POLITECNICO DI TORINO

Master's Degree in
Data Science and Engineering

Master Thesis

Diagnosis methods for predictive maintenance of rolling bearings in an Industry 4.0 scenario

Supervisors
Prof.ssa Tania Cerquitelli
Ilaria Bosi
Ariel Pablo Cedola
Rosaria Rossini

Candidate
Nicolò Bertozzi

Academic Year 2020-2021

Ai miei genitori,
a mio nonno Francesco,
a mia nonna Iole.

Sommario

Il cuscinetto è uno dei componenti più utilizzati in ambito industriale, poiché la sua funzione di riduzione dell'attrito è richiesta in differenti sezioni all'interno di un macchinario, sopratutto in vicinanza di elementi rotativi. Tuttavia, l'onere di essere una pedina fondamentale nell'immensa scacchiera della catena di produzione si contrappone al dovere di trovarsi sempre nelle migliori condizioni possibili, dal momento che su di esso potrebbe gravare il risultato operativo di una azienda. La rottura di uno, o più, di questi elementi potrebbe portare, infatti, a durature interruzioni della fabbricazione, generando un impatto economico negativo non indifferente per l'azienda stessa. Per di più, un singolo cuscinetto, durante la fase di fine vita, emette vibrazioni ad elevata ampiezza che potrebbero portare a ledere anche la componentistica meccanica intorno, con il rischio di creare dei malfunzionamenti a catena. Tenendo in considerazione sia la quantità di capitale necessario per l'acquisto di un macchinario di questa portata, e sia il numero di anni richiesto per un suo ammortamento a bilancio, si rende sempre più necessario mettere a punto strategie profittevoli che permettano di controllare l'andamento della degradazione di questi componenti.

Nell'ottica di garantire un buono stato di conservazione nel tempo, le strategie di riparazione di un elemento meccanico sono principalmente tre: *run-to-break*, manutenzione preventiva e predittiva. La prima prevede che l'elemento guasto venga riparato solamente in seguito alla sua rottura, comportando un certo dispendio in termini temporali ed economici. Al giorno d'oggi, tuttavia, la tecnologia permette anche di automatizzare questa strategia di intervento attraverso l'analisi *real-time* della presenza di un possibile difetto, nonostante la segnalazione tempestiva di un allarme di rottura non rappresenti un vero e proprio vantaggio. D'altro canto, il risvolto positivo consiste dapprima nella possibilità di risparmiare il tempo dedicato alla ricerca del guasto e, in seguito, nella speranza che questo si presenti inizialmente in forma più lieve, ma rilevabile, e solo successivamente in forma più grave. La seconda strategia si basa sulla manutenzione periodica e programmata del macchinario, in modo tale da anticipare la possibile insorgenza di una problematica. Da un lato, questo approccio migliora sensibilmente l'idea preistorica e infruttuosa della *run-to-break*, riuscendo ad evitare la maggior parte delle occasioni di guasto. Dall'altro, non è semplice individuare il giusto intervallo di tempo tra l'esecuzione di una manutenzione e la successiva. Inoltre, è facile che si intervenga su un componente che non necessiti di manutenzione, sprecando materiale di consumo e trascurando una parte che potrebbe invece essere già compromessa. Lo scenario appena menzionato è ancor di più valido per i cuscinetti, la cui vita stimata è affetta da una elevata varianza, che rende molto difficile

l'applicazione della strategia di manutenzione preventiva. L'ultimo approccio risiede nella manutenzione predittiva, che ha lo scopo di massimizzare la resa di un macchinario attraverso l'interruzione della produzione solamente quando è necessario. Questo è possible grazie alla possibilità di conoscere, istante per istante, lo stato di un componente, permettendo al gruppo tecnico di organizzare l'approvvigionamento del materiale di consumo solo quando necessario, e di operare con tempestività e precisione direttamente sulla zona interessata dal malfunzionamento, riducendo anche in questo caso al minimo il tempo sprecato nella ricerca del guasto. Tale strategia porterebbe allo sfruttamento totale della vita utile del cuscinetto che, nel lungo termine, garantirebbe enormi vantaggi, economici ed ecologici.

La manutenzione predittiva è una delle possibili sfaccettature dell'industria 4.0, che ha come target principale la digitalizzazione delle catene di montaggio attraverso l'installazione di una rete sensoriale che possa tenere sotto controllo l'operatività dei vari macchinari. Il budget da sfruttare per la predisposizione e l'acquisto di un numero elevato di sensori, associati ad una corrispondente cablatura, è corposo, anche in caso di trasduttori *wireless*, il cui risparmio nella mancanza della connessione fisica verrebbe comunque compensato dal loro sovraprezzo rispetto ai modelli *wired*.

Risulta quindi non facile la scelta tra un approccio più legato allo *status quo* della manutenzione rispetto ad un'orientamento completamente digitale. Per definire le migliori strategie di investimento e manutenzione viene comunque in aiuto alla classe dirigente dell'azienda la *business intelligence*, un ramo della data science che può essere di sostegno nella definizione dei processi strategici, in base alla salute attuale della compagnia, al suo storico e ai possibili scenari futuri. Vista da quest'ottica, la generazione di informazione può essere rappresentata come un complesso ciclo virtuoso di dati, nel quale le uscite di uno stato diventano poi gli ingressi di un altro, e così via finché il cerchio non si richiude. I dati, il nuovo oro nero del XXI secolo, hanno il potenziale per trasformare interi settori produttivi o, viceversa, di ancorare al passato altre realtà.

Gli scopi di questa tesi sono stati definiti per inserirsi in questo contesto di sviluppo industriale, cercando di collegare il mondo della diagnosi di un guasto al mondo della prognosi e della manutenzione predittiva. Molti dei metodi che rappresentano lo stato dell'arte per la rilevazione di una rottura all'interno di un cuscinetto possono essere trasposti nella loro versione predittiva, invece di essere visti come dei compartimenti a tenuta stagna che non possono essere impiegati in altre applicazioni. Ai suddetti metodi, comunque, vengono affiancate delle tecniche che sono proprie della prognosi, in modo tale da definire nel complesso un sistema ibrido che possa sfruttare al meglio i vantaggi delle due modalità di approccio. Il tutto, infine, viene racchiuso all'interno di un modello di *machine learning* prima, e di *deep learning* poi, che restituisca come uscita la stima della vita residua del cuscinetto o lo stato della sua salute. In conclusione, una piccola sezione verrà dedicata ad una generale analisi costi-benefici, in maniera tale da poter porre le basi di partenza di una possibile strategia di investimento da parte di un'azienda.

Summary

The rolling element bearing is one of the most used components in the industry. Due to its capacity to reduce friction, it is requested in different sections inside machinery, especially nearby rotative elements.

However, the burden of being a key piece in the huge chessboard of the production chain is opposed to the duty of finding itself in the best conditions, since that on it could weigh the operative result of a company. The failure of one, or more, of these elements could lead to long manufacture interruptions, generating a not negligible negative economic impact for the company. Moreover, a single bearing, during its end life phase, emits high magnitude vibrations, which can harm the mechanical components nearby, with the risk of creating a chain of failures. Taking into account both the amount of capital necessary to buy machinery of this scale and the number of years needed for its depreciation on the balance sheet, it is increasingly necessary to develop profitable strategies that allow to control over the time the degradation of these components.

In the view of guarantee a good conservation status as time goes on, the maintenance strategies of a mechanical component are mainly three: *run-to-break*, preventive and predictive maintenance. The first one states that the failure element is replaced only after its complete break, leading to expenditure in terms of time and business. However, nowadays, technology allows to automate this intervention strategy through the real-time analysis of the presence of a possible defect, despite the timely reporting of a breaking alarm is not a true advantage. On the flip side, the silver lining consists firstly in the chance of saving the time dedicated to the research of the failure and, secondly, in the hope that this one is initially mild, but measurable, and only in the future serious. The second strategy is based on periodic and scheduled machinery maintenance, in such a way that it is possible to anticipate the possible onset of a problem. On one hand, this approach improves significantly the prehistoric and unfruitful idea of the *run-to-break*, by avoiding the major part of failures. On the other hand, it is not trivial to define the correct time interval between ywo subsequent maintenance actions. In addition, the intervention probably regards a component that does not need service, wasting consumables and overlooking a part that can be already compromised. The aforementioned scenario is even more valid for the bearings, which life is affected by high variance that makes tricky the application of the preventive maintenance strategy. The last approach is the preventive maintenance, which aims to maximize the machinery rendition by interrupting the production only when it is necessary. This is possible due to the chance of knowing, moment by moment, the state of a component, allowing the technicians to organize the consumable supply only when it

is requested and to operate with promptness and precision directly on the area interested by the failure, minimizing the wasted time dedicated to the research of the failure. Such strategy could lead to the complete exploitation of the useful life of the bearing which, in the long term, would guarantee enormous economic and ecological advantages.

Predictive maintenance is one of the possible faces of industry 4.0, which aims to digitalize the production chain through the implementation of a sensorial network that can keep track of the efficiency of the machines. The budget, including the predisposition and the purchase of many sensors, associated to a corresponding wiring, is substantial, even in the case of wireless transducers, where the saving related to the absence of a physical connection is anyway compensated by their overpricing respect to the wired models.

The choice, between an approach linked to the *status quo* of maintenance and a completely digital orientation, is not easy. In order to define more profitable investments and more precise repairment strategies, the management of the company can rely on *business intelligence*, a branch of the data science that deals with the definition of strategic plans based on the health of the company, on its history, and on the future possible improvements. From this perspective, the generation of information could be treated as a complex virtuous cycle, where the outcomes of a state become the inputs of another one, and so forth until the circle is not closed. Data, the new black gold of the XXI century, have the potential to transform entire sectors or, vice-versa, to anchor to the past other realities.

The purposes of this thesis are defined to fit in this context of industrial development, aiming to links the diagnosis world of a failure to the prognosis and predictive maintenance one. Many of the methods which represent the state-of-the-art for the detection of a break inside a bearing can be transposed into their predictive version, instead of being seen as watertight compartments which have no other applications. Nevertheless, such methods are flanked by prognosis-based techniques, in order to define a complex hybrid system which can exploit as much as possible the advantages of these two approaches. Finally, the whole is firstly incapsulated inside a *machine learning* model and secondly inside a *deep learning* one, which returns as output the estimation of the remaining useful life of the bearing or, equivalently, its health status. In conclusion, a short section will be devoted to a general cost-benefit analysis to lay the groundwork for a possible investment strategy of a company.

Ringraziamenti

Il sentiero di istruzione che mi ha portato oggi a redigere questa tesi, nella mia mente, lo paragono a una strada di montagna, composta da curve pericolose, come i tornanti, e da rettilinei che, al contrario, rendono la guida più scorrevole e meno impegnativa. I momenti di sconforto e di difficoltà sono stati molti, come sono considerevoli anche quelli piacevoli. Tuttavia, a causa del meccanismo umano di sedimentazione dei ricordi, sono i primi che rimangono con più decisione più impressi nel tempo. Credo fermamente che senza le persone che mi sono state accanto durante questo percorso, i risultati che oggi posso vantare sarebbero ridotti al minimo.

Per questo motivo, ringrazio infinitamente i miei genitori, mio nonno Francesco, i miei nonni acquisiti Bertilla e Vasco, la mia ragazza Elena, e tutti coloro che, in un modo o nell'altro, sono stati propedeutici al raggiungimento di questo mio personale, e contemporaneamente collettivo, risultato.

In conclusione, vorrei dedicare un intero pensiero a mia nonna Iole. Nel momento in cui ho dovuto scegliere quei pochi nomi da inserire nella dedica di questo documento, sono stato molto combattuto sulla possibilità di citare solamente lei, l'unica grande assente alla cerimonia di proclamazione. Molto probabilmente sarebbe stata una scelta troppo esclusiva e, per certi versi, discriminatoria. Nonostante ciò, devo ammettere che mia nonna è stata la prima, più di un decennio fa, a vedere nel mio futuro un qualcosa che solo oggi si realizza e che, neanche io, avrei mai potuto solo immaginare. A malincuore, la persona che più ha creduto in me non ha potuto partecipare al raggiungimento di un nessun mio recente e importante traguardo. Forse ne aveva già la certezza o, alternativamente, l'ha fatto all'invisibilità dei miei occhi. In ogni caso, grazie di tutto nonna.

Nicolò

Contents

List of Tables

List of Figures

List of Acronyms

IoT Internet of Things

CBM Condition Based Maintenance

RECLAIM RE-manifaCturing and refurbishment LArge Industrial equipMent

DSF Decision Support Framework

REB Rolling Element Bearing

HI Health Index

PHI Physical HI

VHI Virtual HI

CDF Characteristic Defect Frequencies

BPFO Ballpass Frequency Outer race

BPFI Ballpass Frequency Inner race

BSF Ball Spin Frequency

MD Mahalanobis Distance

FD Frechet Distance

ED Euclidean Distance

RMS Root Mean Square

HS Health Stage

WMA Weighted Moving Average

FPT First Predicting Time

CNN Convolutional Neural Network

LSTM Long-Short Term Memory

ANN Artificial Neural Network

SVM Support Vector Machine

SVC Support Vector Classifier

RBF Radial Basis Function

EBITDA Earning Before Interests, Taxes, Depreciation and Amortization

EBITA Earning Before Interests, Taxes and Amortization

ROI Return of Interests

IRR Internal Rate of Return

KPI Key Performance Indicator

La creatività
è soprattutto la capacità
di porsi continuamente delle domande.
[Piero Angela]

Chapter 1

Industry 4.0

Italy is largely a *transformist* country. Our territory rarely relies on a huge amount of raw materials sources, such as building materials or fossils fuels to be stored to produce petrol for thermal engines. However, on the other hand, Italy can rely on a high number of companies that have the potential of transforming raw elements into fine semi-finished or finished products: the symbol of the Made in Italy brand in the world. In this view, it is essential that these industries are as efficient as possible, optimizing production, incrementing profit, and reducing unnecessary costs.

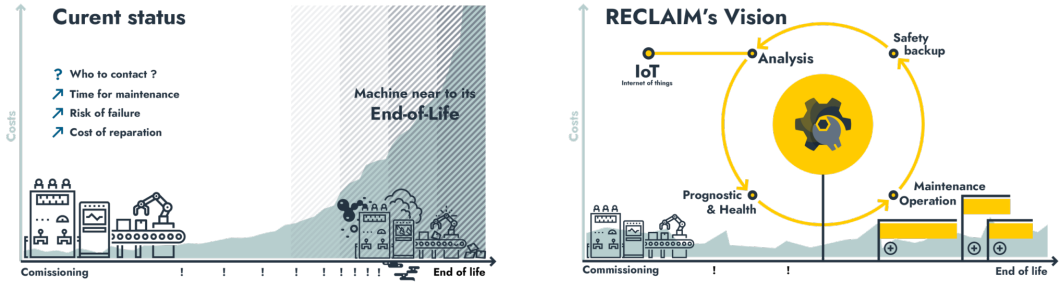
Industry 4.0 helps industries in this aim, through digitalization. The first step consists of the predisposition of a sensorial network that is able to analyze in real-time the health state of a machine or of a component. Then, the set of data generated by such network will represent the input for the next level, i.e. the analysis. In this regard, it is possible to implement different methodologies: machine learning, deep learning, statistical-based, digital twin, and so on. Finally, the decisions made by the analysis are implemented. This architecture is expensive, both in terms of implementation and the cost of electronics. However, in the long term, it is able to lead to higher profitability. This system is practically an investment for the company which must have a certain economic availability to carry out all the technical changes to the industrial plant which the industry 4.0 paradigm requires.

Thus, data analysis is able to cover a wide spectrum of possibilities. The value created by the historical functioning parameters measurements of a production line is huge. In addition, thank to the data enabling learning, this fleet of data could be exploited to improve productivity, which will be characterized by the generation of additional data to be used for the same reasoning, and so forth. In the same manner in which is possible to exploit data in order to increase the efficiency of the industrial plant, it is possible to determine if the choice of industry 4.0 is viable or not. The *business intelligence* aims to define with precision the company strategic plan based on a set of parameters like the income, the sector trends, the percentage of profit, the ROI, and so on.

The objectives of this thesis are inspired by one of the projects funded by the European Union that LINKS FOUNDATION is following at this moment. The project's development path starts from a diagnosis part, which provides the recognition of the presence or not of a defect inside a rolling element bearing, and ends with the estimation of the

remaining useful life of such bearing, in order to substitute it in the case of its health is sufficiently low. Finally, the basic idea of this thesis is to apply different prognosis data-driven methods to a bearing by exploiting the solutions just found on the diagnosis part.

1.1 LINKS Foundation and RECLAIM



(a) Actual situation of industrial plants.

(b) Future situation with the RECLAIM's vision.

Figure 1.1: Actual and future machinery maintenance.

LINKS Foundation (lin) is an Italian research center founded by Politecnico di Torino and Compagnia di SanPaolo which aims to develop cutting-edge technologies in different sectors: industry, logistics, space, Internet of Things (IoT), finance, energy, etc. LINKS tends to collaborate with a number of international institutions like universities, companies, and other research centers, in order to exploit different experiences and different know-how to design an optimal technical solution.

RE-manifaCturing and refurbishment LArge Industrial equipMent (RECLAIM) (rec) is one of the projects developed by LINKS and which has received funding from the European Union's Horizon 2020 program. Following the cooperation required by all the projects granted by the European Commission and the synergy contained in the vision of LINKS, RECLAIM is developed by a set of Spanish, Turkish, Slovenian, Germanic, and Swiss actors, which are divided into non-profit organizations, industrial and end-users. Practically the goals which RECLAIM has to reach are based on specific technological advancements required by the end-users, which depends on the way in which the pilots, another terminology for end-users, have planned to exploit the benefits insured by the industry 4.0 paradigm. The maintenance strategies applied on a production line are often time-consuming, due to the temporal window necessary to inspect the fault machine by the technicians and to repair the damage. In this sense a temporary stop leads also to a lack of profit, because the entire production chain has to be interrupted, generating a series of problems connected to the resource supply and to the marginal costs. In fact, in this case, the amortization of the set of assets used to produce a good is completely uncovered due to the set of goods that are not produced for the temporary stop. Simplifying this sentence,

it is possible to say that in case of stop, an industrial plant represents only a cost for a firm instead of an earning opportunity. In this context, RECLAIM aims to create a circular economy approach for the European industrials which is mainly based on increasing the production performance and efficiency by extending the lifetime of the machinery in order to re-use the production equipment, all encapsulated in a green-oriented approach. To reach this goal, RECLAIM is based on a Decision Support Framework (DSF) which analyze constantly the health status of machinery by taking into account cost models, diagnosis and prognostic reports, and optimization plans.

1.2 Maintenance Methods

Nowadays, the possible strategies which can be exploited to repair a component [Randall]:

- *Run-to-break*: fixing-only strategy. This old-fashioned method consists of running a machine until it broke down. In this way, it is possible to guarantee a long interval between shutdowns but in the meantime, a failure could generate a series of catastrophic consequential damages. Indeed, the production has to be stopped and the total costs incurred for the repairing increase rapidly. When the factory has machinery that is not too complex from a production and maintenance management point of view, it is convenient to use run-to-break maintenance;

- *Preventive maintenance*: in this case maintenance is done at regular intervals. Obviously, in order to prevent a failure, these periodic intervals are shorter than the expected time between failures. Otherwise, the kind of maintenance applied would be run-to-break instead of time-based maintenance. The principal advantage is that the aforementioned catastrophic failures could be avoided. However, the components are replaced before the end of their life and could happen that a number of unforeseen failures can still occur. Preventive maintenance is useful when it is possible to determine accurately the time interval. As an example, the life of a rolling bearing is characterized by a large variance, so it is difficult to apply this methodology;

- *Condition (predictive) maintenance*: this method consists in determining the potential breakdown of a machine. The advantages of this technique are clear, also if they are compared to run-to-break and preventive maintenance methods. The downside is that it required a set of condition monitoring techniques that are limited and normally not correctly applied;

The run-to-break method is applied only when a set of conditions occurs. The failures which could happen on that component must not affect the lifetime of other nearby elements. In addition, also the consequences on the production chain have to be taken into account. As an example, this method is perfectly suitable for an incandescent lamp, but not for developing a model which determines if the lamp has to be replaced or not. Due to its derisory cost, when a failure occurs it is time to change the component. The same speech is valid also for preventive maintenance. This procedure is suitable also for complex machinery but does not make use of any kind of data. The perfect example is the set of periodic reviews of a vehicle provided by the car manufacturer. In this

case, a lot of probable failures are avoided, and it is obvious that a possible mechanism that could alert the car owner of a possible source of damage is quite expensive for the buyer and not convenient for the manufacturer, which in this way could earn by the periodical reviews. Finally, predictive maintenance is the most technological and data-driven approach reported in this list. Predictive maintenance requires data, so inside an industry 4.0 environment it requires sensors that acquire signals. Nowadays machinery is linked to the philosophy of IoT, in which objects are connected to the internet in order to report their condition and to exchange messages between them (M2M technologies). For these reasons, predictive maintenance and IoT work together to reach the same goal and with the same perspectives. The actual horizon of the industrial domains is oriented to have machinery with a lot of sensors, and which alert the employees about their condition, speeding up the production and reducing to a minimum the waste.

1.3 Predictive Maintenance

The Condition Based Maintenance (CBM) strategy aims to control continuously the health status of a component or a system, in order to plan an optimal maintenance intervention. The path which transforms a measurement in a final remaining life value is composed of a set of steps: the data acquisition, the Health Index (HI) construction, the Health Stage (HS) division, and the RUL prediction.

The first one is the data acquisition, in which the signals are sampled by the sensors, stored, and then analyzed. In general, different kinds of measurements can be acquired, e.g., accelerations, acoustic emissions, temperatures, pressures, voltages, currents, and so on. Data are managed by an acquisition system, and all these values are stored in a database and are managed by an event-driven framework like Apache Kafka.

The second step is the definition of an index that can estimate the health status of the machine, used to define the perfect moment in which the RUL prediction has to start. This metric is useful because it gives a good idea of when it is possible, and it is convenient, operate the maintenance of the machine. In the literature, there exist two classes of HI: the Physical HI (PHI) and the Virtual HI (VHI). The former comes from the statistical analysis of the monitoring signals, such as the kurtosis, the skewness, the Root Mean Square (RMS), etc. Instead, the latter is constructed by fusing multiple forms of PHI. Not all the possible metrics can become an HI since the index has to respect a set of parameters that are necessary to affirm if one component is in one status rather than another. More in detail, the HI has to classify the status of a component, or a machine, in a unidirectional and unequivocally manner: two components that have the same health index must lie in the same conditions, possibly with the same expected RUL. In this sense, there exists a set of metrics that can be used to evaluate the goodness of a specific HI. Javed et al. [2014], Liao et al. [2016], Zhang et al. [2016] and Liao [2013] made a work on the monotonicity based on the derivatives of the Hl instead the one studied by Camci et al. [2013] is able also to be robust against a degradation path not perfectly evident to the possible presence of swinging HI values. Another interesting metric is the robustness, present in the work cited previously written by Zhang et al. [2016]. The idea behind this metric is very similar to the definition of monotonicity given by Camci et al.

[2013]. In fact, the robustness is useful to assert if the possible presence of fluctuation in the HI affects the result or not. Javed et al. [2014] and Zhang et al. [2016] defined also the trendability, namely the correlation between the HI and the time, instead Zhao et al. [2013] and Liu et al. [2016] studied the correlation between the HI and the stage sequence, in order to evaluate the ability of a specific HI to classify sequence to the correct HS. Finally, consistency is useful when it is necessary to work with multiple HIs and, in the minds of Liu et al. [2016], these different definitions of indexes, as they try to represent the same situation, have to be correlated.

The third step is the subdivision of the life of a component into different HS, according to the value of the HI. The number of splits is dependent on the type of dataset, which is used for assessing the data-driven methods. If the degradation of the component is gradual over time, a one-stage approach is normally the most suitable choice: for instance, the flank wear of a willing machine.

Instead, if it is possible to define two completely different situations of which the first one has a health index more or less stable, synonymous with a good condition, and a second situation characterized by a completely oscillating and nervous health index, a two-stage approach could be a good choice. The separation between the first stage and the second stage is guaranteed by a sudden growth of the HI, which assumes values out of the previous distribution. In fact, the moment in which the component changes stage is called First Predicting Time (FPT) and there exists multiple methods for computing it. Wang et al. [2016] used the 3σ interval using the Mahalanobis distance for computing the HI. Instead Qian et al. [2014] and Shakya et al. [2014] determined the FPT by computing the Chebyshev inequality function.

The concepts of HI and HS are strictly correlated to the basic principles of reliability engineering (Benbow and Broome [2008]). One of these dogma tries to abstract the life of a mechanical component with a particular function: the bathtub curve (Klutke et al. [2003]). It is reasonable to expect that the failure rate of a component is practically equal to zero at the beginning of the component's life and, as time passes and the life of the component gradually decreases, it reaches not negligible values. However, the real trend is quite different and it has in common with the theoretical one only the last part. The first phase, which would start with a minimal failure rate, has a corresponding RUL not equal to 100% because the single elements which constitute the mechanical component are affected by the manufacturing errors that are then removed only by the wear during the first running period. This behavior is valid also and even more so for the bearings. In fact, the vibrational signal, which corresponds in this case to the classical failure rate, at the beginning is quite high (run-in phase). Then, when the gentle wear harmonizes the defects, the vibration generated by the bearing tends to decrease (useful life). Finally, as soon as a fault re-appears, the magnitude of the vertical and horizontal acceleration exponentially increases, leading the bearing in the terminal part of its life (wear-out). The figure 1.2 depicts exactly this kind of behavior.

Finally, the one-stage and the two-stage divisions can be merged in a multiple-stages split when it is possible to recognize a swinging degradation stage inside the two-stages unhealthy part. In fact, in such situations, it is not possible to affirm that all the samples after the FPT are characterized by a continuous degradation, that this is exponential or linear does not matter. For these reasons, it is possible to split the entire evolution of

the HI into a minimum of three splits, the healthy stage, the degradation stage, and the unhealthy stage. Kimotho et al. [2013] defined a number of splits equal to five, while other researchers prefer focusing their work on exploiting various cluster algorithms to find the optimal health stages. Ramasso et al. [2012] used the K-nearest neighbor (KNN), Javed et al. [2015] and Liu et al. [2016] used the fuzzy k-means and Scanlon et al. [2012] the k-means.

The last step is the RUL prediction, which can be done in different ways. In the literature, there exist two main categories of approaches that can be exploited to reach this goal: statistical approaches (or model-based approaches) and data-driven approaches. The former are principally based on fitting a set of parameters by analyzing continuous observations. Lei et al. [2016] exploited the Kalman filter as the estimator for the stochastic model and Li et al. [2015] used a stochastic model also to predict exactly the value of the FPT. Singleton et al. [2014], Cui et al. [2019] used instead only a Kalman filter in order to fit the HI curve. Hu et al. [2018] exploited a Wiener process, which generally makes use of the Browning motion, and Si et al. [2013] applied also a recursive filter. Sloukia et al. [2013], Tobon-Mejia et al. [2010] used Gaussian hidden Markov models to analyze the degradation process. The latter can be further split into classical machine learning and deep learning approaches. Inside the machine learning algorithms, it is possible to put the Support Vector Machine (SVM), as done by Benkedjouh et al. [2013], Loutas et al. [2013], Yan et al. [2020], or a decision Yang et al. [2009, 2008] used a decision tree in conjunction with a fuzzy system. On other hand, some researchers exploit different Artificial Neural Network (ANN)s in order to predict the bearings' RUL. Yuan et al. [2016] used a Long-Short Term Memory (LSTM) neural network. Ren et al. [2018], Zhu et al. [2018] used a Convolutional Neural Network (CNN) by giving as input the Spectrum-Principal-Energy or the spectrogram. In conclusion, as classical in engineering, there exists also hybrid approaches. For instance, Sloukia et al. [2013] used a mixture of Gaussians hidden Markov model and support vector machine. Wang et al. [2018] used an SVM to find the optimal parameters for the Kalman filter.

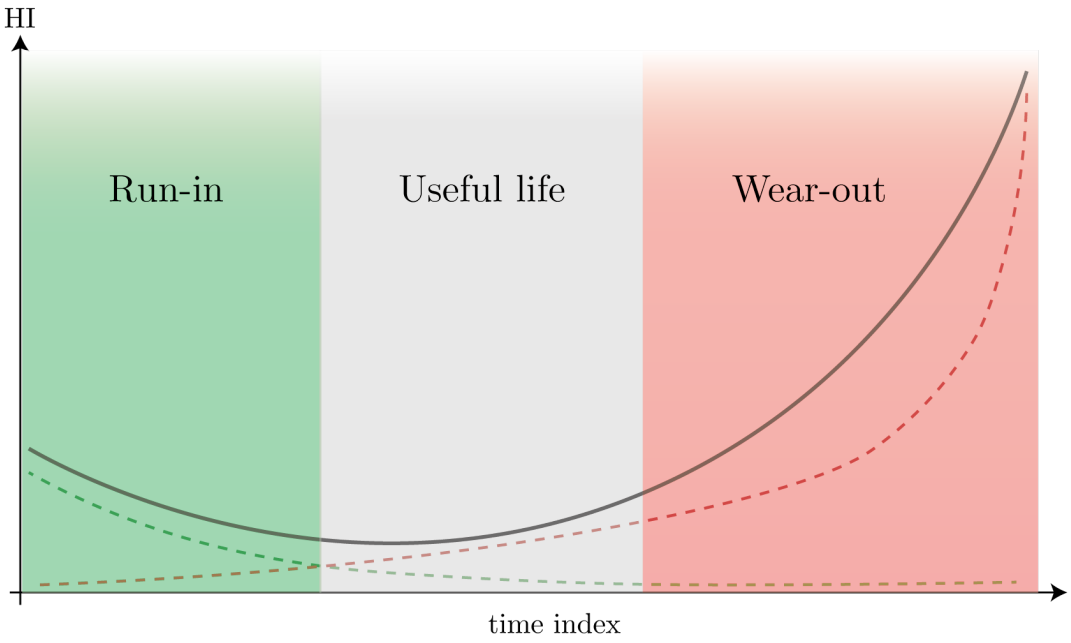


Figure 1.2: Typical bathtub of a mechanical component.

Chapter 2

Rolling Element Bearing

2.1 Introduction

The main purpose of this thesis, in the context of the RECLAIM project, is to predict the RUL of a specific component: the Rolling Element Bearing (REB). A survey conducted by Albrecht et al. [1986] showed that about 41% of motor failures are connected with bearings. This percentage is an example of how bearings are important inside machinery, which however could contain hundreds or thousands of REBs, even if only a part of them are mounted in sensitive areas. On the other side, predicting the life of such components could become strategically rewarding, for all the reasons explained before. In order to contextualize the works presented in this document, a brief introduction about the bearing will be presented in the next paragraph. As the first challenge of this thesis is to transfer the diagnosis methods to the prognosis phase. Furthermore, a study about the fault detection of bearings and other results are also presented in some dedicated sections.

A REB is made of a set of four principal components: the outer ring, the inner ring, the rolling elements, and the cage (Figure 2.1). The rolling elements (balls or cylinders) rotate in the tracks excavated in the internal side of the inner and outer rings. The rolling elements are immersed in a lubricant fluid which aims to reduce drastically the friction generated by the rotation. These elements are kept close but separated thanks to the cage, which prevents the contact between each rolling element and holds the REB in case of damage. Normal application of REBs predicts that the inner ring is mounted on the shaft and, instead, the outer ring is mounted on a stationary housing. A possible case is that damage could interest each of these four main parts and each of them changes the vibrational signal differently. This is due to the fact that these components generate a set of impact vibrations in a very short time interval.

The next sections will explain more in detail the conditions which generate a fault and the particularity of the signal emitted by the REB.

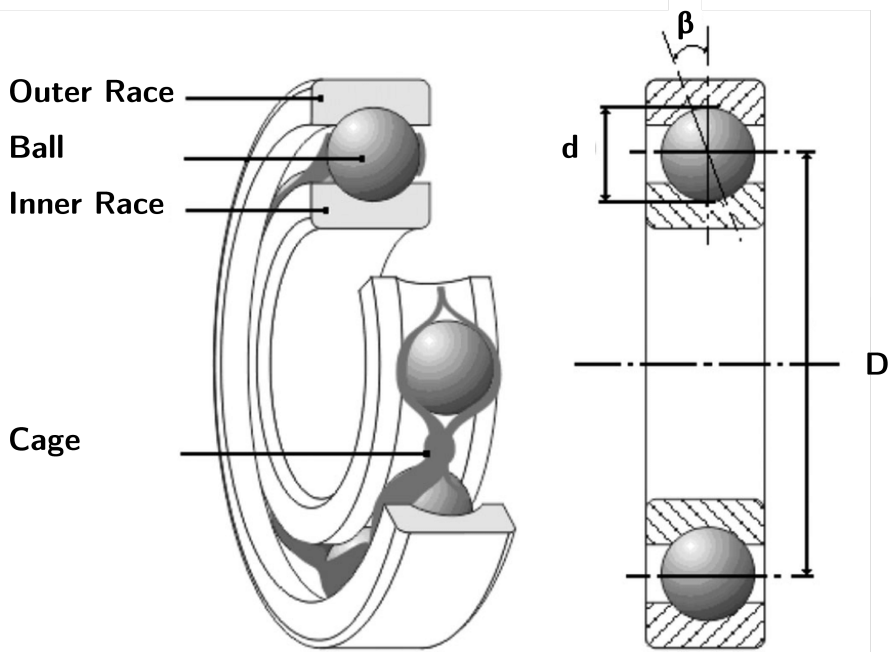


Figure 2.1: Schematic view of a rolling element bearing taken from Koura et al. [2018].

2.2 Sources of Failures

Normal degradation of REBs is obviously linked to wearing. Lubrification is also extremely important in order to prevent increasing friction in metal-to-metal contact between the rolling elements and the track. A poor level of lubrification can only enhance natural wear. High humidity contests can lead to surface oxidation of the bearing and this produces particles which in their turn can degrade the component via abrasion. The same particles can be produced also inside the REB as its behavior is similar to a capacitor. In fact, the outer and the inner ring of the REB could be associated with the capacitor armatures instead the lubricant represents the dielectric between the armatures.

To better explain and analyze other possible scenarios, it is necessary to cite also the situations in which the REB is stressed over its maximum capacities, i.e. slippage of a maximum permissible load. Morgan and Wyllie [1969] conducted a survey in 1969 about the possible failures of a bearing. Table 2.2 extracted from such paper shows that the principal cause of failure is corrosion, which was found in 39 percent of the total number of bearings. This value is double with respect to the second cause of failure, the dirt, and about four times the misalignments, the insufficient grease, and the excessive load. Inspecting more specifically Morgan and Wyllie [1969], it could be noted that the corrosion depends on the kind of lubricant used inside the bearing. However, the authors argued that it is difficult that corrosion depends on the lack of grease. The research has different goals, and one of them is evaluating the consequences generated by choosing one lubricant instead of another. In any case, these further considerations are outside the scope of this report.

Condition	All bearings		Bearings lubricated with XG-274 to DGS/6921A	
	Number	Percentage of bearings	Number	Percentage of bearings
Bearings received*	614	—	439	—
Bearings failed	596	97	429	98
Bearings badly failed	307	50	209	48
Bearings too badly damaged for cause of failure to be assessed	18	3	10	2
Bearings with fractured components	39	6	28	6
Bearings flaked	180	29	129	29
Bearings with severe corrosion	77	13	48	11
Bearings with the following main causes of failure (more than one can be present in one bearing)				
Corrosion	238	39	151	34
Dirt	112	18	91	20
Misalignment	74	12	59	13
Insufficient grease	72	12	56	13
Excessive axial load	63	10	43	10
Hard grease	43	7	5	1
False brinelling	38	6	27	6
True brinelling	21	3	20	5
Grease inadequate	19	3	14	3
Natural fatigue	19	3	17	4
Poor fit	19	3	12	3
Rotating radial load	10	2	7	2
Soft grease	4	< 1	3	< 1
Overheating	4	< 1	4	< 1
Passage of electric current	3	< 1	2	< 1
Too much grease	1	< 1	1	< 1

Figure 2.2: The summary of the main fault causes taken from Morgan and Wyllie [1969].

2.3 Signal Topology

The physics of the REBs is necessary to define a mathematical model which is able to represent the behavior of this component during its life cycle. As said before and based on the principles of the vibration analysis, the vibrations emitted by the REB are different depending on the presence or not of defects, or if it is in a critical state due to damage. However, each vibration is characterized by a particular behavior which is useful in order to determine the state of the component. The spectrum of a gear vibrational signal is different from a spectrum of a REB vibrational signal. In the former, it is possible to notice harmonics in correspondence of frequencies multiples of the shaft rotation speed. In the latter, instead, characteristic frequencies are generally not harmonics associated with the shaft speed.

In addition, different mechanical components have associated vibration signals with different statistical properties. The most important division of a signal is stationary and non-stationary. The former means that the statistical properties do not vary over time. The latter means that the signal does not satisfy the condition for stationarity. Then stationary signals can be further divided into several categories: random, deterministic-periodic, and deterministic-quasi-periodic. The main difference between random and deterministic is that random signals cannot be predicted but their statistical properties remain the same along time. Meanwhile, the value of a deterministic signal can be predicted at any time in the future. Non-stationary signals can be split into a number of subcategories as well, in particular there exist non-stationary cyclostationary signals and non-stationary transient signals. The concepts of energy and power are necessary to distinguish between cyclostationary and transient signals. What it is important to cite is

that cyclostationary signals by definition have constant power and, consequently, infinite energy. A cyclostationary signal is an amplitude modulated with noise. The problem arises due to uniform white noise, as the spectrum of the modulated signal. Therefore, the modulation part is obscured by noise. Signal topology is relevant because it conditions the set of possible processing techniques which have to be applied to raw data. REBs signals are cyclostationary, thus all the issues associated with this particular kind of signal remain valid.

2.4 Fault Detection

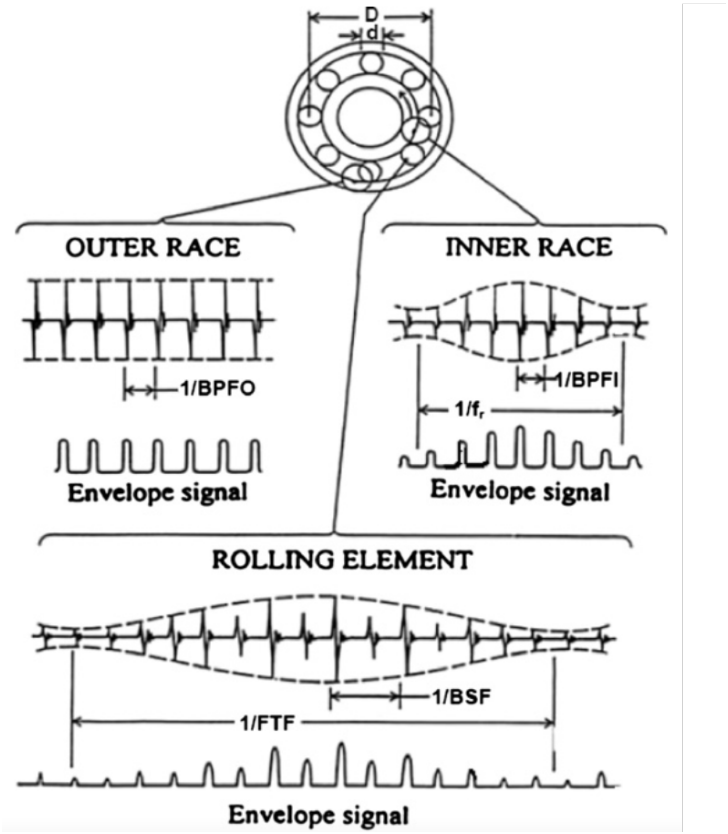


Figure 2.3: Signals generated by various faults in a REB and their envelope signals. Illustration taken from Randall [2021].

From the physical point of view, an impulse is generated when a rolling element strikes a fault on the outer or inner race. This spike excites also the high-frequency resonances of the bearing, i.e. the mechanical parts between the structure and the accelerometer. The same happens when the fault is localized on a rolling element. The only difference is that in this case two shocks are introduced instead of one, due to the hit against the inner and the outer race. The first person who recognized that such faults were to be found in the

frequency band associated with the resonance frequencies is Balderston [1969]. This band has the center localized in a frequency

Each of these impacts is periodic, and the period depends principally on the shaft speed and on the position of the element of which you want to compute the speed. Thus, the information useful for the fault detection is stored in the period of these spikes and not in the resonance frequencies excited. In fact, based on this period and knowing the parameters of the REB, it is possible to determine if there is an inner, outer, or rolling element fault. The frequencies associated with these periods are called Characteristic Defect Frequencies (CDF) and are defined as follows:

$$BPFO = \frac{nf_r}{2}\left(1 - \frac{d}{D}cos\phi\right) \tag{2.1}$$

$$BPFI = \frac{nf_r}{2}\left(1 + \frac{d}{D}cos\phi\right) \tag{2.2}$$

$$FTF = \frac{f_r}{2}\left(1 + \frac{d}{D}cos\phi\right) \tag{2.3}$$

$$BSF = \frac{D}{2d}\left(1 - \left(\frac{d}{D}cos\phi\right)^2\right) \tag{2.4}$$

where n is the number of rolling elements, f_r is the shaft speed in Hz, ϕ is the contact angle, i.e. the angle of load from the radial plane, d is the ball diameter and D is the pitch diameter. Figure 2.3 shows well the signals generated by the REB in the case of inner, outer, and rolling element fault. In the figure, it is evident that these signals are amplitude modulated at the resonance frequency by the aforementioned bursts.

In addition, there is another issue that complicates the analysis. The REB is subject to a load, which is not the same for each portion of the structure. Then, when a ball passes through a loading zone, its radius varies and consequently, also its speed varies. The cage ensures that all the rolling elements rotate at a common speed by maintaining the mean speed constant. The downside of this behavior is that in this manner the rolling elements slip. The slip is in the order of 1-2% or of 0.01-0.02 rad. These issues lead to cataloging the signal emitted by a REB in the category of cyclostationary signals. In this situation and based on what is explained in Section §2.3, it is not possible to retrieve the bearing diagnostic information from the frequency spectrum. In 1979 Braun Braun and Datner [1979] argue that a bearing signal is not completely periodic, and it is affected by a random displacement in time. This result was proved by applying the synchronous averaging technique to the vibrational signal emitted by the REB. Normally, the harmonics associated with the resonance frequencies are orders of magnitude higher compared to the characteristic frequencies and in this situation, they can be masked by other spectrum components. However, in a simple situation in which the random slip is assumed to be 0, it should be sufficient for a diagnostic tool to measure the distance between the harmonics in the frequency band assigned to the resonance frequency to find the desired information. Unfortunately, the presence of the random slip, even very small, leads to compressing the harmonics in the resonance frequencies band on top of each other.

This number of events prunes the possibilities of retrieving such useful diagnostics from the spectrum of the raw signal. Over the years, several methods have been found to solve these issues. Nowadays, by applying a set of specific techniques, it is possible to translate the CDFs in a low-frequency band with extremely high precision.

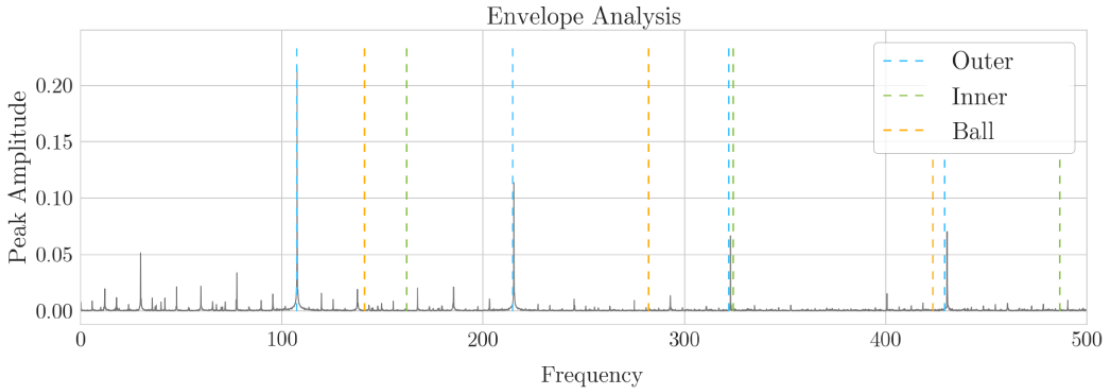


Figure 2.4: Example of a fault diagnosis analysis result.

In this way, the presence or not of a fault is evident by looking at the resulting spectrum. If in correspondence with the theoretical values of the CDFs there is a real harmonic, then it means that the REB has a defect associated with the kind of that CDF, as shown clearly in Figure 2.4. It is possible to compute this kind of spectrum and then you only need to compare the theoretical values of the Ballpass Frequency Outer race (BPFO), Ballpass Frequency Inner race (BPFI) and Ball Spin Frequency (BSF). If around one of them there is a real harmonic, then respectively the outer, the inner or the rolling element defect is present inside the bearing. In this particular example, the bearing is affected by an outer race fault. In fact, in correspondence with the first harmonic of the BPFO, the first blue dashed vertical line is perfectly adjacent to a real spectrum harmonic. As a counterexample, the orange and green dashed lines are in correspondence of harmonics with low magnitude. Instead, the multiples of the BPFO are in their turn near other real harmonics.

Chapter 3

Data Preparation and Analysis

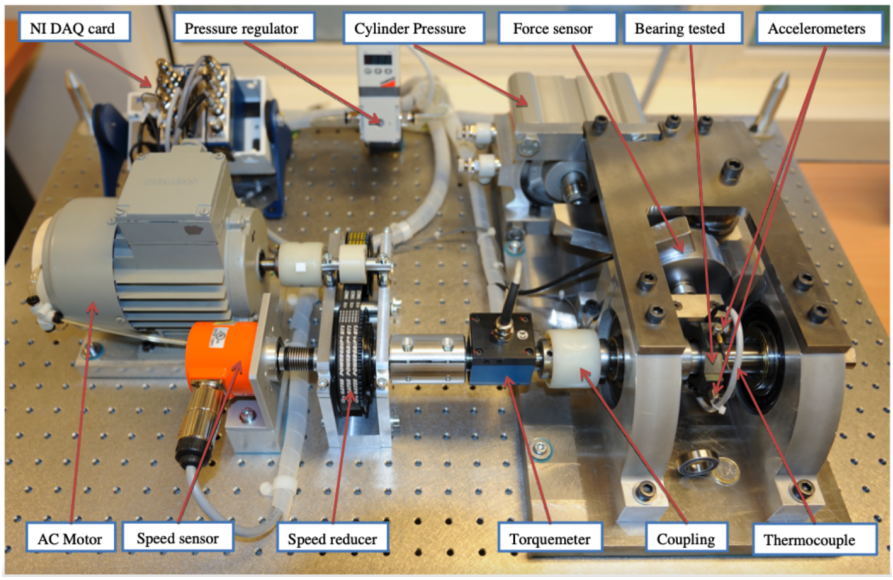
3.1 Datasets

In the literature there exist fundamentally two types of datasets that can be used for our purpose. The first one is the IMS bearing dataset, provided by the Prognostic Data Repository of NASA (Lee et al. [2007]). This dataset is constructed by running a set of four Rexnord ZA-2115 REBs until the total amount of debris produced by wear and collected by a magnetic plug exceeds a predefined threshold. The second one is PRONOSTIA, provided by the FEMTO-ST Institute. In this case, different bearings with different operating conditions run until their vibrational acceleration exceeds the value of 20g.

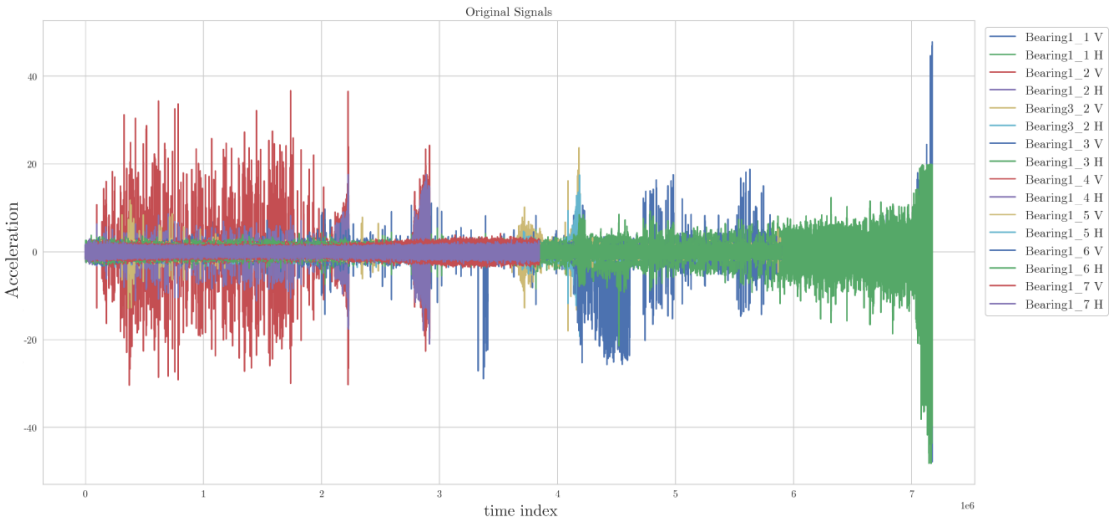
Both these fleet of data are useful to construct the HI or to develop predictive maintenance models in order to estimate the RUL of a bearing, which is also the purpose of this thesis, even if there are some substantial differences between them. First of all the in IMS dataset, it is possible to apply the fault diagnosis methods which are related to the diagnostic approach of such components. In the PRONOSTIA dataset, it is not possible to evaluate these particular features. The second difference is that bearings that compose the IMS dataset are characterized by an anomalous trend, which in some parts is increasing and in others is decreasing, a synonym of degradation which provides a spall generation in the initial life and its consequent grinding due to the gently wear, which however is insufficient in the last phase of the REB's life. Instead, the bearings tested by the PRONOSTIA platform does not present alternative trends, even if the RUL of some of these components is really difficult to be predicted because of the What they have in common is the same variability in the duration of each bearing. In fact, both the IMS and the PRONOSTIA dataset contain bearings with are completely different from each other in terms of total useful life. In particular, the total test time of a REB proved by the PRONOSTIA platform can vary from 1h to 7h, instead, the IMS dataset contains bearings which can vary from 7 days to 35 days

There are no unique motivations to choose one dataset rather than another. However, to the best of my knowledge, the researches found in the literature which is the comparison baseline for this thesis are built with the PRONOSTIA dataset. For this reason, this will be also the choice for this work.

3.2 PRONOSTIA



(a) PRONOSTIA platform, taken from Nectoux et al. [2012], with the specification of the names of all its components.



(b) Vertical and horizontal accelerations of each bearing of condition 1. The letter H indicates the vertical acceleration signal, instead the letter V the vertical one.

Figure 3.1: The test rig (a) and the set of all the signals (b) used in this work.

The data in this thesis come from the IEE PHM2012 Predictor Challenge (Nectoux et al. [2012]) experiment data provided by the FEMTO-ST Institute in Besançon, France. During the various experiments, the 17 test bearings are subjected to 3 different loads and

3 different speeds. In order to sample all these signals, the PRONOSTIA test rig is used, which can be decomposed into three main portions: the rotating part, the loading part, and the measurement part. The platform, with the specification of all its components, is shown in Figure 3.1a.

The rotating part, in turn, contains the asynchronous motor, the gearbox, and the shafts. The motor has a power of 250W and a rated speed of 2830 rpm, which becomes less than 2000 with the addition of the shaft. The loading part is responsible for the exponential decreasing of the REB's life and in the extreme cases of forces higher than 4000 N, it can ruin hopelessly the bearing. In fact, in this situation, an initial fault can propagate through contacts to the regions nearby the initial damaged component. Consequently, it is difficult to recognize a specific failure pattern because multiple issues can occur simultaneously.

The load is generated by a pneumatic jack and the pressure can be managed by a regulator. The data is captured by the measurement part, which is based on four components: the two accelerometers, the temperature sensor, and the NI DAQ card. The vertical and horizontal vibration signals are sampled by the two accelerometers, which are positioned at 90° to each other and the temperature is measured by a thermocouple which is positioned on the external bearing's ring[1]. The sampling frequencies are respectively 25.6 kHz and 10 Hz and the vibration recordings are samples every 10 seconds for 0.1 s. This means that each sampling window contains exactly 2560 samples but the number of sampling windows depends on the life of the bearing under analysis, which can occupy in total from 1h to 7h, as it is visible in Figure 3.1b.

The run-to-failure experiments plan to run the bearing until the amplitude of its vibration signal overpassed 20g. This kind of data is normally difficult to be gathered, because generally, these processes may take several months or also several years. In addition, in a normal situation, machinery which has maintenance problems is stopped before it leads to other security problems. Thus, the datasets normally available can be classified as run-to-maintenance data. Consequently, also the entire acquisition process could take this same period, making this kind of dataset really rare. For these reasons, even if the run-to-failure datasets exist in the commercial or in the research context, they are kept secret in as much as they have an extremely high intrinsic value.

Moreover, it is not possible to exploit the reliability laws such as the L_{10} because the variance of the estimated life of a REB is really large, even if the manufacturer and the model of the bearing are the same. The training and the test sets are organized in folders associated with each bearing. Inside each directory, there is one ASCII file for each sampling window. Regarding the *.csv* s associated with the vibrational data, the format of one row, i.e. one sample, is the following: the first four columns are referred to the time of sampling in hours, minutes, seconds, and μ-seconds and the last two columns contain respectively the horizontal and the vertical acceleration values. Table 3.1 reports the specifications, in terms of radial load and speed, for each operating condition and the organization in folders of the dataset, with the name of each directory. In conjunction with the temperature measurement, all the analyses presented in this document are based

[1] All the analysis presented in this work does not make use of the temperature.

only on condition 1. This choice is shared by the author of the thesis and the supervisors, who have the pleasure of testing more kinds of techniques rather than exploiting all the possibilities associated with one single method. The goal of this project is to find the best predicting life solution in a plausible real scenario, ranging as much as possible in the forest of available data-driven algorithms. In this sense, the parameters and the results which make one solution better or worse than another one are computed only on the test bearings of condition 1. This does not exclude that one particular model performs well on the bearings of conditions 2 and 3, but these results cannot affect the development of such solutions.

	Condition 1	**Condition 2**	**Condition 3**
Radial Load [N]	4000	4200	5000
Speed [rpm]	1800	1650	1500
Learning Set	Bearing1_1	Bearing2_1	Bearing3_1
	Bearing1_2	Bearing2_2	Bearing3_2
Test set	Bearing1_3	Bearing2_3	
	Bearing1_4	Bearing2_4	
	Bearing1_5	Bearing2_5	Bearing3_3
	Bearing1_6	Bearing2_6	
	Bearing1_7	Bearing2_7	

Table 3.1: PRONOSTIA operating conditions and dataset description.

Instead, the bearings analyzed by the PRONOSTIA platform are completely different from each other. This is due, as mentioned before, to the different behavior with which these bearings react to the same speed and the same force load. The correspondent result is that some of them present some anomalies during their running period. For instance, Bearing 1_1 (Figure 3.2a) is characterized by a vibrational signal which increases as quickly as the wear is consuming the RUL of the component. Thus, in this case, the wear-out phase is completely evident, contrarily to the run-in phase which is hidden and for which it is necessary to process the data in order to highlight this trend. The same speech can be done also for Bearing 1_2 (Figure 3.2b), at least for the horizontal acceleration since the vertical signal presents a lot of spikes that are not useful for preparatory to the RUL prediction. On the other hand, Bearings 1_5 (Figure 3.2e) and Bearing1_7 (Figure 3.2g) represent a particular case. The former seems to be more or less stable and constant during its life. In other words, for these components, no proofs of wear are distinguishable in the signal. The latter is very similar to the former except for a different run-in and wear-out phase, in which it is possible to notice a slight decrease of the signal at the beginning and a more evident increase at the end.

For the sake of completeness, it is important to highlight that the raw vibrational signal is not sufficient to retrieve this kind of trend, but it is necessary for a further step and, consequently, a further analysis. Otherwise, a basic and simple vibrational analysis should be sufficient to solve all the problems correlated to predictive maintenance but, in this case, statistics approaches, machine learning, and deep learning algorithms would be useless and time-consuming. This is the principle that is at the foundations of the HI

introduced in Subchapter §1.3.

3.3 Data Preparation

One of the most important phases which have to be performed on the data to diagnose the status of a REB is demodulation. In order to enhance the presence of the features which are relevant in case of fault, the first cornerstone on which to build the RUL prediction regards the application of this diagnosis preprocessing to the prognosis data. More in detail, in a noisy signal, like the one sampled by an accelerometer mounted on the housing of a REB, it is difficult to identify the sidebands of the carrier frequency which modulate the original time series. Since also the noise occupies high frequencies harmonics, the demodulation implies also denoising, incrementing in this way the Signal-to-Noise Ratio (SNR) (K. Kamaras [2016]).

Demodulation implies transferring the information in the low-frequency region, as already explained in Section §2.4. In this way it is possible to enhance the resolution of the diagnosis, achieving better results, and making the solution case-independent.

On the other side, demodulation is a delicate phase, due to the difficulty of choosing the adequate band on which to compute the procedure. In the 70s, the transducer frequency was used to bandpass or high-pass the signal. Unfortunately, H. Engja [1977] demonstrated that this solution was not universal, because of the presence of other excitations which lead to false readings. Over the years, several methods have been proved to find out the perfect band on which to execute the demodulation. Bradshaw and Randall [1983] was the first to use cepstrum analysis for bearing fault detection. This method performs as well as SK, maybe better in some cases, and anyway it requires that the harmonics are separated but as reported in Section §2.4 this is not possible due to the presence of a random slippage. In addition, if the REB is immersed in an environment full of trouble due to the presence of other components like gearboxes, for instance, characterized by a strong vibration signal even in good conditions, the REB vibration signal could be affected, and the resulting diagnosis could become more critical. For these cases, it is possible to apply before the demodulation other kinds of techniques like Linear Prediction (Kay and Marple [1981]), DRS (Antoni and Randall [2004]), and so on.

In conclusion, the vibration signals read from the dataset, before being processed by the data-driven algorithms, are demodulated by using the wavelets Crocker [2007]. Substantially wavelets could be considered as a set of impulse responses of filters. In addition, they perform better at high frequencies and, for this reason, are very suitable for our case. Figure 3.3 show the difference in terms of the original and denoised signal for some bearing of the dataset. At first glance, it seems that there are no differences before and after the application of the technique. A more empirical way to make the comparison lies in the exploitation of the kurtosis: a higher kurtosis means that the fault features are more identifiable and, consequently, the damage is more serious.

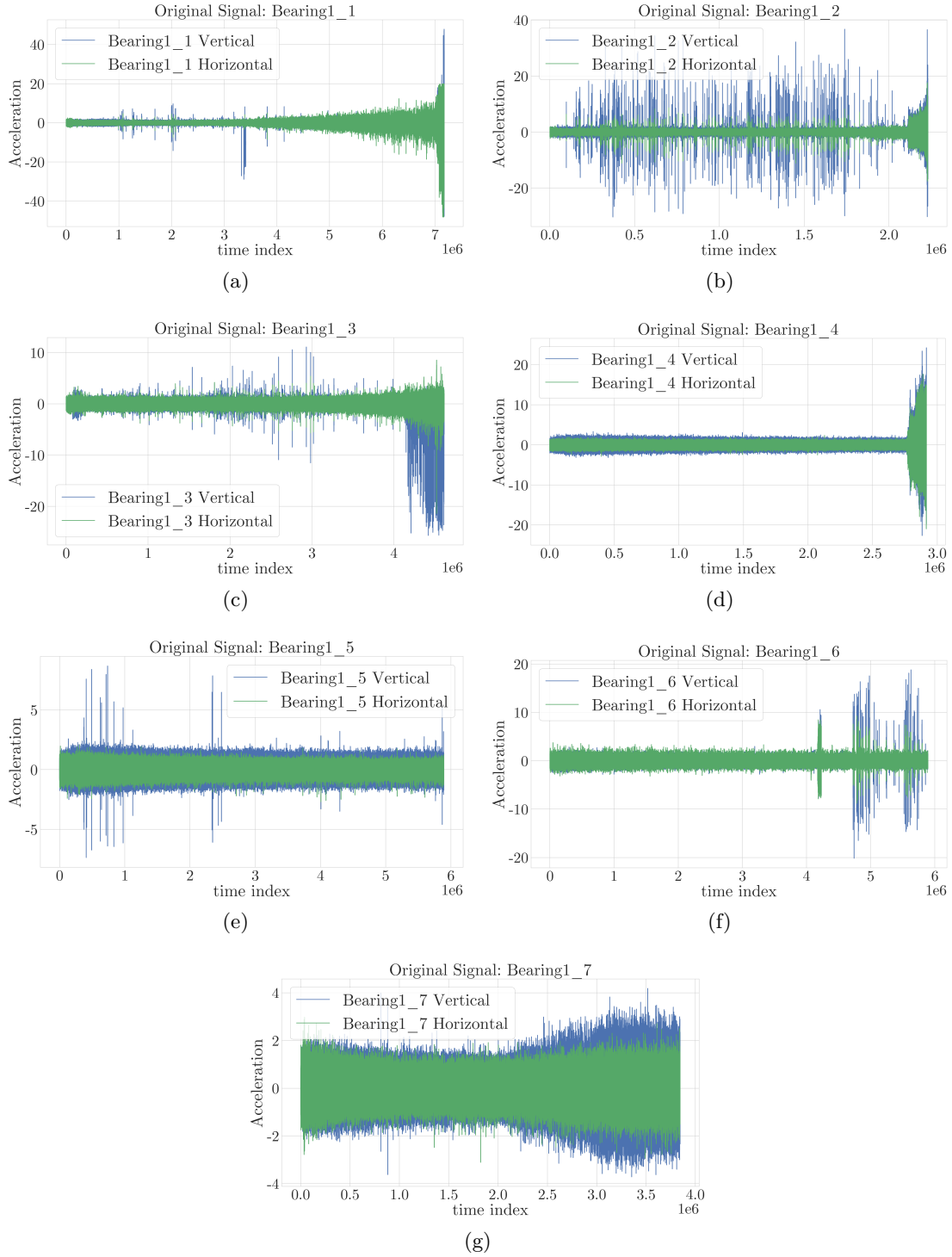
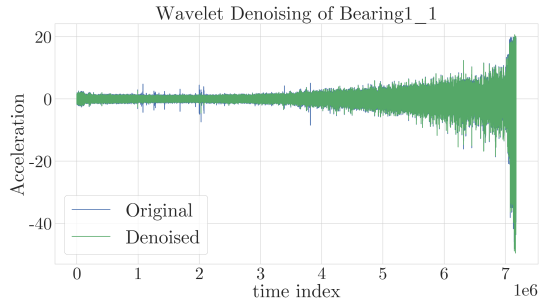
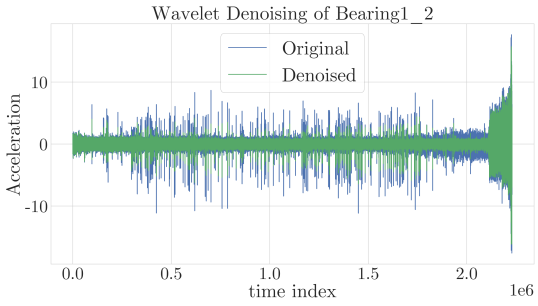


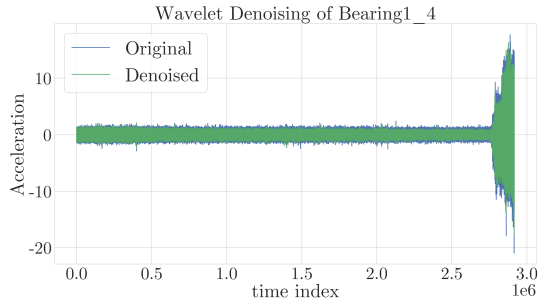
Figure 3.2: Vertical and horizontal acceleration of Bearing 1_1 (a), Bearing 1_2 (b), Bearing 1_3 (c), Bearing 1_4 (d), Bearing 1_5 (e), Bearing 1_6 (f) and Bearing 1_7 (g).



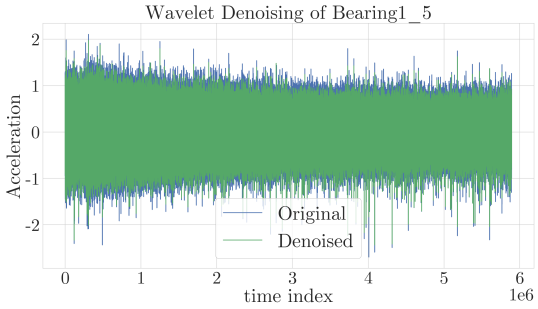
(a) Original kurtosis: 124, denoised kurtosis: 139



(b) Original kurtosis: 40, denoised kurtosis: 60



(c) Original kurtosis: 50, denoised kurtosis: 80



(d) Original kurtosis: .39, denoised kurtosis: .53

Figure 3.3: Vibration signals before and after the application of the wavelet denoising technique.

Chapter 4

RUL Prediction

Subchapter §2.4 is dedicated to the principles of a REB fault diagnosis and it represents the starting base for writing this thesis. The set of methodologies that will be analyzed from now on are derived starting from the diagnosis methods. If on one side it is important to determine if a mechanical component, like a bearing, is breaking at the actual state or not, on the other side it is fundamental to determine the total remaining useful life of such component. In this way, it is possible to estimate, with a certain dose of security and reliability, the most suitable moment to change the component, to the net of the depreciation and the raw material recycling. It is evident that building an infrastructure that brings inside it all the benefits of industry 4.0 is quite expensive and it is up to the company management to trace decisively the narrow boundary between a completely digital orientation of the industrial chain and a loss of production due to the problems generated by the unplanned machinery failures. As explained in the first part of this document, bearings are characterized by high magnitude vibrations during the breaking phase and this could be deleterious for the component and for all its mechanical surroundings. In fact, a set of *chain breaks* can trigger, leading the healthy components to degrade, compromising the whole functioning of the machinery under analysis. On the other hand, it is possible to exploit the prognosis, in order to *squeeze* at most an industrial machine, avoiding its retirement until it runs out all its possible production extent.

4.1 Health Index

The degradation of a REB not always is visible by observing only the vibrational signal. As introduced in Subchapter §1.3, a good condition maintenance algorithm has to transform the raw data gathered from the sensors mounted inside the industrial machinery into a metric that has a development similar to a bathtub curve, typical of a reliability engineering analysis. However, as reported in Subchapter §3.2, some bearing contained in the dataset has a degradation trend which in no way is similar to a bathtub function. The HI is a metric that tries to extrapolate such information, especially in these particular cases. Otherwise, the estimation of an accurate RUL from these components would become quite unreachable. Since the implementation of the predictive maintenance system is expensive,

the final user must have the security that such technologies work in the correct way, or else the entire investment made by the company could be placed at risk.

The well-known method to generate the HI is the following. When the component is installed, its health parameters are computed and for each sampling window, composed of a set of samples which in the case of PRONOSTIA are 2560, the state of the machine is calculated. Then, the deviation of this state with respect to the initial one composes the value of the HI. This procedure is sub-optimal in the sense that in the run-in phase the vibrations emitted by a REB are not at the minimum level. To solve this problem, it is possible to acquire the healthy state of the machine just as soon as the run-in is ended. This period cannot be determined univocally and with precision, even if this uncertainty will not lead to substantial differences in the value of the HI. On the other hand, the way in which these parameters are computed and the metric used to compute the deviation of the HI represents the true challenges that this index has to win. The first challenge consists of determining the parameters which are able to characterize the actual REB's healthy state. First reasoning would bring to the classical statistical features used overwhelmingly in data science and, in particular, during any features extraction phase. Effectively, these parameters can be accepted as a baseline, because also they can fail in certain situations. In this project, the one used for this purpose are: mean 4.1, skewness 4.2, shape factor 4.3, log-ratio 4.4, crest factor 4.5 and kurtosis 4.6.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.1}$$

$$m_k = \frac{1}{(n-1)}\sum_{i=1}^{n}(x_i - \bar{x})^k; \quad \text{skew}_i = \frac{m_3}{{m_2}^{3/2}} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{\left[\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right]^{3/2}} \tag{4.2}$$

$$\text{SF}(x) = \frac{\text{RMS}}{\text{Mean absolute}} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}}{\frac{1}{n}\sum_{i=1}^{n}|x_i|} \tag{4.3}$$

$$\begin{aligned} \text{LR}(x) &= \frac{\text{IF}}{log(\sigma)}\sum_{i=1}^{n}\log(|x_i| + \text{IF}) \\ IF &= \frac{P_k}{\text{Mean absolute}} = \frac{\frac{1}{2}[\text{Max}(x_i) - \text{Mix}(x_i)]}{\frac{1}{n}\sum_{i=1}^{n}|x_i|} \\ \sigma(x) = m_2 &= \sqrt{\frac{1}{(n-1)}\sum_{i=1}^{n}(x_i-\bar{x})^2} \end{aligned} \tag{4.4}$$

$$\text{CF}(x) = \frac{P_k}{\text{RMS}} \tag{4.5}$$

$$\text{kurt}(x) = \frac{m_4}{{m_2}^2} \tag{4.6}$$

The skewness (equation 4.2), is defined for large samples, otherwise, the Fisher-Pearson adjusted standardized moment coefficient has to be used. Another metric very famous in

the literature is the RMS, which has not been selected as feature because both the SF and the CF contain it and, consequently, this could be a source of correlation.

In conjunction with the statistical metrics, the state vector is composed also of the wavelet contribution. The signal of a single sampling window is decomposed in wavelets and the energy of each result is taken as the final parameter. The decomposition will be better explained in Subchapter §4.3.1

The second and final challenge consists of choosing how the deviation between the actual state and the healthy state is achieved; also in this case different methodologies can be applied. The most important one provides to determine the distance between the vector which contains the healthy parameters and the vector which contains the actual health state. In fact, it is possible to treat the REB's parameters as a dimension of a traditional mathematical vector. In this work, the distances formulations compared are the Mahalanobis Distance (MD) and the Frechet Distance (FD). The former is very suitable because the MD is a multi-variate measure that computes the distance between a point and the mean of a group. In this sense, it is necessary that the healthy state is composed of a set of points, and the more points there are in this set, the better is the value of the HI. In any case, this clarification does not affect the gathering process: during the run-in phase, various sampling windows are taken as the baseline to compute the HI[1]. Its formulation is the following: given the observation vector $\mathbf{x} = (x_1, x_2, x_3, \cdots, x_n)$, the mean of the set of observation $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3, \cdots, \mu_n)$ and the covariance matrix $\mathbf{S}$, the MD is equal to equation 4.7.

$$\mathrm{MD}(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{S}^{-1} (\mathbf{x} - \boldsymbol{\mu})} \tag{4.7}$$

Regarding the FD, also this kind of similarity metric is suitable in this context. The idea behind the FD, with respect to the MD and the Euclidean Distance (ED), for instance, is that it takes into account also the ordering and the flow of the points, a typical case of a time-series dataset. Given to arbitrary non-decreasing curves A and B with $A(0) = B(0) = t_1$ and $A(1) = B(1) = t_n$ in a metric space S, the FD is defined as:

$$\mathrm{FD}(A, B) = \inf_{\alpha,\beta t \in [0,1]} \max \left\{ d\Big(A(\alpha(t)), B(\beta(t)) \Big) \right\} \tag{4.8}$$

where $d(\cdot, \cdot)$ is the ED. As for the MD, a lower FD correspond to a higher similarity between the curves A and B. Figure 4.1 reports the different shapes the HI assumes based on the sampling axis, the distance metric used, and the kind of features. The smoothing function is assumed to be the same for all the plots. A further consideration about the difference between the possible techniques that can be applied for this purpose is described more in detail in Subsection §4.1.1. First of all, referring to the function to which the HI has to be similar, i.e. the bathtub depicted in Figure 1.2, the most similar trend is the one of plot 4.1a which make use of the horizontal axis and the MD as deviation metric. Figure 4.1b is based on the vertical axis, which is quite different from the previous one because

[1] As said before, the same is valid after the ending of the run-in phase.

it has a more impulsive growth near the end-life of the component and the run-in is not much evident. Regarding always this initial phase, Figure 4.1c and 4.1d have a starting position that is near to zero, and this is due to the tri-cubic weight function applied at the first samples. In this case, neither the axis nor the chosen parameters can change the situation. In addition, the original non-smoothed signal is an indication of how good is the natural deviation between one point and the health conditions. As it is logical to expect, also Figure 4.1a and 4.1b have a non-smoothed unstable signal, but in those cases, it is possible to extract a general indication of trend. In Figure 4.1d, the smoothing technique resolves the highly impulsive original signal, demonstrating the importance of this post-processing method. Subsequently, regarding the kind of parameters used, each subplot of Figure 4.1 depicts a comparison between a parameter's vector composed only of statistical measures, of wavelet features, and finally of the concatenation of the former two. Here the spectrum of possibilities is more narrowed down to few situations. In general, the best set of parameters is the union between statistical and wavelet. This is plainly visible in plot 4.1a, where the statistical measures perform pretty badly, especially in the run-in phase, and the wavelet ones do not give a good estimation of the REB's degradation. Instead, the conjunction between the latter two is very close to a theoretical bathtub curve. Moreover, these considerations are valid also for the other cases, taking into consideration that, using the FD, there are no differences between the parameters with only statistical measures and the ones with wavelets and statistical.

4.1.1 Smoothing

The vibration signal is completely unstable. Even if it is possible to extract from such time-series a generally increasing trend, there are a lot of fluctuations also choosing a narrow interval of the x-axis. This completely non-linear behavior leads to a subsequent unstable and non-linear RUL prediction. In order to overcome this problem, it is possible to apply after the computation of the HI a smoothing technique that tries to harmonize the curve. The methods analyzed in this work are two: the moving average and the LOESS filter.

Starting from the former there exist different methods to compute the moving average: the simple, the cumulative, the weighted, and the exponential moving average. However, for each of them, the basic principle is the same: given the actual time t, the values between $t - w$ and t are taken in order to compute the average, where w is the size of the window. The simple average makes simply the traditional arithmetic mean over the w values, the cumulative provides a value of w equal to the number of samples n, the weighted assigns a linear decreasing weight start for each time starting from t to $t-w$ and finally the exponential is equal to the weighted except for using an exponential function for the weight decrease. The kind of moving average used in this project is the Weighted Moving Average (WMA), which is implemented as a convolution of the time series vector. The problem arises from the fact that formally it is not possible to compute the average of the samples with a time index t lower than the window w. For this reason, the input vector is left-side padded with $w - 1$ numbers equal to the time sample t_0. The mathematical formulation is reported in equation 4.9.

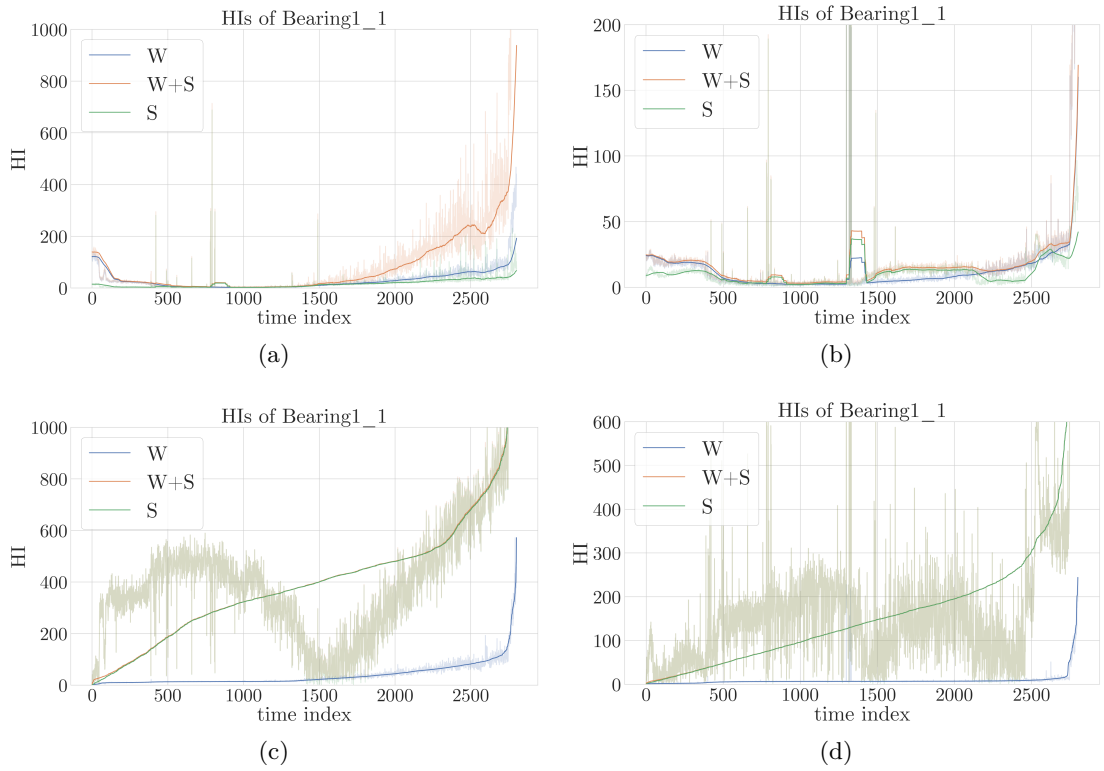


Figure 4.1: Different versions of the HI, based on the parameters, the vibrational axis, and the distance metric. HIs of subplots (a) and (c) are computed on the horizontal axis, instead (b) and (d) on the vertical one. Regarding the distance, (a) and (b) make use of the MD, instead (c) and (d) the FD. There are three possible versions of parameters: only statistical, only wavelet, and statistical plus wavelet. The clouded signals under the colored curves are the HIs before the smoothing with the WMA.

$$\mathrm{WMA}_t = \frac{wx_t + (w-1)x_{t-1} + (w-2)x_{t_2} + \cdots + x_{t-w+1}}{w + (w-1) + (w-2) + \cdots + 1} \tag{4.9}$$

$$w(x) = \begin{cases} (1 - |x|^3)^3 & \text{if } |x| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

The LOESS filter is instead a particular smoothing technique that makes use of linear regression. The basic idea is the following: starting the set of observations $\mathbf{x}$, a subset of the nearest k elements is taken by computing the ED[2]. Then, at each neighbor is assigned a weight using the tri-cubic function, here reported in equation 4.11. Finally,

[2]The same procedure for selecting the k-nearest neighbors in the KNN supervised machine learning algorithm.

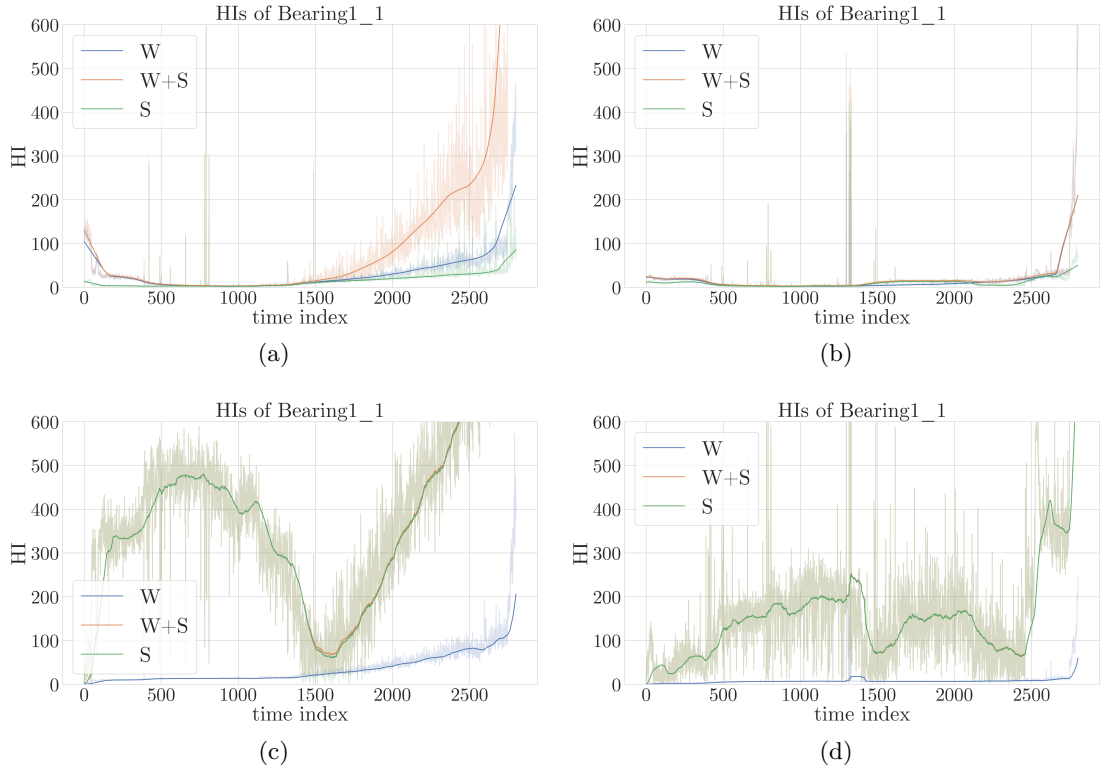


Figure 4.2: Same plot of Figure 4.1 but here the clouded signals under the colored curves are the HIs before the smoothing with the LOESS filter.

using the k neighbors of $\mathbf{x}$ and the weighted function, the response variable $\hat{y}$ can be fount out by using the weighted least-squared estimate with relation 4.12. The substantial difference between a classic linear regression algorithm and the weighted version is that the former requires the hypothesis of homoscedasticity[3], instead the latter, due to the tri-cubic weighted function, violates this assumption in favor of the heteroscedasticity.

$$w(x) = \begin{cases} (1 - |x|^3)^3 & \text{if } |x| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

[3]The errors of the regression are assumed to have a constant variance.

$$
\begin{aligned}
&\mathbf{Y} = \beta_0 + \beta_1 \mathbf{X} + \epsilon \\
&\min \sum_{i=1}^{k} w_i (y_i - \beta_0 - \beta_1 x_i)^2 \\
&\hat{\beta}_0 = \bar{y_w} - \hat{\beta}_1 \bar{x_w} \quad \hat{\beta}_1 = \frac{\sum_{i=1}^{k} w_i (x_i - \bar{x_w})(y_i - \bar{y_w})}{\sum_{i=1}^{k} w_i (x_i - \bar{x_w})^2} \\
&\bar{x_w} = \frac{\sum_{i=1}^{k} w_i x_i}{x_i} \quad \bar{y_w} = \frac{\sum_{i=1}^{k} w_i y_i}{y_i}
\end{aligned}
\tag{4.12}
$$

A comparison between the WMA and the LOESS filter is depicted in Figures 4.1 and 4.2. The first important difference, which becomes useful in the fitting section (§4.2), is the insensitivity to spikes guaranteed by the LOESS filter. The MD, Figure 4.1a, 4.1b, 4.2a and 4.2b, generates two HI impulses at around time index 800 and while the WMA reacts to those spikes with an increasing HI, the function smoothed by the LOESS filter does not present any difference. This is also valid for the monotonicity variations, like the one between 2500 and 2700 of Figure 4.1a and 4.2a, which are taken into account by the WMA and are discarded by the LOESS filter. On the other hand, if the general trend is fluctuating, like Figure 4.1c, 4.1d, 4.2c and 4.2d, WMA is more able to extract a general trend, despite LOESS becomes more sensitive even if, as already discussed in Subchapter §4.1, the aforementioned figures are not perfectly representing the health status of a machine and, suddenly, these cons are limited in importance.

4.1.2 FPT

One of the ways to split the entire life of a REB into two HSs is the FPT. The FPT is a measure used to decree if, at a certain time, the bearing is in a breaking phase or not. From another perspective, the FPT assumes also a different important meaning: the boundary between the diagnosis and the prognosis phase. When the REB's vibrations are not extremely high, i.e. during the run-in and the useful-life phases, it is possible to monitor the presence or not of a fault, since normally at the beginning of the wear-out phase only one kind of break occurs. However, after some time intervals, it will be this failure to generate other spalls and to degrade hopelessly the component and, in this situation, it is not important which kind of break is inside the bearing, but how much is the remaining useful life of the component. For this reason, some research papers indicate the results of the prediction only after the FPT.[4].

There are possible ways to compute this kind of indicator, but the simplest one makes use of kurtosis. Given the time actual time t and the previous times $t-1$, $t-2$, the kurtosis' mean, μ_k, and the kurtosis' standard deviation, σ_k, of all the sampling windows from 0 to $t-3$ are computed. Then if equations 4.13 are all true, the time index t correspond to the FPT.

[4]The approach followed for this project is to predict the overall bearing's RUL, starting from the first samples to the end life of the component.

$$|\mathrm{k}(t) - \mu_k| > 3\sigma_k \wedge |\mathrm{k}(t-1) - \mu_k| > 3\sigma_k \wedge |\mathrm{k}(t-2) - \mu_k| > 3\sigma_k \tag{4.13}$$

The empirical form of this indicator is really suggestive of the goal the FPT has to reach. The image of the vibration signal is quite limited and when the bearing is degrading the codomain of this function changes rapidly. Then, if for a series of samplings, the magnitude of the vibration signal exceeds the *normal* values reached right over, it is quite certain that the bearing is in the wear-out phase and the countdown to the complete break can start.

4.2 Curve Fitting

The bathtub curve can be mathematically represented by the sum of two exponential functions with two different sets of parameters. The first part starts from the run-in phase and continuously decreasing to zero; the second part starts from near zero in the run-in phase and increases over time. The decaying speed of these two exponentials is always a function of the value t, such that the bathtub reaches its minimum value. Thus, in mathematical terms, the HI has the form:

$$\mathrm{HI}(t) = ae^{bx} + ce^{dx} \quad \text{with } a, b, c, d \in \mathbb{R} \wedge b < 0 \tag{4.14}$$

A possible further improvement could lie in the addition of a translation term in the second exponential, since it comes to grow only at the end of the plot. Consequently, to reach the same result without the translation, the parameter d has to assume high values which then spoil the wear-out trend, making this function more similar to a binary signal.

After this first introduction about the mathematical description of the bathtub curve, the ways which exploit this property are now described. Given the HI of a bearing and using the relation 4.14, it is possible to find the four parameters a, b, c and d which allow to transform the real trend of the bearing into a mathematical model of it. In a broad sense, this procedure could embody the basic principles of the digital twin, which aims to replicate the real functioning of machinery with a set of ordinary differential equations. In doing so, it is possible to simulate future scenarios by exploiting the *computational* version of the industrial machine under analysis[5]. Once a, b, c and d have been obtained, through the inverse form of equation 4.14, it is possible to compute when the bearing will reach the failure. The value of the HI which indicates the total break, namely the failure threshold FT, can be calculated using the training set and different but similar approaches, which consists of agglomerating the single FTs of the training REBs. Moreover, the threshold is assumed to be the value of the HI when the complete failure occurs.

[5]It is anyway necessary that the mathematical model follows the real object by constantly and continuously upgrading of the equations' parameters.

$$\begin{aligned} \mathrm{FT} &= \min_i \{\mathrm{FT}(i)\} \text{ with } i \in 1, 2, \cdots, n_{\mathrm{TS}} \\ \mathrm{FT} &= \frac{1}{n_{\mathrm{TS}}} \sum_{i=1}^{n_{\mathrm{TS}}} \mathrm{FT}(i) \end{aligned} \tag{4.15}$$

$$\begin{aligned} \mathrm{RUL}_k &= \inf \{l : \mathrm{HI}(l + t_k) \geq \mathrm{FT}\} = t_{\mathrm{EOL}} - t_k \\ &= \frac{1}{d} \ln \frac{\mathrm{FT}}{c} - t_k \end{aligned} \tag{4.16}$$

The possible ways to compute the FT are reported in equations 4.15 and they generally provide the minimum or the average of the n_{TS} bearings, where n_{TS} is the number of training bearings. Then, relation 4.16 reports the inverse form of 4.14 using only the contribution of the second exponential. This choice depends on two factors: the first one is the simplification of the RUL expression and the second is the FPT. As described in Subsection §4.1.2, the moment which decrees the end of the useful life of the bearing and the contextual beginning of the wear-out phase is the FPT. If the signal $s(t)$ is limited in the range $[\mathrm{FPT}, +\infty)$ the correspondent HI can be represented with only one function: an increasing exponential function. Therefore, the value of RUL at time t_k is the difference between the time at which the bearing, given the actual conditions, will reach the FT minus the actual time t_k. This formulation is the same as taking the value l as the infimum of the set of $\mathrm{HI}(l+t_k)$ which are greater than the FT. The specification *given the actual conditions* of the previous sentence has a high weight. The parameters c and d are better as time goes on. Thus, the RUL predicted at a time t equal to the value of the FPT will not be equivalent to the RUL predicted far away from this time. Under this aspect, the curve fitting is a mathematical follower of the real HI trend and it tends to retrieve the parameters of this function having only a part of it. The possibilities analyzed in this work to reach this goal are two: the NLS and the Kalman Filter.

4.2.1 Non-Linear Least Squares

The non-linear LS is a particular form of the least-squares method which aims to find n unknown parameters of a non-linear model. Given the set of data points $((x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m))$, the non-linear function $y = f(x, \Theta)$ and the set of unknown parameters $\Theta = (\Theta_1, \Theta_2, \cdots, \Theta_n)$, the non-linear least squares (NLS) can be expressed with the following optimization problem:

$$\min \{\sum_{i=1}^{m} (\hat{f}(x_i, \Theta) - y_i)^2\} \tag{4.17}$$

In our case, $f(x, \Theta) = \Theta_1 e^{\Theta_2 x}$. As introduced before, the overall fitting is extremely dependent on the time at which the estimation is done, for different reasons. The first one is due to the translation: the HI of a time index far away from the end-life of the component is characterized by a very gentle slope. As a consequence, the parameters c

and d will replicate this behavior, by predicting an RUL completely different from the real one. The second problem arises from the not perfectly monotonicity of the HI. Even if the global trend of the function is very close to an exponential, it is possible that are present some inflections. The most suitable example is the HI depicted in Figure 4.1a, which is characterized by a trend practically identical to the exponential function, the best of all the combinations of parameters that define the health index. However, between the sampling windows number 2500 and 2700, there is a valley that is deceptive for the application of the NLE. One possible solution to this problem is the execution of other post-processing techniques or the quicker increment of the size of the moving average window.

Figure 4.3 summarizes the different situations generated by the non-linear least square estimates applied to different kinds of smoothed HI with different parameters values. Figure 4.3a and 4.3b illustrate the predicted exponential curves in the case of the FPT is fixed respectively at time index 2000 and 2500 and the RUL analysis is done at sampling windows 2350, 2700, and 2800. Instead, Figure 4.3c and 4.3d make use of an HI smoothed with the LOESS filter. What is evident is that the narrow fluctuation of monotonicity created by the application of the WMA could have two negative sides. The first one is that it can deviate from the application of the NLS algorithm, which tries to always minimize the distance between the fitted curve and the original one, also when the HI is returning to zero. This means that the fitted curve, especially in cases in which the trend is much impulsive, is distant from the original one, leading to a final loss of prediction accuracy, as it is possible to notice in Figure 4.3a. The second is more disastrous than the previous one because if the prediction is made in any of the points where the first derivative of the HI is negative, the resulting RUL is infinity because a decreasing exponential will not intersect with the FT. It is always true that in this case is possible to apply a post-processing technique that tries to solve this problem, but this can not hide the lack of performance demonstrated by the fitting. A comparison with the HI smoothed with the LOESS filter is shown in Figure 4.3d, where the range of time indexes affected by a fluctuation at most varies the final prediction, without returning not physical outcomes. Finally, it has to be taken into account also the different values of the FPT. The value of 2000 is the natural FPT, i.e. the one calculated with equation 4.13, for Bearing1_1, but anyway it is possible to make a comparison with a possible other threshold fixed at sampling window equal to 2500. Even if in the latter case the overall prediction is better, the proximity of the FPT to the analysis points leads to the possibility of estimating an RUL completely wrong.

In light of this, the trend of the HI and the FPT completely influences the success of this method, which is however susceptible to many factors that make the non-linear least square estimation only a starting point for the bearing prognosis.

4.2.2 Kalman Filter

The Kalman Filter is a recursive algorithm that is in general able to determine the state of a system through noisy measurements. The analysis of the analogic reality given by the sensors is not always reliable, due to the uncertainty generated by each element of measure. In these cases is convenient to combine, using the sensor fusion technique, a set of singular untrustworthy and inaccurate measures, in order to obtain an outcome that is better than the lonely contributions. The empirical reason behind this formulation is that assuming

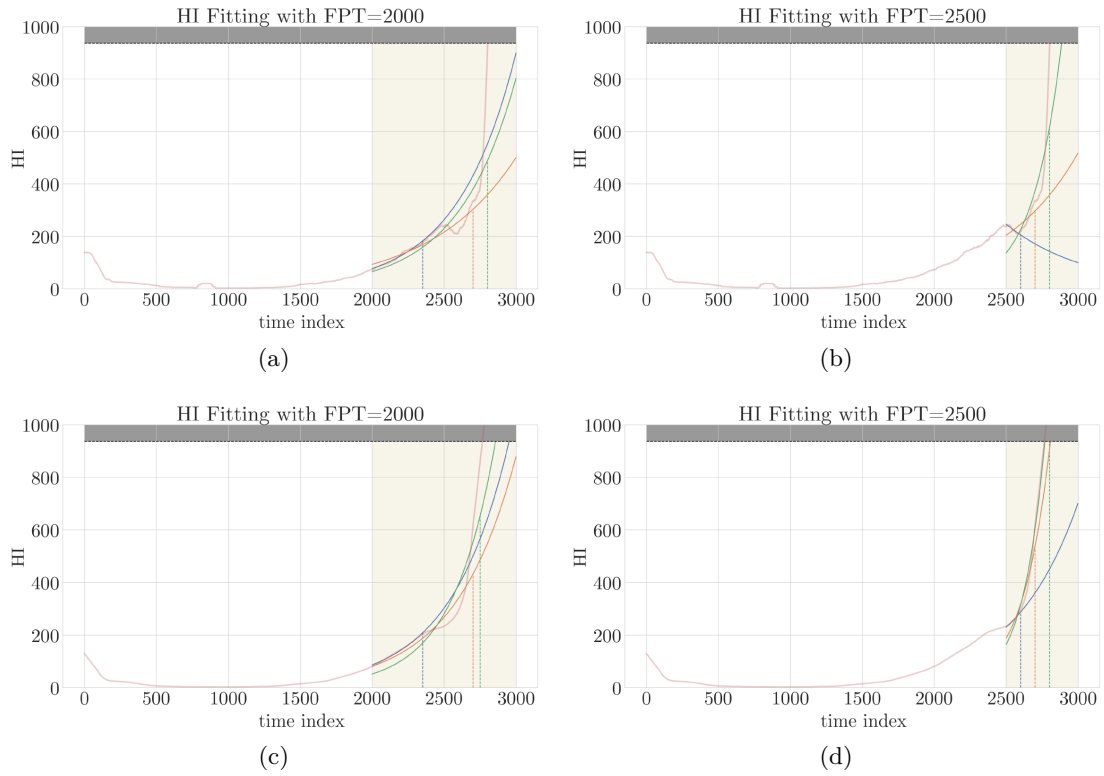


Figure 4.3: Non-linear least square estimates applied to Bearing1_1 using the smoothed HI with two different methods: (a) and (b) with the WMA, (c) and (d) with the LOESS. The FPT, illustrated here as the left side of the shaded light rectangle, is fixed at 2000 for the subplots (a) and (c), and at 2500 for the subplots (b) and (d). The different predicting moments are samples number 2350, 2700, and 2750 for (a) and (c) and 2600, 2700 and 2800 for (b) and (d). The grey zone at the top of the plots represents the FT.

that each sensor reading is a random variable expressible through normal distribution, even with high variance, the product of each measure leads to another gaussian with much less variance. In doing so, the noise which was encapsulated in the readings is removed. This kind of algorithm is useful in particular in this kind of application. The measurements represent the value of the HI computed at the instant time k and the outcomes are the parameters of the exponential function, i.e. the RUL prediction. The design of the filter is the most important step. This phase is based on a series of steps which begin with the definition of the system state, i.e. the state variables. The second step is the definition of the matrices $\mathbf{F}$ and $\mathbf{H}$ which respectively allows determining the next state $\bar{x}$ based on the previous state x and to determine the value of the output z basing on the value of the state vector x. Then, iteratively, for each new incoming reading, the difference between the expected output and the latter determines the estimated state $\hat{x}$. Mathematically speaking, the state vector and the measurement vector at time instance k are equal to:

$$\mathbf{x}_k = \begin{bmatrix} c_k e^{d_k k} & c_k & d_k \end{bmatrix}^T; \quad \mathbf{z}_k = h(\mathbf{x}_k) = \mathbf{x}_k(1) \tag{4.18}$$

$\mathbf{x}_k$ contains the value of the HI at time k and the parameters c_k and d_k which are the final outcome of this filter. The transferring function F is exponential, thus the linear Kalman filter cannot be used. However its extended version (EKF) is suitable for non-linear cases like this and it is based on the determination of the Jacobian matrix of $\mathbf{F}$ and $\mathbf{H}$, as reported by the following equations:

$$\mathbf{F}_k = \left.\frac{\partial f}{\partial x}\right|_{\mathbf{x}_k} = \begin{bmatrix} 0 & e^{d_k k} & k c_k e^{d_k k} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad \mathbf{H}_k = \left.\frac{\partial h}{\partial x}\right|_{\mathbf{x}_k} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{4.19}$$

As the function f is the HI, its partial derivatives with respect to the system state are the ones reported in the matrix $\mathbf{F}$. The same is valid for $\mathbf{H}$, which has only a 1 in the first element as the $\mathbf{z}_k = \mathbf{x}_k(1)$. Finally, the value of the estimated state $\hat{x}$ can be determined by this set of matrix products:

$$\begin{aligned} \mathbf{M}_k &= \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1} + \mathbf{Q}_{k-1} \\ \mathbf{K}_k &= \mathbf{M}_k\mathbf{H}_k^T(\mathbf{H}_k\mathbf{M}_k\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{x} &\leftarrow \hat{x} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{x}) \end{aligned} \tag{4.20}$$

The matrices $\mathbf{Q}$ and $\mathbf{R}$ represent the noise of the system and the measurements. The matrix $\mathbf{K}$ is the Kalman Gain, which allows to ponderate the state upgrade based also on the difference between the real and the estimated reading. In conclusion, the Kalman Filter is a good alternative to progressively follow the real HI, by continually refining the value of the parameters c and d.

4.3 Machine Learning Approaches

This chapter is completely dedicated to the prediction of the remaining useful life of a REB, using machine learning techniques. The possible methods that can be applied in order to estimate the RUL are mentioned in Section §1.3 and they cover a wide spectrum of possibilities. In the case of this work, the statistical methods are not used because the original idea of the author was to compare *plug-n-play* and *ready-to-implement* approaches which do not require sophisticated data mining competencies, like the curve fitting, with more complex ones like SVMs or ANNs.

The following chapter will be organized as follows: the first part will be dedicated to the extraction of the features used to train the machine learning models. Then two words are spent about the features selection phase and finally, one paragraph for each machine learning algorithm will be present. The purpose of this chapter is to present all the methodologies and the variances which involve the data-driven methodologies. The final results will be shown in Chapter §5.

4.3.1 Features Extraction

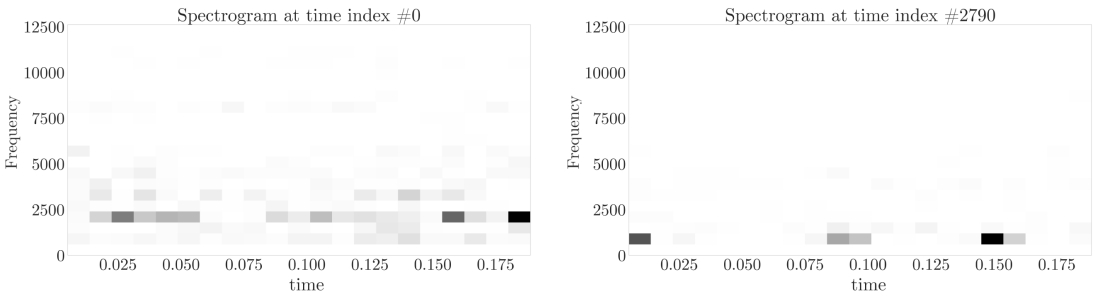
In general, the features that can be extracted by a signal belong to four different domains: the time-domain, the frequency-domain, the time-frequency domain, and finally the wavelet-domain. It is possible to assert, for what is described in Subchapter §4.2, that the HI is a good estimator of the status of a machine. However, it is critical to define with precision the value of the breaking failure, due to the well-known problems related to the high variance which present the bearings estimated life, namely L_{10}. For this reason, it is natural to overcome the limits of the previous approach by improving such solution with redesigned predictors and methods. Before starting the description of the different methodologies, it is important to remember that the signal, before the feature extrapolation phase, is demodulated with the wavelet denoising which enhances the contribution of all the prognosis trends contained in the data.

The features extracted for the machine learning models come from different domains. In particular, the time domain is well-represented by the statistical metrics described in Subchapter §4.1: mean, skewness, crest factor, shape factor, log-ratio, and kurtosis. Regarding the frequency-domain, the most popular approach consists of applying the Fourier transform of the original signal, splitting the obtained results into a set of bins[6] and use these as final features. However, since one of the goals of this thesis is to apply diagnostic techniques in the prognosis context, the frequency domain, in this case, is the envelope spectrum. Thus, in a broad sense, also this feature makes use of the spectrum but the difference lies in the application of the Hilbert transform to the signal before it is passed to the FFT algorithm, which changes completely the final outcome. As the signal is demodulated before the envelope spectrum, the result of the envelope spectrum is the one used in diagnosis to determine if a REB present an inner race, an outer race, or a ball fault. Even if one of the notes which are specified in the paper of the FEMTO Institute which describe PRONOSTIA is that it is not possible to apply diagnostic techniques because multiple faults can occur simultaneously. If this sentence is literally interpreted the envelope spectrum could become useless for this analysis. It is indeed true that always referring to the envelope spectrum, the harmonics, which can be used to assert if a specific fault is present inside the REB, could be all greater than zero, a synonym of the contemporaneity of different kinds of defects. On the other hand, it is also true in this case, contrarily to the diagnosis, it is not important which harmonics are greater than zero, but it is necessary to pay attention to the magnitude of these frequencies. Figure 4.6 illustrates that in the initial run-in phase there are not relevant CDFs in the region $[0, 300]$ Hz, which the first multiples of the BPFOs, BPFIs, and BSFs should be present. Nevertheless, as time goes on, in the last part of the useful life it is possible to notice some groups of harmonics which starting to have a considerable amplitude. This situation comes back amplified in the wear-out phase, where it is possible to detect without a doubt a set of CDFs. In fact, it is possible to surely detect five harmonics in the interval $[0, 250]$ Hz which probably are not multiples of the same kind of CDF. It is logical to expect the first

[6]This passage is done because, normally, the shape of the FFT is too high and the correspondent features set could lead to a final unprecise regression.

multiple of a BPFO, or a BPFI, around 125 Hz and, thus, this high number of harmonics depends on the fact that, as described in Subchapter §3.2, this REB is characterized by the simultaneous presence of multiple faults, due to the extreme operative conditions. Consequently, it is evident that it is not important if it is not possible to assign univocally a kind of failure, which is the goal of the diagnosis; what has to be considered is that these kinds of features could bring improvements to the RUL prediction. In fact, the translation from the envelope spectrum to features is managed by the subdivision of the interval $[0, 300]$ Hz into 6 bins and, for each of them, an aggregation metric is used in order to obtain 6 values. The creation of such frequency intervals depends on the limitations generated by the design of a general solution. The most accurate results could be achieved by specifying the values of the CDF or, equivalently, the manufacturing parameters of the REBs involved in the analysis. However, in a complex industrial real scenario, it is unthinkable to parametrize the predictive maintenance tool with all the bearings under control: medium-sized machinery could contain hundreds, or even thousands, of bearings. As a result, the choice of splitting the envelope spectrum in frequency bins represents the tradeoff between a full-specialized scenario and a generalized predictive maintenance tool.

The time-frequency domain is responsible for describing the variability of the spectrum with respect to time. Differently from the envelope spectrum, in this case, the pure FFT is computed. The outcome of this extrapolation is the spectrogram, which is substantially a matrix such that the columns represent the time dimension and the rows the frequency dimension. Unfortunately, as introduced before, a set of bins must be created in order to have an appropriate number of resulting features. Thus, the number of intervals used in this case is 10. Figure 4.4a and 4.4b depicts respectively the spectrogram computed on the signal of the first sampling window and on the signal of window number 2790[7]. The main difference between them is the diverse concentration of high magnitude harmonics, which are more distributed in the former and more concentrated in the latter, principally due to the presence of the CDFs.



(a) (b)

Figure 4.4: Spectrograms of Bearing 1_1 at time index 0 (a) and time index 2790 (b).

[7]In this case, for exposition reasons, the number of total bins used to divide the frequency range is 21.

The same analysis made for the envelope spectrum can be done also for the time-frequency domain, i.e. it is possible to highlight the difference between a spectrogram referred to an initial sample window and a spectrogram referred to the samples gathered during the end life of the component. Even in this case, it is possible to find out a set of frequencies that change their value and, consequently, which are a possible indicator of the RUL.

The last domain is the wavelet. Formally, this is not a perfect mathematical domain, even if wavelet decomposition is a procedure that is completely different with respect to those introduced before. The limitation of the Fourier transform is that it is not able to capture the information of a signal which has a perturbation confined in time, for instance, an ECG signal[8]. On the other side, the wavelet transform decomposes the original signal into a set of wavelets, by determining the way in which a single wavelet, in terms of location and scale, is contained in the original signal. The number of wavelets that can be used is enormous and the major advantage of using this decomposition is that it is possible to extract local temporal and spectral information. Then, as feature, the total energy of each result is taken. Consequently, with a decomposition depth of 2, the total number of signals is 4 as the total number of added features. Finally, the last value added to the extraction set is the HI computed in Section §4.1, which is recycled also in this part of the work.

The three previous paragraphs could lead the reader to think that the information extracted from the signal has some problems. First of all the frequency-domain and the time-frequency domain are more or less the same. In other words, it seems that the information gain obtained by the former is equal to the one obtained by the latter. In a certain sense, both methods indeed extract knowledge from the magnitude of different harmonics in the spectrum. However, it is also true that the spectrogram evaluates the behavior of a bearing even inside a single sampling window and the frequency intervals targeted by the envelope spectrum are different from the ones taken into account by the time-frequency analysis. Secondly, another possible interpretation is that the ML feature extraction phase is a false copy of the HI, even more, if it is used also in this section as feature. Normally, machine learning models reject the presence of correlated features and, on the other hand, it seems that this phase adds to the results achieved by the HI only some spectral information. First of all the HI is computed with metrics that are not strongly correlated to the parameters which compose its definition. For instance, the MD makes use also of the variance-covariance matrix and its value is an indication of the deviation of one sample, in the view of this paragraph, with another set of healthy samples present in the feature matrix. Thus, in summary, the correlation is weak and the information gain given by the HI is a sort of link between the actual observation and another one characterized by a health condition. It is clear that also the machine learning model is able, by definition, to retrieve this kind of pattern in the data, but how the HI is computed can improve the final result. Regarding the difference between the HI parameters and the feature extraction phase, it is evident that the formers are a subset of the latter because the HI is a general estimator of the status of a machine and it will

[8]This is the most used example for explaining the power of wavelets.

be something weird to overload the health parameters with also the frequency and time-frequency domains. This kind of implementation is completely opposed to the benefits of the aforementioned approach. It is important to remember that the HI construction can lead to good results in terms of final accuracy but, in general, its usage is more suitable with statistical approaches or for a general and immediate *plug-n-play* estimator of the status of the machine. More complex data-driven prediction algorithms could not rely only on this kind of indicator. Finally, as a general rule, if the concatenation of those features leads to degrading the model, it will be the task of the feature selection part to discover this situation by underlying which features are better to discard. Figure 4.5 shows the values assumed by the different features. To obtain a discrete histogram, the possible values are split into ten intervals. Some of these variables, like E1, E5, E6, S8, S9, and S10 present a particular situation because practically only one of the total number of possible columns is much greater than zero. However, with this kind of precision, it is impossible to give a judgment on the utility of these features, even more without additional information like the feature importances.

4.3.2 Features Selection

The set of features generated in the previous subsection (§4.3.1) could not represent optimally the degradation trend of a bearing and, contrarily, some of them could also inject noise in the prediction, decreasing the final accuracy. The goal of this part is to examine the feature matrix, to find the correlations and the importance of the single features, and to plan possible improvements to the features extraction phase.

Feature Correlations

The most immediate way to categorize each feature with its level of correlation implies the usage of the correlation matrix, here illustrated in Figure 4.7, and the metric used for this purpose is the Pearson correlation coefficient. In order to describe correctly all the information the correlation matrix can give, it is convenient to split this figure into several blocks equal to the number of different kinds of features extracted by the dataset: statistical, wavelet, envelope, spectrogram, and HI. Starting from the statistical block the only strong positive correlation present is between F3 (crest factor) and F6 (kurtosis) and the nearly negative strong correlation is between F2 (skew factor) and F6. The remaining features are or weakly or moderately correlated with each other. Thus, it seems that kurtosis could be dropped, even if total security is absent in doing this operation, which can be ensured only by the features importances analysis (Subsection §4.3.2). Regarding the wavelet coefficients, W2 and W3 are positive strongly correlated, as W1 and F5 (log-ratio). One situation which deserves to be discussed is the more or less constant correlation between W1 and all the features starting from E2 to HI, to the net of some slight exceptions for E6 and S8. Consequently, also in this case, W1 is one possible candidate feature to be removed. The envelope features are in a situation such that each spectrum interval is correlated to the other ones. A possible solution to this problem is to change the size of each interval and the starting and final position of interest in the envelope spectrum. This indication can be suggested by the value of the

Figure 4.5: Distribution of the values of the features extracted from the training dataset.

feature importance: the most relevant envelope features are a good estimator of the region more indicative of the spectrum. Then, in a neighborhood of such frequency intervals, it is possible to examine the width and the number of bins that have to be created. Spectrogram features are correlated with each other especially in the region of frequencies represented by S3-S5 and, also for this case, it is possible to re-modulate the frequency bins or to skip an entire region, based on the results of Subchapter §4.3.2. Finally, the HI is, as anticipated before, only weakly correlated with the metrics which has generated it, except for the F5 and W1 which are however redundant for what was described before. Regarding the other kind of domains, there are some strong correlations with some intervals of the envelope spectrum and of the spectrogram, which also in this case depends on the nested

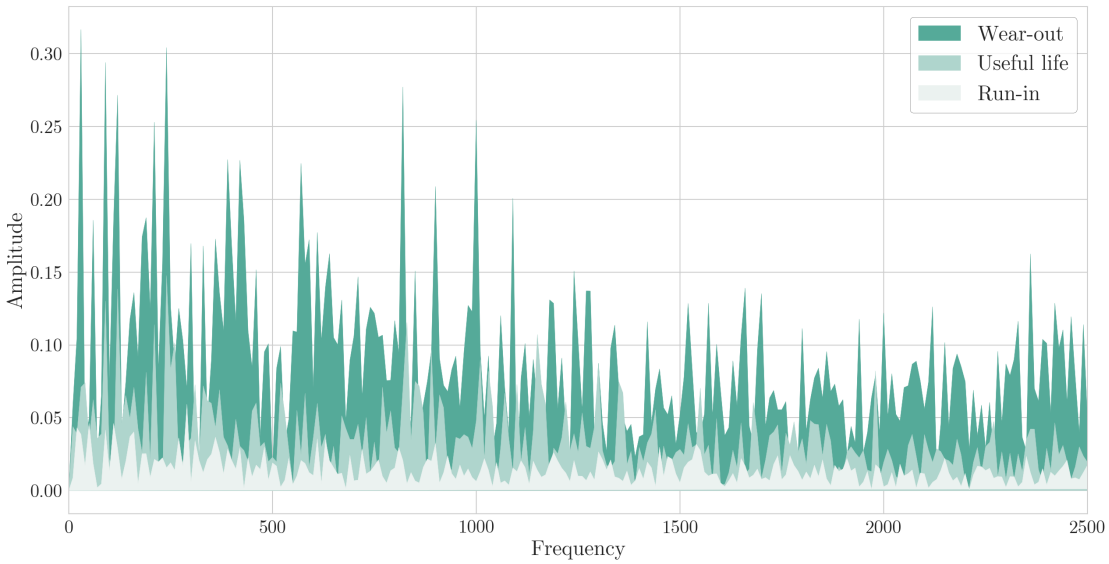


Figure 4.6: Envelope spectrum, limited to the interval $[0, 2500]$ Hz, of bearing Bearing1_1 in a healthy state, in the middle of its useful life and when a fault begins to appear. The time indexes are respectively 0, 1905, and 2693.

correlations analyzed previously.

Feature Importances

There exists multiple ways to compute the feature importances: fisher's score or chi-square test, wrapper methods which add or remove sequentially features to find the best subset, LASSO regularization and, the one used in this work, the random forest. The Random Forest is a bagging algorithm, which splits the total dataset into many groups. Then, each of them is used to train one tree and, consequently, it is possible to obtain the GINI indexes of each split. Finally, the importance value is computed starting from the mean and standard deviation of the accumulation of the impurity decrease within each tree.[9]. Table 4.1 contains the importances values, and the list of the strongly correlated predictors found in Subchapter §4.3.2, of the features extracted in Subchapter §4.3.1. In general, it is possible to decree that only a small subset of features, in particular W4, W1, W2, HI, S2 and F5, compete for the final prediction result, by reaching a cumulative importance of 0.968. The rest of the features have a total importance of 0.032. W4 is the predictor with the highest importance and it is responsible for 61% of the final prediction. W1 has an importance which is more or less 1/4 of W4 and, in addition, is correlated with a lot of other features: E2, F5, HI, and E3. The first three have little, but significant, importance; instead, E3 has a negligible one. For these reasons, E2, E3, HI, and F5 can be discarded as W1 represent the same information of them. The same speech can be done for S3 and

[9] A more detailed explanation about the Random Forest will be present in Subchapter §4.3.6.

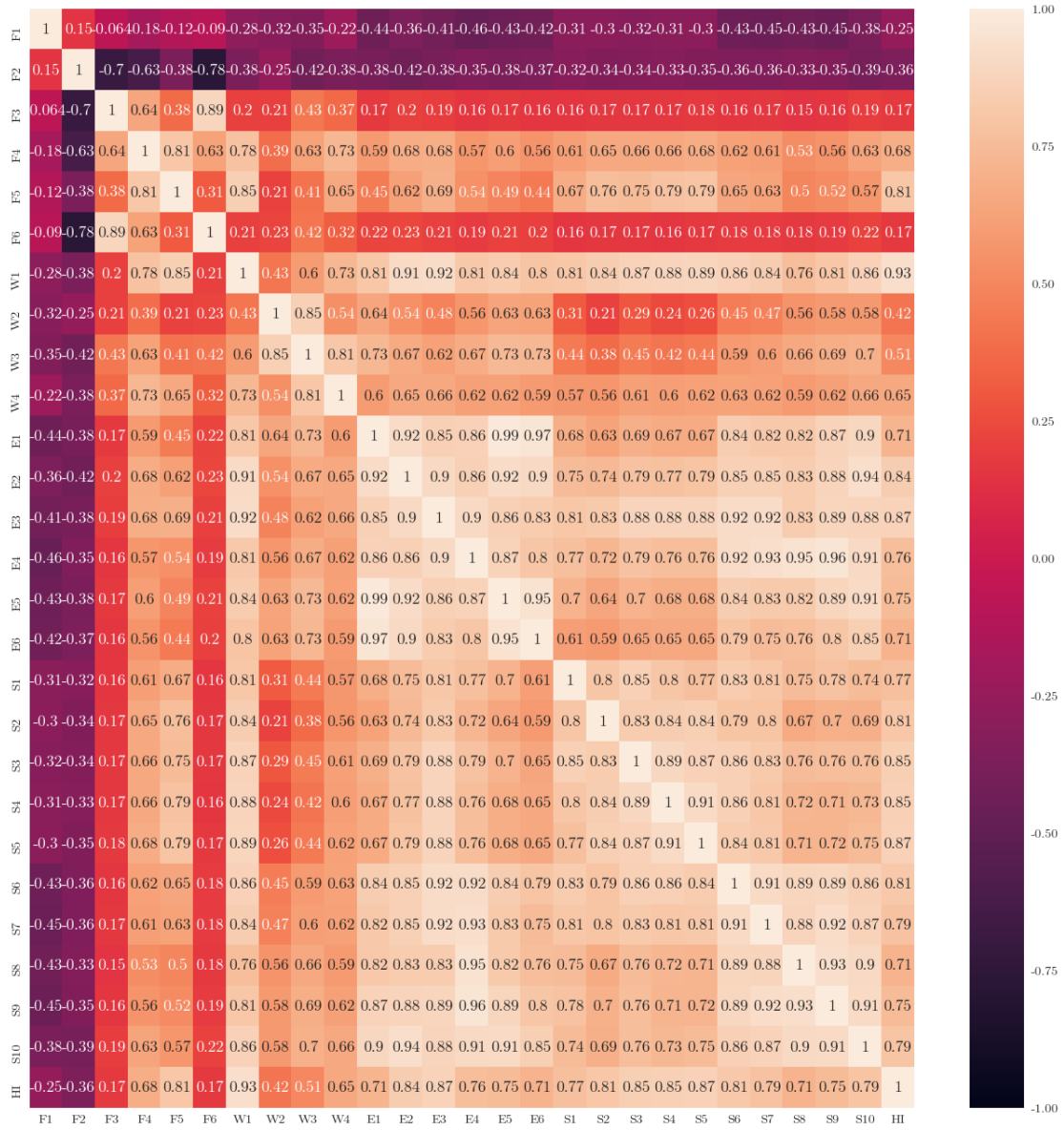


Figure 4.7: Correlation matrix of all the features pointed out in Subchapter §4.3.1.

S5. They are strongly correlated with S2, S3, and S4. Thus, the region of the frequency spectrogram seems that is already covered by S4, which is the feature with the highest importance. Features F6 is correlated with F2 and F6, so it can be removed for the same reasons explained before.

The information given by this table is an indication of possible improvements of the feature extraction phase. However, all these considerations have to be merged with the conclusion of the previous paragraph, a redesign of the intervals used to encode the overall

spectrum into a limited feature vector. The S2 and E4 are the most important *spectrum* features. Thus, it is logical to explode such intervals into two or three parts, in order to aggregate frequencies with a major finesse.

Feature	Domain	Importance	Strong Correlation	Risk
W4	Wavelet	0.6172		
W1	Wavelet	0.1361	E2, E3, HI, F5	
W2	Wavelet	0.1002		
HI	-	0.0535	W1	✓
S2	Time-Frequency	0.0458		
F5	Time	0.015	W1	✓
E4	Frequency	0.0056		
E2	Frequency	0.0035	W1	✓
E1	Frequency	0.0033		
S4	Time-Frequency	0.0027	S3, S5	
S3	Time-Frequency	0.0025	S2, S5	✓
S5	Time-Frequency	0.0016	S3, S4	✓
F2	Time	0.0016	F6	
E5	Frequency	0.0015		
W3	Wavelet	0.0012		
F4	Time	0.001		
F6	Time	0.001	F2, F3	✓
F3	Time	0.001	F6	
S8	Time-Frequency	0.001		
S6	Time-Frequency	0.0009		
E3	Frequency	0.0007	W1	✓
S10	Time-Frequency	0.0007		
S7	Time-Frequency	0.0006		
S9	Time-Frequency	0.0006		
E6	Frequency	0.0005		
F1	Time	0.0003		
S1	Time-Frequency	0.0002		

Table 4.1: Feature importances returned by the Random Forest.

4.3.3 Feature Aggregation

The vibration signal is a time-series dataset, as the order of each sample is due to the moment it is gathered. Given one time instant t and the features extracted from the t-th sampling window, the analysis described until here is only able to predict what those characteristics indicate. Therefore, the model is not able to take into account the data samples which precede it and that, in a certain way, which represent the history of the component. In order to handle this problem, a windowed dataset, with different

configurations, is generated starting from the set of n samples and m features. These three versions are the following: a horizontal concatenation and a long-term aggregation.

Basically, given the width of the time window w, the first one consists of concatenating to each row of the dataset the features of the immediately following w samples, using as response variable the value of the $w + 1$ sample. Figure 4.8 illustrates what mentioned above. This solution is memory-consuming, as the number of total features for each sample are then mw, and with a window equal to 5, m becomes more or less equal to 150. In doing so there are more cons than pros, because this huge memory exploitation has to be justified by a completely redesigned conception of predictive RUL. In other words, with this number of features, it is reasonable to expect that a lot of temporal dependencies are now taken into account. However, even with $w = 5$, the model could be at maximum able to predict the RUL with an *advance* of $10 \cdot 5 = 50$ seconds[10]. The long-term aggregation version makes use of time windows with tens of samples and this leads to anticipate for instance the forecast of $50 \cdot 10 = 500$ seconds, equal to 8 minutes, which starts to be a reasoning advantage. On the other hand, this example has the potential to generate an extremely high number of features. Thus, in order to avoid the horizontal concatenation and exploding the feature space, different kinds of aggregations are used, in such a manner that the total number of predictors is the same as the original one. Different functions have been tested: the average, the maximum, and the minimum. Consequently, for each kind of feature, the average, the maximum, and the minimum of the set of samples with cardinality equal to the window width w is computed. A possible improvement for this solution consists of aggregating the future with different metrics; then all of them are concatenated, as done for the first version. In this way, the possible total number of features is anyway high, but the extension of the time interval remains the same.

4.3.4 Prediction Pipeline

Machine learning and deep learning models are defined over a set of possible configuration parameters which could affect overwhelmingly the final outcomes. Consequently, in order to find the set of values that leads to the best result is necessary to exploit some well-known methods, as cross-validation with k-folds. The mechanism performs as follows. The dataset is firstly divided into two splits: training and test[11]. Then the training data is further split into several parts, normally 3 or 5: one of these is used for validating the performance instead the remaining are used to train the classifier with a specific set of parameters. Finally, an average of the performance metric is done and the process starts again with another set of parameters. The splits instead remain the same. The configuration which reached the best score is chosen and then an assessment is done on the test data. Cross-validation is helpful because, as shown in Figure 4.9, the split used to validate the model is different at each step. A blind subdivision of the dataset in only training and a test part lead to assessing the model with the same data at each time,

[10] 10s is the time between one time window and the following.

[11] The size of the test set is usually 20% or 25% of the total data.

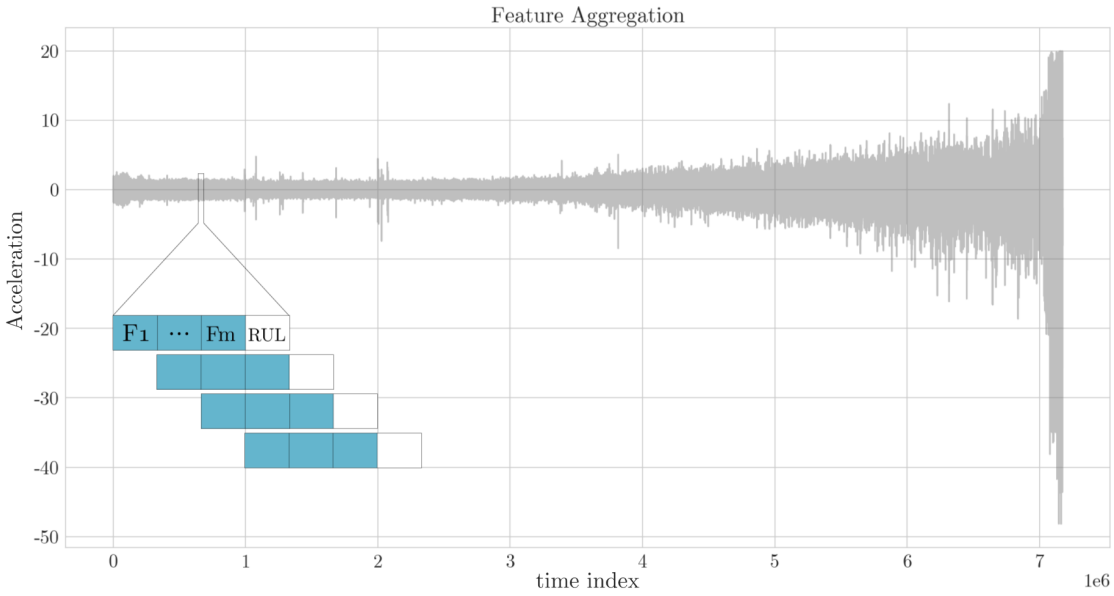


Figure 4.8: Windowed dataset for the Bearing 1_1. A single blue box represents the set of features of each sampling window and the white box the RUL of such time sample. In this example, the step size s is equal to 1 and the window width w is equal to 3.

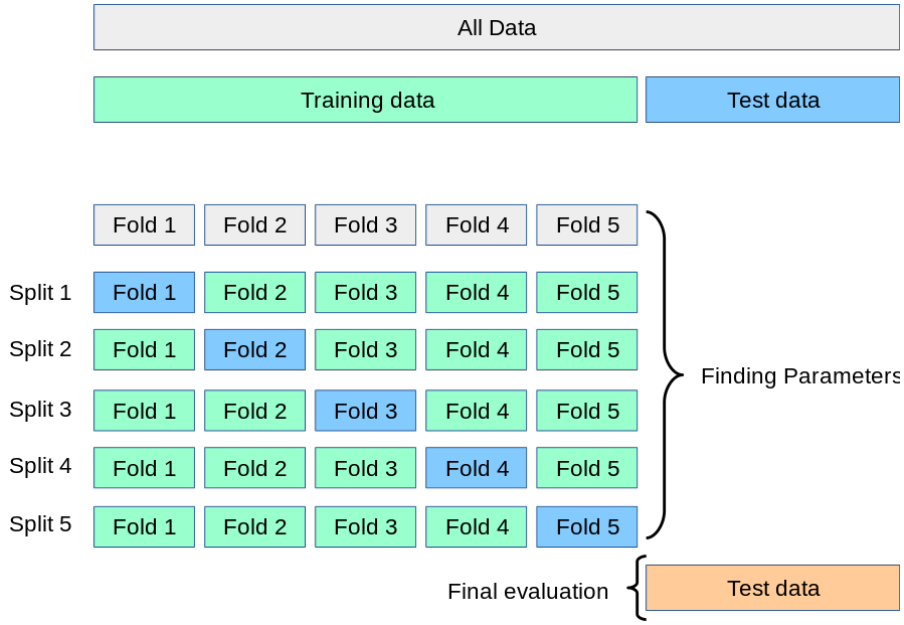


Figure 4.9: Functioning of cross-validation with k-fold, taken from the developers' guide of `sklearn`.

causing a lack of performance in terms of overfitting and underfitting. In fact, a classical tradeoff in machine learning is the one defined by the balance between bias and variance.

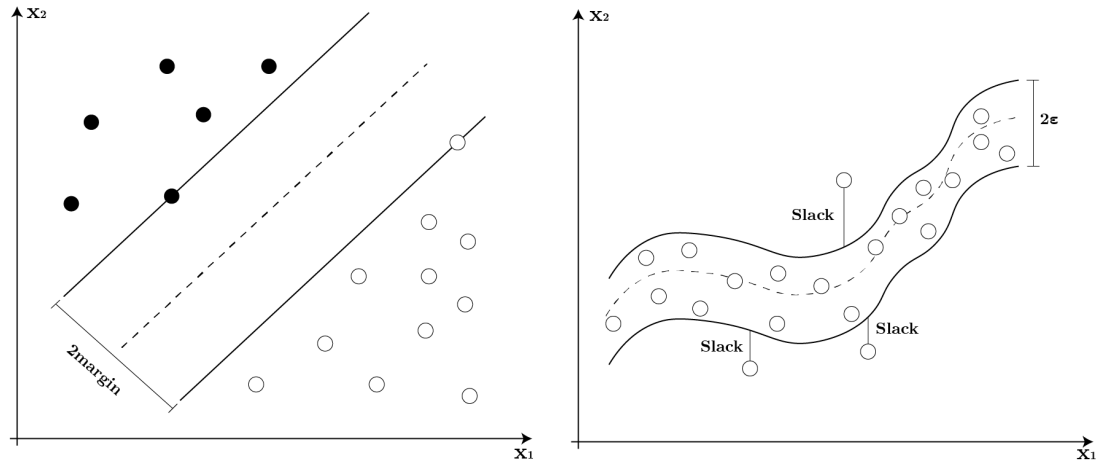
Bias is generated when are made wrong assumptions in the algorithm, i.e. when the model is oversimplified. The extreme case of high bias is underfitting: the condition in which the data-driven approach is unable to find a pattern in the dataset. Variance is present when the model is not able to generalize the assumptions made on the data and, in this case, can happen that the model tends to include also the noise in the algorithm. The high variance could lead to overfitting, which is due to train on much data or with few predictors. If the algorithm is characterized by a high number of parameters, i.e. important assumptions on data and the possible presence of noise, we are in a situation with low bias and high variance. However, the downside is having a low number of parameters, so high bias, and low variance. A possible solution is to find a balance between them, basing also on the specific case study.

In this particular work, the machine learning models will be tuned with a cross-validation method that makes use of 5 folds. Instead, about the deep learning models, it is unthinkable to train five times the same configuration, due to the time required to complete the procedure. A possible solution to this issue is to fix the size of the validation set, by using it as an estimator for the performance of a specific collection of parameters. This way does not apply the principles of k-folds, but it is however efficient to control overfitting and underfitting. From another point of view, bias and variance in deep learning can be managed by looking at the trend of the validation and the training loss. For instance, a validation loss that increased after a long-term decrease indicates overfitting. For these reasons, in this work, a cross-validation hyper-parameter tuning is implemented with a validation set equal to 15% of the total number of samples. In conclusion, a clarification about the RUL is a must. The useful life of bearings is characterized by high variance and this data science project has to be adapted to all the possible REB present in the market. In practice, training a regressor with a specific value of RUL could be misleading because, in general, the model has to recognize a decreasing life as quickly as the predictors grow. Thus, it could happen that, based on the starting conditions and the value of the entire life of the bearing, the same magnitude of a set of harmonics in the envelope spectrum is associated with many RULs. A possible way to limit this behavior is to treat as relative the response variable. More in detail, given the i-th bearing and a sample with time index equal to t, initially the RUL is computed as the distance between t and the total number of samples of bearing i, namely l. Then, this value is divided to l in order to obtain a % RUL. A further distinction has to be made between the training and the test set. The bearings of the training set as a total useful life equal to l, because it is assumed that their life ends at the final sample. Instead, the test bearings have an RUL different from zero, as requested by the problem. In this case, the previous denominator has to be added also the ground truth of each bearing provided by PRONOSTIA.

4.3.5 Support Vector Machine

The SVM is a classification and regression supervised algorithm that aims to separate the data maximizing the distance between the classes in the former fashion and confining the points inside the boundary in the latter. Even though this thesis is related to the

prediction of the RUL of a bearing, a general introduction about the functioning of an SVM in a classification configuration will be reported, in order to finally highlight the differences between the two kinds of problems.



(a) (b)

Figure 4.10: General function of the Support Vector Classifier (a) and the Support Vector Regressor (b).

In general, the SVM extends the cases which can be dealt with the Maximal Margin Classifier. The concept behind the Maximal Margin Classifier is to find a hyperplane that separates the data. Given a *p-dimensional* space, an hyperplane is a flat subspace of dimension $p-1$. The equation is very simple and it is the extension in dimension p of the line formula.

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p = 0 \tag{4.21}$$

Substantially, if the n samples $x_1, x_2, ..., x_n$ are well separated, there is an infinite number of hyperplanes that divide the data. This is the motivation behind the name Maximal Margin Classifier: the flat has to divide data points into two classes (in the case of a binary label) and maximize the distance between the two closest points which belong to the two clusters of data. Figure 4.10a illustrates how the margin, in the case of a 2-D classification problem, is defined as the line which makes a distinction in terms of data point labels. All these words could be translated into the resolution of the following optimization problem:

$$\begin{aligned} &\min_{w,b} \frac{1}{2}||w||^2 \\ &s.t\ y_i[< x_i, w > +b] \geq 1 \end{aligned} \tag{4.22}$$

In the previous expression, $y_i \in \{-1, +1\}$, so the maximum number of different classes is equal to 2. When this number is higher than this value, the task is cataloged as a

multi-class classification problem, so the *one vs rest* strategy represents the only solution to the issue. The aforementioned approach consists in making the distinction between one class and all the others. Then another distinction will be done in the group which contains the discarded classes and so on until the *rest group* is empty.

As already mentioned, this is valid if the set $S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ is linearly separable, and we know that there are some cases in which no separating hyperplanes exits and so it is impossible to define a Maximal Margin Classifier. The solution to this issue is to introduce the Support Vector Classifier (SVC), which simply allows some points to be classified in the wrong class. The way used to include this idea in the minimization problem is called C, and it represents the need to avoid misclassified points in the research. Thus, the problem becomes as described by the following formulas:

$$\begin{aligned} &\min_{w,b} \frac{1}{2}||w||^2 + C\sum_i \xi_i \\ &s.t\ y_i[< x_i, w > +b] \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \end{aligned} \tag{4.23}$$

High values of C, ideally $+\infty$, lead the classifier to choose hyperplanes with a tight margin. Instead, a low value of C leads the classifier to increment the margin, including in the classification also points that do not belong to the appropriate class. Assigning to the hyper-parameter C a small value can worsen also the accuracy of linearly separable problems. The linear separability is a very important characteristic because the layout of the data represents the reasoning behind the choice between the soft margin linear SVC classifier and the hard margin linear SVC classifier. The SVC, however, cannot solve all the possible problems because in some situations data is not separable even if several possible mismatches are allowed. The solution consists of adding one dimension to the shape of the input dataset and trying to separate the samples in such space. However, this solution is resource and time consuming because each sample $x_i \in \mathbb{R}^n$ has to be transformed into a sample $x_i' \in \mathbb{R}^{n+1}$. Then, the product $< x_i`, x_j` >$ is defined in $\mathbb{R}^{n+1}$. The kernel is a particular function that can be exploited to obtain the same results without making those consuming computations. In fact, it is defined as:

$$k(x_i, x_j) =< \phi(x_i), \phi(x_j) > \tag{4.24}$$

There are three main kinds of kernels: the linear kernel, the polynomial kernel, and the Radial Basis Function (RBF) kernel. Linear Kernel is the most simple kind of kernel and it essentially quantifies the similarity of two vectors using the Pearson correlation, which has been used also before in order to analyze the correlation between the features. The expression is the following:

$$k(x_i, x_j) = x_i x_j \tag{4.25}$$

The polynomial kernel is very similar to the linear kernel, except for the power of the product, which is defined by a number $d > 1$.

$$k(x_i, x_j) = (x_i x_j)^d \tag{4.26}$$

The mathematical expression which defines this type of kernel is the following:

$$k(x_i, x_j) = e^{-\gamma(x_i - x_j)^2} \tag{4.27}$$

The value of γ could lead to a model that is not trained opportunely. In fact, an high value of γ generates overfitting. On the other hand, a low value of γ generates underfitting. In this case, in order to avoid such problems, using a classical rule of thumb γ is equal to $1/m$, where m is the number of features.

As introduced at the beginning of this paragraph, a regression problem that is handled with an SVM implies that the data points are enclosed in a tube. All the regression problems try to minimize the difference between the predicted value and the real one. The method used by the SVR is the ϵ-insensitive loss function, which ignores the errors between ϵ and penalizes more the predictions further away from the actual output. This is formally reported in the following relation:

$$\mathcal{L}_\epsilon = \begin{cases} 0, & \text{if } |y - f(x)| < \epsilon \\ |y - f(x)|, & \text{otherwise} \end{cases} \tag{4.28}$$

Figure 4.10b is indicative of this kind of situation, in which the white points are surrounded by the tube, even though some points are allowed to be outside. The width of such tube depends on ϵ: the tolerance of the algorithm becomes smaller as the value of ϵ is smaller. First of all, it is evident how the polynomial kernel performs extremely worse than the radial-basis function kernel. This consideration was annotated in the last paragraphs, especially in the section dedicated to the HI. A component that evolves over time with a non-linear trend, very similar in some cases to an exponential function, is difficult to be represented by a polynomial kernel, even though in this case the value of d is 3. Then, the differences between the other hyper-parameters are evident only with some versions. In particular, the normal version is characterized by a MAE with a small variance when the kernel is *rbf*. Instead, the other versions present a MAE which is a function of the decreasing value of ϵ and the increasing value of C. In fact, three-quarters of the best results are obtained with a radial basis function kernel, a C equal to 100 and ϵ equal to 1e-3. The last quarter is however characterized by the same kernel and C and a lower value of ϵ, 1e-5. In conclusion, it is possible to assert that the SVR tube has to be tight, in order to penalize a lot the misclassification. On the other hand, a possible set of misclassified data points are admitted to the high value of C.

Based on what is explained up here, a grid search with a 5-fold cross-validation method is implemented in order to find the best combination of hyperparameters that minimize the MAE between the actual RUL and the predicted one. In particular, the parameters to tune are three: the kernel type, the regularization parameter C, and tube width ϵ. Regarding the kind of kernels, in this work are tested the *linear* and the *rbf*. Instead, C and ϵ can assume respectively 10 and 100 and finally 1e-1, 1e-3 and 1e-5. In this way, the total number of test runs is 12, which becomes 60 by taking into account a cross-validation with 5 folds. Finally, each grid is trained with a different set of features. Subsections §4.3.1, §4.3.2 and §4.3.3 reports three ways to characterize the dataset, which firstly use as predictors a collection of metrics belonging to different domains, then a subset of them based on the presence of possible mutual linear correlations, and finally

window aggregations over time. In addition to what was reported in the aforementioned paragraphs, also another possible solution built on the combination of feature selection and aggregation will be proved. Tables 4.2 and 4.3 report each result where v indicates the kind of version, k the kind of kernel and the MAE is referred to each single fold. Thus, in order to evaluate which grid is better, an average over the MAE is computed and it is reported in the last column of Table 4.2 and 4.3.

4.3.6 Random Forest

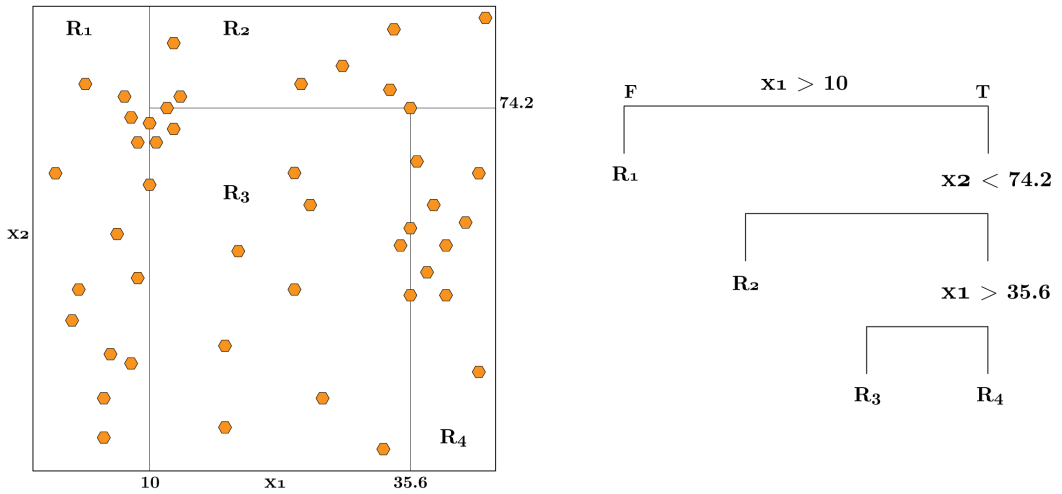


Figure 4.11: Subdivision of the predictor's space into a set of regions (a) and the resulting regression tree (b).

Random Forest is one of the most used and performing ensemble methods in data science. Since it merges together the results of various classification or regression trees, the explanation of this kind of algorithm begins with the description of the general functioning of a regression tree.

In general, a regression tree is a procedural approach that assigns a sample in a region characterized by a specific class. Thus, given the predictor's space which is made of a set of p features $\{X_1, X_2, ..., X_p\}$ which define the data, a set of k regions $\{R_1, R_2, ..., R_k\}$ could be extracted. Then, the response of sample x_0 is equal to the value associated with the region in which the sample x_0 is fallen. Figure 4.11a illustrates how regions are defined over a predictor's space. In this case the number of predictors $p = 2$, equal to $\{x_1, x_2\}$. Then, in this example, the number of regions is equal to $k = 4$ and they are $\{R_1, R_2, R_3, R_4\}$. Each of them should be associated with a specific response value, which is however useless for a brief description like this one. The tree is generated by splitting the predictor X_i into two regions at a certain cut-point s. Hence on one hand all the samples with $X_i < s$ are collected together and on the other hand all the samples with

$X_i \geq s$, as depicted in Figure 4.11b. The choice of the value of s is based on the principle that the split samples have to be as unbalanced as possible in terms of the number of samples that belong to the same response variable. A way to compute this separation is the GINI index or other similar metrics like the entropy for the classification trees and the RSS for the regression trees. Thus, in this case, the construction of the regions depends on the minimization of the RSS, which is reported in equation 4.29.

$$\text{RSS} = \sum_{j=1}^{k} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \tag{4.29}$$

where $\hat{y}_{R_j}$ is the mean response variable of the region R_j. Since it is unthinkable to split the predictor's space in a huge number of partitions, a top-down greedy approach based on a recursive binary splitting is performed. At each step, two new tree branches are defined, and so forth until the end. A problem that affects trees is the high variance. Given a set of n independent observations, the variance of the mean is σ^2/n. Thus, the division of the dataset into many different training sets is equal to reduce variance.

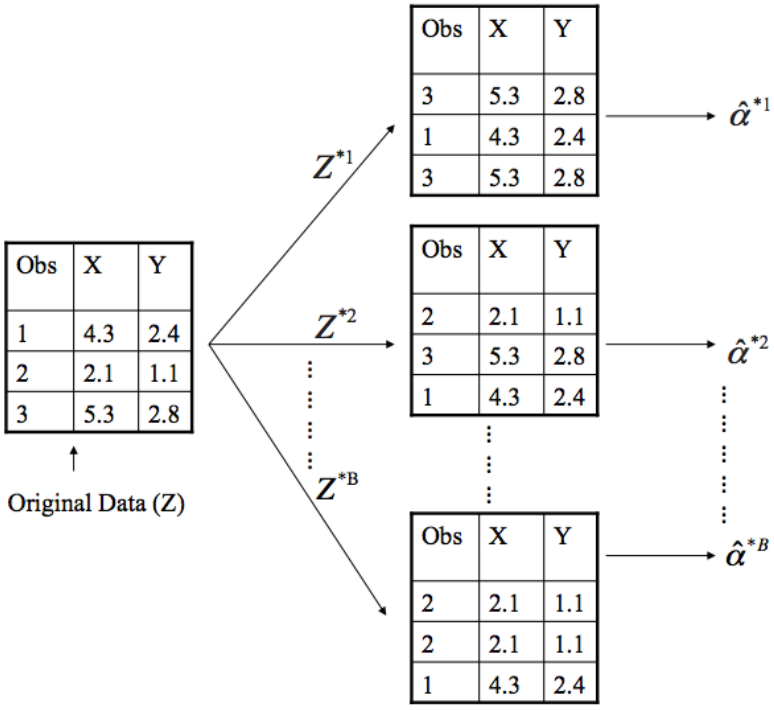


Figure 4.12: Bootstrap technique applied by Random Forest taken from the guide of the University of Cincinnati.

Figure 4.12 shows that, given the original dataset Z, a series of splits have been performed $(Z^{*1}, Z^{*2}, ..., Z^{*B})$ with B equal to the number of training sets. In addition, a single observation can be present multiple times since the sampling is with replacement.

In fact, the observation number 3 is present two times in Z^{*1}. In a classification problem, we can imagine that in the previous figure $\hat{\alpha}^{*i}$, with $i \in \{1, 2, ..., B\}$ is equal to the class predicted by the single tree trained on the dataset Z^{*i}. Thus the final class is chosen with a majority voting mechanism: the most frequent class is then the final one. In conclusion, using bootstrapping and bagging a set of hundreds or thousands of trees could be trained, starting from a set of observations from the original training set. Since each tree has a low bias and a high variance, using the majority voting approach the bias remains almost the same but the variance decreases a lot. In order to create trees that are not correlated, a subset of predictors is used to train each tree. This value m is generally equal to the $\sqrt{p}$, where p is the total number of features.

As done for the SVR, also in this case a 5-fold cross-validation method is implemented to find the best set of hyper-parameters that minimizes the MAE. In this case, the tuned values are the total number of estimators, *ne*, the maximum depth, *md* and the maximum number of features, *mf*. The number of estimators can be equal to 300, 600, or 900. The maximum depth can assume only two values, 30 or 60, and, finally, the maximum number of features can be equal to the square root or the logarithm of the total number of features of the dataset, as explained before about the bagging and the bootstrapping. Then, as for the SVR, also in this case each grid is composed of 12 runs, which becomes 60 by taking into consideration a training for each of the 5-folds, and for each version, all the 12 grids are computed. The outcomes are reported in Table 4.4 and 4.5. In general, the configuration that performs better is characterized by the same value of *md*, that is 60. Then, the number of estimators is equal to 900 for the three-quarters of versions and 600 for the remaining configuration.

4.4 Deep Learning Approaches

The performance of a machine learning model relies on the kind of handcrafted features retrieved by the data. However, it is not easy to extract different levels of detail from such signals, due to the non-linearity that affects the degradation trend of a REB. For this reason, even if they are not interpretable, deep learning models could be more suitable for this task. Moreover, as neural networks require much data and much time to be trained, their increase in performance has to be justified. In other words, they must perform considerately better with respect to the *traditional* machine learning models to be considered as a concrete advantage for the entire predictive maintenance framework.

The following paragraph will be split into two parts: each of them is devoted to describing one single type of architecture. As the same for the other sections, here are introduced only the technical details of such solutions, without reporting the final results, which will be presented in Subchapter §5.

4.4.1 Spectrogram

The architecture depicted in Figure 4.13 is the first version that takes in input the original spectrogram obtained by each sampling window and, after a series of convolutional and dense layers, returns as output the predicted RUL. More in detail, the input spectrogram

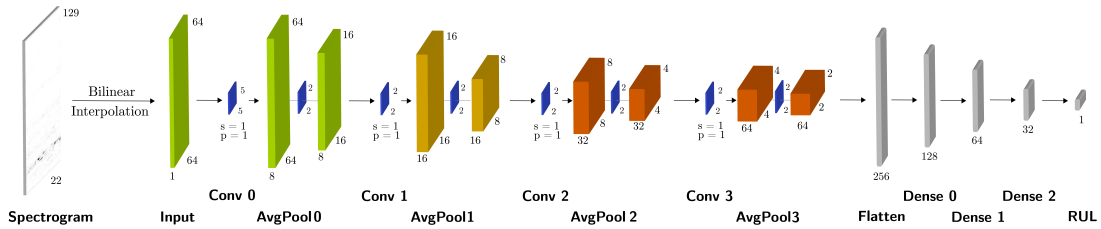


Figure 4.13: Spectrogram neural network.

is firstly transformed into a square image through a bilinear interpolation, the most used resampling technique used in computer vision to adapt the size of an image without losing information. Thus, the original spectrogram has a shape of $[129, 22, 1]$ that is transformed into an image of shape $[64, 64, 1]$. The channel size is 1 because each cell of the time-frequency plot indicates only the magnitude of the harmonic at frequency k and time t; thus, only one dimension is useful in this case to represent the information.

The first neural network layer is a convolutional one with 8 filters, a kernel size of $[5, 5]$ with stride and padding equal to $[1, 1]$. Then, the initial part is made of a series of pooling, convolutions, and batch normalizations, in order to reduce the input dimension, retrieve hierarchical features, and control overfitting, even more so with this number of parameters. In fact, after the first convolutional layer, the number of channels doubled at each hidden level, therefore starting from 8 and arriving at 64. Instead, the output x and y sizes after each pooling layer are half with respect to the input image. Also in this case, the interpolated image has a shape of $[64, 64, 8]$ which becomes $[2, 2, 64]$ after the last feature extraction part. The final part of the architecture is composed of a set of dense layers which gradually decrease the number of units in order to obtain only one neuron to perform the RUL prediction. This is initially made with a flatten layer which takes in input the image with shape $[2, 2, 64]$ to obtain a full hidden layer with 256 neurons, which then become 128, 64, 32, and finally 1.

The loss and the metric for each of these architectures are respectively the MSE and the MAE, which are reported in equation 4.30 and 4.31.

$$\mathcal{L}(y, \hat{y}) = \text{MSE}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{4.30}$$

$$\text{MAE}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{4.31}$$

A possible alternative to the architecture represented in Figure 4.13 consists of substituting all the convolutional layers with their corresponding LSTM version, in order to take into account also some temporal dependencies about the evolution of the spectrogram. More details about this kind of solution will be reported in Subchapter §4.4.2, where the aforementioned architecture will be described.

4.4.2 Wavelet

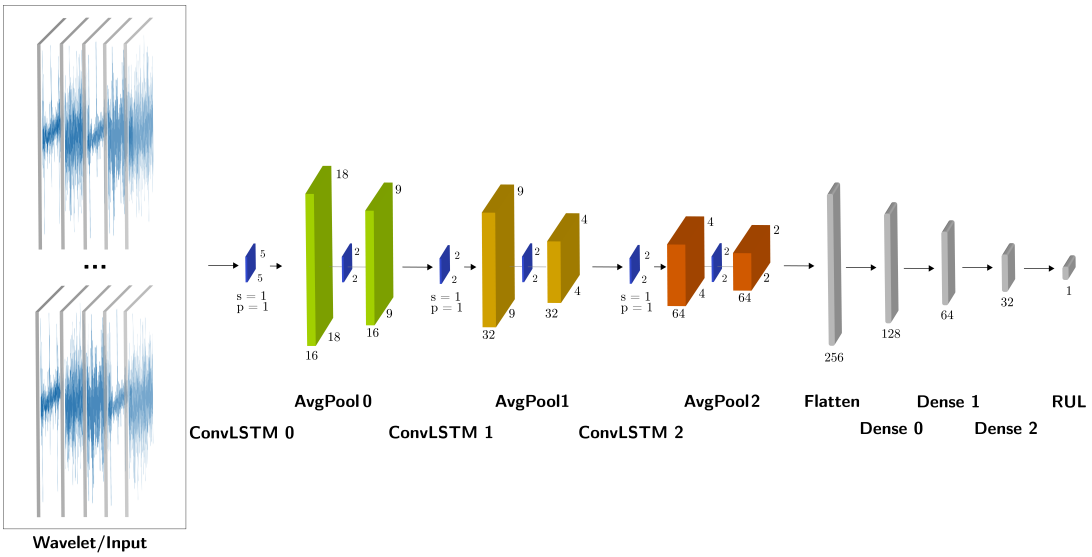


Figure 4.14: Wavelet neural network with ConvLSTM layers instead of convolutional ones.

Table 4.1 shows that spectrograms and wavelets are the most important features for the machine learning regressor. For this reason, the exploitation of the latter kind of input could be determinant to increase the performance obtained by the SpectroCNN. The possible neural network solutions that make use of wavelets are two: the first one accepts as input a single image of shape $[18, 18, 8$ and the second one a collection of images with the same shape previously reported. Solution number one will be not described here, due to the excessive likeness with the architecture described in Subchapter §4.4.1. The input of the second solution, which is depicted in Figure 4.14, is the set of decomposed wavelets extracted from the signal: using a number of tree layers equal to 3, the total number of inputs is $2^3 = 8$. Then as done for the spectrograms, each wavelet is then reshaped through the bilinear interpolation into an image of size 18x18. After that, a series of subsequent sampling windows are used to generate a vector of shape $[w, 18, 18, 8]$, where the w is the total number of sampling intervals taken into account. The principle is the same described in Subchapter §4.3.3: given the time t, all the signals sampled from t to $t + w$ are used to generate a new dataset with RUL equal to the distance between the time index $t + w + 1$ and the end life of the component. The window width w is normally equal to 5. A possible variant of this solution, which can be useful to take into account more time intervals, will be reported in Subchapter §4.4.3 because it requires a revision of the architecture illustrated in Figure 4.14.

Classical RNN suffers from the gradient exploding and vanishing problem. If the former can be contained, the latter requires different technical solutions to be handled. The most effective one relies in the LSTM cell, which thanks to the memory cell and the predisposition of a collection of four gates, is able to solve this issue. Equation 4.32

describe all the computations made by the ConvLSTM at each time instant.

$$\begin{aligned} i_t &= \sigma(w_x^i * x + w_h^i * h_{t-1} + b_i) \\ f_t &= \sigma(w_x^f * x + w_h^f * h_{t-1} + b_f) \\ o_t &= \sigma(w_x^o * x + w_h^o * h_{t-1} + b_o) \\ \tilde{c}_t &= \tanh(w_x^c * x + w_h^c * h_{t-1} + b_{\tilde{c}}) \\ c_t &= \tilde{c}_t \odot i_t + c_{t-1} \odot f_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{4.32}$$

where x is the input, σ is the sigmoid function, i, f, o and $\tilde{c}_t$ are the four gates, h and c are respectively hidden state and memory state.

All the steps between the first convolution and the last one do not change the value of the time dimension. In this manner, it is possible to make accessible to all the memory cells the temporal dependencies between each sampling window. This dataset axis is removed only by the ConvLSTM 2, which returns as output a vector of shape $[4, 4, 64]$. Similarly to the previous architecture, each convolutional step doubles the number of channels and each average pooling layer half the x and y dimension of the output feature maps. The last flatten and fully-connected layers rearrange the last filters into a FFNN, whose output is the RUL explained before.

4.4.3 Mixed

Convolutional neural networks are able to retrieve from the input image a combination of hierarchical features which cannot be easily found with the classical handcraft methods. Thus, a possible improvement of the preceding two architectures relies on the conjunction of the feature maps of such solutions with a final fully connected layer concatenation. In addition, the mixed architecture is characterized by having as first layer a MaxPooling layer. As described in Subchapter §4.3.3, the long-term aggregation performs better compared to the standard dataset. In this view, it is possible to apply the same method also to this scenario. In this case, the input vector has five dimensions, [samples, time, x, y, channels], as already explained in Subchapter §4.4.2. Then, in order to reduce the excessive amount of time features, a MaxPooling layer is applied.

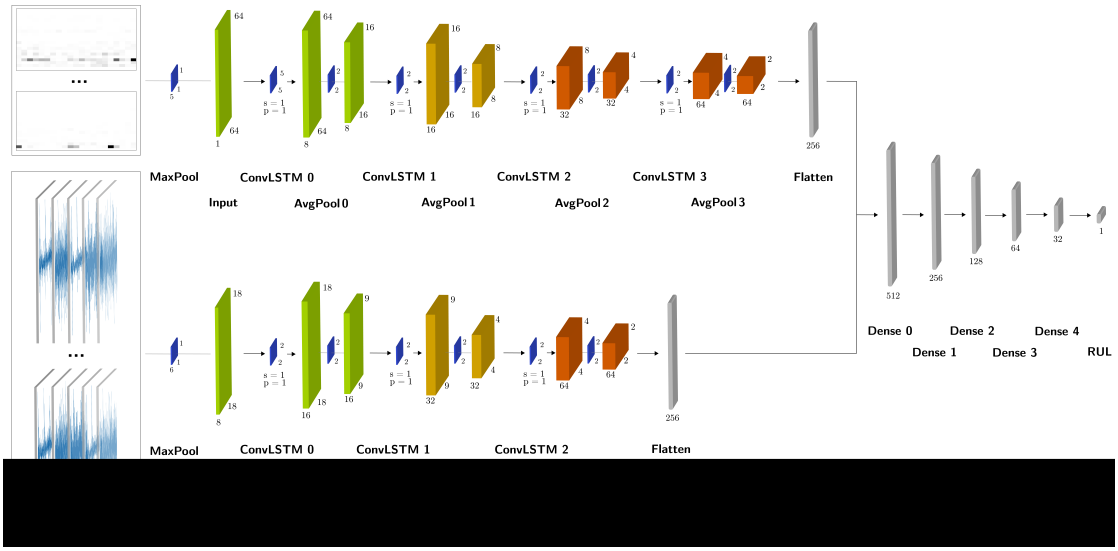


Figure 4.15: Architecture of the neural network which takes in input the spectrogram and the wavelet.

v	**k**	**C**	ϵ	**MAE***					
				1	2	3	4	5	Avg
N	rbf	10	0.1	0.05771	0.05494	0.05783	0.05779	0.05629	0.05691
	poly	10	0.1	0.55494	0.36166	3.34902	0.29769	0.31504	0.97567
	rbf	10	0.001	0.04143	0.03743	0.04018	0.04104	0.03803	0.03962
	poly	10	0.001	0.66413	0.33153	3.09220	0.46353	0.40620	0.99152
	rbf	10	1e-05	0.04139	0.03749	0.04032	0.04107	0.03807	0.03967
	poly	10	1e-05	0.65412	0.33023	3.09265	0.46854	0.40658	0.99042
	rbf	100	0.1	0.05202	0.05126	0.05404	0.05204	0.05430	0.05273
	poly	100	0.1	1.43554	0.48016	11.28947	1.21250	0.73338	3.03021
	rbf	**100**	**0.001**	0.03787	0.03239	0.03826	0.03507	0.03564	**0.03585**
	poly	100	0.001	1.12171	0.76144	13.36313	2.27630	0.93106	3.69073
	rbf	100	1e-05	0.03800	0.03233	0.03840	0.03522	0.03574	0.03594
	poly	100	1e-05	1.07413	0.75815	13.43280	2.29609	0.93362	3.69896
FA	rbf	10	0.1	0.04627	0.04600	0.04942	0.04736	0.04731	0.04727
	poly	10	0.1	0.64382	0.18125	3.16981	0.30303	0.17649	0.89488
	rbf	10	0.001	0.01405	0.01199	0.01252	0.01483	0.01348	0.01337
	poly	10	0.001	1.79745	0.11101	3.43171	0.28157	0.16664	1.15768
	rbf	10	1e-05	0.01404	0.01201	0.01249	0.01485	0.01347	0.01337
	poly	10	1e-05	1.80714	0.11077	3.46701	0.27748	0.16615	1.16571
	rbf	100	0.1	0.04382	0.04421	0.04496	0.04598	0.04515	0.04482
	poly	100	0.1	1.05681	0.24505	6.06334	0.29414	0.23780	1.57943
	rbf	**100**	**0.001**	0.00992	0.00823	0.00910	0.00967	0.00963	**0.00931**
	poly	100	0.001	3.21627	0.20291	4.81646	0.53617	0.43557	1.84147
	rbf	100	1e-05	0.00990	0.00821	0.00907	0.00975	0.00964	0.00932
	poly	100	1e-05	3.17204	0.21018	4.84541	0.51762	0.43293	1.83564
FS	rbf	10	0.1	0.07728	0.07215	0.07690	0.07739	0.07612	0.07597
	poly	10	0.1	2.10326	0.43936	2.11023	0.47990	0.30309	1.08717
	rbf	10	0.001	0.07170	0.06157	0.06766	0.06891	0.06994	0.06796
	poly	10	0.001	2.23742	0.45249	3.03874	0.57818	0.32586	1.32654
	rbf	10	1e-05	0.07173	0.06164	0.06780	0.06892	0.06996	0.06801
	poly	10	1e-05	2.23535	0.45219	3.06278	0.57983	0.32709	1.33145
	rbf	100	0.1	0.07461	0.07100	0.07970	0.07447	0.07680	0.07531
	poly	100	0.1	7.44446	0.76861	5.39092	0.84727	0.42921	2.97609
	rbf	**100**	**0.001**	0.06646	0.05928	0.06801	0.06580	0.06516	**0.06495**
	poly	100	0.001	10.25241	1.09721	8.87021	1.07934	0.53993	4.36782
	rbf	100	1e-05	0.06674	0.05941	0.06806	0.06596	0.06529	0.06509
	poly	100	1e-05	10.26441	1.09576	8.90478	1.08160	0.54244	4.37780

Table 4.2: Part 1: Grid search obtained with the Support Vector Machine on the training set with a cross-validation method with 5 folds. k is the kind of kernel, C is the regularization parameter and ϵ is the homonymous value present in the optimization problem, as reported in equation 4.23. *: the numbers are referred to the MAE computed on each single validation fold.

v	**k**	**C**	ϵ	**MAE***					
				1	2	3	4	5	Avg
FA+FS	rbf	10	0.1	0.05418	0.05114	0.05121	0.05273	0.05375	0.05260
	poly	10	0.1	1.59060	0.18135	1.76614	0.21599	0.23172	0.79716
	rbf	10	0.001	0.02307	0.02191	0.02045	0.02223	0.02350	0.02223
	poly	10	0.001	1.67526	0.16954	1.80037	0.22682	0.18361	0.81112
	rbf	10	1e-05	0.02309	0.02199	0.02043	0.02224	0.02351	0.02225
	poly	10	1e-05	1.67908	0.17171	1.81430	0.22780	0.18347	0.81527
	rbf	100	0.1	0.04616	0.04747	0.04668	0.04739	0.04692	0.04692
	poly	100	0.1	1.21372	0.34360	3.16147	0.26018	0.28106	1.05201
	rbf	100	0.001	0.01469	0.01345	0.01442	0.01508	0.01585	0.01470
	poly	100	0.001	1.61818	0.61665	5.65068	0.18825	0.33368	1.68149
	rbf	**100**	**1e-05**	0.01471	0.01343	0.01423	0.01508	0.01595	**0.01468**
	poly	100	1e-05	1.63567	0.61228	5.65765	0.18663	0.33076	1.68460

Table 4.3: Part 2: Grid search obtained with the Support Vector Machine on the training set with a cross-validation method with 5 folds. k is the kind of kernel, C is the regularization parameter and ϵ is the homonymous value present in the optimization problem, as reported in equation 4.23. *: the numbers are referred to the MAE computed on each single validation fold.

v	**ne**	**md**	**mf**	**MAE***					
				1	2	3	4	5	Avg
N	300	30	sqrt	0.01944	0.01730	0.01888	0.01754	0.01834	0.01830
	600	30	sqrt	0.01945	0.01713	0.01874	0.01740	0.01840	0.01822
	900	30	sqrt	0.01940	0.01716	0.01878	0.01732	0.01833	0.01820
	300	30	log2	0.02137	0.01917	0.02167	0.02033	0.02087	0.02068
	600	30	log2	0.02134	0.01918	0.02134	0.02018	0.02063	0.02053
	900	30	log2	0.02148	0.01920	0.02129	0.02010	0.02057	0.02053
	300	60	sqrt	0.01944	0.01730	0.01888	0.01754	0.01834	0.01830
	600	60	sqrt	0.01945	0.01713	0.01874	0.01740	0.01840	0.01822
	900	**60**	**sqrt**	0.01940	0.01714	0.01876	0.01732	0.01833	**0.01818**
	300	60	log2	0.02137	0.01917	0.02167	0.02033	0.02087	0.02068
	600	60	log2	0.02134	0.01918	0.02134	0.02018	0.02063	0.02053
	900	60	log2	0.02148	0.01920	0.02129	0.02010	0.02057	0.02053
FA	300	30	sqrt	0.00290	0.00262	0.00369	0.00279	0.00321	0.00304
	600	30	sqrt	0.00286	0.00256	0.00363	0.00284	0.00315	0.00301
	900	30	sqrt	0.00289	0.00255	0.00359	0.00283	0.00316	0.00300
	300	30	log2	0.00320	0.00280	0.00378	0.00299	0.00342	0.00324
	600	30	log2	0.00320	0.00280	0.00379	0.00299	0.00338	0.00323
	900	30	log2	0.00317	0.00278	0.00382	0.00298	0.00337	0.00322
	300	60	sqrt	0.00290	0.00262	0.00369	0.00279	0.00321	0.00304
	600	60	sqrt	0.00286	0.00256	0.00363	0.00284	0.00315	0.00301
	900	**60**	**sqrt**	0.00288	0.00254	0.00359	0.00283	0.00316	**0.00299**
	300	60	log2	0.00320	0.00280	0.00378	0.00299	0.00342	0.00324
	600	60	log2	0.00320	0.00280	0.00379	0.00299	0.00338	0.00323
	900	60	log2	0.00317	0.00278	0.00382	0.00298	0.00337	0.00322
FS	300	30	sqrt	0.03663	0.03290	0.03580	0.03440	0.03478	0.03490
	600	30	sqrt	0.03662	0.03273	0.03581	0.03448	0.03457	0.03484
	900	30	sqrt	0.03650	0.03267	0.03592	0.03447	0.03453	0.03482
	300	30	log2	0.03663	0.03290	0.03580	0.03440	0.03478	0.03490
	600	30	log2	0.03662	0.03273	0.03581	0.03448	0.03457	0.03484
	900	30	log2	0.03650	0.03267	0.03592	0.03447	0.03453	0.03482
	300	60	sqrt	0.03663	0.03290	0.03580	0.03440	0.03478	0.03490
	600	60	sqrt	0.03662	0.03273	0.03581	0.03448	0.03457	0.03484
	900	60	sqrt	0.03651	0.03267	0.03592	0.03447	0.03453	0.03482
	300	60	log2	0.03663	0.03290	0.03580	0.03440	0.03478	0.03490
	600	60	log2	0.03662	0.03273	0.03581	0.03448	0.03457	0.03484
	900	**60**	**log2**	0.03651	0.03267	0.03592	0.03445	0.03451	**0.03480**

Table 4.4: Part 1: Grid search obtained with the Random Forest Regressor on the training set with a cross-validation method with 5 folds. *ne* is the number of estimators, *md* the maximum depth and *mf* the maximum number of features. *: the numbers are referred to the MAE computed on each single validation fold.

v	**ne**	**md**	**mf**	**MAE***					
				1	2	3	4	5	Avg
FA+FS	300	30	sqrt	0.00410	0.00342	0.00516	0.00392	0.00430	0.00418
	600	30	sqrt	0.00403	0.00342	0.00517	0.00391	0.00422	0.00415
	900	30	sqrt	0.00403	0.00343	0.00518	0.00395	0.00419	0.00416
	300	30	log2	0.00410	0.00342	0.00516	0.00392	0.00430	0.00418
	600	30	log2	0.00403	0.00342	0.00517	0.00391	0.00422	0.00415
	900	30	log2	0.00403	0.00343	0.00518	0.00395	0.00419	0.00416
	300	60	sqrt	0.00410	0.00342	0.00516	0.00392	0.00430	0.00418
	600	60	sqrt	0.00403	0.00342	0.00517	0.00391	0.00422	0.00415
	900	60	sqrt	0.00403	0.00343	0.00518	0.00395	0.00419	0.00416
	300	60	log2	0.00410	0.00342	0.00516	0.00392	0.00430	0.00418
	600	**60**	**log2**	0.00403	0.00342	0.00516	0.00391	0.00421	**0.00414**
	900	60	log2	0.00403	0.00343	0.00518	0.00395	0.00419	0.00416

Table 4.5: Part 2: Grid search obtained with the Random Forest Regressor on the training set with a cross-validation method with 5 folds. *ne* is the number of estimators, *md* the maximum depth and *mf* the maximum number of features. *: the numbers are referred to the MAE computed on each single validation fold.

Input	**v**	**bs**	**ep**	**lr**	**w**	$\mathcal{L}_{\mathbf{T}}$	$\mathbf{MAE_T}$	$\mathcal{L}_{\mathbf{V}}$	$\mathbf{MAE_V}$
s	N	32	50	1e-05	-	0.00649	0.05990	0.01775	0.10411
		32	50	1e-06	-	0.02612	0.11437	0.01759	0.10029
		64	50	1e-05	-	0.01007	0.07274	0.01456	0.09359
		64	**50**	**1e-06**	-	0.03005	0.11831	0.01447	**0.09084**
	LSTM	**32**	**100**	**1e-05**	5	0.00098	0.02257	0.00773	**0.06450**
		32	100	1e-06	5	0.01629	0.09633	0.01622	0.11150
		64	100	1e-05	5	0.00530	0.05240	0.01117	0.09042
		64	100	1e-06	5	0.01925	0.11392	0.03313	0.16260
	LSTM+MP	32	100	1e-05	20	0.00123	0.02420	0.00437	0.05288
		32	100	1e-06	20	0.01272	0.08258	0.01049	0.08711
		64	**100**	**1e-05**	20	0.00176	0.02897	0.00254	**0.03830**
		64	100	1e-06	20	0.02604	0.13101	0.02488	0.14024
w	N	32	50	1e-05	-	0.00724	0.06540	0.03450	0.12101
		32	50	1e-06	-	0.03505	0.14224	0.02346	0.11590
		64	50	1e-05	-	0.03421	0.07442	0.02370	0.10402
		64	**50**	**1e-06**	-	0.03215	0.12641	0.03267	**0.10105**
	LSTM	**32**	**100**	**1e-05**	5	0.00608	0.05862	0.00523	**0.06235**
		32	100	1e-06	5	0.03178	0.13193	0.01812	0.10736
		64	100	1e-05	5	0.00737	0.06299	0.00608	0.06676
		64	100	1e-06	5	0.03344	0.16195	0.02214	0.14154
	LSTM+MP	**32**	**100**	**1e-05**	20	0.00670	0.06097	0.00254	**0.04326**
		32	100	1e-06	20	0.01259	0.08026	0.00323	0.04766
		64	100	1e-05	20	0.00640	0.05878	0.00284	0.04421
		64	100	1e-06	20	0.01463	0.08589	0.00419	0.05376
m	N	**32**	**100**	**1e-05**	-	0.00271	0.03944	0.00305	**0.04274**
		32	100	1e-06	-	0.00553	0.05240	0.00471	-
		64	100	1e-05	-	0.00301	0.04012	0.00314	-
		64	100	1e-06	-	0.00542	0.05523	0.00553	-
	LSTM	**32**	**100**	**1e-05**	5	0.00271	0.03944	0.00305	**0.04274**
		32	100	1e-06	5	0.00553	0.05240	0.00471	0.05450
		64	100	1e-05	5	0.00301	0.04012	0.00314	0.04900
		64	100	1e-06	5	0.00542	0.05523	0.00553	0.05725
	LSTM+MP	**32**	**100**	**1e-05**	20	0.00213	0.03405	0.00291	**0.04257**
		32	100	1e-06	20	0.00521	0.05440	0.00452	0.05551
		64	100	1e-05	20	0.00294	0.04214	0.00305	0.04890
		64	100	1e-06	20	0.00511	0.05605	0.00502	0.05840

Table 4.6: Grid search for each kind of architecture, indicated by *input*, and for each version, indicated by *v*. *s* stands for spectrogram, *w* for wavelet and *m* for mixed. Regarding the versions, *N* is the normal, *LSTM* includes at least one ConvLSTM layer and *LSTM+MP* also an initial MaxPooling layer. The hyper-parameters are the batch size, *bs*, the number of epochs, *ep*, the learning rate, *lr*, and the time dimension, *w*. $\mathcal{L}_{\mathrm{T}}$ and $\mathcal{L}_{\mathrm{V}}$ are the losses computed respectively on the training and on the validation set, as for the $\mathrm{MAE_T}$ and $\mathrm{MAE_V}$.

Chapter 5

Results

In the first section of Chapter §4 are described all the possible variants that can be used to build the Health Index, while the second part is devoted to reporting the methods applied on the dataset to retrieve the features that can be used by a machine learning or deep learning algorithm. This chapter, instead, reports what was previously found as the best solution for building the HI and the outcomes obtained by training each data science model with different configurations for all the testing bearings. Regarding the paragraphs devoted to the machine learning algorithms, the differences between the predictions with a dataset composed of the entire set of features, as reported in Subchapter §4.3.1, of only a subset of features (Subchapter §4.3.2) or their aggregation (Subchapter §4.3.3) will be reported, using as starting point the grids search of Table 4.2 and Table 4.4. On the other side, regarding the deep learning algorithms, will be analyzed the outcomes of each architecture described in Subchapter §4.4 basing, also in this case, on the grid search reported in Table 4.6. Finally, a series of hierarchical comparisons will be made in order to point out the regression algorithm that performs better on the test bearings. However, before presenting the results, the manner used to assess if one model is more accurate with respect to another will be mentioned, as it represents a fundamental clarification about the kind of analysis performed in this project.

Normally, the research relating the predictive maintenance of a REB concentrate their focus only on the last five hundred sampling windows, paying particular attention to the gap between the predicted and the real RUL of the last dataset sample. Furthermore, this way forward leads to a completely wrong prediction during the initial life of the bearing. Usually, the results of the aforementioned papers are characterized by an infinite RUL before the last five hundred samples and low uncertainty in the final phase of the prediction. In the author's opinion, it is more indicative a general RUL prediction which tries to define the status of a REB during its entire working life. It is accepted that the final samples are more important in a predictive maintenance scenario than the initial ones and it is also true that the FPT is the threshold that decree if the prognosis can be started or not. However, if on one side a prediction made before the FPT can be considered wrong, also a prediction made on a fixed time period is anyway erroneous, with the addition that it is not possible to know *a priori* the total number of samples of a bearing and that a final accurate prediction cannot compensate the false hope and the

precarious situation generated by an initial infinite RUL. For these reasons, the approach used in this section is the following: the goodness of an algorithm is measured by taking into account the closeness between the ground truth and the predicted RUL of the entire life and the last five hundred samples, and the gap between the average of the last ten differences between the real RUL and the predicted one. Formally, it can be represented by relation 5.1.

$$\text{Score}(y, \hat{y}) = \frac{1}{6} \cdot (3 \cdot \text{MAE}(y, \hat{y}) + 2 \cdot \text{MAE}_{\text{L}}(y_L, \hat{y}_L) + \text{DIFF}(y_F, \hat{y}_F)) \tag{5.1}$$

where the MAE is the one reported in equation 4.31, the MAE_{L} is the MAE computed for the last five hundred samples of the predicted RUL, $\hat{y}_L$, and of the actual RUL, y_L, and DIFF is the average of the ten last differences between the prediction and the ground truth, $\hat{y}_F$ and y_F. In a broad sense, the strength of the *score* is that it can be exploited as a hierarchical comparison metric: given two sets of MAE, MAE_{L} and DIFF, it firstly compares the MAE and, in the case of these are similar, it put the attention of the difference of the MAEs_{L}. Finally, if this other paragon fails, it evaluates the DIFF. Therefore, it becomes useful in decreeing which of such outcomes can be considered as the best.

5.1 Health Index

Subchapter §4.1 was completely dedicated to the description of the possible methods that can be exploited to construct the HI. In summary, what is found to be more effective on Bearing 1_1 is the combination of three factors: the horizontal axis instead of the vertical one, the MD instead of the FD, and a smoothing technique based on a LOESS filter instead of a WMA. In order to follow the same path from Chapter §4 to Chapter §5, the features used to train the ML/DL algorithms are extracted from the horizontal signal, and the final RUL prediction is smoothed with a LOESS filter. Figure 5.1 illustrate what mentioned above.

Bearing1_2, Bearing1_3, Bearing1_4, and generally also Bearing1_7 follow the traditional bathtub curve introduced at the beginning of this document. In addition, it is important to remember that Bearing1_2 is a part of the training set, together with Bearing1_1, copiously discussed in Chapter §4. T As the HI is a part of the features extracted in Subchapter §4.3.1, this behavior will lead to one main issue: what is expected by a machine learning, or by a deep learning algorithm,[1] is an exponential-like increasing, which is the trend it learned from the training set. Instead, Bearing1_5 and Bearing1_6 have a different behavior. The latter is the worst because observing only the HI, it seems that its RUL reaches a value near to zero around the time index 1900, and then it comes back to the value reached around the time index 1500. The former has a different trend, which

[1]Despite a DL algorithm does not make use of the features extracted in Subchapter §4.3.1, and thus the HI is not taken into account to predict the RUL, it is possible to assert, using the definition of the health index, that the inputs of such architectures, spectrogram, and wavelets, follow the same trend.

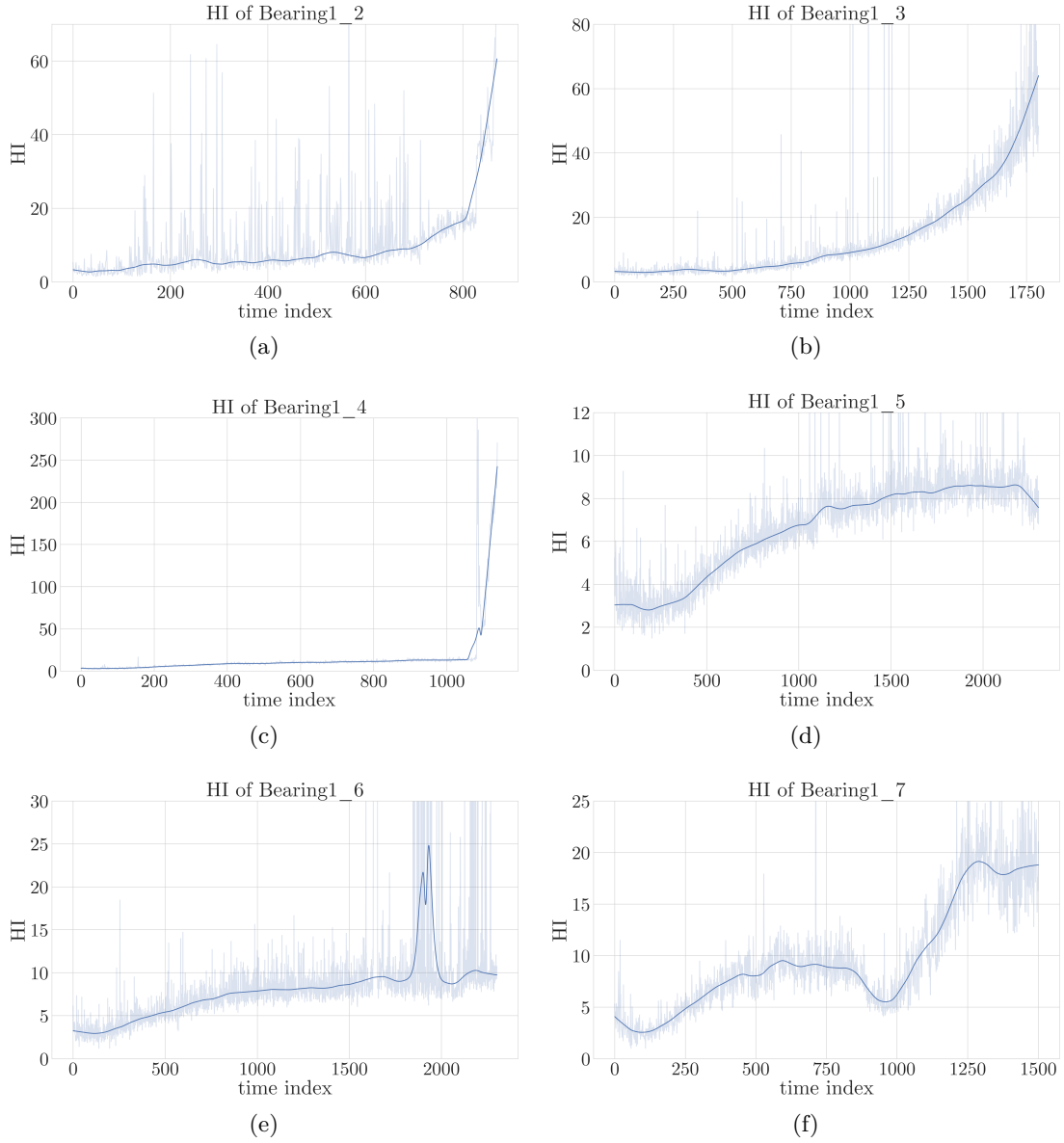


Figure 5.1: HI of the training Bearing1_2 and of test bearings Bearing1_3, Bearing1_4, Bearing1_5, Bearing1_6, and Bearing1_7. The opaque signal is the HI before the application of the smoothing technique, instead, the more intense curve is the result of the LOESS filter.

is however critical due to the final decrease of the HI. For these reasons, also Bearing1_7 would be counted in the not exponential-like trends, and this is in part true. On the other side, despite this trend is checkered, it is possible to define an exponential increase during the entire life of the component. In conclusion, by observing Figure 5.1, some suspects

regarding the performance obtained by each test bearing can be raised.

5.2 Machine Learning

Starting from the results reported in Table 4.2 and Table 4.4, for each kind of regression algorithm and each version, the best configuration of hyper-parameters is used to predict the RUL of the five test bearings. The following two paragraphs report the performance of such algorithms grouped by REB and by version. In order to recall the possible versions, these are four: *N*, *FA*, *FS* and *FA+FS*. The first one is the normal: the set of features are the original ones reported in Subchapter §4.3.1. The second one uses a dataset with a temporal aggregation of features, as described in Subchapter §4.3.3. The third one is based on the selection of only a subset of features, chosen with the help of the correlation matrix of Figure 4.7. The final one is formed by selecting the same subset of features pointed out in Subchapter §4.3.2 but on the dataset returned by the feature aggregation method.

5.2.1 Support Vector Machine

As reported in Table 5.1a, the higher scores are reached by versions *N* and *FA+FS*, even if their average score computed for all the test bearings are very different. As it can be seen from Table 5.1b, version *N* has a mean score of 0.14515 and version *FA+FS* reaches a mean score of 0.21438. Practically, this depends on the performance of the latter version on Bearing1_5, which is completely worse than the former. In this way, it is evident like the average score reported in Table 5.1b is useful to decree the best version. As will be also analyzed in Chapter §7, dedicated to the conclusions, it is not sufficient that a version predicts correctly only a subset of bearings. Instead, the discriminant for the choice of the best solution is that the regressor has to obtain appreciable results for all the testing REBs. The goodness of the normal version is also visible in Figure 5.3.

5.2.2 Random Forest

The RFR performs generally better with respect to the SVR, and this is clearly visible by observing the average scores reported in Table 5.2b. However, also in this case the version *N* and *FA+FS* are the ones that reach the best outcomes with Bearing1_3, Bearing1_5, Bearing1_6, and Bearing1_7. Despite these considerations, the most performing version in the *FS*, which is only better respect the others on Bearing1_4. In fact, by observing Figure 5.3b, it is evident that the feature aggregation for this test dataset does not increase the quality of the prediction, as the versions *FA* and *FA+FS* have a MAE and a MAE_L definitely much higher respect to the other two configurations, despite the DIFF is similar.

5.3 Deep Learning

As done for the previous subchapter, this section is devoted to presenting the performances reached by each ANN with different configurations: *N*, *LSTM* and *LSTM+MP*. Starting

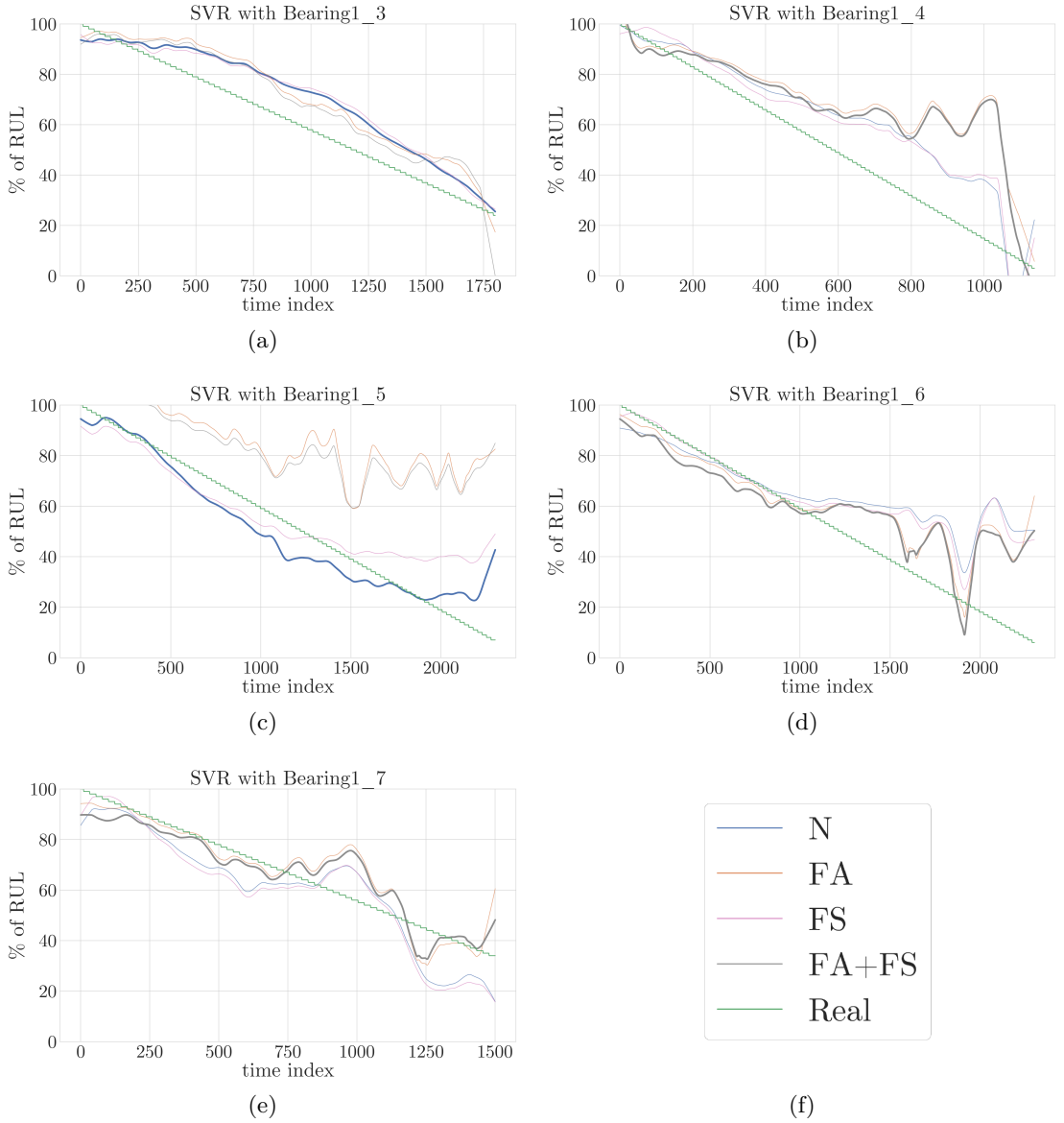


Figure 5.2: RUL Prediction of each test bearing using the Support Vector Machine with different features configurations.

from a classical CNN represented by version *N*, as reported in Subchapter §4.4, the second configuration implies the addition of one or more ConvLSTM layers, depending on the kind of architecture, and the third is characterized by having an initial MaxPooling layer which contracts the time dimension.

Fig	Bearing	Version	MAE	MAE_L	DIFF	Score
5.2a	Bearing1_3	**N**	0.09504	0.07766	0.01769	**0.07635**
		FA	0.09857	0.09669	0.05735	0.09108
		FS	0.09660	0.08237	0.02560	0.08002
		FA+FS	0.08887	0.09946	0.21828	0.11397
5.2b	Bearing1_4	N	0.12560	0.18462	0.15984	0.15098
		FA	0.19788	0.32786	0.04408	0.21557
		FS	0.12176	0.18861	0.08788	**0.13840**
		FA+FS	0.18219	0.30498	0.07080	0.20456
5.2c	Bearing1_5	**N**	0.06885	0.09473	0.34792	**0.12399**
		FA	0.31645	0.58904	0.75251	0.47999
		FS	0.09455	0.23474	0.41525	0.19473
		FA+FS	0.29312	0.56489	0.77208	0.46354
5.2d	Bearing1_6	N	0.14821	0.34631	0.44509	0.26373
		FA	0.11375	0.26717	0.56619	0.24030
		FS	0.12340	0.31206	0.40696	0.23354
		FA+FS	0.12017	0.25043	0.43898	**0.21673**
5.2e	Bearing1_7	N	0.08538	0.11768	0.17272	0.11070
		FA	0.05613	0.07893	0.24279	0.09484
		FS	0.09372	0.12977	0.17603	0.11945
		FA+FS	0.05929	0.06445	0.13187	**0.07311**

(a)

Version	MAE	MAE_L	DIFF	Score
N	0.10462	0.16420	0.22865	**0.14515**
FA	0.15656	0.27194	0.33258	0.22435
FS	0.10600	0.18951	0.22234	0.15323
FA+FS	0.14873	0.25684	0.32640	0.21438

(b)

Table 5.1: Results obtained by the application of a Support Vector Machine with the best set of parameters for each test bearing and each kind of version (a). Average of the metrics grouped by version (b). The MAE_L is the Mean Absolute Error computed with the last five hundred samples and finally *DIFF* is the average between the real and the predicted RUL of the last ten samples. The *score* is a weighted average of the previous three metrics.

5.3.1 Spectrogram NN

The neural network with input the spectrogram performs particularly well with a normal configuration, which does not imply the usage of any LSTM cell or MaxPooling layer. This is confirmed by the results reported in Table 5.3a. Except for Bearing1_4 which obtain better predictions with the LSTM version, the other testing REBs reach a lower

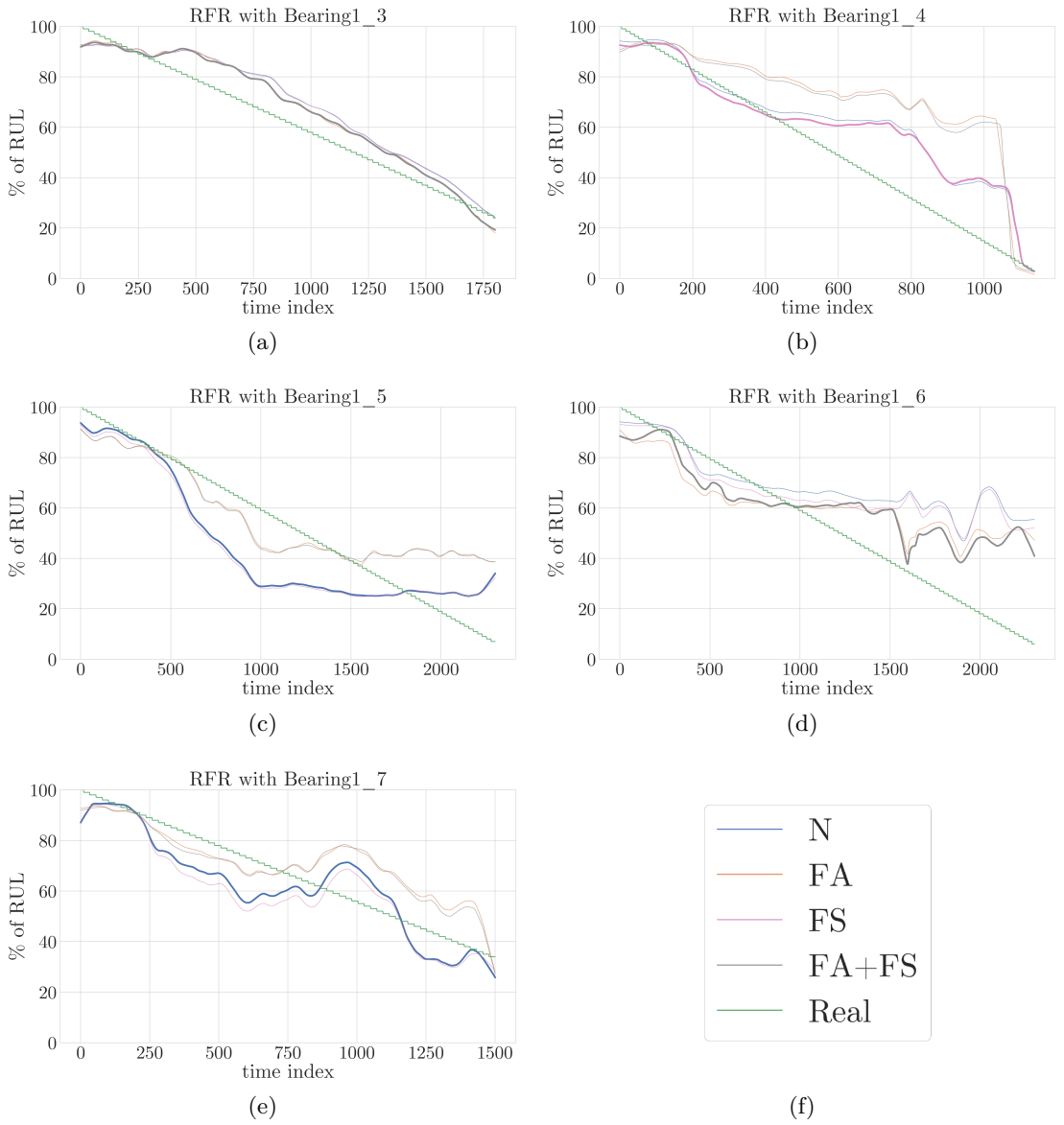


Figure 5.3: RUL Prediction of each test bearing using the Random Forest with different features configurations.

score with the normal version. In this scenario, it is also important to delineate, with respect to machine learning, two kinds of bearings. Bearing1_5 and Bearing1_6 have a bad DIFF with all the versions. In other words, in these two cases, it is not possible to find out a particular neural network configuration that is more able to decrease the value of the score. Instead, Bearing1_3, Bearing1_4, and Bearing1_7 present some differences, caused by the different configurations, in terms of MAE, MAE_L and DIFF. In particular,

Fig	Bearing	Version	MAE	MAE_L	DIFF	Score
5.3a	Bearing1_3	N	0.07583	0.05319	0.00425	0.05636
		FA	0.06353	0.04258	0.05773	0.05558
		FS	0.07484	0.05080	0.00266	0.05480
		FA+FS	0.06368	0.04094	0.04667	**0.05326**
5.3b	Bearing1_4	N	0.11354	0.19618	0.00011	0.12218
		FA	0.21772	0.33807	0.01449	0.22397
		FS	0.11114	0.19529	0.00051	**0.12075**
		FA+FS	0.20761	0.32499	0.00508	0.21298
5.3c	Bearing1_5	**N**	0.12819	0.10166	0.26571	**0.14227**
		FA	0.10027	0.24946	0.31571	0.18591
		FS	0.13673	0.10135	0.25002	0.14382
		FA+FS	0.10226	0.25034	0.31630	0.18729
5.3d	Bearing1_6	N	0.17827	0.41659	0.49399	0.31033
		FA	0.15752	0.32655	0.41558	0.25687
		FS	0.16665	0.40059	0.46100	0.29369
		FA+FS	0.14147	0.29946	0.35690	**0.23004**
5.3e	Bearing1_7	**N**	0.08281	0.07380	0.07455	**0.07843**
		FA	0.08815	0.14598	0.04478	0.10020
		FS	0.09446	0.07029	0.05063	0.07910
		FA+FS	0.08733	0.13457	0.05124	0.09706

(a)

Version	MAE	MAE_L	DIFF	Score
N	0.11573	0.16828	0.16772	0.14191
FA	0.12544	0.22053	0.16966	0.16450
FS	0.11676	0.16366	0.15296	**0.13843**
FA+FS	0.12047	0.21006	0.15524	0.15613

(b)

Table 5.2: Results obtained by the application of the Random Forest Regressor with the best set of parameters for each test bearing and each kind of version (a). Average of the metrics grouped by version (b). The MAE_L is the Mean Absolute Error computed with the last five hundred samples and finally *DIFF* is the average between the real and the predicted RUL of the last ten samples. The *score* is a weighted average of the previous three metrics.

it is indicative the case of Bearing1_4. In this case, as it can be seen in Figure 5.4b, the MAE obtained by version *LSTM+MP* is half of the MAE of the best performing version, the *LSTM*. However, if the two MAE_L are quite similar, there is a huge gap in terms of DIFF. For this reason, even if the two total scores differ little, the better score is reached using the configuration with the ConvLSTM layer. In conclusion, as it is reported in Table 5.3b, the better predictions are obtained with the normal version.

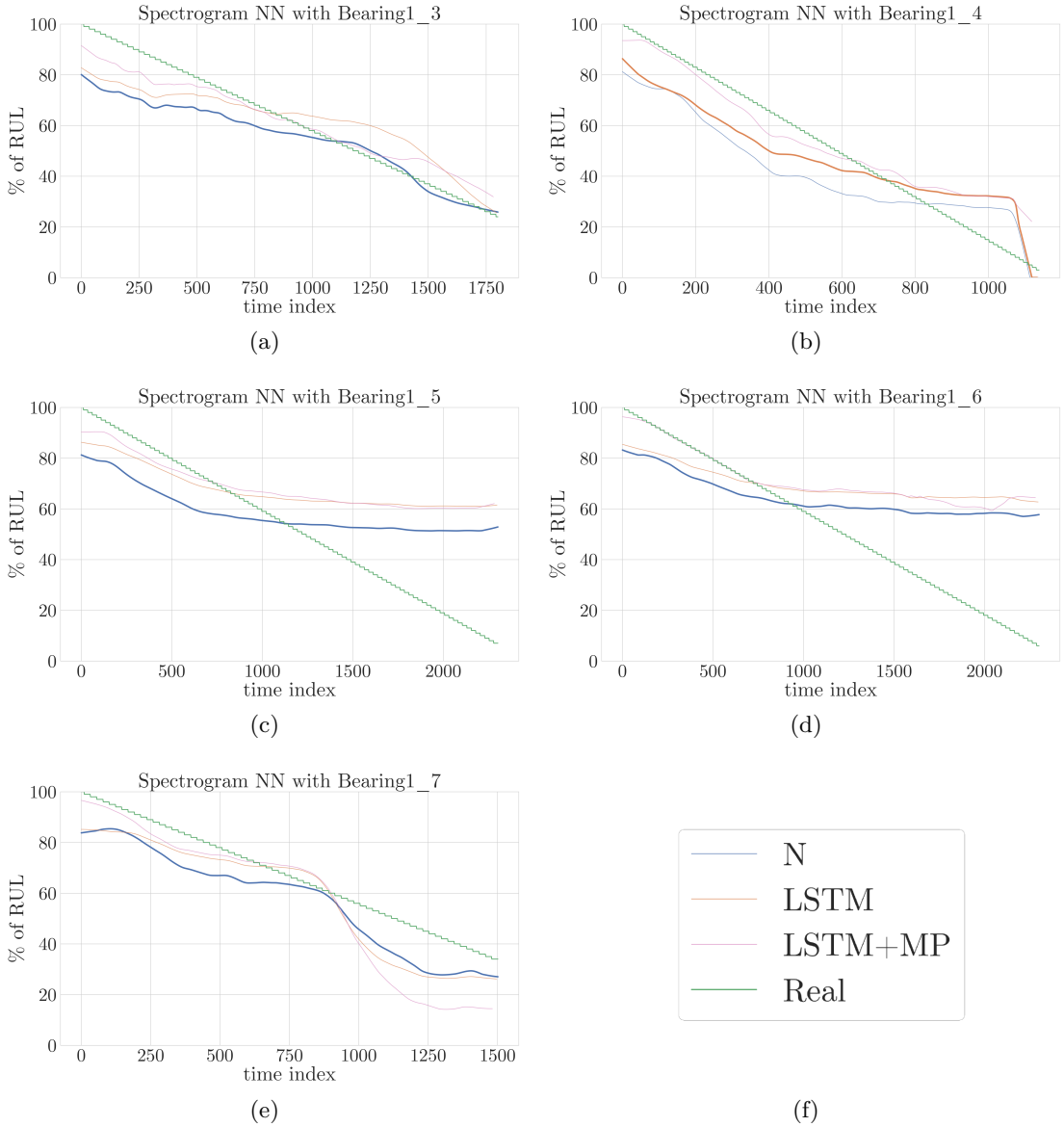


Figure 5.4: RUL prediction for each test bearing using the Spectrogram Neural Network with different versions of architectures.

5.3.2 Wavelet NN

The Wavelet NN is in general more accurate with a configuration that implies the usage of a ConvLSTM and a MaxPooling layer, as reported in Table 5.4b. This is also confirmed by the scores of Table 5.4b: the *LSTM+MP* is the best configuration for four-fifths of the bearings, differently respect to the previous outcomes where it was not immediate that the best version was also the one which obtained the best predictions. Moreover,

Fig	Bearing	Version	MAE	MAE_L	DIFF	Score
		N	0.08187	0.01669	0.01835	**0.04956**
5.4a	Bearing1_3	LSTM	0.08944	0.08601	0.01694	0.07621
		LSTM+MP	0.04987	0.07802	0.08043	0.06435
		N	0.14003	0.07913	0.03200	0.10172
5.4b	Bearing1_4	**LSTM**	0.11007	0.08661	0.03200	**0.08924**
		LSTM+MP	0.06688	0.11698	0.19632	0.10515
		N	0.17347	0.34850	0.45767	**0.27918**
5.4c	Bearing1_5	LSTM	0.19125	0.44442	0.54437	0.33449
		LSTM+MP	0.18301	0.43761	0.54994	0.32903
		N	0.18864	0.41752	0.51674	**0.31962**
5.4d	Bearing1_6	LSTM	0.21282	0.48027	0.56747	0.36108
		LSTM+MP	0.19075	0.45921	0.58441	0.34584
		N	0.09870	0.12299	0.06913	**0.10187**
5.4e	Bearing1_7	LSTM	0.09075	0.14957	0.07781	0.10820
		LSTM+MP	0.10314	0.23914	0.19538	0.16385

(a)

Version	MAE	MAE_L	DIFF	Score
N	0.13654	0.19697	0.21878	**0.17039**
LSTM	0.13887	0.24937	0.24772	0.19384
LSTM+MP	0.11873	0.26619	0.32130	0.20164

(b)

Table 5.3: Results obtained by the application of the Spectrogram NN with the best set of parameters for each test bearing and each kind of version (a). Average of the metrics grouped by version (b). The MAE_L is the Mean Absolute Error computed with the last five hundred samples and finally *DIFF* is the average between the real and the predicted RUL of the last ten samples. The *score* is a weighted average of the previous three metrics.

this kind of model presents consistent gaps between the performances obtained by each single version of Bearing1_4, Bearing1_5, and Bearing1_7. Regarding the latter, the predictions with an architecture that contains also ConvLSTM and MaxPooling layers are much worse with respect to the standard configuration. It is also true that in this case, Bearing1_7 is the only test set that reaches a lower score with the normal version. In this sense, this REB is penalized, because the choice of whatever configuration except the first one degrades a lot the final accuracy. However, in the view of ensuring that the predictive maintenance solution tries to predict well all the test bearings, some results are inevitably more accurate and, otherwise, some predictions lose precision. Chapter §7 will discuss more in detail this behavior, which is common for various models. and it will also present some possible solutions to this kind of issue.

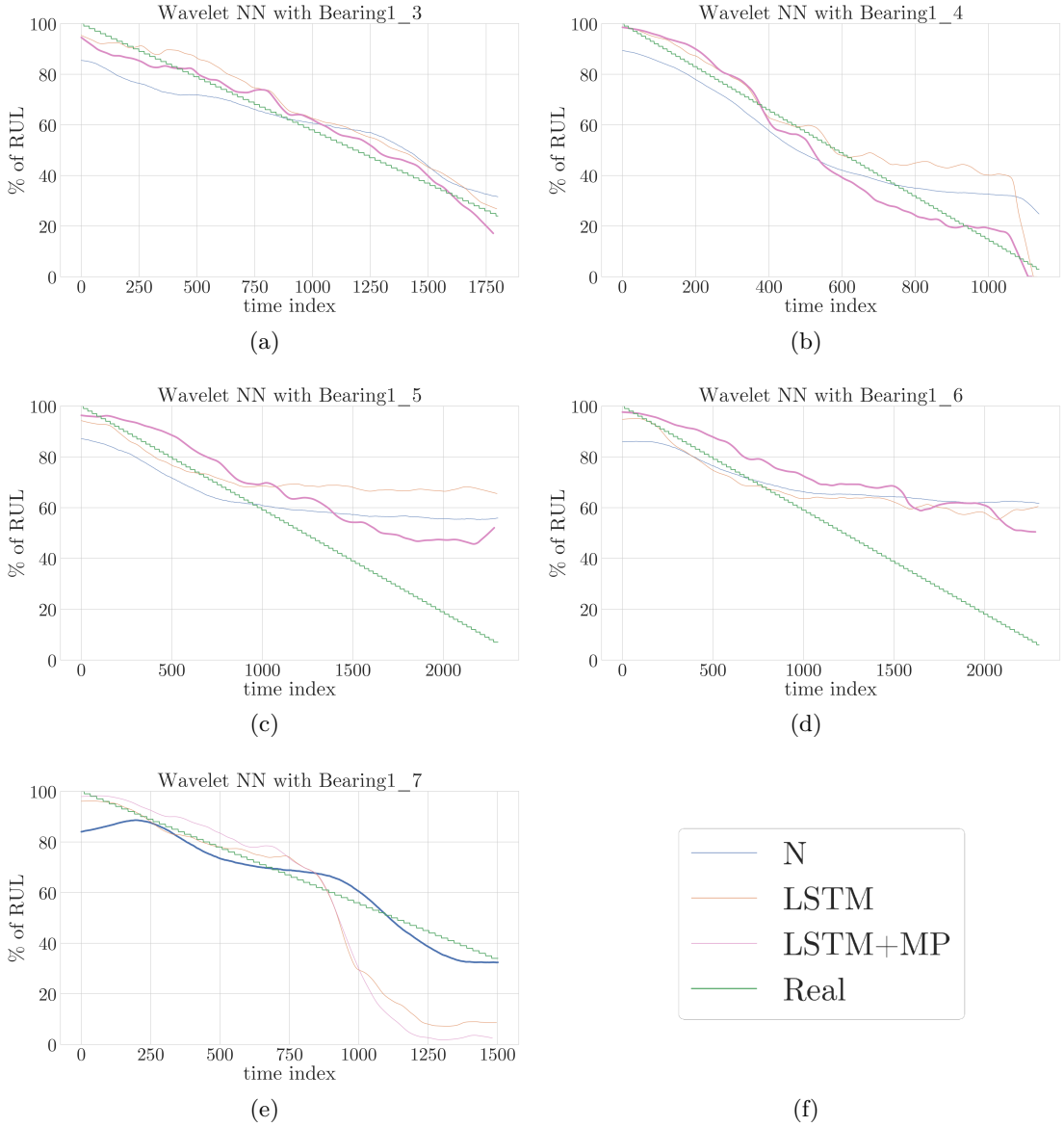


Figure 5.5: RUL Prediction of each test bearing using the Wavelet Neural Network with different versions of architectures.

5.3.3 Mixed NN

Similarly to the previous neural network, the Mixed NN performs on average well with the *LSTM+MP* configuration. However, as it can be seen in Figure 5.5b, the average scores are very similar to each other. In particular, the architectures which contain a ConvLSTM layer have an average score which differs of 0.00001. This is because they have a close MAE, MAE_L and DIFF contemporarily. Nevertheless, for the aforementioned reasons,

Fig	Bearing	Version	MAE	MAE_L	DIFF	Score
5.5a	Bearing1_3	N	0.07943	0.04495	0.06725	0.06591
		LSTM	0.05637	0.05971	0.02865	0.05286
		LSTM+MP	0.03986	0.03664	0.06531	**0.04303**
5.5b	Bearing1_4	N	0.13424	0.11141	0.38308	0.16810
		LSTM	0.08694	0.16143	0.03200	0.10261
		LSTM+MP	0.05323	0.05325	0.03200	**0.04970**
5.5c	Bearing1_5	N	0.17797	0.38957	0.50521	0.30304
		LSTM	0.20972	0.50343	0.58609	0.37035
		LSTM+MP	0.15198	0.30723	0.44754	**0.25299**
5.5d	Bearing1_6	N	0.20152	0.45298	0.56167	0.34536
		LSTM	0.17157	0.41862	0.54357	0.31592
		LSTM+MP	0.20685	0.41097	0.44458	**0.31451**
5.5e	Bearing1_7	**N**	0.06820	0.08615	0.02172	**0.06644**
		LSTM	0.12899	0.31366	0.25400	0.21138
		LSTM+MP	0.16187	0.36358	0.31375	0.25442

(a)

Version	MAE	MAE_L	DIFF	Score
N	0.13227	0.21701	0.30779	0.18977
LSTM	0.13072	0.29137	0.28886	0.21063
LSTM+MP	0.12276	0.23433	0.26064	**0.18293**

(b)

Table 5.4: Results obtained by the application of the Wavelet NN with the best set of parameters for each test bearing and each kind of version (a). Average of the metrics grouped by version (b). The MAE_L is the Mean Absolute Error computed with the last five hundred samples and finally *DIFF* is the average between the real and the predicted RUL of the last ten samples. The *score* is a weighted average of the previous three metrics.

the version which is more accurate on the higher number of bearings is the normal, even if the average score obtained by this configuration is the worst for this kind of architecture. The motivation behind this fact is that, as for the results described above, on Bearing1_5 and Bearing1_6, version *N* is more inaccurate, either regarding the MAE and the final predictions embodied in the metric DIFF.

5.4 Comparisons

An ideal predictive maintenance tool has to try to predict, as better as possible, all the bearings under analysis. In order to reach this goal, a collection of possible solutions can be implemented. In this section, only one of such solutions will be finely described, due to the amount of time requested to assess also other possible implementations. Moreover, Chapter §7 will take into account a possible valid alternative to what will be presented in

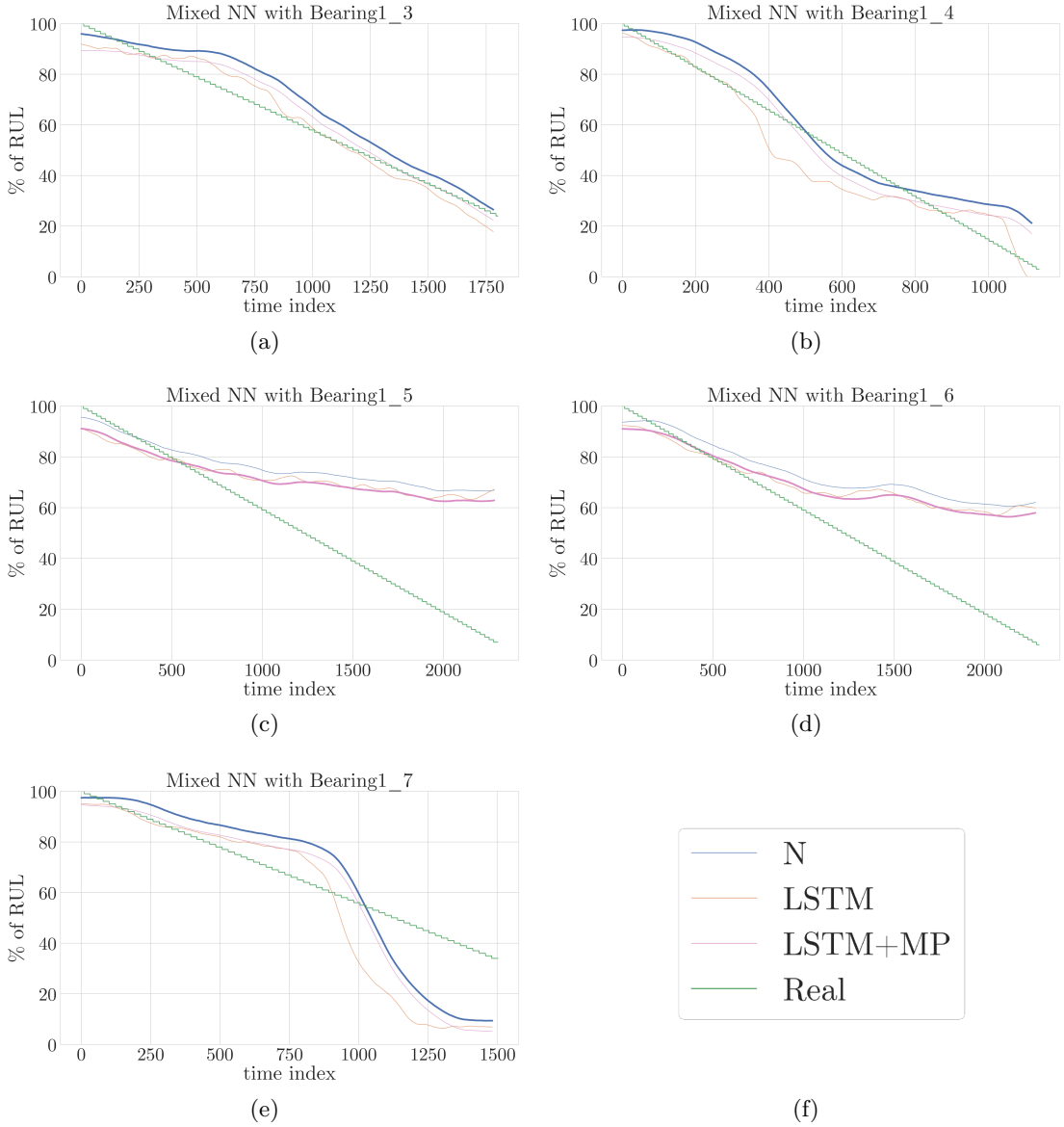


Figure 5.6: RUL Prediction of each test bearing using the Mixed Neural Network with different versions of architectures.

this subchapter.

The basic idea behind the definition of a complete predictive maintenance tool is the choice of the best model between machine learning and deep learning. In addition, inside each of these categories, also the most performing algorithm has to be chosen. The following two subchapters will make some comparisons between firstly the SVR and the RFR, and secondly between the Spectrogram NN, the Wavelet NN, and the Mixed NN.

Fig	Bearing	Version	MAE	MAE_L	DIFF	Score
5.6a	Bearing1_3	**N**	0.04517	0.01724	0.03374	**0.03395**
		LSTM	0.03793	0.02957	0.05992	0.03881
		LSTM+MP	0.04134	0.04626	0.07347	0.04833
5.6b	Bearing1_4	**N**	0.06496	0.06649	0.00465	**0.05542**
		LSTM	0.06213	0.06083	0.03200	0.05668
		LSTM+MP	0.06808	0.06328	0.05334	0.06402
5.6c	Bearing1_5	N	0.23471	0.50190	0.61927	0.38787
		LSTM	0.21996	0.47802	0.60049	0.36940
		LSTM+MP	0.21140	0.46144	0.58147	**0.35642**
5.6d	Bearing1_6	N	0.20252	0.45292	0.56250	0.34598
		LSTM	0.18449	0.42896	0.53932	0.32512
		LSTM+MP	0.17565	0.41219	0.52324	**0.31243**
5.6e	Bearing1_7	**N**	0.14200	0.28818	0.24985	**0.20870**
		LSTM	0.13850	0.31261	0.27088	0.21860
		LSTM+MP	0.14129	0.32800	0.28434	0.22737

(a)

Version	MAE	MAE_L	DIFF	Score
N	0.13787	0.26535	0.29400	0.20638
LSTM	0.12860	0.26200	0.30052	0.20172
LSTM+MP	0.12755	0.26223	0.30317	**0.20171**

(b)

Table 5.5: Results obtained by the application of the Mixed NN with the best set of parameters for each test bearing and each kind of version (a). Average of the metrics grouped by version (b). The MAE_L is the Mean Absolute Error computed with the last five hundred samples and finally *DIFF* is the average between the real and the predicted RUL of the last ten samples. The *score* is a weighted average of the previous three metrics.

Finally, after having found out the best machine learning and deep learning model, the last part of this section is devoted to comparing the two previous models, taking into account also the times required for training such models.

5.4.1 Machine Learning

Inside the category of machine learning, the only paragon is done between the performances obtained by the SVR and the RFR. In order to proceed with the comparison, it could be useful to reiterate the versions which perform better for both models: the SVR makes use of a dataset with all the features extracted in Subchapter §4.3.1, instead the RFR with only the features selected in Subchapter §4.3.2.

Regarding the score, as reported in Table 5.6a, for three-fifths of bearings the RFR has a value tightly lesser than the SVR. This is valid for Bearing1_3, Bearing1_4, and

Bearing1_7. This behavior is also confirmed by looking at Table 5.6b, where the average score of the Random Forest is lower more or less of only 0.01 points of score. The major differences between the two models are visible in Bearing1_4, Bearing1_5, and Bearing1_7. The first one is characterized by a more accurate final prediction. Figure 5.7b shows how for the SVR in the final samples the bearing increments its life, with respect to the outcome of the RFR, which follows precisely the actual RUL. The second one has a final similar prediction but a different MAE, especially around time index 1000. Finally, it is possible to make the same considerations for Bearing 1_7: in this case, the RUL predicted by the SVR is better compared to the one predicted by the RFR between the time index 250 and 1000. However, as the score takes into account also the last five hundred samples and the final ten gaps between the actual and the predicted remaining useful life, the score of the RFR is better also in this case.

Fig	Bearing	Algorithm	MAE	MAE_L	DIFF	Score
5.7a	Bearing1_3	SVR	0.09504	0.07766	0.01769	0.07635
		RFR	0.07484	0.05080	0.00266	**0.05480**
5.7b	Bearing1_4	SVR	0.12560	0.18462	0.15984	0.15098
		RFR	0.11114	0.19529	0.00051	**0.12075**
5.7c	Bearing1_5	**SVR**	0.06885	0.09473	0.34792	**0.12399**
		RFR	0.13673	0.10135	0.25002	0.14382
5.7d	Bearing1_6	**SVR**	0.14821	0.34631	0.44509	**0.26373**
		RFR	0.16665	0.40059	0.46100	0.29369
5.7e	Bearing1_7	SVR	0.08538	0.11768	0.17272	0.11070
		RFR	0.09446	0.07029	0.05063	**0.07910**

(a)

Algorithm	MAE	MAE_L	DIFF	Score
SVR	0.10462	0.16420	0.22865	0.14515
RFR	0.11676	0.16366	0.15296	**0.13843**

(b)

Table 5.6: Comparison between the results reached by the best versions of the machine learning models grouped by test bearing (a) and model (b).

5.4.2 Deep Learning

The situation reported in Table 5.7a differs with respect to the one of Table 5.6a, as in this case all the test datasets, except for Bearing1_7, reach a lower score with the Wavelet NN. It is also true that, by observing Table 5.7b, the average score of the Spectrogram NN and the Mixed NN are similar, as they differ only for 0.0001 points of MAE. However, it is also evident that the metrics MAE_L and DIFF are better if the prediction is made by the Wavelet NN. A particular situation, which has been introduced in the previous subchapters, regards Bearing1_7. Despite Figure 5.5e depicts how it is possible, with

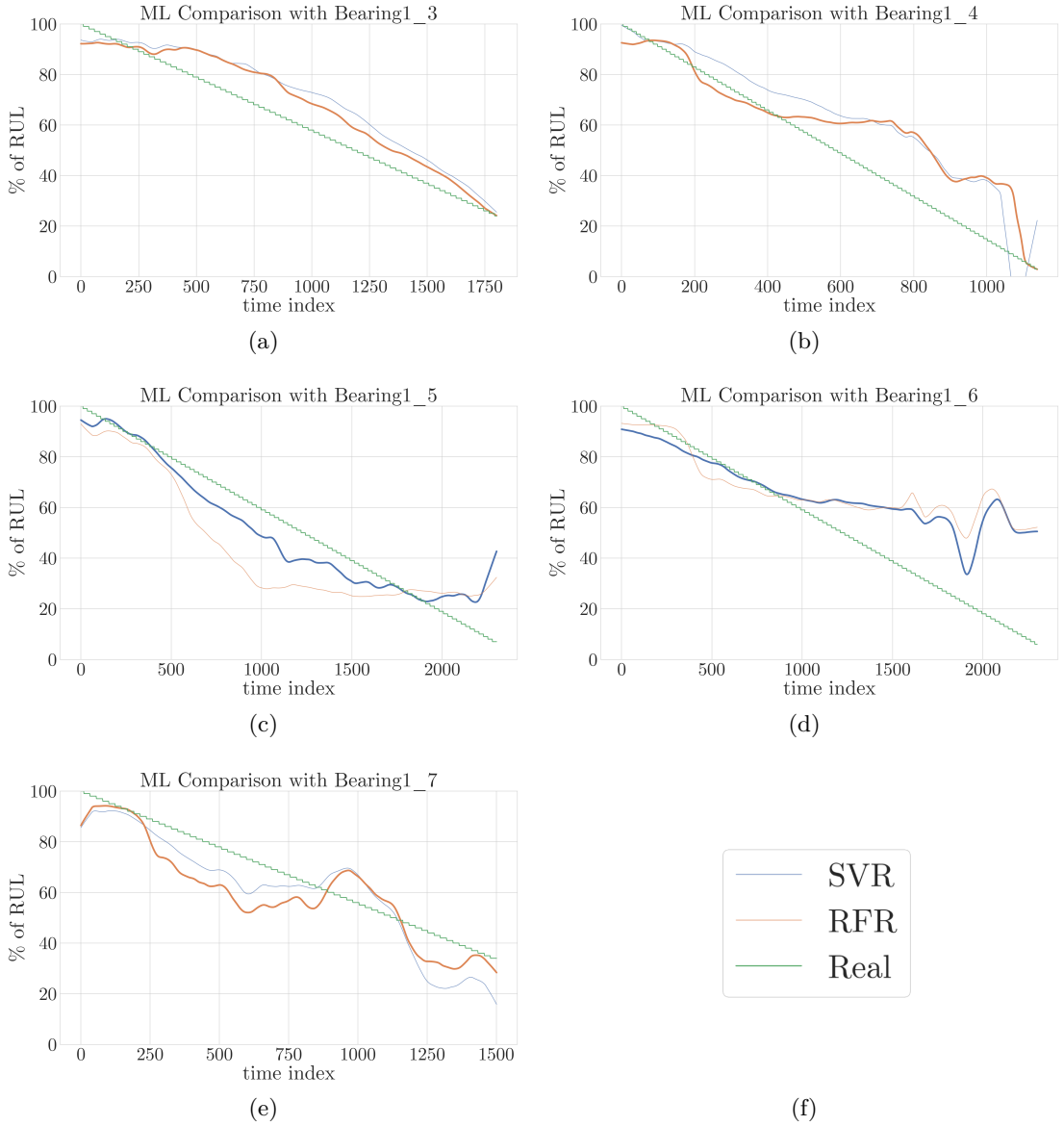


Figure 5.7: Comparison between the best-performing version of each machine learning model.

all the models explained so far, to predict an RUL for this component with a relatively low mean absolute error, the model selection steps based on a hierarchical pruning of the regressors with a high average score penalizes this component in favor of Bearing1_5 and Bearing1_6. Instead, regarding Bearing1_3 and Bearing1_4 the choice of an architecture respect than another does not change significantly the final result.

Fig	Bearing	Architecture	MAE	MAE_L	DIFF	Score
5.8a	Bearing1_3	S	0.04987	0.07802	0.08043	0.06435
		W	0.03986	0.03664	0.06531	**0.04303**
		M	0.04134	0.04626	0.07347	0.04833
5.8b	Bearing1_4	S	0.06688	0.11698	0.19632	0.10515
		W	0.05323	0.05325	0.03200	**0.04970**
		M	0.06808	0.06328	0.05334	0.06402
5.8c	Bearing1_5	S	0.18301	0.43761	0.54994	0.32903
		W	0.15198	0.30723	0.44754	**0.25299**
		M	0.21140	0.46144	0.58147	0.35642
5.8d	Bearing1_6	S	0.19075	0.45921	0.58441	0.34584
		W	0.20685	0.41097	0.44458	**0.31451**
		M	0.17565	0.41219	0.52324	0.31243
5.8e	Bearing1_7	**S**	0.10314	0.23914	0.19538	**0.16385**
		W	0.16187	0.36358	0.31375	0.25442
		M	0.14129	0.32800	0.28434	0.22737

(a)

Architecture	MAE	MAE_L	DIFF	Score
Spectrogram	0.11873	0.26619	0.32130	0.20164
Wavelet	0.12276	0.23433	0.26064	**0.18293**
Mixed	0.12755	0.26223	0.30317	0.20171

(b)

Table 5.7: Comparison between the results reached by the best versions of the various neural network architectures grouped by test bearing (a) and model (b).

5.4.3 Overall

After having decreed which is the most performing machine learning model and the most performing neural network, a final paragon has to be made in order to define the regressor which is able to make, on average, better predictions. However, in this case, it is not possible to prefer one model only observing the value of the score. In fact, it is well-known that a deep learning model requires some time to be trained, which usually is extremely higher with respect to the time required by a machine learning model. For this reason, Table 5.9 reports either the hardware used to train such regressor and the training times for each machine learning and deep learning model. In addition, it is important to highlight a particular consideration: the hardware presented in Table 5.9a is the one made available by Google to all the developers that use their Colaboratory platform which, with respect to a traditional personal computer, is characterized by a higher computational power.

Regarding the differences in terms of performance, Table 5.9a reports the results obtained by the best machine learning model, the Random Forest Regressor, and the best

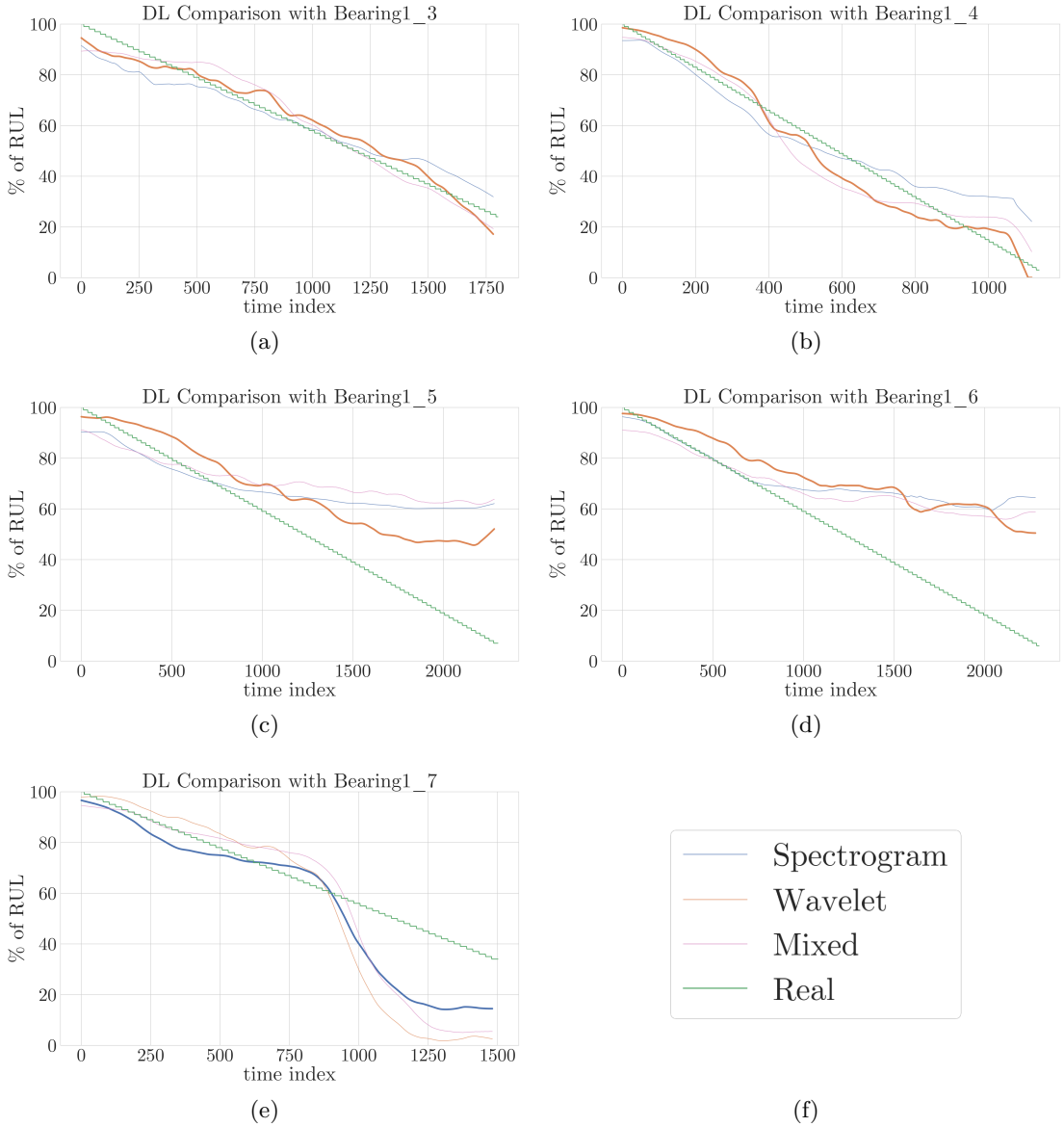


Figure 5.8: Comparison between the best-performing version of each neural network architecture.

neural network, the Wavelet NN. Subsequently, the score of the two regressors is similar only for Bearing1_3 and Bearing1_6. Instead, Bearing1_4, Bearing1_5, and Bearing1_7 are characterized by the fact that or the RFR or the WNN have a score that is much lower compared to the other. In particular, the Bearing1_4 is the only which is predicted better by the WNN. For these reasons, the average score reported in Table 5.9b is in favor of the RFR, mainly because this model is more accurate respect the WNN in the last part

of the aforementioned bearings.

In conclusion, based on all what is described in this chapter, the most performing model is the Random Forest Regressor. A possible reasoning behind the lower accuracy represented by the Wavelet NN is the number of samples used for its training. It is consolidated in the literature that a neural network needs, in order to obtain better results with respect to the machine learning models, a great amount of data. It is dutiful to remember that the dataset used for this work relies on a number of training bearings which is less than half of the total number of test bearings. Moreover, as already introduced in Subchapter §5.1, two of the testing bearings have a trend that is not even close to the behavior of bearings belonging to the training set. However, this issue will be also discussed in Chapter §7.

Fig	Bearing	Model	MAE	$\mathbf{MAE_L}$	DIFF	Score
5.9a	Bearing1_3	RFR	0.07485	0.05080	0.00266	0.05480
		Wavelet	0.03986	0.03664	0.06531	**0.04303**
5.9b	Bearing1_4	RFR	0.11192	0.19529	0.00051	0.12114
		Wavelet	0.05323	0.05325	0.03200	**0.04970**
5.9c	Bearing1_5	**RFR**	0.13728	0.10135	0.25002	**0.14409**
		Wavelet	0.15198	0.30723	0.44754	0.25299
5.9d	Bearing1_6	**RFR**	0.16754	0.40059	0.46100	**0.29413**
		Wavelet	0.20685	0.41097	0.44458	0.31451
5.9e	Bearing1_7	**RFR**	0.09420	0.07029	0.05063	**0.07897**
		Wavelet	0.16187	0.36358	0.31375	0.25442

(a)

Model	MAE	$\mathbf{MAE_L}$	DIFF	Score
RFR	0.11716	0.16366	0.15296	**0.13863**
Wavelet	0.12276	0.23433	0.26064	0.18293

(b)

Table 5.8: Comparison between the results reached by the best machine learning and deep learning model, grouped by test bearing (a) and model (b).

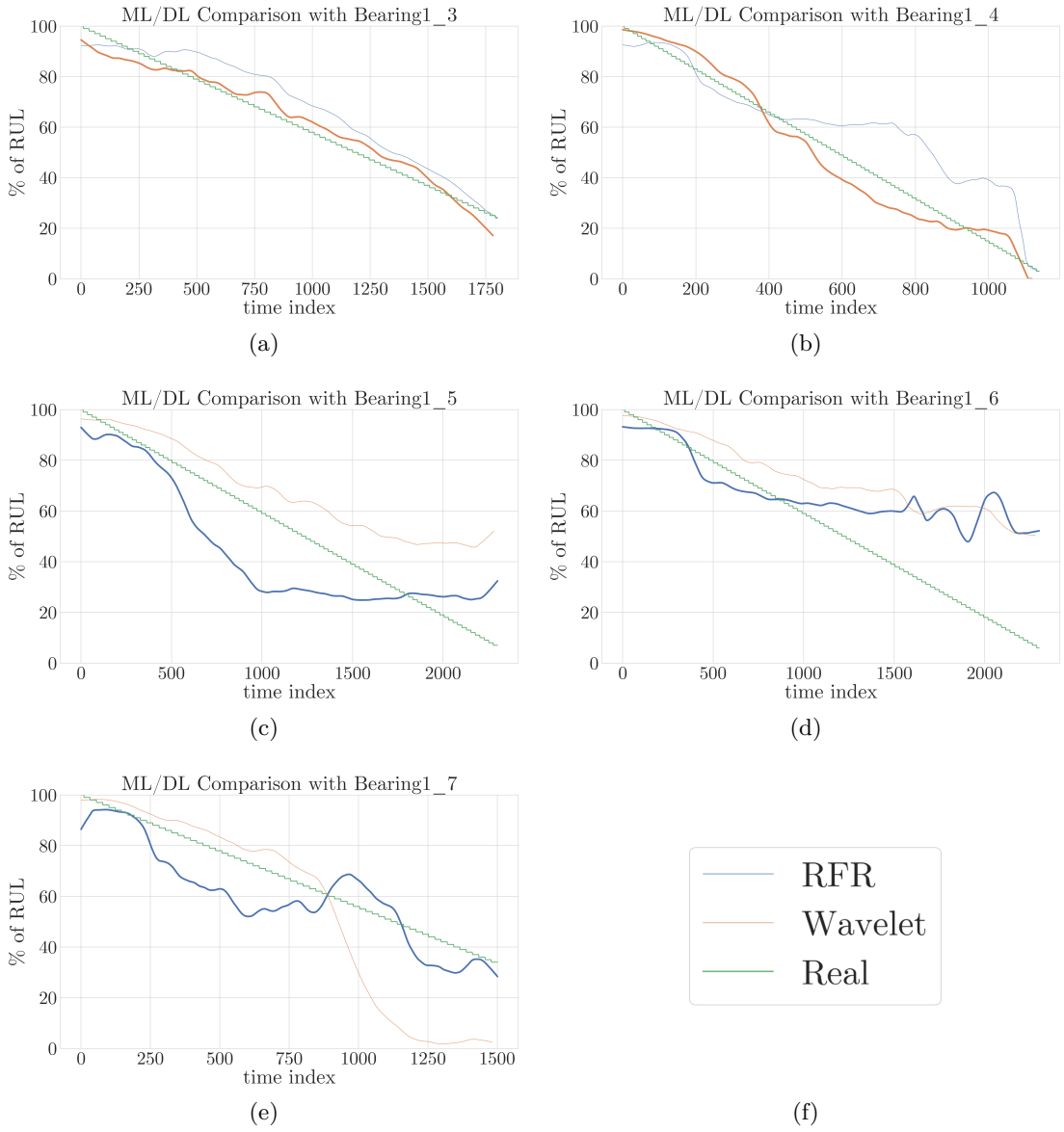


Figure 5.9: Comparison between the best machine learning and deep learning model.

Component	Specifications	
	Feature	**Detail**
CPU	Intel(R) Xeon(R)	
	Socket(s)	1
	Core(s) per socket	1
	Thread(s) per core	2
	L3 cache	39424 KB
	CPU	2000.178 MHz
GPU	NVIDIA Tesla K80	
	CUDA Cores	2496
	VRAM	12 GB
	Type	GDDR5
RAM	13 GB	
Disk	33 GB	

(a)

ML/DL	Model	Dataset Time	Training Time	Total
ML	SVR	240s	49s	289s
	RFR	240s	35s	275s
DL	Spectrogram	330s	70s/epoch	100 min/run
	Wavelet	350s	83 s/epoch	120 min/run
	Mixed	440s	100s/epoch	145 min/run

(b)

Table 5.9: Hardware specifications of a Google Colaboratory instance (a). List of all the time required for generating the dataset and training the model (b). **Note**: each combination of hyper-parameters affects the training time. For this reason, the *time* is an average over all the possible grid configurations.

Chapter 6

Cost-Benefit Analysis

The paradigm of industry 4.0 can completely change the productivity of an industry for the better. On the other side, each single case study is particular and dependent on the context where the company is situated. For this reason, it is not possible to define a general threshold that can be used to decide if the digitalization of the entire production chain will return considerable advantages in terms of profit or not. It is necessary to evaluate case by case the situation of each industry, in order to design the best collection of strategic investments for such reality.

This basic cost-benefit analysis tries to highlight the possible considerations that must be taken into account when the company management assesses this kind of decision. Preventive maintenance and diagnostic methods, normally outsourced to other firms, can reach considerable levels of reliability and performance, but their major disadvantages are the extreme amount of consumable used, the store of a certain number of components, and the lack of certainty due to a bearing that should be repaired but that it is not part of the list of maintenance. In addition, diagnostic and prognosis methods share more or less the same sunk costs: the implementation of an IoT infrastructure that is able to analyze time by time the status or to predict the RUL of a component is very similar for both case studies. Probably, direct costs like manpower and indirect costs like overhead can increase from diagnosis to prognosis, despite the most pronounced differences in terms of money spent for the run-up phase rely on the materials heading. For this reason, it is possible to assert that the boundary between an *old-fashioned* industry configuration and a completely digitalized one is represented by the separation between the preventive maintenance and the diagnosis/prognosis methods. In this sense, all the considerations that will be further made in this paragraph will take into account the transition from the *run-to-break* or the preventive maintenance to the predictive maintenance. Preventive maintenance and *run-to-break* have in common the possible failure scenarios which can occur during the production. The application of one strategy rather than the other can only influence the gravity of the resulting failures but it can not avoid them, which is only guaranteed by predictive maintenance.

The clear advantages provided by the implementation of the industry 4.0 paradigm have to be compared to the effective costs of such solution. This rapid analysis aims to equate the loss of profit due to the stop of production and the costs linked to the

digitalization of the industry. The report is based on plausible data which come from a real textile manufacturer. However, for the sake of clarity, a set of important notes has to be highlighted: firstly, to respect the industrial secret of this company, real values have been rounded up or down in a not predictable manner, in such a way that the final considerations remain more or less valid despite the veracity of the outcomes. Secondly, this analysis is based only on a production segment of this company, in particular on the bleaching of cotton. Thus, the benefits and the costs are compared to the loss of production generated by a set of failures of the bleaching machine. It is obvious that it is not possible to assert that the raw material and the finished products depend only on single machinery, but it is also true that the way to proceed is the same even in the case of more complex systems. In such situations, it is sufficient to explode the analysis for all the production parts to obtain a final cumulative evaluation.

The starting point is represented by the calculation of the Earning Before Interests, Taxes and Amortization (EBITA)[1], which can be used as a basic estimator of the value of production. Normally, this balance sheet heading is referred to a financial year but in this case, in order to compare it to the hours needed to solve the various failures, is referred to one hour of production. The EBITA is defined as the total value of revenue generated by the operative activities, to the net of the depreciation but to the gross of amortization: this report will not take into account borrowing costs, financial income, and taxes. These headings mainly depend on the management and on the country where the company is located. Thus, in this context, they would only weigh down the analysis without any informative gain. The calculation of the EBITA pass through the definition of the Earning Before Interests, Taxes, Depreciation and Amortization (EBITDA)[2], which, as suggests the name, is the EBITA before depreciation and it can be computed by subtracting to the total value of production the external costs, the cost of raw materials and the cost of personnel. The value of production and the total costs depend on different factors: on one side the total income relies on the value of the finished product and on the other side costs depend on the raw materials, manpower, overhead, and tangible fixed assets. In order to simplify the analysis, the total income is based on the selling price of reels of bleached cotton.

Table 6.1b reports the calculation to derive the unit price for square meter of a reel of bleached cotton, exploiting its total price and its dimensions. Table 6.1a reports the same information for the raw material. However, in this case, the price is computed by analyzing the market quotation of such feedstock, which is expressed as $/pound. Then, after some currency conversion and some trivial physical computation, it is possible to obtain the unit price for cubic meter[3]. In this sense, it is fundamental one clarification: the unit price of raw cotton and the unit price of bleached cotton is expressed with two different space measures, the former in square meters and the latter in cubic meters. However, it is also true that the thickness of the fabric is useless for the bleaching machine, which production

[1]It can be translated into Italian as MON, Margine Operativo Netto.

[2]It can be translated into Italian as MOL, Margine Operativo Lordo.

[3]These prices are VAT excluded.

Feature	Value	Unit
Volume	0,0003	m^3
Density	40	kg/m^3
Weight	0,012	kg
Price	121,12	€/kg
Unit Price	1,4534	€/m^3

(a) Price of raw cotton.

Feature	Value	Unit
Width	0,7	m
Height	100	m
Surface	70	m^2
Price	187,68	€
Unit Price	2,68	€/m^2

(b) Price of bleached cotton.

Table 6.1: Prices of raw material and finished product.

is computed for surfaces of textile and, practically, for lengths of reels. For this reason, the unit price of raw cotton is treated as defined for a single square meter. Consequently, assuming that the textile in input to a bleaching machine is raw and the output fabric is finished and it could be immediately palletized, the value of production can be computed by multiplying the hourly production of this kind of machinery, reported in the first raw of Table 6.4, for the unit price of the bleached cotton reported in Table 6.1b. Subsequently, from the total income have to be subtracted the various costs items. These are formed out of the sum of the cost of raw material, which can be computed as for the value of production, and the external and personnel costs. The external costs contain the cost of energy and the overhead. The former is obtained by multiplying the estimated hourly power consumption of a bleaching machine, 12 KWh, for the average kilowatt cost equal to 0.38 €/KWh. The latter is a standard value that brings together general expenses. The personnel costs are reported in Table 6.2, where there are specified for each different position the correspondent net annual salary, the number of human resources, and the final cumulative salary for each position. Then, this total annual cost is divided by the total work hours in 12 months, assuming 24 workdays per month and 8 work hours per day, in order to compensate the holiday periods. The resulting value is the EBITDA, which is reported in Table 6.4.

Position	Net Salary	Quantity	Total Cost
Management	80.000 €	3	240.000 €
Commercial	30.000 €	3	90.000 €
Engineers	40.000 €	10	400.000 €
Skilled Workers	26.000 €	7	182.000 €
Employees	20.000 €	25	500.000 €

Table 6.2: List of the human resources with their corresponding salary and quantity.

The last step consists of computing the EBITA that, as already introduced before, is obtained by subtracting the depreciations from the EBITDA. A possible list of tangible fixed assets are reported in Table 6.3, where for each asset is specified the price, the

number of years of depreciation, and the final cost per hour[4]. The investments in assets are normally depreciated on a typical useful lifetime period, except for the human capital which follows different rules.

Asset	**Price**	**Unit**	**Depreciation**	**Cost per Hour**
Employees	1.412.000	€/y	-	612,85 €
Bleaching Machine	1.250.000	€	10 y	54,25€
Industrial Plant	4.000.000	€	20 y	86,81€
IT Infrastructure	230.000	€	5 y	8,68 €
Cars and Trucks	100.000	€	3 y	33,28 €
Furniture	40.000	€	5 y	3,47 €
Lifting Equipment	260.000	€	10 y	11,28 €

Table 6.3: List of possible assets with their depreciation and unit cost per hour of production.

For this reason, the bleaching machine is assumed to have a useful life of 10 years. In the market, it is possible to find used machinery at half of the cost and with a past of 15 years of production. However, even if these machines are good for 20 years, a decade is sufficient to split their cost. Over this threshold, this asset has no budgeted costs. Thus, all the production hours that exceed 10 years are, in a broad sense, free. One of the objectives of RECLAIM is to lead machines to their complete exhaustion and the causes rely on this particular situation. A bleaching machine that has to be substituted before a decade is a double sunk cost because it is necessary to buy another asset before the complete depreciation of the preceding. On the other hand, machinery that can be productive after a decade represents only an advantage for the company, because it is possible to slit the purchase of new machinery or, even better, to parallelize the production between the new asset and the old one. Clearly, there will be differences in terms of efficiency, but in this way, the cumulative productivity is higher compared to relying on a single machine, at least for a period. The industrial plant is assumed to be depreciated in 20 years, which is also a plausible time period for a mortgage. A possible alternative, which has been considered, is the lease. In this case, the rent of a plant is recurrent, normally monthly, cost that however it is not an asset because it does not influence the patrimonial state. The IT infrastructure is accounted for 5 years, due to the speed of the current technological advancement. Cars and Trucks can be depreciated in 3 years but this value has the same limits as the bleaching machine because it depends on the annual kilometers traveled which could lead to an increase of the useful life. In addition, this value is recurrent in the long-term rental[5] contracts because it is expected that after 3 years it is not further convenient to maintain the vehicle. Finally, furniture is accounted for 5 years and the lifting equipment, which is indispensable to transport

[4]Also in this case are assumed 12 work months, 24 workdays per month and 8 work hours per day.

[5]Long-term rental is an alternative to the cars and trucks purchase. The reasons are the same of the industrial plant.

internally the cotton reels and the finished pallets, is accounted for 10 years, similarly to the bleaching machine since this is composed of mechanical components too.

Heading	**Value**	**Unit**
Hourly Production	9.025,92	m^2/h
+ Value of Production	24.199,78	€/h
- Cost of Raw Materials	13.118,63	€/h
- External Costs	6.004,94	€/h
- Cost of Personnel	612,85	€/h
= **EBITDA**	4.463,36	€/h
- Depreciations	197,77	€/h
= **EBITA**	4.265,59	€/h

Table 6.4: List of possible assets with their depreciation and unit cost per hour of production.

At this point, it is possible to compute the EBITA by subtracting the total amount of depreciation, in this case referred to a single work hour, from the EBITDA. Such value can be used from now on to quantify the loss of production in terms of euro per hour, gross of the financial profits, costs, and taxes. In a certain sense, the EBITA is the key data on which it is possible to build the rest of the CBA. In order to take into consideration different situations, a set of possible failure scenarios regarding the bleaching machine are considered. Each scenario is characterized by a level of gravity, minor, moderate, and catastrophic, linked to the percentage of fault bearings in the same moment. This can occur easily because, as described in the summary and in the introduction, a broken REB could damage also the mechanical components nearby, which could be in turn bearings, and so forth. Thus, given the list of all the REBs contained in the bleaching machine, which are more than 1200, the four most critical and most sensitive models of bearings are used to define the aforementioned three damage scenarios. The *minor* situation relies on the 10% of bearings, the moderate situation on the 20% and, finally, the catastrophic situation on the 40%.

The number of REBs under control influences all the correlated costs and benefits reported in Table 6.1. Starting from the latter, the number of hours dedicated to the maintenance depends on the number of broken components. In this sense, a more serious scenario leads to higher costs, either in terms of loss of production and in terms of manpower. In Table 6.1a, for every single scenario, a set of data is propaedeutic for the definition of what was mentioned before. The loss of production is determined by multiplying the EBIT, defined here per hour, for the total number of hours needed to restart the bleaching machine. The consumables are computed by multiplying the number of requested REBs for their unitary cost, depending on the model. Instead, the labor cost is obtained by treating these work hours as overtime, in order to remedy the incident as soon as possible. The total benefit can be derived by assigning a probability to each situation. This step, normally, could be the result of a statistical analysis that determines the relative frequencies of each scenario on the total number of events. In this CBA, these percentages are assumed to be as conservative as possible, by assigning a higher

probability to the *minor* and *moderate* scenarios and a lower chance to the most critical one.

Regarding the costs incurred in the implementation of an architecture that can digitalize an industry, it is possible to define three different categories: the costs for all the transducers involved in the analogic-to-digital conversion, the costs for the work needed to install those devices, and the software and hardware which perform the analysis.[6] Regarding the first and the most expensive category, a lot of money are invested in the purchase of a professional accelerometer[7], with its wire, and the number of DAQ cards required for this task, which depends on the total number of channels. Therefore, by dividing the price of a professional analogic-to-digital converter by its number of inputs, it is possible to estimate the cost of each channel which is subsequently multiplied by the number of bearings. The second category takes into account the total number of hours that have to be dedicated to the implementation of this architecture. The single work value is an average of the time needed to install a single sensor, to pull the cable from the bleaching machine to the DAQ rack, and to configure the predictive maintenance software. Finally, the costs of the engineering solution, either in terms of software and hardware, complete this part of CBA.

The benefits generated by an industry 4.0 architecture are generally double the costs incurred in the predisposition of such solutions. What is absent in Table 6.1a is the time period interested by the failures, which consequently affects the Return of Interests (ROI), i.e. a financial metric used for measuring the profitability of an investment. A general method to compute this value is the following:

$$\text{ROI} = \frac{\text{Return on Investment}}{\text{Cost of Investment}} * 100 \tag{6.1}$$

However, equation 6.1 cannot be used in this context because it involves equal cash flows. Instead, the total benefit value is not split uniformly in five years. In fact, it is possible to assume that each year the company has an economical damage equal to a percentage of 214.090,42 €. In this case, it is necessary to compute the ROI using the Internal Rate of Return (IRR) function, which can be computed with equation 6.2.

$$\sum_t \frac{C_t}{(1+\text{IRR})^t} = 0 \tag{6.2}$$

where C_t is the cash flow of period t.

Consequently, assuming a subdivision of the total value of benefits as reported in Table 6.5, the value of the IRR becomes equal to 11,41%. Such percentages follow a specific reasoning: in the first two years, it is more probable that will happen only *minor* or *moderate* scenarios, taking into account also the success percentages reported in Table

[6] In this case the hardware is not referred to the set of sensors, wires, and DAQ cards, but it is inherited to the upgrade of the existent IT infrastructure already accounted in the list of assets reported in Figure 6.3.

[7] The price of one accelerometer is considered equal to 260 € + VAT. The model of such sensor is Dytran 3035B, the same used in the PRONOSTIA platform.

Benefits		**Scenarios**		
		Minor	**Moderate**	**Catastrophic**
% of bearings		10	20	40
Coverage		80	160	321
Data	Single Work	0,3 h		
	Total Hours	24	48	96
Loss of Production		102.630,06 €	205.260,12 €	410.520,24 €
Maintenance	6013 2Z/C3	1.995,00€	3.990,00 €	7.980,00 €
	INA GYE 30 KLLHB	3.281,00 €	6.562,00 €	13.124,00 €
	INA GYE 30 KRRB	1.185,00 €	2.370,00 €	4.740,00 €
	RHP 1050-50 G	338,00 €	676,00 €	1.352,00 €
	Manpower	360,90 €	721,80 €	1.443,60 €
	Total	7.159,90 €	14.319,80 €	28.639,60 €
Costs		109.789,96 €	219.579,92 €	439.159,84 €
Probability		45%	35%	20%
Benefit		214.090,42 €		

(a) Possible list of benefits of industry 4.0.

Costs		**Scenarios**		
		Minor	**Moderate**	**Catastrophic**
% of bearings		10	20	40
Coverage		80	160	321
Hardware	6013 2Z/C3	10.374,00 €	20.748,00 €	41.496,00 €
	INA GYE 30 KLLHB	5.018,00 €	10.036,00 €	20.072,00 €
	INA GYE 30 KRRB	4.018,00 €	8.216,00 €	16.432,00 €
	RHP 1050-50 G	1.352,00 €	2.704,00 €	5.408,00 €
	DAQ [€/channel]	150		
	DAQ	12.030,00 €	24.060,00 €	48.120,00 €
	Total	32.882,00 €	65.764,00 €	131.528,00 €
Manpower	Single Work	0,5 h		
	Total Hours	40	80	160
	Total	601,50 €	1.203,00 €	2.406,00 €
Infrastructure	Software	10.000,00 €		
	HW Upgrade	3.000,00 €		
	Total	13.000,00 €		
Cost		46.483,50 €	79.967,00 €	146.934,00 €
€/h*		4.04	6.94	12.75

(b) Costs incurred for the predisposition of a digitalized industry.

Figure 6.1: Detailed benefits derived by the implementation of an Industry 4.0 architecture and its corresponding costs. *: with depreciation in 5 years.

	Cash Outflow	**Benefit Percentage**	**Cash Inflow**
Year 0	(143.934,00 €)	-	-
Year 1	-	12%	25.690,85 €
Year 2	-	12%	25.690,85 €
Year 3	-	16%	34.254,47 €
Year 4	-	25%	53.522,61 €
Year 5	-	35%	74.931,65 €
Total	(143.934,00 €)	100%	214.090,42 €
IRR		11,41 %	

Table 6.5: Calculation of ROI with unequal yearly cashflows.

6.1a. More in detail, if the random variable $X_{\{min,mod,cat\}}$ is a Bernoulli distribution which describes if a *minor*, a *moderate* or a *catastrophic* scenario happen or not, the probability that the same scenario happen after three, four or five years, can be modeled as a Binomial distribution. Consequently, it is possible to assert that, over five years, it is more probable that the more serious scenarios happen later with respect to the less critical ones. For this reason, the amount of benefit associated with the first two years is lower compared to the other values.

The resulting IRR is equal to 11,41%, which indicates that the CBA is in favor of a digitalized production chain. In conclusion, in the long-term, the investment made in the implementation of the paradigm of industry 4.0 is profitable, leading the company to gain a competitive advantage compared to the competitors if this amount of saved money is reinvested in other assets or the growth of the production. Due to the period of five years requested by the depreciation of the entire architecture, this cannot be done immediately but, with the mechanism of the retained earnings, after this time window, the company should have the appropriate cash to make this further evolutionary step.

Chapter 7

Conclusions and Future Work

The purpose of this thesis was to try to adapt diagnosis methods to the prognosis phase. The first step was concentrated to define precisely the most successful way to characterize the status of a rolling element bearing, based on the vibrational signal of the component. Then, a collection of metrics about such signal was gathered, in order to train a machine learning model. Finally, some of the same features were used to build a deep learning model. The scenario on which this project is collocated is the industry 4.0 and one of the most tricky tasks about this paradigm is the predisposition of an appropriate cost-benefit analysis which aims to demonstrate if the digitalization of the production chain could have an economical return or not.

Normally, the distance between the diagnosis and the prognosis is high, and this is due to the completely different kinds of targets that each technique tries to reach. The former is focused on identifying if one component is broken and, by definition, this implies that it is not possible to schedule maintenance far away from the time it happened. The latter aims to predict the state of the bearing in order to program future maintenance interventions. Consequently, it is a must to point out that, from a general point of view, the diagnosis and the prognosis are completely separated and they cannot be linked in any way. However, by investigating the reasons which lead a fault diagnosis tool to recognize a break, it is possible, at least theoretically, to exploit such details to transform a real-time break detector into a real-time RUL predictor.

It is possible to assert that this goal has been reached by the various methods described in this thesis. In particular, to the gross of the distinction between a machine learning and a deep learning approach, the diagnosis techniques increase the performance of the bearings which have a trend similar to the one used to train the model. In other words, a diagnosis-based prognosis algorithm is able to predict the degradation trend of a component better with respect to the predictive pipelines which do not make use of any diagnosis preprocessing or feature extraction method. On the other side, bearings that have a behavior extremely different respect to the training set are penalized, for different reasons. The first one is inherited to the principles of data science. The training set is composed of two similar bearings which generate more or less 3000 samples. Instead, the number of test bearings is five and some of them have a trend that is in any way correlated to the training ones. As described in Subchapter §4.3.4, a classical dataset is

split into two parts, one for training and one for testing, which contain respectively the 80% and the 20% of the total number of samples. In this case, the situation is completely flipped. In the literature, it is possible to find some research that, based on the evolution of the RMS of some test bearings, consider them as broken before the threshold defined by PRONOSTIA. In addition, as such components normally differ from the ones already used for training, they recycle them as training samples. Naturally, this procedure could only increment the performance obtained by their models, for all the reasons about the dimensionality of the dataset explained before; but it is also true that in this manner it is not possible to compare the results to the papers which use the traditional training and test set. The second reason relies on a more physical aspect. The most important assumption made on the data is that the multi-domain metrics described in Subchapter §4.3.1 are affected by the degradation of the bearing. If for whatever reason, this does not occur, the entire data-driven solution is useless. For instance, Bearing1_4 and Bearing1_6 are very similar in this sense. It seems that the former and the latter do not present any degradation form before final samples, but Bearing1_4 with respect to Bearing1_6 is characterized by an unexpected and uncontrolled drop in terms of performance during its end life. This is a further proof of the extreme non-linearity of bearings, whose health state can change rapidly and without any form of a preventive alert. In this sense, or the bearing degrades *normally*, or it is necessary to discover other metrics whose trend is correlated to the aging of the component. The prediction of an eruption of a volcano is a situation very similar to the problem analyzed in this work. It is known that an eruption is anticipated from the occurrence of a collection of precursors. However, their efficacy is relevant only for the 24 hours preceding the event, and not always with the same precision. In this sense the precursors are useful and thank them it is possible to put into practice a series of actions capable of safeguarding the population nearby the site of the eruption, even if nowadays it could happen, with a sufficient dose of certainty, that they fail their goal, leading to a catastrophic natural and human disaster. At the moment a lot of geological research focuses their attention on this topic and it is probable that, in the not too distant future, this limit of knowledge will be overpassed. An additional source of inaccuracy is embodied in the choice of the best data-driven model. Subchapter §5.4 pointed out some comparisons in order to determine the best model between the SVR and the RFR on the machine learning side, and the best architecture between Spectrogram, Wavelet, and Mixed on the deep learning side. This passage is necessary because the final solution has to be as flexible as possible, even if some bearing is penalized by this choice. It is indicative the situation of the Wavelet NN: in this case, Bearing1_7 has a low prediction MAE with a normal configuration. Instead, the version of architecture that performs better is the *LSTM+MP*, with a gap in terms of score respect to the normal one of 0.007. This difference is generated by the slightly more accurate prediction of Bearing1_5 and Bearing1_6, where the most performing architecture gains some points of score. In a broad sense and without applying blindly the average between the set of test bearings, another solution could rely on the computation of the overall score without considering the unpredictable REBs. In other words, assuming that a better prediction on Bearing1_5 and Bearing1_6 does not affect overwhelmingly their final outcome, it would be more convenient predicting with higher accuracy the Bearing1_7. However, in order to define the most flexible solution and in the view of ensuring the best overall prediction

for each REB, an average-based pruning method has been implemented. Another possible solution that have also a possible real implementation in RECLAIM is the prediction fusion. Instead of choosing only one model to estimate the RUL, it is possible to make an average over the outcomes of several algorithms. Practically, this is the same as applying an ensemble technique not inside a single model, but inside the entire architecture. In addition, it is also possible to assign a weight to each contribution based on the accuracy reached by the corresponding model on the entire test set.

Moreover, in the view of further improving the technical solution described before, a more detailed proof about the parameters used to train the different models could be implemented. For instance, the feature aggregation phase uses a fixed window size of 20, the same value which characterizes the *LSTM+MP* versions in the deep learning grids search. This value is found empirically, by generally observing the results without any form of rigorous outcomes tracking. Consequently, it is not possible to assert that 20 is the window width that reaches the best results and the same speech is valid also for the definition of the set of hyper-parameters of each model. It is unthinkable to test too many grids with respect to the ones presented in this document, even more for the deep learning models, whose training times are excessively high. However, on the other side, a procedural approach could be implemented. Starting from a set of hyper-parameters far removed from each other, the one which leads to a better performance deserves to be further and more deeply investigated by running other grids which focus their attention on the neighborhood of such parameter.

The application of the industry 4.0 paradigm to an old-fashioned production chain is not easy. This task requires several professional, which comes from completely different domains: electronic, information technology, management, economics, etc. Therefore, not only the definition of an accurate engineering solution requires a lot of effort, but also the predisposition of a sensorial network, the scheduling of all the activities involved in the implementation, and the economical considerations about the convenience of this kind of investment. This thesis has tried to solve two-thirds of the aforementioned steps. Naturally, in order to answer as many questions as possible, some aspects have to be neglected. On the other side, it is also true that a deeper analysis, especially regarding the CBA, should require more applicational details, which for one reason or another, are not taken into account in this work. For instance, always about the CBA, in order to define in a more precise and accurate manner the list of costs and the correlated benefits, also the Key Performance Indicator (KPI) has to be evaluated. In particular, a more correct way forward could start from the list of KPIs to subsequently define the cost model and the collection of goals that industry 4.0 has to reach, transforming the CBA into an optimization problem.

Bibliography

URL http://linksfoundation.com.

URL https://www.reclaim-project.eu/.

P.F. Albrecht, J.C. Appiarius, R.M. McCoy, E.L. Owen, and D.K. Sharma. Assessment of the reliability of motors in utility applications - updated. *IEEE Transactions on Energy Conversion*, EC-1(1):39–46, 1986. doi: 10.1109/TEC.1986.4765668.

J. Antoni and R.B. Randall. Unsupervised noise cancellation for vibration signals: part ii—a novel frequency-domain algorithm. *Mechanical Systems and Signal Processing*, 18(1):103–117, 2004. ISSN 0888-3270. doi: https://doi.org/10.1016/S0888-3270(03)00013-X. URL https://www.sciencedirect.com/science/article/pii/S088832700300013X.

Harvey L Balderston. The detection of incipient failure in bearings. *Mater Evaluation*, 27 (6):121–128, 1969.

D. W. Benbow and H. W. Broome. *The Certified Reliability Engineer Handbook*. American Society for Quality, 2008.

Tarak Benkedjouh, Kamal Medjaher, Noureddine Zerhouni, and Saïd Rechak. Remaining useful life estimation based on nonlinear feature reduction and support vector regression. *Engineering Applications of Artificial Intelligence*, 26(7):1751–1760, 2013.

P Bradshaw and RB Randall. Early detection and diagnosis of machine faults on the trans alaska pipeline. In *MSA session, ASME Conference, Dearborn, Mich*, 1983.

S. Braun and B. Datner. Analysis of Roller/Ball Bearing Vibrations. *Journal of Mechanical Design*, 101(1):118–125, 01 1979. ISSN 0161-8458. doi: 10.1115/1.3454009. URL https://doi.org/10.1115/1.3454009.

Fatih Camci, Kamal Medjaher, Noureddine Zerhouni, and Patrick Nectoux. Feature evaluation for effective bearing prognostics. *Quality and reliability engineering international*, 29(4):477–486, 2013.

Malcolm J Crocker. *Handbook of noise and vibration control*. John Wiley & Sons, 2007.

Lingli Cui, Xin Wang, Huaqing Wang, and Jianfeng Ma. Research on remaining useful life prediction of rolling element bearings based on time-varying kalman filter. *IEEE Transactions on Instrumentation and Measurement*, 69(6):2858–2867, 2019.

J. Lippe H. Engja, M. Rasmussen. Vibration analysis used for detection of rolling element bearing failures. Technical report, Norwegian Maritime Research, 1977.

Yaogang Hu, Hui Li, Pingping Shi, Zhaosen Chai, Kun Wang, Xiangjie Xie, and Zhe Chen. A prediction method for the real-time remaining useful life of wind turbine bearings based on the wiener process. *Renewable energy*, 127:452–460, 2018.

Kamran Javed, Rafael Gouriveau, Noureddine Zerhouni, and Patrick Nectoux. Enabling health monitoring approach based on vibration data for accurate prognostics. *IEEE Transactions on industrial electronics*, 62(1):647–656, 2014.

Kamran Javed, Rafael Gouriveau, and Noureddine Zerhouni. A new multivariate approach for prognostics based on extreme learning machine and fuzzy clustering. *IEEE transactions on cybernetics*, 45(12):2626–2639, 2015.

I. Dimitrakopoulos K. Kamaras. Vibration analysis of rolling element bearings using spectral kurtosis and envelope analysis. Technical report, FNT Condition Monitoring Services, 2016.

Steven M Kay and Stanley Lawrence Marple. Spectrum analysis—a modern perspective. *Proceedings of the IEEE*, 69(11):1380–1419, 1981.

James Kuria Kimotho, Christopher Sondermann-Wölke, Tobias Meyer, and Walter Sextro. Machinery prognostic method based on multi-class support vector machines and hybrid differential evolution–particle swarm optimization. *Chemical Engineering Transactions*, 33, 2013.

Georgia-Ann Klutke, Peter C Kiessler, and Martin A Wortman. A critical look at the bathtub curve. *IEEE Transactions on reliability*, 52(1):125–129, 2003.

Mohamed Boudiaf Koura, Ahmed Hamida Boudinar, et al. Comparaison entre la technique vibratoire et la technique des courants statoriques: Application au diagnostic des roulements à billes. In *2018 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*, pages 1–6. IEEE, 2018.

J Lee, H Qiu, G Yu, J Lin, et al. Rexnord technical services: Bearing data set. *Moffett Field, CA: IMS, Univ. Cincinnati. NASA Ames Prognostics Data Repository, NASA Ames*, 2007.

Yaguo Lei, Naipeng Li, and Jing Lin. A new method based on stochastic process models for machine remaining useful life prediction. *IEEE Transactions on Instrumentation and Measurement*, 65(12):2671–2684, 2016.

Naipeng Li, Yaguo Lei, Jing Lin, and Steven X Ding. An improved exponential model for predicting remaining useful life of rolling element bearings. *IEEE Transactions on Industrial Electronics*, 62(12):7762–7773, 2015.

Linxia Liao. Discovering prognostic features using genetic programming in remaining useful life prediction. *IEEE Transactions on Industrial Electronics*, 61(5):2464–2472, 2013.

Linxia Liao, Wenjing Jin, and Radu Pavel. Enhanced restricted boltzmann machine with prognosability regularization for prognostics and health assessment. *IEEE Transactions on Industrial Electronics*, 63(11):7076–7083, 2016.

Zhiliang Liu, Ming J Zuo, and Yong Qin. Remaining useful life prediction of rolling element bearings based on health state assessment. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 230(2):314–330, 2016.

Theodoros H Loutas, Dimitrios Roulias, and George Georgoulas. Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic e-support vectors regression. *IEEE Transactions on Reliability*, 62(4):821–832, 2013.

A. W. Morgan and D. Wyllie. A survey of rolling-bearing failures. *Proceedings of the Institution of Mechanical Engineers, Conference Proceedings*, 184(6):48–56, 1969. doi: 10.1243/PIME_CONF_1969_184_143_02. URL https://doi.org/10.1243/PIME_CONF_1969_184_143_02.

Patrick Nectoux, Rafael Gouriveau, Kamal Medjaher, Emmanuel Ramasso, Brigitte Chebel-Morello, Noureddine Zerhouni, and Christophe Varnier. Pronostia: An experimental platform for bearings accelerated degradation tests. In *IEEE International Conference on Prognostics and Health Management, PHM'12.*, pages 1–8. IEEE Catalog Number: CPF12PHM-CDR, 2012.

Yuning Qian, Ruqiang Yan, and Shijie Hu. Bearing degradation evaluation using recurrence quantification analysis and kalman filter. *IEEE Transactions on Instrumentation and Measurement*, 63(11):2599–2610, 2014.

Emmanuel Ramasso, Michele Rombaut, and Noureddine Zerhouni. Joint prediction of continuous and discrete states in time-series based on belief functions. *IEEE transactions on cybernetics*, 43(1):37–50, 2012.

R. B. Randall. *Vibration-based Condition Monitoring, Introduction and Background*, chapter 1, pages 1–23. John Wiley and Sons, Ltd. ISBN 9780470977668. doi: https://doi.org/10.1002/9780470977668.ch1. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470977668.ch1.

Robert Bond Randall. *Vibration-based condition monitoring: industrial, automotive and aerospace applications*. John Wiley & Sons, 2021.

Lei Ren, Yaqiang Sun, Hao Wang, and Lin Zhang. Prediction of bearing remaining useful life with deep convolution neural network. *IEEE Access*, 6:13041–13049, 2018.

Patricia Scanlon, Darren F Kavanagh, and Francis M Boland. Residual life prediction of rotating machines using acoustic noise signals. *IEEE Transactions on Instrumentation and Measurement*, 62(1):95–108, 2012.

Piyush Shakya, Makarand S Kulkarni, and Ashish K Darpe. A novel methodology for online detection of bearing health status for naturally progressing defect. *Journal of Sound and Vibration*, 333(21):5614–5629, 2014.

Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, Mao-Yin Chen, and Dong-Hua Zhou. A wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation. *Mechanical Systems and Signal Processing*, 35(1-2):219–237, 2013.

Rodney K Singleton, Elias G Strangas, and Selin Aviyente. Extended kalman filtering for remaining-useful-life estimation of bearings. *IEEE Transactions on Industrial Electronics*, 62(3):1781–1790, 2014.

F Sloukia, Mohamed El Aroussi, Hicham Medromi, and Mohamed Wahbi. Bearings prognostic using mixture of gaussians hidden markov model and support vector machine. In *2013 ACS International Conference on Computer Systems and Applications (AICCSA)*, pages 1–4. IEEE, 2013.

Diego A Tobon-Mejia, Kamal Medjaher, Noureddine Zerhouni, and Gerard Tripot. A mixture of gaussians hidden markov model for failure diagnostic and prognostic. In *2010 IEEE International Conference on Automation Science and Engineering*, pages 338–343. IEEE, 2010.

Biao Wang, Yaguo Lei, Naipeng Li, and Ningbo Li. A hybrid prognostics approach for estimating remaining useful life of rolling element bearings. *IEEE Transactions on Reliability*, 69(1):401–412, 2018.

Yu Wang, Yizhen Peng, Yanyang Zi, Xiaohang Jin, and Kwok-Leung Tsui. A two-stage data-driven-based prognostic approach for bearing degradation problem. *IEEE Transactions on industrial informatics*, 12(3):924–932, 2016.

Mingming Yan, Xingang Wang, Bingxiang Wang, Miaoxin Chang, and Isyaku Muhammad. Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model. *ISA transactions*, 98:471–482, 2020.

Bo-Suk Yang, Myung-Suck Oh, Andy Chit Chiow Tan, et al. Machine condition prognosis based on regression trees and one-step-ahead prediction. *Mechanical Systems and Signal Processing*, 22(5):1179–1193, 2008.

Bo-Suk Yang, Andy Chit Chiow Tan, et al. Multi-step ahead direct prediction for the machine condition prognosis using regression trees and neuro-fuzzy systems. *Expert systems with applications*, 36(5):9378–9387, 2009.

Mei Yuan, Yuting Wu, and Li Lin. Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network. In *2016 IEEE international conference on aircraft utility systems (AUS)*, pages 135–140. IEEE, 2016.

Bin Zhang, Lijun Zhang, and Jinwu Xu. Degradation feature selection for remaining useful life prediction of rolling element bearings. *Quality and Reliability Engineering International*, 32(2):547–554, 2016.

Xiaomin Zhao, Ming J Zuo, Zhiliang Liu, and Mohammad R Hoseini. Diagnosis of artificially created surface damage levels of planet gear teeth using ordinal ranking. *Measurement*, 46(1):132–144, 2013.

Jun Zhu, Nan Chen, and Weiwen Peng. Estimation of bearing remaining useful life based on multiscale convolutional neural network. *IEEE Transactions on Industrial Electronics*, 66(4):3208–3216, 2018.