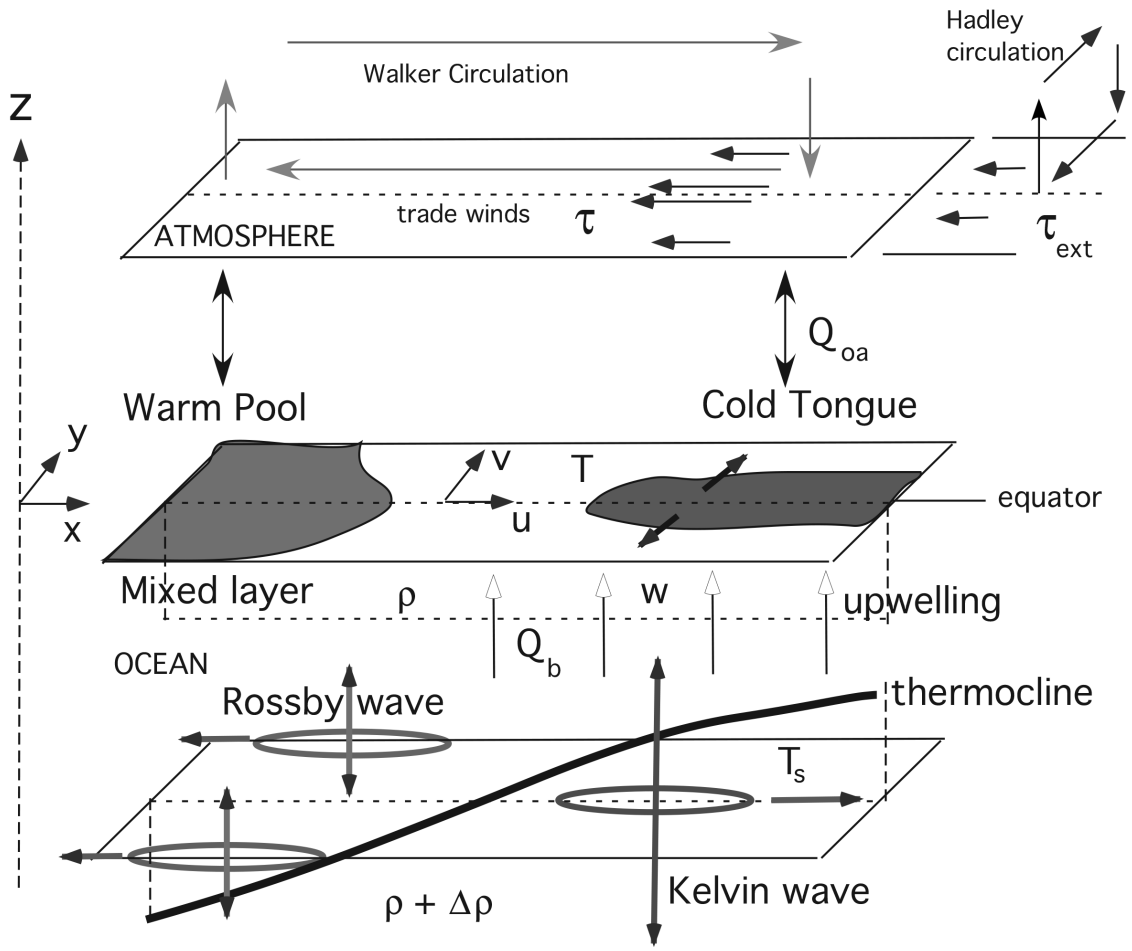


Figure 3.1: Schematic of the Zebiak-Cane model, where a shallow-water ocean model is coupled to a shallow-water atmosphere model through a mixed-layer ocean model with temperature T. The ocean-atmosphere coupling involves a heat flux Q_{oa} and a wind stress vector τ. This image was taken from Dijkstra (2005)

Effect	μ	δ	δ_s
distorted wave speed	2.7	0.5-1.5 (0.1)	0.3
distorted wave speed	2.7	0.5-1.5 (0.1)	0.3
distorted upwelling feedback	2.7	1.0	0.1-0.6 (0.05)
distorted upwelling feedback	2.7	1.0	0.1-0.6 (0.05)
reference	2.7	1.0	0.3

Table 3.1: Parameter settings of the ZC model used to generate the data used in the distorted physics experiments with the parameter step size shown within brackets. Parameter ranges are chosen to cover roughly a 50% increase and decrease compared to the reference value, step size is chosen to get around 10 points within this range. The parameters are from left to right: coupling strength μ, wave speed parameter δ and upwelling feedback parameter δ_s. The value of $\mu = 2.7$ is subcritical in the ZC model.

tions (it described only the dynamics of the Pacific). The model also cannot adequately represent a Tropical Pacific seasonal cycle and hence such a seasonal cycle is prescribed in the model.

3.3 Distorted physics experiments

The model experiments broadly consist of two steps: first, the ZC model is run for standard parameter values to produce reference case data; and then it is run again but for a range of values around the standard parameter value (shown in Table 3.1) to get the distorted data. This ultimately results in three different kinds of datasets: reference case, distorted wave speed, and distorted upwelling feedback. There are no simulations where more than one parameter is distorted simultaneously. In the second step, the distorted datasets are used as training data for the DL model whose performance is then determined by using the reference case as the test set. As a consistency check, the DL models are also trained on the reference case data and then tested on reference case data. This should produce the highest performance because the DL models are tested on data they have already seen.

3.4 1-D CNN for ZC data

To leverage the essential feature of the 1-D CNN (see section 1.1.3) models of encoding the sequentiality of the patterns contained in the simulated data, we fed the 1-D CNN with the simulated Time-Series obtained by rearranging the output of the ZC model. Indeed, the average value of the integration grid (representing the NINO3.4 region) is computed for each physical observable at each simulation step (corresponding to almost one month). More specifically, this synthetic collection of datasets gives a possible description of the temporal evolution in the NINO3.4 region of the following physical observables of interest, that is *thermocline depth,* SST *, wind speed, and zonal wind stress*. From the simulation of these four observables along the time domain, we extracted the instances that we propagated into the 1-D CNN model. We chunked the synthetic simulation of all observables in a sequence of overlapping time windows of 48 months and a stride interval equal to 1. As discussed in section 1.1.3, we precise that a "stride interval of 1" means that the two subsequent chunks have been extracted from two time-subdomain that differs by one month. Thus, every

single instance is a tensor of rank two whose dimensions are the time length (48 pixels, sampling frequency one month) and the number of time series features (i.e., the four physical observables of interest).

Labeling the instances was performed by equipping each instance with the corresponding ONI value. So we labeled one ENSO event whenever the ONI value was either greater than 0.5 (El Niño event) or lower than -0.5 (La Niña event).

The instances were pre-processed via standardization (each time-series feature is set with zero mean and unit variance) and grouped into the training sets and test sets; the validation of the 1-D CNN model is performed by means of the 5-fold cross-validation.

We evaluated the AUROC (a.k.a, AUC, i.e., Aera Under the Curve) on each fold; the mean value and the standard error mean provided the degree of accuracy of the CNN model and its error, respectively.

The 1-D CNN model we designed for this application consisted of the sequence of one *Convolutional layer* (64 filters, kernel size 9) with a Rectified Linear Unit (ReLU) *activation function* (i.e., $\mathrm{ReLU}(x) = \max\{0, x\}$), followed by a *max-pooling Layer* with pooling size 2. Dropout layers (Srivastava et al., 2014) with a dropout rate of 0.50 are also employed to reduce overfitting; no stride is applied during the convolutions (see section 1.1.3). After repeating this block of hidden layers twice, the resulting feature map is flattened via a flattened layer; a final *Fully-Connected Layer* with sigmoid activation function returned the output of the 1-D CNN model. During the training phase, the ADAM (Kingma and Ba, 2014a) algorithm was used as an optimizer for the Binary-Cross Entropy loss function; the batch size and learning rate were set equal to 128 and 0.005, respectively.

SMOE scale method (Mundhenk et al., 2019) was employed to construct robust saliency maps; as we described in section 1.3.2, the saliency maps generated by this method appear to be much more efficient and computationally faster than popular gradient methods. We exploited the capability of the SMOE scale method to detect those patterns within those spatial domains that mainly indicate the approaching or the occurrence of the ENSO events. Thus, we proceeded with analyzing the profile of the saliency maps to evince possible analogies and differences between the patterns learned during the training phase and the patterns contained in the test datasets.

In order to complete the analysis provided by the SMOE scale method, we focused on how the predictions can change when only a spectral sub-band of the input instances is propagated through the hidden layers of the 1-D CNN model. Through this approach, we aimed to investigate how oscillations present in ZC data under a specific regime can represent a particular aspect of the predictive performance of the 1-D CNN model. Therefore, we progressively applied a digital Butterworth (Butterworth et al. (1930), Hamming (1998)) filter of order three as either a bandpass filter or low-pass filter to select the instances of the 1-D CNN model. The ensemble of low-pass and bandpass filters we used was designed to analyze the whole spectral domain of the instances while being non-overlapping simultaneously. We imposed that the cutoff frequencies of each bandpass filter were in a ratio of 1:2; starting from the Nyquist frequency ν_0, the first digital filter selected the frequency band $[\frac{\nu_0}{2}, \nu_0]$, the second one $[\frac{\nu_0}{4}, \frac{\nu_0}{2}]$, and so on; see Table 3.2. Once again, we choose the cut-off frequency of the low-pass digital filters according to a dyadic scale, i.e. the first filter had the cut-off frequency ν_0 (Nyquist Frequency), the second one $\frac{\nu_0}{2}$, the third one $\frac{\nu_0}{4}$ and so on; see Table 3.2. Note that this digital filtering-based approach was repeated on each fold of the 5-fold cross-validation; that is, after processing the instances via the digital filters, we propagated the filtered instances through the fitted 1-D CNN model and then evaluated the AUROC precisely as we did in the validation step of the 1-D CNN model. It is important to stress that at this stage, we

Frequency bands (Period in months)	Cut-off period (months)
$[2, 4)$	2
$[4, 8)$	4
$[8, 16)$	8
$[16, 32)$	16
$[32, 48)$	32

Table 3.2: Frequency bands and cut-off frequencies for the band-pass and low-pass digital filters, respectively.

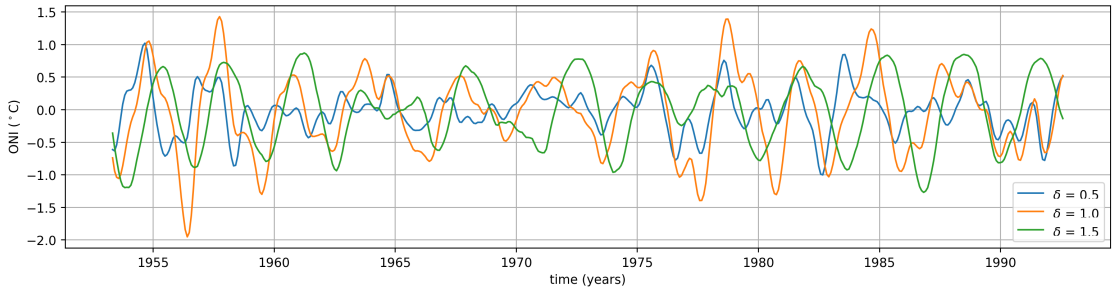


Figure 3.2: Time-series of ONI calculated from ZC model simulations using δ parameter values of 0.5 (blue), 1.0 (orange), and 1.5 (green).

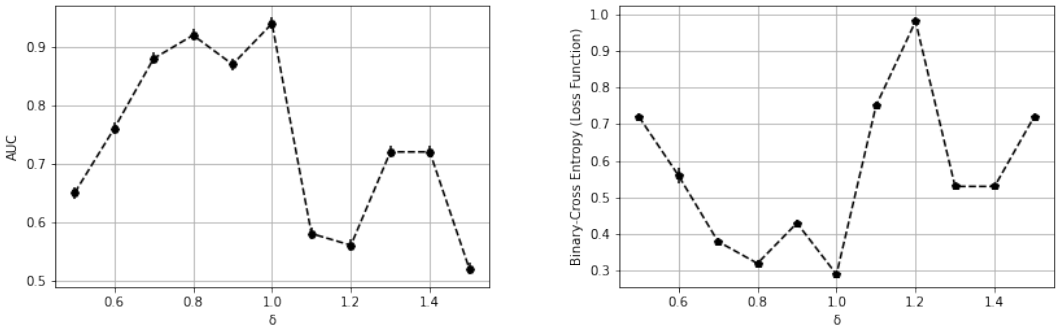
neither refit the 1-D CNN model on the filtered instances nor alter the architecture of the 1-D CNN model fitted already.

Hence, we used this approach to reveal which time scales were dominant in those patterns that characterized the ENSO events in ZC data (i.e., those patterns retrieved by the SMOE scale). In particular, we tried to understand how the periodic behavior of ZC-data was an essential feature that the 1-D CNN model tended to capture to solve the classification task. The distorted physics experiment led to an alteration of this feature and, consequently, a decrease in the 1-D CNN model capability of classifying the ENSO events in ZC data.

3.5 Analysis of saliency maps for ZC data

3.5.1 Distortion of equatorial wave dynamics

Time series of the ONI for the different δ values, as computed from the ZC model, are shown in figure 3.2. Changing the δ value causes the amplitude of the oscillation to become much smaller for $\delta < 1$, so much even that by definition only ENSO neutral conditions ($-0.5 < ONI < 0.5$) are present. Increasing δ above the reference value of 1.0 initially leads to an increase in the oscillation amplitude, then decreases again for higher values of δ. This is expected because the ENSO period depends on the speed of Rossby and Kelvin waves crossing the Pacific basin. In the study of the classification performance of the 1-D CNN, we took a prediction lead time of 9 months. The propagation of the δ-distorted data through the 1-D CNN leads to substantial changes when testing the model's accuracy on the reference data. By construction, the AUROC score (figure 3.3a) attains excellent results at $\delta = 1.0$ (AUROC 0.94) as the CNN is trained on the reference data. The



(a) (b)

Figure 3.3: The AUROC score (a) and the loss function (b) as a function of the equatorial wave speed δ. Each point represents the mean AUROC over five different folds; error bars are evaluated via standard error mean.

AUROC scores tend to remain relatively high (peak of AUROC 0.91 at $\delta = 0.8$) as the δ parameter is slightly decreased from its reference value. Instead, as δ is reduced up to value 0.5, we observed a severe degradation of the accuracy with respect to the reference case; from $\delta = 0.7$, the evaluation of the AUROC metrics decreases monotonically (AUROC 0.66 at $\delta = 0.5$). At values $\delta > 1.0$, we observed a total reduction of the AUROC values. Specifically, models trained for $\delta = 1.1$ and $\delta = 1.2$ show low AUROC values as 0.58 and 0.56 but the lowest value (AUROC 0.51) is reached at $\delta = 1.5$.

When the reference data are propagated through the CNN models, the evaluation of the loss function (see 3.3b) confirms the scenario described for the AUROC score. Indeed, the global minimum value is achieved at $\delta = 1.0$, and a relative minimum is also present at $\delta = 0.8$. When δ is decreased or augmented towards the bound values $\delta = 0.5$ and $\delta = 1.5$, respectively, we can observe the loss function tends to reach higher values. In particular, an increase or decrease in the AUROC along the δ domain is followed, respectively, by a decrease or an increase in the loss function.

The application of the SMOE Scale (see 1.3.2) on the mean instance (i.e., the instance obtained by averaging all samples of the test data of the reference case associated with one event of interest such as El Niño and La Niña) can help identify which patterns of ZC data are captured by the CNN to obtain (more accurate or degraded) ONI predictions. The reason why we focused on the analysis of the mean instance is that it is the most representative instance containing those patterns that are related to the main features of ZC data; In addition, the visualization of one main saliency map would help to draw some conclusions. In contrast, the interpretation of the saliency maps of all instances would be impractical. The mean instances of both the events El Niño and La Niña are represented infigure 3.4a andfigure 3.5a, respectively. Thus, after propagating the mean instance through a fitted 1-D CNN model, we extracted the activated feature maps for the hidden layer of the model and then computed the saliency map via the SMOE scale method. We individualized the domains (expressed in months) of the saliency maps achieving the highest values; in those salient domains of the mean instance, we visualized the patterns that were mainly captured by the 1-D CNN model.

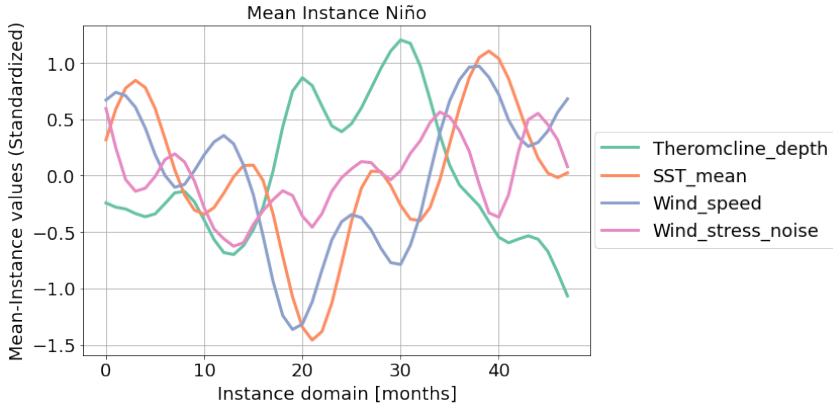
Taking into account the event El Niño, the saliency map of $\delta = 1$ reference case (green line in-figure 3.4a) shows two peaks with an intensity of 0.6 and 0.9 around months 18 and 36, respectively (note that the instances are 48 months long and that the lead time is 9 months). These two regions turn out to be the most salient along the whole domain of the mean instance. At month 18 we find a peak in the thermocline depth (Fig. 3.4a, green line) and a trough in both sea surface temperature (Fig. 3.4a, orange line) and wind speed (Fig. 3.4a, indigo line). Conversely, in the neighborhood of month 36, we find the thermocline is descending towards a trough, while both sea surface temperature and wind speed are reaching a peak value. Both these two combinations of patterns represent the main characteristic that mostly defines the event El Niño according to the recognition activity of the CNN model.

Likewise, we found some similar results for the event class La Niña. The spatial locations, where the saliency map (figure 3.5 b, green line) achieves values close to unity correspond to one interval domain (months 0-7) of the mean instance (see figure 3.5a) where the thermocline depth attains a peak while both the sea surface temperature and the wind speed descend towards a trough.

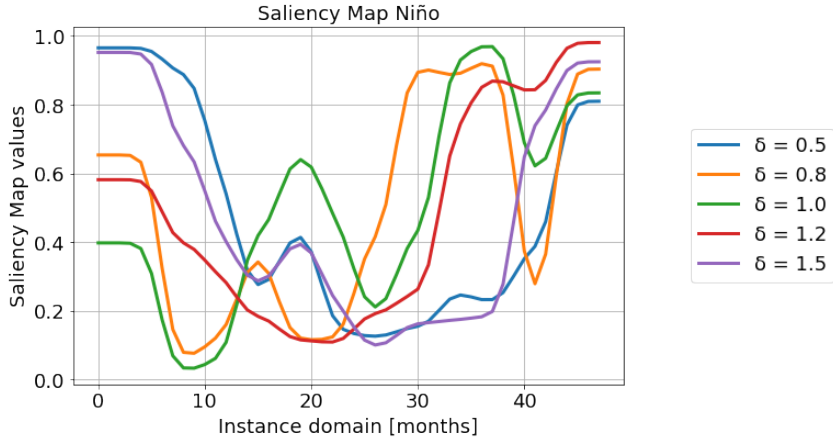
When considering other distorted cases, such as $\delta = 0.5$ and $\delta = 0.8$ (where waves are propagating faster than the reference case), the combination of peaks in the thermocline and troughs in the sea surface temperature (and vice versa) still represents those relevant time-series patterns that the CNN model captures during the learning phase. If we focus our attention on the event El Niño, the saliency map of case $\delta = 0.8$ (seefigure 3.4b, orange line) shows at months 30-38 a broad region with intensity larger than 0.8. Whereas, the saliency map of case $\delta = 0.5$ (Fig. 3.4b, blue line) shows intensities close to unity in the region 0-10 months. In the first case, we find that the high saliency region corresponds to a peak in the thermocline and a trough in both the sea surface temperature and the wind speed, while in the latter case we find the thermocline depth shows a soft minimum value as opposed to the high valued peak in the sea surface temperature and wind speed.

Similar results are also obtained for the event La Niña. The saliency map of case $\delta = 0.5$ (seefigure 3.5 b, blue line) shows at months 35-40 a saliency region with intensity higher than 0.85. Infigure 3.5 a, we see that this high saliency region corresponds to a peak in the thermocline depth and a trough in both the sea surface temperature and the wind stress. The saliency map of case $\delta = 0.8$ (seefigure 3.5 b, orange line) reveals a broad high-valued peak (maximum intensity 0.81) around month 18 and a flat salient region (intensity close to 0.9) at months 40-48. By looking at those temporal regions infigure 3.5 a, we find a peak in both the sea surface temperature and the wind stress together with a deep trough in the thermocline depth around month 18, whereas the domain months 40-48 show a soft trough in the thermocline and a prominent peak in both sea surface temperature and wind speed.

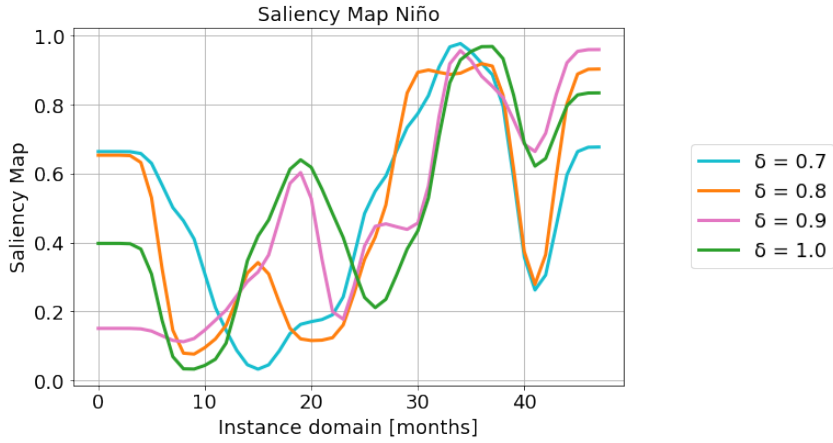
A deeper insight into the region $\delta = 0.7 - 1.0$ (where the AUROC takes the highest values) reveals that all CNN models tend to capture a specific type of time-series patterns when they have to deal with the recognition of the event El Niño. For all cases considered in this interval, the saliency maps indicate as interesting the region months 32-38 (see Fig 3.4 c.); where the values attained are larger than 0.8. Thus, we find that the results previously discussed for both cases $\delta = 0.8$ and $\delta = 1.0$ (where we discussed the behavior of the mean instance around month 36) are still valid even when we consider both cases $\delta = 0.7$ and $\delta = 0.9$. Interestingly, the case $\delta = 0.9$ shows a recognition activity that is similar to that of the model trained under the reference case because the saliency maps appear to be partially overlapped (see both the pink and the green line of figure 3.4 c at months 32-38). Moreover, the saliency map of case $\delta = 0.9$ points out some other aforementioned



(a)

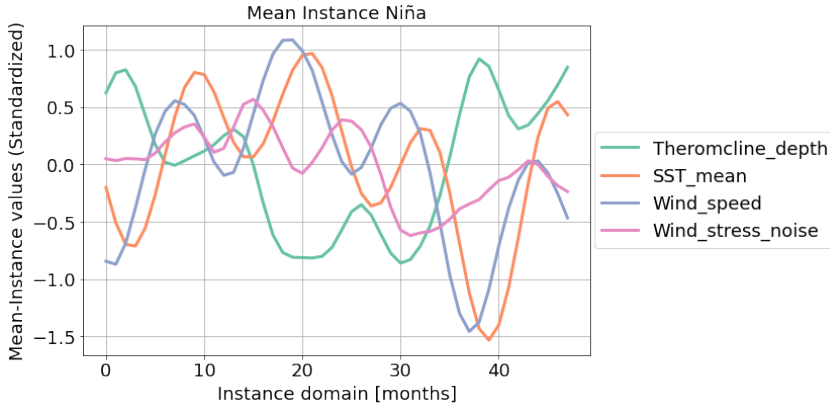


(b)

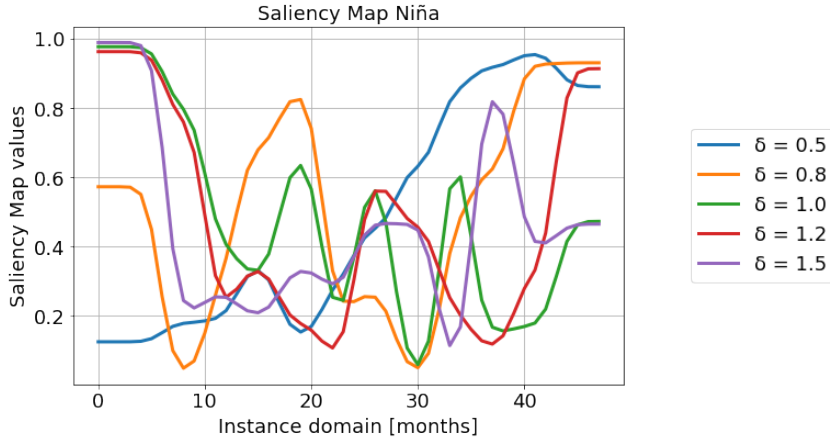


(c)

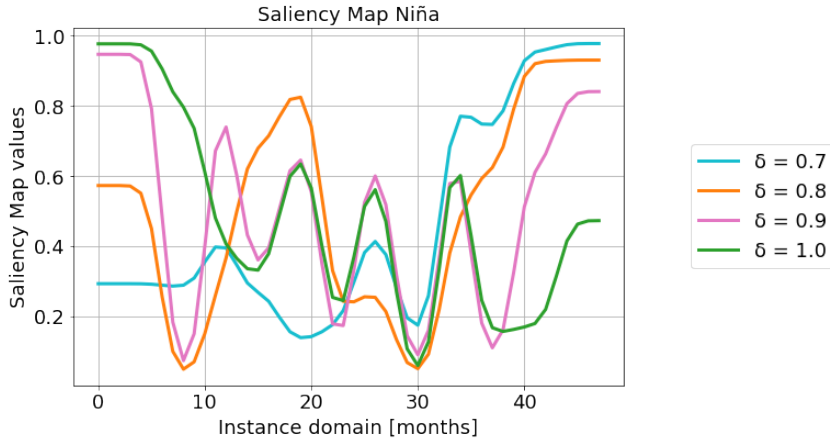
Figure 3.4: The mean instance considering all the El Niño event instances in the test data (reference case) (a). Saliency Maps of CNN models (b)-(c) trained with the wave distorted data (variation of δ) considering the cases (b) $\delta = 0.5, 0.8, 1.0, 1.2, 1.5$ and (c) $\delta = 0.7, 0.8, 0.9, 1.0$.



(a)

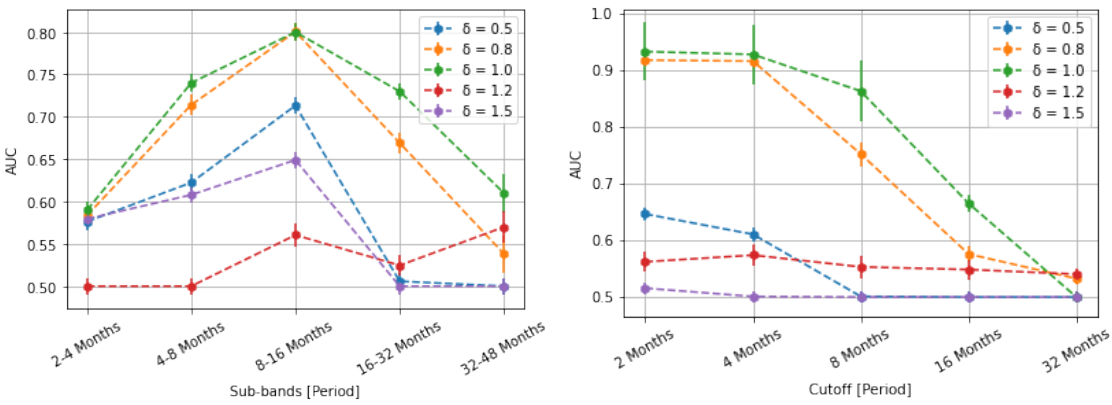


(b)



(c)

Figure 3.5: The mean instance considering all the event La Niña instances in the test data (reference case) (a). Saliency Maps of CNN models (b)-(c) trained with the wave distorted data (variation of δ) considering the cases (b) $\delta = 0.5, 0.8, 1.0, 1.2, 1.5$ and (c) $\delta = 0.7, 0.8, 0.9, 1.0$



(a) (b)

Figure 3.6: The AUROC score for different values of δ for the event El Niño as a function of (a) the band-pass frequency range and (b) the cut-off frequency, obtained by filtering the data by (a) band-pass Butterworth digital filter and (b) a low-pass Butterworth digital filter.

details of interest, e.g. those corresponding to the peak with intensity 0.6 at month 18; as shown by the pink line in figure 3.4 c.

For La Niña event, instead, the saliency map of case $\delta = 0.7$ (Fig. 3.5c, cyan line) presents some analogies with case $\delta = 0.8$ (Fig. 3.5c, orange line), individualizing one highly salient region at months 40-48 with an intensity around. Similarly to case $\delta = 1.0$ (Fig. 3.5c, green line), the saliency map of case $\delta = 0.9$ (Fig. 3.5c, pink line) individualizes a salient region in proximity of the left edge of the instance domain (months 0-5) with an intensity around to 0.95. It is interesting to note that both the saliency maps of cases $\delta = 0.9$ and $\delta = 1.0$ are overlapped at the middle region (months 15-35); both the two CNN models show a similar approach to capturing some low relevant features to identify the event La Niña.

For the cases $\delta = 1.2$ and $\delta = 1.5$ (where waves are propagating slower), the saliency maps (figure 3.4 b, red and purple line) reveal that the region around month 18 is no longer salient as in the reference case for El Niño event. For case $\delta = 1.2$, we find a salient region at months 36-48, where the saliency map takes values larger than 0.8. This corresponds to the presence of one broad peak in both the sea surface temperature and the wind speed with a less important contribution (than in the reference case) in the thermocline depth located at months 32-48. For case $\delta = 1.5$ we can observe that the saliency map is similar and almost completely overlapped with that of case $\delta = 0.5$; in this case, the analysis of the most salient time-series patterns will lead to some results that have already discussed for the case $\delta = 0.5$.

When considering the event La Niña, the saliency map of case $\delta = 1.2$ (seefigure 3.5 b, red line) attains values with an intensity close to unity at months 0-10. This temporal domain is characterized by the opposite feature, i.e. a broad peak in the thermocline depth and a trough in the sea surface temperature, located at months 0-10. The same feature can be also found for the case $\delta = 1.5$. Indeed, the saliency map (seefigure 3.5 b, purple line) shows high saliency regions at either months 0-10 (intensity values close to unity) and month 36 (a peak with a maximum of 0.81). In particular, at month 36 the sea surface temperature and the wind speed reach a deeper trough with respect to that of region months 0-10.

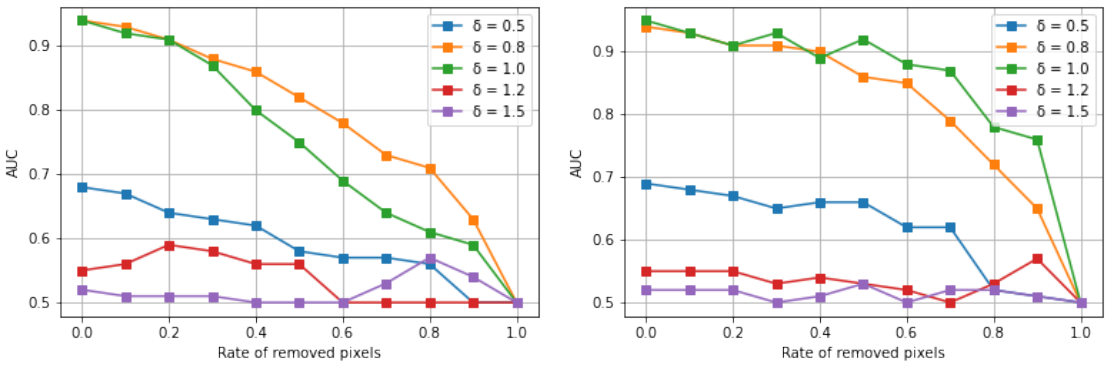


Figure 3.7: Evaluation of AUROC when the ROAR method (a) or the replacing at random strategy is applied (b); on the x-axis the ratio of pixels that are replaced and on the y-axis the AUROC value.

The application of a band-pass filter on all the instances included in the test dataset (reference case data) reveals that the propagation of one specific frequency band through the CNN models can retrieve most of the AUROC scores obtained with the non-filtered data; as shown in figure 3.6. Specifically, the model trained under the reference case turns out to be very sensitive to the frequency band corresponding to periods 8-16 months, where the AUROC is equal to 0.80 (Fig. 3.6a, green line). On the contrary, the complete degradation of AUROC scores is attained when propagating lower and higher frequency bands, e.g. both intervals 16-32 months and 2-4 months, where the AUROC value is equal to 0.61 and 0.58, respectively. Similar results can be found for other cases taken under consideration, as the case $\delta = 0.5$ and $\delta = 0.8$ (Fig. 3.6a, blue and orange lines). For both these cases, the band 8-16 months turns out to be the most predictive one with a net degradation of AUROC score as soon as slower frequency bands are considered.

In particular, case $\delta = 0.8$ still shows some analogies with the reference case; the frequency band 8-16 months is still the most predictive with an AUROC score of 0.80, and net degradation occurs at either lower or higher frequency bands. Such a result offers further details in interpreting the saliency maps, i.e. the CNN models tend to capture oscillating trends with specific carrier frequencies within the low-medium band of frequencies. It is important to highlight that the presence of details on a shorter frequency scale (i.e. period 16-32 months) is still fundamental and needed to allow the CNN to make an accurate classification of the ENSO events. The smoothing of the sample instances with a low-pass filter (Fig. 3.6b) reveals the instances tend to substantially lose many of their discriminating patterns at cutoff frequencies as 8 or 16 months. For example, in the cases $\delta = 0.8$ and $\delta = 1.0$ (Fig. 3.6b, orange and green lines) we can observe a decrease in the predictive power with degradation of 0.1 AUROC at 8 months and 0.3 AUROC at 16 months. Hence, medium-low frequency patterns (4-8 months) as those contained in the thermocline depth or in the wind-noise time series can play an important role in the detection of ENSO events.

To ensure the correct implementation of the combined SMOE Scale and guarantee the validity of the results obtained, we used (and adapted to this analysis) the metrics ROAR (Remove and Retain) introduced in Mundhenk et al. (2019). The replacement in the validation sets of an increasing amount of salient spatial locations with zero-valued pixels rapidly deteriorates the predictive char-

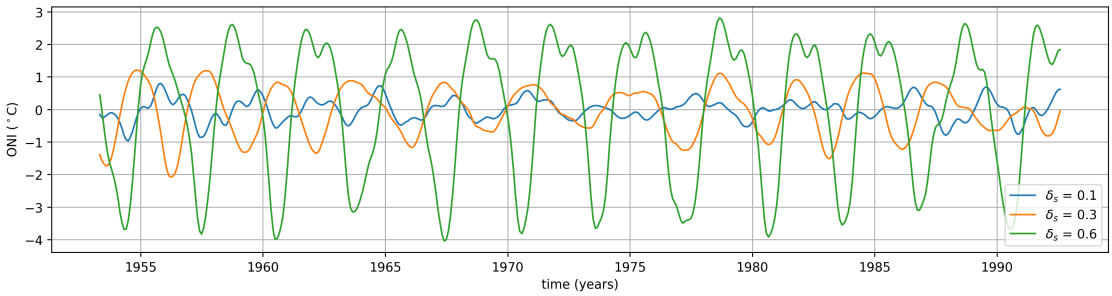
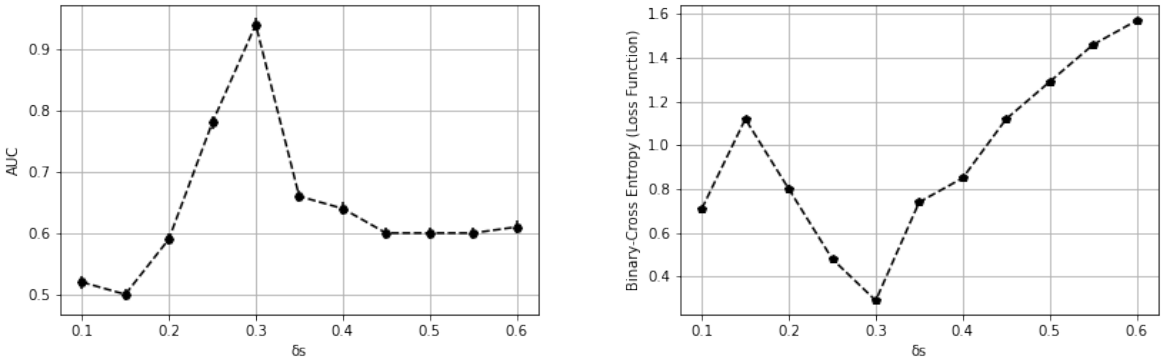


Figure 3.8: Several time-series of ONI calculated from ZC model simulations using δ_s parameter values of 0.1, 0.3, and 0.6 using $\mu = 2.7$.

acteristics of the data; as shown in figure 3.7. It is important to remember that the CNN models do not make use of any bias term either in the convolutional layers or in the dense layers. Accordingly, the CNN model considers the zero-valued patterns as absolutely non-informative, i.e. the propagation of such a pattern through the CNN is designed to prevent the activation of any stimulus along the hidden layers. Infigure 3.7a, we can observe that the removal of the top 50% salient pixels via ROAR (actually 24) guarantees a considerable decrease in the AUROC; under the reference case model, the AUROC scores present a loss equal to 0.20. Contrary to this, when randomly replacing the 50% pixels with zero-valued pixels we can still observe a slighter decrease in the AUROC curve under the reference, i.e. a loss equal to 0.03; seefigure 3.7b. Likewise, similar results can be found when even considering all the other distorted physics cases.

3.5.2 Distortion of upwelling feedback

We next considered the distortion of the model data due to a wrong representation of the upwelling feedback, controlled via the parameter δ_s in the ZC model. Figure 3.8 shows that the ONI's amplitude increases (decreases) for larger (smaller) values of δ_s. This behavior is expected because the upwelling feedback is a positive one, enhancing the existing sea surface temperature anomaly further and consequently increasing the amplitude of the ONI. The AUROC score versus δ_s curve (Fig. 3.9a) reveals that a particular tuning of the parameter δ_s strongly affects the accuracy of the CNN models when trained with distorted data. By construction, the AUROC score attains the highest score at the reference value $\delta_s = 0.3$ (AUROC 0.94). For $\delta_s < 0.3$ the profile of the curve suggests a net degradation in the AUROC scores with the lowest score attained at $\delta_s = 0.15$ (AUROC 0.5), whereas at $\delta_s > 0.3$ the AUROC scores remain stable, but still attaining values lower than 0.7. The profile of the AUROC has a plateau at values of 0.6 as δ_s goes towards the boundary value $\delta_s = 0.6$. The evaluation of the loss function (Fig. 3.9b) as a function of the parameter δ_s confirms the results obtained above. At $\delta_s = 0.3$ the global minimum is achieved, and the net degradation occurring at lower and higher $\delta_s = 0.3$ are still present; the loss function increases monotonically in both cases. Similarly to the analysis provided for the distortion of the δ parameter, we next consider the mean instances (of the test data of the reference case) and their saliency maps (figure 3.10a and 3.11a). For the event El Niño, we can observe that different regions of saliency can be associated to different variations of δ_s, i.e. for $\delta_s < 0.3$ the saliency maps (figure 3.10 b, blue and orange lines) indicate the left part of the instance as the most predictive, while for $\delta_s > 0.3$ the right part (figure 3.10 b, red and purples lines). In particular, the saliency map of



(a) (b)

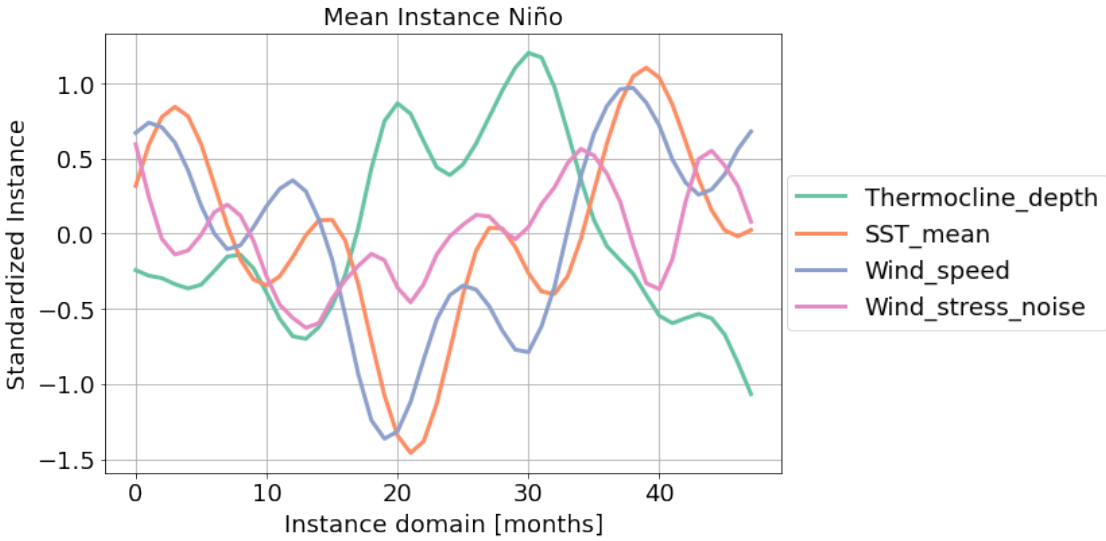
Figure 3.9: The AUROC score (a) and the loss function (b) as a function of the upwelling feedback parameter δ_s. Each point represents the mean AUROC over 5 different folds; error bars are evaluated via standard error mean.

cases $\delta_s = 0.10$ and $\delta_s = 0.25$ (figure 3.10 b, blue and orange lines) turn out to be very salient at 0-8 months, with intensity above 0.8. In that region, the mean instance presents a peak occurring in both the sea surface temperature and the wind speed time-series features. On the contrary, for cases $\delta_s = 0.45$ and $\delta_s = 0.60$ the saliency maps (figure 3.10 b, red and purple lines) achieve intensities larger than 0.8 around 32-48 months and capture one single broad oscillating peak in both the sea surface temperature and the wind speed time-series features.

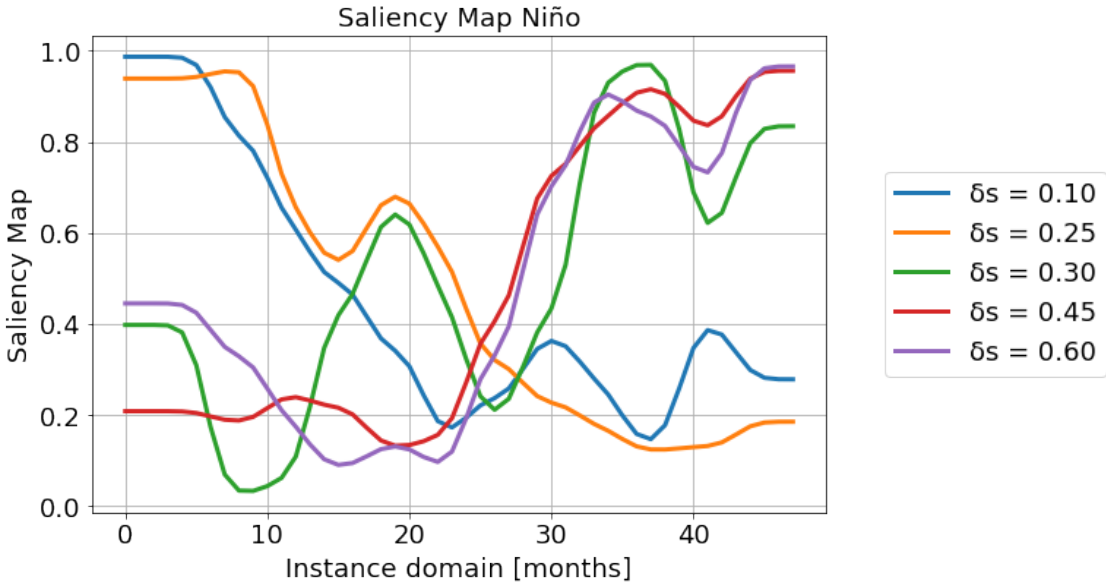
For the event La Niña, we refer to figure 3.11. In particular, the saliency maps of cases $\delta_s = 0.10$ and $\delta_s = 0.25$ (figure 3.11 b, blue and orange lines) present intensities larger than 0.8 at 42-48 months. It is interesting to observe that the saliency map of case $\delta_s = 0.25$ presents a plateau around 32-48 months; in opposition to the event El Niño, the CNN here captures a deep trough in the sea surface temperature time-series feature.

The application of band-pass and low-pass filters on the sample instances brings to light a result similar to the analysis done for the parameter δ; as shown in figure 3.12. When applying a band-pass filter with bandwidth 8-16 months, the case $\delta_s = 0.25$ (Fig. 3.12a, orange line) can partially retrieve the original prediction with AUROC 0.70, whereas for other cases such as $\delta_s = 0.6$ (Fig. 3.12a, purple line)the original prediction can be retrieved only by oscillations lying within the frequency band corresponding to 32-48 months. The smoothing of the instances via a low-pass filter (Fig. 3.12b) shows that the removal of high-frequency patterns oversimplifies the data, and so the classification task cannot be solved by the information contained in the low-frequency data only.

As confirmed by the filtering of the instances, the frequency bands 4-8 months and 8-16 months represent the main frequency bands in the reference case ($\delta_s = 0.3$). Capturing one of these two can retrieve a considerable amount of skill. The case $\delta_s = 0.25$ focuses on a large number of relevant patterns mainly in the frequency band 8-16 months. The filtering with a low-pass digital filter also reveals that a cut-off frequency of 16 months can reduce the AUROC in both cases, but a cut-off frequency of 8 months leads to degradation for the reference case only. In the latter scenario, we register a loss of 0.1 AUROC, i.e. a degradation on the same order of magnitude as when testing the reference case data and the data of case $\delta_s = 0.25$. Hence, this example shows how manipulation

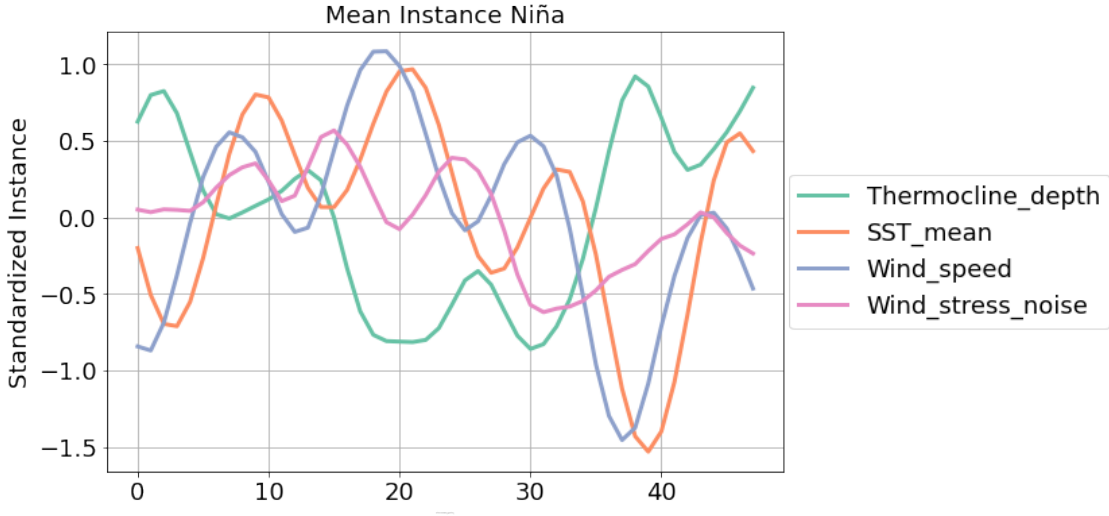


(a)

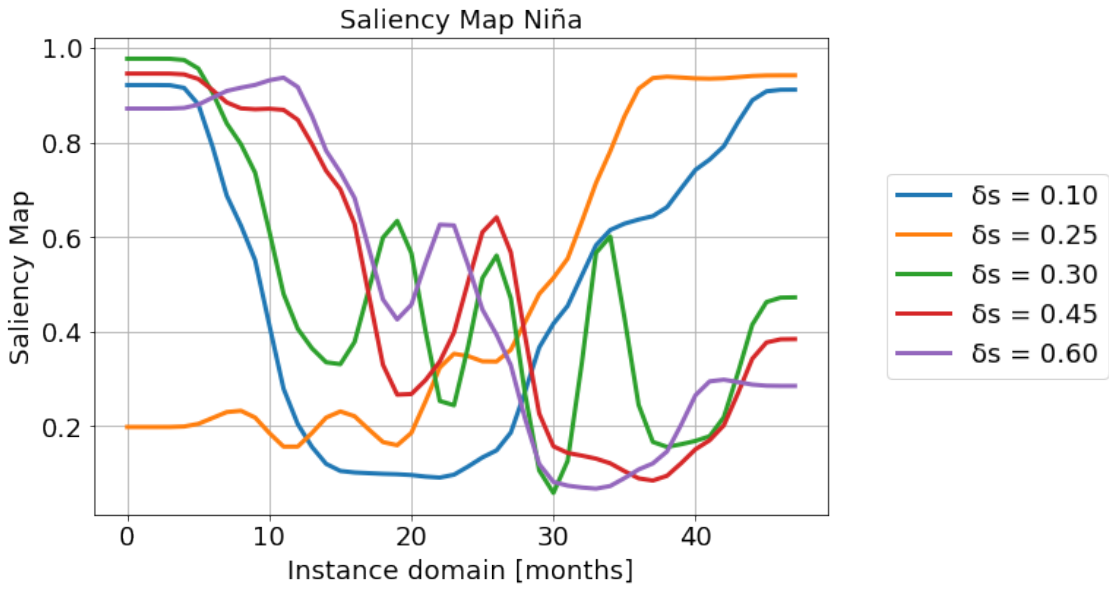


(b)

Figure 3.10: The mean instance (a) of all the El Niño event instances in the test data (reference case). Saliency maps of CNN models (b) trained with the upwelling distorted data (variation of δ_s). Specifically, cases $\delta_s = 0.10, 0.25, 0.30, 0.45, 0.60$ are considered.

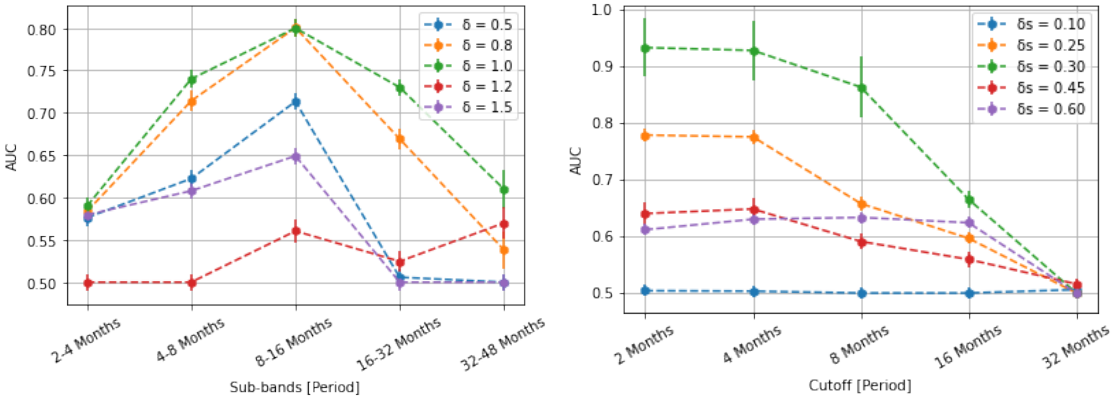


(a)



(b)

Figure 3.11: The mean instance (a) of all the La Niña event instances in the test data (reference case). Saliency Maps of CNN models (b) trained with the upwelling distorted data (variation of δ_s). Specifically, cases $\delta_s = 0.10, 0.25, 0.30, 0.45, 0.60$ are considered.



(a) (b)

Figure 3.12: The AUROC score for different values of δ_s for the event El Niño as a function of (a) the band-pass frequency range and (b) the cut-off frequency, obtained by filtering the data by (a) band-pass Butterworth digital filter and (b) a low-pass Butterworth digital filter.

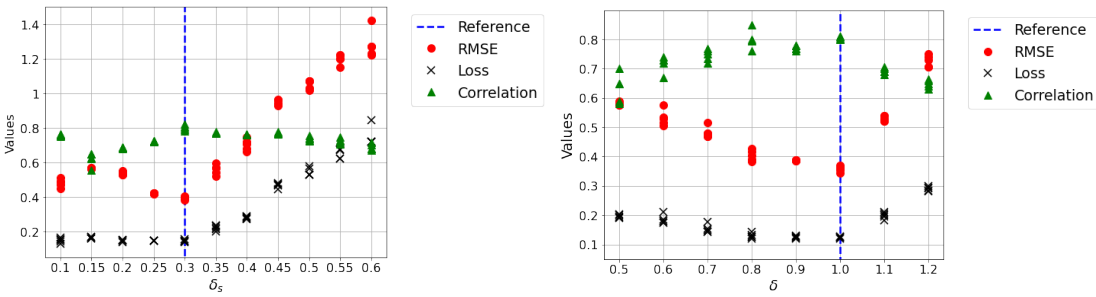
of the intrinsic characteristic of the instances can lead to a reduction and oversimplification of the instances, i.e. the distortion of the periodicity of data provokes a reduction or missing of some patterns that are fundamental in the classification of the reference case data.

3.6 Comparison of 1-D CNN and GDN

To provide a comparison with another DL method, we also applied the distorted physics approach in the Gaussian Density Neural Network (GDN), as used in Petersik and Dijkstra (2020). The variable to be predicted (or target variable) is also the ONI at a (lead) time in the future. The features used in the GDN are described by Petersik and Dijkstra (2020). All feature datasets are normalized before training.

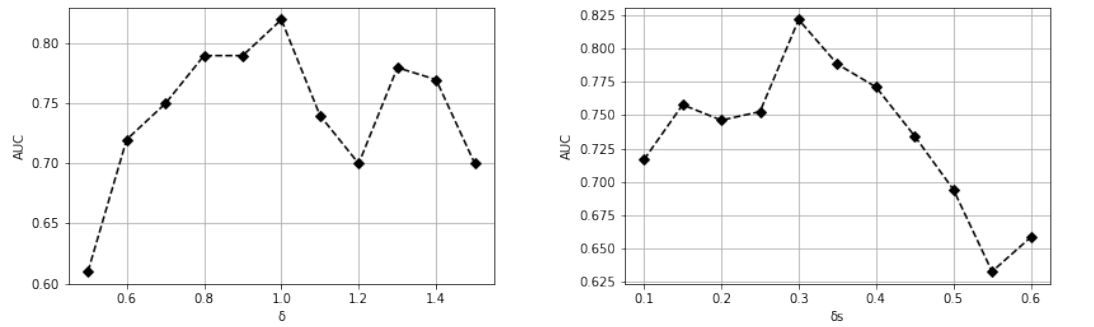
Training the GDN consists of a number of ensemble members that are trained in parallel. Each of the members is trained for 100 iterations over 500 epochs with a batch size of 100. The training starts with a random selection of hyperparameters within bounds defined by the user and is then optimized using the ADAM algorithm (Kingma and Ba, 2014a) with a user-specified learning rate, dropout, and Gaussian noise. The resulting ensemble members each predict a mean and standard deviation of the target variable, and these predictions are then averaged over the ensemble for the final prediction. Once again, we opted for a lead time of 9 months.

We used two different measures for the performance of the GDN: the Root Mean Squared Error (RMSE) and the *Pearson correlation*; also the loss function is shown (see figure 3.13). Different simulations gave different networks and different performance values. The GDN, when trained on distorted physics data, still performed consistently when varying δ or δ_s. However, a change in the ONI's amplitude in the training data (such as for higher than reference δ_s) is poorly corrected, leading to a large overestimation of the predicted variable (e.g., see $\delta_s = 0.40$ in figure 3.13 a). The attempt of comparing the capability of both CNN and GDN in detecting El Niño events were made complicated by the intrinsic design of both models. Although both models are trained to solve the same problem, we had to take into account that the 1-D CNN model has trained a binary classifier,



(a) (b)

Figure 3.13: Performance of the GDN when trained on distorted ZC model data using several values of (a) δ_s and (b) δ.



(a) (b)

Figure 3.14: AUROC metric for the GDN when considered as a classifier for both the wave distorted case (a) and the upwelling distorted case (b). On the x-axis the values of ZC parameters (δ, δ_s), and on the y-axis the AUROC score.

while the GDN was designed to solve a regression problem. In addition, the fact that both models optimize the same loss function does not ensure that a similarity lies in what the two models learn during the training phase. Another difference between both two models lies in the probabilities that they estimate; the 1-D CNN estimates the probability of the event itself, whereas the GDN estimates the probability distribution of the ONI index. However, the ENSO events are based on the behavior of the ONI index and we exploited this fact to make the outputs of the GDN close to those of the CNN. After training the GDN, we used the estimation on the Gaussian density to estimate the probability of El Niño events, i.e. the probability that the absolute value of the ONI index is greater than 0.5 $^{\circ}C$. Thereafter, we used the AUROC metric to compare the performance of the two models.

As shown in figure 3.14, the GDN model appears to be less accurate than the 1-D CNN model. The reference case data show a lower AUROC (compare to Figure 3.3), and we can observe a general reduction of 0.1 AUROC with respect to the results obtained with the 1-D CNN model. When feeding the GDN model with ZC data with a different tuning of parameters δ, we can observe that GDN tends to be more degraded at $\delta < 1$ than the CNN model (compare to Fig. 8a); in fact,

the AUROC can lose up to 0.21 with respect to the reference case. Note that the same tuning of parameter δ would reveal a plateau in the AUROC score whose value is much closer to that one attained in the reference case. When considering the distortion of parameter δ_s, we observed a degradation at values lower than 0.3. However, the decrease in the AUROC scores appears milder (~ 0.1) with respect to that shown for the CNN model. On the contrary, as $\delta_s > 0.3$, there is a significant reduction in the AUROC scores; with respect to the reference case, the AUROC scores can now be reduced up to 0.2.

3.7 Summary and discussion

This work was strongly motivated to understand the high skill in ENSO prediction obtained with the 1-D CNN approach in Ham et al. (2019b) in particular at long lead times. Although heat maps were presented in Ham et al. (2019b), their analysis does not connect immediately to the detailed processes of ENSO dynamics, which is also tricky because of the wide range of data they used. In this work, we introduced distorted physics simulations with the well-known ZC model (Zebiak and Cane, 1987a) to determine how a CNN can perform on accurate data when trained on data from 'wrong' model simulations.

The behavior of the ZC model can be elegantly described by a delay-differential equation (Suarez and Schopf, 1988; Jin, 1997)

$$\frac{dT(t)}{dt} = aT(t) - bT(t-d) - cT^3(t); \tag{3.2}$$

for the eastern Pacific temperature T as a function of time t. Here the constant a indicates the strength of the positive feedback, b that of the delayed negative feedback (with a delay d due to equatorial wave dynamics), and c measures the strength of the nonlinear equilibration.

By distorting the δ parameter in the ZC model, we modified the delay d in (3.2) and hence mostly the adjustment processes in the equatorial Pacific. When the equatorial wave speeds are distorted, there is an asymmetry in the skill of the 1-D CNN. For faster waves $\delta < 1$, the performance remains good, whereas for $\delta > 1$ (slower waves), it deteriorates. For example, in case $\delta = 1.2$, the El Niño event appears to be mainly constituted by slower oscillations, even though the behavior of the large-scale thermocline depth and sea surface temperature s similar to the reference case. However, the loss of details on shorter time scales leads the 1-D CNN model to solve the classification task still reasonably.

By distorting the parameter δ_s, we basically modify the feedback parameter a in (3.2) and hence the amplitude of the El Niño events. However, also the stability properties of the background climate state are changed, as seen through the shift in the Hopf bifurcation with δ_s (van der Vaart et al., 2000). For increasing δ_s and constant μ (as is done here), the background destabilizes as can also be seen in figure 3.8. The case $\delta_s = 0.1$ (reference case $\delta_s = 0.3$) offers a clear example of how the manipulation in the upwelling feedback can degrade the AUROC, i.e., the distortion of the patterns in the data leads to misplacement and misalignment and reduce the capability of the network in capturing the correct patterns at the right (temporal) location. For other cases (e.g., $\delta_s = 0.25$, $\delta_s = 0.45$ and $\delta_s = 0.6$), the skill of the CNN predictions is reduced less because the right combination of peaks and valleys in the time series are present. Indeed, the absence of oscillating terms located at the frequency band 4-8 months does not allow the CNN to capture all the relevant patterns but only a part of them.

The results indicate that the accuracy of the classification of the El Niño and La Niño events for lead times of 9 months using a 1-D CNN approach is strongly related to the capability of the CNN to capture the wave adjustment and feedback processes. The exact combination of specific patterns like peaks and valleys occurring at particular regions of the time domain of all features is essential to generate skill in the CNN predictions. The distorted physics approach can be advantageous in looking at how a 1-D CCN-based prediction scheme can represent other processes. For example, it is well known that connections between the Indian-Pacific (Izumo et al., 2010) and Atlantic-Pacific (Ham et al., 2013), as well as extratropical-tropical connections (Zhao and Di Lorenzo, 2020), are essential for the skill of ENSO predictions. The latter interactions have been described as ocean-atmosphere meridional modes and can influence ENSO and tropical variability on decadal time scales from both hemispheres independently (Amaya, 2019). Also, the effect of climate change on ENSO prediction skills and how a CNN would capture this is an interesting future line of work. However, one cannot use the ZC model for such studies and needs to do such distorted physics simulations with more sophisticated global climate models

3.8 Future developments: physics captured by DeepGreen Network

In this section, we shall present future developments in physics captured by ANN. In particular, we focus here on Green's function; this function plays an essential role in either Physics or Applied Mathematics. It is broadly used to solve boundary and initial values problems. In many contexts, Green's function coincides with the inverse operator of several linear differential operators. Accordingly, it represents a source of information about the linear differential operator one aims to study. For example, the ocean component of ZC equations (see Appendix C) admits a Green's function, which is a combination of both Kelvin and Rossby waves (Dijkstra, 2005). We shall present here an example of how using ANN to estimate the Green's function of a non-linear operator as the Van der Pol operator; we shall also focus on the limits of this approach.

3.8.1 Green's function

Let $\mathcal{L}$ be a linear differential operator acting on a collection of generalized functions over a subset $\Omega \subset \mathbb{R}^d$; the Green's function (Arfken and Weber, 1999; Evans, 2010) $G(x, y)$ is that function that satisfies

$$\mathcal{L}G(x, y) = \delta(x - y); \tag{3.3}$$

with $\delta(\cdot)$ denoting the Dirac's delta. Briefly, Dirac's delta is a *generalized function* (Kolmogorov and Fomin, 1957). According to Kolmogorov and Fomin (1957), a generalized function is any linear functional $T(\varphi)$ defined on the space $\mathcal{D} \subset \mathbb{R}$. $T(\varphi)$ satisfies the following conditions:

1. $T(a\varphi_1 + b\varphi_2) = Ta(\varphi_1) + bT(b\varphi_2)$;
2. $\varphi_n \to \varphi$, then $T(\varphi_n) \to T(\varphi)$. The notation $\varphi_n \to \varphi$ denotes that
   (a) There exists an interval external of which φ_n and φ are equal to zero.
   (b) The derivative $\varphi_n^{(k)}$ (with $k = 0, 1, 2 \dots$) converges uniformly to φ on that interval.

Dirac's delta can be heuristically defined as

$$\begin{cases} \delta(x) = 0 \quad \text{for } x \neq 0, \\ \delta(0) = \infty; \end{cases}$$

see Landau and Lifshitz (2013); in addition,

$$\int_{\mathbb{R}} dx \delta(x) = 1.$$

Alternatively, Dirac's delta can be defined via the Fourier transform (Landau and Lifshitz, 2013), namely

$$\delta(x) = \frac{1}{2\pi} \int_{\mathbb{R}} dk \exp(ikx) \leftrightarrow 1 = \int_{\mathbb{R}} dx \exp(-ikx)\delta(x).$$

Green's function turns out to help solve many boundary problems. For example, in many linear boundary value problems defined on a bounded domain $D \subset \mathbb{R}^d$, with $d \geq 1$, one aims to find out the solution $u(x)$ such that

$$\begin{cases} \mathcal{L}u = f \quad \text{in } D \\ u = 0 \quad \text{in } \partial D; \end{cases} \tag{3.4}$$

where $f : D \longrightarrow \mathbb{R}$ represent the source (or forcing) term; ∂D denotes the boundary of D. The solution of the (3.4) can be written:

$$u(x) = \int_D d^d y G(x, y) f(y); \tag{3.5}$$

Green's function acts, therefore, as the inverse kernel of the differential operator $\mathcal{L}$; its visualization enables ones to understand better the features of the differential operator that one wants to study. Note that (3.5) represents the inhomogeneous solution of (3.4). The complete solution of (3.4) can be expressed as

$$u(x) = u_{hom} + \int_D d^d y G(x, y) f(y);$$

with

$$\begin{cases} \mathcal{L}u_{hom} = 0 \quad \text{in } D \\ u_{hom} = 0 \quad \text{in } \partial D. \end{cases}$$

When considering a non-linear operator $\mathcal{N}$, Green's function can still be used to solve initial value problems. Frasca and Khurshudyan (2018); Frasca (2008) provided a formula for solving Ordinary Differential Equations of type

$$\frac{d^2 u}{dt^2} + \mathcal{N}(u, t) = f(t), \quad t > 0; \tag{3.6}$$

that is, they obtained the following short-time solution

$$u(t) = a_0 \int_0^t d\tau G(t - \tau) f(\tau) + \sum_{k=1}^{\infty} a_k \int_0^t d\tau (t - \tau)^k G(t - \tau) f(\tau); \tag{3.7}$$

with a_k are determined in terms of the quantities $\frac{d^k u}{du^k}|_{t=0}$. Note that in (3.7) Green's function $G(t,\tau)$ is still meant to solve the differential equation

$$\frac{d^2G}{dt^2} + \mathcal{N}(G,t) = \delta(t-\tau), \quad t > 0, \quad \tau > 0.$$

In several non-linear boundaries and initial condition problems, Frasca's formula arrested on the first term, i.e.

$$u(t) \simeq \int_0^t d\tau G(t-\tau) f(\tau), \tag{3.8}$$

can provide a satisfying numerical approximation consistent with the numerical solution obtained by the method of lines (Frasca, 2008; Khurshudyan, 2018a,b,c).

3.8.2 DeepGreen Network

In Gin et al. (2021), the usage of ANN to estimate Green's function is presented, i.e. the DeepGreen Network (DGN). Similarly to MDN, the DGN represents only a design of ANN (it can be composed of several MLP, CNN, RNN, or a combination of those) with the scope estimating some quantities of interest. Thus, DGN are trained to solve a regression problem where the targets are the solutions of a linear operator $\mathcal{L}$ (either linear boundary values or initial values problems) associated with different source terms; the source terms are the inputs of the DGN. Therefore, the task of DGN is estimating both the homogenous and inhomogeneous solution of $\mathcal{L}$. In particular, the latter is estimated by (3.5); therefore, the DGN is designed to evaluate Green's function associated with $\mathcal{L}$. A scheme of DGN is represented in figure 3.15

3.8.3 Learning Green's function of the Van der Pol oscillator

We can now focus on applying DGN on a specific problem. To be coherent with the work done in the previous sections of this chapter, we opted for a model exhibiting a specific property of ZC equations, i.e., supercritical Hopf bifurcation (Dijkstra, 2005). Thus, we consider the following non-linear homogenous ordinary differential equation, that is

$$\ddot{x} - \mu(1-x^2)\dot{x} + x = f(t); \tag{3.9}$$

where the dot notation denotes the time derivative, and μ is a real-valued parameter; $f(t)$ denotes a forcing term. The (3.9) is usually referred to as the equation of the *Van der Pol oscillator*. Similarly to the harmonic oscillator, the Van der Pol oscillator is characterized by a second-order differential equation. It has a unique equilibrium point at the origin, i.e., $\frac{d}{dt}x|_{x=0} = 0$. The introduction of the non-linear term μx^2, however, leads to the different behavior of the system's stability around the origin. Depending on the value of μ, the origin can either be a stable or an unstable equilibrium point; we can summarize the stability of the origin as follows

- $\mu \geq 0$; stable equilibrium point
- $\mu = 0$; the (3.9) reduces to the equation of the harmonic oscillator. This value represents the critical value of a supercritical Hopf bifurcation; a stable limit cycle surrounds the unstable equilibrium point at the origin.

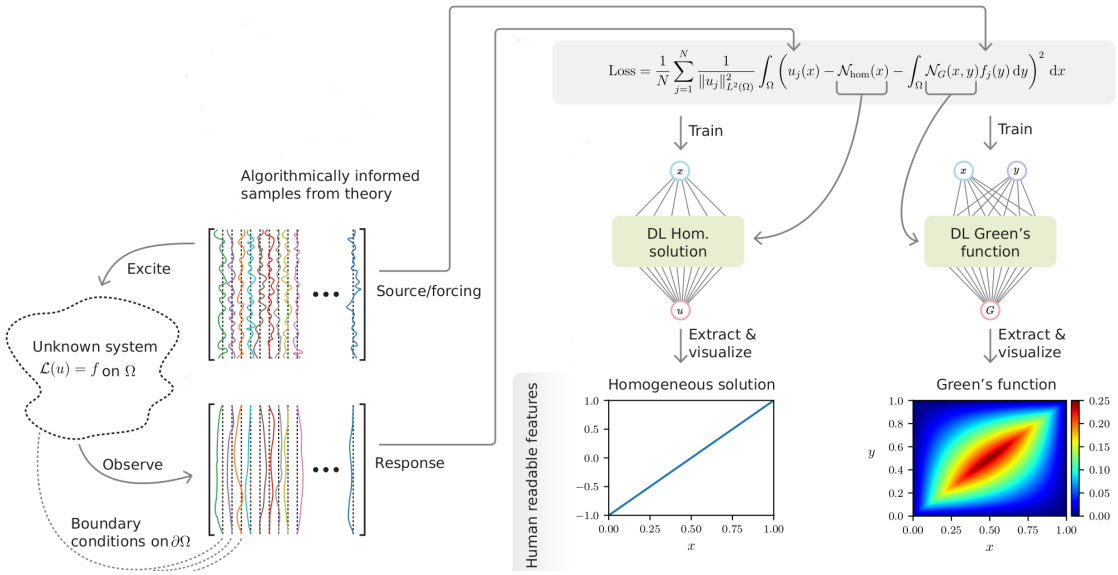


Figure 3.15: Scheme of the DGN used to estimate the Green's function of a linear differential operator The image was taken from Boullé et al. (2022).

.

- $0 < \mu < 2$; unstable spiral point; presence of a limit circle
- $\mu = 2$; unstable degenerate node
- $\mu > 2$; unstable node

A representation of the phase portraits at different values of μ is shown in figure 3.16.

In this framework, we want to exploit DGN to understand better the physics of some data that are known to follow the Van der Pol equation, even though they are affected by a perturbation. Thus, we want to use DGN to visualize the Green's function associated with the perturbed Van der Pol oscillator to have more profound knowledge about the number of physical features that an ANN can capture. This scheme wants to mimic the distorted physics approach studied in section 3.5; we will train a DGN for different values of $\mu \in [0, .1]$ and will try to draw some conclusions about how the GDN captures some partial physics of interest from some observations or generated data that are related to that physics of interest. In our example, we aim to visualize which features of the Van der Pol oscillator can still be captured by the DGN when the data are affected by a perturbation. Thus, we follow this strategy:

1. We generate a number N of source functions, with $N = 1000$. In doing so, we use the same approach of Boullé et al. (2022): we generate the source functions from a Gaussian process with covariance matrix $\Sigma(t, \tau) = \exp(-\frac{\beta}{2}(t-\tau)^2)$ with $t > 0$, and $\tau > 0$ (we prescribed $\beta = 16$).

2. We generate a dataset of source terms, and we denote it with $\{f\}_{i=1}^{N}$. We also generate data by solving

$$\ddot{u_i} - \mu(1-x^2)\dot{u_i} + u_i + \xi u_i^3 = f_i(t); \tag{3.10}$$

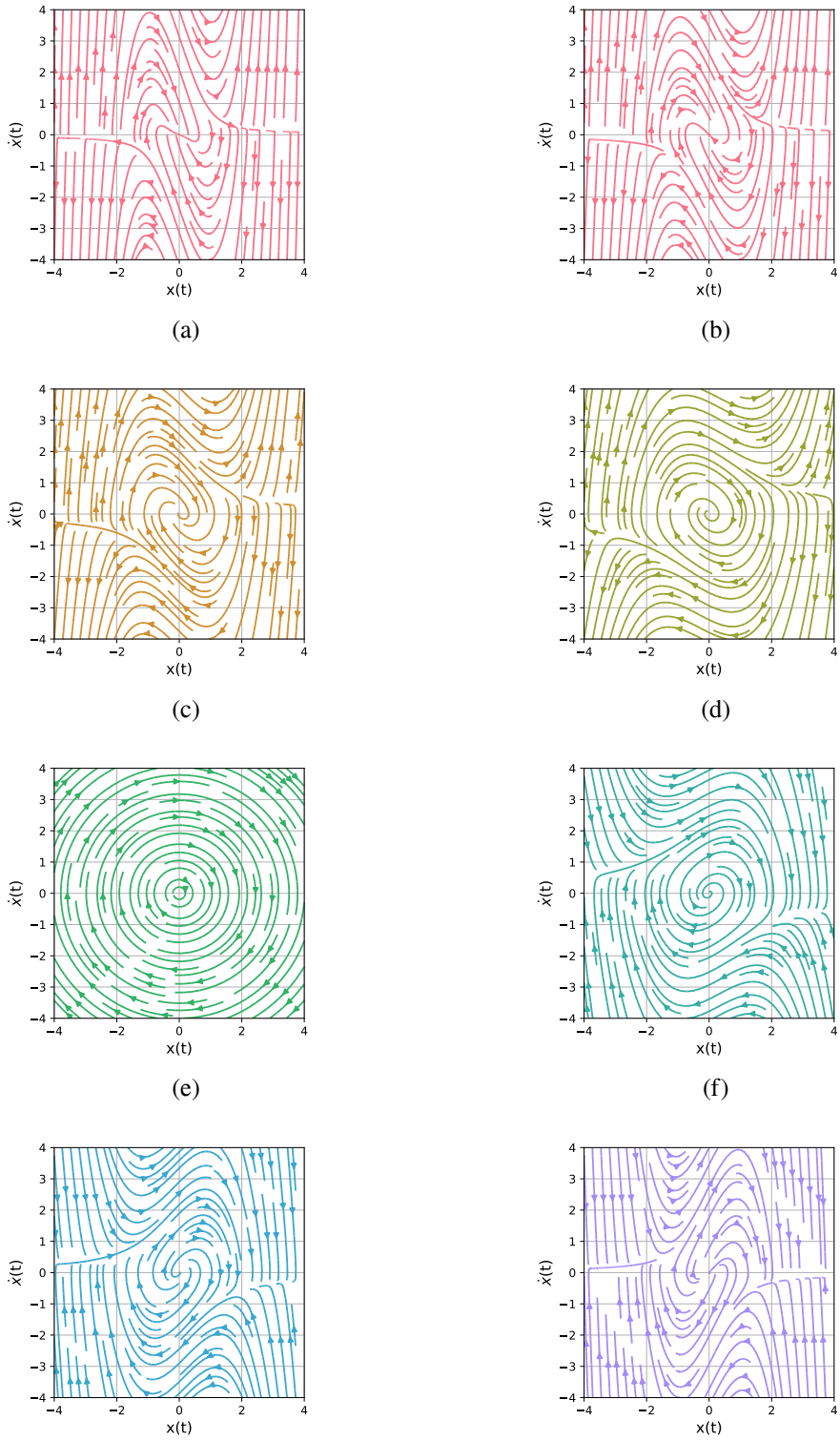


Figure 3.16: Streamlines of a vector flow of the Van der Pol oscillator. as a function of the parameter μ. (a) $\mu = -3$, (b) $\mu = -2$, (c) $\mu = -1$, (d) $\mu = -0.5$, (e) $\mu = 0$, (f) $\mu = 0.5$, (g) $\mu = 1$, $\mu = 2$.

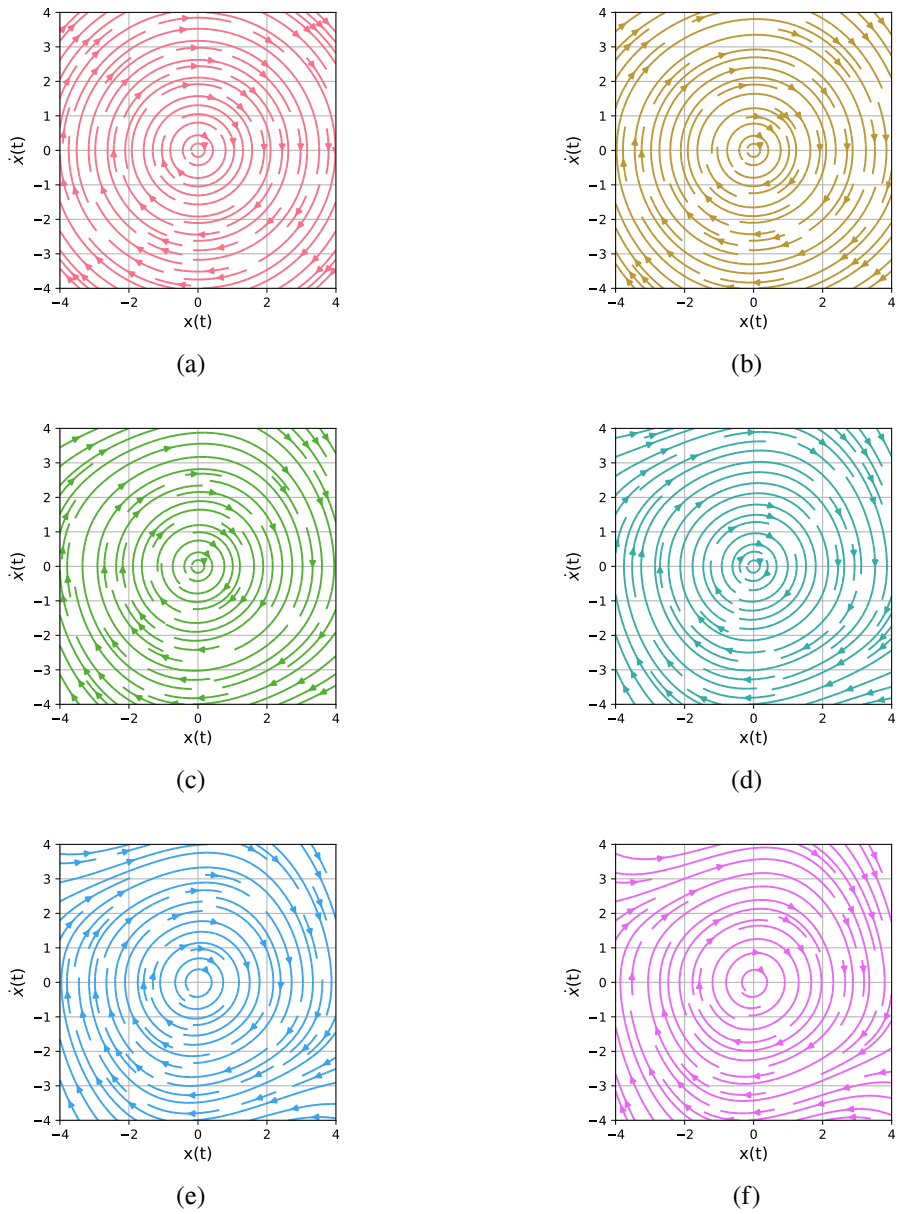


Figure 3.17: Streamlines of the vector flows of the perturbed Van der Pol oscillator as a function of the parameter μ. (a) $\mu = 0$, (b) $\mu = .02$, (c) $\mu = .04$, (d) $\mu = .06$, (e) $\mu = .08$, (f) $\mu = .1$.

by means of the Runge-Kutta-Felhberg method. With ξu_i^3 denoting a small perturbation (we prescribed $\xi = 10^{-3}$). We generated six different datasets, and each time, we prescribed another value of $\mu \in \{0, 0.02, 0.04, 0.06, 0.08, 0.10\}$; see figure 3.17 Note that each solution u_i is associated with the source term f_i. Moreover, both perturbative terms do not alter the stability of the equilibrium point.

3. We design a DGN as a MLP with one hidden layer and linear activation function. The inputs

are the forcing terms f_i, while the outputs $\tilde{u}_i$ are the estimation of the generated data u_i. Hence, the GDN minimizes the loss function

$$\frac{1}{N}\sum_{n=0}^{N}\|\tilde{u}_i - u_i\|^2 + \lambda\|W\|^2;$$

with W the kernel of the hidden layer and λ a shrinkage parameter equal to 10^{-3} The idea behind this architecture is simple; a single hidden layer enables one to estimate

$$\tilde{u}_i(t_k) = \sum_{j=0}^{T} w_{kj} f_i(t_j) \simeq \int_0^T d\tau G(t,\tau) f_i(\tau);$$

with T the total time of the forcing terms f. Similarly to (3.8), the DGN attempts to represent the solution u_i through a kernel matrix.

4. When training the DGN, we opted for a one-held-out selection of data; we once run the model after splitting the data in 80% training set and 20% test set.

5. The goodness of the DGN is evaluated utilizing the mean explained variance score, i.e.

$$\tilde{\Upsilon} = \frac{1}{N}\sum_{n=1}^{N} 1 - \frac{\mathbf{Var}(\tilde{u_i} - u_i)}{\mathbf{Var}(u_i)}; \tag{3.11}$$

the error is evaluated via SEM. The more the fraction of explained variance, the more the DGN is accurate in retaining all the details of the physics of (3.10). The visualization of the estimation of Green's function completes our discussion about the physics captured by the GDN.

From figure 3.18, we can see that the goodness of the DGN tends to decrease as the parameter μ increases. The reduction of non-linear terms ($\mu = 0$) reduces (3.9) to the equation of a perturbed harmonic oscillator. In this case, the DGN can retrieve 99% of the explained variance, and so the estimated Green's function turns out to contain all necessary information about the physics of the process. As the non-linear terms are getting more prominent (e.g., $\mu = 0.1$), we observe that the DGN cannot represent all salient characteristics of the generated data.

After training the DGN, we also evaluated the mean explained variance score of the outputs of the DGN with respect to the solutions of the Van der Pol oscillator without perturbation. That is, we integrated the solution of (3.9) for each source term f_i; we denote the collection of these solutions with $\{v_i\}_{i=0}^{N}$. Then, we evaluated

$$\Upsilon_v = \frac{1}{N}\sum_{n=1}^{N} 1 - \frac{\mathbf{Var}(\tilde{u_i} - v_i)}{\mathbf{Var}(v_i)}.$$

The quantity Υ_v points out the amount of variance captured by the DGN with respect to the solution of the Van der Pol equation. The orange line in figure 3.18 shows that the GDN can always capture a smaller piece of information more when the Van der Pol oscillator is not subjected to any perturbation. Anyway, the approximation (3.8) turns out to retrieve only a few salient characteristics of data ($\mu = .1$, $\Upsilon_v = 0.84$).

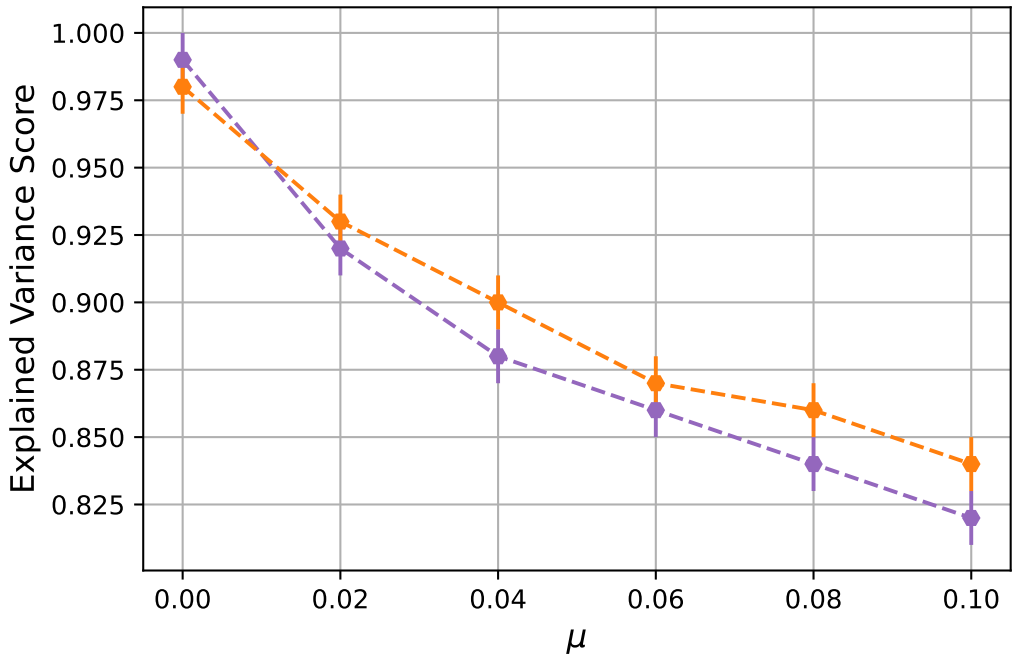
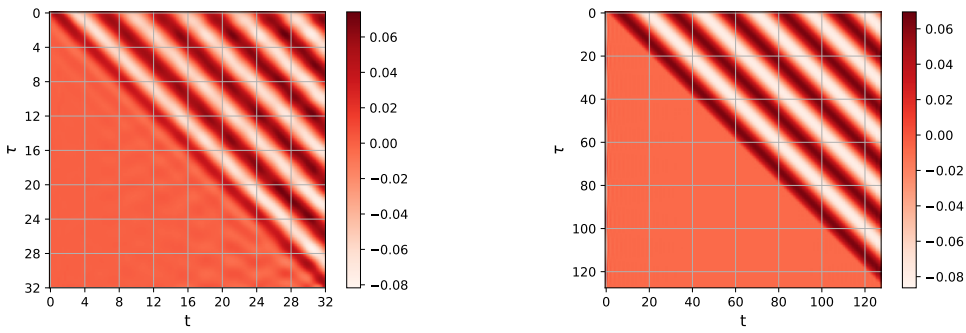


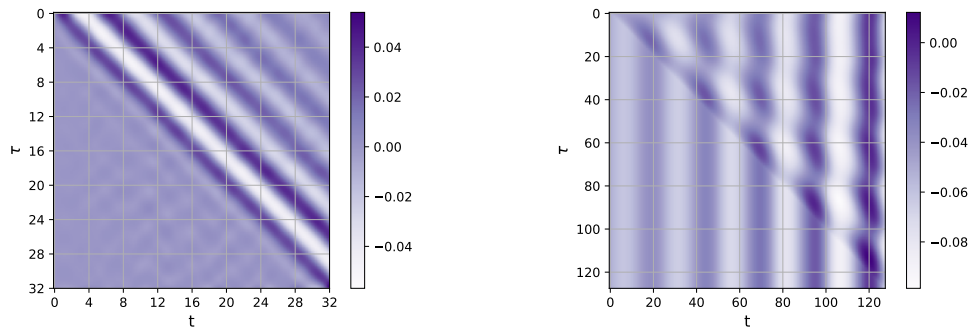
Figure 3.18: Explained Variance Score of the solutions estimated by the DGN as a function of the parameter μ. Purple line: numerical solutions of (3.10). Orange line: numerical solutions of (3.9)

.



(a) (b)

Figure 3.19: Case $\mu = 0$; (a) Green's function estimated via DGN and (b) Green's function estimated via numerical methods (RK-45).



(a) (b)

Figure 3.20: Case $\mu = .1$; (a) Green's function estimated via DGN and (b) Green's function estimated via numerical methods (RK-45).

For case $\mu = 0$, the visualization of Green's function points out that the DGN constructs a kernel matrix that keeps a trace of the characteristic oscillation that one would expect to find in the theoretical Green's function; see figure 3.19.

Whereas case $\mu = .1$ reveals a limit of the approximation (3.8), that is it is valid only for short intervals of time. Indeed, in figure 3.20, one can observe that the Green's function estimated by the DGN can essentially capture the oscillations occurring at the beginning of the processes, that is when $|t-\tau| \ll 1$. At later times the non-linear term μx^2 provides a limit cycle orbit that the DGN do not manage to capture. In this case, the implementation of higher-order terms of (3.7) should be taken to account to have a major control on the characteristics that can be learned by the DGN.

In figure 3.19 and 3.20, we compared the Green's function estimated by the DGN with the Green's function obtained via numerical methods. To obtain the numerical the solution of

$$\ddot{\tilde{G}} - \mu(1-\tilde{G}^2)\dot{\tilde{G}} + \tilde{G} + \xi\tilde{G}^3 = \delta(t-\tau); \tag{3.12}$$

we assumed that $\tilde{G} = \tilde{G}_0 + \mu\tilde{G}_1$; with

$$\ddot{\tilde{G}}_0 - \mu\dot{\tilde{G}}_0 + \tilde{G}_0 = \delta(t-\tau).$$

In our case, G_0 is the Green's function of the harmonic oscillator, namely

$$G(t,\tau) = \theta(t-\tau)\exp\left(-\frac{\mu}{2}(t-\tau)\right)\frac{\sin\psi(t-\tau)}{\psi}, \quad \psi = \sqrt{1-\frac{\mu}{4}},\ t>0,\ \tau>0;$$

with $\theta(\cdot)$ the Heaviside function. Hence, one needs to find G_1 from

$$\mu\ddot{\tilde{G}}_1 - \mu^2\dot{\tilde{G}}_1 + \mu\tilde{G}_1 + \mu\tilde{G}^2\dot{\tilde{G}} + \tilde{G} + \xi\tilde{G}^3 = 0; \tag{3.13}$$

that is, we integrated this differential equation by means of the Runge-Kutta-Fehlberg method.

Chapter 4

Dynamic prediction of ICUAI via two-step modeling

Hein …ah! oui, …oui c'est pour me singulariser. Tu comprends, pour un médecin, faire sa médecine, c'est banal!…Tandis que pour un couturier…

G. Feydeau. *Tallieur pour dames*

O binómio de Newton é tão belo como a Vénus de Milo. O que há é pouca gente para dar por isso.

F. Pessoa. *Poesias de Álvaro de Campos*

This chapter will present an application of the ANN model with hospital data. In particular, a ANN model was developed to improve both the forecasts of ICUAI and the extraction of information from the EHR recorded in the ICU monitors. When speaking of ICUAI, we refer to the nosocomially acquired infection that one patient might develop after being admitted to the ICU. The early detection of the onset of ICUAI remains the best strategy to contrast those fragile clinical conditions that often worsen in more severe conditions such as sepsis and, eventually, septic shock. Therefore, its accurate prediction can help reduce mortality in ICU. Unlike other popular outcomes of observational studies involving ICU patients (e.g., the death of a patient admitted to intensive care), the intrinsic difficulty of detecting one ICUAI episode at some precise time remains the main obstacle in providing a model that can predict this event with high precision; the time of the onset of ICUAI cannot be observed. Practically, the occurrence of an ICUAI is usually detected after the worsening of one ICU patient's clinical condition is overt; the ascertaining is generally made via the analysis of blood culture.

Here, the improvement driven by the ANN methods consists of analyzing the large amount of

high-detailed information coming with the continuous monitoring of ICU patients (see section 4.1). The ANN are then utilized to extract extra information not contained in traditional covariates (e.g., baseline covariates and other routine measures of vital parameters); the patterns that such a model analyze are different from the summary statistics that are usually included in survival models. In our, the implementation of an ANN method has the specific scope of formulating a risk variable (we shall refer to it as the *risk score of infection* or *infection risk score*) that should inform clinicians about the approaching of an infectious episode. Thus, ANN are fed with the EHR sampled at each minute; that is, we used the potentiality of ANN models to analyze a type of data that is usually not included in longitudinal studies. The novelty introduced in this work is, therefore, the exploitation of the risk score of infection to improve the performance prediction of a CR model (see, Appendix D.1), including a broad collection of traditional ICU predictors. Hence, we fit the CR model to a dataset including the risk score of infection and a wide range of traditional clinical variables. It's important to stress that the inclusion of the risk score of infection into the longitudinal dataset of clinical variables reflects our intention of approaching the prediction of the onset of ICUAI by means of two different data types, i.e., the low-frequency and high-frequency data. With the term *low-frequency data*, we refer to the traditional time-dependent clinical predictors that are sampled with a low sampling frequency, usually hours or days (see 4.3.4). Whereas *high-frequency data* are the EHR data which are sampled with a 1-minute sampling frequency (as mentioned, the fitting of ANN models to these data has the scope of including risk variables arising from the analysis of a highly-detailed source of information into a definitive collection of ICU predictors). Note that baseline covariates (e.g., sex, age, ICU admission type, and others) are also included among the ICU predictors (we shall discuss it in section 4.3.4).

Dynamic predictions (see section 4.3.5) are thus formulated to evaluate the probability that one ICUAI episode occurs in a time window of either 24 or 48 hours. In order to estimate the risk of infection, we used the Landmark model (Van Houwelingen, 2007; van Houwelingen and Putter, 2011), as described in section 4.3.6. Specifically, we implemented a Landmark Competing Risk Cox model. The idea behind this model is relatively simple and, in our case, it consists of fitting a Cox model on both time-fixed and time-dependent covariates in order to estimate the probability of ICUAI at any *horizon time* (i.e., the time where we want to make our prediction), given the patients' histories at some previous time denoted as *landmark time* (more details are discussed in section 4.3.7); an illustration is shown in figure 4.1. With the term patients' histories, we refer here to the low and high-frequency information and the baseline covariates introduced above. The reasons why we opted for the Landmark model are several; so we have

1. Although an ANN-based prediction exploiting both low and high-frequency data and baseline covariates would theoretically be more performative, such an approach is often regarded as *black-box* method whose predictions are not straightforwardly related to some interpretable quantities. The fitting of a CR model leads to the estimation of some interpretable quantities, such as the *hazard rate* and the *survival function* (see Appendix D).

2. The Landmark model allows us to include time-dependent covariates in a CR model while keeping feasible the prediction of cumulative incidences and survival probabilities (Cortese and Andersen, 2010); in our case, this is achieved by means of the estimation of the cumulative incidences at different moments of the ICU stay.

3. When integrated with a Cox model, the Landmark model is robust and relatively simple and

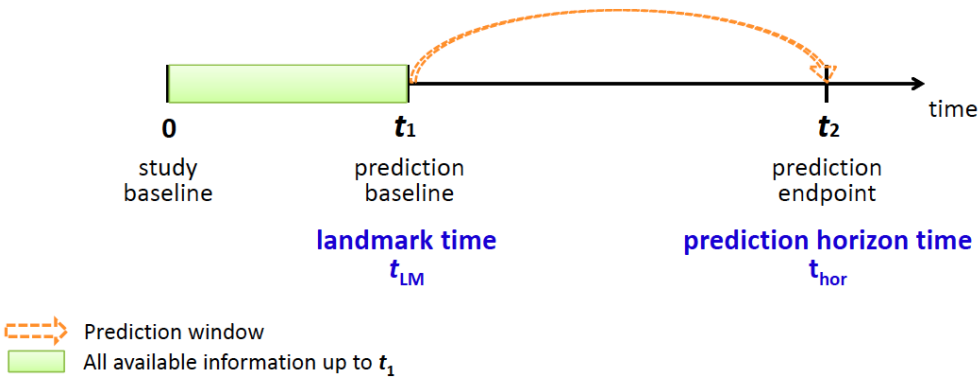


Figure 4.1: Scheme of the Landmark model.

offers an accurate estimation of cumulative incidences, even though the hazard proportionality assumption might be violated (Putter and van Houwelingen, 2017)

4. In our case, where a massive amount of data is involved while fitting the model, a more detailed model such as the *joint-modeling* would turn out to be impractical because of the high computational costs that such a model requires; as we shall discuss in section 4.3.5. Although the Landmark model is less accurate than correctly specified joint models in simulation studies (Putter and van Houwelingen, 2022); the simple structure of the Landmark model enables one to formulate accurate dynamic predictions rapidly. When the mean model is misspecified, however, simulation has shown that joint models may be inferior to a landmarking approach (Putter and van Houwelingen, 2022).

5. No censoring observations are included in our longitudinal data; the continuous monitoring of the ICU patients enabled us to be acquainted with the patients' clinical histories until either the discharge from the ICU or an infectious episode occurs (see section 4.2.1). In addition, the survival predictions cannot suffer from the fact of being underestimated because of an eventual neglection of the *censored observations* (Watt et al., 1996).

So, in summary, we fitted here an ANN model on the ICU EHR to evaluate a risk score of infection indicating the future occurrence of an ICUAI episode; namely, a novel variable based on the analysis of the high-frequency patterns recorded in the ICU monitors. The ANN we selected is the 1-D CNN (see section 1.1.3) model; we shall discuss the model selection in section 4.3.3. The infection risk score is then included in a wide range of longitudinal clinical predictors; this collection of explanatory variables contains information about the evolution of patients' trajectories at both low and high resolution. When evaluating the survival probability, we fitted a Landmark Competing Risk Cox model on all the longitudinal data in our posses. We shall refer to this methodology as *two-step modeling* (see section 4.3). Specifically, when using the two-step modeling, we focused on predicting either first or secondary ICUAI episodes. We shall refer to the model based on our two-step modeling as the *Deep-LCPH model*.

After investigating the possibility of improving the standard prediction modeling of ICUAI onset via 1-D CNN, we pass to interpret the predicting power of this methodology. In this case, we decided to restrict our focus on Intensive Care Unit First-episode Acquired Infections (ICUFAI)

(i.e., the clinical history of those patients who develop primary or secondary infections is restricted up to the first ICUAI episode) with a prediction time window of 24 hours. Including recurrent infection can enhance the data's complexity and variability. Thus, we applied the SMOE scale (see section 1.3.2) algorithm to visualize which patterns of the continuously monitored EHR are recognized and captured by the 1-D CNN model. Our goal is, therefore, the interpretation of those high-frequency clinical predictors that support the 1-D CNN model in early detecting ICUAI; we aim then at constructing some non-linear and complex predictors that improve the traditional way of extracting information from in the ICU setting; we shall discuss it in section 4.6.

For completeness, we also try to invert the forward operations of 1-D CNN model by means of Deconvolutional Neural Network (DNN) (Zeiler et al., 2010). In this case, we focused on the binary classification of ICU patients who increased their C-Reactive Protein (CRP) level in such a way as being suspected of experiencing an ICUFAI. We opted for the forecast of a high increase of CRP level because this laboratory value enables good clinical diagnostic and prognostic value for patients with sepsis and septic shock (Cui et al., 2019). In the ICU, the presence of Systemic Inflammatory Response Syndrome (SIRS) is determined daily after routinely measuring the concentration of CRP (Yentis et al., 1995; Povoa et al., 1998); SIRS indicates an inflammation that involves the whole body and is usually the main symptom related to sepsis (Chakraborty and Burns, 2019). The forecast of the high variation of CRP level in ICU patients represents, therefore, an alternative strategy to improve the early identification of ICUFAI. Traditional linear models have shown that the daily monitoring of CRP level represents a promising tool for the early recognition of ICUFAI events (Póvoa et al., 2006); likewise, ANN methods have been proposed as an emergent and precise methodology to forecast CRP Time-Series for medical purposes (Dorraki et al., 2019).

DNN are sequence rules designed to approximate the inverse operations of the hidden layers of a CNN model. As a result, DNN allows us backward to propagate the latent representation of the hidden layers to reconstruct the relevant piece of information contained in the input data. After reconstructing the input EHR, we focused on the extraction of some *macroscopic features* of the input EHR (e.g., mean value, standard deviation, mean change, and minimal and maximal value) at different moments of a 24-hour time window (e.g., sub-domains of 6 hours). We inspected the extracted macroscopic features via U-test (Mann and Whitney, 1947); this method allowed us to find those characteristics of EHR that mainly support the prediction formulated by the 1-D CNN (see section 1.1.3) model. The interpretation of these results will lead us to conclude that the prediction formulated by 1-D CNN can be regarded in terms of some clinical situations that usually prelude the occurrence of ICUFAI episodes.

4.1 The ICUAI problem

Nosocomial multiple infections represent one major cause of morbidity and mortality in the Intensive Care Unit (ICU) (Maki et al., 2008). The early detection of these events can help physicians in the treatment of the complications that might arise from the worsening of a patient's prognosis (Dantes and Epstein, 2018). Methods enabling the recognition of subjects potentially developing an ICUAI could practically improve the outcomes for these patients. ANN methods represent one of the most preferred approaches in clinical prediction literature nowadays (Pettit et al., 2021). The massive expansion in the usage of this method has been supported by the availability of ever larger and more complex data sets, e.g., the data extracted by the continuous monitoring of ICU patients. ANN methods, therefore, enable us to profoundly investigate the similarities and the features of

patterns characterizing subjects belonging to the same class of events; with a flexible approach, we do not need to meet any *a priori* theoretical assumption for model specification. On the other hand, we recall that these methods construct a decision function that cannot be expressed in a closed form (see Sec1.1.3); as a result, the prediction proposed cannot be straightforwardly interpreted.

The ICU represents the perfect environment where applying DL methods. Bedside measurements and observations have already become digitized mainly with comprehensive high-time resolution (e.g., 1-minute sampling frequency) clinical data. For example, ANN methods are often used to detect anomalous patterns of continuously monitored clinical data, such as trends, fluctuation, and periodicity; these irregularities in clinical data might often be informative over the clinical deterioration of patients (Tonekaboni et al., 2018; Henry et al., 2015; Suresh et al., 2017; Brand et al., 2018). Recently success of ANN in predicting more and more accurately healthcare outcomes has given the prospection that intelligent, high-performance algorithms could analyze an increasingly wide range of detailed healthcare data in order to predict and manage patient outcomes in a way that is not humanly possible; this has consequently generated a great deal of excitement (Topol, 2019; Zeng et al., 2022; Ivanov et al., 2022; Komorowski, 2020).

As sepsis remains one major cause of morbidity and mortality worldwide (Schlapbach et al., 2020; Alrawashdeh et al., 2022), methods enabling the earlier detection of patients potentially developing nosocomial infection have already been pursued by clinical researchers (Komorowski, 2020; Henry et al., 2015). Howsoever, the dynamic prediction of nosocomial infections represents a unique and challenging approach due to the frequently uncertain nature of these outcomes. We recall that unlike the modeling of other events of interest occurring in ICU, such as the death of a patient admitted to intensive care, the early detection of ICUAI episodes reveals an intrinsic difficulty in being detected at some exact moment of the ICU stay. The overt worsening of one ICU patient's prognosis is often a sign suggesting the occurrence of an ICUAI episode. In addition, secondary infections can sequentially occur one after the other in a short interval of days. Therefore, the prompt and accurate recognition of acquired infections remains a challenging problem for either statistical researchers or medical staff; it still represents an active area of automatic learning research in the ICU (Fleuren et al., 2020).

4.2 MARS ICU cohort

4.2.1 Study design

This study was conducted within the framework of the Molecular Diagnosis and Risk Stratification of Sepsis (MARS) study (ClinicalTrials.gov identifier NCT01905033), a prospective ICU cohort, for which the institutional review board approved an opt-out method of informed consent (protocol number 10-056C). We considered a broad collection of longitudinal EHR data from 5538 patients admitted to a medical-surgical tertiary ICU of the University Medical Center Utrecht (UMCU) in the Netherlands between January 2011 and December 2018. We considered patients older than 18 years only and with ICU stay longer than 48 hours.

Our outcome was the onset of suspected ICUAI. Infectious episodes occurring at times more extensive than 48 hours after the ICU admission were identified using data from the MARS-data repository (Klouwenberg et al., 2013). The reference time of nosocomial infection diagnosis was defined by either the beginning of empirical antibiotic treatment or any blood culture ascertaining the infectious state, whichever occurs first.

We selected candidate predictors among several variables based on literature review, a priori consensus of clinical importance, and prevalence in the study population. These covariates include both time-fixed variables reflecting the baseline risk of infection, as well as time-dependent data representing the dynamics of the clinical evolution of patients over time, e.g., laboratory values and physiological response and organ function parameters; see Table 4.1 and 4.2. In addition, we also considered 1-minute resolution data from monitored vital signals (i.e., EHR); these data were collected from the University Medical Centrum Utrecht ICU database.

We designed our model by means of a two-step modeling in order to exploit all longitudinal clinical data and include all data with different temporal resolutions, namely:

1. We first used ANN methods (1-D CNN and 2-D CNN; see section 4.3.3) to investigate on longitudinal evolution of EHR data. In our case, the EHR data are those high-frequency vital signals that are recorded in the ICU monitors and are sampled with a sampling frequency of 1 minute. Thus, ANN methods are leveraged to evaluate the *risk score of infection* (or more simply *risk score*), i.e., we introduced here a novel predictor based on the recognition, capturing, and analyzing of those patterns contained in the EHR data that are mainly connected with the onset of ICUAI. The infection risk score is a condensed form of the information contained in high-frequency physiological signals of an ICU patient about his/her clinical history. As a result, the risk score of infection reflects the risk that one patient might develop a nosocomial infection in the next 24 or 48 hours.

2. We modeled a Landmark CR Cox model, including all the explanatory variables in our possess. That is, we included in a unique longitudinal dataset both low-frequency and high-frequency predictors and the baseline covariates. The baseline covariates represent all the time-fixed covariates expressing the characteristics of the ICU patients, such as sex, age, ICU admission type, and comorbidities; see Table 4.1. Low-frequency predictors are all time-dependent covariates expressing the physiological evolution of ICU patients with the variability of hours or days (e.g., circulation, respiration, consciousness score, laboratory measurements, and bacterial colonization); see Table 4.2 High-frequency data coincide with the EHR used to train the ANN model; incorporating the risk score of infection in the longitudinal dataset has precisely the scope of ensuring the inclusion of such a type of high-detailed information. By definition, low and high-frequency data are not sampled at the same time; for this reason, the evaluation of the risk score enables us to integrate the information arising from the high-frequency data at the time stamps where the low-frequency predictors were sampled. The implementation of the Cox proportional hazards CR is designed to have two failure causes only: the onset of an acquired infection or the occurrence of one of some other exclusive events, such as the death of the subject or his/her discharge from the ICU.

In this work, we considered the ANN outputs as condensed information about the possible approaching of the onset of one infectious episode in the early future. Note that the ANN output is based on the analysis of EHR data only. We recall (as introduced in section 1.1.3) that during the learning phase, the ANN model constructs a specific transformation that aims to reduce the high dimensionality of the input data in order to make the input features linearized at most, i.e., the ANN solves a classification problem by transforming and reducing the number of input features in such a way that a hyper-plane can separate the classes of events. Accordingly, the skill of ANN in classifying infectious EHR data in a way that is not humanly possible is exploited to improve the early recognition of ICUAI.

Variable name	Variable description
Sex	Sex (male/female)
Age	Age at ICU admission
Immunodeficiency	Immunocompromised status; defined as having acquired immune deficiency syndrome, the use of corticosteroids in high doses (equivalent to prednisolone of >75 mg/day for at least 1 week), current use of immunosuppressive drugs, current use of antineoplastic drugs, recent hematologic malignancy, or documented humoral or cellular deficiency
Readmission	Previous ICU admission during current hospitalization period
Primary specialty	Diagnostic category of ICU admission (cardiovascular, gastrointestinal, neurological, respiratory, post-transplantation, trauma, other)
Diabetes Mellitus	Medical history of diabetes mellitus
Chronic corticosteroid use	Chronic medication use: systemic corticosteroids
Chronic organ failure	Presence of chronic organ insufficiency with one of the following conditions documented in medical history: • Chronic heart failure defined as the medical history of chronic NYHA class 2-4 or documented ejection fraction <45% (on echography in 2 years prior to ICU admission) or orthopnea with chronic diuretic use • Severe cardiovascular insufficiency defined as angina or dyspnea in rest or during minimal exercise (NYHA IV) • Chronic renal insufficiency defined as chronically elevated serum creatinine >177 μmol/L or chronic dialysis • Chronic restrictive, obstructive, or vascular pulmonary disease leading to severe functional impairment • Chronic liver failure with portal hypertension (with positive liver biopsy) and/or upper gastrointestinal bleeding due to portal hypertension and/or episode of hepatic encephalopathy/coma due to medical history of liver failure
APACHE admission diagnosis category	Admission to a medical/surgical tertiary ICU

Table 4.1: Table with all the baseline predictors.

Variable name	Variable description
Heart rate	Median of 1-hour mean heart rate (bpm)
Blood pressure	Median of 1-hour mean blood pressure, either invasive mean arterial blood pressure measurement or non-invasive cuff (mmHg)
Oxygen saturation	Median of 1-hour mean oxygen saturation (%)
Respiratory rate	Median of 1-hour mean respiratory rate (rpm)
Pulse	Median of 1-hour mean pulse pressure (difference between systolic and diastolic blood pressure, mmHg)
Invasive mechanical ventilation	Last observed mechanical ventilation status
FiO2	Last observed FiO2 (inspired oxygen concentration) value in 8 hours
Chronic corticosteroid use	Chronic medication use: systemic corticosteroids
Fever	Presence of fever in last 8 hours (>38 degrees Celsius)
Fluid balance	Fluid balance (mL) over the past 8 hours
Urine output	Total urine output (mL) in 8-hour window
Suctioned sputum	Total number of times sputum was suctioned and observed within an 8-hour time window
Worsening CNS status	Either decrease in consciousness (either a decrease in GSC M-score or worsening RASS score) or onset of new delirium episode in the past 8 hours
CRP (last value)	Last observed CRP (mg/L)
CRP (change)	Unit change in CRP relative to CRP 24 hours earlier (mg/L)
White blood cell count (last value)	Last observed white blood cell count (x10^9/L)
White blood cell count (change)	Unit change in white blood cell (WBC) count relative to WBC hours earlier (x10^9/L)
Platelet count (last value)	Last observed platelet count (x10^9/L)
Platelet count (change)	Unit change in platelet count relative to platelet count 24 hours earlier (x10^9/L)
Prothrombin time (last value)	Last observed prothrombin time (seconds)
Creatinine (last value)	Last observed creatinine (µmol/L)
Creatinine (change)	Unit change in creatinine relative to creatinine 24 hours earlier (µmol/L)
Total bilirubin (last value)	Last observed total bilirubin (µmol/L)
Total bilirubin (change)	Unit change in total bilirubin relative to bilirubin 24 hours earlier (µmol/L)
Bicarbonate (change)	Unit change of bicarbonate relative to bicarbonate 24 hours earlier (mmol/L)
Lactate (last value)	Last observed lactate (mmol/L)
Increase in vasopressor rate	Increase in mean norepinephrine dose relative to previous 8-h window
Increase in insulin dose	Increase in mean insulin dose relative to previous 8-h window
Gram+ in respiratory culture	Gram-positive bacteria cultured in the airway (the result of the most recent culture)
Candida in respiratory culture	Candida species cultured in the airway (the result of the most recent culture)

Table 4.2: Table with all the time-dependent predictors

The advantage that a landmark model (Van Houwelingen, 2007; van Houwelingen and Putter, 2011) offers is the possibility of making flexible and interpretable dynamical predictions when dealing with a huge amount of data. Landmarking model, therefore, allows us to estimate the survival probabilities at a precise *time-horizon* (i.e., the time where we want to make a prediction) given the observations at one precise *landmark time* (i.e., the time of the observations that one uses to make predictions at the time-horizon); the core of this approach consists of fitting a supermodel, i.e., a Cox model fitted on a large dataset containing the observations of ICU patients that are still at risk of acquiring an infection at some desired landmark times (different moments of ICU stay). We shall discuss it in detail in section 4.3.6.

4.2.2 Study population

The population we considered consists of 5075 ICU admissions in 4444 critically ill, amounting to 43143 observation days for model development. We collected 1197 cases of suspected ICU-acquired infections in 954 (18.8%) admissions. The absolute risk (i.e., the incidence) of ICUAI remains relatively constant during the stay of any sample patient in the ICU; we observed a mean risk of 0.04 with a standard deviation of 0.01 over the first ten days; see figure 4.2a. The visualization of the box plot for the times at which the infectious episodes were observed reveals that the median time lies at day 8.66, with an inter-quantile range between days 4.66 and 16; see figure 4.2d. When considering infectious episodes (920 episodes) only, the median time lies at day 6.33, with an inter-quantile range between days 4 and 10; see figure 4.2b. 869 cases of 1172 episodes (74.2%) with suspected infection met the criteria of *sepsis-3* definitions (Singer et al., 2016). In 10.8% of these episodes (94 cases), the patient suffered from septic shock. Patients with at least one ICUAI episode appear to have a significantly prolonged ICU stay with respect to patients who never had suspected ICU-AI episodes. Specifically, ICU patients suspected to develop an acquired infection reveal a median stay of 16 days (inter-quantile range 10-28 days), while patients who were never suspected of developing an ICUAI episode stay in the ICU with a median time of 5 days within the inter-quantile range of 3-8 days; see figure 4.2c.

4.3 Two-step modeling

4.3.1 High-Frequency covariates

With the term High-Frequency covariates (or even High-Frequency information), we refer to 5 EHR that we used in this work. The EHR we considered are Heart Rate, Mean Arterial Blood Pressure, Pulse, SaO2, and Respiratory Rate; all these predictors are sampled with a sampling rate equal to one minute and are arranged like a Time-Series.

Starting from the admission time, we divided all physiological vital signals (i.e., EHR) in a sequence of overlapping time windows with an amplitude of 24 or 48 hours. In other words, starting from the admission time τ_0, we selected the first 24-hour (or 48-hour) time window (i.e., that time window that encloses the EHR whose edge points are $(\tau_0, \tau_0$+24 hours)). Next, we selected all other time windows that are generated by recursively applying a forward displace of 8 hours on the first time window, i.e., we collected the windows with edges at (τ_0+24 hours), (τ_0+32 hours), (τ_0+32 hours, τ_0+40 hours), and so on. This strategy allows chunking the longitudinal evolution of physiological vital signals coherently with the way we extracted the low-frequency time-dependent

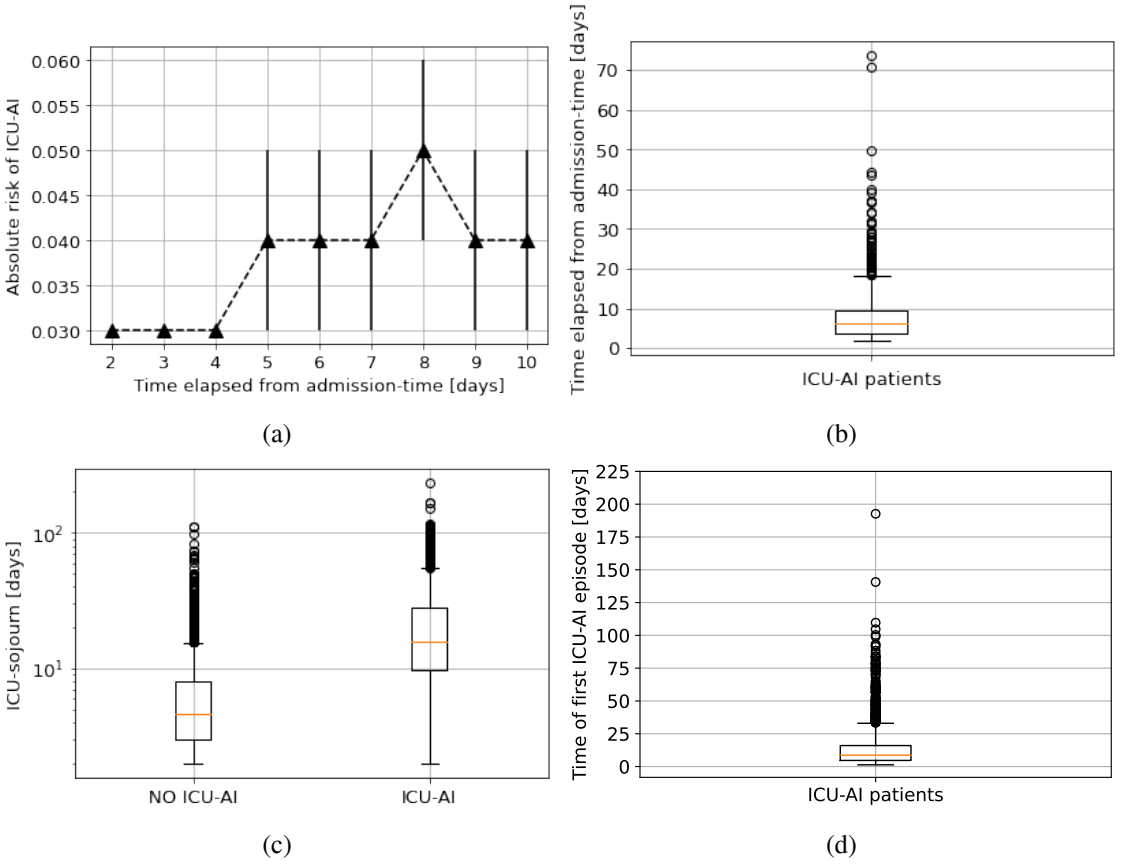


Figure 4.2: Plots with some characteristics of the study population: (a) the absolute risk of suspected ICU-acquired infection, (b) box-plot for the time at which the first infectious episode occurs, and (c) the box plots of the ICU stay of both classes of patients who experienced or not at least one ICU-acquired infection episode, and (d) the box-plot for the time at which one infectious episode occurs.

covariates. We stress the point that the vital signals we extracted within one 24-hour time window represent the collection of five vital signals evolving during one day of therapy; we shall refer to this kind of datum as one *Time-Series instance*; see example in figure 4.3. Time-Series instances represent grid-structured data (see 1.1.3), namely, the ideal type of data structure to be propagated through ANN models. So, in summary, we selected and extracted the Time-Series instances as follows:

1. We first removed the last 24 hours of records for all patients who passed away during the therapy because the clinical deterioration occurring in these cases might represent a bias or can eventually confound the ANN model during the learning phase.

2. Per each patient who does not acquire infection in the ICU, we divided his/her complete stay in a sequence of overlapping time windows with an amplitude of 24 hours. Each one of those time windows is denoted as one *not-infected* instance.

3. Per each patient who acquired infection during his stay in the ICU, we chunked his/her complete stay like in the non-infected case, but we label as *infected* only all time-windows where an ICUAI event occurred (i.e., if that time-windows includes the time-stamp at which the ICUAI episode have been recorded). In addition, we labeled as infected all time windows whose next 24 hours will enclose the ICUAI episode that emerged at the previous point. Consequently, all the remaining time windows were treated as not-infected. In case of multiple infections, we, however, had to censor the next 48 hours of therapy after the ICUAI episode was recorded; after that period of 48 hours, we then readmitted that patient to the cohort with a new identity.

4. When a patient who acquired a nosocomial infection was readmitted after being temporarily censored, we selected and extracted the remaining part of the vital signals as introduced before.

In addition, we equipped each Time-Series instance with an extra Time-Series, monitoring the presence of missing values. In practice, one Time-Series that points out the fraction of missing records among the five variables under consideration at each time stamp. Hence, each Time-Series instance can be described by a 6×1440 matrix, whose rows represent the number of *Time-Series features* (i.e., Heart rate, Arterial blood pressure, Pulse, SaO2, Breath Rate and Missing records) and the columns of the time domain. The illustration of one sample Time-Series instance is shown in Fig .4.3.

Missing values of EHR were imputed by using a zero-order spline, i.e., the Last Occurrence Carried Forward (LOCF) method. The reason why we opted for the LOCF method is motivated by the absence of a physical constraint about the temporal evolution of EHR. That is, we opted for the most practical and easy solution. However, we recall that each instance is equipped with a Time-Series that take trace of the actual presence of missing values; the inclusion of such a Time-Series help ANN to recognize the correct informativeness of flat patterns, that is, whether a flat pattern is due to the LOCF method or not. In figure 4.4, the occurrences of the Time-Series instances (before and after applying the LOCR imputation method) are shown; a descriptive statistic of both scenarios (before and after LOCR) is also shown in Table 4.3. As one can see, the imputation via LOCR does not distort the overall statistics of EHR, since most of the statistical estimators are subject to a relative change equal to 1-3%.

4.3.2 ANN for the analysis of High-Frequency data

To analyze the 1-minute physiological vital signals we opted for a 1-D CNN model. More specifically, we chose a pure convolutional network; the architecture of this ANN is composed of convolutional, pooling, and dense layers only. Such a choice has been drawn after comparing the accuracy of either traditional or ANN-based models in classifying ICUAI episodes from the EHR of the subjects; see section 4.3.3. In particular, we considered methods such as the LR model, linear Supported Vector Machine, MLP, 1-D CNN, and 1-D CNN-LSTM. The last is a particular type of ANN whose architecture is identical to 1-D CNN, but an LSTM layer (see section 1.1.4) replaces the *flatten layer*. When testing the accuracy of these models, we evaluate the AUROC score (often denoted as AUC, i.e., Area Under the Curve). The evaluation of the AUROC score is always performed by means of a 5-fold cross-validation, that is, one evaluates the AUROC of the model per

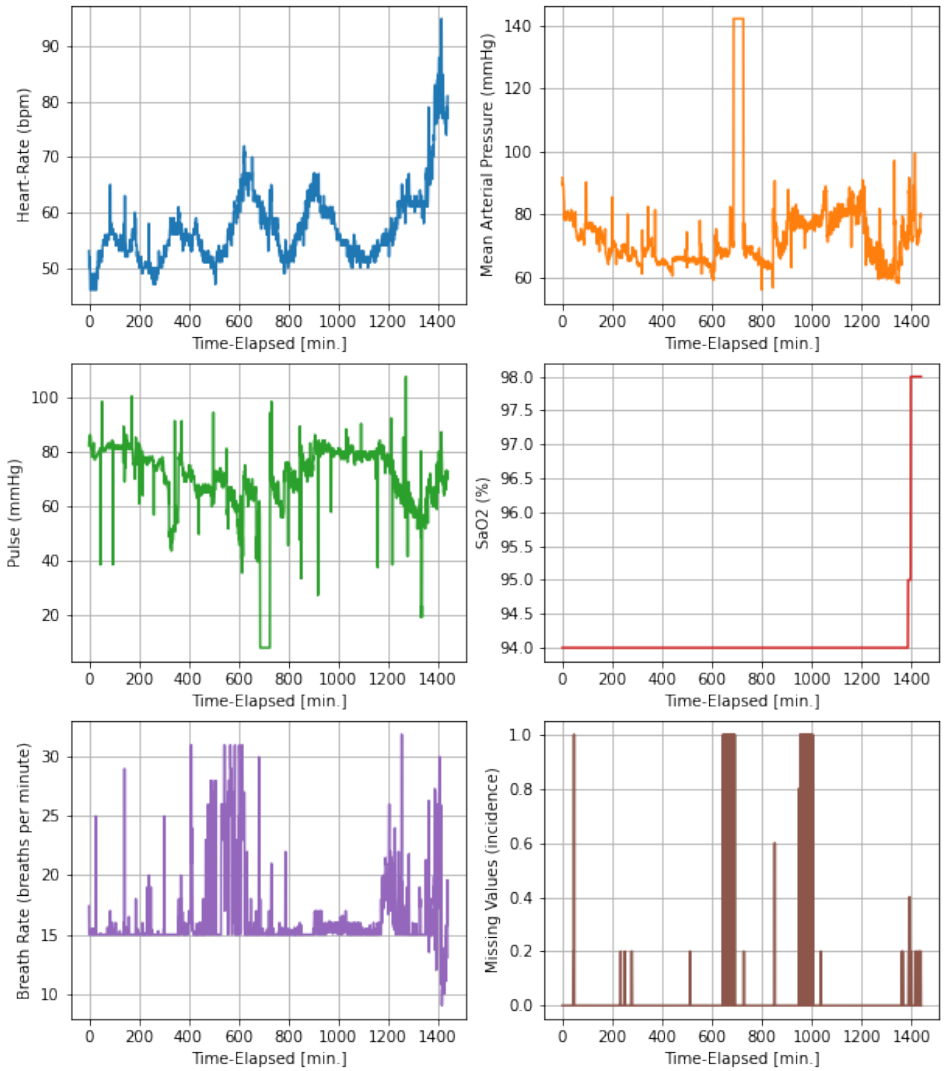
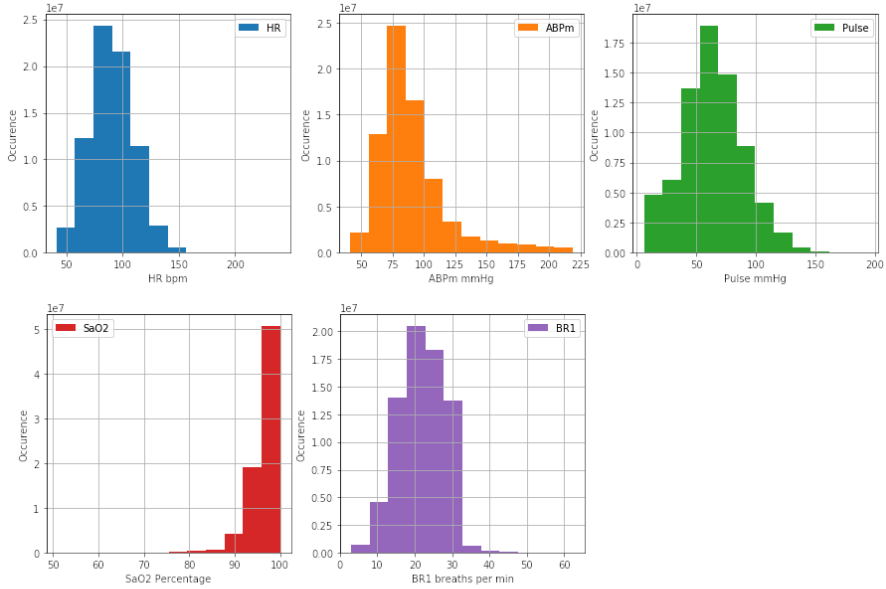
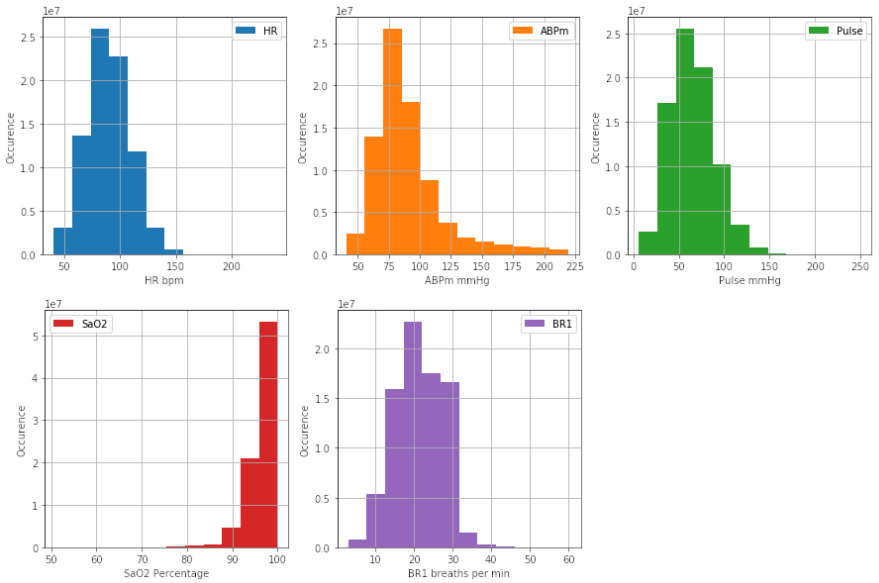


Figure 4.3: Example of Time-Series instance. Each subplot represents one single Time-Series feature (i.e., one vital signal). x-axis: Time-domain (24 hours). y-axis: the values taken by each Time-Series feature. Specifically, heart rate in blue, arterial blood pressure in orange, pulse in green, saturation (SaO_2) in red, breath rate in purple, and the auxiliary Time-Series (with the missing values incidence) in brown.



(a)



(b)

Figure 4.4: Occurrences for all the sampled records of the EHR values; (a) raw data (before correcting the missing values) and (b) after imputing missing values with a zero-order spline. Specifically, heart rate in blue, arterial blood pressure in orange, pulse in green, saturation (SaO_2) in red, and breath rate in purple.

Heath Rate	Before Imputation (Beats/min)	After Imputation (Beats/min)	Relative Error
Mean	92.145	92.049	0.001
Standard Deviation	19.831	19.796	0.002
Median	91.0	91.0	0.001
Min	41.0	41.0	0
Max	239.0	239.0	0
Skewness	0.313	0.299	0.005
Kurtosis	3.215	3.156	0.018
Number of samples	76538986	80829572	0.053
Mean Arterial Pressure	Before Imputation (mmHg)	After Imputation (mmHg)	Relative Error
Mean	83.264	83.148	0.001
Standard Deviation	16.452	16.347	0.006
Median	80.716	80.716	0.001
Min	41.036	41.036	0
Max	219.087	219.087	0
Skewness	1.186	1.171	0.013
Kurtosis	6.896	6.797	0.014
Number of samples	74040855	80829572	0.084
Pulse	Before Imputation (mmHg)	After Imputation (mmHg)	Relative Error
Mean	64.827	64.231	0.001
Standard Deviation	24.682	24.536	0.003
Median	62.671	61.964	0.001
Min	6.021	6.021	0
Max	249.834	249.834	0
Skewness	0.512	0.541	0.011
Kurtosis	3.071	3.111	0.081
Number of samples	74328386	80829572	0.081
SaO2	Before Imputation (%)	After Imputation (%)	Relative Error
Mean	96.266	96.378	0.001
Standard Deviation	3.599	3.315	0.079
Median	97.0	97.0	0.001
Min	51.0	51.0	0
Max	99.0	99.0	0
Skewness	-3.422	-3.279	0.047
Kurtosis	23.292	23.309	0.001
Number of samples	76661931	80829572	0.052
Breath Rate	Before Imputation (Breath/min)	After Imputation (Breath/min)	Relative Error
Mean	20.882	20.979	0.001
Standard Deviation	24.53	8.402	0.006
Median	19.96	19.951	0.001
Min	1.02	1.02	0
Max	89.01	89.01	0
Skewness	1.592	1.762	0.013
Kurtosis	9.247	10.151	0.014
Number of samples	73371367	80829572	0.088

Table 4.3: Overall descriptive statistics for raw EHR and EHR imputed via LOCF.

each fold, and then the overall measure of accuracy with its error is given by the mean value and the SEM, respectively.

Although 1-D convolutional layers represent the most natural choice to analyze Time-Series data, a 2-D convolutional-based approach is always possible if one provides a 2-D representation of sequential data. For example, the method developed by Ye et al. (2019) offers the possibility to give a 2-D representation of Time Series data, i.e. a 2-D binning is performed, where each 2-D bin counts the number of records falling in a specific range of values and at some precise moments along the time domain; see the example in figure 4.5. A 2-D grid-structured representation of the EHR enabled us to investigate the possibility of using a 2-D CNN model to early identify the onset of ICUAI. After selecting the best architecture of the 1-D CNN (see section 1.1.3) model, we also tested whether the LOCF method affected the pattern recognition activity of the 1-D CNN model. In particular, we made a comparison between two methods: we compared the pattern recognition activity of the 1-D CNN model after imputing the data via either LOCF or another imputation method which imputes the missing values depending on the length of the interval of missing records. In the latter, the missing values of an interval of missing records with a length larger than 4 hours are replaced with an out-of-range value; for intervals of missing values with a length lower than 4 hours, all missing values are imputed with a null constant value. We shall refer here to this imputation as *ICUAI-imputation* method. The alternative proposed by the ICUAI-imputation method mirrors the pragmatic medical point of view of dealing with EHR. A time scale larger than 4 hours usually represents the typical duration of surgical operations, while shorter time scales are usually related to the usual causes of interruptions of the ICU monitoring, e.g., the detaching of one or more devices for shorter periods (either unconscionably by one patient or because of a malpositioning of the devices themselves).

4.3.3 ANN model selection

When testing the level of accuracy of both the LR and the SVM model, and MLP, we needed to aggregate Time-Series in order to make the structure of the data suitable for the correct employment of these ML techniques. Thus, we extracted some traditional statistics (i.e., mean value, standard deviation, skewness, kurtosis, minimum, and maximum value) of the features of the Time-Series instance. We recall that each Time-Series instance represents a collection of physiological vital signals evaluated in a time window of 24 hours (or 48 hours); we evaluated these statistics on each physiological vital signal. This way, we obtained a total of 31 input features. It is important to mention that the input features extracted in this way are subjected to a standardization, i.e., each feature is linearly transformed in order to set the mean value and the standard deviation equal to zero and one, respectively.

The LR and the SVM models are penalized with a ridge regularization. We used the inverse of the shrinkage parameter (here denoted as C) as the unique hyper-parameter of both the models; We searched for the best C that optimizes the AUROC score; see figure 4.6. In this case, we observed that both models cannot achieve an AUROC score larger than 0.59. Such a result is present for both the 24-hour and the 48-hour prediction models.

For the MLP (see section 1.1.2), we know that the best accuracy of the model is intimately connected with the search for the best set of hyperparameters. Unlike the previous case, we did not have to deal with one single hyper-parameter, but we needed to consider a few more parameters, such as the *number of units, deepness, dropout rate, learning rate, activation function,* and, the *batch*

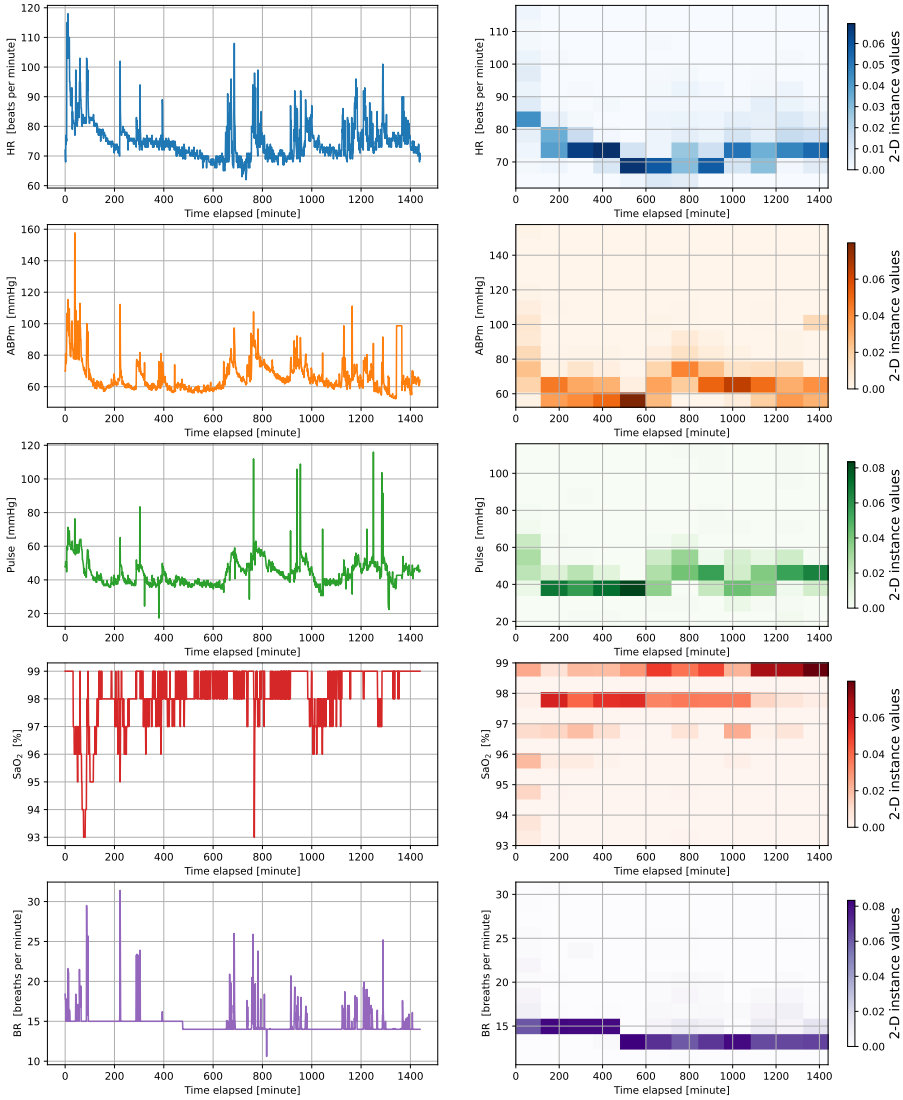
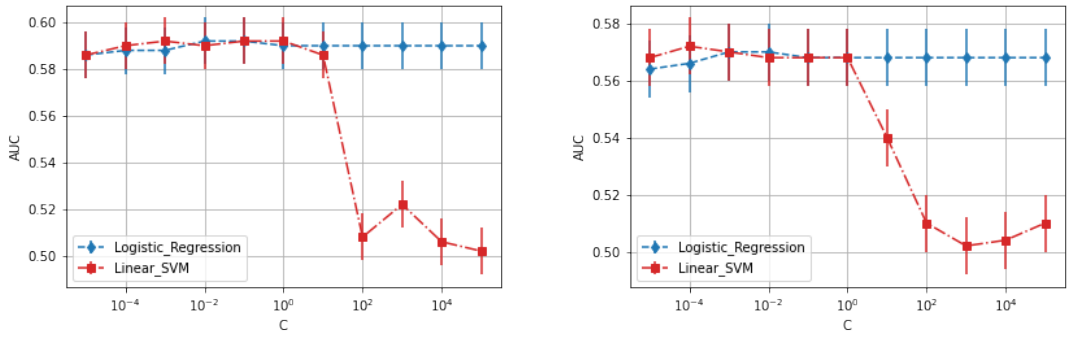


Figure 4.5: Example of 2-D representation of a Time-Series instance. On the left column the Time-Series features (EHR), while on the right columns their 2-D representation.



(a) (b)

Figure 4.6: AUROC as a function of the inverse shrinkage parameter for (a) the 24-hour and (b) the 48-hour prediction model. The red curve concerns the SVM model, while with the blue line, the LR model is represented. The error bars refer to the Standard Error mean.

size (see section 1.1.2, and 1.2). Therefore, tuning these hyperparameters consisted of searching for the set of parameters that maximize the AUROC score. Such a search was performed over a fine grid of hyperparameters. In this case, the MLP model was designed to optimize the Binary Cross-Entropy by means of the ADAM (Kingma and Ba, 2014a) optimizer. Note that the depiction of the curve of the AUROC score as a function of the hyperparameters is quite impractical because of the large number of parameters we had to consider. Anyway, we regarded all the configurations that meet one precise constraint (e.g. we consider all configurations with deepness equal to 3 or a number of units equal to 16) and there we selected that one with the highest AUROC score. We shall refer to this as the *maximal AUROC* (of a configuration). The representation of all these maximal AUROC scores should help visualize the change of the maximal AUROC with respect to one single hyper-parameter. In figure 4.7a and 4.7b are shown MLP models characterized by a ReLU activation function (i.e., $\text{ReLU}(x) = \max(0, x)$); as one can see, either the augmentation of the number of units per layer or more deep architecture does not enable the model to achieve an AUROC score higher than 0.63. Likewise, the choice of a hyperbolic tangent (tanh) activation function would present a similar result; see figure 4.7c and 4.7d.

The next model that we analyzed is the 1-D CNN (see section 1.1.3). Similarly to the MLP, we needed to optimize a fewer number of hyper-parameters, i.e. the *number of convolutional filters, kernel size, deepness, learning rate*, and *the batch size*. In this case, the activation function has not been included in the hyperparameters to tune; unlike the MLP model, we only considered the activation function ReLU. We motivate this choice after noting that several tests with a *one held-out* approach revealed that sigmoidal activation functions (e.g., sigmoid or hyperbolic tangent) affected the predictiveness of the model; one obtained AUROC always lower than 0.60 for different combinations of *power* (i.e., the number of filters $times$ dropout rate), *deepness* (i.e., number of hidden layers), and *receptive field* (i.e., the combination of kernel and max-pooling layers of different size).

Before propagating the vital signs through the 1-D CNN model, a few pre-processing steps must be executed. Firstly, one linear transformation was applied to all the instances to give a compact representation in the range [-1, 1]. Note that we applied the same-type transformation to all instances; according to the Time-Series feature to rescale, a precise linear transformation was applied. For

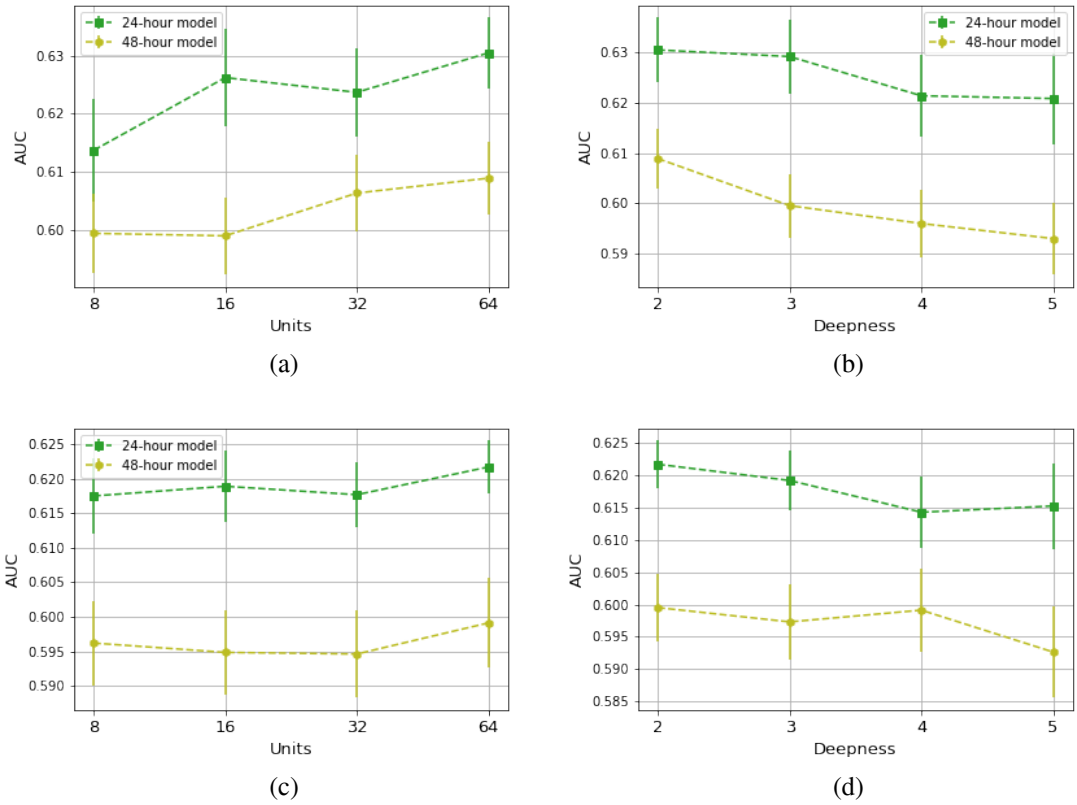


Figure 4.7: MLP model. Maximal AUROC as a function of the hyper-parameters *Units* (a.k.a, number of units) and *Deepness*. Each plot presents the behavior of both the 24-hour (green line) and 48-hour models (yellow line). The following cases are considered: (a) Number of units and ReLU activation function, (b) Deepness and ReLU activation function, (c) Number of units and tanh activation function, (d) Deepness and tanh activation function.

example, we used the same linear transformation to rescale all Heart Rate signals contained in all instances; but for all Breath Rate signals, we developed and used a different one. Specifically, these linear transformations are based on the descriptive statistics of Table 4.3; so for each Time-Series feature, we constructed a linear map that rescales both maximum and minimum values to 1 and -1, respectively. For example, if we consider the Heart Rate predictor in the Time-Series instances, from Table 4.3, we know that the minimum and the maximum value are 41 and 239 beats per minute, respectively. Accordingly, if we denote with $X_i^{HR}(t)$ the Heart Rate feature of the i-th Time Series instance, the transformation we shall apply is

$$X_i^{HR}(t) \to \frac{2X_i^{HR}(t) - 41\text{bpm} - 239\text{bpm}}{239\text{bpm} - 41\text{bpm}}.$$

Unlike standardization (i.e., one imposes that all time-series features have unitary variance and zero mean value), the application of these data-based linear transformations does not drastically distort proper characteristics of the vital signals such as scale (i.e., the mean) and energy (i.e., the

empirical second moment) value. In addition, data were processed using the Piecewise Approximate Aggregation (PPA) method. Instead of representing all very-high-scale details, we obtained a reduced but informative representation of vital signals while maintaining the lower bound of distance measurements in Euclidean Space (Chen and Qi, 2019). Therefore, we used PPA to aggregate time intervals of 9 minutes.

Similarly to the MLP case, the visualization of the maximal AUROC scores can help describe how a change of one hyperparameter can affect the performance of the 1-D CNN model. In figure 4.8a, one can see that the more the number of filters, the finer the capturing activity of the interesting patterns within the hidden layers. A high number of filters, such as 128, make both 24-hour and 48-hour models accurate with AUROC 0.72 and 0.67, respectively. The composition of many hidden layers is another key feature of enabling the model to be performative. In figure 4.8b, either shallow or deep architectures ensure good predictions for the 24-hour model (AUROC in the range [0.70, 0.72]), whereas 6 convolutional layers are needed to enable the 48-hour model to achieve the highest AUROC equal to 0.67. Conversely, the amplitude of the convolutional masks reduces the AUROC values, especially if one considers masks of size 17 or 33; see figure 4.8c. Convolutional masks of size 3 enable keeping the AUROC 0.72 and 0.67 for both the 24-hour and 48-hour models. Thus, after searching for the best configuration, our investigation revealed that powerful (i.e., with many filters) and deep networks with small-sized kernels are the type of 1-D CNN models to employ.

For completeness, we compared a pure convolutional approach (i.e., 1-D CNN model) with a 1-D CNN-LSTM model. As mentioned above, the latter has precisely the same architecture as the 1D-CNN model, except for the fact that an LSTM layer (see section 1.1.4) replaces the *Flattern Layer* of the 1-D CNN model (see section 1.1.3). The LSTM layer possesses only one relevant hyper-parameter, i.e., the *number of units* denoting the number of items used to encode the impute data. Thus, we considered a 1-D CNN-LSTM with the best hyperparameters found for the optimization of the 1-D CNN model. Still, the parameter "number of units" is the unique variable assuming different values. In figure 4.9, one can see that an increasing change in the number of units does not apport a significant augmentation of the AUROC score. For the 24-hour model, the plateau region starting at unit 64 reveals that the CNN-LSTM model is as accurate as the 1-D CNN model, i.e., AUROC score equal to 0.72 ± 0.01. The 48-hour model cannot achieve AUROC values larger than 0.6, given any configuration of the LSTM units.

The last class of models that we tested is the 2-D CNN. Despite sharing a similar structure with the 1-D CNN (see section 1.1.3), the implementation of 2-D CNN required tuning a larger number of hyper-parameters. Such an increase in hyper-parameters is mainly due to the 2-D structure of data; unlike the 1-D case, in the 2-D case, both the width and the height of the convolutional masks need to be optimized (see section 1.1.3) as well as the height and the width of the 2-D bins representing each Time-Series feature. As with the 1-D CNN model, we optimally tuned the model on a fine grid of parameters: *number of filters, kernel size (on both the 2 dimensions), deepness, dropout, learning rate, batch size*, and both *width and height of the 2-D bins*. In figure 4.10, one can see that drastic changes in the architecture of the 2-D CNN model do not cause relevant changes in the evaluation of the maximal AUROC score. That is, opting for several configurations in the number of filters (see figure 4.10a), in the deepness (see figure 4.10b), in the size of the convolutional masks (figure 4.10c), and in the height and width of the 2-D bins (figure 4.10d) do not lead both the 24-hour and the 48-hour models to achieve AUROC scores larger than 0.63.

Among all the models considered so far, the 1-D CNN model did not turn out to be the model with the highest predictive performance; but we also chose it to model the risk score of infection

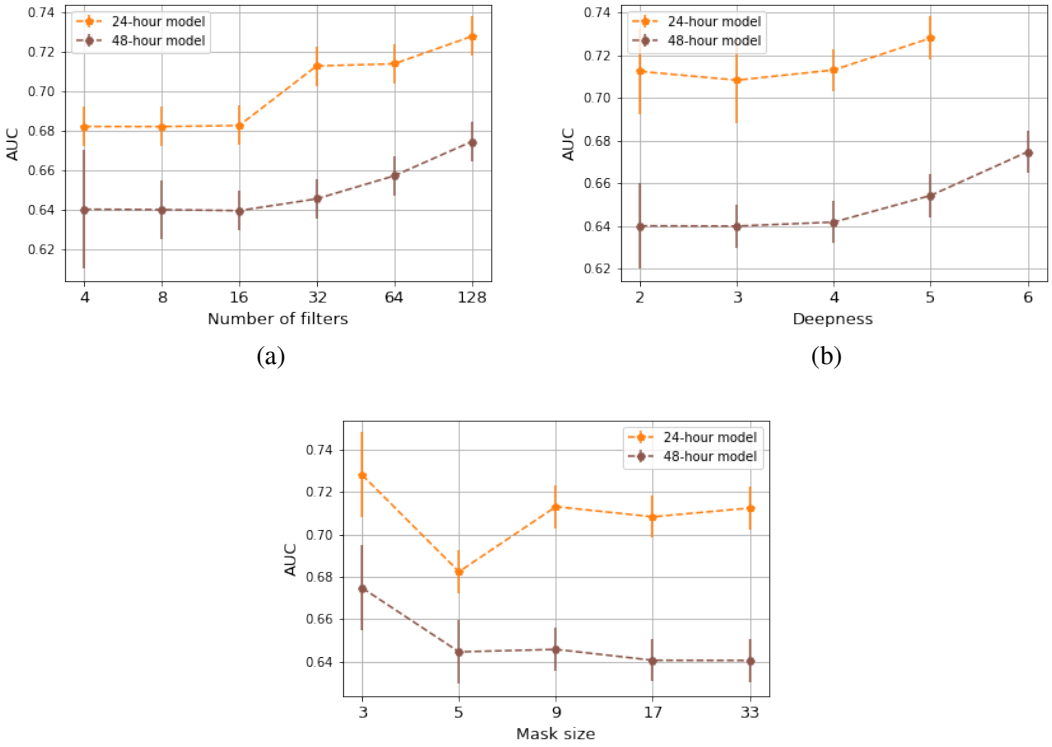


Figure 4.8: 1-D CNN model. Maximal AUROC as a function of the hyper-parameters (a) *number of filters*, (b) *deepness*, and (c) *kernel size*. Each plot presents the behavior of both the 24-hour and 48-hour models.

for both the 24-hour and the 48-hour models. Although the 24-hour CNN-LSTM model could be slightly more accurate than the 1-D CNN, we observed that the latter showed more precise predictive performances even with 48-hour Time-Series instances. The difference in terms of AUROC between both models is marginal for the 24-hour model but instead evident for the 48-hours models. In contrast, the 1-D CNN model showed good predictive power in both contexts. This is why we opted for the 1-D CNN. Moreover, we opted for a 1-D CNN (see section 1.1.3) model also because we want to explain the activity of pattern recognition via a robust XAI method such as the SMOE scale (see section 1.3.2); this method is designed to work with CNN models only (we shall discuss it in section 4.6.1).

For the 24-hour model, we propose the following optimal architecture:

- *Convolutional Layers*: the number of filters on each layer is 128, and each filter has a size of 3 (pixels). The result of these convolutions is referred to as *feature maps*.

- *Activation Layer*: the ReLU function is applied after each convolution operator. This application of a non-linear activation function on the feature maps gives birth to the *activated feature maps*.

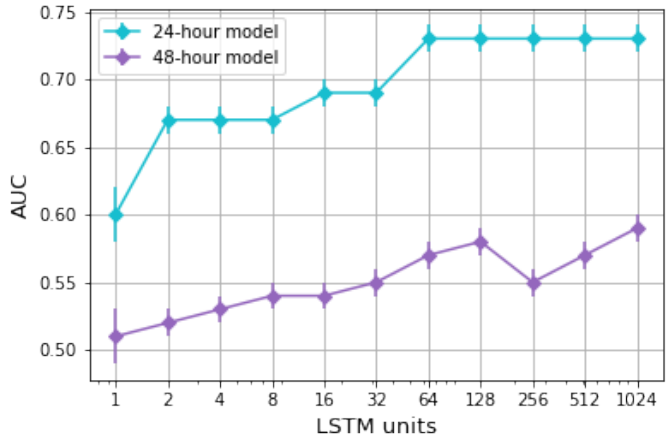


Figure 4.9: 1-D CNN LSTM model. AUROC as a function of the number of units of the LSTM layer. The choice of all other hyperparameters of the 1-D CNN-LSTM model is identical to the ones of the optimal 1-D CNN model. The error bars are computed via SEM.

- *Max-pooling layer*: the activated feature maps are resampled via a Max-pooling operator with a pooling size of 2.

This sequence of hidden layers is repeated five times. The architecture also encloses a *Dropout layer* after each Max-Pooling layer. The Dropout layer has a dropout rate of 0.25. The last feature map is flattened into an array and then propagated in a *fully-connected* layer (a.k.a., Dense layer) with a sigmoid activation function. The activation function returns a positive output between 0 and 1, that is, the risk score denoting the chance of a patient developing an ICUAI episode. As usual, the loss function is the binary cross-entropy, and the optimizer is the ADAM algorithm. For the 48-hour model, the architecture is identical to the 24-hour one, except for the fact that the sequence of convolutional and max-poling layers is repeated 6 times.

When testing the effects of a different imputation of the EHR on the performance of the 1-D CNN, the results point out that the alternative proposed to the LOCF can really worsen the performance of the 1-D CNN. In figure 4.11, one can see that the ICUAI-Imputation method decreases the AUROC score at values lower than 0.6; we tuned several configurations for the number of filters (figure 4.11a), deepness (figure 4.11b), and size of the convolutional masks (figure 4.11c). These results, therefore, confirm that an LOCF imputation method results to be much more robust and performative with respect to the ICUAI-Imputation.

4.3.4 Low-frequency covariates

All time-dependent data have been prepared using a landmarking approach, i.e., one chunks the longitudinal evolution of time-dependent covariates time into equally spaced intervals (i.e., landmarking whose amplitude is equal to 8 hours) The most recent predictor variable value at each landmarking time point has been used to predict the probability of infection. We opted for 8-hour intervals to mirror the typical time period between consecutive measurements and the assessment of time-dependent variables. Thus, landmarking time points were thus set at 8-hour intervals starting

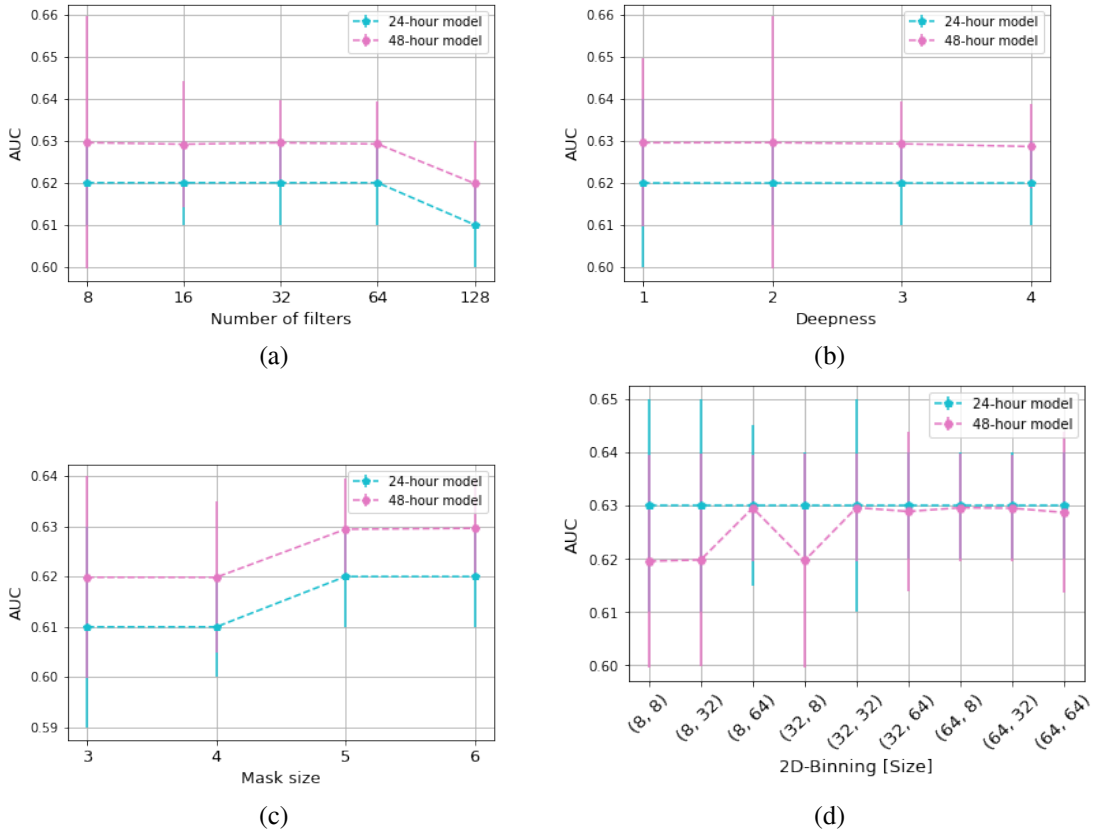


Figure 4.10: 2-D CNN model. Maximal AUROC as a function of the hyper-parameters *number of filters* (a), *deepness* (b), *kernel size* (c), and the *dimensionality of the 2-D bins* (d). Each plot presents the behavior of both the 24-hour (cyan) and 48-hour (pink) models.

from 48 hours after ICU admission.

Whenever one ICUAI event occurs, we censor the 48-hour interval that immediately follows each timestamp of infection. After this recovery period, the patient who experienced the ICUAI event can be newly readmitted to the cohort but with a new identity and admission time. We opted for this strategy because we want to guarantee the presence in the cohort of those patients who might experience multiple acquired infections. Consequently, observations acquired at different intervals, but produced by the same patient, are also used to estimate the dynamic probabilities of infection. All observations extracted at different landmarking times were treated as independent observations.

The strategy we adopted to model recurrent acquired infectious episodes has already been proposed in the literature. For example, Musoro et al. (2018) used a landmark Cox proportional hazard model to predict recurrent infections after kidney transplant. From a modeling point of view, the censoring of one patient's history at the time of the event and his/her subsequent readmission as a new independent subject does not conciliate with the fact that the recurrent events of one patient are theoretically correlated by the fact to belong to the same patient's history. For completeness, we need to mention that when running the landmark competing risk Cox model, we also consider introducing a frailty term (a random effect for each individual) to control the correlation among

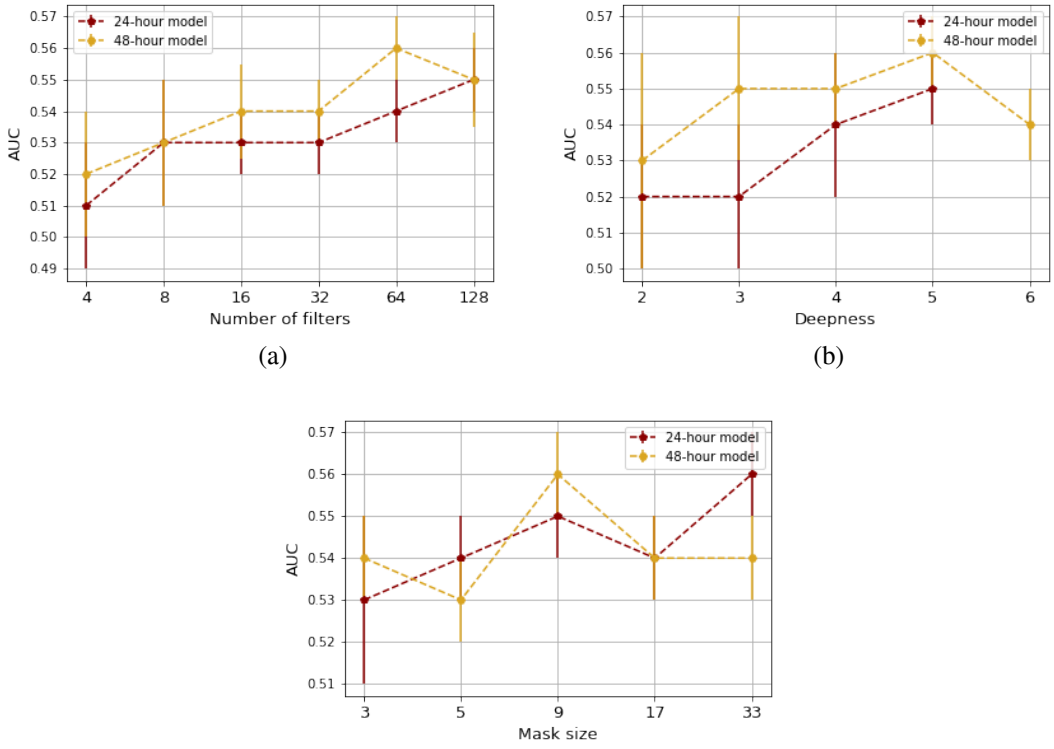


Figure 4.11: 1-D CNN model fed by EHR data imputed via the ICUAI-Imputation method. Maximal AUROC as a function of the hyper-parameters of the 1-D CNN model. (a) *number of filters*, (b)*deepness*, (c) *kernel size*. Each plot presents the behavior of both the 24-hour and 48-hour models.

recurrent infectious episodes. Using a frailty term did not reveal a substantial change in the hazard rate estimation. However, in terms of computational costs, it made expensive the estimation of the Cumulative Incidence Function of ICUAI episodes; theoretically, one cannot utilize the Aalen-Johansen estimator but should directly compute the Cumulative Incidence Function after simulating all patients' trajectories.

For time-varying predictors, missing values were observed in the range 1% to 10% of 8 hours observation windows; see Table 4.4. Missing values between measurements, including missing data points due to up-sampling, were imputed with a zero-order spline (i.e., LOCF). Instead, missing values occurring before the first actual measurement were imputed using multiple imputation procedures based on all other predictors and outcomes.

4.3.5 Dynamic Predictions

Dynamic predictions represent one of the most interesting aspects of CR models, especially when dealing with time-dependent covariates. In dynamic predictions, the main focus is the estimation of the probability of one desired event at some time t_{hor} (i.e., the horizon time) given the longitu-

Time-Varying variable	Missing values (incidence)
Heart rate (bpm)	0.00
Blood pressure	0.00
Pulse	0.00
Respiratory Rate	0.00
Oxygen saturation	0.00
Invasive mechanical ventilation	0.00
Fever	0.00
Urine output	0.00
Fluid balance	0.00
Worsening CNS status	0.00
FiO2	0.00
Suctioned sputum	0.00
CRP (last value)	0.01
CRP (change)	0.01
White blood cell count (last value)	0.00
White blood cell count (change	0.10
Platelet count (last value)	0.00
Platelet (change)	0.01
Prothrombin time (last value)	0.00
Creatinine (last value)	0.00
Creatinine (change)	0.03
Total bilirubin (last value)	0.00
Bicarbonate (change)	0.10
Lactate (last value)	0.10
Gram+ in respiratory culture	0.00
Candida in respiratory culture	0.00
Increase in insulin dose	0.00
Increase in vasopressor rate	0.00

Table 4.4: Missing values (incidence) of the time-dependent variables.

dinal observations of some subjects as some fixed time s (often denoted as $\mathbf{Z}(s)$). Such estimation is therefore based on the individual event history. When applied to healthcare studies, dynamic predictions often aim to see if particular clinical strategies (e.g., some time-dependent clinical covariates expressing a patient's therapy) can establish a dependence on the probability that one of competing risk to occur (e.g., the amelioration of the worsening of one patient's prognosis) within a specific interval of time.

Due to their versatility, dynamic predictions have found in the last decades many interesting developments in both theoretical and applied contexts (van Houwelingen and Putter, 2011). The estimation of the dynamical prediction probability can be made in several ways; a famous one is the joint-modeling approach (Proust-Lima and Taylor, 2009; Rizopoulos, 2011, 2012). According to (Ibrahim et al., 2010), the idea behind joint modeling is based on two basic components:

1. **The longitudinal component:** it consists of fitting a linear model with random effects on all the subjects' trajectory; the model has the structure

$$Y_{ij} = X_{ij} + \varepsilon_{ij}; \tag{4.1}$$

with Y_{ij} the i-th outcome at j-th time stamp, X_{ij} the trajectory of the i-th patient at the j-th time stamp, and ε_{ij} a multinormal uncorrelated gaussian noise for the i-th patient at the j-th time-stamp. The single trajectory X_{ij} is assumed to have form

$$X_{ij} = \theta_{0i} + \theta_{1i} t_{ij} + \gamma Z_i; \tag{4.2}$$

with θ_{0i} + θ_{1i} some random variable and multivariate normal distributed, t_{ij} is a linear function of time, Z_i one covariate for the i-th subject, and γ a coefficient assessing the effect of the k-th covariate.

2. **The survival component**: the evaluation of the survival probability is made after modeling the hazard rate via a Cox model involving the patients' trajectory, namely

$$\lambda(t_{ij}) = \lambda_0(t_{ij}) \exp\big(\alpha X_{ij} + \beta Z_i\big); \tag{4.3}$$

with λ the hazard rate, λ_0 the baseline hazard rate, α the association between the trajectory of one subject and its time to event, and β is the direct effect of the k-th covariate on the time-to-event.

Note that we only considered the case with one covariate; a complete discussion can be found in (Rizopoulos, 2012). Although the joint-modeling approach offers a more robust way to estimate the dynamical prediction probabilities, the theoretical assumptions one must meet might often make applying this method impractical when dealing with vast amounts of data. The presence of shared random effects and multiple submodels might make the joint modeling complicated to fit on data (Furgal et al., 2019).

For this reason, in the last decade, much attention has been focused on the *landmark approach*. Landmark model (Van Houwelingen, 2007; van Houwelingen and Putter, 2011) represents a valid approach to dynamical predictions. The idea behind this method consists of dividing the explanatory variables' time domain into fixed landmark times. The term *landmark time* refers to one prefixed time point (i.e. "landmark"). When combined with a CR Cox model, the advantage offered by the landmark approach consists in fitting a Cox model on each landmark time; individual Cox regressors are then assumed to have a time dependency expressed via a cubic spline. The relatively simple structure of the landmark model enables one to accurately estimate the behavior on time of Cox regressors while avoiding a software implementation that turns out to be time-consuming, especially in the case of a big dataset leading to extensive runnings.

4.3.6 Landmark model for competing causes

In this subsection, we shall present the Landmark model Van Houwelingen (2007), but extended to J competing risk model (Nicolaie et al., 2013). The framework is always the CR models (see section D.1).

Suppose to have a cohort composed of N subjects, and we consider a CR model where each of the N subjects can experience one of J distinct failures (i.e., failing causes). We denote with $\tilde{T}$

the time of failure, C the censoring time, D the case of failure, and $\mathbf{Z}(\cdot)$ and array of covariates (we assume covariates might be either time-fixed or time-dependent, or eventually both). For each i-th subject, the tuple $(T_i, \Delta_i, \mathbf{Z}_i(\cdot))$ represents the observation at time $T_i = \min(\tilde{T}_i, C_i)$ (i.e., the earliest of failure and censoring time), the cause of failing $\Delta_i = \mathbf{1}(\tilde{T}_i < C_i)D_i$ (with $\mathbf{1}(\cdot)$ the indicator function; note that in case of censoring $\Delta_i = 0$), and $\mathbf{Z}_i(\cdot)$ the acquisition of the covariates up to time T_i. We stress that the subscript $i \in \{1, 2, \dots, N\}$. Likewise, we shall adopt the subscript j to refer to the competing causes with $j \in \{1, 2, \dots, J\}$.

In the landmark model, one aims to make a dynamic prediction (i.e., estimating the probability of one specific event at some time horizon, denoted as t_{hor}) for the i-th subject with respect to one of the J causes (e.g., $\Delta_i = j$); the estimation of the survival probability is conditional to the observations available at some fixed time t_{LM}, i.e., the *landmark time*. Thus, we aim to make a dynamic prediction at the time horizon

$$t_{hor} = t_{LM} + w; \tag{4.4}$$

that is, we aim to estimate the following two quantities

$$S_{LM}(t_{hor}|\mathbf{Z}(t_{LM}), t_{LM}) = \mathbf{P}(T > t_{hor}|\mathbf{Z}(t_{LM}), t_{LM}), \tag{4.5}$$

$$F_{j,LM}(t_{hor}|\mathbf{Z}(t_{LM}), t_{LM}) = \mathbf{P}(T \leq t_{hor}, \Delta = j|\mathbf{Z}(t_{LM}), t_{LM}). \tag{4.6}$$

Note that we denoted with w the prediction window or interval (i.e., the lead time).

The landmark approach consists now of two steps:

1. We first divide the time domain of our observations (e.g., $[s_0, s_1]$) into some equispaced landmark points denoted as t_{LM}; this operation is adaptive to the individual subject's history. The grid of selected landmark points will be applied to all subjects' histories, i.e., one establishes *a priori* a partition into landmark times which is valid for all subjects. Next, we fix the prediction window, i.e., w. The selection of w implies selecting the horizon times via (4.4). For each landmark time t_{LM}, we extract all the observations $\mathbf{Z}(t_{LM})$ of all patients that are still at risk at time t_{LM}. The observations are administratively truncated in the interval $[t_{LM}.\ t_{LM} + w]$. Finally, we vertically stack all these datasets to generate a *stacked dataset*.

2. The second step is dedicated to fitting a CR Cox model (see section D.1). We fit a *landmark supermodel* (Nicolaie, 2014) on the stacked dataset. The peculiarity of the *landmark supermodel* is the stratification per landmark time (i.e., a landmark competing risk Cox model stratified per landmark time); the hazard rates estimated at different landmark times are then connected via cubic spline. The final goal of the *landmark supermodel* is to make dynamical predictions with prediction windows w at any time within $[s_0, s_1]$.

We can now pass to formally introduce the notion of *landmark supermodel*. Firstly, we fit a Cox model to estimate the j-th cause-specific hazard rate λ_j at some prediction interval $[t_{LM}, t_{LM}+w]$, namely

$$\lambda_j(t|t_{LM}, \mathbf{Z}(t_{LM})) = \lambda_{0j}(t|t_{LM}) \exp\left[\beta_j^T(t_{LM})\mathbf{Z}(t_{LM})\right]; \tag{4.7}$$

with $\lambda_{0j}(t|t_{LM})$ denotes the unspecified baseline hazards and $\beta_j(t_{LM})$ the set of regressors specific for the j-th cause in within the interval interval $[t_{LM}, t_{LM}+w]$. Note that t in (4.7) is assumed to lay

in range $[t_{LM}, t_{LM} + w]$. Once again, we stress that one fits one model like (4.7) to the observation available at each landmark time t_{LM}. Next, we can use the estimation of $\lambda_j(t|t_{LM}, \mathbf{Z}(t_{LM}))$ to estimate both the survival and the cumulative incidence function, namely

$$S(t_{hor}|\mathbf{Z}(t_{LM}), t_{LM}) = \exp\left(-\int_{t_{LM}}^{t_{hor}} du\ \lambda(u|t_{LM}, \mathbf{Z}(t_{LM}))\right), \tag{4.8}$$

$$F_j(t_{hor}|\mathbf{Z}(t_{LM}), t_{LM}) = \int_{t_{LM}}^{t_{hor}} du\ S(u|\mathbf{Z}(t_{LM}))\lambda_j(u|t_{LM}, \mathbf{Z}(t_{LM})). \tag{4.9}$$

Once again, the estimation of both (4.8) and (4.9) referes to a single prediction interval, i.e. $[t_{LM}, t_{LM}+w]$. Thus, the fit of independent Cox models to each dataset extracted at each landmark time t_{LM} cannot help us to estimate the desired dynamic prediction probabilities within a more general interval $[s_0, s_1]$. Such an approach does not take into account the time-evolution of the observations $\mathbf{Z}(\cdot)$; i.e., one does not consider that $\lambda_j(t)$ is sensitive to the changes of $\mathbf{Z}(\cdot)$ along the time-domain. To solve this, however, one can suppose that $\lambda_j(t)$ (as well as the regressors $\beta_j(t)$) could not change drastically on time. This implies that we model the time-evolution of $\beta_j(t)$ by means of a smooth function, namely

$$\beta_j(s) = f_j(s, \beta_j); \tag{4.10}$$

with s a generic time value, $\beta_j = (\beta_{1j}, \dots, \beta_{Mj})$ a vector of regression parameter for M covariates and $f_\beta(\cdot)$ a parametric function on time, e.g., a cubic spline. Equation (4.10) provides a complete evolution on time of the regressors β_j. Fitting the model characterized by both (4.7) and (4.7) is equivalent to maximizing the pseudo-partial log-likelihood (Nicolaie, 2014), namely

$$\begin{aligned}\text{ipl}(\beta) = \sum_{j=1}^{J}\sum_{i=1}^{N} \mathbf{1}\{D_i = j\} \sum_{t_{LM}: t_{LM} \le t_i \le t_{hor}} \Big[\mathbf{Z}_i(t_{LM})^T \beta_j(t_{LM}) - \\ \log \sum_{t_{LM}: t_{LM} \le t_i \le t_{hor}} e^{\mathbf{Z}_j(t_{LM})\beta_j(t_{LM}))}\Big];\end{aligned} \tag{4.11}$$

with $\mathbf{1}\{\cdot\}$ the indicator function.

Likewise, we can assume the baseline hazards of (4.7) to have a smooth dependence on time. When the estimation of $\lambda_{0j}(t|t_{LM})$ is made via Breslow-type estimators (see, section D.1), we can model such dependence as

$$\lambda(t|t_{LM})_0 = \lambda_{0j}(t) \exp(\gamma_j(t_{LM})); \tag{4.12}$$

and therefore assume

$$\gamma_j(s) = g_j(s, \gamma_j), \tag{4.13}$$

with s a generic time value, $\gamma_j = (\gamma_{1j}, \dots, \gamma_{Mj})$ a vector of the regression parameters for M covariates and $g_j(t_{LM}, \gamma_j)$ a parametric function dependent on time; e.g a quadratic spline. Hence, the complete description of the time evolution of the regressors $\beta_j(t)$ and the baseline hazards $\lambda_j(t)$ can be given by means of equations (4.7), (4.10), (4.12), and (4.13). The model involving the direct application of all these equations is the so-called *landmark supermodel*.

However, the landmark supermodel can be equivalently described through the maximization of a pseudo-likelihood (Nicolaie, 2014), namely

$$\text{ipl}^*(\beta,\gamma)=\sum_{j=1}^{J}\sum_{i=1}^{N}\mathbf{1}\{D_i=j\}\log\left(\frac{\sum_{t_{LM}:t_{LM}\le t_i\le t_{hor}}\exp\left[\mathbf{Z}_i(t_{LM})^T\beta_j(t_{LM})+\gamma_j(t_{LM})\right]}{\sum_{t_{LM}:t_{LM}\le t_i\le t_{hor}}\sum_{t_k:t_k\ge t_i}\exp\left[\mathbf{Z}_k(t_{LM})^T\beta_j(t_{LM})+\gamma_j(t_{LM})\right]}\right) \tag{4.14}$$

Accordingly, we can estimate the baseline hazard, that is

$$\hat{\lambda}_{0j}(t_i)=\frac{\#(t_{LM}\le t_i\le t_{hor},\Delta_i=j)}{\sum_{t_{LM}:t_{LM}\le t_i\le t_{hor}}\sum_{t_k:t_{LM}\le t_i\le t_k\le t_{hor}}\exp\left[\mathbf{Z}_k(t_{LM})^T\hat{\beta}_j(t_{LM})+\gamma_j(t_{LM})\right]}, \tag{4.15}$$

and we denote the estimated cause-specific cumulative baseline hazard with

$$\hat{\Lambda}_{0j}(t)=\sum_{t_i\le t}\hat{\lambda}_{0j}(t_i). \tag{4.16}$$

Hence, we can estimate the desired survival function, namely

$$\hat{S}_{LM}(t_{hor}|Z(t_{LM}),t_{LM})=\exp\left(-\sum_{j=1}^{J}\exp\left(Z(t_{LM})\hat{\beta}_j(t_{LM})+\hat{\gamma}_j(t_{LM})\right)\left[\hat{\Lambda}_{0j}(t_{hor})-\hat{\Lambda}_{0j}(t_{LM})\right]\right), \tag{4.17}$$

and

$$\hat{F}_{j,LM}(t_{hor}|Z(t_{LM}),t_{LM})=\sum_{t_{LM}<t_i\le t_{hor}}\hat{\lambda}_{0j}(t_i|Z(t_{LM}))\hat{S}_{LM}(t_{hor}|Z(t_{LM}),t_{LM}). \tag{4.18}$$

Both the equations (4.17) and (4.18) represent the estimation of the desired dynamic prediction probabilities, i.e., the survival function and the cumulative incidence function for j-th cause at the horizon time t_{hor} given the information on the subjects' histories at time t_{LM}. Note that in both (4.17) and (4.18), we denote with the hat notation $\hat{(\cdot)}$ the estimated quantities, e.g., $\hat{\beta}_j(t_{LM})$ refers to the estimate of the regressors of the Cox-model for the j-th failure cause evaluated at time t_{LM}.

4.3.7 Landmark model in the Deep-LCPH model

When modeling the dynamic predictions of ICUAI episodes, one competing risk model with two competing and exclusive causes is employed, i.e., the onset of an acquired infection or one of both the events "death in the ICU" and "discharge"; see figure 4.12. Such modeling of the competing events enables us to deal with no censoring observation due to either the death of a patient or the discharge from the ICU department.

Following the notation used in section 4.3.6, we denote with $\tilde{T}$ the time of failure, D the case of failure (i.e., $D=1$ denotes an ICUAI episode, while $D=2$ the other event), and $\mathbf{Z}(\cdot)$ and array of covariates both time-fixed and time-dependent. For each i-th subject the tuple $(T_i,\Delta_i,\mathbf{Z}_i(\cdot))$ represents the observation at time $T_i\equiv\tilde{T}_i$, the cause of failing $\Delta_i\equiv D_i$, and $\mathbf{Z}_i(\cdot)$ the acquisition of the covariates up to time T_i. We stress that in our case, we denote with $\Delta=1$ the onset of an infectious state, while with $\Delta=2$ the other competing cause.

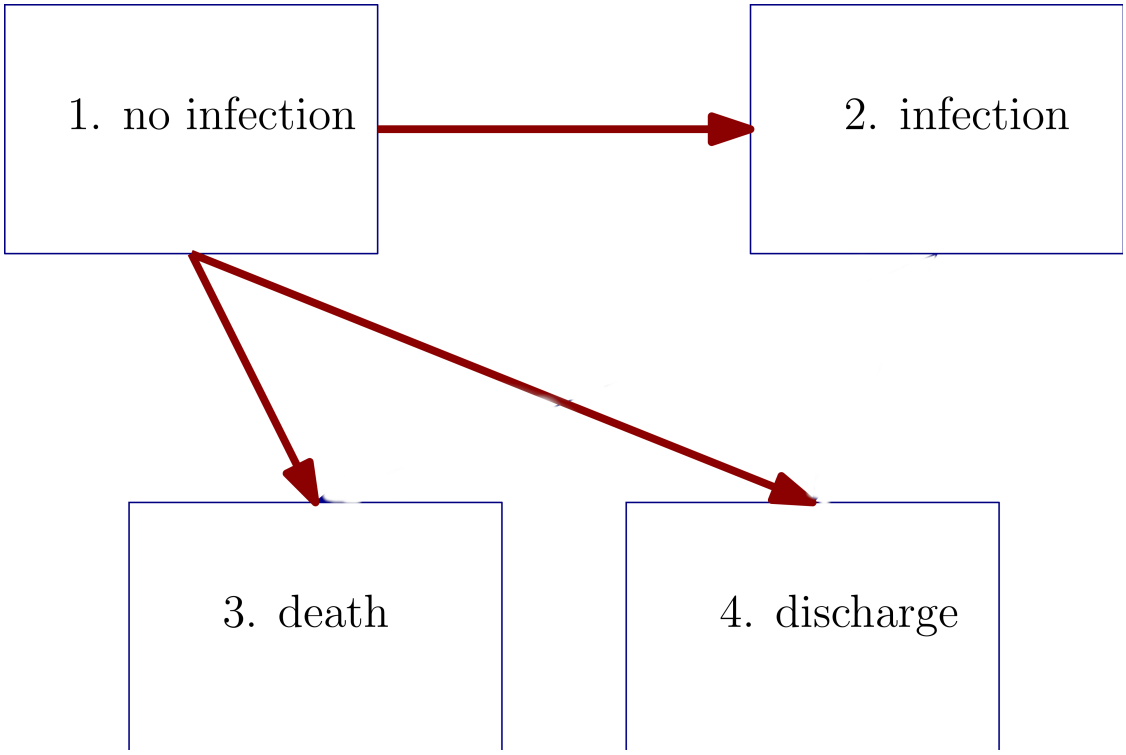


Figure 4.12: Scheme of the occupational states of the CR model. The red arrows denote the transitions that one ICU admitted patient ("1. no infection") could make towards the two causes of interest, i.e., the ICUAI episode ("2. infection") or the end of the ICU stay (either "3. death: or "4. discharge"). For completeness, the transitions a patient would do after getting infected in real life have been depicted; see the blue arrows.

In this study, therefore, we considered two realizations of the prediction window w, i.e., $w = 24$ hours and $w = 48$ hours. The grid selecting the landmark times t_{LM} in the time-domain $[s_0, s_1]$ has a width of 8 hours, i.e., two subsequent landmark times are distant 8 hours.

After incorporating the risk score of infection into the dataset with the selected ICU predictors, the fit of the landmark supermodel on this collection of data aims to estimate the following quantities

$$\hat{S}_{LM}(t_{hor}|Z(t_{LM}), t_{LM}) = \exp\left(-\sum_{j=1}^{2} \exp\left(Z(t_{LM})\hat{\beta}_j(t_{LM}) + \hat{\gamma}_j(t_{LM})\right)\left[\hat{\Lambda}_j(t_{hor}) - \hat{\Lambda}_j(t_{LM})\right]\right), \tag{4.19}$$

and

$$\hat{F}_{1,LM}(t_{hor}|Z(t_{LM}), t_{LM}) = \sum_{t_{LM} < t_i \leq t_{hor}} \hat{\lambda}_1(t_i|Z(t_{LM}))\hat{S}_{LM}(t_{hor}|Z(t_{LM}), t_{LM}). \tag{4.20}$$

While (4.17) informs about the chance that one subject stays in the ICU without both developing an ICUAI episode or being discharged or dying, (4.18) gives more information on the distribution probability of ICUAI episodes at different moments of the ICU stay.

4.3.8 Deep-LCPH model

Incorporating the risk score of infection into the traditional variables extracted from the ICU setting represents the innovative point that we proposed to improve the traditional modeling of the early detection of the onset of ICUAI episodes. The risk score of infection is evaluated by means of the 1-D CNN (see section 1.1.3) model; see section 4.3.2. When fitting the model to the input EHR data, we used only a portion of the total amount of available EHR data. Indeed, we undersampled the overall amount of EHR data to avoid the training and the test set being too imbalanced. Note that when considering either the 24-hour model or the 48-hour model, the number of Time-Series instances in the case group (i.e., those instances representing the ICUAI episodes) are about one-twentieth of the total amount of Time-Series instances in the control group (i.e., those instances not representing the ICUAI episodes). Thus, we fit the 1-D CNN model on a population of Time-Series instances with a *control-case ratio* of 8:1 (i.e., for each Time-Series instance in the case group one has eight Time-Series instances from the control group). It is important to stress that when undersampling the EHR data, we only apply a random undersampling on the control group.

We specify that the application of a random under-sampling has the scope of ensuring that the infection risk scores incorporated in the Landmark model do not coincide with the risk scores evaluated in the testing phase of the CNN model. In other words, we do not want to employ the totality of the time-series instances associated with the longitudinal covariates (i.e., baseline and low-frequency covariates) to train and test the CNN; this way, when incorporating the infection risk scores into the landmark model, one ensures that the CNN-based risk scores arise from time-series instances that have never been propagated thought the CNN before. Thus, when incorporating the infection risk scores, we want that most of the CNN-based risk scores to arise from time-series instances that have never been propagated through CNN before. As a result, most of the infection risk scores embedded in the Landmark model are generated from time-series instances that have never been involved in the training phase of the CNN.

We need to mention that the extraction of the time series instances is accomplished by starting from the admission time to the ICU, which is usually different with respect to the initial time of the low-frequency covariates. This fact implies that the time series instances used to evaluate the infection risk score and those used in the learning phase of the CNN are not precisely equal; one expects that the latter are anticipated or delayed with respect to the former by a mean time shift of four hours. Nevertheless, the introduction of such a possible shift does not dramatically alter the quality of the risk scores of infection because of the translational invariance property of CNN. The combination of both these details (i.e., the under-sampling of time-series instances and the slight time-shift intercurring between the time-stamp of the training time-series instances and the actual time-stamp at which the risk scores are evaluated) ensures that the large majority of instances employed in the evaluation of the infection risk score have never been employed in the learning phase of CNN previously.

Thus, the procedure to evaluate and incorporate the risk scores into the Landmarking model is the following

- Consider the EHR of one patient (Hear Rate, Arterial Blood Pressure mean, Pulse, SaO2, and Respiratory rate) and all his/her other auxiliary Time-Series.

- Starting from the admission time to the ICU, extract 24-hour (or 48-hour) Time-Series instances by means of an 8-hour sliding time window (see section 4.3.1).

- Propagate the Time-Series instances through the hidden layers of the 1-D CNN model and evaluate the risk-score
- Assign the risk score to the corresponding time-stamp (i.e., day-month-hour-minute).

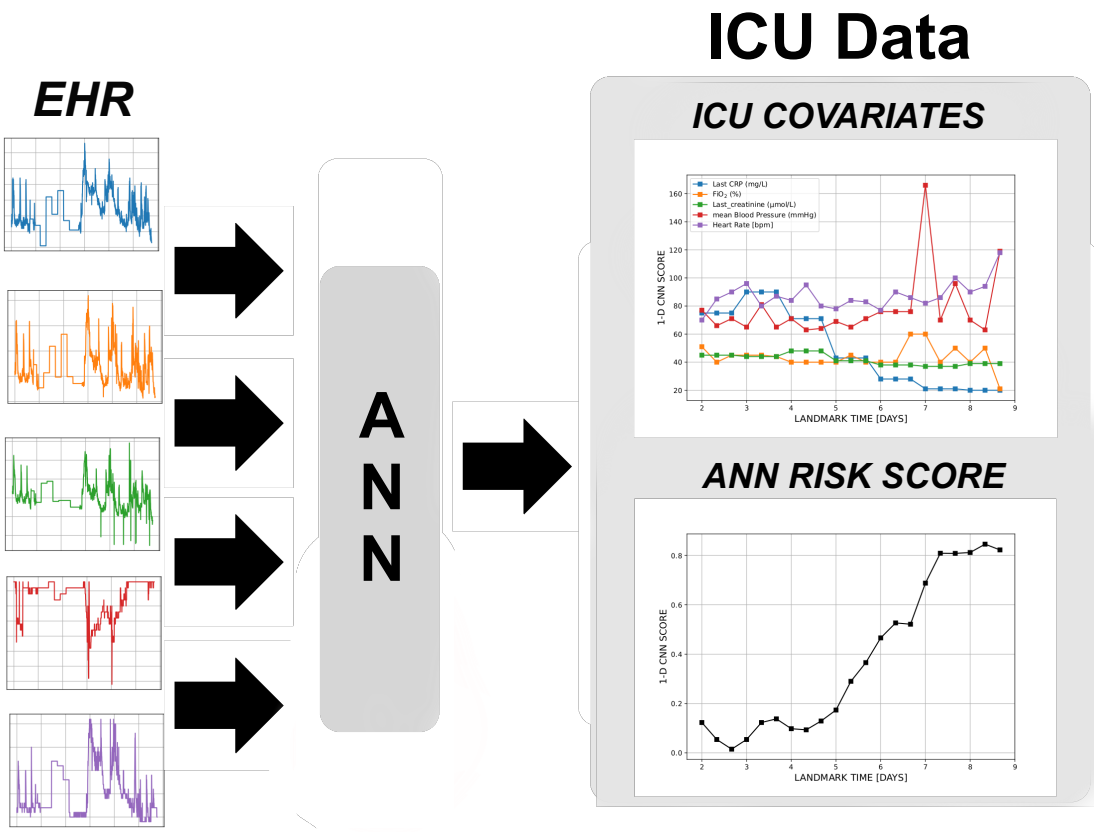


Figure 4.13: Schematic representation of the inclusion of the ANN-based risk score in the ICU cohort data.

A scheme of how we incorporated the risk score into the ICU traditional ICU covariates is depicted ins illustrated in figure 4.13. Depending on the time window length, we can therefore build two types of risk scores to model the onset of one event with two different lead times, i.e., 24 or 48 hours. The Landmark CR model, including the risk scores of infection, is therefore referred to as the *Deep-LCPH* model. The basic one, i.e., the Landmark Competing Risk model with only the traditional ICU variables, is referred to as the *LCHP* model.

To exploit the potentiality of CNN-based risk scores at best, we opted to include a secondary risk score evaluated by a 2-D CNN model. Although the 2-D CNN would not represent a performative class of models, we can argue that rearranging 1-D data into 2-D data produces a different description of the patterns contained in the vital signals. This implies that the CNN-2D could base its pattern recognition activity on the search for some spatial-temporal characteristics of EHRs that might not be considered in the 1D model. However, The poor impact that such an alternative risk

score would show in the Deep-LCHP model led us to cautiously consider the 2D-CNN score with a minor relevance.

4.3.9 Overall evaluation of the model

In this study, we trained 1-D CNN to solve a binary classification task. Therefore, AUROC score represents the most appropriate choice to assess the goodness of the learning phase. Although we are not interested in the prediction formulated by the 1-D CNN, we need to guarantee, after all, that the 1-D CNN model is exceptionally skilled in classifying the Time-Series instances. We recall that the primary goal of the 1-D CNN (see section 1.1.3) was the encoding in a single score (i.e., the risk score) of all those crucial patterns that are recorded in EHR with a resolution higher than hours and days (i.e., the time-scales that are frequent in a clinical longitudinal study involving ICU patients), i.e., a resolution of one minute. The evaluation of the AUROC has the scope of guaranteeing here that the risk score evaluated by the 1-D CNN model can represent an informative feature describing the impending onset of one acquired infection.

When evaluating the accuracy of both the LCHP and Deep-LCHP model, we can still use the AUROC metric to evaluate the prediction made at each single landmark time. Still, when considering an overall measure for the Deep-LCHP model, the evaluation of a global AUROC needs to consider that the dynamic of failing causes is time-dependent and not homogeneous on time. Similarly to what is proposed by Spitoni et al. (2018) for the *Prediction Error* of dynamic predictions in *Muti-State models*, the evaluation of the overall AUROC of the Deep-LCHP model needs to take into account both the number of subjects still at risk different landmark times and the presence of censored data. However, the absence of censored data in our longitudinal dataset allows us to express the overall AUROC score as

$$\text{AUROC}_{\text{global}} = \frac{\sum_{k=1}^{N_{LM}} \nu(t_{LM,k})\, \text{AUROC}(t_{LM,k})}{\sum_{k=1}^{N_{LM}} \nu(t_{LM,k})}; \tag{4.21}$$

with $t_{LM,k}$ the k-th landmark time, N_{LM} the total number of landmark times, and $\nu(t_{LM,k})$ the size of the risk set at $t_{LM,k}$.

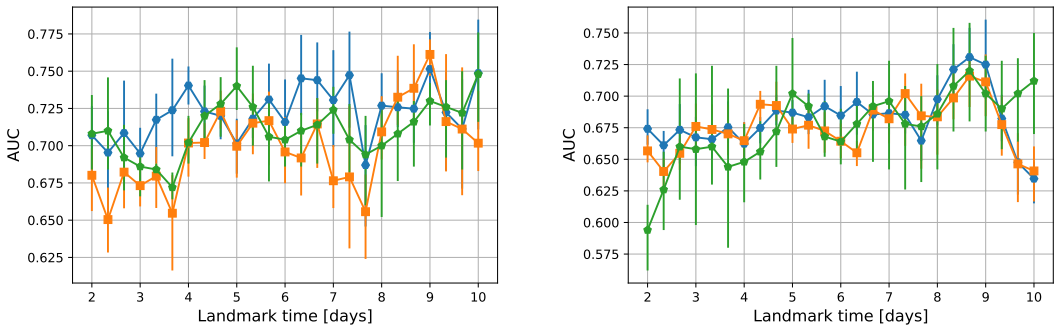
The influence that individual predictor variables have within the landmark model has been visualized by means of heat maps. We compute the relative variation of the overallAUROC between one basic model, including all predictors, and another one excluding only one single predictor. Thus, we construct a heat map representing the relative change in AUROC due to one single predictor at different moments of the stay in the ICU. At last, we remark that internal validation was performed using the *K-folds cross-validation* method. When validating the performance of 1-D CNN models as binary classifiers, the data were split into 5 folds, while for the landmark competing risk model, we used 10 folds. The overall AUROC is the average over the 5 fold; SEM is used to provide the uncertainty of the AUROC score. In both the LCHP and the Deep-LCPH model, the uncertainty of the mean predictive performance was obtained via a 95% bootstrap confidence interval (CI).

4.4 Dynamical prediction of ICUAI

The evaluation of the predictive performances of the modeling has been conducted for two specific prediction windows, i.e. 24 and 48 hours. Thus, we predicted the onset of ICUAI with a lead time

of 24 or 48 hours. To evidence the impact due to the 1-D CNN, we made a comparison between three models mainly, that is, the basic LCHP, the Deep-LCHP, and the 1-D CNN model treated as a classifier. The evaluation of the latter model consists of evaluating the AUROC score from the risk score provided by the 1-D CNN model. It is essential to specify that the evaluation of the accuracy level of the 1-D CNN (see section 1.1.3) model, when treated as a binary classifier, is performed by fitting the 1-D CNN model on an almost complete set (80%) of the Time-Series instances. In contrast, the employment of a more significant percentage (e.g., 100%) of the Time-series instances would have generated substantial computational costs and overlong training phases and, therefore, would have given rise to difficulties in obtaining adequate results.

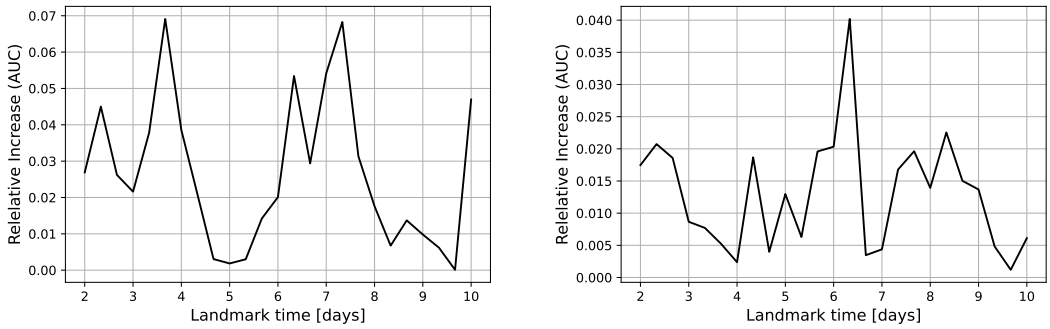
Starting with the 24-hour prediction models, we observed that the LCHP model shows an overall AUROC score equal to 0.69 (95%CI 0.68-0.70), while the Deep-LCHP shows a slight increase in the AUROC score, i.e., 0.72 (95%CI 0.71-0.73). The 1D-CNN model alone shows an AUROC equal to 0.71 (95%CI 0.69-0.73). The 48-hour prediction models show lower AUROC scores instead. The LCHP model shows an AUROC score equal to 0.67 (95%CI 0.66-0.69), while the Deep-LCHP has a AUROC score of 0.68 (95%CI 0.67-0.69) The Deep-LCHP shows a relative increase of the AUROC is equal to 0.01. In this case, we have an overall AUROC score equal to 0.69 (95%CI 0.67-0.72). With respect to the 24-hour model, the 48-hour 1-D CNN classifier model, instead, presents a slightly poorer performance, with an overall AUROC score of 0.68 (95%CI 0.64-0.72).



(a) (b)

Figure 4.14: Overall AUROC score (y-axis) as a function of the landmark times (x-axis) when predicting ICUAI episodes. The three curves represent the predictive performance of three models; the basic LCHP (orange), the Deep-LCHP (blue), and the 1-D CNN (green) model. The error bars denote the 95% bootstrap confidence intervals. (a) 24-hour models, (b) 48-hour models.

When considering the AUROC score at different moments of the stay in the ICU (e.g. we considered all 8-hour landmark times until day 10), all three models can reveal different behaviors; see figure 4.14. The results of the 24-hour prediction models are shown in figure 4.14a. The LCHP model reaches the minimum AUROC score of 0.65 (95%CI 0.63-0.67) at day 2.33; instead, at day 7.66, the Deep-LCHP reaches the minimum AUROC score of 0.68 (95%CI 0.65-0.71). The LCHP model reaches the maximum AUROC score of 0.76 (95%CI 0.75-0.77) at day 9; the same result is valid for the Deep-LCHP model as well, with AUROC 0.75(95%CI 0.74-0.75) The 1-D CNN as classifier reaches a minimal AUROC score of 0.67 (95%CI 0.66-0.68) at day 3.66 and a maximal



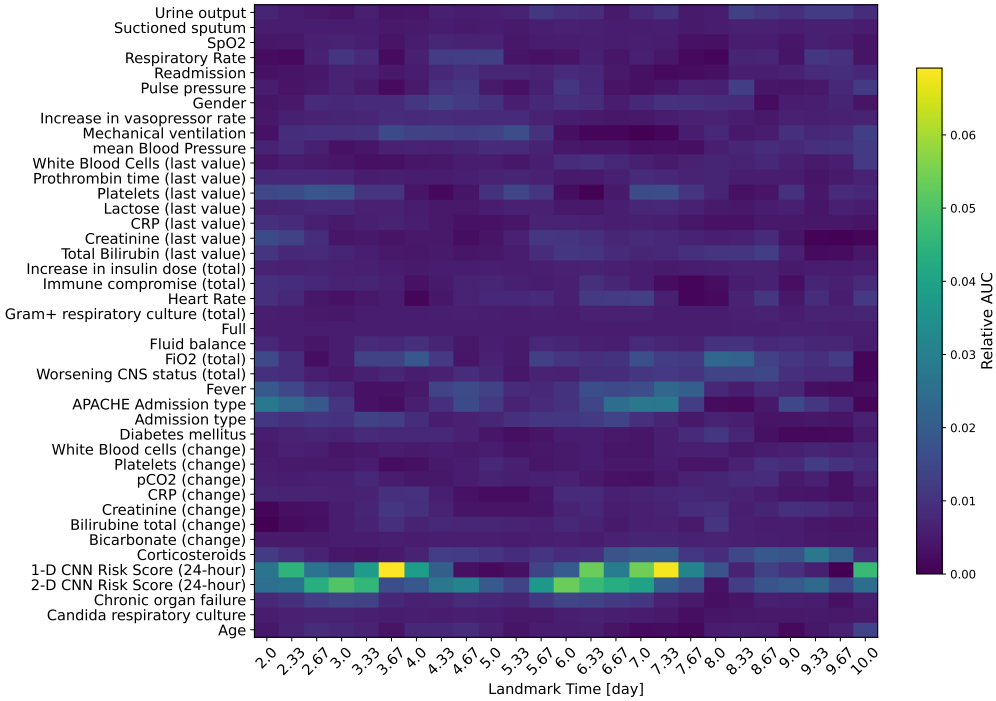
(a) (b)

Figure 4.15: ICUAI episodes. Overall Relative increase of AUROC score of LCHP model (y-axis) as a function of the landmark times (x-axis) when including 1-D CNN-based risk score of infection. (a) 24-hour models, (b) 48-hour models.

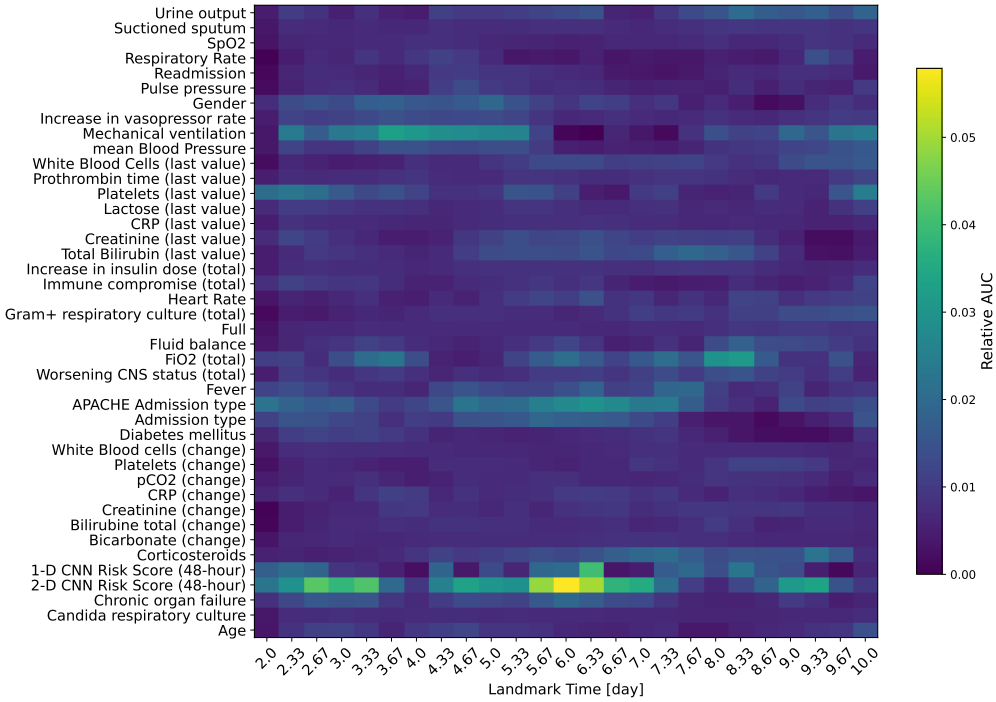
AUROC score of 0.75 (95%CI 0.67-0.77) at day 10. The 48-hour prediction model shows that the LCHP model has a minimum AUROC of 0.63 (95%CI 0.61-0.66) at day 10 and a maximal AUROC of 0.72 (95%CI 0.71-0.73) at day 8.66; for the Deep-LCHP, the minimum AUROC is 0.63 (95%CI 0.61-0.66) at day 10 and a maximal AUROC score of 0.73 (95%CI 0.65-0.75) at day 8.66, and for the 1-D CNN as binary classifier, AUROC 0.59 (95%CI 0.56-0.63) at day 2 and 0.71 (95%CI 0.67-0.75) at day 10; see figure 4.14b.

The impact of each explanatory variable on the Deep-LCHP is shown in figure 4.16. The evaluation of the AUROC score by adding a covariate to the others reveals that the risk scores provided by the CNN models are the covariates with the largest impact. Despite incorporating such a covariate does not seem to give the overall AUROC a relative increase of 2%, one can see that such an increase attains different values at different moments of the ICU stay. For the 24-hour model, the risk scores provided by the CNN models with 24-hour time windows (see figure 4.16a; the corresponding covariates are named *1-D CNN Risk Score (24-hour* and *2-D CNN Risk Score (24-hour* for the 1D-CNN and 2D-CNN model, respectively) can give larger contributes in the proximity of day 4, 6 and 7. The risk score of the 1-D CNN can relatively increase the 6-7% of the base AUROC scores at two precise moments, i.e. days 3.66 and 7.33. Whereas a smaller contribution of 5% is attained at day 6.33; see figure 4.16a. The scores of the 2D-CNN model do not seem to contribute in a very substantial manner; as shown in figure 4.16a, this covariate can increase the AUROC base by 3-4% in the proximity of days 3 and 6. Likewise, for the 48-hour model, the 1D-CNN risk score can give larger but limited contributions in the proximity of days 6 only (more precisely at day 6.33 with an increase equal to 4%), while the 2D-CNN risk score contributes with a relative increase of 3-4% in the range of days 4 and 6; see figure 4.16b (see covariates *1-D CNN Risk Score (48-hour* and *2-D CNN Risk Score (24-hour*).

Despite being not so incisive in increasing the relative predictive performance of the landmarking model, when integrated into that, the CNN risk scores turned out to be the most important predictors. The evaluation of the cause-specific hazard ratio (i.e the β regressors specific for each CNN risk score) for the 1-D CNN risk scores achieve values of 4.8 (95%CI 3.05-6.72) and 7.3 (95%CI 4.44-15.91) for the 24-hour and 48-hour models, respectively. Instead, the 2-D CNN risk



(a)



(b)

Figure 4.16: AUC heatmaps evaluating the impact of each predictor in the Deep-LCPH model when predicting ICUAI episodes; (a) 24-hour prediction model and (b) 48-hour prediction model. The color of each pixel denotes the magnitude of the impact (relative AUROC increase) of one covariate (y-axis) with respect to the days elapsed since the admission time in the ICU (x-axis).

scores do not reveal an important impact as predictors; the cause-specific hazard ratios are equal to 0.67 (95%CI 0.24-1.82) and 0.5 (95%CI 0.22-1.25) for the 24-hour and 48-hour models, respectively. The complete tables with the cause-specific hazard ratio are reported in Table 4.5 (24-hour model) and Table 4.6 (48-hour model).

In this study, the 1D-CNN risk score consistently emerged as a strong predictor of ICU-AI and achieved similar risk discrimination using only vital signs monitor data without requiring extensive predictor selection or engineering. The dual approach modeling, however, leads to marginal improvements in the predictive performance when evaluated with respect to the Landmarking Competing Risk model. Our findings appear to agree with an extensive systematic review of 71 studies that found no substantial benefits of clinical prediction models based on machine learning compared to logistic regression (Christodoulou et al., 2019). Likewise, two recent investigations looking at machine learning models for predicting outcomes in the domain of cardiology (Khera et al., 2021; Desai et al., 2020) found limited improvement over logistic regression. Although using landmarking for dynamic prediction is still relatively new, Cox proportional hazards models represent a widely applied and accepted statistical approach. Similarly to our approach, Tanner et al. (2021) reported similar cross-validated predictive performance between a machine learning ensemble survival model and an LCPH model.

Although deep learning did not significantly impact forecasting ICU-acquired infections, the CNN-based risk scores offer potentially outstanding advantages. Whereas model-based methods, as well as most ML methods, require hand-engineered input variables or features, CNN models can automatically and adaptively recognize spatial hierarchies within data that is readily available in EHR. Furthermore, we found that the accuracy of the CNN-based risk scores and the basic LCPH model are much similar. While the former needs five vital signals from the continuous monitoring of EHR, the latter needs a much broader range of clinical predictors. Moreover, other clinical predictors varied in predictive strength across time, a CNN-based risk score demonstrated strong predictive potential across ICU stay. Apart from the high skill of ANN methods in extracting highly detailed complex information from a few Time-Series, such a method offers the possibility of dynamically supporting the detection of ICU-acquired infections. Refining the CNN-based risk scores offers the perspective of a new tool in any ICU monitor.

4.5 Dynamical prediction of ICUFAI

When dealing with ICUFAI episodes, the dynamic predictions are evaluated utilizing the same approach of section 4.4. In this case, the two-step modeling takes into account only one 1-D CNN model, i.e., the 24-hour 1-D CNN (see section 1.1.3). This implies that the Deep-LCPH model is embedded with the risk scores formulated by analyzing the 24-hour 1-D CNN model. When implementing the landmark model, we considered two prediction windows mainly, i.e., 24 and 48 hours. Unlike section 4.4, we do not consider the 1-D CNN model treated as a binary classifier because the goal of this section is to assess if the embedding of the 1-D CNN risk score into the LCHP model can help increase the predictive power of traditional clinical predictors when considering ICUFAI episodes. For this reason, we decided not to compare the predictive performance of Deep-LCHP with the 1-D CNN model.

For the 24-hour models, the LCHP model shows an overall AUROC score equal to 0.69 (95%CI 0.68-0.70), while the Deep-LCHP shows an overall AUROC score of 0.75 (95%CI 0.73-0.76). While for the 48-hour models, the LCHP model shows an overall AUROC score of 0.71 (95%CI

Variable name	β	β-CI
Urine output	1	0.99-1.01
Suctioned sputum	1	0.99-1.02
SpO2	0.97	0.95-0.98
Respiratory Rate	1	0.99-1.01
Readmission	0.92	0.85-1.11
Pulse pressure	1	0.99-1.01
Gender (male)	1.4	1.20-1.53
Increase in vasopressor rate	1.4	1.25-1.49
Mechanical ventilation	1	0.88-1.21
mean Blood Pressure	1	0.99-1.01
White Blood Cells (last value)	1	0.99-1.01
Prothrombin time (last value)	1	0.99-1.01
Platelets (last value)	1	0.99-1.01
Lactose (last value)	0.98	0.95-1.04
CRP (last value)	1	0.99-1.01
Bilirubin (total)	1	0.99-1.01
Increase in insulin dose (total)	1	0.99-1.01
Immune compromise (total)	1.2	1.01-1.54
Heart Rate (total)	1	0.95-1.05
Gram+ respiratory culture (total)	1.1	1.01-1.54
Fluid balance	1	0.99-1.01
FiO2 (total)	1	0.99-1.01
Worsening CNS status (total)	1.1	1.02-1.18
Fever	2	1.86-2.75
APACHE-Trauma	1	0.92-1.34
APACHE-Transp	0.97	0.86-1.26
APACHE-Respir	0.67	0.54-0.77
APACHE-Other	0.58	0.33-0.96
APACHE-Neuro	1.2	1.01-1.48
APACHE-Gastro	0.71	0.54-0.96
Admission Type (surgical)	1.3	1.16-1.44
Diabetes mellitus	0.82	0.73-0.99
White Blood cells (change)	1	0.99-1.01
Platelets (change)	1	0.99-1.01
pCO2 (change)	1	0.99-1.01
CRP (change)	1	0.99-1.01
Bilirubine total (change)	1	0.99-1.01
Bicarbonate (change)	1	0.99-1.01
Corticosteroids	1.2	0.99-1.48
1-D CNN Risk Score (24-hour)	4.8	3.05-6.72
2-D CNN Risk Score (24-hour)	0.67	0.24-1.82
Chronic organ failure	1.1	0.95-1.28
Candida respiratory culture	0.82	0.72-0.91
age	1	0.99-1.01

Table 4.5: Cause-specific hazard ratio for the 24-hour model.

Variable name	β	β-CI
Urine output	1	0.99-1.01
Suctioned sputum	1	0.99-1.01
SpO2	0.97	0.94-1.02
Respiratory Rate	1	0.99-1.01
Readmission	0.92	0.91-1.13
Pulse pressure	1	0.99-1.01
Gender (male)	1.4	1.17-1.44
Increase in vasopressor rate	1.4	1.25-1.48
Mechanical ventilation	1.2	0.85-1.29
mean Blood Pressure	1	0.99-1.01
White Blood Cells (last value)	1	0.99-1.01
Prothrombin time (last value)	1	0.99-1.01
Platelets (last value)	1	0.99-1.01
Lactose (last value)	0.98	0.96-1.01
CRP (last value)	1	0.99-1.01
Bilirubin (total)	1	0.99-1.01
Increase in insulin dose (total)	1	0.99-1.01
Immune compromise (total)	1.2	1.05-1.47
Heart Rate (total)	1	0.99-1.01
Gram+ respiratory culture (total)	1.1	1.02-1.43
Fluid balance	1	0.99-1.01
FiO2 (total)	1	0.99-1.01
Worsening CNS status (total)	1.1	0.98-1.22
Fever	1.8	1.65-2.54
APACHE-Trauma	1.1	0.92-1.14
APACHE-Transp	0.98	0.88-1.32
APACHE-Respir	0.7	0.51-1.01
APACHE-Other	0.57	0.41-0.99
APACHE-Neuro	1.1	0.97-1.54
APACHE-Gastro	0.72	0.45-1.46
Admission Type (surgical)	1.3	1.05-1.53
Diabetes mellitus	0.84	0.64-0.90
White Blood cells (change)	1	0.99-1.01
Platelets (change)	1	0.99-1.01
pCO2 (change)	1	0.99-1.01
CRP (change)	1	0.99-1.01
Bilirubine total (change)	1	0.99-1.01
Bicarbonate (change)	0.99	0.99-1.01
Corticosteroids	1.2	0.98-1.65
1-D CNN Risk Score (48-hour)	7.3	4.44-15.91
2-D CNN Risk Score (48-hour)	0.5	0.22-1.25
Chronic organ failure	1.1	0.93-1.21
Candida respiratory culture	0.82	0.69-0.95
Age	1	0.99-1.01

Table 4.6: Cause-specific hazard ratio for the 48-hour model.

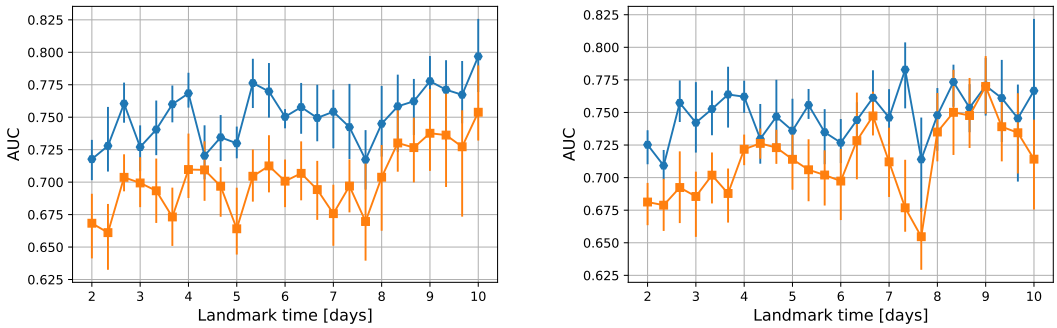


Figure 4.17: Overall AUROC score (y-axis) as a function of the landmark times (x-axis). The three curves represent the predictive performance of three models when predicting ICUFAI episodes; the basic LCHP (orange), the Deep-LCHP (blue), and the 1-D CNN (green) model. The error bars denote the 95% bootstrap confidence intervals. (a) 24- hour models, (b) 48-hour models.

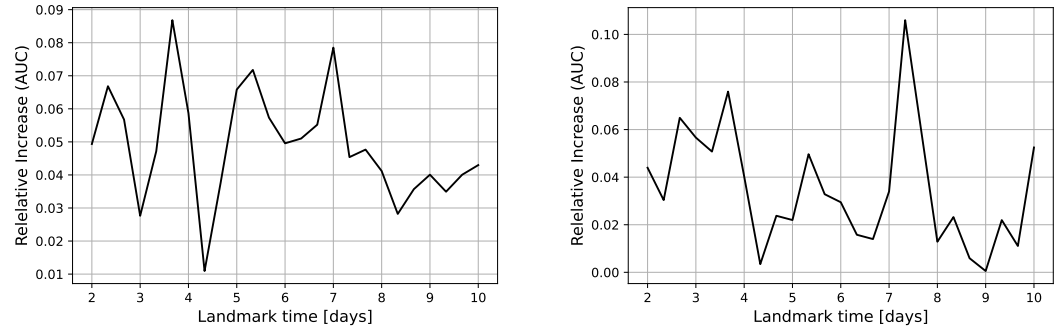


Figure 4.18: ICUFAI episodes. Overall Relative increase of AUROC score of LCHP model (y-axis) as a function of the landmark times (x-axis) when including 1-D CNN-based risk score of infection. (a) 24-hour models, (b) 48-hour models.

0.72-0.74), while the Deep-LCHP shows an overall AUROC score equal to 0.75 (95%CI 0.74-0.76).

The AUROC score at different moments of the stay in the ICU (i.e., we considered all 8-hour landmark times from admission until day 10, as in section 4.4), are shown in figure 4.17. The predictive performance of both the 24-hour and 48-hour prediction windows are shown in figure 4.17a and 4.17b, respectively. As we can see, the LCHP model always shows lower predictive performance with respect to the Deep-LCHP; this is shown for both 24 and 48-hour prediction windows. For the 24-hour prediction window (see figure 4.17a), the LCHP model reaches the maximum AUROC score of 0.75 (95%CI 0.73-0.78) at day 10; likewise, the Deep-LCHP reaches the maximum AUROC at day 10, but with score 0.80 (95%CI 0.77-0.82). The Deep-LCHP model reaches the minimum AUROC score of 0.71 (95%CI 0.70-0.72) at day 7.66; instead, the LCHP with AUROC score of 0.66 (95%CI 0.63-0.68) at day 2.33. For the 48-hour prediction window (see figure 4.17b), the Deep-LCHP model reaches the maximum value of AUROC score of 0.78 (95%CI 0.75-0.80) at day 7.33; whereas, the AUROC of the LCHP model attains the maximum value 0.77

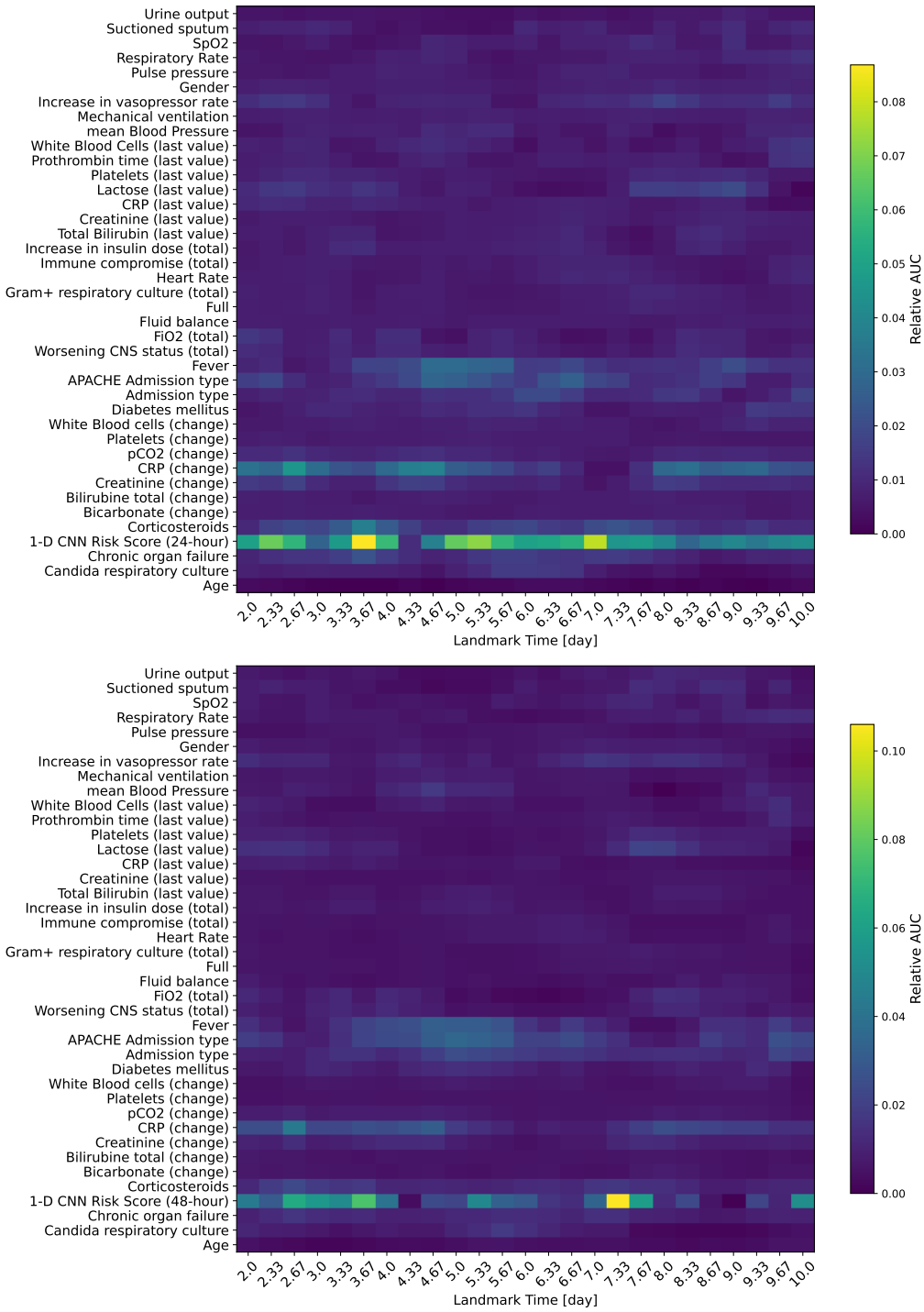


Figure 4.19: AUC heatmaps evaluating the impact of each predictor in the Deep-LCPH model when predicting ICUFAI episodes; (a) 24-hour prediction model and (b) 48-hour prediction model. The color of each pixel denotes the magnitude of the impact (relative AUROC increase) of one covariate (y-axis) with respect to the days elapsed since the admission time in the ICU (x-axis).

(95%CI 0.75-0.79) at day 9. The minimum value of AUROC is attained by both Deep-LCHP at day 7.66 with AUROC 0.71 (95%CI 0.65-0.75); LCHP models shoes the minimum AUROC 0.65 (95%CI 0.63-0.68) at time 7.66.

Similarly to the approach used in section 4.4, the impact of the explanatory variable involved in the Deep-LCHP model (including the 1-D CNN risk scores) is shown in figure 4.19. For the 24-hour prediction window, the evaluation of the AUROC score by excluding a covariate from the others reveals that the 1-D CNN risk scores have a relevant impact in the early detection of ICUFAI; at the beginning of the ICU stay (day 2-4) and around day 7, the 1-D CNN can improve the prediction of traditional ICU clinical covariates of 0.08 points of AUROC; see figure 4.19a. Specifically, one can see that day 3.66 represents the maximal impact (AUROC increases of 0.08 points), while an impact of 0.01 is observed on day 4.66. For the 48-hour prediction window, the evaluation of the impact of the 1-D CNN scores (after excluding that covariate from the others) reveals that the risk scores can marginally increase the AUROC of 0.02-0.04 values at almost all landmark times; see figure 4.19b. However, a peak of 0.07 is observed at the landmark interval of the ICU stay days 2-4 (see day 3.66). A maximum increase of 0.10 AUROC points is observed around day 7.66; see figure 4.19b.

From the results shown in figure 4.17, one can see that when considering ICUFAI events, the two-step modeling can effectively lead to a considerable increase in the accuracy of ICUFAI predictions. Including a CNN-based risk score of infection to the information traditionally extracted from the ICU setting led to a relative increase of 0.06 in the global AUROC, when considering the 24-hour Deep-LCHP model. For the 48-hour model, the relative increase in the global of AUROC is 0.04. Note that, for the 24-hour model, the AUROC at each landmark time is always larger in the Deep-LCHP with respect to the LCHP model. This implies that the impact that a CNN-risk score variable has on the early detection of ICUFAI events is robust and reflects the intuition that high-frequency data represent a source of detailed data that can be exploited to improve patients' prognosis. Likewise, the heatmaps of figure 4.19 show that, with respect to the other covariates, the CNN-based risk score of infection has a substantial impact in the prediction formulated at almost all landmark time for both the 24-hour and the 48-hour Deep Deep-LCHP model. Although the risk score of infection can only marginally improve the performance prediction of the LCHP model when dealing with ICUAI episode, we can see that the reformulation of a less challenging and less dispersive outcome shows that the potentiality of ANN models can offer a new perspective for improving the early detection and the treatment of ICUAI episodes.

4.6 Explanability of 1-D CNN-based prediction of ICUFAI

In this section, we present our attempt to make interpretable the activity of 1-D CNN model when dealing with the prediction of ICUFAI. As shown in section 4.5, the CNN-based risk score shows an added value in predicting ICUFAI events. For this reason, we shall attempt to investigate which characteristics of the EHR data are selected by the 1-D CNN model. We exploited the potentiality of one XAI algorithm (i.e., the SMOE scale; see section 1.3.2) to visualize and analyze the most relevant chunks of the Time-Series instances.

4.6.1 Explanability via SMOE

After training a 1-D CNN model to classify the clinical history of patients who experienced an ICUFAI episode, we leveraged the power of saliency maps (see section 1.3) to visualize in the Time-Series features the most relevant 8-hours patterns. We remark that those relevant patterns contain the features that the 1-D CNN select to discern the forthcoming ICUFAI events fully. Similar to the discussion of section 4.3, we trained the 1-D CNN utilizing the 24-hour Time-Series instance, i.e., chunks 24-hour vital signals continuously recorded by the ICU monitors (sampling frequency one minute). We recall these vital signals are Heart Rate, Arterial Blood Pressure (mean), Pulse, SaO2, and Breath Rate. To construct the saliency maps, we used the method developed by Mundhenk et al. (2019) (See section 1.3.2); this algorithm helped us better comprehend the statistics of the activated feature maps of the hidden layers. Hence, we deeply investigated how the feature maps get activated as the Time-Series instances are propagated through the 1-D CNN model.

The approach we developed can be divided into five main steps:

1. This is a preliminary step. The 1-D CNN model is trained on a subset of Time-Series windows (EHR vital signals, amplitude 24 hours) that partially cover the whole stay of a group of patients admitted in the ICU. We recall that one assumes that each patient is subjected to two competing events, i.e., the ICUFAI episode or the other event represented by either the death or the discharge from the ICU. All EHR acquired after the ICUFAI episode are neglected. We stress the point that we are not interested here in considering the 1-D CNN (see section 1.1.3) as a binary classifier, but rather as a feature extractor. That is, we are not interested in evaluating the high skill of CNN in solving a binary classification problem but instead in studying the ability of CNN to pre-process the input data to make easy-to-solve a binary classification problem. In other words, the evaluation of the accuracy of the 1-D CNN will be rather intended as a measure of the goodness of the pre-processing performed by the hidden layers of the 1-D CNN.

2. After assessing that 1-D CNN can solve the proposed binary problem (i.e., ICUFAI early detection), we fitted a 1-D CNN model for each *landmark time*, i.e., every 8 hours after 48 hours from the admission in the ICU. For instance, we take the first landmark time at time *56 hours* (from the admission in ICU); and accordingly, let the 1-D CNN model solve the classification of ICUFAI events only by analyzing the vital signals of those patients that are still at risk in the interval of landmark times 48 and 56 hours. Again, let's take the landmark time corresponding to day 10 from the admission to the ICU. We will allow the 1-D CNN (see section 1.1.3) to solve the binary classification tasks by analyzing only the vital signals of those patients that still are at risk in the interval of landmark times 232 and 240 hours. All these models are validated via 5-fold cross-validation.

3. We study the pattern recognition performed by the hidden layer, and we make it interpretable via *SMOE scale* (Mundhenk et al., 2019) (see section 1.3.2). Through this method, we can visualize the regions of the input data with the highest saliency. Specifically, for each model developed at every landmark time, we construct and visualize the saliency maps of the test set only. Note that we repeat this action for each test set of each cross-validation fold. From each saliency map, we extracted the 8-hour interval with the highest cumulative saliency value. We remark that an 8-hour interval mirrors the typical period between consecutive measurements and the assessment of time-dependent variables in ICU; see section 4.3.4.

4. After extracting the most relevant 8-hour patterns from each Time-Series instance, we focused on their interpretation. Thus, per each relevant pattern, we evaluated some traditional statistics (e.g., mean value, minimum value, and maximum value) and some other more complex statistics (e.g., approximate entropy (Pincus, 1991), permutation entropy (Bandt and Pompe, 2002), CID (Batista et al., 2014)). In other words, we described each 8-hour salient pattern by means of a broad set of different statistics; to be more close to the ML terminology, we shall refer to the *statistics extracted from the most salient 8-hor patterns* as the *new features*. Univariate feature selection was performed to select only those features exhibiting significant differences between the two classes of events (i.e., *presence of acquired infection in the next 24 hours* and *no infection*). Specifically, we utilized the one-parametric *Mann-Withney U-test* (Mann and Whitney, 1947). This step represents the key point of this study: *the essential characteristic of patterns captured by 1-D CNN is made interpretable by means of some specific statistics*.

5. We finally evaluate the goodness of the features we proposed in the last step. Thus, once again, we considered the binary classification of ICUFAI events. In particular, we paid attention to the predictions proposed by two models, i.e., a Multinomial Logistic Regression (MLR) fitted with the traditional ICU explanatory variables (the same introduced in section 4.3) and the new features obtained in the last step. After fitting the MLR models at each landmark time, we evaluated its goodness through the AUROC score. We stress that We opt for a MLR in place of the standard LR because we need to separate the effects caused by an acquired infection and the effects due to competing events as *death in ICU*, and *discharge from ICU*. Note that when fitting the MLR model at one landmark time, we utilized the information available at that landmark time only.

The architecture of the 1-D CNN model (see section 1.1.3) that we fitted in this work is the following:

- Convolutional layers with ReLU activation functions (i.e., $\phi(x) = \max\{0, x\}$), max-pooling layers and Dropout layers (Srivastava et al., 2014). This sequence of hidden layers is repeated 3 times
- All convolutional layers are made of 128 convolutional filters with an amplitude of 3 pixels (i.e., the kernel size); neither bias terms nor stride are involved when the convolutions are computed. The convolutions are performed without introducing artifacts at the edge; i.e., we used a *valid padding*
- Max-pooling layer selects the maximum value of the activated feature maps every 2 pixels
- The dropout layer prevents overfitting by disabling the action of a random portion of filters during the training (see section 1.2). We set a dropout rate equal to 0.25
- After that, the activated feature maps are flattened (feature maps are usually represented as a tensor; *flatten layers* makes their representation more practical, for example, like a 1-D array).
- Lastly, we use a *fully-connected layer* with sigmoidal activation to make the final prediction. In other words, the flattened feature maps are propagated through a *fully-connected layer* to obtain a value between 0 and 1. As in section 4.3, the output of the 1-D CNN model denotes the risk that a patient will get an infection in the next 24 hours.

- The model minimizes the *Binary Cross-Entropy* loss function. The optimizer is the ADAM algorithm (Kingma and Ba, 2014b). Learning rate and batch size are equal to 10^{-3} and 128, respectively.

4.6.2 Non-linear statistics for SMOE Scale interpretation

We can now pass to describe the statistics involved in the interpretation of the most salient 8-hour patterns proposed by the SMOE scale method. As introduced above, the statistical description of these patterns is therefore performed by combining traditional statistics (i.e., mean value, maximum value, minimum value, and second moment) with different types of non-linear estimators. For the non-linear estimators, we opted for Approximate entropy (Pincus, 1991; Pincus et al., 1991), Permutation Entropy (Bandt and Pompe, 2002), CID C3 (Batista et al., 2014), and C3 statistic (Schreiber and Schmitz, 1997).

Approximate entropy and Permutation Entropy are powerful statistics widely used to estimate the randomness of Time-Series, regardless of any assumption on the dynamics ruling the data. Therefore, their applicability is limitless and have been used in a wide variety of medical research (Pappalettera et al., 2022; Engoren, 1998; Varela et al., 2005; Kalpakis et al., 2015; Ciarrocchi et al., 2019).

We start from the Approximate Entropy. Given a Time-Series $\{X(t)\}_{t\in T}$, we define the distance

$$d[X(i), X(j)] = \max_{k\in\{0,1,\dots,m\}} \{|X(i+k) - X(j+k)|\}; \tag{4.22}$$

with both $X(i)$ and $X(j)$ denoting the chunks of the Time-series $\{X(t)\}$ with fixed length m, and starting, respectively, at the time points i and j, respectively. The quantity m is the so-called *scale length*. We also recall the total number of chunks that one can obtain from $\{X(t)\}_{t\in T}$ is equal to $T-m+1$, with T the total length of $\{X(t)\}_{t\in T}$. Thus, we can define an entropy measure based on the number of chunks that meet the condition $d[X(i), X(j)] \leq r$; where r is a positive real-valued constant termed *filtering*, namely

$$\Phi^m(r) = \frac{1}{T-m+1} \sum_{i=0}^{T-m+1} \log C_i^m(r); \tag{4.23}$$

where

$$C_i^m(r) = \frac{1}{T-m+1} \sum_{j=0}^{T-m+1} \mathbf{1}_{d[X(i),X(j)]\leq r}; \tag{4.24}$$

with $\mathbf{1}(\cdot)$ denoting the indicator function. Hence, Approximate Entropy is therefore defined as

$$\text{ApEn}(X(t), m, r) = \Phi^m(r) - \Phi^{m+1}(r), \tag{4.25}$$

The definition of *Permutation Entropy* is expressed as follows

- We denote with $\{X(t)\}_{t\in T}$ a Time-Series. We also define the *embedding time delay* positive real-valued constant termed and we denote it with τ; we denote with D an integer constant the *embedding dimension*.

- We construct $\chi = \{\{X(t)\}_{t\in(\tau,\tau+D)}, \dots, \{X(t)\}_{t\in(T-\tau-D,T-\tau)}\}$, i.e. we chunk $X(t)$ in intervals of length D and stride τ. We denote with χ_i the i-th element of χ.

- We make a rank statistic of each element χ_i, the values of χ_i are mapped into the ordinal ranks. For instance, if we denote $\chi_i = \{X_i, X_{i+1}, X_{i+2}\}$ (we prescribed τ= 1, and D = 2), then, according to the ranks of X_i, X_{i+1}, and X_{i+2}, we will map $\{X_i, X_{i+1}, X_{i+2}\}$ into one of the $D!$ possible permutations, i.e., $\{\{0,1,2\},\{0,2,1\},\{1,0,2\},\{1,2,0\},\{2,1,0\},\{2,0,1\}\}$. More specifically, if $X_i > X_{i+1} > X_{i+2}$, then χ_i will be mapped into $\{2,1,0\}$; again, if $X_i < X_{i+1}$ and $X_{i+1} > X_{i+2}$ and $X_{i+2} < X_i$, then χ_i will be mapped into $\{1,2,0\}$.
- We consider all the ranked elements of χ and we denote with π_k the relative occurrence of the k-th permutation.
- Permutation Entropy is given by

$$\text{PE}(\{X(t)\}_{t\in T}, D) = -\sum_{k=0}^{D!} \pi_k(\{X(t)\}_{t\in T}, \tau, D) \log_2 \pi_k(\{X(t)\}_{t\in T}, \tau, D). \quad (4.26)$$

We used both Approximate and Permutation Entropy to understand the complex behavior of the Time-Series instances; when using the Approximate Entropy we set $m = 10$ (corresponding to one and a half hours) and $r = 0.001$ (when considering Heart Rate, it corresponds to 0.2 beats per minute; for SaO_2, 0.1% of O_2 level; for Arterial Blood Pressure (mean) 0.05 mmHg; for Pulse 0.04 mmHg; and for Respiratory Rate 0.01 Breath per minute), while for Permutation Entropy we opted for a $\tau = 1$, and $D = 6$.

Other statistics such as CID C3 (Batista et al., 2014) and C3 statistics (Schreiber and Schmitz, 1997) are valid alternatives to get more in touch with the non-linearity of data. CID C3 statistics is often presented as a measure of the complexity of a Time-Series (Batista et al., 2014); this estimator expresses how ample the variety of patterns of a Time-Series is, i.e. the more the presence of valleys and peaks, and the more is the magnitude of this estimator. More in specific, if $\{X(t)\}_{t\in T}$ is a Time-Series, then CID is defined as

$$\text{CID-C3}(\{X(t)\}_{t\in T}) = \sqrt{\sum_{i=0}^{T} (X(i+1) - X(i))^2}, \quad (4.27)$$

that is a measure of the energy of the estimated first derivative of $X(t)$. Instead, the C3 statistic is more focused on measuring the non-linearity of a Time Series; therefore for the Time-Series $\{X(t)\}_{t\in T}$, C3 is defined as

$$\text{C3}(\{X(t)\}_{t\in T}) = \frac{1}{T} \sum_{i=0}^{T} \text{L}_{2k}(X(i)) \text{L}_k(X(i)) X(i), \quad (4.28)$$

with L_k defining the lag operator of order k, i.e., when applying this operator to the generic Time-Series evaluated at time t, i.e., $X(t)$. Thus, one has

$$\text{L}_k(X(t)) = X(t-k).$$

The number of new explanatory variables that we can extract with this method is therefore dependent on both the number of features of the Time-Series instances (five) and the number of statistics (eight) involved (mean value, standard deviation, mean minimum value, maximum value,

approximate entropy, permutation entropy, CID-C3, and C3). For each feature of each Time-Series instance, we evaluated the statistics we introduced above. Not all these new estimators could be predictive primarily to achieve a sufficient level of discrimination of ICUFAI episodes. Indeed, most of these new variables could not turn out to help improve the early classification of ICUFAI. For this reason, it is necessary to reduce the number of new explanatory variables; we aim to select only those that give rise to an improved classification task. Univariate feature selection is a strategy to make feature selection; this approach is based on univariate statistical tests aiming to point out those features showing the highest level of separation. From a very general point of view, the unique drawback of such a class of methods lies in the impossibility of selecting the features regarding the tangled effects between two or more features that might characterize the type of events. However, the univariate feature selection finds an effortless application in binary problems and represents a powerful tool to select all those variables whose independent contribution can eventually be as weak (but not utterly uninformative) as uncorrelated noise.

The method we employed is the *non-parametrical Mann-Whitney U-test* (Mann and Whitney, 1947). This non-parametric statistical test allows us to ascertain if the distributions of two independent sets of samples are equal. The null hypothesis is that two sets of samples follow the same distribution. The way we use this method is the following: we want to find an evident difference between the sample distributions of the new explanatory variables conditioned to the events *first-episode acquired infection*. For instance, let $Q = \{q_0, q_1, ..., q_N\}$ (the subscript denotes the observations) be the values of the q-th new explanatory variable (e.g., Approximate entropy for Hearth Rate records) and let be $Q^0 \subset Q$ and $Q^1 \subset Q$ the sets with observations conditioned, respectively, to the class "not-infected" and "first-episode acquired infection." Note that $Q^0 \cap Q^1 = \emptyset$ and $Q^0 \cup Q^1 = Q$. Then, we apply the U-test to investigate if the two distributions, followed by the values of both Q^0 and Q^1, are the same. The rejection of the null hypothesis occurs with a p-value lower than 0.05; when it is the case, it is equivalent to saying that the new explanatory q shows a significant difference between the two classes of events. Hence, when applying this test to all the new explanatory variables, we shall only select those rejecting the null hypothesis.

4.6.3 Comparison of MLR models

In this section, we shall discuss the predictive performance of four models: 1-D CNN, a MLR model fitted on the traditional ICU covariates, a MLR model fitted on the new covariates extracted via the analysis of the most salient 8-hour patterns (i.e., the *new features* obtain after visualizing the saliency maps obtained via SMOE scale) and a MLR fitted to both the covariates of the last two MLR models (namely, both the traditional ICU covariates and the *new features*). We used the AUROC score to evaluate the predictive performance of all these methods; errors are estimated by SEM. The results at three specific landmark times (i.e.,(days 3, 7, and 10) are summarized in figure 4.20. Focusing our analysis only on three exact moments of the ICU stay (i.e., day 3, 7, and 10), we observed that the 1-D CNN starts with AUROC 0.60 ± 0.02 at day 3, AUROC 0.62 ± 0.03 at day 7, and achieve the highest performance with AUROC 0.78 ± 0.03 at day 10. In contrast with this, the MLR model embedded with the new explanatory variables we extract via the SMOE scale cannot alone be very predictive; in fact, we observed AUROC 0.61 ± 0.01 at day 3, 0.61 ± 0.01 at day 7, and 0.70 ± 0.01 at day 10. Similar to this, the MLR model embedded with the traditional ICU clinical covariates (both time-dependent and fixed) keeps a low predictive power, especially at the beginning of the ICU stay, i.e., we observed that AUROC 0.63 ± 0.01 at

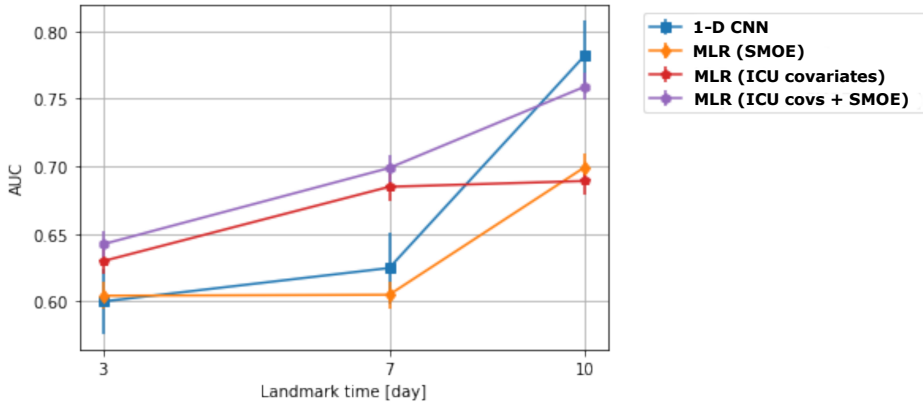


Figure 4.20: AUC scores (a.k.a. AUROC score) for four different classifiers at different landmark times (days 3, 7, and 10): 1-D CNN (blue), MLR embedded with covariates extracted via SMOE scale (orange), MLR with embedded with traditional covariates extracted from the ICU setting (red), and MLR including both the two sets of explanatory variables (i.e., the traditional ones and the those extracted via SMOE scale)(purple). On the x-axis, the time-domain (landmark times), and on the y-axis, the AUROC scores. Error bars are evaluated via SEM.

day 3, AUROC 0.67 at day 7, and AUROC 0.69 at day 10. Finally, the MLP model, including both the variables extracted via the SMOE scale and the traditional ICU covariates, reveals how incorporating the *new features* can increase either marginal or substantially the predictive power of the conventional ICU covariates. Indeed, we observed AUROC 0.64 ± 0.01 at day 3; a marginal increase is achieved on day 7, where AUROC 0.70 ± 0.01. Finally, a relevant contribution is present on day 10, where AUROC is 0.76 ± 0.01. On days 3 and 7, we observed an increment of order 0.02 (points of AUROC) compared to the MLR fitted on the traditional ICU variables only. On day 10, our analysis highlighted a more consistent increment equal to 0.08 (points of AUROC).

We tried to interpret the situation described in figure 4.20. On day 3, the correct diagnosis of ICUFAI for a patient might still be premature since the ICU therapy has not progressed enough; as we expected, all the models cannot retrieve some patterns that straightforwardly discern the two classes of events (i.e., *onset of ICUFAI* and *No infection*). The information contained in the data at this precise moment of the ICU stay might therefore turn out to be too dispersive; both 1-D CNN and MLR model embedded with traditional ICU clinical predictors did non show a AUROC score higher than 0.65. As a result, the extraction of the new explanatory variables via the SMOE scale is affected by the poor intrinsic discriminative power of data; the relative contribution that could arise from them is also weak.

After one week from the admission in the ICU (when ICUFAI episodes are overly likely to occur, see section 4.2.2), the MLR models fed with traditional ICU covariates can better discern the onset of acquired infections (AUROC 0.68 ± 0.01). Whereas, 1-D CNN still presents poor accuracy (AUROC 0.63 ± 0.02). In this case, the contribution arising from the analysis of the activation of the 1-D CNN can only give a marginal increase in the AUROC (i.e., 0.01) to the MLR with traditional ICU predictors.

On day 10, we can see that traditional ICU covariates are not a satisfying set of predictors. However, the addition of extra information arising from the activity of the 1-D CNN (see section

1.1.3) brings a substantial increase of 0.06 points in AUROC (once again, we compared the MLP fitted on traditional ICU predictors with the MLP fitted on both traditional ICU predictors and the *new features*).

4.6.4 Data-driven clustering of salient patterns

To complete this analysis, we focus our attention on the following point: *how can the activity of pattern recognition within the 1-D CNN model be connected with some real scenarios of the classification of patients when the ICUFAI episode is approaching? Which proper characteristics of EHR can be captured by the 1-D CNN model, when an acquired infection is approaching?* For this purpose, we explored, once again, the activity of the 1-D CNN model by means of the SMOE scale. Then, we grouped the most salient pattern in some *data-driven clusters*.

In order to make this analysis simple as well as interpretable, we avoid using clustering methods for Time-Series (such as K-means or K-shape (Paparrizos and Gravano, 2015)) that would not succeed in this case. Both the high complexity of Time-Series instances and the heterogeneity of samples would dramatically affect this attempt to find a finite number of well-separate clusters by minimizing the *within-cluster variance*. Instead, we tackled this problem through this easier strategy:

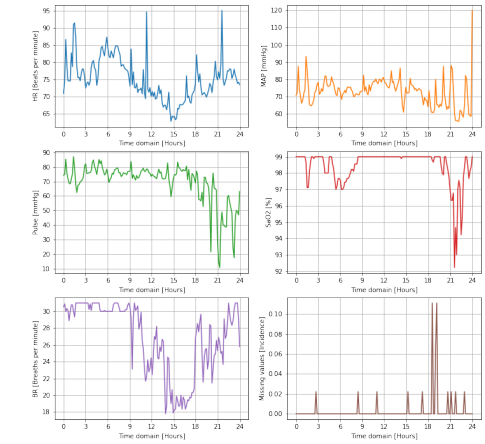
1. Similar to the approach of section 4.6.1, after training the 1-D CNN binary classifier, we applied the SMOE scale to find, in the test set, the most predictive patterns with amplitude 8 hours; see 4.21.

2. We considered four clinical critical conditions, i.e., *tachycardia*, *hypotension*, *desaturation*, and *hyperventilation* (see Table 4.7).

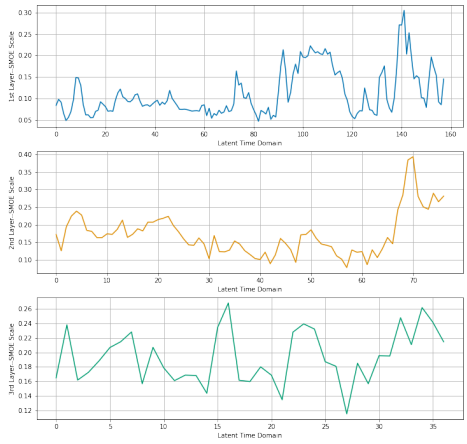
3. We evaluated the mean value of each 8-hour pattern extracted via the SMOE scale. Depending on the values obtained, we checked the presence of each clinical critical condition introduced in the previous step. Each clinical critical condition is represented as one of the four dichotomous items of a binary array of length four. For each item of that array, we assign the value 1 when one of the four conditions defining one precise critical clinical situation is met; otherwise, 0. The binary array of length four is therefore constructed to describe 16 different clinical situations of interest.

4. The ensemble of all these steps represents our *data-driven clustering*.

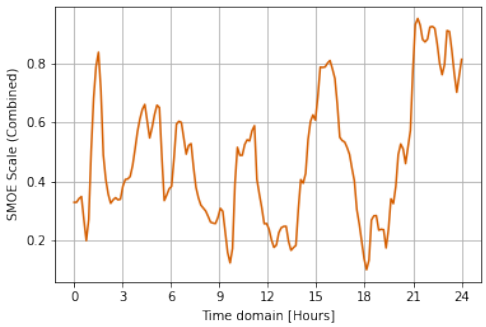
Thus, we described the skill of the 1-D CNN in recognizing ICUFAI episodes by means of a data-driven clustering based on the clinical conditions of the patients. Interestingly, our data-driven can be represented as clustering on a 16-node graph, where each one describes one precise critical condition (e.g., tachycardia and hyperventilation, hypertension and tachycardia, hypertension, and so on) that could predict the approaching of one ICUFAI episode. The full description of all the 16 data-driven clusters (classes) is reported in Table 4.7, while the criteria adopted to highlight the presence of the critical clinical conditions are listed in Table 4.8. The data-driven clusters reflect the main symptoms of SIRS (Chakraborty and Burns, 2019). In figure 4.22, a schematic graph representing the 16 nodes hyper-cube is shown. Conditions like tachycardia, hypotension, and hyperventilation are quite spread in the ICU and hospitalized patients and are usually mentioned in general guidelines for the ascertainment of SIRS (Comstedt et al., 2009). For the criteria reported in Table 4.8 we referred to Comstedt et al. (2009); in specific for Desaturation, we referred to (Hafen



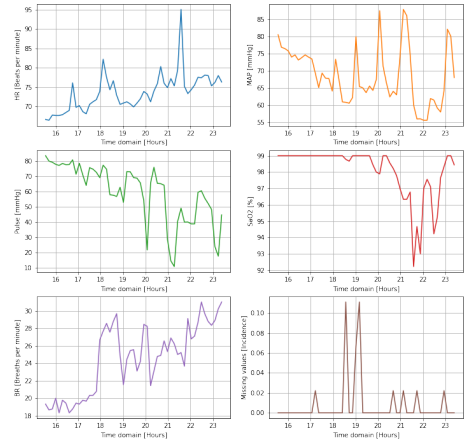
(a)



(b)



(c)



(d)

Figure 4.21: Schematic example of the extraction of the most salient patterns in the 24-hours Time-Series instances: (a) Example of Time-Series instance, (b) SMOE scale applied on each activation feature map of the 1-D CNN, (c) Averaged saliency map (weighted average of SMOE scales on individual hidden layers), (d) extraction of the most salient interval of (a).

Class	Data Driven Cluster (Critical Clinical Conditions)
0	None
1	Tachycardia
2	Hypotension
3	Hypotension, Tachycardia
4	Desaturation
5	Desaturation, Tachycardia
6	Desaturation, Hypotension
7	Desaturation, Hypotension, Tachycardia
8	Hyperventilation
9	Hyperventilation, Tachycardia
10	Hyperventilation, Hypotension
11	Hyperventilation, Hypotension, Tachycardia
12	Hyperventilation, Desaturation
13	Hyperventilation, Desaturation, Tachycardia
14	Hyperventilation, Desaturation, Hypotension
15	Hyperventilation, Desaturation, Hypotension, Tachycardia

Table 4.7: All 16 critical clinical conditions (classes) proposed with the data-driven clustering approach.

Critical Condition	Criterion
Tachycardia	Hearth Rate $\geq$90 beats per minute
Hypotension	Arterial Blood Pressure (mean) $\leq$ 80mmHg
Desaturation	$SaO_2 \leq 95\%$
Hyperventilation	Breath Rate $\geq$ 24 breaths per minute

Table 4.8: Criteria used in the data-driven clustering approach.

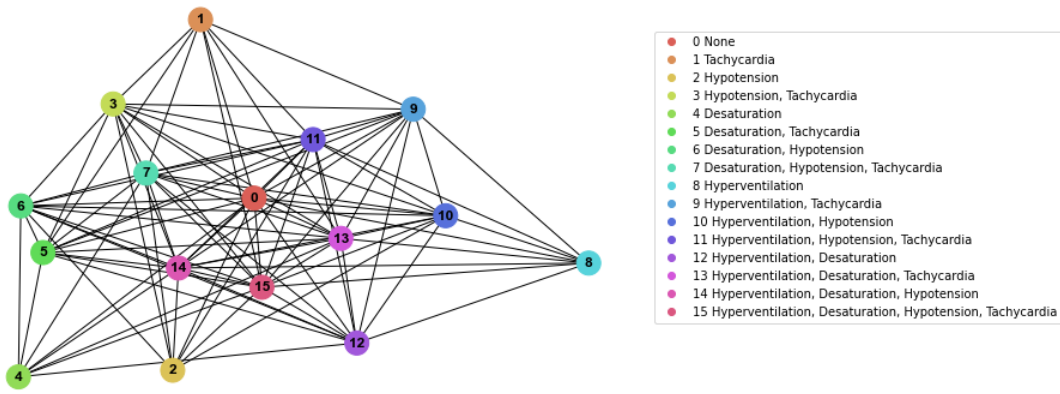


Figure 4.22: Scheme of the hyper-cube whose nodes represent the 16 data-driven clusters.

and Sharma, 2018).Therefore, we motivate this choice of using a *data-driven clustering* approach as the attempt to connect the activity of 1-D CNN in terms of some common clinical scenarios in the ICU.

To show some results, we worked within the same framework used in section 4.6.3. We considered the 1-D CNN model with training samples restricted to landmark time days 3, 7, and 10. Then, we extracted the 8-hour most salient patterns (via SMOE scale) from the test sets and made a clustering according to the data-driven clustering method proposed here. Histograms with the relative frequencies of the 16 data-driven clusters are shown in Fig. 4.23. We used the U-test (see section 4.6.2) to provide a significant difference in the description provided by the data-driven clusters in discerning ICUFAI episodes.

For day 3 (see figure 4.23 a and 4.23 b), two-sample Kolmogorov-Smirnov test (Hodges, 1958). reveals that the sample distributions of events ”not-infected” and ”first-episode acquired infection” are not significantly different. This implies that the data-driven clustering approach shows no substantial difference between a patient with ICUFAI and the others. In addition, this result supports the low prediction of 1-D CNN (see section 1.1.3) which cannot be connected to the occurrence of some critical clinical conditions anyhow. As argued in section 4.6.2, the intrinsic high dispersion of EHR at day 3 impedes the 1-D CNN from recognizing an ICUFAI episode following generally accepted criteria.

In opposition to this, we can observe a completely different scenario on both days 7 and 10 (see figure 4.23 d-f), where the null hypothesis of the two-samples Kolmorogov-Smirnov test is rejected (once again, for the 16 data-driven clusters, we compared the sample distribution of the ICUFAI population and all the other instances that do not represent an ICUFAI event). Hence, this shows that different clinical conditions could represent an essential feature of the patterns that the 1-D CNN model captures during the learning phase. For instance, for infectious events, at day 10, the prevalence of at least one of these 16 conditions is around 94%, while 79% at day 7; see figure 4.23 d-f. Precisely, on day 10, events with hyperventilation correspond at 70% of samples, and in combination with tachycardia 23%. While a day 7 tachycardia is much more relevant and occurs in 50% of infectious samples.

A deeper insight into the ICUFAI events on days 7 and 10 reveals that the signals containing

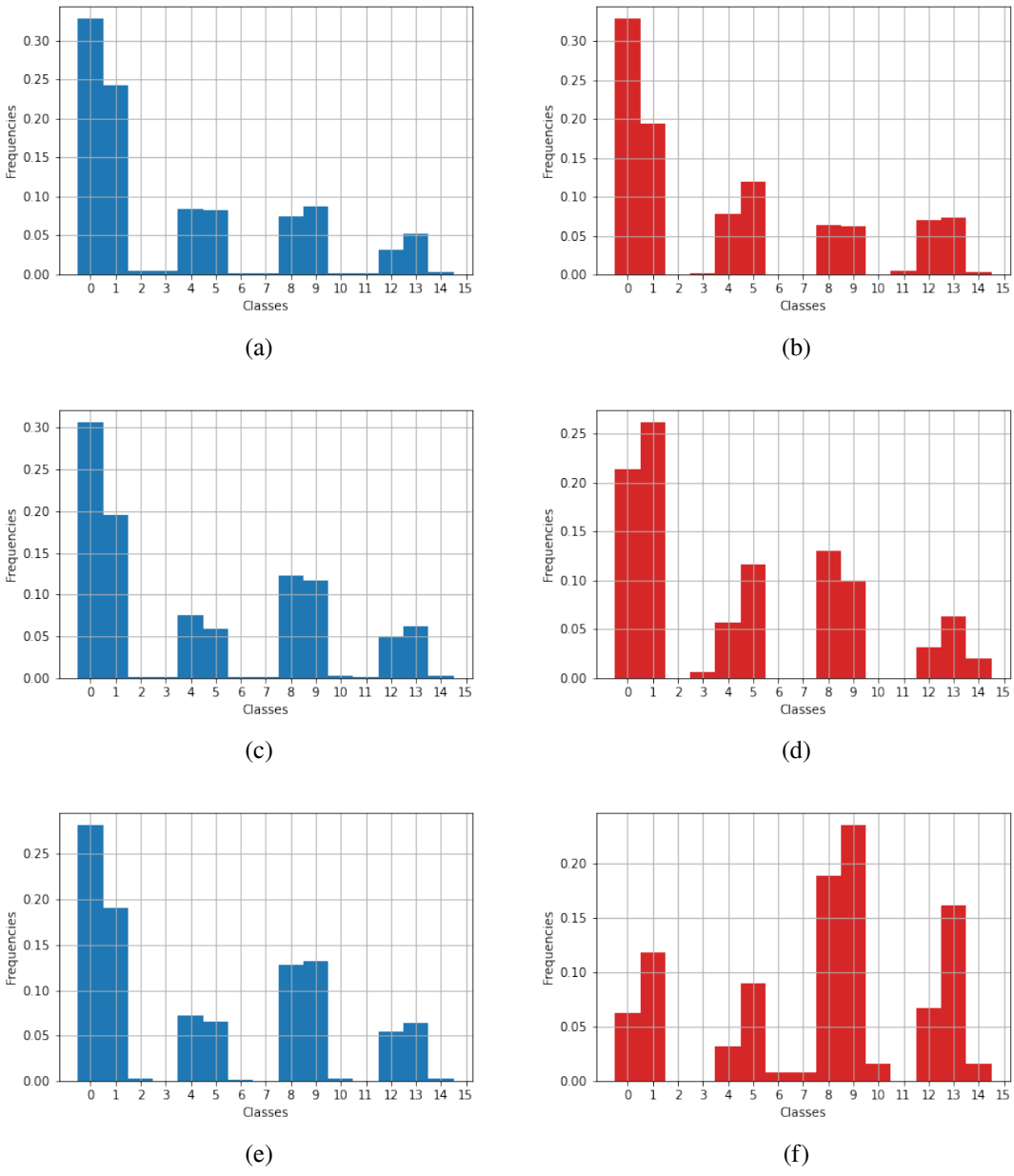


Figure 4.23: Histograms the data-driven clustering approach. Bins on the x-axis represent the 16 classes based on ill physiological states. Blue histograms concern the non-infected population, whereas the red ones the infectious events. 1-D CNN trained on day 3 is described by (a) and (b), on day 7 by (c) and (d), and on day 10 by (e) and (f).

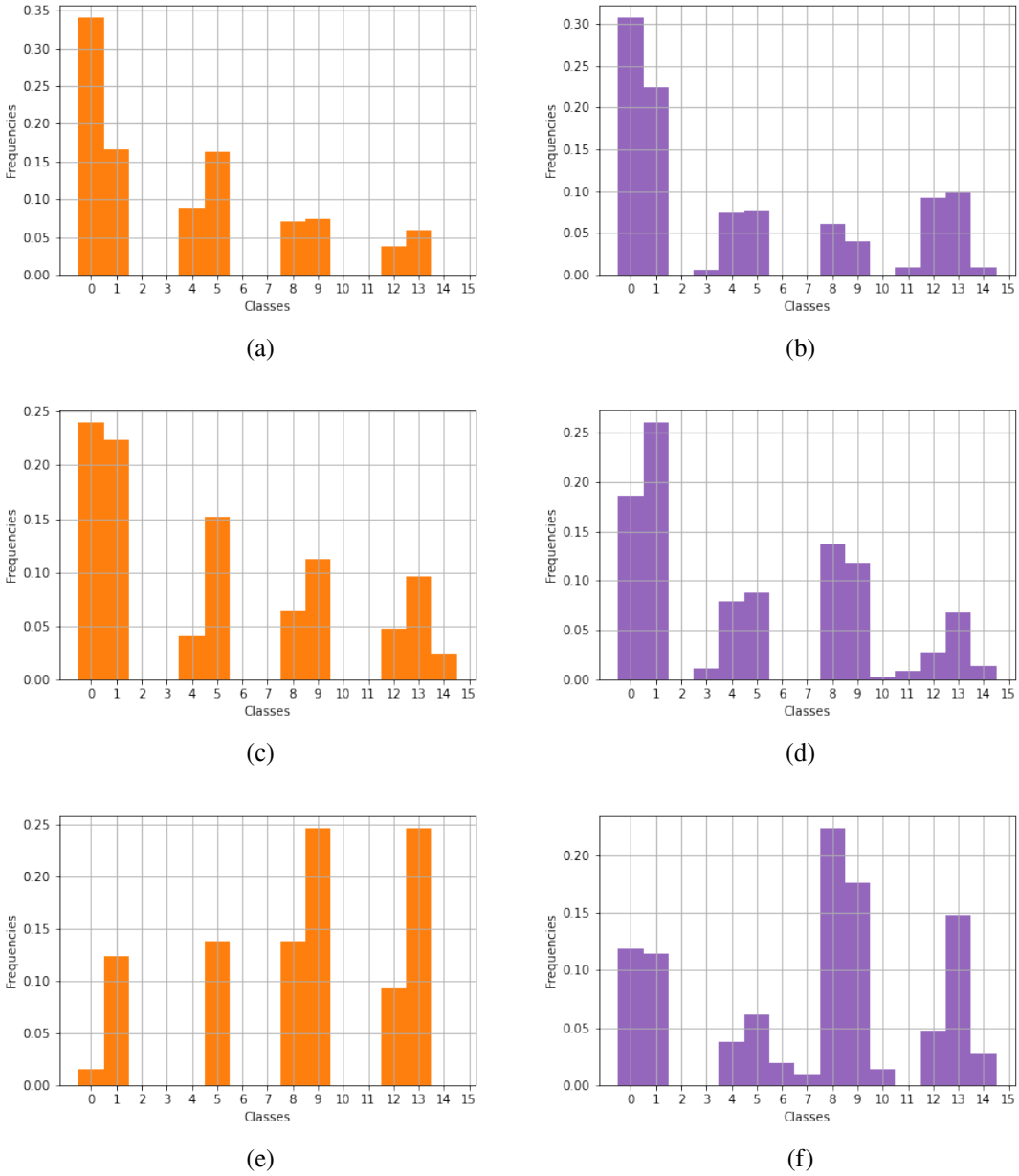


Figure 4.24: Histograms of the data-driven clustering approach for patients with ICUFAI only. Bins on the x-axis represent the 16 classes based on ill physiological states. Orange histograms concern the population that will experience an ICUFAI in the next 24 hours, whereas the purple ones concern infectious events that occurred in an interval lower than 24 hours. 1-D CNN trained on day three is described by (a) and (b), on day seven by (c) and (d), and on day ten by (e) and (f).

the ICUFAI event are significantly different than the non-infected ones. We recall that the Time-Series instances representing ICUFAI events are extracted in two different ways; we considered either 24-hour Time-Series instances laying in the 24 hours before the event or the Time-Series instances containing the event itself (see section 4.3.1). In this case, we still visualized and studied the histograms of the data-driven clusters. When considering the patterns of the Time-Series instances containing an ICUFAI event, we observed a significant difference for the most salient patterns of the Time-Series instances that are not labeled as an ICUFAI event. We used the two-samples Kolmogorov-Smirnov, and we observed that the null hypothesis is rejected when comparing the histograms on both days 7 and 10; we compared figure 4.23 c and e with, respectively, figure 4.24 d and f. The instances extracted 24 hours before one ICUFAI event turned out to be significantly different from non-infectious situations only on day ten only. Kolmogorov-Smirnov test confirmed this result after rejecting the null hypothesis on the equivalence of the sample distribution of the 16 data-driven clusters; see figure 4.24e.

4.7 Future directions: Multi-Branch CNN model for forecasting a high CRP increase

C-Reactive Protein (CRP) is a protein composing the sanguine fluid whose concentration often represents a response to bacterial and viral inflammations (Black et al., 2004); for example, the concentration of CRP in blood samples is a laboratory value used to ascertain the presence of a SIRS (Comstedt et al., 2009) which is a common symptom in septic ICU patients. A higher concentration of CRP is also frequent in other clinical conditions, such as ill patients with cancer (Srimuninnimit et al., 2012). Nevertheless, when considering infectious states in ICU, it has also been shown that the concentration of CRP represents a suitable biomarker for clinical purposes (e.g., diagnostic of infectious states in ICU) (Reny et al., 2002).

As mentioned in the introduction of this Chapter, the forecast of CRP levels in ICU patients represents a challenging and different approach to modeling the early identification of ICUAI. The attention posed to the prediction of CRP concentration has already shown promising results; traditional linear models (e.g., ARIMA and ARMA) have shown that predictions based on the continuous monitoring of CRP level can improve the early recognition of ICUFAI events (Póvoa et al., 2006; Dorraki et al., 2019). In addition, ANN methods have recently emerged as a finer methodology with significantly reduced prediction error in forecasting CRP concentration for medical purposes (Dorraki et al., 2019).

In this section, however, we shall present another way to model CRP evolution. Unlike the DL-based regression proposed in Dorraki et al. (2019), here, we shall implement a DL-based classification to forecast a high increase in the CRP level. More specifically, we want to classify all those conditions in ICU that are not infectious at some moment of the ICU stay (i.e., a patient with a low CRP level at some time τ of his/her stay in the ICU) and are suspected to be infectious in a horizon time of 24 hours (i.e., the same patient has now a much higher CRP level at some time τ + 24 hours). Therefore, we implemented a CNN-based binary classifier and made its prediction explainable via a DNN. DNN are a class of algorithms designed to reconstruct the input data from the hidden layer of a CNN-based model. DNN are often utilized to reveal the connection between the input data and the patterns captured at any hidden layer of a CNN model. The visualization and analysis of those patterns enable one to understand which input data feature supports the high

predictive performance of a CNN classifier. Similarly to work done in section 4.4 and 4.5, the classification task is based on the recognition of the EHR data introduced in section 4.3.1. As a result, DNN will be utilized to reveal which clinical conditions are connected with the forthcoming occurrence of a suspected ICUFAI event. To simplify the implementation of the DNN, we opted for a 1-D Multi Branch binary classifier; more details will be given in 4.7.1.

The main feature of DNN consists of backpropagating the activation feature maps of a CNN model up to its input layer (Zeiler et al., 2010). Note that DNN has no learning phase, even though it has the same layered structure as CNN. Thus, a DNN is composed of a sequence of not-trainable layers that mimic the inversion operation of CNN hidden layers. For example, we recall that 1-D CNN (see Section 1.1.3) are mainly composed of convolutional and max-pooling layers; the operation executed in those layers are, technically, not always invertible. In such a particular case, the DNN model is conceived to approximate the invert those two types of layers. More specifically, when attempting to invert the max-pooling layer, the *unpooling layer* (Zeiler and Fergus, 2014) offers a good approximation of the original activated features maps; the unpooling operation is often accomplished by means of linear interpolation. *Transposed convolutions* (Aggarwal et al., 2018) represents an excellent strategy to invert the convolutional layers; in practice, one applies a convolutional layer whose filters (i.e., the convolutional masks) are a transposed version of the convolutional filters of the convolutional layer one aims to invert. A detailed description of the DNN implemented to invert the Multi-Branch 1-D CNN will be presented in section 4.7.3.

Hence, the methodology that we present in this section can be summarized as follows: we first train a 1-D CNN Multi Branch model with the EHR presented in section 4.3.1. The aim is to forecast a critical increase of the CRP level in those ICU patients still at risk of experiencing a possible infectious condition in ICU (section 4.7.2). Next, DNN is used to invert the pattern recognition activity of the 1-D CNN Multi Branch model. The statistics of the patterns reconstructed via DNN are then visualized and analyzed to individualize which are the main clinical conditions associated with an abnormal increase of CRP (section 4.7.4). This study is based on the fact that the forecast of CRP level represents another valid approach to improve the detection of the symptomatology associated with ICUFAI. Similarly to work presented in section 4.4, the work we present here has the scope of providing another example of how the extraction of interpretable information from the activity of CNN models can potentially be exploited to improve the early detection of ICUFAI episodes.

4.7.1 Multi-Branch 1-D CNN

With the term Multi-Branch 1-D CNN, we refer to a particular feed-forward CNN composed of a set of separated one-node output 1-D CNN (see section 1.1.3) models connected in series. The output of each 1-D CNN branch is then conveyed into one single *Dense layer* with a single output; an illustration of such an ANN architecture is shown in figure 4.25. Each branch of the Multi-Branch 1-D CNN is therefore composed of a 1-D CNN model fed by a single Time-Series feature, i.e., Time-Series instances, containing one 1-D Time-Series. We opted for a Multi-Branch 1-D CNN because it allows a more performative inversion via DNN; Multi-Branch 1-D CNN turns out to be advantageous when one is interested in having convolutional and pooling layers to be devoted to the analysis of single Time-Series features. Because the branches of the Multi-Branch 1-D CNN are specialized in the analysis of single Time-Series features, the inversion via DNN turns out to be more precise, and the introduction of notable artifacts is prevented during the reconstruction of

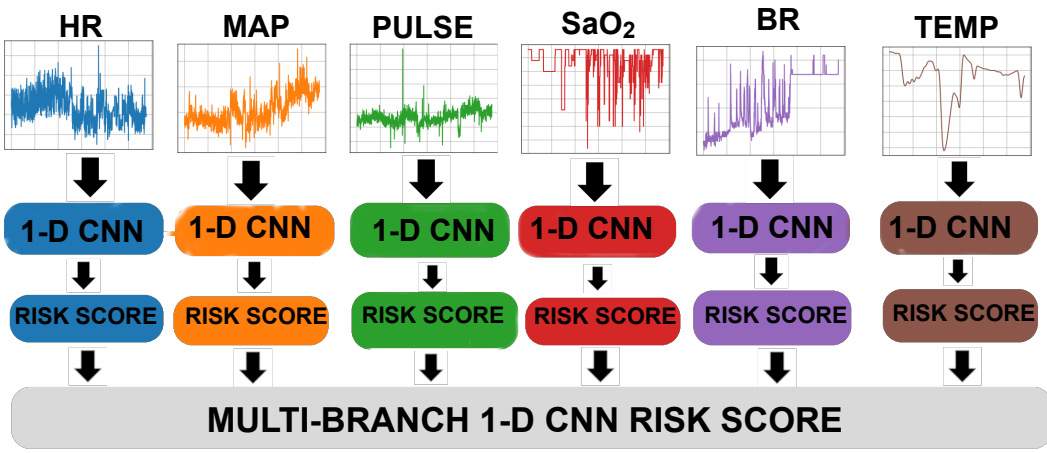


Figure 4.25: Scheme of the Multi-Branch 1-D CNN model.

the input Time-Sereis instances. When feeding this model with the EHR data of section 4.3.1, the output of each branch represents a sort of risk score of infection coming with the analysis of single Time-Series features such as Heart Rate, Arterial Blood pressure, Pulse, Saturation, Breath Rate, and (body) Temperature. The final output of the Multi-Branch 1-D CNN is still a weighted risk score arising from the contribution of single predictors (Time-Series features) analysis.

Each 1-D CNN branch of the model is therefore structured as follows

- The data are propagated through *operational blocks* that are composed by the sequence of one *convolutional layer*, followed by a *non-linear activation function*, a *max-pooling layer*, and a *gaussian dropout layer* (see section 1.2). The architecture of the operational blocks is the same for each 1-D CNN branch of the model.
- Convolutional layers have 16 filters with convolutional masks of amplitude 3. The activation function is the hyperbolic tangent. Max-pooling layers have a pooling size equal to 2, while the gaussian dropout layer injects 1-mean multiplicative gaussian noise with a standard deviation equal to $\sqrt{\frac{0.25}{0.75}} \simeq 0.57$ (it is equivalent to a dropout layer with dropout rate equal to 0.25; see section 1.2)
- The number of operational blocks in each branch is equal to three, so each 1-D CNN branch has a deepness equal to three.
- The final latent representation of the operational blocks of each branch is then flattened (via Flatten Layer) and conveyed into a Dense layer with one output node and sigmoidal activation function. Each of these outputs corresponds to the output of each 1-D CNN branch.
- The outputs of each branch are propagated into a Dense layer with one output node and sigmoidal activation function. The Dense layer of this stage returns the final output of the Multi-Branch 1-D CNN model.
- The loss function is the binary-cross-entropy (see section 1.1.1); Adam algorithm is the optimizer.

This architecture has been chosen after selecting that choice of hyper-parameters giving the highest AUROC value; we used the one-held-out split of the instances (64% training, 16% validation, and 20% test set).

4.7.2 Forecasting an increase CRP concentration via Multi-Branch 1-D CNN

We want to apply now the Multi-Branch 1-D CNN to classify the EHR data associated with an increase of the CRP level of ICU patients that have not experienced any inflammatory state. Thus, we aim to classify an increase of CRP level in the next 24 hours for those patients that have been admitted to ICU but are still at risk of worsening their clinical condition (e.g., they might develop an inflammatory state that could degenerate in an infectious or septic state). We recall that we trained the model with the EHR data presented in section 4.3.1; so, we fed the Multi-Branch model with the 24-hour Time-Series instances used in section 4.4. The instances are labeled according to the CRP values at different moments of the ICU stay. Note that CRP is one of the low-frequency covariates reported in table 4.2; this allowed us to associate a CRP value with the Time-Series instances selected to train the model. Since we possess the records of the continuous monitoring of the CRP values for all patients under examination, we can match each Time-Series instance with the corresponding exact value of the level of CRP along the time domain (i.e., the ICU stay of one ICU patient). Thus, the labeling of the instances is the following

1. We fix two specific quantities: we denote them as the *threshold level* of CRP (with notation CRP_0) and the *minimal abrupt increase* of CRP (denoted as Δ_{CRP_0}).

2. Each Time-Series instance is recorded at various times of the ICU stay of a patient; we denote a generic time with τ. Thus, for any patient, we consider the set of observations *EHR at time τ, CRP at time τ, and CRP at time τ+ 24 hours*. Similarly to section 4.3.4, we used a landmarking approach with a stride of 8 hours (in other words, all τ times are equispaced along the whole stay of a patient and form subintervals of 8 hours).

3. When evaluating the CRP level at time τ and τ+24 hours, we denote these two quantities as $\text{CRP}(\tau)$ and $\text{CRP}(\tau + 24 \text{ hours})$.

4. We evaluate the corresponding increasing of CRP level at time τ+24 hours; we denote this observable as $\Delta_{CRP}(\tau + 24 \text{ hours}) = CRP(\tau + 24 \text{ hours}) - CRP(\tau)$.

5. We label as an *event of interest* (case group) all the Time-Series instances whose corresponding CRP value meets
$$\begin{cases} \text{CRP}(\tau) \leq 10 \text{ mg/L}, \\ \Delta_{\text{CRP}}(\tau + 24 \text{ hours}) \geq 40 \text{ mg/L}. \end{cases} \tag{4.29}$$
All remaining Time-Series instances that do not meet that criterium are placed into the control group.

We stress the fact that CRP_0 is a variable constructed to denote a physiological response to a viral or bacterial inflammation; we combined it with Δ_{CRP_0} to label all patients that might not have experienced inflammation in ICU yet (i.e., low values of CRP), but might do it in the next 24 hours (i.e., a high value of Δ_{CRP_0}). Both the thresholds are based on the fact that a CRP level of $10 - 40$ mg/L is typical in bland inflammatory processes or viral infections (Nehring et al., 2017). Still, a

value around 4 mg/L is a good warning sign for detecting mild inflammation. Hence, we modeled the probable presence of a non-severe inflammation (CRP lower than 10 mg/L) that might worsen into a more serious clinical condition in the next 24 hours of one ICU stay. We considered a total of 31494 Time-Series instances (extracted as in 4.3.1); the portion of the population labeled as an event is barely 2%. The Multi-Branch 1-D CNN we used to classify these data is shown in section 4.7.

When pre-processing the Time-Series Instances, all Time-Series instances have been standardized with zero-mean and unitary variance. To train the model, we used a *one-held-out* split of the data; thus, 80% training set and 20% test. In addition, we split the training set again in 80% training set, and 20% validation set again. Hence, the split of the original data can be summarised as 64% training set, 16% validation set, and 20% test set. We used the training set during the learning phase, and the validation set was used for model selection and parameter tuning (a sort of test set uniquely used for designing the best choice). The test set was used to evaluate the accuracy of the model. The one-held-out strategy is used to tune the best hyperparameter configuration of the network, i.e., that configuration reaches the highest AUROC score. section 4.7.1.

After establishing the best hyperparameters, we use that configuration to train the model. Thus, we validate it via 5-fold cross-validation; after training the Multi-Branch 1-D CNN, we obtained an overall AUROC equal to 0.80 and SEM equal to 0.01.

To determine the goodness of the predictive power of the Multi-Branch model, we made a comparison with a LSTM-based classifier. The choice of utilizing LSTM (see section 1.1.4) is motivated by the fact that Dorraki et al. (2019) proposed an LSTM neural network to show that ANN is a promising methodology in CRP level forecasting. Note that Dorraki et al. (2019) implemented an LSTM to solve a regression task; because we are interested in solving a classification task, we readapted the LSTM network of Dorraki et al. (2019) to our needs.

Thus, the LSTM classifier we implemented has the following structure:

1. EHR data are propagated throught an LSTM layer (see section B). The number of output units is equal to 128. The LSTM layer has thus the scope of giving a flatter latent representation of the input Time-Series instances.

2. The latent description given by the LSTM layer is then propagated through a *Dense layer* with one output and ReLU activation function.

3. A further *Dense layer* with one node output and sigmoidal activation function is located underneath the *Dense layer* with ReLU activation function. Note that the former (i.e., the dense layer with sigmoidal activation function) is equivalent to a *Platt scaling* (Platt et al., 1999). Also, as known, the sigmoidal activation function returns an output value between 0 and 1; this fact enables one to correctly use the *binary cross-entropy* as a loss function. The optimizer is the Adam algorithm Kingma and Ba (2014a).

The best hyperparameters are chosen after searching the highest AUROC value on a fine grid of possible parameters. To evaluate all the hyperparameters configurations, we used the one-held-out approach; and so we used 64% of instances in the training phase, 20% in the validation (i.e., we used them in the early stopping criterium; see 1.2), and 20% to test the model. The AUROC is evaluated utilizing the test data only.

After selecting the best configuration, we validated the model via 5-fold cross-validation; we obtained an overall AUROC equal to 0.61 and SEM equal to 0.01. This result shows that an LSTM

approach turns out to be unsuited to classify a high increase of CRP in ICU patients. Still, the results obtained with the Multi-Branch 1-D CNN revealed that a CNN-based approach can better distinguish when a non-critical or absent inflammation state in ICU might worsen into a more severe inflammation in a horizon time of 24 hours.

4.7.3 A brief introduction to DNN

In this section, we present the structure of the DNN model used to invert the Multi Branch 1-D CNN. Usually, the inverse-mapping of a DNN is designed to invert the typical sequence of operations of the CNN hidden layers, i.e., convolutional, non-linear activation function, and max-pooling layers. As known, CNN model follows the scheme convolution $\rightarrow$ non-linear activation function $\rightarrow$ max-pooling layer (see Sec 1.1.3); as a result, DNN is characterized by the inverse scheme, i.e., unpooling $\rightarrow$ inverse activation function $\rightarrow$ transposed convolutional layer. This sequence of operations is repeated until the original input space is covered; see figure 4.26.

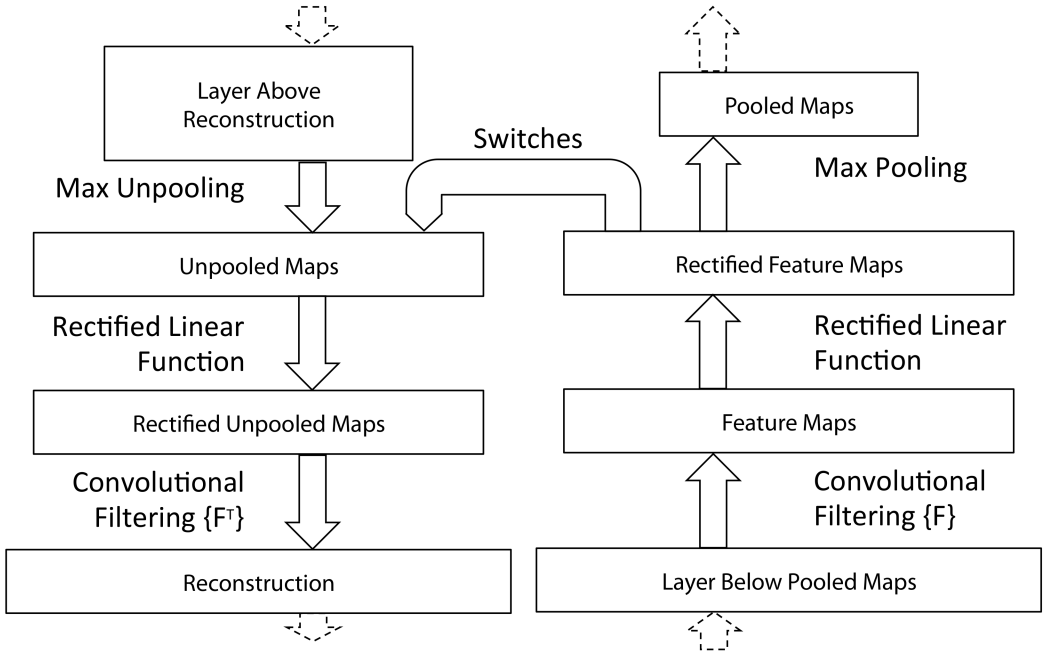


Figure 4.26: Example of DNN model. This image was taken from (Zeiler and Fergus, 2014)

More specifically, the *unpooling layer* is designed to invert the max-pooling layer; in implementing this layer, we follow the approach of Zeiler and Fergus (2014). Thus, after forward propagating the input data through the CNN hidden layers, we first record the exact positions of the points (pixels) of the activated feature map, which will be selected by the max-pooling layer. Then, when unpooling the max-pooled activated feature maps, we reassign the actual value to the recorded positions while all remaining positions are imputed with a null value. This particular approach in unpooling the max-pooled activated feature maps was proposed by Zeiler and Fergus (2014), and we shall refer to it as unpooling layer via *switch variables*; an illustration is shown in figure 4.27.

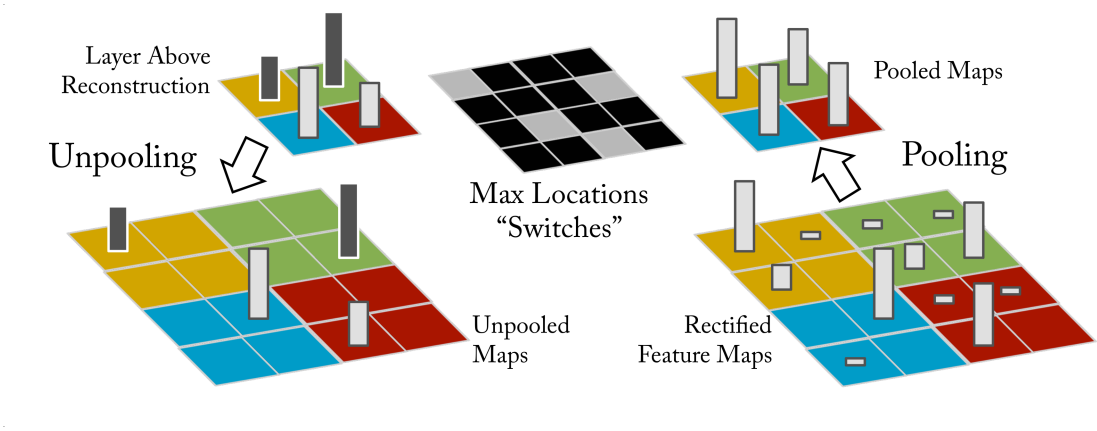


Figure 4.27: Example of unpooling layer via *switch variables*. This image was taken from (Zeiler and Fergus, 2014)

The mechanism of switch variables helps preserve the original activated feature maps' structure and avoid introducing artifact patterns in the final reconstruction.

The inversion of the activation function depends on the functional form of the function. For instance, since both are monotone and bijective, activation functions such as *hyperbolic tangent* or *softplus* can be straightforwardly inverted. The ReLU function cannot be inverted theoretically, even though it is often inverted by means of the Relu function itself, provided the fact that the ReLU activation map has been implemented after all convolutional layers (Zeiler and Fergus, 2014).

Convolutional layers are inverted via transposed convolutional layers; the latter is identical to the former, except that the filters are the transposed versions of the filters of the latter (Zeiler and Fergus, 2014). Although the transposed version of the convolutional filters does not coincide with a deconvolution operation (i.e., the inverse operation of convolutions), the transposed convolutions enable one to reconstruct the exact spatial resolution that a deconvolution would generate. However, transposed convolutions are naturally developed by convolutional autoencoders (i.e., a type of ANN designed to encode unlabeled input data via convolutional operators efficiently) when decoding the latent representation of the deepest layers (Aggarwal et al., 2018); this also motivates the usage of transposed convolutional layers to attempt to reconstruct the patterns encoded via convolutional layers.

Hence, the running of a DNN model is similar to the forward propagation of CNN model but does not involve any learning phase. In particular, the activated feature maps of a CNN are propagated through the inverse layers constituting the DNN to obtain a possible reconstruction of the input data that gave rise to the activated feature maps in question. For example, one first forwardly propagates the instance X through a CNN model and generates the activated feature maps $\tilde{X}$; thus, one propagates $\tilde{X}$ through the DNN model to obtain a reconstruction $\hat{X}$ of the original instance X. A study of the features of $\hat{X}$ should be informative about the activity pattern recognition of CNN; that is, what patterns the model captures during its learning phase.

4.7.4 DNN applied to Multi-Branch 1-D CNN

In this section, we present an application of DNN (see section 4.7.3) to the Multi-Branch 1-D CNN (see section 4.7.1) trained to classify clinically relevant increase of CRP (see section 4.7.2). The main concern of this section is the statistical analysis made on the patterns reconstructed via the DNN. The scope of such an analysis is to outline some significant macroscopic characteristics that support the prediction made by the Multi-Branch 1-D CNN. We recall that the Multi-Branch 1-D CNN was fed with EHR data; in this framework, the analysis of the reconstructed pattern aims to reveal the main macroscopic feature of EHR that might be associated with a particular critical physiological dynamics in ICU patient.

Thus, the analysis made on the reconstructed patterns consists of

1. All test set instances are propagated through the Multi-Branch 1-D CNN. DNN is then used to invert the activated feature maps of the Multi-Branch 1-D CNN; as a result, one obtains a reconstruction of the input data of the Multi-Branch 1-D CNN.

2. The EHR reconstructed via DNN are structured as the Time-Series instances propagated thought the Multi-Branch 1-D CNN; an example of reconstructed instance is shown figure 4.28. Note that in that figure, the reader can visualize the original instance and its reconstruction obtained by the DNN method.

3. From the reconstructed EHR, we extracted some summary statistics such as mean value, standard deviation, mean change, minimum value, and maximum value per each one of the Time-Series features. Also, the extraction of the summary statistics is made on four precise sub-intervals of the time domain, i.e., at intervals 0-6 hours, 6-12 hours, 12-18 hours, and 18-24 hours. We recall that each Time-Series instance has an amplitude of 24 hours and, only in this case, contains the evolution of the EHR of one patient from the midnight of a calendar day up to the midnight of the day after. Hence, when extracting the summary statistics mentioned above, we first chunked each Time-Series instance in four precise sub-intervals; we evaluated and extracted these summary statistics on each chunk. For example, when evaluating the mean value of Heart Rate, instead of considering an overall value over the 24 hours domain, we extracted the same statistics at intervals 0-6 hours, 6-12 hours, 12-18 hours, and 18-24 hours.

4. To reveal whether the extracted statistics were significant or not, we used the non-parametric U-test (the same approach used in section 4.6.1).

Before utilizing DNN, we need to specify that we clustered the activated feature maps according to the output value returned by the Multi-Branch 1-D CNN. As stated before, the Multi-Branch 1-D CNN has been constructed with the scope of giving the output of each branch the meaning of a risk score; the final output is dependent on the weighted average of these risk scores. Thus, the clustering we made consists of gathering all the Time-series instances and their activated feature maps according to the quartile ranges of the corresponding output score of the Multi-Branch 1-D CNN model. We denoted as *low-risk class* all instances that return an output value lower than the 25th quantile of the empirical distribution of the output scores. *Mild-risk class* includes all instances associated with an output score falling between the 25th quantile and the 50th quantile of the empirical distribution of the output scores. Likewise, *middle-risk class* encloses all instances

associated with an output value between the 50th and 75th quantile of the empirical distribution of the output scores. All instances in the *high-risk class* have an output score with values larger than the 75th quantile of the empirical distribution of the output scores. For example, when forecasting an increase in the level of CRP of ICU patients, the *low-risk class* represents all potentially dangerous clinical conditions that are misclassified. In contrast, the *high-risk class* represents all clinical conditions of interest rightly classified by the Multi-Branch 1-D CNN. Note that the empirical distribution of output scores is evaluated over the test set only. In addition, we specify that, when using the DNN and analyzing the pattern reconstructed, we focus on the *low-risk class* and *high-risk class* instances.

To quantify the quality of the DNN reconstructions, we can refer to figure 4.29, where the histograms of the Pearson correlation coefficients and the Explained Variance Score between the true Time-Series instances and their DNN reconstruction are shown. In particular, we plotted the mean Pearson coefficient per Time-Series feature, i.e., we assessed the goodness of the reconstruction of one particular Time-Series feature. Likewise, the mean Explained Variance Score (we introduce this metric in section 1.1.5) is used to assess the quality of DNN reconstruction on each single Time-Series feature. Note that since we standardized the Time-Series instances, the Explained Variance Score is equivalent to the portion of the energy (i.e., the empirical second moment in a Time-Series) of the true Time-Series instance captured by the DNN reconstruction. As we can see, all reconstructed Time-Series features have at least a 0.70 correlation score (Pearson Coefficient) compared to the original ones; the Explained Variance Score can reveal similar results. In figure 4.29, the histograms of the Pearson correlation coefficients and the Explained Variance Score between the true Time-Series instances and their DNN reconstruction are shown. In particular, we plotted the mean Pearson coefficient per Time-Series feature, i.e., we asses the goodness of the reconstruction of one particular Time-Series feature. Likewise, the mean Explained Variance Score (we introduce this metric in section 1.1.5) is used to assess the quality of DNN reconstruction on each single Time-Series feature. Note that since we standardized the Time-Series instances, the Explained Variance Score is equivalent to the portion of the energy of the true Time-Series instance (i.e., the empirical second moment in a Time-Series) captured by the DNN reconstruction. As we can see, all reconstructed Time-Series features have at least a 0.70 correlation score (Pearson Coefficient) compared to the original ones; the Explained Variance Score can reveal similar results. The evaluation of the Explained Variance score reveals that all Time-Series features can be reconstructed with about 50% of their original energy. The difficulty of reconstructing high-frequency details makes the reconstruction of the breath rate Time-Series feature poorer; in this case, the captured energy is about 0.4. On the contrary, we can see that the Temperature is that Time-Series feature whose reconstruction has both the highest correlation and explained variance ratio. This result is also supported by the fact that the Temperature Time-Series slowly varies with variations of order 1 hour; in this case, the DNN does not need to reconstruct some high-scale details.

The relevant macroscopic characteristics of the DNN reconstructions are summarized in table 4.9. In evaluating the significance level of the U-test, we operated, as described above, on four different sub-domains of the time-domain of the Time-Series instances. That is, since the Time-Series instance represents the development of the ICU monitor signals within 24 hours, we opted to restrict our search for the discriminative macroscopic features on sub-domains of 6 hours. Unlike the Time-Series instances selected in section 4.3.1, we extracted, in this case only, the Time-Series instances by imposing that the time-domain starts at the midnight of one ICU stay day and ends at the midnight of the day after. In this way, when we analyzed the patterns of 6-hour chunks, we could

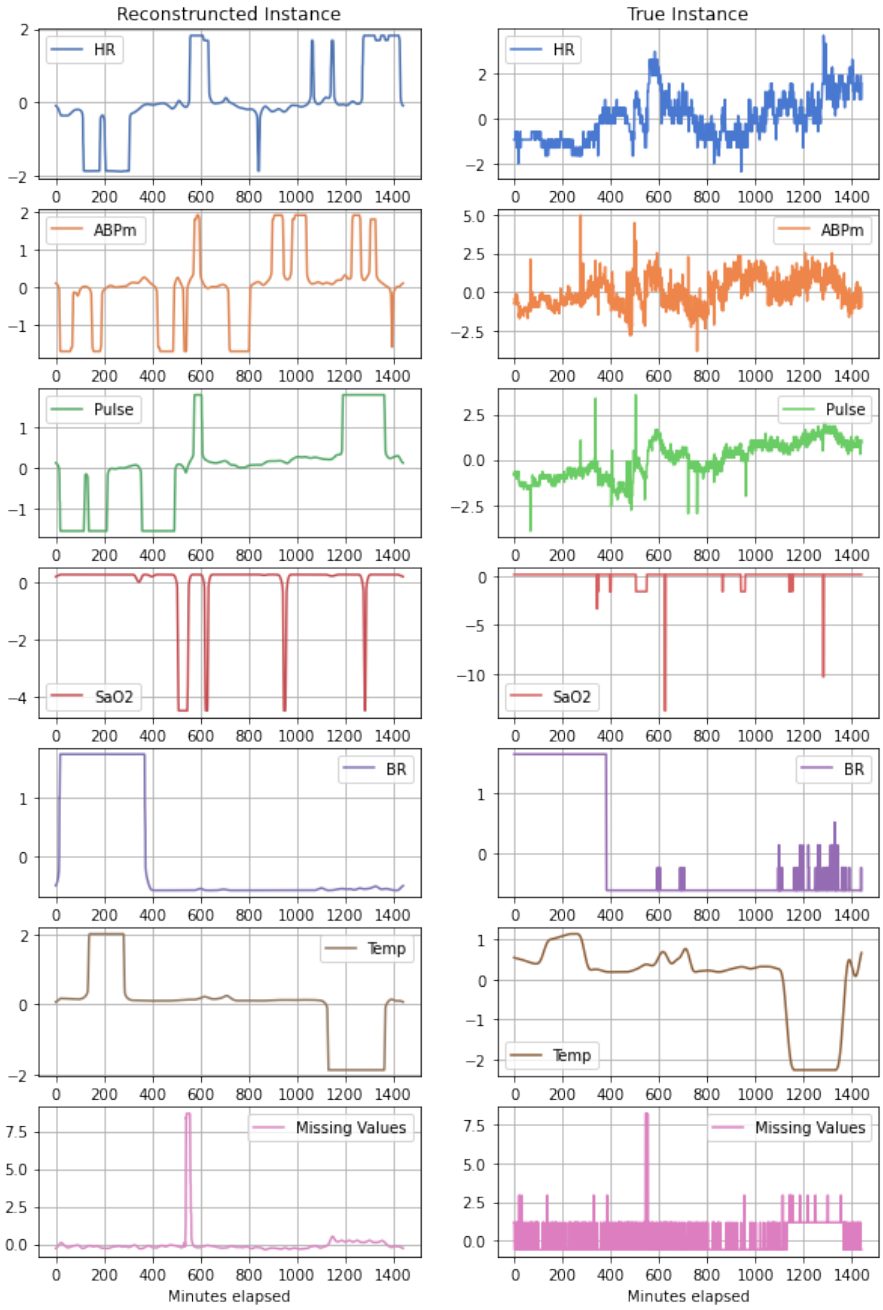


Figure 4.28: Example of a Time-Series instance reconstructed via DNN. Each row represents one single Time-Series feature (Heart Rate in blue, Arterial Blood Pressure in orange, Pulse in green, SaO_2 in red, Breath Rate in purple, Temperature in brown, and a Missing Value indicator in pink). On the left side, the reconstruction proposed by the DNN, while on the right, the original (standardized) instance.

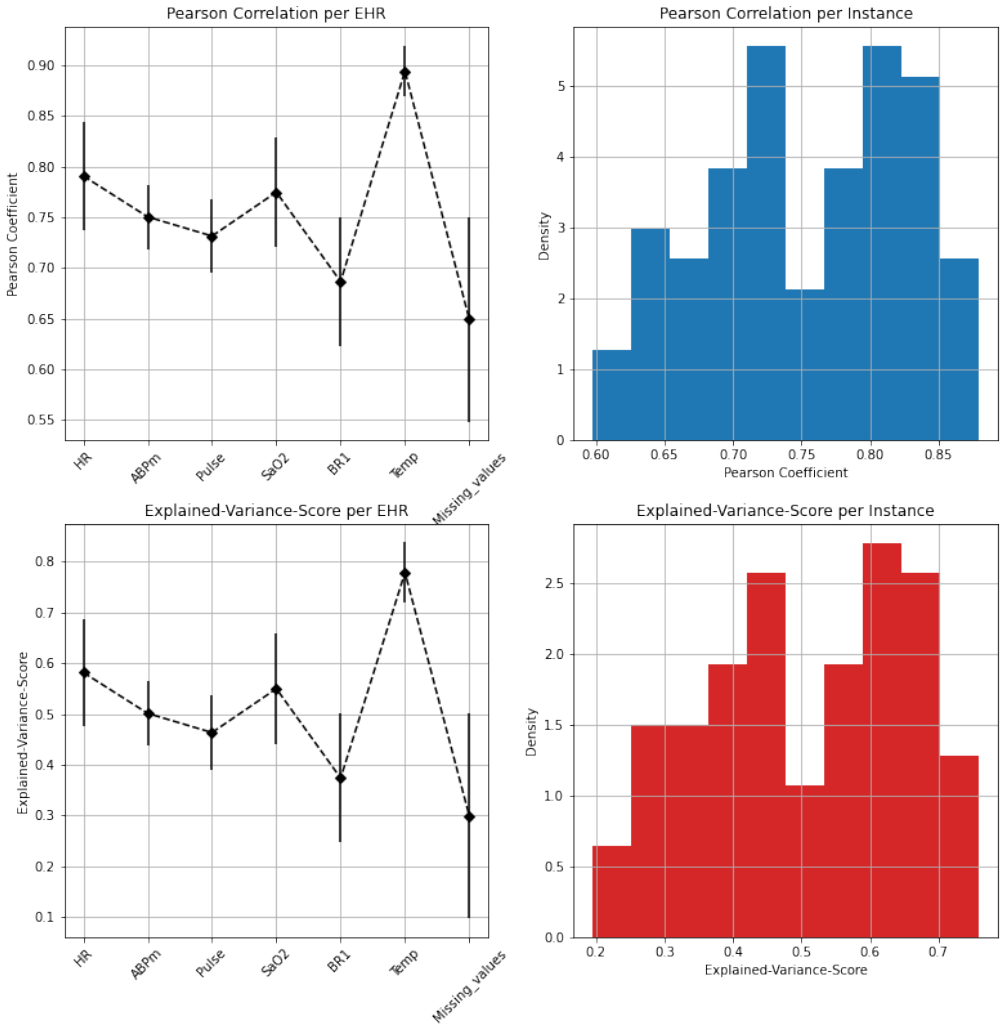


Figure 4.29: Goodness of the reconstruction proposed by the DNN. On the top left is the mean Pearson coefficient per Time-Series feature. The mean Explained Variance Score per Time-Series feature is on the bottom left. Error bars denote the standard deviation. On the top left, the histograms of Pearson coefficient per Time-Series instance. On the bottom left, the histograms of Explained Variance Score per Time-Series instance.

relate the macroscopic extracted features to a precise moment of the ICU routine; in particular, this choice should help highlight some differences in the behavior of the EHR of ICU patients depending of the moment of a daytime.

For Hearth Rate, we observed that the mean value could be regarded as a discriminative feature when restricted at hours 6-12 of the instance domain (time window). Indeed, high-risk subjects present a lower mean value (87.5 bpm vs. 95.3 bpm) with also a lower standard deviation (9.9 vs. 16.3 bpm); likewise, the maximum value is discriminative at hours 6-12 (120 bpm vs. 135 bpm).

Extracted Feature	**High-Risk Class**	**Low-Risk Class**	**U-test pval**
HR Mean [beats per min.] (0-6 H)	90.25085	89.03179	0.880
HR Mean [beats per min.] (6-12 H)	87.49373	95.30343	0.012
HR Mean [beats per min.] (12-18 H)	93.19080	91.92562	0.302
HR Mean [beats per min.] (18-24 H)	97.46462	92.13916	0.580
HR Std [beats per min.] (0-6 H)	8.97833	10.69220	0.351
HR Std [beats per min.] (6-12 H)	9.85511	16.34760	0.004
HR Std [beats per min.] (12-18 H)	12.64849	12.81104	0.960
HR Std [beats per min.] (18-24 H)	9.47137	14.06765	0.070
HR Mean change [beats per min.] (0-6 H)	0.02583	0.05565	0.013
HR Mean change [beats per min.] (6-12 H)	0.01677	-0.00008	0.687
HR Mean change [beats per min.] (12-18 H)	-0.00291	-0.02336	0.279
HR Mean change [beats per min.] (18-24 H)	-0.04009	-0.03238	0.389
HR Min [beats per min.] (0-6 H)	77.35360	76.81347	0.546
HR Min [beats per min.] (6-12 H)	78.49758	77.33552	0.365
HR Min [beats per min.] (12-18 H)	79.24613	77.20709	0.159
HR Min [beats per min.] (18-24 H)	77.49865	76.72498	0.352
HR Max [beats per min.] (0-6 H)	109.04804	118.18172	0.365
HR Max [beats per min.] (6-12 H)	119.51550	135.29394	0.006
HR Max [beats per min.] (12-18 H)	119.83628	126.88359	0.314
HR Max [beats per min.] (18-24 H)	116.90331	129.88280	0.258
ABPm Mean [mmHg] (0-6 H)	80.49195	80.48983	0.980
ABPm Mean [mmHg] (6-12 H)	79.30919	84.60495	0.083
ABPm Mean [mmHg] (12-18 H)	85.76048	84.64698	0.580
ABPm Mean [mmHg] (18-24 H)	86.83838	82.65823	0.191
ABPm Std [mmHg] (0-6 H)	11.82015	11.53544	0.840
ABPm Std [mmHg] (6-12 H)	13.59302	12.32282	0.392
ABPm Std [mmHg] (12-18 H)	11.12005	12.16563	0.505
ABPm Std [mmHg] (18-24 H)	8.76737	11.77275	0.237
ABPm Mean change [mmHg] (0-6 H)	-0.04968	-0.02743	0.101
ABPm Mean change [mmHg] (6-12 H)	0.00418	0.00667	0.900
ABPm Mean change [mmHg] (12-18 H)	0.00883	-0.00984	0.377
ABPm Mean change [mmHg] (18-24 H)	0.03833	0.03166	0.686
ABPm Min [mmHg] (0-6 H)	55.21539	57.21134	0.782
ABPm Min [mmHg](6-12 H)	52.68058	54.88221	0.687
ABPm Min [mmHg] (12-18 H)	60.11351	57.56059	0.513
ABPm Min [mmHg] (18-24 H)	69.42735	56.76904	0.046
ABPm Max [mmHg] (0-6 H)	97.54514	97.76943	0.706
ABPm Max [mmHg] (6-12 H)	97.60905	97.82886	0.744
ABPm Max [mmHg] (12-18 H)	97.58216	97.72379	0.763
ABPm Max [mmHg] (18-24 H)	97.57394	97.78144	0.725
Pulse Mean [mmHg] (0-6 H)	49.87360	65.28850	0.005

Pulse Mean [mmHg] (6-12 H)	57.97655	61.78814	0.513
Pulse Mean [mmHg] (12-18 H)	71.41099	66.14890	0.218
Pulse Mean [mmHg] (18-24 H)	78.33887	64.37446	0.001
Pulse Std [mmHg] (0-6 H)	18.09502	16.41711	0.597
Pulse Std [mmHg] (6-12 H)	18.11551	17.74296	1.000
Pulse Std [mmHg] (12-18 H)	12.81818	16.57801	0.078
Pulse Std [mmHg] (18-24 H)	6.20644	15.60477	0.001
Pulse Mean change [mmHg] (0-6 H)	-0.10163	-0.05140	0.097
Pulse Mean change [mmHg] (6-12 H)	0.03908	-0.01113	0.128
Pulse Mean change [mmHg] (12-18 H)	0.03469	0.01475	0.545
Pulse Mean change [mmHg] (18-24 H)	0.03084	0.04908	0.691
Pulse Min [mmHg] (0-6 H)	15.69380	28.33167	0.279
Pulse Min [mmHg] (6-12 H)	17.73147	22.68110	0.580
Pulse Min [mmHg] (12-18 H)	41.55295	25.17268	0.022
Pulse Min [mmHg] (18-24 H)	60.56395	28.96096	0.001
Pulse Max [mmHg] (0-6 H)	83.40401	85.87289	0.039
Pulse Max [mmHg] (6-12 H)	81.10065	84.93752	0.021
Pulse Max [mmHg] (12-18 H)	83.81834	85.55192	0.070
Pulse Max [mmHg] (18-24 H)	83.81834	85.92848	0.074
SaO_2[%] Mean (0-6 H)	95.68374	95.75371	0.980
SaO_2[%] Mean (6-12 H)	95.37707	95.22388	0.920
SaO_2[%] Mean (12-18 H)	96.43572	95.81463	0.392
SaO_2[%] Mean (18-24 H)	95.49098	96.19160	0.546
SaO_2[%] Std (0-6 H)	2.59366	2.37663	0.880
SaO_2[%] Std (6-12 H)	1.54377	3.01945	0.001
SaO_2[%] Std (12-18 H)	2.48077	2.58407	0.960
SaO_2[%] Std (18-24 H)	2.46606	2.83974	0.218
SaO_2[%] Mean-change (0-6 H)	0.01390	0.01460	0.421
SaO_2[%] Mean-change (6-12 H)	0.00043	-0.00465	0.295
SaO_2[%] Mean-change (12-18 H)	0.00197	0.00388	0.899
SaO_2[%] Mean-change (18-24 H)	-0.01640	-0.01387	0.763
SaO_2[%] Min (0-6 H)	90.18080	91.15595	0.801
SaO_2[%] Min (6-12 H)	92.93062	91.41240	0.024
SaO_2[%] Min (12-18 H)	92.38295	92.22514	0.379
SaO_2[%] Min (18-24 H)	90.15090	91.14041	0.940
SaO_2[%] Max (0-6 H)	98.91408	98.75195	0.506
SaO_2[%] Max (6-12 H)	98.01306	99.43753	0.001
SaO_2[%] Max (12-18 H)	99.12854	99.57908	0.162
SaO_2[%] Max (18-24 H)	98.62984	99.49638	0.093
BR Mean [breaths per minute] (0-6 H)	17.09256	20.26258	0.191
BR Mean [breaths per minute] (6-12 H)	20.34110	21.75134	0.365
BR Mean [breaths per minute] (12-18 H)	23.41751	20.73557	0.092
BR Mean [breaths per minute] (18-24 H)	23.14883	21.25052	0.314
BR Std [breaths per minute] (0-6 H)	2.45312	4.15823	0.059
BR Std [breaths per minute] (6-12 H)	5.64672	6.30033	0.811
BR Std [breaths per minute] (12-18 H)	6.31248	4.59354	0.092
BR Std [breaths per minute] (18-24 H)	4.82449	5.97242	0.352
BR Mean-change [breaths per minute] (0-6 H)	0.00753	0.02071	0.004
BR Mean-change [breaths per minute](6-12 H)	0.01644	0.00382	0.217
BR Mean-change [breaths per minute] (12-18 H)	-0.00173	-0.01199	0.174

BR Mean-change [breaths per minute](18-24 H)	-0.02185	-0.01288	0.462
BR Min [breaths per minute] (0-6 H)	15.20466	13.63193	0.006
BR Min [breaths per minute] (6-12 H)	15.32299	14.38983	0.063
BR Min [breaths per minute] (12-18 H)	17.38230	14.09325	0.000
BR Min [breaths per minute] (18-24 H)	15.26297	13.58740	0.005
BR Max [breaths per minute] (0-6 H)	23.74639	31.38915	0.047
BR Max [breaths per minute] (6-12 H)	35.08708	35.20759	0.880
BR Max [breaths per minute] (12-18 H)	39.34894	31.98470	0.050
BR Max [breaths per minute] (18-24 H)	35.53892	35.39954	0.821
Temp. Mean [$^{\circ}C$] (0-6 H)	36.16267	36.42349	0.352
Temp. Mean [$^{\circ}C$] (6-12 H)	36.29840	36.72074	0.125
Temp. Mean [$^{\circ}C$] (12-18 H)	36.16128	36.29120	0.314
Temp. Mean [$^{\circ}C$] (18-24 H)	36.70718	36.16458	0.166
Temp. Std [$^{\circ}C$] (0-6 H)	0.69357	1.01472	0.097
Temp. Std [$^{\circ}C$] (6-12 H)	0.69731	0.91340	0.242
Temp. Std [$^{\circ}C$] (12-18 H)	0.94789	0.80114	0.443
Temp. Std [$^{\circ}C$] (18-24 H)	0.81764	1.02345	0.258
Temp. Mean-change [$^{\circ}C$] (0-6 H)	0.00331	0.00379	1.000
Temp. Mean-change [$^{\circ}C$] (6-12 H)	-0.00069	0.00052	0.339
Temp. Mean-change [$^{\circ}C$] (12-18 H)	0.00068	-0.00178	0.152
Temp. Mean-change [$^{\circ}C$] (18-24 H)	-0.00323	-0.00272	0.406
Temp. Min [$^{\circ}C$] (0-6 H)	35.14877	35.01549	0.258
Temp. Min [$^{\circ}C$] (6-12 H)	35.40194	35.32136	0.258
Temp. Min [$^{\circ}C$] (12-18 H)	35.14777	35.25335	0.880
Temp. Min [$^{\circ}C$] (18-24 H)	35.18189	34.98828	0.159
Temp. Max [$^{\circ}C$] (0-6 H)	37.39307	38.77970	0.083
Temp. Max [$^{\circ}C$] (6-12 H)	37.59037	37.99943	0.529
Temp. Max [$^{\circ}C$] (12-18 H)	38.06050	37.73343	0.633
Temp. Max [$^{\circ}C$] (18-24 H)	38.12165	37.98701	0.365

Table 4.9: Mean values of the statistics evaluated per each Time-Series features for the two classes considered (High-Risk Class, Low-Risk Class). The column **U-test p-val** contains the p-value of the U-test between the two classes.

In this case, we can see that the Multi-branch model recognizes better patients that are not in tachycardia (HR $\geq$ 90 bpm, see section 4.6), with respect to those that show one of the main symptoms associated with SIRS Comstedt et al. (2009). Although Arterial Blood Pressure cannot offer a net distinction between the two classes, we can see that Pulse can reveal a risk when measured at hours 18-24 of the instance domain. In that interval, the mean value is higher for the high-risk subjects (78.3 mmHg vs. 64.4 mmHg) as well as both the standard deviation (6.2 mmHg vs. 15.6 mmHg) and the Minimum value (60.5 vs. 29.0 mmHg). Therefore, we can see that a lower heart activity combined with a high pumping activity (the Pulse measures the difference between Systolic and Diastolic Blood Pressure) can evidence the chance of being subjected to a high increase in CRP level. Note that this implies an increase in the relative pulse (i.e., the ratio between Pulse and Diastolic Blood Pressure) as well. We recall that the relative pulse is expressed as

$$\frac{p}{m - \frac{p}{3}},$$

with p and m denoting the *Pulse* and the *Mean Arterial Blood pressure*, respectively. Provided

the fact that the physical limit $p/m << 3$, we can see that pulse and relative pulse are related as described above. When considering the respiratory aspects of ICU patients (SaO2), we can see that at hours 6-12 of the domain of the instances the intensity of deviations and extremal values are predictive, e.g. standard deviation (1.5 vs. 3.0), Minimum value (93.0 vs. 91.4), and Maximum value (98.0 vs. 99.4). For the Breath Rate, we can see that the Minimum value appears to play a fundamental role (note that the p-value is always lower than 0.05 for all four chunks along the 24 hours). Nonetheless, Minimum values are here slightly higher by a few breaths per minute in the high-risk class; this points out a marginal contribution from this Time-Series feature. At last, we can see from Temperature that none of the extracted features can contribute to distinguishing patients among the two classes.

Appendix A

Backpropagation algorithm for CNN

A.1 BP algorithm for 1-D CNN

In section 1.2.2, we have already shown how to backpropagate through the dense layers of a MLP. The rules derived, however, cannot be used under the context of the convolutional and max-pooling layers. We derive here the BP algorithm for a 1-D CNN (i.e., the network is fed with 1-D data; see section 1.1.3). Therefore, we consider an 1-D CNN consisting of the sequential repetition of the following hidden layers:

1. Convolutional layer: for the j-th Time-Series feature at the spatial location ξ_k located on the n-th hidden layer, we have

$$a_j^n[\xi_k] = \sum_{i=1}^{I} (w_{ji}^n * \zeta_i^{n-1})[\xi_k], \tag{A.1}$$

   that is the sum of the I convolutions (with I the number of Time-Series features) between the filters (convolutional mask) w_{ji}^n and the elements of the (max-pooled) Time-series features ζ_i^{n-1} ($(n-1)$-th layer, i-th feature map). As in (1.5), the convolutional operator is discrete.

2. Activation layer: the activation function $\phi(\cdot)$ is applied on all the elements computed by the convolutional layer. Thus, we have

$$\eta_j^n[\xi_k] = \phi(a_j^n[\xi_k]); \tag{A.2}$$

   with $\eta_j^n[\xi_k]$, representing the elements of the so-called *activated feature map*.

3. Maxpooling layer: the dimensionality of each activated feature map after pooling the maximal values. When considering a 1-D CNN, the maxpooling operations are the following:

$$\zeta_j^n[\xi_k] = \sum_{h=0}^{h^*} \Delta_j^n(k,h)\eta_j^n[\xi_h]; \tag{A.3}$$

   with $\Delta_j^n(k,h) = 1$, if $\eta_j^n[\xi_h]$ is precisely the value of the max-pooled activated feature map at the spatial position ξ_k; 0 otherwise. With h^*, we denote the number of spatial locations (i.e., elements) of the single activated feature map η_j^n.

Thus, we search for the gradient of the loss function $\Lambda(\mathbf{w})$ (see section 1.2) with respect to the elements of the convolutional masks. We make use of the chain rule for derivatives, and so we obtain

$$\frac{\partial \Lambda}{\partial w_{ji}^n[\xi_k]} = \sum_{l=0}^{l^*} \frac{\partial \Lambda}{\partial \zeta_j^n[\xi_l]} \frac{\partial \zeta_j^n[\xi_l]}{\partial \eta_j^n[\xi_l]} \frac{\partial \eta_j^n[\xi_l]}{\partial a_j^n[\xi_l]} \frac{\partial a_j^n[\xi_l]}{\partial w_{ji}^n[\xi_k]};$$

with l^* denotes the number of spatial locations of $\zeta_j^n[\cdot]$. Note that w_{ji}^n is dependent with all the variables a_j^n, as well as with all η_j^n and ζ_j^n, because all these quantities involve the weight w_{ji}^n in the convolution (A.1).

Each partial derivative, therefore, can be associated with a specific quantity of the forward rules (A.1)-(A.3). Thus, we have

- A direct application of (A.1) leads to

$$\frac{\partial a_j^n[\xi_l]}{\partial w_{ji}^n} = \eta_i^{n-1}[\xi_k + M].$$

- A direct application of (A.2) leads to

$$\frac{\partial \eta_j^n[\xi_l]}{\partial a_j^n[\xi_l]} = \phi'(a_j^n[\xi_l]).$$

- A direct application of (A.3) leads to

$$\frac{\partial \zeta_j^n[\xi_l]}{\partial \eta_j^n[\xi_l]} = \Delta_j^n(l, l).$$

- Finally, we can consider all connections between ζ_j^n and all nodes ζ_j^{n+1} nodes, so we have

$$\frac{\partial \Lambda}{\partial \zeta_j^n[\xi_l]} = \sum_{m=0}^{m^*} \frac{\partial \Lambda}{\partial \eta_j^{n+1}[\xi_m]} \frac{\partial \eta_j^{n+1}[\xi_m]}{\partial a_j^{n+1}[\xi_m]} \frac{\partial a_j^{n+1}[\xi_m]}{\partial \zeta_j^n[\xi_l]};$$

with m^* all the spatial locations of η_{n+1}. We can highlight the *error terms* (see section 1.2.2), i.e.,

$$\delta_j^{n+1}[\xi_m] = \frac{\partial \Lambda}{\partial \eta_j^{n+1}[\xi_m]} \frac{\partial \eta_j^{n+1}[\xi_m]}{\partial a_j^{n+1}[\xi_m]} = \frac{\partial \Lambda}{\partial a_j^{n+1}[\xi_m]};$$

also, by means of (A.1), we obtain

$$\frac{\partial a_j^{n+1}[\xi_m]}{\partial \zeta_j^n[\xi_l]} = w_{ji}^n[\xi_l - M].$$

Reordering all the results obtained above, the form of the desired gradients is given by

$$\frac{\partial \Lambda}{\partial w_{ji}^n[\xi_k]} = \eta_i^{n-1}[\xi_k + M] \sum_{l=0}^{l^*} \sum_{m=0}^{m^*} \phi'(a_j^n[\xi_l]) \Delta_j^n(l, l) \delta_j^{n+1}[\xi_m] w_{ji}^n[\xi_l - M]. \tag{A.4}$$

As shown in (A.4), the backpropagation in 1-D CNN is not so different from what was discussed for the MLP case. In the forward phase, we propagate the input data through the hidden layers and evaluate all terms given by the forward rules (A.1)-(A.3). We evaluate the error term δ starting from the output node. Finally, we use (A.4) to backpropagate the hidden layers of the 1-D CNN. This scheme is used recursively to evaluate the desired derivative for any hidden layer.

Appendix B

Backpropagation algorithm for LSTM

In this appendix, we present the BP algorithm for one particular class of RNN, i.e., LSTM. To better introduce the reader to this topic, we shall first review the problem of the exploding gradient in the Vanilla RNN, and lately, we will present the BP algorithm for the LSTM.

B.1 Exploding gradient in RNN

Let's consider the Vanilla RNN as introduced in 1.1.4. The equation governing the updating of the hidden states at any time t and the evaluation of the output is the following:

$$\begin{cases} y_t^{(j)} = \theta(h_t^{(j)}) \\ h_t^{(j)} = \phi\left(\sum_{j=0}^{N} U_{ij} x_t^{(j)} + \sum_{j=0}^{N} V_{ij} h_{t-1}^{(j)}\right); \end{cases} \tag{B.1}$$

with $y_t^{(j)}$ the j-th output of the Vanilla RNN at time t, $h_t^{(j)}$ the j-th component of the hidden recurrent state at time t, $x_t^{(j)}$ the j-th component of the input data at time t, U and V a couple of fixed weights, and $\theta(\cdot)$ and $\phi(\cdot)$ a couple of activation functions. As introduced in 1.2, the learning phase of ANN usually exploits iterative algorithms to minimize a loss function. This minimization involves the iterative computation of the gradient with respect to those weights one needs to optimize, i.e., one wants to find that set of weights that minimizes the loss function. For example, in Time-Series forecasting, one aims to predict the outcome of a class of Time-Series at some time t given the history of those Time-Series at the previous times. In this case, the *mean squared error* function represents a suited choice as a loss function; thus, we write

$$\Lambda(y, \bar{y}) = \frac{1}{N} \sum_{k=0}^{N} (y_\tau^{(k)} - \bar{y}_\tau^{(k)})^2, \tag{B.2}$$

with $\bar{y}_\tau^{(k)}$ the true value of the k-th Time-Series at time τ, and $y_\tau^{(k)}$ the output predicted by the RNN in question for the k-th Time-Series.

When training a RNN, we need to compute the gradients $\frac{\partial \Lambda}{\partial U_{ij}}$ and $\frac{\partial \Lambda}{\partial V_{ij}}$. In this case, the BP algorithm remains a valid approach; accordingly, one computes the desired gradient by considering the nested relation intercurring from the output up to the input node. The couple of derivatives for

the output layer is the following:

$$\begin{cases} \frac{\partial \Lambda}{\partial U_{ij}} = \frac{\partial \Lambda}{\partial y_t} \frac{\partial y_t}{\partial h_t^{(i)}} \frac{\partial h_t^{(i)}}{\partial U_{ij}} \\ \frac{\partial \Lambda}{\partial V_{ij}} = \frac{\partial \Lambda}{\partial y_t} \frac{\partial y_t}{\partial h_t^{(i)}} \frac{\partial h_t^{(i)}}{\partial V_{ij}}; \end{cases}$$

we used both (B.1) and the chain rule to consider the nested relation between the output and the hidden state at time t. Once again, we can make use of the chain rule combined with (B.1), and so we have

$$\begin{cases} \frac{\partial h_t^{(i)}}{\partial U_{ij}} = \sum_{k=0}^{K} \frac{\partial h_t^{(k)}}{\partial h_t^{(k-1)}} \frac{\partial h_t^{(k-1)}}{\partial U_{ij}} \\ \frac{\partial h_t^{(i)}}{\partial V_{ij}} = \sum_{k=0}^{K} \frac{\partial h_t^{(k)}}{\partial h_t^{(k-1)}} \frac{\partial h_t^{(k-1)}}{\partial V_{ij}}; \end{cases}$$

with K the number of components of the hidden state at time $t-1$. Applying (B.1) we can see that

$$\frac{\partial h_\tau^{(i)}}{\partial h_{\tau-1}^{(i)}} = \phi' \left(\sum_{j=0}^{N} U_{ij} x_\tau^{(j)} + \sum_{j=0}^{N} V_{ij} h_{\tau-1}^{(j)} \right) V_{ij}.$$

Assembling all together, we obtain

$$\begin{cases} \frac{\partial \Lambda}{\partial U_{ij}} = \delta_t \theta'(h_t^{(i)}) x_0^{(j)} \prod_{\tau=1}^{t} \left[\phi' \left(\sum_{k=0}^{N} U_{ik} x_\tau^k + V_{ik} h_{\tau-1}^{(k)} \right) \sum_{k=0}^{K} V_{ik} \right] \\ \frac{\partial \Lambda}{\partial V_{ij}} = \delta_t \theta'(h_t^{(i)}) h_0^{(j)} V_{ij}^t \prod_{\tau=1}^{t} \left[\phi' \left(\sum_{k=0}^{N} U_{ik} x_\tau^k + V_{ik} h_{\tau-1}^{(k)} \right) \sum_{k=0}^{K} V_{ik} \right]; \end{cases} \tag{B.3}$$

with δ_t the *error terms* (see section 1.2.2) at time t, (i.e., $\delta_t = \frac{\partial \Lambda}{\partial y_t}$); we recall that in case the mean square error function is used as a loss function, these kinds of gradients are proportional with the difference between the true observations and the output estimated by the model. As shown in (B.3), both the gradients are actually dependent on terms like $\left(\sum_{k=0}^{K} V_{ij} \right)^t$, i.e., the t-th power of $\sum_{k=0}^{K} V_{ik}$. As soon as these weights attain large (or small) values, the computation of the gradient is affected by an explosive (vanishing) term that might make harder and unpractical the minimization of the loss function. As a result, the prediction made by the Vanilla RNN can often be inefficient or unreliable. The following section is dedicated to the BP algorithm for LSTM layer; we shall show there how a refinement of the RNN structure can help contrast the issue of the exploding gradient.

B.2 BP algorithm for one LSTM layer

Let's consider one ANN composed of a single LSTM layer. For convenience, as in (B.2), we denote with $\Lambda(y, \bar{y})$ the loss function of the RNN in question. The forward equations of the following:

$$\begin{cases} I_t^{(j)} = \sigma \left(\sum_{j=0}^{N} V_{Iij} x_t^{(j)} + U_{Iij} h_{t-1}^{(j)} \right) \\ F_t^{(j)} = \sigma \left(\sum_{j=0}^{N} V_{Fij} x_t^{(j)} + U_{Fij} h_{t-1}^{(j)} \right) \\ O_t^{(j)} = \sigma \left(\sum_{j=0}^{N} V_{Oij} x_t^{(j)} + U_{Oij} h_{t-1}^{(j)} \right) \\ S_t^{(j)} = \tau \left(\sum_{j=0}^{N} V_{Sij} x_t^{(j)} + U_{Sij} h_{t-1}^{(j)} \right) \\ c_t^{(j)} = F_t^{(j)} c_{t-1}^{(j)} + I_t^{(j)} S_t^{(j)} \\ h_t^{(j)} = O_t^{(j)} \sigma(c_t^{(j)}) \\ y_t^{(j)} = \theta(h_t^{(j)}); \end{cases} \tag{B.4}$$

with $x_t^{(j)}$ the j-th component of the observations at time t and $y_t^{(j)}$ the j-th component of the output at time t; $I_t^{(j)}$ is the j-th component of the *input gate* at time t, V_{Iij} and U_{Iij} the matrix weights of the input gate; $S_t^{(j)}$ is the j-th component of the *new cell gate* at time t, V_{Sij} and U_{Sij} the matrix weights of the new cell gate, $F_t^{(j)}$ is the j-th component of the *forget gate* at time t, V_{Fij} and U_{Fij} the matrix weights of the forget gate; $O_t^{(j)}$ is the j-th component of the *output gate* at time t, V_{Oij} and U_{Oij} the matrix weights of the output gate; h_t^j and c_t^j are the j-th components at time t of the hidden state and the cell state, respectively; $\sigma(\cdot)$ and $\tau(\cdot)$ denote the sigmoid and the hyperbolic tangent function. Finally, $\theta(\cdot)$ is a generic activation function.

Let's start with the *input gate*. To compute the desired gradients, we can make use of the chain rule for composite functions, i.e., the calculation of the gradient with respect to both weights V_{Iij} and U_{Iij} is expressed as follows:

$$\begin{cases} \frac{\partial \Lambda}{\partial U_{Iij}} = \frac{\partial \Lambda}{\partial y_t^{(i)}} \frac{\partial y_t^{(i)}}{\partial h_t^{(i)}} \frac{\partial h_t^{(i)}}{\partial c_t^{(i)}} \frac{\partial c_t^i}{\partial U_{Iij}} \\ \frac{\partial \Lambda}{\partial V_{Iij}} = \frac{\partial \Lambda}{\partial y_t^{(i)}} \frac{\partial y_t^{(i)}}{\partial h_t^{(i)}} \frac{\partial h_t^{(i)}}{\partial c_t^{(i)}} \frac{\partial c_t^i}{\partial V_{Iij}}. \end{cases} \tag{B.5}$$

As in the previous section, the terms $\frac{\partial \Lambda}{\partial y_t^{(i)}}$ represent the *error terms*, and we shall denote them with $\delta_t^{(i)}$. So we have,

$$\begin{cases} \frac{\partial \Lambda}{\partial U_{Iij}} = \delta_t^{(i)} \theta'(h_t^{(i)}) O_t^{(i)} \frac{\partial c_t^i}{\partial U_{Iij}} \\ \frac{\partial \Lambda}{\partial V_{Iij}} = \delta_t^{(i)} \theta'(h_t^{(i)}) O_t^{(i)} \frac{\partial c_t^i}{\partial V_{Iij}}. \end{cases} \tag{B.6}$$

To evaluate the term $\frac{\partial c_t^i}{\partial U_{Iij}}$ we can expolit the nested relation between c_t^i with respect to both c_{t-1}^i and $I_t^{(j)}$. Also, we apply the same approach for evaluating $\frac{\partial c_t^i}{\partial V_{Iij}}$. So we have,

$$\begin{cases} \frac{\partial c_t^i}{\partial U_{Iij}} = \frac{\partial c_t^{(i)}}{\partial c_t^{(i-1)}} \frac{\partial c_t^{(i)}}{\partial U_{Iij}} + \frac{\partial c_t^{(i)}}{\partial I_t^{(i)}} \sum_{k=0}^{K} \frac{\partial I_t^{(i)}}{\partial h_{t-1}^{(k)}} \frac{\partial h_{t-1}^{(k)}}{\partial U_{Iij}} \\ \frac{\partial c_t^i}{\partial V_{Iij}} = \frac{\partial c_t^{(i)}}{\partial c_t^{(i-1)}} \frac{\partial c_t^{(i)}}{\partial V_{Iij}} + \frac{\partial c_t^{(i)}}{\partial I_t^{(i)}} \sum_{k=0}^{K} \frac{\partial I_t^{(i)}}{\partial h_{t-1}^{(k)}} \frac{\partial h_{t-1}^{(k)}}{\partial V_{Iij}}; \end{cases}$$

and this is equivalent to

$$\begin{cases} \frac{\partial c_t^i}{\partial U_{Iij}} = F_t^{(i)} \frac{\partial c_t^{(i)}}{\partial U_{Iij}} + s_t^{(i)} \sum_{k=0}^{K} \left[\sigma' \left(\sum_{j=0}^{N} V_{Iij} x_t^{(j)} + U_{Iij} h_{t-1}^{(j)} \right) U_{Iik} \frac{\partial h_{t-1}^{(k)}}{\partial U_{Iij}} \right] \\ \frac{\partial c_t^i}{\partial U_{Iij}} = F_t^{(i)} \frac{\partial c_t^{(i)}}{\partial V_{Iij}} + s_t^{(i)} \sum_{k=0}^{K} \left[\sigma' \left(\sum_{j=0}^{N} V_{Iij} x_t^{(j)} + U_{Iij} h_{t-1}^{(j)} \right) U_{Iik} \frac{\partial h_{t-1}^{(k)}}{\partial V_{Iij}} \right]. \end{cases} \tag{B.7}$$

When considering the *new state cell gate*, we repeat the same approach used for deriving the equations (B.6) and (B.7). So, we have

$$\begin{cases} \frac{\partial \Lambda}{\partial U_{Sij}} = \delta_t^{(i)} \theta'(h_t^{(i)}) O_t^{(i)} \frac{\partial c_t^i}{\partial U_{Sij}} \\ \frac{\partial \Lambda}{\partial V_{Sij}} = \frac{\partial \Lambda}{\partial U_{Sij}} = \delta_t^{(i)} \theta'(h_t^{(i)}) O_t^{(i)} \frac{\partial c_t^i}{\partial V_{Sij}}; \end{cases} \tag{B.8}$$

$$\begin{cases} \frac{\partial c_t^{(i)}}{\partial U_{Sij}} = F_t^{(i)} \frac{\partial c_t^{(i)}}{\partial U_{Sij}} + I_t^{(i)} \sum_{k=0}^{K} \left[\tau' \left(\sum_{j=0}^{N} V_{Sij} x_t^{(j)} + U_{Sij} h_{t-1}^{(j)} \right) U_{Sik} \frac{\partial h_{t-1}^{(k)}}{\partial U_{Sij}} \right] \\ \frac{\partial c_t^i}{\partial U_{Sij}} = F_t^{(i)} \frac{\partial c_t^{(i)}}{\partial V_{Sij}} + I_t^{(i)} \sum_{k=0}^{K} \left[\tau' \left(\sum_{j=0}^{N} V_{Sij} x_t^{(j)} + U_{Sij} h_{t-1}^{(j)} \right) U_{Sik} \frac{\partial h_{t-1}^{(k)}}{\partial V_{Sij}} \right]. \end{cases} \tag{B.9}$$

Likewise, for the forget gate one gets

$$\begin{cases} \frac{\partial \Lambda}{\partial U_{Fij}} = \delta_t^{(i)} \theta'(h_t^{(i)}) O_t^{(i)} \frac{\partial c_t^i}{\partial U_{Fij}} \\ \frac{\partial \Lambda}{\partial V_{Fij}} = \delta_t^{(i)} \theta'(h_t^{(i)}) O_t^{(i)} \frac{\partial c_t^i}{\partial V_{Fij}}; \end{cases} \tag{B.10}$$

$$\begin{cases} \frac{\partial c_t^i}{\partial U_{Fij}} = F^{(i)} \frac{\partial c_t^{(i)}}{\partial U_{Fij}} + c_{t-1}^{(i)} \sum_{k=0}^{K} \left[\sigma' \left(\sum_{j=0}^{N} V_{Fij} x_t^{(j)} + U_{Fij} h_{t-1}^{(j)} \right) U_{Fik} \frac{\partial h_{t-1}^{(k)}}{\partial U_{Fij}} \right] \\ \frac{\partial c_t^i}{\partial U_{Iij}} = F^{(i)} \frac{\partial c_t^{(i)}}{\partial V_{Fij}} + c_{t-1}^{(i)} \sum_{k=0}^{K} \left[\sigma' \left(\sum_{j=0}^{N} V_{Fij} x_t^{(j)} + U_{Iij} h_{t-1}^{(j)} \right) U_{Fik} \frac{\partial h_{t-1}^{(k)}}{\partial V_{Fij}} \right]. \end{cases} \tag{B.11}$$

Finally, for the output gate, one gets

$$\begin{cases} \frac{\partial \Lambda}{\partial U_{Oij}} = \delta_t^{(i)} \theta'(h_t^{(i)}) \sigma(c_t^{(i)}) \frac{\partial O_t^i}{\partial U_{Oij}} \\ \frac{\partial \Lambda}{\partial V_{Oij}} = \delta_t^{(i)} \theta'(h_t^{(i)}) \sigma(c_t^{(i)}) \frac{\partial O_t^i}{\partial V_{Oij}}; \end{cases} \tag{B.12}$$

$$\begin{cases} \frac{\partial O_t^{(i)}}{\partial U_{Oij}} = \sum_{k=0}^{K} \left[\sigma' \left(\sum_{j=0}^{N} V_{Oij} x_t^{(j)} + U_{Oij} h_{t-1}^{(j)} \right) U_{Oik} \frac{\partial h_{t-1}^{(k)}}{\partial U_{Oij}} \right] \\ \frac{\partial O_t^{(i)}}{\partial V_{Oij}} = \sum_{k=0}^{K} \left[\sigma' \left(\sum_{j=0}^{N} V_{Oij} x_t^{(j)} + U_{Oij} h_{t-1}^{(j)} \right) U_{Oik} \frac{\partial h_{t-1}^{(k)}}{\partial V_{Oij}} \right]. \end{cases} \tag{B.13}$$

Hence, during the forward phase, one can use (B.4) to evaluate the error terms and all components of h_t. After that, during the backward phase, one can first use (B.6), (B.8), (B.10), and (B.12) and then iterate (B.7), (B.9), (B.11), and (B.13) to evaluate all gradients at previous times t. Note that, unlike (B.3), rules (B.7), (B.9), (B.11) are no longer subject to the exploding (or vanishing) gradient issue. Indeed, the evaluation of the gradient does not involve the product of the matrix weights but instead the sum of terms proportional to the output value of the forget gate.

Appendix C

The Zebiak-Cane model

We shall introduce in this appendix a deep insight into the equations of ZC model (Zebiak and Cane, 1987b). We recall that the ZC model has been leveraged to generate the Time-Series of the work presented in Chapter 3. The ZC model is a set of equations with the scope of describing the coupled ocean-atmosphere flow on the equatorial β-plane. In geophysical fluid dynamics, according to Dijkstra (2005), the β-plane approximation consists of assuming that the ocean is small with respect to the zonal direction. In this context, one denotes with $L = r_0\Delta\theta$ the characteristic horizontal scale of the flow; with r_0 the radius of the Earth and $\Delta\theta$ a difference of latitudes. Instead of using spherical coordinates as the longitude ϕ and the latitude θ, the motion of the ocean on the Earth is described via a Cartesian coordinates system. The β-plane approximation, therefore, is valid of small values of $\gamma = \frac{L}{r_0} = \Delta\theta$; and accordingly, the local dimensionless Cartesian Coordinates takes the form

$$\begin{cases} x = (\phi - \phi_0)\cos\theta_0 \\ y = \frac{\theta-\theta_0}{\Delta\theta_0}. \end{cases} \tag{C.1}$$

with ϕ and θ the values, respectively, of the longitude and the latitude; while the couple (θ_0, ϕ_0) is a central point of a neighborhood where considering the β-plane approximation. We recall that in a neighborhood of $\theta = \theta_0$ we can always write

$$\begin{cases} \sin\theta = \sin\theta_0 + (\theta - \theta_0)\cos\theta_0 + \mathcal{O}(\theta^2) \\ \cos\theta = \cos\theta_0 - (\theta - \theta_0)\sin\theta_0 + \mathcal{O}(\theta^2). \end{cases} \tag{C.2}$$

The *Rossby parameter* (denoted with β) describes the longitudinal variations of the Coriolis force due to the sphericity of the Earth, and it is defined as

$$\beta = \frac{\partial f}{\partial y} = \frac{2\Omega\cos\phi}{r_0};$$

with Ω the angular velocity of the Earth ($7.2921 \cdot 10^{-5}\,\frac{rad}{s}$), and f the module of the Coriolis force, where

$$f = 2\Omega\sin\phi.$$

When considering the approximation (C.2), one can simply considers

$$\beta = \frac{2\Omega\cos\phi_0}{r_0} \tag{C.3}$$

as the Rossby parameter. Similarly to this, one can approximate the vorticity gradient β_0 as

$$\beta_0 = \frac{\beta L^2}{U}; \tag{C.4}$$

where U is the characteristic velocity of the oceanic flow in the horizontal direction; with $U = \mathcal{O}(10\text{ km})$. For a detailed derivation of the β-plane approximation, the reader will find a complete discussion in Pedlosky et al. (1987).

The *ocean component* of the ZC model describes the motion of the oceanic flow and is modeled via a *reduced gravity model*. Taking Dijkstra (2005) as a reference, the reduced gravity model assumed that the flow motion takes place on a two-layer ocean; where the bottom layer is at rest with density ρ, while the motion of long waves is considered on the top layer with density ρ. The top layer is assumed to have a depth equal to H_1, while the deepest layer has a depth denoted as H_2; the whole two-layer ocean model has a depth $H = H_1 + H_2$. The equations ruling the dynamics of the top (shallow) layer are

$$\begin{cases} \frac{\partial u}{\partial t} + a_m u - \beta_0 y v + g' \frac{\partial h}{\partial x} = \frac{\tau_x}{H}, \\ \beta_0 y u + g' \frac{\partial h}{\partial y} = 0, \\ \frac{\partial h}{\partial t} + a_m h + c_0^2 \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0; \end{cases} \tag{C.5}$$

with u and v are, respectively, the horizontal and the vertical velocities and h the depth of the top layer; τ^x is the zonal wind stress, $g' = \frac{g\rho}{\Delta\rho}$ is the reduced gravity (with g the constant of gravity), and $c_0 = \sqrt{g'H}$ is the *phase of speed* of the first baroclinic Kelvin mode, and a_m a linear damping coefficient. In (C.5) the ocean is assumed to be bounded by walls at $x = 0$ and $x = L$; the boundary conditions at the walls are

$$\begin{cases} \int_{\infty}^{\infty} dy\ u(0, y, t) = 0 \\ u(L, y, t) = 0; \end{cases} \tag{C.6}$$

all variables are bounded in the vertical direction y.

The motion for the surface layer is described as

$$\begin{cases} a_s u_s - \beta_0 y v_s = \frac{H_2 \tau^x}{H H_1 \rho} \\ a_s v_s - \beta_0 y u_s = 0. \end{cases} \tag{C.7}$$

with u_s and v_s the surface velocity, respectively, in the horizontal and the vertical direction; a_s is the linear damping coefficient.

The *sea surface temperature*, i.e. the evolution of the temperature of the surface layer, is described by

$$\frac{\partial T}{\partial t} + a_T (T - T_0) + \frac{w_1}{H_1} \mathcal{H}(w_1)(T_s(h)) + u_1 \frac{\partial T}{\partial x} + v_1 \frac{\partial T}{\partial y}; \tag{C.8}$$

with a_T a linear damping coefficient, $u_1 = u_s + u$, $v_1 = v_s + v$, $\mathcal{H}(\cdot)$ is a continuous approximation of the Heaviside function, and T_0 is the *radiation equilibrium tempererature*. The quantity $T_s(h)$ express the dependence of the sea surface temperature T with respect to the *thermocline depth* h, namely

$$T_s(h) = T_{s0} + (T_0 - T_{s0}) \tanh\left(\frac{h + h_0}{\hat{H}} \right); \tag{C.9}$$

with h_0 and $\widehat{H}$ represent some parameters that control the steepness and the scale of T_s, while T_{s0} is the characteristic temperature of the surface layer.

The *atmosphere component* of the ZC equations are described via Gill model (Gill, 1980). The equations of the atmosphere model are

$$\begin{cases} \frac{\partial u_a}{\partial t} + A_m u_a - \beta_0 y v_a - \frac{\partial \Theta}{\partial x} = 0 \\ \frac{\partial v_a}{\partial t} + A_m v_a + \beta_0 y u_a - \frac{\partial \Theta}{\partial y} = 0 \\ \frac{\partial \Theta}{\partial t} + A_m \Theta - c_a^2 \left(\frac{\partial u_a}{\partial x} + \frac{\partial v_a}{\partial x} \right) = 0; \end{cases} \tag{C.10}$$

with u_a and v_a, respectively, the horizontal and vertical velocities of atmosphere flux, Θ is the geopotential height, and A_m is a linear damping coefficient. α_T is the proportionality constant that linearly relates the anomalies of the sea surface temperature to the heat fluxes from the ocean that drive the atmosphere flux (Feng and Dijkstra, 2017). The zonal wind stress τ^x is divided into two components, namely

$$\begin{cases} \tau^x = \tau^x_{ext} + \tau^x_c, \\ \tau^x_{ext} = -\tau_0 e^{-\frac{1}{2}\left(\frac{y}{L_a}\right)^2}. \end{cases} \tag{C.11}$$

τ^x_{ext} is the external wind stress, while τ^x_c is the zonal wind field. The external wind stress is assumed to be symmetric with respect to the equator; L_a is the Rossby deformation radius (Feng and Dijkstra, 2017).

Appendix D

Survival Analysis

Survival Analysis (SA) is a branch of statistics whose primary concern is the estimation of the time at which one or multiple events occur, given the property of a sample population (Allison, 2014). This field of statics finds numerous applications in medicine and healthcare (Ohno-Machado, 2001; Flynn, 2012; Putter et al., 2007). In many clinical studies, for example, SA is typically utilized for estimating the time-to-death T of some patients belonging to a cohort of study; see Salinas-Escudero et al. (2020). Also, SA focuses the attention on the characteristics defining one sample population and how their inhomogeneity might affect the estimation of the survival times (Aalen et al., 2008). In addition, another peculiarity of SA is the possibility of estimating the survival times of homogenous subsamples; and thus understanding the dynamical evolution of samples all sharing the same intrinsic characteristics (Aalen et al., 2008).

The most fundamental quantity in survival analysis is undoubtedly the *Survival Function* $S(t)$, i.e., the probability of one subject to experience one particular event at some precise time T, given the fact that any other event has not occurred before; in formulae, one has

$$S(t) = \mathbf{P}(T > t). \tag{D.1}$$

By complementarity, $S(t) = 1 - \mathbf{P}(T \leq t)$, with $\mathbf{P}(T \leq t)$ the the *distribution function of the events*. We shall denote the distribution function of the events with

$$f(t) = \mathbf{P}(T \leq t). \tag{D.2}$$

In addition, one can introduce another fundamental quantity, expressing the risk that one event occurs in an infinitesimal interval, i.e., the *hazard rate*. So, we have

$$\alpha(t) = \lim_{h \to 0} \frac{\mathbf{P}(t \leq T \leq t + h) | T \geq t)}{h}. \tag{D.3}$$

One can prove that the hazard rate is directly related to the survival function and vice versa, namely

$$\alpha(t) = \frac{S(t)}{f(t)};$$

or alternatively

$$S(t) = \exp\left[- \int_0^t du \, \alpha(u) \right]. \tag{D.4}$$

Another relevant aspect of SA is represented by the *censoring* of data (Putter et al., 2007; Allison, 2014). We refer to the censoring of data as that condition in which the observations are missing. Such a loss of information raises numerous problems about the correctness of the estimation that one can obtain. In other words, the absence of data points in some intervals can lead to the impossibility of formulating correct predictions. Depending on the regression model one uses, many strategies are often available to contrast this sort of issue. Anyway, censoring can be categorized as follows:

- Left censoring: observations are missing before (at left of) some time τ.
- Interval censoring: observations are missing in a precise interval (say, $[\tau_0, \tau_1]$)
- Right censoring: observations are missing after (at right of) some time τ.
- Random (or non-informative) censoring represents a situation in which the censoring time U has not been fixed a prior but is independent with respect to the time-to-event T. Hence, the i-th subject will end his/her trajectory at the time min (U_i, T_i). Consequently, when the relation $T_i > U_i$ is valid for a patient, we can consider that patient as right censored at U_i.

D.1 Competing risk models

Competing Risk (CR) models belong to a particular class of Multi-States Model (Putter et al., 2007). To better introduce some theoretical backgrounds about this class of models, we can consider the following toy example: *we have a cohort of study consisting of a class of subjects who may die from breast cancer or from a stroke but cannot die from both.* Thus, after being kept under observation for a certain amount of time, each patient ends his/her therapy with one of the two failing causes; in this toy example, these events are "die because of breast cancer" or "die because of a stroke." In this thesis, we applied this class of models in section 4.3.6. In CR models, one main interest is to quantify how much the baseline characteristics (e.g., sex, age, treatment, or comorbidities) and some other time-dependent characteristics (e.g., some time-dependent clinical predictors) can affect the exposure of patients to one specific cause. In doing so, it is fundamental to consider that other causes of failure might happen. CR models attempt to address the estimation of the survival probability of subjects susceptible to several types of failing causes. Usually, the observations for the time-to-event variable T represent what is mainly of interest in these kinds of models. This variable usually indicates the moment one competing event occurs.

We denote with δ the causes of failure ($\delta = j$) or the censoring ($\delta = 0$) and with $\mathbf{Z}$ the realization of some covariates (i.e., both baseline and time-dependent characteristics). Therefore a CR process (denoted as $(X_t)_{t \in T}$), associated with a CR model, points to the states where a subject is at every time t, i.e., in our case we consider $X_t \in \{0, 1, \dots, J\}$. Each subject starts in the initial state 0 at $t = 0$, namely

$$\mathbf{P}(X_0 = 0) = 1.$$

Each subject can move into the state of either the event of interest or in one of the remaining $J - 1$ competing causes. The time at which one subject leaves the initial state is defined as

$$T = \min\{t > 0 | X_t \neq 0\}. \tag{D.5}$$

When considering the censoring time C (e.g., right censoring) the actual time that one subject leaves the initial state is

$$\tilde{T} = \min\{T, C\}; \tag{D.6}$$

where T is the time-to-event as defined in (D.5), while the event observed is

$$\tilde{\delta} = \mathbf{1}(T \leq C) \cdot X_t; \tag{D.7}$$

where $\mathbf{1}(\cdot)$ is the indicator function.

In CR models, the definition of the hazard rate (see (D.3)) is opportunely adapted by taking into account the diversity of events that a subject might experience. Thus, one defines the *cause-specific hazard* (for the j-th cause) as

$$\lambda_j(t) = \lim_{h \to 0} \frac{\mathbf{P}(t \leq T \leq t + h, X_T = j | T \geq t)}{h}; \quad j \in \{0, 1, ..., J\}. \tag{D.8}$$

Note that

$$\lambda(t) = \sum_{j=0}^{J} \lambda_j(t),$$

i.e. the sum of all the risk accounted to each competing cause is equal to the hazard rate stated by (D.3); the hazard rate expressing the risk of an "all-causes" event. Practically, $\lambda_j(t)$ represents a quantity that can be directly estimated from observations. For this reason, the estimation of cause-specific hazards has represented the broadest and most widely used approach in medical research (Prentice et al., 1978). Similar to the cause-specific hazard, one can also define the *cumulative cause-specific hazard*, namely

$$\Lambda_j(t) = \int_0^t dt\ \lambda_j(t). \tag{D.9}$$

Another interesting quantity involving the cause-specific hazard rate is the *cumulative incidence function*, namely

$$\mathbf{P}(T \leq t; X_T = j) = \int_0^t du\ \mathbf{P}(T > u-)\lambda_j(u-); \tag{D.10}$$

where $\mathbf{P}(T > u-)$ denotes the survival function evaluated just before the time u.

The estimation of the cause-specific hazards remains, therefore, the main target of parametric, non-parametric models, and semi-parametric models. We now consider one of the most famous and popular semi-parametric models: Cox proportional hazards model (Cox, 1972); a.k.a Cox-model. In this model, the j-th cause-specific hazard takes the following form:

$$\lambda_j(t|\mathbf{Z}) = \lambda_{0j}(t) \exp(\beta_j^T \mathbf{Z}); \tag{D.11}$$

where $\lambda_{0j}(t)$ is the so-called baseline hazard function (i.e., a covariate-free hazard term), β_j denotes the regressors (i.e., an array of unknown regression parameters to fit on the observations) associated with the j-th cause, and $\mathbf{Z}$ is the array of the covariates (observations). Note that when the covariates are time-fixed and assumed as non-informative (i.e., $\mathbf{Z} = 0$), then the hazard rate is uniquely described by the baseline hazard $\lambda_{0j}(t)$. Conversely, the presence of informative covariates ($\mathbf{Z} \neq 0$, for example $\mathbf{Z} = 1$) introduces a modulation term $\exp(\beta_j^T \mathbf{Z})$ in the baseline hazard $\lambda_{0j}(t)$. As a result, each of the regressors β is exclusively associated with one of the covariates of $\mathbf{Z}$.

The more the magnitude of one regressor β_j, the more the corresponding covariate $\mathbf{Z}$ contributes to determining the high risk of exposure to the j-th failure cause. The Cox model usually does not make any assumption on the shape of $\lambda_{0j}(t)$. On the other hand, the so-called *hazard ratios* (i.e., the effects due to a single covariate as expressed by the term $\exp(\beta_j^T \mathbf{Z})$) remain unchanged at any time t. In some cases, one can relax the proportionality assumption when it appears too strict; that is, one can consider $\mathbf{Z}$ and regressors β as dependent on time.

The parameters of (D.11) can be estimated from data via Maximum Likelihood Principle (Beyersmann et al., 2011); the likelihood function is the following:

$$\mathcal{L}(\beta) = \prod_t \prod_{i=0}^{N} \prod_{j=1}^{J} \left(\frac{\exp \beta_j Z_{ji}}{S_j^{(0)}(\beta, t)} \right)^{\Delta N_{ij}(t)} . \tag{D.12}$$

In (D.12), the first product runs over all time points at which observations were acquired, the second product runs over all patients, and the third product over all competing events. Note that Z_{ji} represents the j-th cause specific covariate of the i-th subject. The quantity $S_j^{(0)}(\beta, t)$ is the *weighted risk set* and it is defined as

$$S_j^{(0)}(\beta, t) = \sum_{i=1}^{N} \exp \beta_j Z_{ji} Y_i(t); \tag{D.13}$$

with $Y_i(t)$ the individual at-risk process for the i-th subject, i.e.

$$Y_i(t) = \mathbf{1}(L_i < t \leq \min\{T_i, C_i\}). \tag{D.14}$$

Also, in (D.12), $N_i j(t)$ denotes the *cause-specific counting process* of the i-th patient with respect to the j-th cause, namely

$$N_{ij}(t) = \mathbf{1}(\min\{T_i, C_i\}, L_i < T_i, \leq C_i, X_{T_i} = j); \tag{D.15}$$

and accordingly, its increment at time t is

$$\Delta N_{ij}(t) = N_{ij}(t) - N_{ij}(t-). \tag{D.16}$$

In (D.16), L_i denotes the *left-truncation* censoring time, while C_i the *right-truncation* censoring time. NOthe that the same notation has been used for both (D.13) and (D.14).

It is worthwhile to mention that functional form in (D.12) is the *partial likelihood* (Cox, 1975). Formally, one expects the likelihood function to contain both the baseline hazard of (D.12) and the regressors β. However, terms as λ_{0j} are *nuisance parameters* and are left entirely unspecified. The time intervals between observed event times should not contain any information on β (Beyersmann et al., 2011).

Summary

Convolutional Neural Networks (CNN) are the primary concern of this thesis. This type of DL algorithm represents a subclass of ANN designed to solve both classification and regression problems; they are often applied in visual imaging analysis. The idea behind these models lies in the sequential combination of *convolutions*, *non-linear activation functions*, and *max-pooling* operators to encode different classes of data distinctively. The goal is, therefore, to match different encodings to equally different data classes. The best way to encode the instances of several classes of data is established during the learning phase, where the ANN learn which are the most salient feature to capture in a sample image by means of the recursive application of convolution and max-pooling operators.

The most relevant aspect of this thesis lies in the interpretation of the analysis provided by CNN by means of the construction of saliency maps, revealing which areas of the input data are captured by CNN during or after its learning phase. CNN and all other types of ANN are often regarded as *black-box* models, namely those models over which one has neither complete control nor a little knowledge about what relation has been established to generate a specific output value from the input data. In the last decades, several algorithms have been developed in the framework of Explainable Artificial Intelligence (XAI); these techniques enable one to investigate how the flux of information of the input data is propagated throughout CNN. All these methods aim to construct saliency maps, i.e., a graphical representation of the degree of relevance in each subdomain of the input data. Therefore, we exploited these techniques' potentiality to interpret the activity of CNN models.

In this thesis, the results of 3 different applications of CNN in 3 different contexts are discussed and interpreted:

1. **Binary Classification:** a 1-D CNN was implemented to classify the Raman spectra of both healthy and tumor cells. Raman spectroscopy is a technique based on the elastic scattering of incident photons over a sample, i.e., Raman scattering. We used only one type of monochromatic incident radiation to study the response of a biological sample due to the excitation caused by the incident radiation. In particular, we focused on the Raman spectra, which represent the intensity of Raman radiation as a function of its frequency; usually, one prefers to replace the latter with the difference in wavenumbers between the scattered and the incident radiation, i.e., the Raman shift. Thus, we analyzed the Raman spectra derived from two different types of cellular culture, namely human skin cells and cells extracted from the human intestinal mucosa.

   In the first case, one considers 3 binary classification problems: human skin cells and malignant melanoma cell line A375; human skin cells and malignant melanoma cell line SK-MEL-28; and cell samples belonging to both the malignant cell lines A375 and SK-MEL-28.

Similar to the first case, in the second case, one considers 3 binary classification problems: human colorectal healthy cells and malignant cells of line HT-29; human colorectal healthy cells and malignant cells of line CaCO; and cell samples belonging to both the malignant cell lines HT-29 and CaCO.

Unlike the standard ML algorithms, the analysis of the results revealed that 1-D CNN is exceptionally accurate in recognizing both healthy and malignant cells; similar results have been observed with either high-definition or low-definition spectra. The interpretation provided through saliency maps revealed that 1-D CNN captures the information concerning four specific physicochemical properties of the cellular samples: the metabolism of proline in cancer cells, changes in the lipid membrane of cancer cells, the alteration of CH groups in the DNA of cancer cells and the peculiarity of cancer cells to be unevenly arranged on the nanowire substrate on which each sample is placed to be analyzed in the spectrometer.

Hence, we outlined how the 1-D CNN can distinguish the Raman spectra of cancer cells. The predictions are intrinsically connected with the recognition of specific physicochemical properties of the samples analyzed. Therefore, CNNs are expected to be a promising opportunity to enhance the analysis of nanomedicine data.

2. **Time-Series Classification:** Several 1-D CNN were implemented to classify Time-Series concerning events such as *el Niño* e *la Ninã*. We used synthetic data from the Zebiac-Cane (ZC) model. ZC model is a set of coupled partial differential equations designed to simulate the motion of the Southern Pacific Ocean and the evolution of its superficial temperature. In this work, a 1-D CNN model has been exploited in *dystorted physics experiments*; that is, the 1-D CNN was trained on data generated with a specific configuration of the parameters ruling the velocity of the equatorial waves and the *upwelling*. Upwelling refers to the rising of cold and dense masses of water from the deepest layers toward the superficial layers of the ocean. The configuration of the parameters has been conceived to simulate some scenarios that could be either similar or dissimilar to the actual oceanographic observations. For each configuration, we trained a 1-D CNN model. The interpretation of the predictions by means of saliency maps revealed that the analysis performed by the 1-D CNN models, with a prediction window of 9 months, aims to capture precise oscillating patterns of the equatorial waves or feedback processes in the thermocline. For instance, the individuation of the exact combination of patterns as peaks and valleys in specific regions of the input data turned out to be an essential feature in the activity recognition of the 1-D CNN. The study of the 1-D CNN predictions on distorted physics data offers the benefit of studying which features can be captured by such a model, even when dealing with more complex processes.

   The predictive performance of 1-D CNN models have been compared with another class of ANN broadly used in the forecasting of *el Niño* and *la Niña*, i.e., the Gaussian Density Network (GDN). GDN models turned out to be less predictive than the 1-D CNN, with 0.10 less of AUROC.

3. **Classification of hospital data:** a 1-D CNN was implemented to include the information contained in *high-frequency data* into a longitudinal study aiming to predict the onset of *nosocomially acquired infections* in the Intensive Care Unit. We used 48-hour prediction windows.

This study was conducted within the *Molecular Diagnosis and Risk Stratification of Sepsis study* (MARS); therefore one included a total of 5538 patients in the cohort. These patients had a stay larger than 48 hours in the Universiteit Medical Centrum Utrecht from January 2011 to December 2018.

In this work, we refer to high-frequency data as those data acquired from the ICU monitors (these data are sampled with a sampling frequency equal to one minute); they differ from the *low-frequency data* denoting all other explanatory variables typically used in this kind of longitudianl studies (e.g., laboratory values, bacterial colonization, patients' consciousness scores, temperature records, other variables describing the quality of both the respiration and the circulation of patients). The low-frequency have been sampled every 8 hours. In addition, we considered the baseline characteristics of patients, such as age, sex, comorbidities, etc. The event we aim to study is, therefore, identified by either the beginning of the antibiotic treatment or the ascertaining of an infectious episode by means of a blood culture.

The analysis of data and the prediction of the onset of an acquired infection is modeled through a *two-steps modeling*:

(a) a 1-D CNN model is implemented to investigate the longitudinal evolution of the electronic health records of each patient (we considered 5 vital signals, such as heart rate, arterial blood pressure, pulse, saturation level, and breath rate). All these 5 signals are sampled with a sampling frequency of one minute. The 1-D CNN is trained to capture those features of these 5 vital signals that are the most predictive; the model, therefore, outputs the *risk score of infection*.

(b) A *Landamrk model* is implemented to estimate the probability of infection of one patient given his/her previous clinical history at any moment of the ICU stay. Specifically, we implemented the *Landmark Cox model* with two competing events to model the *hazard rate* that one patient gets infected. The competing risks are the onset of an acquired infection or one of the two disjointed events, "death in ICU" or "discharge from ICU."

The Landmark Cox model was fitted on both the low-frequency data and the baseline characteristics (LCHP model); next, we fitted another Landmark Cox on the data used with the LCHP model and the risk score of infection elaborated by the CNN (DeepLCHP model).

The results showed that the risk score of infection could increase by 0.03 the AUROC of the 48-hour LCHP model. If one considers only the first infectious episodes (i.e., a patient is discarded after getting infected for the first time), then the risk score of infection incremented by 0.06 the AUROC of the LCHP model.

The interpretation of the 1-D CNN risk score of infection has been accomplished only when considering the first infectious episodes because, in this case, the 1-D CNN model is more accurate. Thus, the saliency maps have been constructed to highlight the most relevant 8-hour vital signals. Per each of these 8-hour signals, we evaluated both traditional statistics (e.g., mean value, standard deviation, etc.) and more complex non-linear statistics (e.g., Approximate Entropy, Permutation Entropy). All these statics represented the explanatory variables of a MLR model. More specifically, we compared the predictive performance of four models on three exact moments of patients' ICU stay (data 3, day 7, and day 10): the 1-D CNN

model, a MLR model fitted on the statics evaluated on the most salient 8-hours signals, a MLR model fitted on both the low-frequency signals and the baseline characteristics, and a MLR model fitted on the low-frequency signals, the baseline characteristics, and the statistics evaluated on the most salient 8-hours signals. From comparing the predictive performances of all 4 models, we observed that the statistics extracted from the 8-hour most salient signals can increase the prediction formulated by means of the low-frequency only. On day 3, we observed an increase of 0.02 in the AUROC, on day 7 of 0.03, and day 10 of 0.07. Thus, the recognition activity of the most salient features of 5 vital signals can be expressed by means of the combination of simple and complex statistics.

The interpretation of the activity recognition of the 1-D CNN model was connected to the representation of authentic clinical situations On days 3 and 10, the 1-D CNN mainly recognizes those patients experiencing symptoms of systemic inflammatory response syndrome (SIRS). Whereas on day 3, when ICU care may not have progressed decisively, the identification of symptoms related to infection remains somewhat premature.

This represents theoretical evidence that CNN models can make an exciting and decisive contribution in predicting the first episode of ICU-acquired infections while basing their predictions on the tracing of an inflammatory stage that identifies with sepsis when it occurs together with an infection.

Samenvatting

Convolutionale Neurale Netwerken (CNN) staan centraal in dit proefschrift. Dit type DL-algoritme vertegenwoordigt een subklasse van ANN die is ontworpen om zowel classificatie- als regressieproblemen op te lossen; ze worden vaak toegepast bij visuele beeldvormingsanalyse. Het idee achter deze modellen ligt in de sequentiële combinatie van *convolutions*, *niet-lineaire activeringsfuncties* en *max-pooling*-operators om verschillende gegevensklassen onderscheidend te coderen. Het doel is daarom om verschillende coderingen af te stemmen op even verschillende gegevensklassen. De beste manier om de instanties van verschillende gegevensklassen te coderen, wordt vastgesteld tijdens de leerfase, waar de ANN leert welke de meest opvallende eigenschap is om in een voorbeeldafbeelding vast te leggen door middel van de recursieve toepassing van convolutie- en max-pooling-operators.

Het meest relevante aspect van dit proefschrift ligt in de interpretatie van de analyse van CNN door middel van de constructie van saliency maps, die onthullen welke delen van de invoergegevens door CNN worden vastgelegd tijdens of na de leerfase. CNN en alle andere soorten ANN worden vaak beschouwd als *black-box*-modellen, namelijk die modellen waarover men geen volledige controle heeft, noch enige kennis heeft over welke relatie er tot stand is gebracht om een specifieke uitvoerwaarde te genereren uit de invoergegevens . In de afgelopen decennia zijn er verschillende algoritmen ontwikkeld in het kader van Explainable Artificial Intelligence (XAI); deze technieken stellen ons in staat om te onderzoeken hoe de informatiestroom van de invoergegevens door CNN wordt verspreid. Al deze methoden zijn gericht op het construeren van opvallende kaarten, d.w.z. een grafische weergave van de mate van relevantie in elk subdomein van de invoergegevens. Daarom hebben we het potentieel van deze technieken benut om de activiteit van CNN-modellen te interpreteren.

In dit proefschrift worden de resultaten van 3 verschillende toepassingen van CNN in 3 verschillende contexten besproken en geïnterpreteerd:

1. **Binaire classificatie:** een 1-D CNN werd geïmplementeerd om de Raman-spectra van zowel gezonde cellen als tumorcellen te classificeren. Raman-spectroscopie is een techniek die is gebaseerd op de elastische verstrooiing van invallende fotonen over een monster, d.w.z. Raman-verstrooiing. We gebruikten slechts één type monochromatische invallende straling om de respons van een biologisch monster te bestuderen vanwege de excitatie veroorzaakt door de invallende straling. We hebben ons met name gericht op de Raman-spectra, die de intensiteit van Raman-straling weergeven als functie van de frequentie; meestal geeft men er de voorkeur aan om de laatste te vervangen door het verschil in golfgetallen tussen de verstrooide en de invallende straling, d.w.z. de Raman-verschuiving. Zo hebben we de Raman-spectra geanalyseerd die zijn afgeleid van twee verschillende soorten celcultuur, namelijk menselijke huidcellen en cellen die zijn geëxtraheerd uit het menselijke darmslijmvlies.

   In het eerste geval beschouwt men 3 binaire classificatieproblemen: menselijke huidcellen en

kwaadaardige melanoomcellijn A375; menselijke huidcellen en kwaadaardige melanoomcellijn SK-MEL-28; en celmonsters die behoren tot zowel de kwaadaardige cellijnen A375 als SK-MEL-28. Vergelijkbaar met het eerste geval, beschouwt men in het tweede geval 3 binaire classificatieproblemen: menselijke colorectale gezonde cellen en kwaadaardige cellen van lijn HT-29; menselijke colorectale gezonde cellen en kwaadaardige cellen van lijn CaCO; en celmonsters die behoren tot zowel de kwaadaardige cellijnen HT-29 als CaCO.

In tegenstelling tot de standaard ML-algoritmen, onthulde de analyse van de resultaten dat 1-D CNN uitzonderlijk nauwkeurig is in het herkennen van zowel gezonde als kwaadaardige cellen; vergelijkbare resultaten zijn waargenomen met high-definition of low-definition spectra. De interpretatie die via saliency-kaarten werd gegeven, onthulde dat 1-D CNN de informatie vastlegt over vier specifieke fysisch-chemische eigenschappen van de cellulaire monsters: het metabolisme van proline in kankercellen, veranderingen in het lipidemembraan van kankercellen, de verandering van CH-groepen in het DNA van kankercellen en de eigenaardigheid van kankercellen om ongelijk gerangschikt te zijn op het nanodraadsubstraat waarop elk monster wordt geplaatst om in de spectrometer te worden geanalyseerd.

Daarom hebben we geschetst hoe de 1-D CNN de Raman-spectra van kankercellen kan onderscheiden. De voorspellingen zijn intrinsiek verbonden met de herkenning van specifieke fysisch-chemische eigenschappen van de geanalyseerde monsters. Daarom wordt verwacht dat CNN's een veelbelovende kans zijn om de analyse van nanogeneeskundegegevens te verbeteren.

2. **Tijdreeksclassificatie:** Er zijn verschillende 1-D CNN geïmplementeerd om tijdreeksen te classificeren met betrekking tot gebeurtenissen zoals *el Niño* en *la Ninã*. We gebruikten synthetische data van het Zebiac-Cane (ZC) model. Het ZC-model is een set van gekoppelde partiële differentiaalvergelijkingen die zijn ontworpen om de beweging van de zuidelijke Stille Oceaan en de evolutie van de oppervlaktetemperatuur te simuleren. In dit werk is een 1-D CNN-model gebruikt in *vervormde fysica-experimenten*; dat wil zeggen, de 1-D CNN werd getraind op gegevens die werden gegenereerd met een specifieke configuratie van de parameters die de snelheid van de equatoriale golven en de *opwelling* bepalen. Opwelling verwijst naar het opstijgen van koude en dichte watermassa's van de diepste lagen naar de oppervlakkige lagen van de oceaan. De configuratie van de parameters is bedacht om enkele scenario's te simuleren die al dan niet vergelijkbaar kunnen zijn met de daadwerkelijke oceanografische waarnemingen. Voor elke configuratie hebben we een 1-D CNN-model getraind. De interpretatie van de voorspellingen door middel van saliency-kaarten onthulde dat de analyse uitgevoerd door de 1-D CNN-modellen, met een voorspellingsperiode van 9 maanden, gericht is op het vastleggen van precieze oscillerende patronen van de equatoriale golven of feedbackprocessen in de thermocline. Zo bleek de individualisering van de exacte combinatie van patronen als pieken en dalen in specifieke regio's van de invoergegevens een essentieel kenmerk te zijn in de activiteitsherkenning van de 1-D CNN. De studie van de 1-D CNN-voorspellingen op vervormde natuurkundige gegevens biedt het voordeel om te bestuderen welke kenmerken kan door zo'n model worden vastgelegd, zelfs als het gaat om complexere processen.

De voorspellende prestaties van 1-D CNN-modellen zijn vergeleken met een andere klasse van ANN die algemeen wordt gebruikt bij het voorspellen van *el Niño* en *la Niña* , d.w.z. het

Gaussian Density Network (GDN). GDN-modellen bleken minder voorspellend te zijn dan de 1-D CNN, met 0,10 minder van AUROC.

3. **Classificatie van ziekenhuisgegevens:** een 1-D CNN werd geïmplementeerd om de informatie in *hoogfrequente gegevens* op te nemen in een longitudinaal onderzoek met als doel het begin van *nosocomiaal opgelopen infecties* op de Intensive Care te voorspellen. We gebruikten voorspellingsvensters van 48 uur.

   Deze studie is uitgevoerd binnen de *Molecular Diagnosis and Risk Stratification of Sepsis study* (MARS); daarom omvatte men in totaal 5538 patiënten in het cohort. Deze patiënten verbleven van januari 2011 tot december 2018 langer dan 48 uur in het Universitair Medisch Centrum Utrecht.

   In dit werk verwijzen we naar hoogfrequente gegevens als die gegevens die zijn verkregen van de ICU-monitoren (deze gegevens worden bemonsterd met een bemonsteringsfrequentie gelijk aan één minuut); ze verschillen van de *low-frequency data* die alle andere verklarende variabelen aangeven die doorgaans worden gebruikt in dit soort longitudinale studies (bijv. laboratoriumwaarden, bacteriële kolonisatie, bewustzijnsscores van patiënten, temperatuurgegevens, andere variabelen die de kwaliteit van zowel ademhaling en de circulatie van patiënten). De lage frequenties zijn elke 8 uur bemonsterd. Daarnaast hebben we gekeken naar de basiskenmerken van patiënten, zoals leeftijd, geslacht, comorbiditeit, enz. De gebeurtenis die we willen bestuderen, wordt daarom geïdentificeerd door het begin van de antibioticabehandeling of door het vaststellen van een infectieuze episode door middel van een bloedkweek.

   De analyse van gegevens en de voorspelling van het begin van een verworven infectie wordt gemodelleerd door middel van een *modellering in twee stappen*:

   (a) een 1-D CNN-model is geïmplementeerd om de longitudinale evolutie van de elektronische medische dossiers van elke patiënt te onderzoeken (we hebben 5 vitale signalen overwogen, zoals hartslag, arteriële bloeddruk, polsslag, verzadigingsniveau en ademhalingsfrequentie). Al deze 5 signalen worden bemonsterd met een bemonsteringsfrequentie van één minuut. De 1-D CNN is getraind om die kenmerken van deze 5 vitale signalen vast te leggen die het meest voorspellend zijn; het model voert daarom de *risicoscore van infectie* uit.

   (b) Er wordt een *Landamrk-model* geïmplementeerd om de kans op infectie van een patiënt te schatten, gegeven zijn/haar eerdere klinische geschiedenis op elk moment van het verblijf op de IC. We hebben met name het *Landmark Cox-model* geïmplementeerd met twee concurrerende gebeurtenissen om de *hazard rate* te modelleren waarmee één patiënt geïnfecteerd raakt. De concurrerende risico's zijn het begin van een verworven infectie of een van de twee onsamenhangende gebeurtenissen, "overlijden op de IC" of "ontslag van de IC".

   Het Landmark Cox-model werd aangepast op zowel de laagfrequente gegevens als de basiskenmerken (LCHP-model); vervolgens hebben we nog een Landmark Cox gemonteerd op de gegevens die zijn gebruikt met het LCHP-model en de risicoscore van infectie uitgewerkt door de CNN (Deep-LCHP-model).

   De resultaten toonden aan dat de risicoscore van infectie zou kunnen toenemen met 0,03

de AUROC van het 48-uurs LCHP-model. Als men alleen rekening houdt met de eerste infectieuze episoden (d.w.z. een patiënt wordt weggegooid nadat hij voor de eerste keer is geïnfecteerd), dan wordt de risicoscore van infectie verhoogd met 0,06 de AUROC van het LCHP-model.

De interpretatie van de 1-D CNN-risicoscore van infectie is alleen bereikt bij het beschouwen van de eerste infectieuze episodes, omdat in dit geval het 1-D CNN-model nauwkeuriger is. Daarom zijn de saliency-kaarten geconstrueerd om de meest relevante 8-uurs vitale signalen te markeren. Per elk van deze 8-uursignalen evalueerden we zowel traditionele statistieken (bijv. gemiddelde waarde, standaarddeviatie, enz.) als complexere niet-lineaire statistieken (bijv. Geschatte entropie, permutatie-entropie). Al deze statica vertegenwoordigden de verklarende variabelen van een MLR-model. Meer specifiek vergeleken we de voorspellende prestaties van vier modellen op drie exacte momenten van verblijf op de IC van de patiënt (gegevens 3, dag 7 en dag 10): het 1-D CNN-model, een MLR-model aangepast aan de statische gegevens geëvalueerd op de meest opvallende 8-uurssignalen, een MLR-model aangepast aan zowel de laagfrequente signalen als de basislijnkarakteristieken, en een MLR-model aangepast aan de laagfrequente signalen, de basislijnkarakteristieken, en de statistieken geëvalueerd op de meest opvallende 8-uursignalen. Door de voorspellende prestaties van alle 4 de modellen te vergelijken, hebben we vastgesteld dat de statistieken die zijn geëxtraheerd uit de meest opvallende signalen van 8 uur, de voorspelling kunnen verhogen die is geformuleerd door middel van alleen de lage frequentie. Op dag 3 zagen we een stijging van 0,02 in de AUROC, op dag 7 van 0,03 en op dag 10 van 0,07. Zo kan de herkenningsactiviteit van de meest opvallende kenmerken van 5 vitale signalen worden uitgedrukt door middel van de combinatie van eenvoudige en complexe statistieken.

De interpretatie van de activiteitsherkenning van het 1-D CNN-model werd gekoppeld aan de representatie van authentieke klinische situaties Op dag 3 en 10 herkent de 1-D CNN voornamelijk patiënten met symptomen van het systemische inflammatoire responssyndroom (SIRS). Terwijl op dag 3, wanneer de IC-zorg misschien nog niet doorslaggevend is gevorderd, de identificatie van symptomen die verband houden met infectie nog wat voorbarig is.

Dit vertegenwoordigt theoretisch bewijs dat CNN-modellen een opwindende en beslissende bijdrage kunnen leveren bij het voorspellen van de eerste episode van ICU-verworven infecties, terwijl ze hun voorspellingen baseren op het opsporen van een ontstekings stadium dat zich identificeert met sepsis wanneer het samen met een infectie optreedt.

Riepilogo

Le reti neurali artificiali di tipo convoluzionale (Convolutional Neural Networks, CNN) sono l'argomento principale di questa tesi. Questo tipo di algoritmi di apprendimento profondo (Deep Learning, DL) rappresentano una classe di reti neurali artificiali (Artificial Neaural Networks, ANN) ideata per risolvere sia problemi di classificazione che di regressione; sono spesso applicate nell'analisi di immagini. L'idea dietro questi modelli sta nella combinazione di *convoluzioni*, *funzioni di attivazione non lineari* e operatori di *max-pooling* al fine di codificare in maniera distintiva diverse classi di dati. L'obbiettivo è quindi quello di far corrispondere codifiche diverse a differenti classi di dati. Il modo per stabilire la migliore codifica da attribuire a diverse classi di dati viene ottenuto durante la fase d'apprendimento dove le CNN apprendono quali sono le caratteristiche più salienti da catturare in un immagine mediante l'applicazione ricorsiva di opportune convoluzioni e operatori di max-pooling.

L'aspetto piú rilevante di questa tesi sta nell'interpretazione delle analisi fornite dalle CNN mediante la costruzione di *mappe di salienza*. Queste mappe rivelano quali aree dei dati ingresso vengono meglio catturate dalla CNN durante o dopo la sua fase di apprendimento. Le CNN, così come tutte le reti neurali artificiali, vengono spesso considerate delle scatole nere, ovvero modelli del quale non si ha il pieno controllo, né la minima conoscenza su quale relazione venga stabilita per generare un certo valore in uscita dai dati in ingresso. Numerosi algoritmi per la decifrabilitá dell'intelligenza artificiale (Explainable Artificial Intelligence, XAI) permettono di investigare su come il flusso di informazione dei dati in entrata venga analizzato all'interno delle CNN. Questi metodi sono finalizzati alla costruzione di mappe di salienza. Abbiamo quindi sfruttato la potenzialità di queste tecniche al fine di interpretare l'attività delle CNN.

In questa tesi vengono discussi ed interpretati i risultati di tre differenti applicazioni delle CNN in tre diversi contesti

1. **Classificazione binaria**: una rete convoluzionale 1-D (1-D CNN) è stata implementata per classificare gli spettri Raman di cellule sane e tumorali. La spettroscopia Raman è una particolare tecnica spettroscopica basata sulla diffusione elastica di fotoni incidenti su un campione, ovvero la diffusione Raman. Abbiamo usato un solo tipo di radiazione incidente monocromatica per vedere la risposta del campione all'eccitamento dovuto alla radiazione incidente. In particolare abbiamo studiato lo spettro Raman di un campione che rappresenta come i valori dell'intensità della radiazione (Raman) emessa dal campione possano variare rispetto alla sua frequenza; di solito si preferisce sostituire l'ultimo con la differenza in numeri d'onda tra radiazione diffusa e radiazione incidente, cioè il Raman shift. Quindi abbiamo analizzato gli spettri Raman ottenuti da due tipi differenti di colture cellulari, ovvero cellule cutanee umane e cellule estratte dalla mucosa intestinale umana. Questo lavoro è stato diviso in due fasi. Nella prima fase, si considerano tre problemi di classificazione binaria: cellule

cutanee sane contro melanoma maligno della linea cellulare A375, cellule cutanee sane contro melanoma maligno della linea cellulare SK-MEL-28 e cellule appartenenti ad entrambe le linee cellulari maligne A375 e SK-MEL-28. Analogamente al primo caso, nel secondo caso, si considerano tre problemi di classificazione binaria: cellule intestinali sane contro cellule tumorali della linea cellulare HT-29, cellule intestinali sane contro cellule tumorali della linea cellulare CaCO e cellule intestinali tumorali di entrambe le linee cellulari maligne HT-29 e CaCO.

A differenza dei metodi tradizionali di apprendimento automatico (Machine Learning, ML) l'analisi dei risultati ha rivelato che la 1-D CNN è estremamente accurata sia nel riconoscere cellule sane dalle cellule tumorali che diversi tipi di linee cellulari maligne; risultati accurati sono stati ottenuti sia nel caso di spettri con un'alta risoluzione che per spettri con una risoluzione più bassa. L'interpretazione mediante le mappe di salienza ha rivelato che la rete convoluzionale 1-D cattura le informazioni inerenti a quattro specifiche propietà fisicochimiche dei campioni cellulari: il metabolismo di prolina nelle cellule tumorali, cambiamenti nella mebrana lipidica delle cellule cancerogene, l'alterazione dei gruppi CH nel DNA delle cellule cancerogene e la particolarità delle cellule cancerogene di disporsi in maniera non omogenea sul substrato di nanofili su quale ogni campione viene posizionato per essere analizzato nello spettrometro.

Quindi in questo lavoro, viene evidenziato come una rete convoluzionale 1-D possa distinguere gli spettri Raman di cellule cancerogene e come le predizioni formulate siano legate al riconoscimento di precise propietá fisico-chimiche dei campioni analizzati. Le CNN si prefigurano quindi come un'opportunità valente per potenziare l'analisi dei dati di nanomedicina.

2. **Classificazione di Time-Series:** Diverse 1-D CNN sono state implementate per classificare eventi come *el Niño* e *la Ninã* da dati simulati mediante il modello Zebiak-Cane (ZC). Il modello ZC è un insieme di equazioni differenziali accoppiate alle derivate parziali usate per simulare il moto dell'oceano pacifico e l'evoluzione della sua temperatura superficiale nel tempo. In questo lavoro le 1-D CNN sono state coinvolte in esperimenti di *fisica distorta*, ovvero la 1-D CNN è stata addestrata dopo aver generato dati con una specifica configurazione dei dati che governano la velocità delle onde equatoriali e lo *upwelling*, ovvero la risalita di masse dense di acqua fredda dagli strati piú profondi verso la superficie dell'oceano. La configurazione dei parametri mirava a simulare scenari molto simili o molto dissimili dalle osservazioni oceanografiche reali. Per ogni configurazione abbiamo addestrato un modello 1-D CNN. L'interpretazione dei risultati mediante la costruzione di mappe di salienza indica che l'analisi eseguita dalla 1-D CNN con una finestra di previsione di 9 mesi mira a catturare precise caratteristiche delle onde equatoriali o di processi di feedback nel termoclino. Ad esempio, l'individuazione dell'esatta combinazione di pattern come picchi e valli in regioni specifiche dei dati di input si è rivelata una caratteristica essenziale nel riconoscimento dell'attività della 1-D CNN. Lo studio delle predizioni delle CNN su dati di fisica distorta offre il vantaggio di studiare quali caratteristiche possano essere catturate da un tale modello anche per processi più complessi.

   L'abilità delle CNN è stata confrontata con un'altra classe di reti neurali artificiali largamente usato nella predizione di eventi come *el Niño* e la *la Niña*, cioè la Gaussian Density Network (GDN). Il modello GDN rispetto al CNN risulta essere meno accurato di circa 0.10 punti di AUROC.

3. **Classificazione di dati ospedalieri:** un modello 1-D CNN è stato implementato per includere l'informazione proveniente da dati ad alta frequenza in uno studio longitudinale che mira a predire l'insorgenza di un *infezione nosocomiale acquisita* nel dipartimento di terapia intensiva. La finestra di predizione che si usa è di 48 ore. I dati utilizzati appartengono alla coorte di studi *Molecular Diagnosis and Risk Stratification of Sepsis study* (MARS); quindi sono stati inclusi nello studio un totale di *5538 pazienti* con un soggiorno superiore a 48 ore nel reparto di terapia intensiva del Universiteit Medical Centrum Utrecht nel periodo Gennaio 2011 - Dicembre 2018.

   Nello specifico ci riferiamo ai dati ad alta frequenza come ai quei dati provenienti dai monitor della terapia intensiva (dati campionati con un frequenza di campionamento pari ad un minuto); si differenziano dai *dati a bassa frequenza* che comprendono tutte le altre variabili tipicamente usate in questo tipo di studi longitudinali (valori di laboratorio, colonizzazioni batteriche, punteggi sulla coscienza dei pazienti, valori di temperatura, dati sulla respirazione e sull'attività cardiaca dei pazienti) la cui frequenza di campionamento è stata impostata a 8 ore. Oltre a questi vengono anche considerate le caratteristiche di base dei pazienti come l'età, il sesso, le comorbilità, etc. L'evento che si vuole studiare, cioè l'insorgenza di un'infezione acquisita, viene perciò identificato o come l'inizio della terapia antibiotica o l'accertamento di stato infettivo attraverso emocultura.

   L'analisi dei dati e la previsione della comparsa di un'infezione acquisita nel reparto di terapia intensiva viene analizzata attraverso una modellizzazione in due passaggi:

   (a) un modello 1-D CNN viene implementato per investigare l'evoluzione longitudinale della cartella clinica elettronica di ogni paziente (ogni cartella contiene 5 segnali vitali come frequenza cardiaca, pressione arteriosa, polso arterioso, saturazione e frequenza respiratoria) campionata ogni minuto. Il modello CNN viene addestrato a catturare le caratteristiche di questi 5 segnali vitali che si rivelano le più predittive; il modello infine elabora un punteggio di rischio di infezione.

   (b) Un *modello di Landmark* viene poi implementato per stimare la probabilità dell'insorgenza di un'infezione nosocomiale in qualunque momento del soggiorno di un paziente in terapia intensiva. Nello specifico è stato implementato un modello *Landmark di Cox* per modellizzare il *tasso di rischio* (hazard rate) di due eventi competitivi, ovvero l'insorgenza di un'infezione oppure la dimissione o la morte di un paziente in terapia intensiva. Il modello di Landmark di Cox viene quindi adattato sia per i dati bassa frequenza e le caratteristiche di base dei pazienti (modello LCHP) che per i dati usati nel modello LCHP insieme al punteggio di rischio di infezione elaborato dall'analisi dei dati ad alta frequenza mediante il modello CNN (Deep-LCHP).

   I risultati mostrano che il punteggio di rischio di infezione può incrementare di 0.03 punti di AUROC nella predizione a 48 ore del modello LCHP. Tuttavia se si restringe l'analisi ai soli primi episodi infettivi (cioè un paziente non può essere riammesso alla corte come un nuovo paziente dopo essersi infettato la prima volta), allora il punteggio di rischio del modello CNN incrementa le previsioni del modello LCHP di 0.06 punti di AUROC.

   L'interpretazione dei punteggi di rischio della 1-D CNN sono stati interpretati soltanto quando si considerano i primi episodi infettivi, poiché in questo caso il modello CNN risulta essere più accurato. Quindi le mappe di salienza sono state utilizzate per evidenziare quei pezzi di

segnali vitali di ampiezza 8 ore che risultano essere i più rilevanti. Per ognuno di questi pezzi sono state valutate sia delle statistiche tradizionali come la media o la deviazione standard che altre statistiche non lineari più elaborate come la entropia approssimata o l'entropia di permutazione. L'insieme di queste statistiche valutate sui pezzi più rilevanti dei segnali vitali sono state utilizzate per adattare un modello di Regressione Logistica Multipla (MLR). Nello specifico, per tre precisi momenti del soggiorno in terapia intensiva (giorno 3, giorno 7 e giorno 10), abbiamo confrontato le prestazioni di quattro modelli: il modello CNN, il modello MLR con le statistiche calcolate sui pezzi di più segnali rilevanti, il modello MLR con le informazioni a bassa frequenza insieme alle caratteristiche di base e un modello MLR che comprende le informazioni a bassa frequenza, le caratteristiche di base e le statistiche calcolate sui pezzi più salienti dei segnali vitali. Dal confronto dei risultati si osserva le statistiche estratte dai segnali di otto ore più salienti aumentano le predizioni formulate dai dati a bassa frequenza; al giorno 3 si ha un'incremento di 0.02 punti di AUROC, giorno 7 0.03 punti di AUROC e giorno 10 0.07 punti di AUROC. Quindi l'attività di riconosimento delle caratteristiche più predittive dei segnali vitali può essere espressa mediante la combinazione di statistiche più o meno complesse.

L'interpretazione dell'attivittà di riconoscimento del modello 1-D CNN è stato anche connesso a delle situazioni clinicamente riscontrabili. A giorno 7 e giorno 10 la 1-D CNN identifica sopratutto i pazienti che manifestano i sintomi di una Sindrome da Risposta Infiammatoria Sistemica (SIRS), mentre a giorno 3, quando le cure in terapia intensiva possono non aver progredito in maniera decisiva, l'identificazione di sintomatologie correlate ad un infezione rimane piuttosto prematura. Ciò rappresenta una prova teorica che i modelli CNN possono dare un contributo interessante e decisivo nella predizione del primo episodio di infezioni acquisite in terapia intensiva pur basando le loro predizioni sul rintracciamento di uno stadio infiammatorio che si identifica con una sepsi quando si manifesta insieme ad un'infezione.

Code Availability

Python codes and modules are available on GitHub.

For the work presented in chapter 2, the reader can refer to https://github.com/glancia93/Cancer-diagnosis-via-Raman-Spectroscopy.

For the work presented in chapter 3, the reader can refer to *https://github.com/glancia93/Physics-captured-by-data-based-methods-in-El-Nino-prediction_PyCODE*

For the work presented in chapter 4, the reader can refer to *https://github.com/glancia93/ICUAI-dynamic-prediction/blob/main/ICUAI_module.py*.

Bibliography

Aalen, O., Borgan, O., and Gjessing, H. (2008). *Survival and event history analysis: a process point of view*. Springer Science & Business Media.

Abramowitz, M. and Stegun, I. A. (1964). *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office.

Aggarwal, C. C. et al. (2018). Neural networks and deep learning. *Springer*, 10:978–3.

Allison, P. D. (2014). *Event history and survival analysis: Regression for longitudinal event data*, volume 46. SAGE publications.

Alrawashdeh, M., Klompas, M., Simpson, S. Q., Kadri, S. S., Poland, R., Guy, J. S., Perlin, J. B., Rhee, C., Program, C. P. E., et al. (2022). Prevalence and outcomes of previously healthy adults among patients hospitalized with community-onset sepsis. *Chest*.

Amaya, D. J. (2019). The Pacific Meridional Mode and ENSO: a Review. *Current Climate Change Reports*, pages 1–13.

Anand, R. S., Stey, P., Jain, S., Biron, D. R., Bhatt, H., Monteiro, K., Feller, E., Ranney, M. L., Sarkar, I. N., and Chen, E. S. (2018). Predicting mortality in diabetic icu patients using machine learning and severity indices. *AMIA Summits on Translational Science Proceedings*, 2018:310.

Arfken, G. B. and Weber, H. J. (1999). Mathematical methods for physicists.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115.

Balmaseda, M. A., Davey, M. K., and Anderson, D. L. T. (1995). Decadal and Seasonal Dependence of ENSO Prediction Skill. *J. Climate*, 8(11):2705–2715.

Bandt, C. and Pompe, B. (2002). Permutation entropy: a natural complexity measure for time series. *Physical review letters*, 88(17):174102.

Barnston, A. G., Tippett, M. K., L'Heureux, M. L., Li, S., and DeWitt, D. G. (2012a). Skill of Real-Time Seasonal ENSO Model Predictions during 2002–11: Is Our Capability Increasing? *Bull. Amer. Meteor. Soc.*, 93(5):631–651.

Barnston, A. G., Tippett, M. K., L'Heureux, M. L., Li, S., and DeWitt, D. G. (2012b). Skill of Real-Time Seasonal ENSO Model Predictions during 2002–11: Is Our Capability Increasing? *Bull. Amer. Meteor. Soc.*, 93(5):631–651.

Batista, G. E., Keogh, E. J., Tataw, O. M., and de Souza, V. (2014). Cid: an efficient complexity-invariant distance for time series. *Data Mining and Knowledge Discovery*, 28(3):634–669.

Beyersmann, J., Allignol, A., and Schumacher, M. (2011). *Competing risks and multistate models with R*. Springer Science & Business Media.

Bishop, C. M. (1994). Mixture density networks.

Bishop, C. M. (2009). *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, NY, corrected at 8th printing 2009 edition. OCLC: 845772798.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

Black, S., Kushner, I., and Samols, D. (2004). C-reactive protein. *Journal of Biological Chemistry*, 279(47):48487–48490.

Borovykh, A., Bohte, S., and Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*.

Boullé, N., Earls, C. J., and Townsend, A. (2022). Data-driven discovery of green' s functions with human-understandable deep learning. *Scientific reports*, 12(1):1–9.

Boullé, N., Nakatsukasa, Y., and Townsend, A. (2020). Rational neural networks. *Advances in Neural Information Processing Systems*, 33:14243–14253.

Brand, L., Patel, A., Singh, I., and Brand, C. (2018). Real time mortality risk prediction: A convolutional neural network approach. In *HEALTHINF*, pages 463–470.

Braun, A., Nordlund, D., Song, S.-W., Huang, T.-W., Sokaras, D., Liu, X., Yang, W., Weng, T.-C., and Liu, Z. (2015). Hard x-rays in–soft x-rays out: An operando piggyback view deep into a charging lithium ion battery with x-ray raman spectroscopy. *Journal of Electron Spectroscopy and Related Phenomena*, 200:257–263.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Buchanan, M. (2015). Depths of learning. *Nature Physics*, 11(10):798–798.

Butterworth, S. et al. (1930). On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541.

Caianiello, E. R. (1961). Outline of a theory of thought-processes and thinking machines. *Journal of theoretical biology*, 1(2):204–235.

Castelvecchi, D. (2016). Can we open the black box of ai? *Nature News*, 538(7623):20.

Chakraborty, R. K. and Burns, B. (2019). Systemic inflammatory response syndrome.

Chang, Y.-J., Yeh, M.-L., Li, Y.-C., Hsu, C.-Y., Lin, C.-C., Hsu, M.-S., and Chiu, W.-T. (2011). Predicting hospital-acquired infections by scoring system with simple parameters. *PloS one*, 6(8):e23137.

Chen, D. and Cane, M. A. (2008). El Niño prediction and predictability. *Journal of Computational Physics*, 227(7):3625–3640.

Chen, Y. and Qi, B. (2019). Representation learning in intraoperative vital signs for heart failure risk prediction. *BMC medical informatics and decision making*, 19(1):1–15.

Chimmula, V. K. R. and Zhang, L. (2020). Time series forecasting of covid-19 transmission in canada using lstm networks. *Chaos, Solitons & Fractals*, 135:109864.

Choi, S. C. and Wette, R. (1969). Maximum likelihood estimation of the parameters of the gamma distribution and their bias. *Technometrics*, 11(4):683–690.

Christodoulou, E., Ma, J., Collins, G. S., Steyerberg, E. W., Verbakel, J. Y., and Van Calster, B. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *Journal of clinical epidemiology*, 110:12–22.

Ciarrocchi, N., Quiróz, N., Traversaro, F., San Roman, E., Risk, M., Goldemberg, F., and Redelico, F. O. (2019). The complexity of intracranial pressure as an indicator of cerebral autoregulation. *Communications in Nonlinear Science and Numerical Simulation*, 75:192–199.

Cimpoi, M., Maji, S., and Vedaldi, A. (2014). Deep convolutional filter banks for texture recognition and segmentation. *arXiv preprint arXiv:1411.6836*.

Comstedt, P., Storgaard, M., and Lassen, A. T. (2009). The systemic inflammatory response syndrome (sirs) in acutely hospitalised medical patients: a cohort study. *Scandinavian journal of trauma, resuscitation and emergency medicine*, 17(1):1–6.

Cortese, G. and Andersen, P. K. (2010). Competing risks and time-dependent covariates. *Biometrical Journal*, 52(1):138–158.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202.

Cox, D. R. (1975). Partial likelihood. *Biometrika*, 62(2):269–276.

Cui, N., Zhang, H., Chen, Z., and Yu, Z. (2019). Prognostic significance of pct and crp evaluation for adult icu patients with sepsis and septic shock: retrospective analysis of 59 cases. *Journal of International Medical Research*, 47(4):1573–1579.

Dantes, R. B. and Epstein, L. (2018). Combatting sepsis: a public health perspective. *Clinical infectious diseases*, 67(8):1300–1302.

Dean, J. (2022). A golden decade of deep learning: Computing systems & applications. *Daedalus*, 151(2):58–74.

Desai, R. J., Wang, S. V., Vaduganathan, M., Evers, T., and Schneeweiss, S. (2020). Comparison of machine learning methods with traditional models for use of administrative claims with electronic medical records to predict heart failure outcomes. *JAMA network open*, 3(1):e1918962–e1918962.

Dijkstra, H. A. (2005). Nonlinear physical oceanography: a dynamical systems approach to the large scale ocean circulation and el nino.

Dijkstra, H. A., Petersik, P., Hernández-García, E., and López, C. (2019). The application of machine learning techniques to improve el niño prediction skill. *Frontiers in Physics*, page 153.

Dimitri, A. and Talamo, M. (2018). The use of data mining and machine learning in nanomedicine: A survey. *Frontiers in Nanoscience and Nanotechnoogy*, 4.

Dorraki, M., Fouladzadeh, A., Salamon, S. J., Allison, A., Coventry, B. J., and Abbott, D. (2019). Can c-reactive protein (crp) time series forecasting be achieved via deep learning? *IEEE Access*, 7:59311–59320.

Durastanti, C., Cirillo, E. N., De Benedictis, I., Ledda, M., Sciortino, A., Lisi, A., Convertino, A., and Mussi, V. (2022). Statistical classification for raman spectra of tumoral genomic dna. *arXiv preprint arXiv:2203.10867*.

Engoren, M. (1998). Approximate entropy of respiratory rate and tidal volume during weaning from mechanical ventilation. *Critical care medicine*, 26(11):1817–1823.

Evans, L. C. (2010). *Partial differential equations*, volume 19. American Mathematical Soc.

Fedorov, A., Harper, S., Philander, S., Winter, B., and Wittenberg, A. (2003). How predictable is El Niño? *Bulletin of the American Meteorological Society*, 84(7):911–919.

Feng, Q. Y. and Dijkstra, H. A. (2017). Climate network stability measures of El Niño variability. *Chaos*, 27(3):035801–15.

Feretzakis, G., Loupelis, E., Sakagianni, A., Kalles, D., Martsoukou, M., Lada, M., Skarmoutsou, N., Christopoulos, C., Valakis, K., Velentza, A., et al. (2020). Using machine learning techniques to aid empirical antibiotic therapy decisions in the intensive care unit of a general hospital in greece. *Antibiotics*, 9(2):50.

Fleischmann, M., Hendra, P. J., and McQuillan, A. J. (1974). Raman spectra of pyridine adsorbed at a silver electrode. *Chemical physics letters*, 26(2):163–166.

Fleuren, L. M., Klausch, T. L., Zwager, C. L., Schoonmade, L. J., Guo, T., Roggeveen, L. F., Swart, E. L., Girbes, A. R., Thoral, P., Ercole, A., et al. (2020). Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy. *Intensive care medicine*, 46(3):383–400.

Flynn, R. (2012). Survival analysis. *Journal of Clinical Nursing*, 21(19pt20):2789–2797.

Frasca, M. (2008). Green functions and nonlinear systems: Short time expansion. *International Journal of Modern Physics A*, 23(02):299–308.

Frasca, M. and Khurshudyan, A. (2018). General representation of nonlinear green's function for second order differential equations nonlinear in the first derivative. *arXiv preprint arXiv:1806.00274*.

Furgal, A. K., Sen, A., and Taylor, J. M. (2019). Review and comparison of computational approaches for joint longitudinal and time-to-event models. *International Statistical Review*, 87(2):393–418.

Furtak, T. (1983). Current understanding of the mechanism of surface enhanced raman scattering. *Journal of Electroanalytical Chemistry and Interfacial Electrochemistry*, 150(1-2):375–388.

Geng, P., Qin, W., and Xu, G. (2021). Proline metabolism in cancer. *Amino Acids*, pages 1–9.

Gill, A. (1980). Some simple solutions for heat-induced tropical circulation. *Quart. J. Roy Meteor. Soc.,*, 106:447–462.

Gin, C. R., Shea, D. E., Brunton, S. L., and Kutz, J. N. (2021). Deepgreen: Deep learning of green' s functions for nonlinear boundary value problems. *Scientific reports*, 11(1):1–14.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Guo-yan, X., Jin, Z., Cun-you, S., Wen-bin, H., and Fan, L. (2019). Combined hydrological time series forecasting model based on cnn and mc. *Computer and Modernization*, (11):23.

Hafen, B. B. and Sharma, S. (2018). Oxygen saturation.

Ham, Y.-G., Kim, J.-H., and Luo, J.-J. (2019a). Deep learning for multi-year enso forecasts. *Nature*, 573(7775):568–572.

Ham, Y.-G., Kim, J.-H., and Luo, J.-J. (2019b). Deep learning for multi-year ENSO forecasts. *Nature Publishing Group*, pages 1–17.

Ham, Y.-G., Kug, J.-S., Park, J.-Y., and Jin, F.-F. (2013). Sea surface temperature in the north tropical Atlantic as a trigger for El Niño/Southern Oscillation events. *Nature Geosci*, 6(2):112–116.

Hamming, R. W. (1998). *Digital filters*. Courier Corporation.

Hassabis, D., Kumaran, D., Summerfield, C., and Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory*. Wiley.

Henry, K. E., Hager, D. N., Pronovost, P. J., and Saria, S. (2015). A targeted real-time early warning score (trewscore) for septic shock. *Science translational medicine*, 7(299):299ra122–299ra122.

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Hodges, J. L. (1958). The significance probability of the smirnov two-sample test. *Arkiv för Matematik*, 3(5):469–486.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.

Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574.

Hyndman, R. J. and Fan, Y. (1996). Sample quantiles in statistical packages. *The American Statistician*, 50(4):361–365.

Ibrahim, J. G., Chu, H., and Chen, L. M. (2010). Basic concepts and methods for joint models of longitudinal and survival data. *Journal of Clinical Oncology*, 28(16):2796.

Irie, K., Tüske, Z., Alkhouli, T., Schlüter, R., Ney, H., et al. (2016). Lstm, gru, highway and a bit of attention: an empirical overview for language modeling in speech recognition. In *Interspeech*, pages 3519–3523.

Ivanov, O., Molander, K., Dunne, R., Liu, S., Masek, K., Lewis, E., Wolf, L., Travers, D., Brecher, D., Delaney, D., et al. (2022). Accurate detection of sepsis at ed triage using machine learning with clinical natural language processing. *arXiv preprint arXiv:2204.07657*.

Izumo, T., Vialard, J., Lengaigne, M., de Boyer Montegut, C., Behera, S. K., Luo, J.-J., Cravatte, S., Masson, S., and Yamagata, T. (2010). Influence of the state of the Indian Ocean Dipole on the following year's El Niño. *Nature Geosci*, 3(3):168–172.

Janiesch, C., Zschech, P., and Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3):685–695.

Jin, F.-F. (1997). An equatorial recharge paradigm for ENSO. I: Conceptual Model. *J. Atmos. Sci.*, 54:811–829.

Jin, F.-F., Neelin, J., and Ghil, M. (1994). El Niño on the devil's staircase: Annual subharmonic steps to chaos. *Science*, 264:70–72.

Jolliffe, I. T. and Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202.

Kagaya, H., Aizawa, K., and Ogawa, M. (2014). Food detection and recognition using convolutional neural network. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1085–1088.

Kalpakis, K., Yang, S., Hu, P. F., Mackenzie, C. F., Stansbury, L. G., Stein, D. M., and Scalea, T. M. (2015). Permutation entropy analysis of vital signs data for outcome prediction of patients with severe traumatic brain injury. *Computers in biology and medicine*, 56:167–174.

Khera, R., Haimovich, J., Hurley, N. C., McNamara, R., Spertus, J. A., Desai, N., Rumsfeld, J. S., Masoudi, F. A., Huang, C., Normand, S.-L., et al. (2021). Use of machine learning models to predict death after acute myocardial infarction. *JAMA cardiology*, 6(6):633–641.

Khurshudyan, A. Z. (2018a). New green' s functions for some nonlinear oscillating systems and related pdes. *International Journal of Modern Physics C*, 29(04):1850032.

Khurshudyan, A. Z. (2018b). Nonlinear green' s functions for wave equation with quadratic and hyperbolic potentials. *Advances in Mathematical Physics*, 2018.

Khurshudyan, A. Z. (2018c). Nonlinear implicit green' s functions for numerical approximation of partial differential equations: Generalized burgers' equation and nonlinear wave equation with damping. *International Journal of Modern Physics C*, 29(07):1850054.

Kingma, D. P. and Ba, J. (2014a). Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*. arXiv: 1412.6980.

Kingma, D. P. and Ba, J. (2014b). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klouwenberg, P. M. K., Ong, D. S., Bos, L. D., de Beer, F. M., van Hooijdonk, R. T., Huson, M. A., Straat, M., van Vught, L. A., Wieske, L., Horn, J., et al. (2013). Interobserver agreement of centers for disease control and prevention criteria for classifying infections in critically ill patients. *Critical care medicine*, 41(10):2373–2378.

Kolmogorov, A. N. and Fomin, S. V. (1957). *Elements of the theory of functions and functional analysis. Volume 1: Metric and normed spaces*. Graylock Press Rochester.

Komorowski, M. (2020). Clinical management of sepsis can be improved by artificial intelligence: yes.

Kong, K., Kendall, C., Stone, N., and Notingher, I. (2015). Raman spectroscopy for medical diagnostics—from in-vitro biofluid assays to in-vivo cancer detection. *Advanced drug delivery reviews*, 89:121–134.

Kwon, D., Natarajan, K., Suh, S. C., Kim, H., and Kim, J. (2018). An empirical study on network anomaly detection using convolutional neural networks. In *ICDCS*, pages 1595–1598.

Kyriakidou, M., Anastassopoulou, J., Tsakiris, A., Koui, M., and Theophanides, T. (2017). Ft-ir spectroscopy study in early diagnosis of skin cancer. *in vivo*, 31(6):1131–1137.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 6402–6413. Curran Associates, Inc.

Landau, L. D. and Lifshitz, E. M. (2013). *Quantum mechanics: non-relativistic theory*, volume 3. Elsevier.

Latif, M., Anderson, D., Barnett, T., Cane, M., Kleeman, R., Leetmaa, A., O'Brien, J. J., Rosati, A., and Schneider, E. (1998). A review of the predictability and prediction of ENSO. *Journal of Geophysical Research*.

Latif, M. and Barnett, T. P. (1994). Causes of decadal climate variability over the North Pacific and North America. *Science*, 266:634–637.

L'Heureux, M. L., Takahashi, K., Watkins, A. B., Barnston, A. G., Becker, E. J., Di Liberto, T. E., Gamble, F., Gottschalck, J., Halpert, M. S., Huang, B., Mosquera-Vásquez, K., and Wittenberg, A. T. (2017). Observing and Predicting the 2015/16 El Niño. *Bull. Amer. Meteor. Soc.*, 98(7):1363–1382.

Li, J., Mohamed, A., Zweig, G., and Gong, Y. (2015). Lstm time and frequency recurrence for automatic speech recognition. In *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*, pages 187–191. IEEE.

Lian, T., Chen, D., Tang, Y., and Wu, Q. (2014). Effects of westerly wind bursts on El Niño: A new perspective. *Geophys. Res. Lett.*, 41(10):3522–3527.

Lin, D., Feng, S., Pan, J., Chen, Y., Lin, J., Chen, G., Xie, S., Zeng, H., and Chen, R. (2011). Colorectal cancer detection by gold nanoparticle based surface-enhanced raman spectroscopy of blood serum and statistical analysis. *Optics express*, 19(14):13565–13577.

Liu, Y. H. (2018). Feature extraction and image recognition with convolutional neural networks. In *Journal of Physics: Conference Series*, volume 1087, page 062032. IOP Publishing.

Liu, Z., Parida, S., Prasad, R., Pandey, R., Sharma, D., and Barman, I. (2021). Vibrational spectroscopy for decoding cancer microbiota interactions: Current evidence and future perspective. In *Seminars in Cancer Biology*. Elsevier.

Livieris, I. E., Pintelas, E., and Pintelas, P. (2020). A cnn–lstm model for gold price time-series forecasting. *Neural computing and applications*, 32(23):17351–17360.

Lou, G. and Shi, H. (2020). Face image recognition based on convolutional neural network. *China Communications*, 17(2):117–124.

Maki, D. G., Crnich, C. J., and Safdar, N. (2008). Nosocomial infection in the intensive care unit. *Critical care medicine*, page 1003.

Mann, H. B. and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

McPhaden, M. J. (2003). Tropical Pacific Ocean heat content variations and ENSO persistence barriers. *Geophysical Research Letters*, 30(9):2705–2709.

McPhaden, M. J., Timmermann, A., Widlansky, M. J., Balmaseda, M. A., and Stockdale, T. N. (2015). The curious case of the El Niño that never happened: A perspective from 40 years of progress in climate research and forecasting. *Bull. Amer. Meteor. Soc.*, 96(10):1647–1665.

Milani, A., Tommasini, M., Russo, V., Bassi, A. L., Lucotti, A., Cataldo, F., and Casari, C. S. (2015). Raman spectroscopy as a tool to investigate the structure and electronic properties of carbon-atom wires. *Beilstein journal of nanotechnology*, 6(1):480–491.

Montavon, G., Binder, A., Lapuschkin, S., Samek, W., and Müller, K.-R. (2019). Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209.

Movasaghi, Z., Rehman, S., and Rehman, I. U. (2007). Raman spectroscopy of biological tissues. *Applied Spectroscopy Reviews*, 42(5):493–541.

Mundhenk, T. N., Chen, B. Y., and Friedland, G. (2019). Efficient saliency maps for explainable ai. *arXiv preprint arXiv:1911.11293*.

Musoro, J., Struijk, G., Geskus, R., Ten Berge, I., and Zwinderman, A. (2018). Dynamic prediction of recurrent events data by landmarking with application to a follow-up study of patients after kidney transplant. *Statistical methods in medical research*, 27(3):832–845.

Mussi, V., Ledda, M., Polese, D., Maiolo, L., Paria, D., Barman, I., Lolli, M. G., Lisi, A., and Convertino, A. (2021). Silver-coated silicon nanowire platform discriminates genomic dna from normal and malignant human epithelial cells using label-free raman spectroscopy. *Materials Science and Engineering: C*, 122:111951.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., and Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of big data*, 2(1):1–21.

Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M., and Han, K. (2018). Enhanced network anomaly detection based on deep neural networks. *IEEE access*, 6:48231–48246.

Neelin, J. D., Battisti, D. S., Hirst, A. C., Jin, F.-F., Wakata, Y., Yamagata, T., and Zebiak, S. (1998). ENSO theory. *Journal of Geophysical Research*, 103(C7):14261–14290.

Nehring, S. M., Goyal, A., Bansal, P., and Patel, B. C. (2017). C reactive protein.

Nicolaie, M., Van Houwelingen, J., De Witte, T., and Putter, H. (2013). Dynamic prediction by landmarking in competing risks. *Statistics in medicine*, 32(12):2031–2047.

Nicolaie, M. A. (2014). *Dynamic aspects of competing risks with application to medical data*. PhD thesis, Universiteit Leiden.

Ohno-Machado, L. (2001). Modeling medical prognosis: survival analysis techniques. *Journal of biomedical informatics*, 34(6):428–439.

Paparrizos, J. and Gravano, L. (2015). k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1855–1870.

Pappalettera, C., Miraglia, F., Cotelli, M., Rossini, P. M., and Vecchio, F. (2022). Analysis of complexity in the eeg activity of parkinson ' s disease patients by means of approximate entropy. *GeroScience*, pages 1–9.

Pedlosky, J. et al. (1987). *Geophysical fluid dynamics*, volume 710. Springer.

Petersik, P. J. and Dijkstra, H. A. (2020). Probabilistic forecasting of el niño using neural network models. *Geophysical Research Letters*, 47(6):e2019GL086423.

Petry, R., Schmitt, M., and Popp, J. (2003). Raman spectroscopy—a prospective tool in the life sciences. *ChemPhysChem*, 4(1):14–30.

Pettit, R. W., Fullem, R., Cheng, C., and Amos, C. I. (2021). Artificial intelligence, machine learning, and deep learning for clinical outcome prediction. *Emerging topics in life sciences*, 5(6):729–745.

Phillips, P. J., Hahn, C. A., Fontana, P. C., Broniatowski, D. A., and Przybocki, M. A. (2020). Four principles of explainable artificial intelligence. *Gaithersburg, Maryland*.

Pincus, S. M. (1991). Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88(6):2297–2301.

Pincus, S. M., Gladstone, I. M., and Ehrenkranz, R. A. (1991). A regularity statistic for medical data analysis. *Journal of clinical monitoring*, 7(4):335–345.

Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Povoa, P., Almeida, E., Moreira, P., Fernandes, A., Mealha, R., Aragao, A., and Sabino, H. (1998). C-reactive protein as an indicator of sepsis. *Intensive care medicine*, 24(10):1052–1056.

Póvoa, P., Coelho, L., Almeida, E., Fernandes, A., Mealha, R., Moreira, P., and Sabino, H. (2006). Early identification of intensive care unit-acquired infections with daily monitoring of c-reactive protein: a prospective observational study. *Critical Care*, 10(2):1–8.

Pratt, G. A. (2015). Is a cambrian explosion coming for robotics? *Journal of Economic Perspectives*, 29(3):51–60.

Preisendorfer, R. W. (1988). *Principal Component Analysis in Meteorology and Oceanography*. Elsevier, Amsterdam, The Netherlands.

Prentice, R. L., Kalbfleisch, J. D., Peterson Jr, A. V., Flournoy, N., Farewell, V. T., and Breslow, N. E. (1978). The analysis of failure times in the presence of competing risks. *Biometrics*, pages 541–554.

Proust-Lima, C. and Taylor, J. M. (2009). Development and validation of a dynamic prognostic tool for prostate cancer recurrence using repeated measures of posttreatment psa: a joint modeling approach. *Biostatistics*, 10(3):535–549.

Putter, H., Fiocco, M., and Geskus, R. B. (2007). Tutorial in biostatistics: competing risks and multi-state models. *Statistics in medicine*, 26(11):2389–2430.

Putter, H. and van Houwelingen, H. C. (2017). Understanding landmarking and its relation with time-dependent cox regression. *Statistics in biosciences*, 9(2):489–503.

Putter, H. and van Houwelingen, H. C. (2022). Landmarking 2.0: Bridging the gap between joint models and landmarking. *Statistics in Medicine*, 41(11):1901–1917.

Reny, J.-L., Vuagnat, A., Ract, C., Benoit, M.-O., Safar, M., and Fagon, J.-Y. (2002). Diagnosis and follow-up of infections in intensive care patients: value of c-reactive protein compared with other clinical and biological variables. *Critical care medicine*, 30(3):529–535.

Rigas, B., Morgello, S., Goldman, I. S., and Wong, P. (1990). Human colorectal cancers display abnormal fourier-transform infrared spectra. *Proceedings of the National Academy of Sciences*, 87(20):8140–8144.

Rioul, O. and Vetterli, M. (1991). Wavelets and signal processing. *IEEE signal processing magazine*, 8(4):14–38.

Riser, A. P. and Cassarly, W. J. (2001). Analysis of single lens arrays using convolution. *Optical Engineering*, 40(5):805–813.

Rizopoulos, D. (2011). Dynamic predictions and prospective accuracy in joint models for longitudinal and time-to-event data. *Biometrics*, 67(3):819–829.

Rizopoulos, D. (2012). *Joint models for longitudinal and time-to-event data: With applications in R*. CRC press.

Roddier, F. (1999). Adaptive optics in astronomy.

Roimi, M., Neuberger, A., Shrot, A., Paul, M., Geffen, Y., and Bar-Lavie, Y. (2020). Early diagnosis of bloodstream infections in the intensive care unit using machine-learning algorithms. *Intensive Care Medicine*, 46(3):454–462.

Rosenblatt, F. (1960). Perceptron simulation experiments. *Proceedings of the IRE*, 48(3):301–309.

Rosenblatt, F. (1961). Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Sagheer, A. and Kotb, M. (2019). Time series forecasting of petroleum production using deep lstm recurrent networks. *Neurocomputing*, 323:203–213.

Saha, S., Moorthi, S., Wu, X., Wang, J., Nadiga, S., Tripp, P., Behringer, D., Hou, Y.-T., Chuang, H.-y., Iredell, M., et al. (2014). The NCEP climate forecast system version 2. *Journal of Climate*, 27(6):2185–2208.

Sainath, T. N., Kingsbury, B., Mohamed, A.-r., and Ramabhadran, B. (2013). Learning filter banks within a deep neural network framework. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 297–302. IEEE.

Salinas-Escudero, G., Carrillo-Vega, M. F., Granados-García, V., Martínez-Valverde, S., Toledano-Toledano, F., and Garduño-Espinosa, J. (2020). A survival analysis of covid-19 in the mexican population. *BMC public health*, 20(1):1–8.

Sarker, I. H. (2021). Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6):1–20.

Savitzky, A. and Golay, M. J. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639.

Schlapbach, L. J., Kissoon, N., Alhawsawi, A., Aljuaid, M. H., Daniels, R., Gorordo-Delsol, L. A., Machado, F., Malik, I., Nsutebu, E. F., Finfer, S., et al. (2020). World sepsis day: a global agenda to target a leading cause of morbidity and mortality.

Schmid, T. and Dariz, P. (2019). Raman microspectroscopic imaging of binder remnants in historical mortars reveals processing conditions. *Heritage*, 2(2):1662–1683.

Schreiber, T. and Schmitz, A. (1997). Discrimination power of measures for nonlinearity in a time series. *Physical Review E*, 55(5):5443.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V. K., and Soman, K. (2017). Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*, pages 1643–1647. IEEE.

Serov, N. and Vinogradov, V. (2022). Artificial intelligence to bring nanomedicine to life. *Advanced Drug Delivery Reviews*, page 114194.

Sezer, O. B., Gudelek, M. U., and Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90:106181.

Shen, X., Tian, X., Liu, T., Xu, F., and Tao, D. (2017). Continuous dropout. *IEEE transactions on neural networks and learning systems*, 29(9):3926–3937.

Siami-Namini, S., Tavakoli, N., and Namin, A. S. (2018). A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 1394–1401. IEEE.

Silverman, R. A. et al. (1972). *Special functions and their applications*. Courier Corporation.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Singer, M., Deutschman, C. S., Seymour, C. W., Shankar-Hari, M., Annane, D., Bauer, M., Bellomo, R., Bernard, G. R., Chiche, J.-D., Coopersmith, C. M., et al. (2016). The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, 315(8):801–810.

Solanki, C., Thapliyal, P., and Tomar, K. (2014). Role of bisection method. *International Journal of Computer Applications Technology and Research*, 3(8):535–535.

Spitoni, C., Lammens, V., and Putter, H. (2018). Prediction errors for state occupation and transition probabilities in multi-state models. *Biometrical Journal*, 60(1):34–48.

Srimuninnimit, V., Ariyapanya, S., Nimmannit, A., Wonglaksanapimon, S., Akewanlop, C., and Soparattanapaisarn, N. (2012). C-reactive protein as a monitor of chemotherapy response in advanced non-small cell lung cancer (cml study). *J Med Assoc Thai*, 95(Suppl 2):S199–S207.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Staar, B., Lütjen, M., and Freitag, M. (2019). Anomaly detection with convolutional neural networks for industrial surface inspection. *Procedia CIRP*, 79:484–489.

Stiles, P. L., Dieringer, J. A., Shah, N. C., and Van Duyne, R. P. (2008). Surface-enhanced raman spectroscopy. *Annu. Rev. Anal. Chem.*, 1:601–626.

Suarez, M. and Schopf, P. S. (1988). A delayed action oscillator for ENSO. *J. Atmos. Sci.*, 45:3283–3287.

Suresh, H., Hunt, N., Johnson, A., Celi, L. A., Szolovits, P., and Ghassemi, M. (2017). Clinical intervention prediction and understanding using deep networks. *arXiv preprint arXiv:1705.08498*.

Talari, A. C. S., Movasaghi, Z., Rehman, S., and Rehman, I. U. (2015). Raman spectroscopy of biological tissues. *Applied spectroscopy reviews*, 50(1):46–111.

Tang, Y., Zhang, R.-H., Liu, T., Duan, W., Yang, D., Zheng, F., Ren, H., Lian, T., Gao, C., Chen, D., and Mu, M. (2018). Progress in ENSO prediction and predictability study. *National Science Review*, 5(6):826–839.

Tanner, K. T., Sharples, L. D., Daniel, R. M., and Keogh, R. H. (2021). Dynamic survival prediction combining landmarking with a machine learning ensemble: Methodology and empirical comparison. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 184(1):3–30.

Timmermann, A., An, S.-I., Kug, J.-S., Jin, F.-F., Cai, W., Capotondi, A., Cobb, K., Lengaigne, M., McPhaden, M. J., Stuecker, M. F., Stein, K., Wittenberg, A. T., Yun, K.-S., Bayr, T., Chen, H.-C., Chikamoto, Y., Dewitte, B., Dommenget, D., Grothe, P., Guilyardi, E., Ham, Y.-G., Hayashi, M., Ineson, S., Kang, D., Kim, S., Kim, W., Lee, J.-Y., Li, T., Luo, J.-J., McGregor, S., Planton, Y., Power, S., Rashid, H., Ren, H.-L., Santoso, A., Takahashi, K., Todd, A., Wang, G., Wang, G., Xie, R., Yang, W.-H., Yeh, S.-W., Yoon, J., Zeller, E., and Zhang, X. (2018). El Niño–Southern Oscillation complexity. *Nature*, pages 1–11.

Tonekaboni, S., Mazwi, M., Laussen, P., Eytan, D., Greer, R., Goodfellow, S. D., Goodwin, A., Brudno, M., and Goldenberg, A. (2018). Prediction of cardiac arrest from physiological signals in the pediatric icu. In *Machine Learning for Healthcare Conference*, pages 534–550. PMLR.

Topol, E. J. (2019). High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25(1):44–56.

Tuli, S., Dasgupta, I., Grant, E., and Griffiths, T. L. (2021). Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*.

Tziperman, E., Toggweiler, J. R., Bryan, K., and Feliks, Y. (1994). Instability of the thermohaline circulation with respect to mixed boundary conditions: Is it really a problem for realistic models? *Journal of Physical Oceanography*, 24(2):217–232.

van der Vaart, P. C. F., Dijkstra, H. a., and Jin, F. F. (2000). The Pacific Cold Tongue and the ENSO Mode: A Unified Theory within the Zebiak–Cane Model. *Journal of the Atmospheric Sciences*, 57(7):967–988.

van Houwelingen, H. and Putter, H. (2011). *Dynamic prediction in clinical survival analysis*. CRC Press.

Van Houwelingen, H. C. (2007). Dynamic prediction by landmarking in event history analysis. *Scandinavian Journal of Statistics*, 34(1):70–85.

Varela, M., Calvo, M., Chana, M., Gomez-Mestre, I., Asensio, R., and Galdos, P. (2005). Clinical implications of temperature curve complexity in critically ill patients. *Critical care medicine*, 33(12):2764–2771.

Vilone, G. and Longo, L. (2021). Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76:89–106.

Watt, D., Aitchison, T., Mackie, R., and Sirel, J. (1996). Survival analysis: the importance of censored observations. *Melanoma research*, 6(5):379–385.

Weber, C. E., Luo, C., Hotz-Wagenblatt, A., Gardyan, A., Kordaß, T., Holland-Letz, T., Osen, W., and Eichmüller, S. B. (2016). mir-339-3p is a tumor suppressor in melanomamirnas affecting melanoma cell invasion. *Cancer research*, 76(12):3562–3571.

Werbos, P. (1974). Beyond regression:" new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*.

Widjaja, E., Zheng, W., and Huang, Z. (2008). Classification of colonic tissues using near-infrared raman spectroscopy and support vector machines. *International journal of oncology*, 32(3):653–662.

Wikipedia (2022a). Raman spectroscopy. `https://en.wikipedia.org/wiki/Raman_spectroscopy`.

Wikipedia (2022b). Recurrent neural network. `https://en.wikipedia.org/wiki/Recurrent_neural_network`.

Yan, J., Mu, L., Wang, L., Ranjan, R., and Zomaya, A. Y. (2020). Temporal convolutional networks for the advance prediction of enso. *Scientific reports*, 10(1):1–15.

Ye, Y., Jiang, J., Ge, B., Dou, Y., and Yang, K. (2019). Similarity measures for time series data classification using grid representation and matrix distance. *Knowledge and Information Systems*, 60(2):1105–1134.

Yentis, S., Soni, N., and Sheldon, J. (1995). C-reactive protein as an indicator of resolution of sepsis in the intensive care unit. *Intensive care medicine*, 21(7):602–605.

Young, A. T. (1981). Rayleigh scattering. *Applied optics*, 20(4):533–535.

Ypma, T. J. (1995). Historical development of the newton–raphson method. *SIAM review*, 37(4):531–551.

Zebiak, S. E. and Cane, M. A. (1987a). A model El Niño-Southern Oscillation. *Monthly Weather Review*, 115:2262–2278.

Zebiak, S. E. and Cane, M. A. (1987b). A Model El Niñ–Southern Oscillation. *Mon. Wea. Rev.*, 115(10):2262–2278.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Zeiler, M. D., Krishnan, D., Taylor, G. W., and Fergus, R. (2010). Deconvolutional networks. In *2010 IEEE Computer Society Conference on computer vision and pattern recognition*, pages 2528–2535. IEEE.

Zeng, Z., Hou, Z., Li, T., Deng, L., Hou, J., Huang, X., Li, J., Sun, M., Wang, Y., Wu, Q., et al. (2022). A deep learning approach to predicting ventilator parameters for mechanically ventilated septic patients. *arXiv preprint arXiv:2202.10921*.

Zhang, L., Tan, J., Han, D., and Zhu, H. (2017). From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug discovery today*, 22(11):1680–1685.

Zhang, Z., Ren, B., and Zheng, J. (2019). A unified complex index to characterize two types of ENSO simultaneously. *Scientific Reports*, pages 1–8.

Zhao, Y. and Di Lorenzo, E. (2020). The impacts of Extra-tropical ENSO Precursors on Tropical Pacific Decadal-scale Variability. *Scientific Reports*, pages 1–12.

Zheng, H., Fu, J., Mei, T., and Luo, J. (2017). Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 5209–5217.

Acknowledgments

> Bravo! Me dispiace che il Signor Presidente se n'è andato, ma mò che torna sono sicuro d'avere una sentenza d'assoluzione e di perdono per tutti quanti. Mò 'nce ne jammo a mangià tutte quante 'ncoppa a dù Pallino. E…stanotte chissà ca succede pure 'o fatto apposta! Mò ca tutto s'è sistemato, spero con un lungo e rumoroso applauso, di avere l'assoluzione anche da questo nobile e rispettabile pubblico!
>
> E. Scarpetta. *'O scarfalietto*

Most of the work done in this thesis is undoubtedly the result of continuous discussions and speculations with my supervisor, Cristian. I appreciated your way of doing science and working together; during our weekly lunch meetings, there was neither too much statistic nor too much computer science, but just enough physics. Thanks for supporting me during these years and giving me the certainty that I'd have reached the goal sooner or later. You've taught me a lot.

I also feel grateful to all I collaborated with: Emilio, Valentina, Annalisa, Claudio, Henk, Ivo, Olaf, and Meri. Thanks to their essential contribution, I practiced and improved my skills in analyzing large amounts of complex data.

The Mathematical Institute has been the ideal place where I had the opportunity to get acquainted with several exciting and dynamic personalities: I'm thankful to my colleagues Jetze, Sergej, Jack, Francesco, Luca, Leandro, Antonella, Stefano, Yuqin, Leonardo, Sebastián, and Pu.

Special thanks are to Matteo; we shared a lot of coffee breaks inside the Institute and tasteless pizzas outside the Institute. I'm grateful to you for being one of the few sincere friends I've fraternized with so far.

I owe a lot to another colleague, Xin; I spent the most beautiful moments of this PhD journey with you. I found in you a very happy, sensitive, and spirited friend.

My family has always been supporting me in every moment of my life. A thousand thanks to my brothers Carlo and Michele: you are the best older brothers one would have ever had; thanks for constantly encouraging me and trusting in my abilities. I'm very grateful to both my sisters-in-law, Melania and Kama: thank you for the affection you have always given me. A super special thank goes to my little nephew Riccardo: you let me realize that life is not working more than expected,

but instead, paying attention and appreciating the little things. Also, I'm very grateful to my grandpa, Vito. I always appreciated his interest in listening to my research progress; he had always instilled in me a significant curiosity about scientific subjects. I will never stop being grateful to my parents: thank you for letting me free to make all my decisions; I know it's been hard to accept that all your sons were many kilometers away. We have lost many happy days and will catch them soon.

My long-life friends, Mario, Eleonora, Martina, Chiara, and Gioia, have never stopped encouraging me; thank you both for being close to me, even at a distance.

Science and Scientific Research need a constant source of creativity. I want to express all my gratitude to all fellows of the *Korego Theater Group*. In particular, I'm grateful to Carmelinda: you taught me the motto *Io ti aiuterò* (I shall help you), and I shall keep putting it into practice every moment of my life. I say thanks to Sebastiano, Gabriele (*Quale fu la ricompensa di questo viaggio?*), Gianpaolo (*Tutto è perduto! Tutto è perduto!*), Erica (*Oronzo, quando sei partito?*), Heriana (*Un'altra volta che vengono, farò le panzarelle con la ricotta*), Ilaria (*Ottomila lire a seduta, si faceva pagare? E poi la moglie gli dava da mangiare il polmone?*), Gabriella (*Come dilaga l'immoralità!*), Elena (*penso che questo Lamberto Genova non l'ho proprio mai conosciuto!*), Margot (*Non vuole aiutarmi signor Delegato?*) Marinella (*Mi viene ancora dietro come un canuzzo*), Loredana (*Mi mandarono a chiamare il delegato Spanò*), Eros (*Spanò avete detto?*), Mirella (*Spanò creatura di tuo padre ha potuto fare questo senza sconsigliartelo?*), Natalino (*Nina, aspetta: Corda civile. Riverenza, occhi bassi e diritta a casa*), Velia (*Serva Vostra, sono!*), Federica, Jarno, Giorgio, Sandra, Valerio, Valeria, Francesca, Francesco, Chiara, Beata, Maura, Raffaele, Carla, Iridiana, and Leonardino. I spent the most exciting moments of my stay in The Netherlands with you. You all made me feel familiar, *as a piece of undeniable reality, like stones and trees*.

And, finally, Federica. Our future is just in front of us; wherever it leads, we shall figure it out together: thank you for always bearing with me and supporting me.

Curriculm vitae

Giacomo Lancia was born on September 14, 1993, in Velletri, Italy. After graduating from a scientific lyceum in 2012, he enrolled in the Physics bachelor's program at Università di Roma Tor Vergata, which he completed in October 2016. In September 2018, he completed the *Physics of complex systems and big data* master's studies at Università di Roma Tor Vergata. After that, in October 2018, he started a Ph.D. position at Utrecht Universiteit, Department of Mathematics, and Universitair Medisch Centrum Utrecht, IC department, under the guidance of Cristian Spitoni and Jason Frank. In April 2023, he started a PostDoc position at Università di Genova, Department of Mathematics.

Dertien, een getal van de nevel
het verlies van de richting, de weg
naar het verlaten gebouw
de plaats van de dans.

handen in handed, dan langdurig
zitten en wachten, wat is avond,
van wie is de kraai, van wie is de schildpad,
het vuur in de verte?

Geen antwoord is altijd een antwoord,
de karper wordt later een walvis
het kleine woordt groot
en koestert het kleine

tot de dood erop volgt.

C. Nooteboom. Monniksoog