



Universidad de Oviedo

UNIVERSITY OF OVIEDO

**Official Doctoral Program in Computer Science
Intelligent Systems**

Computer Science and Artificial Intelligence

Doctoral Thesis

**Explainable condition monitoring from
imprecise information**

**D. Costa Cortez, Nahuel
Supervisor: D. Sánchez Ramos, Luciano**

December 2022



RESUMEN DEL CONTENIDO DE TESIS DOCTORAL

1.- Título de la Tesis	
Español: Monitoreo de condición explicable a partir de información imprecisa	Inglés: Explainable condition monitoring from imprecise information
2.- Autor	
Nombre: Nahuel Costa Cortez	DNI/Pasaporte/NIE:
Programa de Doctorado: Informática	
Órgano responsable: Centro Internacional de Postgrado de la Universidad de Oviedo	

RESUMEN (en español)

Los avances computacionales que han sucedido a lo largo de los últimos años, especialmente en la última década, han permitido el desarrollo de algoritmos de Inteligencia Artificial (IA) que han formado parte activa en el progreso de diferentes campos tales como la industria, la medicina o el arte. Ejemplos de ello pueden ser los sistemas de visión por computador que permiten a los coches autónomos modelar el entorno que les rodea, algoritmos de detección precoz del cáncer o la generación de imágenes realistas a partir de descripciones de texto.

Debido a su buen rendimiento en la resolución de problemas complejos sin la necesidad de recurrir a conocimiento experto, estos avances han atraído enormemente el interés en el monitoreo de condición, que abarca problemas en los que se monitorizan las fuentes de información de un sistema para detectar cambios que provoquen fallos o deterioros. Históricamente es un concepto ligado a la industria, pero hoy en día las fuentes de información son muy diversas, por lo que la monitorización puede aplicarse a la inspección de fallos en un motor, así como también a la detección de anomalías en las constantes vitales de una persona en la UCI o en la predicción del mercado de valores.

La IA puede tomar un papel fundamental en el desarrollo de este campo, sin embargo, existen una serie de limitaciones que hacen que su aplicación se vea aún restringida. Por ejemplo, muchos de los grandes progresos están ligados a la capacidad de reutilizar el conocimiento de modelos pre-entrenados, algo que se conoce como Transfer Learning. Esta reutilización es posible en datos estáticos como las imágenes o el texto, pero no en datos dinámicos como es el caso de los datos de monitorización, que normalmente son registros históricos que se almacenan a lo largo del tiempo. Esto se debe a que, a diferencia del texto o las imágenes, la cuantía física de los datos a monitorizar cambia entre problemas, es decir, un modelo entrenado sobre datos de motores es difícilmente aprovechable para el estudio de las constantes de una persona en la UCI.

Por otra parte, los datos registrados por los sistemas a monitorizar están sujetos a una serie de inconvenientes que entorpecen su procesamiento. Por ejemplo, es habitual que la captura de datos sea una tarea dificultosa y, por consiguiente, pueden existir restricciones para crear modelos debido a defectos o imprecisiones en los datos, ya sea por su dimensionalidad, su complejidad o simplemente por su escasez.

Además, la aplicación de modelos de IA se basa normalmente en predicciones o clasificaciones crudas donde existe una fuerte carencia en la interpretación de los resultados. Esta condición además se ve agravada cuando los datos son imprecisos o inexactos, puesto que los algoritmos tienen mayores dificultades a la hora de modelar la realidad o en el caso de hacerlo, están sujetos a sesgos. Por esta razón, se antoja necesario la creación de modelos que sean capaces de explicar las decisiones que toman.



El desarrollo de esta tesis engloba el tratamiento de los problemas mencionados con soluciones generalistas que pueden ser aplicadas a diferentes campos. Las investigaciones realizadas persiguen la creación de modelos dinámicos capaces de reflejar el comportamiento real del sistema a monitorizar de forma que permitan explicar la naturaleza de los datos y a su vez de interpretar los resultados más allá de simples predicciones numéricas. Además, esto conlleva la dificultad añadida de trabajar con datos de baja calidad y con un alto nivel de incertidumbre. De esta manera, se pretende contribuir a ampliar el rango de aplicación de algoritmos de IA a problemas donde, a día de hoy, su aplicación es limitada.

RESUMEN (en Inglés)

The computational advances that have taken place over the last few years, especially in the last decade, have enabled the development of Artificial Intelligence (AI) algorithms that have played an active role in the progress of different fields such as industry, medicine or art. Examples include computer vision systems that allow autonomous cars to map their surroundings, early cancer detection algorithms or the generation of realistic images from text descriptions.

Due to their remarkable performance in solving complex problems without the need for expert knowledge, these advances have attracted enormous interest in condition monitoring. Condition monitoring refers to the process of observing a system's information sources to identify changes that may cause an impending failure or deterioration. It is a concept that has its roots in industry, but with the diversity of information sources available today, monitoring can also be applied to other fields such as medicine for the detection of abnormalities in a person's vital signs or finance for stock market prediction.

AI can play a fundamental role in the development of this field, however, there are still some limitations that hinder its effective application. For example, many of the major advances in AI are possible due to the ability to reuse knowledge from pre-trained models. This reutilization, known as Transfer Learning, can be applied to static data like images or text, but not to dynamic data such as monitoring information, which are usually historical records that are stored over time. This is because, unlike text or images, the physical quantities of the data to be monitored changes between problems, e.g., the knowledge of a model trained on engine data can hardly be used to study the constants of a person in the ICU.

What is more, the data recorded by the systems often present some problems that hamper their processing. It is common for data capture to be a difficult task, so there may be restrictions to create models due to defects or inaccuracies in the data, either because of its dimensionality, its complexity or simply because of its scarcity.

On the other hand, the application of AI models is usually based on raw predictions where there is a strong lack of interpretation of the results. This condition is further aggravated when the data are imprecise or inaccurate, since the algorithms have greater difficulty in modeling reality or, if they do, are subject to biases. For this reason, it is necessary to create models that can explain the decisions they make.

The development of this thesis addresses the mentioned problems with solutions that can be applied to different fields. The research carried out is aimed at creating dynamic models capable of reflecting the real behavior of the system to be monitored in order to explain the nature of the data and, in turn, to interpret the results beyond simple numerical predictions. In addition, this entails the added difficulty of working with low quality data and with a high level of uncertainty. In this way, the aim is to contribute to broaden the range of application of AI algorithms to problems where, to date, their application is limited.

Acknowledgments

First and foremost, I would like to begin by expressing my gratitude to my advisor, Dr. Luciano Sánchez Ramos, who made this research project possible. His excellent mentorship, guidance and support throughout this thesis is deeply appreciated. I would also like to thank José Ranilla Pastor, Inés Couso Blanco and David Anseán González for their support and collaboration.

I am also greatly thankful to Dr. Matthieu Dubarry at the Hawai'i Natural Energy Institute (HNEI) at the University of Hawaii at Manoa for his kind support and highly valuable guidance during my stay at his laboratory. Several lab mates at the HNEI deserve special recognition for their warm welcome and their generous hospitality, including David Beck, Jacob Morton and Alexa Fernando.

Most specially, I would like to thank my family, María and all my friends outside academia for their love and unconditional support.

Finally, I am also greatly thankful to the Spanish Ministry of Science and Innovation (MIC) for partly providing the funding (PID2020-112726-RB-100) of this work.

Contents

INTRODUCTION	13
1.1. CONTEXT.....	14
1.1.1. Atrial Fibrillation.....	14
1.1.2. Aircraft engines	15
1.1.3. Lithium-ion batteries.....	16
1.1.4. Similarities	18
1.2. CONTRIBUTIONS.....	18
1.3. STRUCTURE OF THE DOCUMENT	20
RELATED WORK	22
2.1. DEEP LEARNING MODELS.....	23
2.1.1. Autoencoders	23
2.1.2. Convolutional Neural Networks.....	24
2.1.3. Recurrent Neural Networks	25
2.2. DEEP LEARNING APPLICATIONS IN CONDITION MONITORING.....	26
RESEARCH FINDINGS.....	29
3.1. DATA PROCESSING	29
3.1.1. Simulation models.....	29
3.1.2. Data preprocessing.....	35
3.1.3. Summary	41
3.2. DEEP LEARNING-BASED SOLUTIONS	42
3.2.1. Variational Autoencoder (VAE)	42
3.2.2. Generative Adversarial Networks (GAN).....	47
3.2.3. Convolutional Neural Networks	51
3.3. EXPLAINABILITY	52
3.3.1. GAN ensembles	52
3.3.2. Variational encoding	54
3.4. NUMERICAL RESULTS	58
3.4.1. Graphical analysis of the progression of atrial arrhythmia using recurrent neural networks	58
3.4.2. Semi-supervised recurrent variational autoencoder approach for visual diagnosis of atrial fibrillation.....	60

3.4.3. Variational encoding approach for interpretable assessment of remaining useful life estimation	61
3.4.4. Li-ion battery degradation modes diagnosis via Convolutional Neural Networks...	62
CONCLUSIONS	65
4.1. FUTURE WORK	66
PUBLICATIONS.....	69
4.2. JOURNAL PUBLICATIONS.....	69
4.3. CONFERENCE PROCEEDINGS.....	70
4.4. UNDER REVIEW	70

List of Figures

Figure 2.1.- Autoencoder architecture.....	24
Figure 2.2.- CNN architecture.....	25
Figure 2.3.- RNN architecture.....	26
Figure 3.1.- The morphology of the superficial ECG (left) is not maintained in the intracardiac ECG (right), where there is only one peak per beat.	31
Figure 3.2.- State diagram of the model of AF episodes.....	32
Figure 3.3.- Daily percentage of time in arrhythmia of a sample patient (left). Same sample with a smoothing applied (right).	35
Figure 3.4.- Time window framing.	37
Figure 3.5.- Capacity vs voltage curve (left). IC representation (right): voltage vs derivative of capacity.....	37
Figure 3.6.- Euclidean distance between each pair of points of the two sequences displayed on a grid (a). Every warping path represented as a set of pixels (b). In both images the optimal warping path is marked in blue.....	38
Figure 3.7.- IC signatures from the initial state (solid line) for each degradation in the dataset: LLI (a), LAMPE(b) and LLI(c) at 20% degradation.....	39
Figure 3.8.- DTW Images for 20% LLI degradation for three different cell configurations.	40
Figure 3.9.- Simplified representation of the compression resulting from a vanilla Autoencoder (left) and a VAE (right). When the latent space is continuous, the organization of the data allows decoding a meaningful figure, in this case a cross between a rectangle and a triangle, thus favoring the generation of new data.....	43
Figure 3.10.- Network structure of the proposed recurrent VAE. The blue and green blocks are the encoder and decoder respectively and the red block represent a downstream task such as classification or regression.	44
Figure 3.11.- Latent representations learned by the encoder for one of the aircraft datasets. The figure on the left shows the regular training of a VAE, while the figure on the right shows the result with the proposed model, which does not include the decoder but a regression model that adds a penalty for wrong predictions.	46
Figure 3.12.- Workflow of GAN networks.	47

Figure 3.13.- Ensemble of GAN discriminators, each trained with different classes of AF. When feeding evaluation data to the ensemble at least one of the discriminators is expected to be activated.....	48
Figure 3.14.- Discriminator network structure.	49
Figure 3.15.- Generating network structure.	50
Figure 3.16.- Model architecture for Li-ion degradation modes diagnosis. Conv1 to Conv4 represent the convolution layers followed by the max pooling layers. The features extracted are condensed in a flatten layer from which the 3 degradation modes are predicted.....	51
Figure 3.17.- Left: projection of AF events using the Markov model, with $\beta = 90$ and $\alpha = 0.999$. Right: projection of a sequence of AF events from a real pacemaker. The map is consistent with a value of $\beta \approx 200$ and a rapid progression toward permanent AF, $\alpha = 0.998$	54
Figure 3.18.- Learned representation of simulated AF events (left). Latent projection of a real patient (right).	55
Figure 3.19.- Left: Latent projection of one of the CMAPSS training sets. The red trail corresponds to the RUL evolution of a testing engine. Right: RUL evolution of six real engines in two different time steps, cycle 0 and cycle 1000.	57
Figure 3.20.- Left: GAN ensemble confusion matrix. Right: Random Forest confusion matrix. Similar classes should be nearby on the map, thus classification errors should be close to the diagonal.	59

List of Tables

Table 3.1.- Classes generated to model AF behavior.....	33
Table 3.2.- C-MAPSS datasets details.....	34
Table 3.3.- Summary of data, preprocessing steps and types of problems.....	41
Table 3.4.- Accuracy of the different classifiers, six types of AF.....	58
Table 3.5.- Accuracy of the different classifiers, six types of AF.....	60
Table 3.6.- Family of hypotheses ordered by p-value.....	61
Table 3.7.- Evaluation metrics of different approaches for RUL estimation on C-MAPSS datasets.....	62
Table 3.8.- RMSE results for each degradation mode and cycle for the LFP test set.....	63
Table 3.9.- RMSE results summary for the LFP test set calculated as the average and the standard deviation of predictions in all cycles for all cells.....	64

List of abbreviations

AI: Artificial Intelligence

LLM: Large Language Model

AF: Atrial Fibrillation

RUL: Remaining Useful Life

LIBs: lithium-ion batteries

CV: Computer Vision

NLP: Natural Language Processing

AE: Autoencoder

VAE: Variational Autoencoder

CNN: Convolutional Neural Network

RNN: Recurrent Neural Networks

ECG: Electrocardiogram

iECGs: Intracardiac Electrocardiogram

EDA: Exploratory Data Analysis

IC: Incremental Capacity curve

FOI: Features of Interests

DTW: Dynamic Time Warping

GAN: Generative Adversarial Network

Chapter 1

Introduction

Advances in computing capabilities and new technologies have made possible the availability of different sources of information from which useful knowledge can be extracted for later decision-making. In fact, data collection is of increasing interest in a wide variety of fields and, consequently, data management seems to be necessary to ensure the correct performance of any company or organization.

In this context, it is important to note that there is a large number of situations in which data is recorded over time and so, it is essential to keep track of possible changes for a proper analysis. Condition monitoring is a concept that precisely refers to this claim as it is understood as the process of controlling the information sources of a system in order to detect changes that may be significant for decision-making. Despite being historically linked to industry, nowadays the sources of information are very diverse, allowing to apply this monitoring to different problems such as preventive maintenance [1], the analysis of vital signs of a person in the ICU [2], or the prediction of the stock market [3], to name a few examples.

On the other hand, Artificial Intelligence (AI) has been incorporated as another tool in data analysis, especially in recent years with the rise of new architectures and novel paradigms that allow its effective use. However, the greatest advances have been brought by the ability to reuse the knowledge of a model and adapt it to solve a different problem, something known as Transfer Learning [4]. This is possible because large, trained AI models can be leveraged in the creation of new models so that they do not learn from scratch but rather start from a more advanced stage. In contrast to these cases, when dealing with systems that store historical records, the physical nature of the data changes in every dataset, making such reuse impossible. For example, a Large Language Model (LLM) that encodes text can be easily reused for other tasks because the data is still text, the only thing that changes between datasets is the objective (sentiment analysis, text prediction, classification....). However, when dealing with data stored over time, it represents different physical quantities (heart monitoring, sensor readings, battery degradations...) therefore, in addition to the objective, the data also changes.

Furthermore, unlike images or text, which are easily accessible, monitoring data is difficult to collect because it is often specific to each field of study. Additionally, it is vulnerable to

deficiencies that hinder its processing. For example, it is common to find restrictions in data capture due to problems such as sensor failure, sensitivity to noise, or there may simply be data shortage.

Finally, despite being incredibly good at some tasks most AI models, especially Deep Learning (DL) models behave like black boxes. This is because they rely on feeding an input to an algorithm that results in a numerical prediction [5], and this might not be enough in many problems. The ultimate goal of developing such models should be that they can be easily used by people in any field outside of AI. Therefore, it is essential to promote explainability, i.e., to describe how knowledge of their different parts affects the learning of the model as a whole; and also, interpretability of the results, which allow a cause to be easily associated with an effect.

This thesis provides solutions to these problems, with a focus on DL solutions for data that is of low quality and changes over time. In addition, the algorithms developed have a strong explanatory component, providing an understanding of the inner workings of the models and a visual and intuitive interpretation of the results that help understand the knowledge generated about the system being monitored.

1.1. CONTEXT

One of the main goals of this thesis is to provide solutions that are not tailored to a specific field but to common limitations that may be present in different fields. Consequently, the methods developed have been studied in three different areas: atrial fibrillation, aircraft engines and lithium-ion batteries. These topics will be a recurring element throughout the development of the thesis and serve to illustrate the achievement of the proposed objectives.

1.1.1. Atrial Fibrillation

Heart diseases are among the leading causes of death worldwide [6]. Atrial Fibrillation (AF) is the most common cardiovascular disorder, and its treatment involves the use of pacemakers to regulate the heart rhythm. Unfortunately, there is currently no standard procedure for diagnosing the disease state beyond the check of the data captured by medical specialists.

The disease usually evolves from paroxysmal arrhythmias (arrhythmias that come and go spontaneously) to persistent arrhythmias (episodes that last more than 7 days and do not end without external intervention) or to permanent arrhythmias (uninterrupted episodes). It is caused by a problem in the electrical system of the heart that leads to an irregular heartbeat

in which the upper chambers of the heart, the atria, fibrillate. Because of the loss of synchrony in the heart rhythm, symptoms such as chest pains or dizziness may appear, and in the worst scenarios, it can cause the formation of clots inside the heart which, if they reach the bloodstream, can result in serious problems such as stroke [7].

The records stored by pacemakers are generally limited, as a long history would mean the individual is already in the final stage of the disease. This limitation makes it difficult to obtain a model that fits the characteristics of the data and understand the patient's condition. Furthermore, the disease evolves over time, making the data non-stationary and it is precisely the changes in the properties of the data that determine the progression from paroxysmal to permanent arrhythmias.

There are even more difficulties, the algorithm used by pacemakers to determine the duration of arrhythmia episodes is not completely reliable [8]. The device parameters are adjusted prioritizing safety, so the false positive rate is high. This results in long AF episodes that are sometimes erroneously reported as sets of short episodes [9], so preprocessing is needed to account for these spurious events, which in turn causes the number of available episodes to be further reduced.

Given these facts, the progression of AF is a complex process that depends on many different factors. Ideally, the aim is to find a model that would be able to learn, from a few dozen recordings, the specific properties of the disease and at the same time can extrapolate the breaking point between paroxysmal and permanent AF. In this way, it would be possible to provide, from intracardiac data, a diagnosis of the patient's situation that would make it possible to show how the disease will progress in the near future. This would enable medical specialists to prevent any future complications, and thus improve the health and quality of life of the patient.

1.1.2. Aircraft engines

The advancement of technology has resulted in significant improvements in the safety and maintenance of aircraft engines. Prognostic technologies have been essential in this process since aircrafts are subjected to different conditions throughout their life cycle that cause degradation and ultimately lead to failure. Thus, being able to provide safe monitoring in these systems translates to a notable increase in reliability and also an extension in their useful life, which in the end also means remarkable savings for manufacturing companies.

In order to monitor performance and prevent operation under undesirable conditions, the engines have several built-in sensors from which data is routinely collected. Over the years, the amount of information collected has increased and this has paved the way for more complex analysis in favor of life-extending maintenance. However, traditional strategies

such as scheduled preventive maintenance or corrective maintenance are increasingly unable to meet the growing industrial demand in terms of efficiency and reliability. In recent years, metrics such as Remaining Useful Life (RUL) [10] have gained popularity and have been established as key elements to improve maintenance and avoid engineering, safety and reliability failures. Consequently, this would make it possible to determine engine deterioration, increase flight time and reduce maintenance costs.

Traditionally, anomaly detection is widely adopted in the field [11], [12] but not so much the study of the phenomena that cause the breakpoint between normal and abnormal operation. However, in RUL estimation the aim for a complete and interpretable diagnosis should be to model the evolution of the system over time to know at what speed it evolves towards anomalous situations. This means, not to pursue the identification of anomalies, but rather to find when the engine start deteriorating at a different rate than it did before.

The presence of anomalies is indeed correlated with RUL as they usually correspond to low RUL values. However, two systems can be in the same initial state but have a different evolution over time, so successive system states cannot be studied independently as is done in anomaly detection. Instead, RUL estimation must be linked to the temporal analysis of complete historical periods.

In addition, monitoring data is subject to limitations. Although the number of sensors is tending to increase, the number of recorded events is still limited and not all sensors are of equal diagnostic importance. Also, many of the records are incomplete, have spurious values or are sensitive to natural factors such as wind, the number of passengers or trajectory changes and can cause noticeable peaks in each signal, which ultimately corrupts the data.

An end-to-end model capable of dealing with the above-mentioned problems should greatly reduce the likelihood of an aircraft experiencing an unforeseen event, which represents an invaluable economic saving for manufacturing companies.

1.1.3. Lithium-ion batteries

Since their commercialization in the early 1990s, lithium-ion batteries (LIBs) have been widely used in key commercial and industrial applications, ranging from portable electronics and transportation to storage systems. Unfortunately, the performance of LIBs decreases with operation due to parasitic reactions taking place at the positive and negative electrodes (PE and NE respectively), as well as in the electrolyte [13]. In addition, some specific side reactions, such as lithium plating, can create safety risks [14]. Both decreased performance and safety issues are a major concern for deployed LIB systems, especially when reliable and durable applications are critical. To assess LIB performance and ensure overall safety and reliability, it is necessary to determine the state of health (SOH) [15].

Recently, a new paradigm of non-invasive methodologies for assessing battery SOH has emerged thanks to recent improvements in processor capabilities and communications, and AI is expected to have a profound influence on future LIBs diagnostic and prognostic systems. However, the existing methodologies are still in their early stages [16] and critical issues remain to be solved.

Degradation of LIBs is the result of a complex interaction between physical and chemical mechanisms within the battery, leading to loss of capacity and power. Degradation is path dependent and different uses such as temperatures, charging currents or cut-off voltages, can inhibit or exacerbate specific degradation mechanisms. These mechanisms are varied and can be grouped into degradation modes, which are loss of lithium inventory (LLI) and loss of active material (LAM) at the negative and positive electrodes (LAMNE and LAMPE respectively).

Although the degradation modes are widely known in the field, the underlying causes and effects in the battery are not always straightforward. As an example, LAM degradations are difficult to detect since they do not usually leave signals in the history of capacity loss but after a few cycles of regular operation, they manifest by causing the capacity to decay suddenly. Because of this, they are known in the literature as "silent" or "hidden" modes [17].

In addition to the increasing sophistication of the algorithms required, the amount of data needed for training is also critical, as battery data generation is challenging and time-consuming. The reality is that existing datasets, while providing valuable information, are sparse and only provide data from a few cells [18]. This poses a major obstacle to the application of AI algorithms, as large amounts of data are generally required for the training process. In addition, models trained with these datasets can lead to a false sense of confidence in their performance, as the capacity loss decays linearly in most cases and tests are usually performed with a low variety of duty cycles that are quite often disconnected from real applications (e.g., constant current cycles). The actual data is much more sporadic and sub and supralinear degradations are more common.

Both future datasets and AI algorithms must take these limitations into account to realistically contribute to the diagnosis and prognosis of LIBs. Thus, their life cycles can be extended, and they can even be reused in other systems with less demanding requirements.

1.1.4. Similarities

Having outlined the different topics of study, it is interesting to note that, despite having apparently little relation, they share common characteristics. Specifically:

- The systems to be studied, regardless of whether it is a person, an engine or a battery, tend to degrade and are therefore susceptible to monitoring. There exists an interest in constant monitoring to detect changes that condition future events.
- Degradation evolves over time. The data to be monitored, despite coming from different sources, share that they are events that occur over time, i.e., they are mostly time series.
- The rate of deterioration is not stationary. System degradation is subject to a deterioration that evolves until it reaches a point where sudden degradation occurs. The interest of these problems lies in anticipating these events in order to extend the lifetime of the system to be monitored.
- Since the properties of the systems are different between datasets, even if it is the same application field, the use of transfer learning is not possible. That is, the knowledge learned by pre-trained models cannot be used as a starting point for similar problems.
- Data is imperfect. In all problems, there are data deficiencies, mainly associated with scarcity, but also with inaccuracy in capture and spurious events.

1.2. CONTRIBUTIONS

The main contributions of this thesis are:

1. Development of mechanisms to manage data deficiencies. To model the behavior of a system, Deep Learning algorithms need a large amount of data that may not be sufficient, may not be completely labeled or may be inaccurate or faulty. Different solutions are proposed depending on the type of problem.
 2. Implementation of explainable AI algorithms. The solutions proposed in this thesis have a strong explanatory component in order to facilitate the understanding of the inner workings of the models and thus contribute to understanding the nature of the
-

data, the relationship between the different variables and the interpretation of the results.

3. Creation of visual tools for decision-making. The proposed solutions are oriented to data diagnosis and prognosis to provide an intuitive perspective of the operation and status of the system to be monitored and in turn its evolution in the future. These features are of high priority for decision-making to extend the useful life of the systems.
 4. Development of models capable of anticipating future failures. Unlike anomaly diagnosis, where problems can generally be identified with more traditional methods, what is proposed in this thesis is the focus on the detection of small changes that do not involve an alteration in the normal operation of the system to be monitored but that may suddenly impact as a severe problem in the future.
 5. Provide open-source code that guarantees the reproducibility of results and allows the practical application of the models developed. Open-source code eases accessibility to the research carried out and reinforces its credibility. Therefore, all the work developed is public both for the reproducibility of results and for their easy adaptation to problems of a similar nature.
-

1.3. STRUCTURE OF THE DOCUMENT

This PhD dissertation is structured as follows. The next chapter describes the related work. Chapter 3 contains the results of the research carried out during the course of the thesis, which in turn is divided into four main sections. Section 1 describes the mechanisms used to deal with data scarcity and inaccuracy. The Deep Learning models developed are discussed in section 2 while the explainability of the proposed approaches is commented in section 3. The discussion of results is presented in section 4. Finally, in Chapter 4 the conclusions are drawn, and future lines of work are identified.

Chapter 2

Related work

In this chapter, related work on Deep Learning applied to condition monitoring will be presented and the main limitations of existing methods will be discussed.

Condition monitoring emerges as a critical approach for any industry in order to ensure the safety and reliability of the systems to be monitored, as well as for cost reduction. Within this context, there are a few steps that are typically common regardless of the field of application. This includes collecting and manipulating data and subsequent fault detection, diagnosis and prognosis, with the aim of extracting knowledge that can be critical for decision-making [19].

The methods used for the analysis of the collected data can be classified as model-based, data-driven or hybrid approaches [20]. The former, although usually highly accurate, requires expert knowledge of the subject and may have limitations due to complex behaviors that can be easily overlooked in the modeling. On the other hand, data-driven methods rely on the study of historical records to determine the state of health and remaining useful life of the element to be monitored. Combining both model-based and data-driven approaches, hybrid approaches aim to utilize the advantages of both approaches. However, there are additional difficulties such as the extraction of useful features, for which expert knowledge is needed so that they can then feed other learning algorithms. These hand-crafted features can be subjective, which implies low efficiency and high labor cost, especially in methods that require a large number of labeled samples for training. It is difficult to meet this requirement in many real-world applications where experiments are expensive or even not allowed. For this reason, the current is turning in recent years in favor of data-driven methods [21].

Within the field of Artificial Intelligence, Deep Learning algorithms have made significant impacts largely because of their ability to automate feature engineering, learn internal representations, and create feature vectors from raw data without human intervention [22], thus alleviating the need for expert knowledge.

The following subsections will briefly explain the main Deep Learning models applied to condition monitoring problems and describe some of their applications and limitations.

2.1. DEEP LEARNING MODELS

In recent years, several network architectures have been proposed for fault detection, diagnosis and prognosis. More specifically, these are variants of models that have had a significant impact and are known for their remarkable performance, especially in Computer Vision (CV) and Natural Language Processing (NLP). These architectures have been adapted to deal with different types of monitoring data such as vibration signals, health constants or multi-sensor fusion. The following section briefly explains the most widely used models: Autoencoders (AE), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), as well as their variants.

2.1.1. Autoencoders

Autoencoders are a family of neural networks that are designed to reconstruct the input data and, at the same time, learn a compressed representation, known as the latent space (see Figure 2.1). To do this, it has two elements: an encoder, which compresses the information to a lower dimensionality (the latent space), and the decoder, which reconstruct the data from that compression.

Training an autoencoder requires minimizing the average reconstruction loss, usually the squared error function, over a given training set. The reconstruction loss of the decoder depends on the encoder output, so a good reconstruction means that the latent space learned by the encoder retains representative features of the input.

The family of autoencoders is broad and includes several variants:

- 1) Denoising autoencoder (DAE): they add arbitrary noise to the input data to corrupt it. This is intended to ensure that, just as humans can recognize objects that are not completely clear, the network learns to reconstruct them effectively. In addition, it prevents overfitting [23], thus making the reconstruction more robust.
 - 2) Sparse autoencoder (SAE): Again, in order to avoid overfitting and improve robustness, SAEs add sparsity constraints to the neurons [24]. This is achieved by adding a penalty term that causes learning sparse feature representations that favor data compression and reconstruction.
 - 3) Contractive autoencoder (CAE): Like the sparse autoencoder, the contractive autoencoder [25] encourages the learned representation to remain in a contractive
-

space. It adds a term in the loss function, the Frobenius norm of the Jacobian of the nonlinear mapping to penalize the representation from being too sensitive to the input, and thus improve robustness to small perturbations around the input data.

- 4) Variational autoencoder (VAE): Deeply rooted in variational Bayesian methods, VAEs [26], instead of learning to encode information to a fixed latent space, encode it to a probability distribution, thus forming a continuous latent space. This property allows the decoder to reconstruct data from an area of the latent space that does not belong to the compression of any input data but an interpolation. In this way, it can generate new data samples, which is why they are also known as generative models.

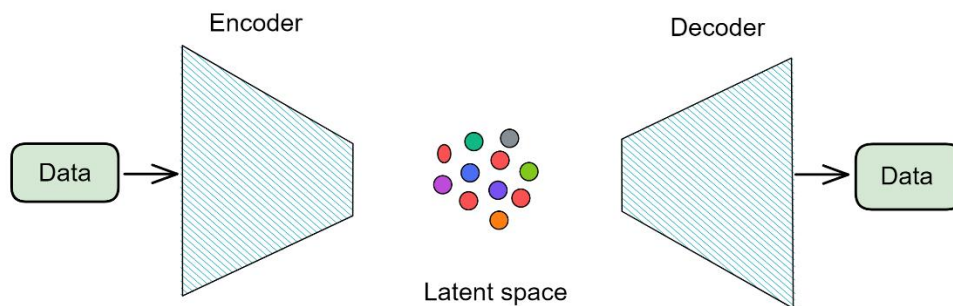


Figure 2.1.- Autoencoder architecture.

2.1.2. Convolutional Neural Networks

CNNs are a type of neural network commonly oriented to analyze visual imagery. Since they were proposed in a handwritten digit recognition task [27], researchers have repeatedly demonstrated their success in various applications such as computer vision, NLP and speech recognition and with multiple derivative architectures like AlexNet [28] or U-NET [29]. CNNs consist of multiple layers of neurons in which two fundamental operations are performed: convolution and pooling (see Figure 2.2). The convolution layers consist of multiple filters, also known as kernels, that are applied to the image to highlight certain features that make it unique. The resulting images are known as feature maps. Pooling reduces the spatial size of the feature maps and learns to ignore irrelevant and redundant information, so the dimension of the data is reduced in each layer.

Stacking multiple convolutional and pooling layers allows a CNN to learn hierarchical feature representations from the input. Filters applied on the first layers obtain the feature maps that primarily characterize the image, while the later layers typically include more filters to capture finer details such as color intensity or brightness.

A variant of the CNNs is the 1D CNN, which employs 1D filters to convolve along a single dimension of its inputs. Although applicable to 2D inputs, 1D CNNs are mainly suited to 1D inputs such as sequential data.

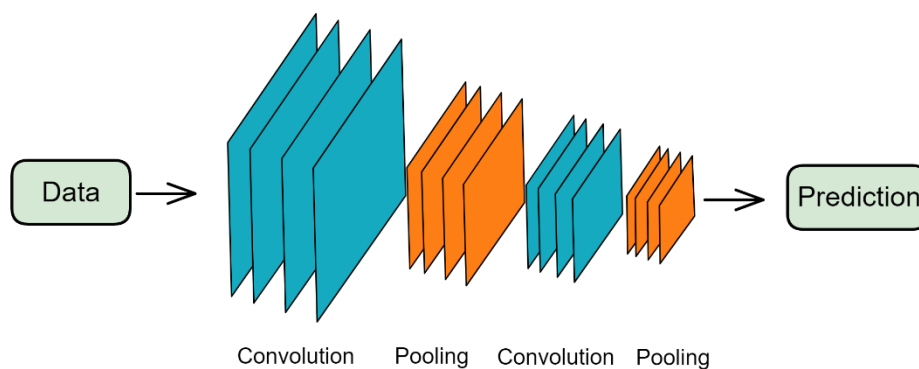


Figure 2.2.- CNN architecture.

2.1.3. Recurrent Neural Networks

Recurrent neural networks are designed to deal with sequential data. They have forward connections, but also backward connections so that the dynamic behavior of the sequences can be modeled, i.e., the output of a neuron can be calculated based on what it received in the past. Thus, we have a network structure where the outputs (or states) of the neurons depend not only on the input they receive but also on the states they had previously (see Figure 2.3).

There are several topologies based on RNNs, such as Elman and Jordan networks [30] or Neural Turing Machines [31]. However, they have become obsolete due to the effectiveness demonstrated by LSTM [32] and GRU [33] networks. One of the reasons why LSTM and GRU work so well is that they avoid a problem known as Vanishing Gradient [34]. A small or zero gradient implies that the network parameters are not updated correctly, consequently, these architectures introduce a gating mechanism, which allows the neural network to avoid this problem by maintaining information through multiple units for a long series of steps.

RNNs analyze data forward, preserving information from the past through hidden states. But it is also possible to preserve future information by processing the input data from front to back. This is the working principle of Bidirectional RNNs (BRNNs) [35]. They make a path first from past to future and then from future to past, preserving information from both

periods. Using information from the future can help the network understand what kind of information to predict.

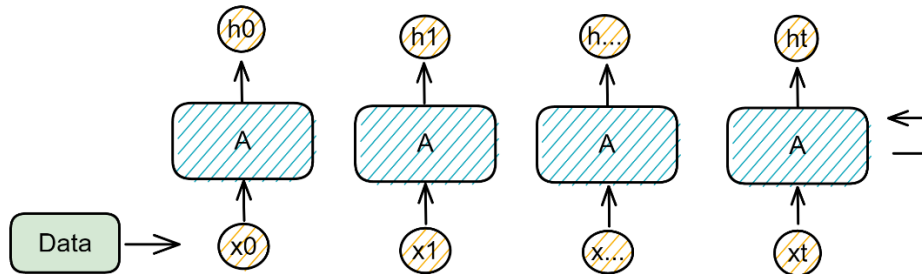


Figure 2.3.- RNN architecture.

2.2. DEEP LEARNING APPLICATIONS IN CONDITION MONITORING

Deep learning is becoming increasingly popular in condition monitoring applications, however, there is no common framework that can be automatically applied to different problems. Most works study subfields individually, mainly associated with anomaly detection, diagnosis and prognosis.

Anomaly detection refers to the process of detecting data instances that deviate significantly from the majority of data instances [36]. It can be understood as a binary classification task, i.e., classifying whether the item of interest is performing well or whether something has gone wrong.

Once a failure is identified, it is necessary to diagnose the severity associated with it. This diagnosis aims to determine what went wrong and must be more rigorous than the detection of the anomaly in its predictive accuracy and results as it can directly affect the operation or maintenance adjustments.

Finally, after the diagnosis, it is expected to infer the remaining useful life of the system to be monitored, i.e., to make a future forecast. This is known as prognosis, and it is key to provide an accurate estimate because a premature prediction can lead to excessive maintenance and a late prediction can lead to irreparable failures.

There are different approaches based on Deep Learning to deal with this set of processes, which can be divided into supervised and unsupervised methods. In the first group, one can find contributions using mainly CNNs or RNNs to predict previously labeled instances. For

example, in [11] the authors used RNNs, specifically LSTM and GRU networks to identify 11 different types of faults associated with flight records of an aircraft, or in [37] a one-dimensional CNN for fault diagnosis in rotating machinery was presented.

On the other hand, in unsupervised methods, labeled data is not available, so dimensionality reduction techniques such as autoencoders [38], [39], [40] are usually used to model normal behavior and identify faults as deviations from normal instances based on different thresholds. Intuitively, the aim is to find a latent space that can encapsulate most of the informative features, from which samples can be then reconstructed with minimal information loss. If a test sample cannot be reconstructed well from its latent space, then it is likely an abnormality. From this process further downstream tasks can be performed for diagnosis and prognosis [41], [42], [43], [44].

There are still some open problems in most of these works given that there are several challenges, mainly associated with data quality and models explainability [45]. Many supervised methods assume that the labeled training data is clean, which can be vulnerable to noisy instances that are erroneously labeled as an opposite class label. In these cases, unsupervised methods can be employed, but there is often unlabeled data contaminated by large-scale anomalies [46]. Furthermore, in many domains it can be dangerous to use the developed algorithms as black-box models. For example, instances of data reported as anomalies can lead to potential bias against minority groups that show up in the data, such as underrepresented groups. An effective approach to mitigate this type of risk is to have explanatory algorithms that provide direct clues about the models' decision. The development of inherently interpretable models is also crucial, but it remains a major challenge to balance model interpretability and effectiveness.

Chapter 3

Research findings

The methodologies developed during this thesis have been validated in three different fields of study: atrial fibrillation, aircraft engines and lithium-ion batteries. This chapter is organized into four sections that cover the completion of the objectives pursued. The content of the published contributions will be explained according to these objectives and so the appearance of each of the different topics will be recurrent in each section.

3.1. DATA PROCESSING

This section presents details on the data used, as well as the mechanisms employed to deal with data inaccuracies and limitations.

3.1.1. Simulation models

One of the most recurrent claims in any condition monitoring problem is that the data is limited, inaccurate or of low quality. This is a major constraint when applying Deep Learning models, which generally require large amounts of data for training. One of the most common approaches to alleviate this problem is the development of physical or mathematical models that can faithfully model the system to be monitored, also known as digital twins.

Digital twins are simply virtual replicas of a system that allow simulations that can predict how a product or process will work [47]. They are used to avoid failures in physical objects and to perform advanced analysis, monitoring and forecasting. The main benefit is that new data can be generated under certain configuration parameters which can then be used to train Deep Learning algorithms as if they were real data.

Nevertheless, building a digital twin is complex, there is no standardized process to do so, nor is it entirely clear what technology is needed to build and implement them. Because of this, there is a heavy reliance on expert knowledge for their development.

In the fields of study used in this thesis, two approaches have been taken: the use of models previously developed by experts or the elaboration of a digital twin that adapts to the needs of the problem to be modeled. After generating the necessary data to train Deep Learning

algorithms, they must be able to be used in real environments, so the models developed in this thesis are also validated on real data.

Atrial Fibrillation

As mentioned in the introduction, there is no standard procedure for diagnosing AF beyond reviewing the intracardiac recordings produced by pacemakers. This is because it is a very specific field and consequently, there is no model capable of reflecting the behavior of the disease.

The aim is to have a dataset that includes a wide range of different types of arrhythmias that AF patients may suffer. Gathering a dataset of these characteristics with records from real people is practically impossible, first because of the sensitivity of the data and then because of the scarcity of data from real patients, so the creation of a customized model is necessary.

There are additional difficulties, unlike surface electrocardiograms (ECGs), which record all types of electrical activity (Figure 3.1, left), intracardiac electrocardiograms (iECGs) simply represent the potential difference between two points in contact with the muscular tissue of the heart, the myocardium, and this causes the morphology of cardiac activity to be lost (Figure 3.1, right). For this reason, the dates and lengths of arrhythmia episodes are considered to provide more information.

Pacemaker data capture is not constant, but certain events trigger their recording, namely episodes of high atrial frequency. When they come into action, they emit an electrical discharge that activates the heart cells to promote cardiac contraction. The pacemaker algorithm has different modes of operation, which rotate according to the activity recorded. When an episode of arrhythmia occurs with a higher-than-normal atrial rate, it is an indicator that the patient is suffering an episode of AF, therefore, the pacemaker changes its mode of operation from rest to electrical discharge emission to force ventricular excitation. The change of mode to act in the presence of an arrhythmia is part of a process known as Automatic Mode Switching (AMS) and when it occurs, the information capture is activated, which will be the date on which the episode occurs, its duration and the resulting iECG.

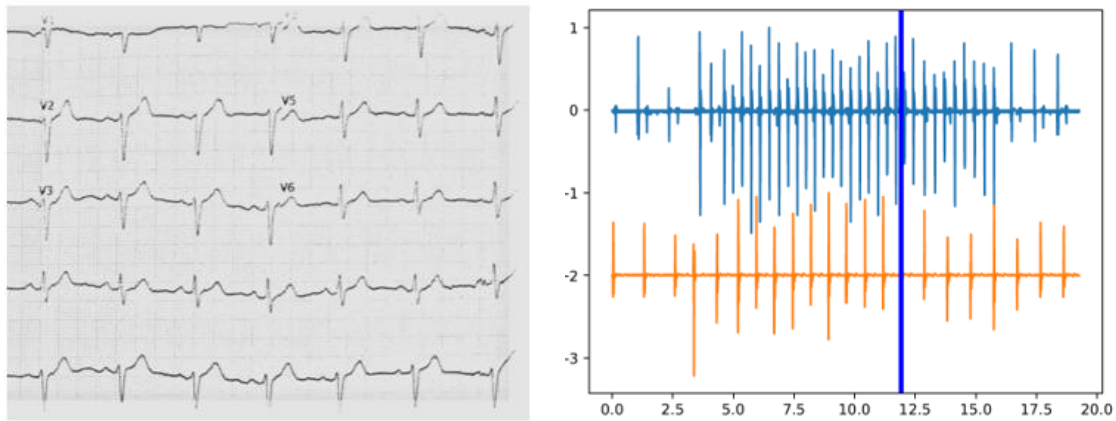


Figure 3.1.- The morphology of the superficial ECG (left) is not maintained in the intracardiac ECG (right), where there is only one peak per beat.

For efficient operation of the algorithm, it is necessary to configure the pacemaker parameters, which are specific to each patient. Generally, iECGs are used to adjust these parameters. The problem with the AMS process is that, since false positives are prioritized so as not to generate ventricular excitation at inappropriate times, there are long arrhythmia episodes that are reported as several short ones. That is, it is common for the algorithm to detect the presence of an arrhythmia, therefore, it generates an AMS event and then consider it over, only to discover seconds later that the episode is still occurring, so it generates another AMS event to attenuate the arrhythmia and subsequently, when successful, the mode returns to the initial state. These cases do not have any transcendental consequences for either the patient or the device, but they do affect the fidelity of the recorded data, causing inaccurate information to be captured.

All this information was obtained in collaboration with medical experts for the development of a model that is able to simulate the behavior that pacemakers follow to capture AF events.

The model is presented in Figure 3.2 as a continuous Markov model where there are three states: "Normal", "Arrhythmia" and "False Normal". A patient is in the "Normal" state until an arrhythmia is detected and the device outputs an AMS event, then the patient switches to the "Arrhythmia" state. There are two possible paths out of this state: back to "Normal" when the episode ends, or a transition to "False Normal" when a false end of episode is emitted. In this second case, the patient remains in the "False Normal" state until a new AMS episode is dispatched and returns to the "Arrhythmia" state. In this way, AMS events mark the beginning of a true AF episode or the end of a "False Normal" state. This second class of AMS events is abnormal and should be removed, but there is no simple procedure to remove them from the pacing data, so these events will be present in real patients, and therefore the digital twin must produce these spurious events as well.

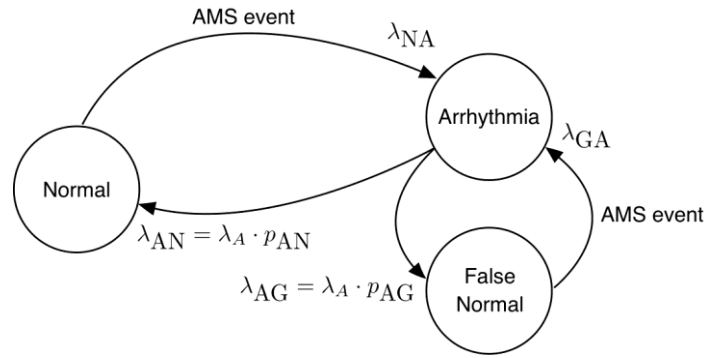


Figure 3.2.- State diagram of the model of AF episodes.

The time between two episodes is assumed to follow an exponential distribution with parameter λ_{NA} . The duration of an episode also follows an exponential distribution with parameter λ_A . The progression from paroxysmal to permanent AF is measured by the rate of change of these two parameters: as the cardiac condition worsens, the time between episodes is shorter and the episodes are longer. The speed of progression is modeled by a parameter $\alpha \in [0,1]$,

$$\lambda_{NA}(t) = \lambda_{NA}(0) \cdot \alpha^t, \quad (1)$$

$$\lambda_A(t) = \lambda_A(0) \cdot \alpha^{-t}, \quad (2)$$

where $\alpha = 1$ is a stable patient and values below 1 are patients with rapid progression to permanent arrhythmia. It is also assumed that the transition from "Arrhythmia" to "Normal" state can occur with a probability p_{AN} . The probability of the transition from "Arrhythmia" to "False Normal" is therefore $p_{AN} = 1 - p_{AG}$. p_{AG} is the fraction of false positives, which is the probability that the AF detection algorithm in the pacemaker signals the end of an episode too early.

In summary, the proposed generative model is a continuous-time Markov model characterized by 5 parameters (λ_{NA} , λ_{GA} , λ_A , p_{AG} y α). With this model, it is possible to generate a list of events that can be interpreted as a hypothetical patient whose AF type is defined by the above parameters. Specifically, six classes are generated (see Table 3.1) by varying the main parameters of the model: α y λ_{NA} . α measures the speed of AF progression and the inverse of λ_{NA} is used to model the mean time (measured in days) between two AF episodes. The values of α used are 999, denoting slow progression and 998, denoting rapid progression. For simplicity, to refer to the inverse of λ_{NA} , β will be used and the values chosen are 10, 30 and 180 days. Validation is then performed on real intracardiac data obtained from pacemakers and defibrillator systems.

α/β	10	30	180
999	Class 1	Class 2	Class 3
998	Class 4	Class 5	Class 6

Table 3.1.- Classes generated to model AF behavior.

Aircraft engines

In the case of aircraft engines, the Commercial Modular Aero-Propulsion System Simulation, also known as C-MAPSS, is used. It is a realistic data simulation tool for large commercial turbofan engines developed by NASA [48]. Each simulated flight is a combination of a series of flight conditions with a reasonable linear transition period to allow the engine to change from one condition to the next. The flight conditions are arranged to cover a typical climb from sea level to 35K feet and a descent to sea level. The failures or degradations attempt to reflect realistic situations and are injected at specific times during flights and persist over consecutive flights, effectively increasing the aging of the engine.

The C-MAPSS [49] dataset, generated with this simulation model, is widely known and used in the literature [50], [51], [52], [53]. The objective of this problem is to identify which flight and at what time of the flight the failure occurred and to infer its remaining useful life. This dataset consists of multivariate time series obtained from twenty-one sensors and is further divided into 4 subsets (see Table 3.2). In each subset, a training set and a test set are provided, of which there is a slight difference. The training set comprises the operating and failure data. That is, although each engine unit starts with different degradation states that are unknown, these are considered healthy and, as time progresses, the engine units degrade to failure, so the last data sample corresponds to the time cycle in which the engine is declared unhealthy (RUL = 0). In contrast, the sensor records in the test sets terminate sometime before system failure and the actual RUL value for these engines is provided. The aim is to estimate the RUL of each engine in the test sets.

C-MAPSS dataset is used for training and data sampled on actual Turbofan engines under different conditions of use is proposed for validation in a more realistic context.

	FD001	FD002	FD003	FD004
Train trajectories	100	260	100	249
Test trajectories	100	259	100	248
Operating conditions	1	6	1	6
Fault conditions	1	1	2	2

Table 3.2.- C-MAPSS datasets details.

Lithium-ion Batteries

As in aircraft engines, a simulation model widely known in the literature, Alawa [54], is used. Alawa is a mechanistic model that can enable battery diagnostics and prognostics. The model can simulate various "what-if" scenarios of battery degradation modes through a synthetic approach based on the specific behavior of the electrodes with appropriate adjustment of the charge ratio and the degree of degradation at and between the two electrodes. With the mechanistic understanding of battery degradation processes and failure mechanisms, it offers a unique high-fidelity simulation to address the path dependence of battery degradation.

Associated with this simulation model, the authors published a dataset for training diagnostic and prognostic algorithms [55]. The mechanistic approach combines modeling and experimental techniques to provide a universal tool for creating synthetic voltage curves that are practically indistinguishable from real data. This approach offers the advantages of broad applicability of the model to various cell chemistries, designs and modes of operation, as well as the high fidelity inherent in the detailed extraction of the experimental data. The data generated consists of two datasets, one intended for diagnostics, containing over 700,000 individual voltage versus capacity curves and a prognostics dataset with over 130,000 individual degradation trajectories for commercial batteries of different chemistries, LFP, NMC, and NCA. Despite being a recent dataset, its application is spreading rapidly [56], [57], [58].

For real data, cycling records from two commercial high-power graphite//LFP cells manufactured by A123 Systems (ANR26650M1, 2.3 Ah) were used. They were tested under different conditions, particularly under multistage fast charging and under dynamic stress test (DST) driving schedules.

3.1.2. Data preprocessing

Simulation models attempt to reproduce the behavior of real systems, therefore, the data they produce must also be consistent with the data recorded in these systems. Generally, raw registers provide very little information and need to be processed to obtain clean data that can be fed to Deep Learning models. This stage is known as data preprocessing and is particular to each problem, as we will see below.

Atrial Fibrillation

The data recorded by the pacemakers allow the start and end dates of each arrhythmia episode to be known. With this, it is possible to calculate for each day recorded the time that the patient was in AF to subsequently obtain the daily percentage of time in AF. This information can be plotted as in Figure 3.3 (left).

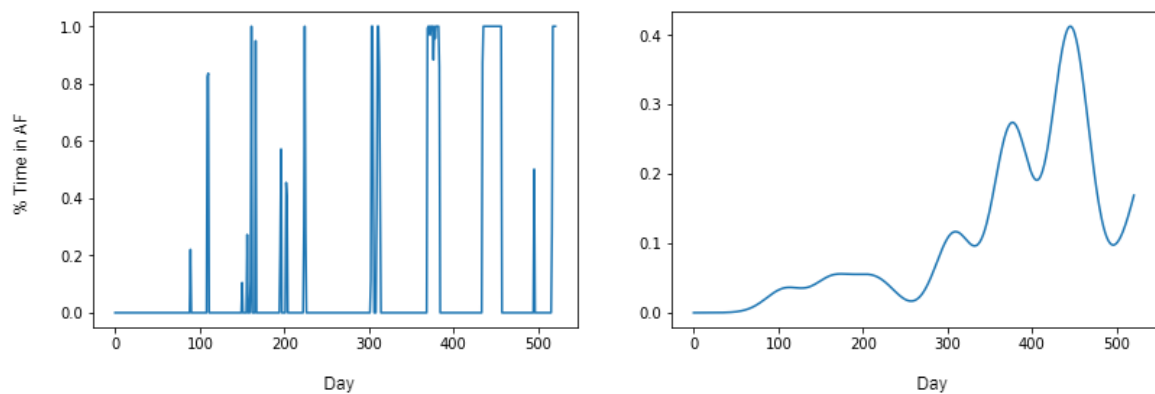


Figure 3.3.- Daily percentage of time in arrhythmia of a sample patient (left). Same sample with a smoothing applied (right).

As the events are sporadic, the morphology of the sequence does not resemble a classic time series because the sequence has very few time steps that provide important information. Moreover, this would be aggravated in those patients who are not in critical condition because the percentage of daily time spent in arrhythmia is very low or directly zero. This would cause that if it was decided to use this data as a training set, the performance of the network would be negligible. Therefore, from the previous step, a smoothing with Gaussian kernel is applied, taking as window width a few days. With this technique, the morphology of the previously obtained sequence is transformed in such a way that the evolution from paroxysmal to permanent arrhythmias can be appreciated.

Figure 3.3 (right) shows an example of the result obtained: the beginning of the plot corresponds to spontaneous arrhythmias, which, as time goes by, occur more frequently and are of longer duration, with the end of the plot coinciding with the transition to chronic arrhythmias, where the daily time percentage in arrhythmia is dangerously high.

With this transformation, the data would be ready to be fed to a neural network as they are also in a suitable range (between 0 and 1). The only drawback depends on the length of the sequences since each patient will have a different set of records. For this reason, the approach followed was to normalize and reduce the number of samples to a fixed size taking as the value of each sample the mean of the closest ones. Despite drastically reducing the length of the sequences, no information is lost because the morphology of the series is still maintained.

Aircraft engines

The records stored by the engines correspond to the readings of different sensors. It is worth mentioning that the engines can operate under different conditions, as shown in Table 3.2, therefore a simple Exploratory Data Analysis (EDA) would yield little or no information concerning the sensors because operating conditions may change throughout cycles, which makes analyzing and predicting RUL much more complex. This is essential when it comes to normalizing data, because, if all data is normalized in the same way, the meaning of the signals may be different depending on the mode of operation. As an alternative, a condition-based standardization is used. With this approach, all records of the same operating condition are grouped together and scaled using a standard scaler. The application of this type of scaling will bring the average of the grouped operating conditions to zero. As this technique is applied for each operating condition separately, all signals will receive an average of zero, thus making them comparable.

On the other hand, although sensor data does have a trend, it is known that it is subject to local oscillations, mainly caused by high-frequency sensors, which lead to noise [44], [59]. To ease the processing of the series, an exponential weighted moving average is carried out. It takes the current value and the previous filtered value into account when calculating the filtered value. Depending on the smoothing effect stationarity may be lost therefore the trade-off between smoothing and stationarity has to be taken into account.

Also, the data is split into sequences for better prediction performance. That is, multivariate series are not processed for each engine but are sliced into fixed-size windows as shown in Figure 3.4. At each time step, data is picked from sensors within the time window to form a high-dimensional feature vector used as input to the network to predict the RUL. Thus, each input sample contains a fixed size of samples that are extracted from the sensors, which are another hyperparameter of the network. The aim is to find patterns in those time windows that can lead to an adequate RUL estimation. There may be cases in which the partitioning

of the sequences for a particular engine in the last few cycles may not have enough data to complete the length of the window. In those cases, a masked value is used and will be treated in the network by simply ignoring those values. In this way, as much information as possible is used.

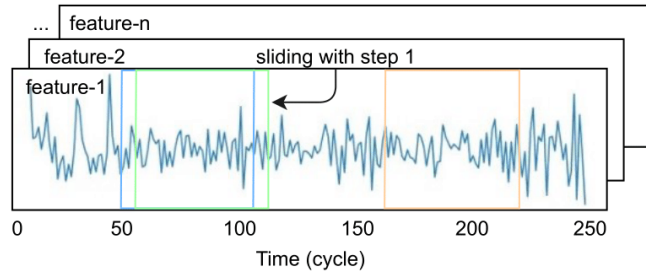


Figure 3.4.- Time window framing.

Lithium-ion Batteries

The data collected by batteries is limited, especially in real-world situations, although voltage and capacity records can usually be obtained over the cycling history of the battery. The representation of capacity vs. voltage curves (Figure 3.5 , left), on the contrary, provides little information because the changes are not significant between different types of degradation. Because of this, a more widespread representation in the battery field is the use of the derivative of capacity vs. voltage curve (Figure 3.5, right), known as Incremental Capacity curve (IC). This representation produces a series of peaks, also known as Features Of Interests (FOIs), that depend directly on the chemistry from which the battery is made and are deeply studied in the literature [60], [61], [62]. The evolution of these FOIs throughout the cycles that the battery undergoes is investigated to know the type of degradation that is occurring and thus diagnose its state of health.

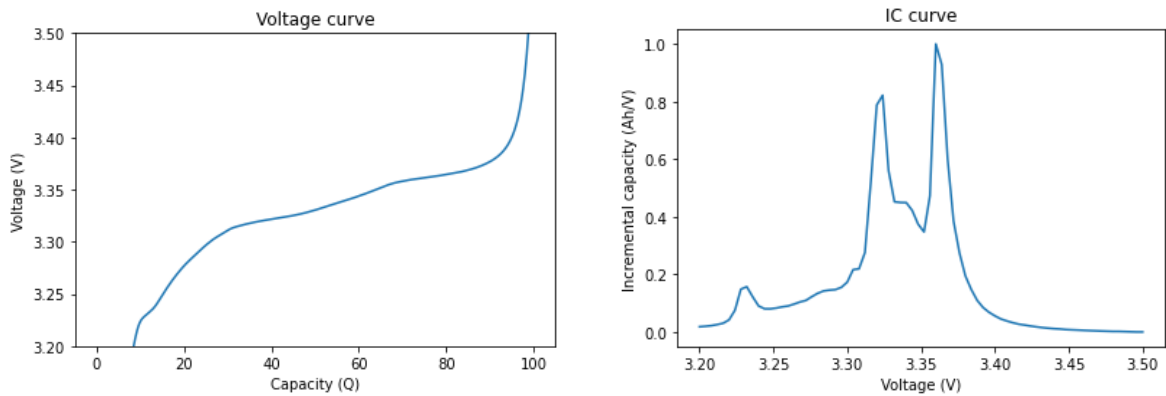


Figure 3.5.- Capacity vs voltage curve (left). IC representation (right): voltage vs derivative of capacity.

This process can be automated with the use of Deep Learning models and in fact there are several works that point in that direction [57], [63], [64], [65]. However, the representation of the data depends on several factors such as battery usage or the internal configuration parameters of the cell, so the application of a model trained on simulated data is hardly applicable to real-world data. Because of this gap, a new representation of battery degradation data was sought in order to benefit from advanced DL models. It was proposed a representation based on Dynamic Time Warping (DTW) consisting of an image highlighting the differences between the IC curves of a pristine and aged battery [58].

DTW [66] is an algorithm used to measure the similarity between two sequences. First, the Euclidean distance between each pair of points between the two sequences is calculated in a matrix. Among these distances different warping paths can be found, that is, possible deformations that a sequence should follow in order to be as similar as possible to the other. The method quantifies the similarity between the sequences by finding the best warping path, which corresponds to the one with the smallest accumulated distance. Figure 3.6 (left) presents the example of the application of DTW to two sine waves, referred to as Sin #1, located in the left part of the grid and Sin #2, located in the upper part of the grid, which shows a small deformation in the second period. The best path found in the matrix is marked in blue and indicates that for the Sin #2 to be the same as Sin #1, the deformation to follow is to slightly raise the values between 15 and 20. The similarity between the two sequences can be quantified with the resulting distance, i.e., the accumulated Euclidean distances of the path, which is 0.1946. At the lower left and upper right corners, the values are marked as inf (infinite) because there are no deformation paths that extend that far, so they are not calculated in order to reduce computation time. The method developed originally for speech recognition, and it is widely used for classification and clustering tasks [67], [68], [69], [70].

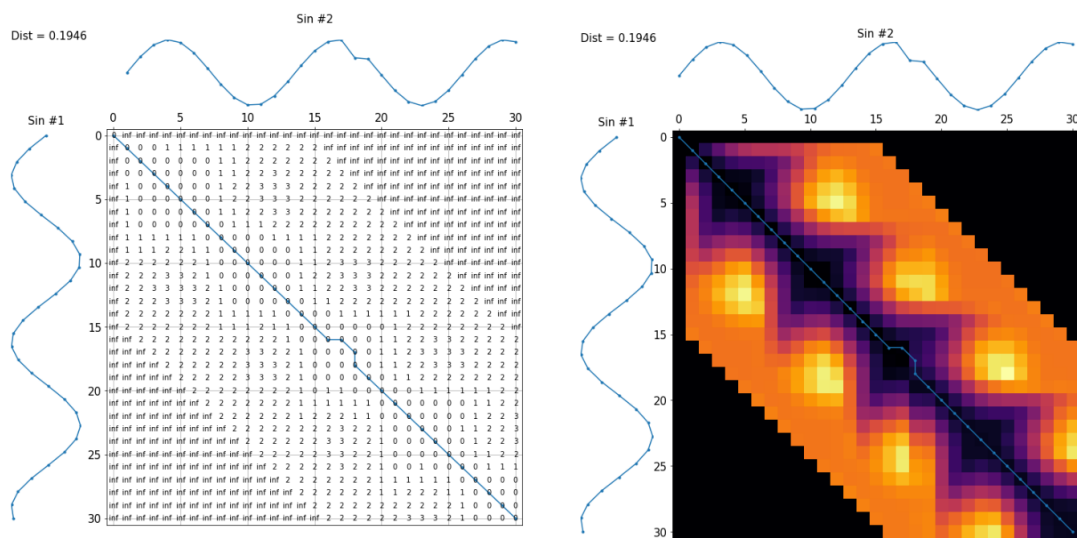


Figure 3.6.- Euclidean distance between each pair of points of the two sequences displayed on a grid (a). Every warping path represented as a set of pixels (b). In both images the optimal warping path is marked in blue.

DTW was already applied to the estimation of Li-ion battery capacity [71], [72], [73] as well as for augmenting the data obtained from different operating conditions [74]. However, these works make use of the similarities found in the best warping paths. Instead, it was proposed for the first time to use the full matrix, represented as a set of pixels (see Figure 3.6, right) and thus as an image. IC curves were used as sequences, one pristine and one aged. An image can be generated for each sample in the data and each image thus represents the similarity between the corresponding IC curve and the pristine one. Since each degradation path leads to a unique voltage response, it will also result in a unique image. As an example, Figure 3.7 depicts the IC curves corresponding to 20% of each of the three degradation modes considered: LLI, LAMPE and LAMNE (dashed lines) with the reference IC curve (solid lines) and their resulting images, labeled with the final DTW distance. This is to showcase that, just as the IC curves after different degradation are unique, the images are too. In these images, changes are reflected in shape, symmetry and colors. Note in the first degradation, LLI, the main peak located at 3.37 V is lost while in LAMPE the peak that disappears is the minor one, located at 3.23 V. The images associated with these degradations also change, specifically in the intensity of the purple color, as well as in the symmetry, which is mainly lost in the first image, and consequently, the distance is greater, 0.77 vs. 0.31. In the LAMNE degradation, the appearance of the peak at 3.45 V represents a sign of lithium plating in LFP cells [75]; On the image, this translates to the appearance of a lighter color band that coincides exactly with the position of the peak. The changes in this degradation are much more significant, and accordingly, the final calculated distance is greater: 1.53. In the end, just as with studying FOI variations, the degradation modes are decipherable from these unique images and so image processing algorithms such as CNNs can be undertaken.

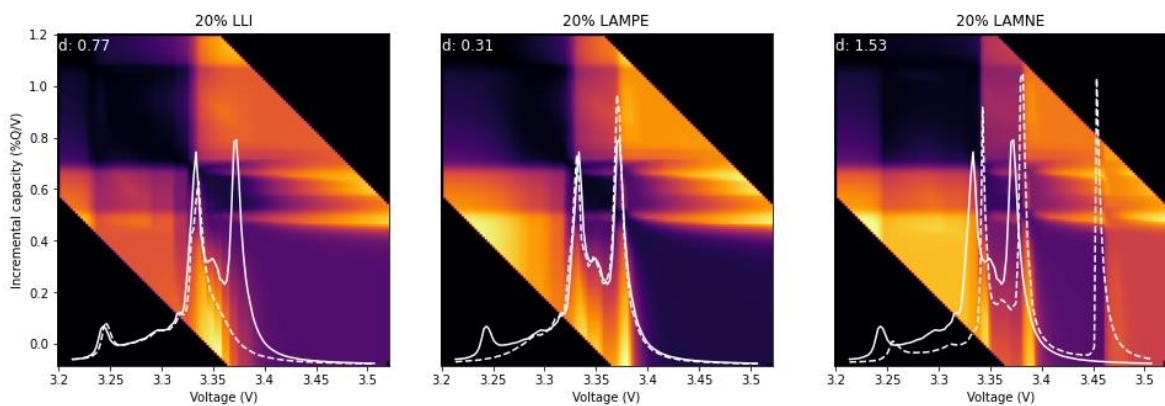


Figure 3.7.- IC signatures from the initial state (solid line) for each degradation in the dataset: LLI (a), LAMPE(b) and LLI(c) at 20% degradation.

A key property of these images is that they preserve the representation of the degradation modes regardless of the cell configuration. While the images were gathered from a dataset composed of synthetic curves, the differences between the pristine and aged IC curves should be similar for cells with slightly different cell configurations. In the simulation model, a cell is defined by its active materials and two additional parameters, the loading ratio (LR), which corresponds to the electrode capacity ratio and the offset (OFS), which corresponds to their slippage compared to one another. Based on cell-to-cell variations studies [76], variations of LR by ± 0.2 and OFS by $\pm 2\%$ were estimated possible within a batch. As an example, images associated with different cell configurations for the same degradation (20% of LLI) are presented in Figure 3.8, with varied parameters to simulate cells from the same batch with slightly different properties (± 0.01 for LR, $\pm 1\%$ for OFS). Visually, the three images are almost identical, and this is confirmed by the final DTW distance that were 0.65, 0.66, 0.62, respectively to be compared to the 0.77, 0.31, 1.53 for LLI, LAMPE and LAMNE degradations on Figure 3.7. This is a key factor when applying the procedure to batteries with different operating modes or cell configurations, especially since batteries from the same batch have some cell-to-cell variations and batteries from different manufacturer might not use the same materials, additives or loading. This differentiates the proposed method from other models trained on synthetic data that might not be applicable to real data.

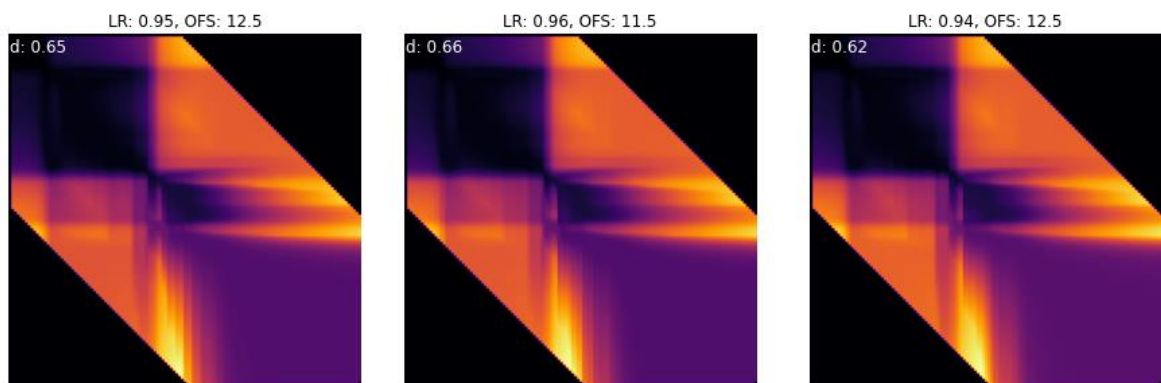


Figure 3.8.- DTW Images for 20% LLI degradation for three different cell configurations.

To demonstrate the performance of the approach in a more realistic application context, a demo is provided in <https://huggingface.co/spaces/NahuelCosta/DTW-CNN>. Different cycles associated with three LFP test cells can be selected to display their IC curves, the corresponding DTW image and the final diagnosis given as the percentage of each predicted degradation mode using a CNN.

3.1.3. Summary

Table 3.3 presents a summary of what has been explained in this section. The rows show the simulation models to produce the training data (labeled as “Digital twin”), the real data to evaluate the developed DL models, the preprocessing steps followed and the type of problem to be solved in each topic.

	FA	Aircraft engines	Batteries
Digital twin	Markov model	C-MAPSS	Alawa
Real data	Registers from pacemakers	Turbofan engines	Commercial high-power LFP cells
Preprocessing	Smoothing + downsampling	Smoothing + condition standardization + time window framing	IC + DTW
Type of problem	AF estimation: Classification	RUL estimation: Regression	Degradation modes estimation: Regression

Table 3.3.- Summary of data, preprocessing steps and types of problems.

3.2. DEEP LEARNING-BASED SOLUTIONS

This section presents the models developed during the realization of this thesis, which have been validated on the different topics explained above.

3.2.1. Variational Autoencoder (VAE)

In Chapter 2 the relevance of autoencoders in condition monitoring was discussed. The workflow of these models resembles an hourglass structure where at the bottleneck (latent space) a compressed encoding of the input data is learned and then reconstructed back to its original dimension (remember Figure 2.1). The main limitation of conventional autoencoders is that the inputs are encoded to a fixed set of vectors in the latent space. This makes the model suitable for reconstructing the input data, however, if an area of the latent space not associated with any of the learned vectors is taken as a starting point for reconstruction, the output would be arbitrary.

In contrast, VAEs [26] do not encode the information to a set of fixed vectors but learn the probability distribution of the input data. This allows the decoder to reconstruct data from areas of the latent space that do not belong to the compression of any input, thus generating completely new samples. An example of this can be seen in Figure 3.9: the left side of the figure corresponds to the latent representation of geometric figures by a conventional autoencoder. There are samples that are not similar that are encoded in nearby areas of the latent space, so that, if some intermediate point is chosen, for example, between the rectangle and the rounded triangle, the resulting decoding does not make sense (figure in red). On the other hand, on the right-side similar data are encoded in nearby areas, therefore, if it is interpolated again between the rectangle and the rounded triangle the resulting figure will be a cross between the two figures, thus favoring the generation of new data.

This organization in the latent space is achieved by forcing the encoder produce not one vector of fixed size, but two vectors: a vector of means μ and another vector of standard deviations σ , which define the probability distribution of the data. In this way, the reconstruction is done from a sample of the latent space that follows the same distribution, which allows the VAE to be a generative model.

Generative models in recent years have had an extraordinary impact in several areas [77], [78], [79], mainly due to models like VAEs, GANs [80] or the recent diffusion models [81], [82]. However, VAEs are also actively applied in anomaly detection [83], [84], [85], as by learning the distribution of the data they are trained on, they can detect instances that do not specifically follow the learned distribution.

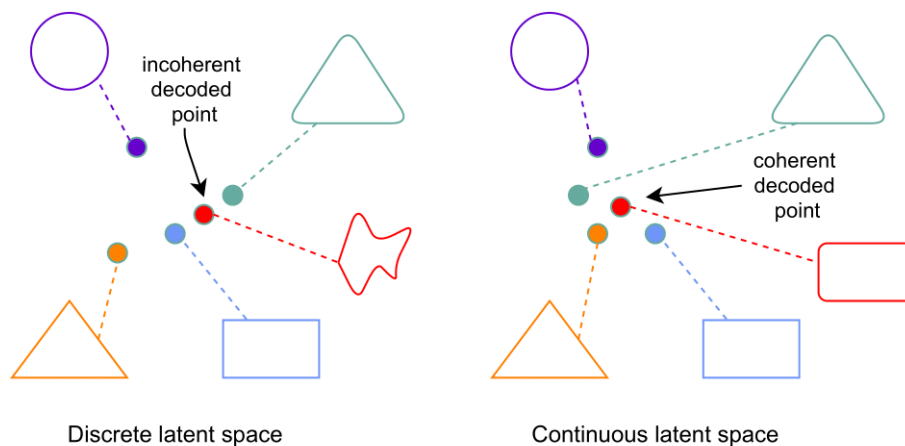


Figure 3.9.- Simplified representation of the compression resulting from a vanilla Autoencoder (left) and a VAE (right). When the latent space is continuous, the organization of the data allows decoding a meaningful figure, in this case a cross between a rectangle and a triangle, thus favoring the generation of new data.

Achieving such a latent space is an example of a Representation Learning approach. The performance of Deep Learning models is highly dependent on the representations learned. Typically, an algorithm capable of learning the features that best represent the underlying distribution of the data is required, so it makes it easier to perform other tasks such as classification or regression. A model such as a VAE is potentially useful in this regard, as its feature extraction capabilities make it easily interpretable. Other unsupervised techniques, such as clustering algorithms, can also be applied for this purpose as they do prioritize grouping data of a similar nature, however, the visual disposition of the clusters can often be arbitrary. On the other hand, neither can the VAE latent space be used for clustering since the encoded data tend to be overlapped to prioritize the generative process.

Furthermore, although VAEs have been shown to be efficient in multiple domains, mainly related to Computer Vision and NLP problems, there is a large gap in research on these models for sequential data. In [15] the authors presented a VAE that can map time series to a latent vector representation, but the model has become obsolete due to more recent advances in recurrent architectures. Other promising work has started to emerge: in [86] LSTM networks are used to model the temporal complexion of data, while in [87] the authors propose to use echo-state networks for the same purpose. Although these works combine recurrent architectures with VAEs, their goal differs from ours, as they aim to detect anomalies based on reconstruction errors or anomaly scores, while what we pursue is an interpretable assessment of the time series.

The solution proposed is to introduce a recurrent version of the VAE to deal with temporary data. If a VAE is trained with monitoring data of several systems that have undergone a wide range of degradations, the resulting model will be capable of projecting different types of degradation of the systems in different areas of the latent space. The key is that similar

samples are located in nearby areas of the latent space. Also, this representation can be used for other downstream tasks such as the classification of different types of arrhythmias or the estimation of RUL in the case of aircraft engines (Figure 3.10). In fact, the inclusion of a classifier or a regressor in the training process helps control the regularization of the latent space, thus preventing the resulting clusters from overlapping.

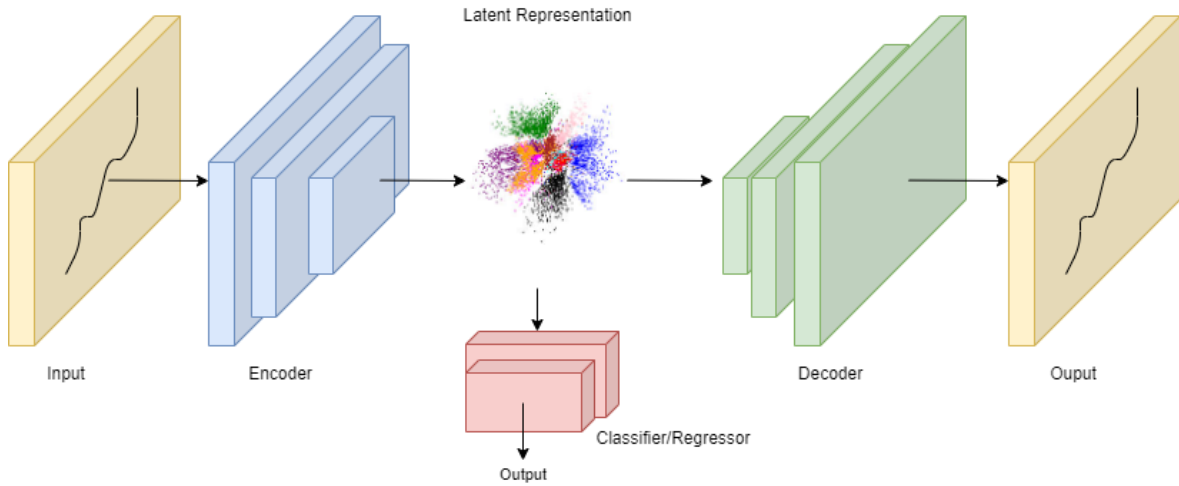


Figure 3.10.- Network structure of the proposed recurrent VAE. The blue and green blocks are the encoder and decoder respectively and the red block represent a downstream task such as classification or regression.

The model is trained with simulation data so that when a sample from a real system is fed to the model, the projection on the latent space is expected to be located on a specific group corresponding to those samples of the training set with similar properties. In this way, the latent representation can be considered as a projection of the simulation model parameters that best fit the criticality of the given sample, thus being able to know its degradation stage.

The following subsection is intended to explain the core parts of the model, which are mainly the encoder and the loss function.

3.2.1.1 Model settings

In a VAE the training is regularized to avoid overfitting and to ensure that the latent space has good properties that allow the generative process. Precisely, these properties contribute to the input data being mapped in the latent space in such a way that similar data are nearby, and that this representation can be used as a feature extractor.

A VAE given an input, tries to find a latent vector that can describe it and at the same time has the instructions to generate it again. The process can be described as: $p(x) = \int P(x|z)p(z)dz$. Given that the integral of this formula is intractable due to the continuous domain of z , the Bayesian variational inference is needed via the lower bound of the log-likelihood, L_{VAE} ,

$$L_{VAE} = E_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}(q_{\phi}(z|x)||p_{\theta}(z)), \quad (3)$$

The first term is the reconstruction of x (the input data) that tends to make the coding-decoding scheme as efficient as possible by maximizing the log-likelihood $[\log p_{\theta}(x|z)]$ with sampling from $q_{\phi}(z|x)$, which represents the encoder, whose output are the parameters of a multivariate Gaussian: a mean and a diagonal covariance matrix. The second term tends to regularize the organization of the latent space by causing the distributions returned by the encoder to approach a standard normal. It regularizes the latent variables (represented by z) by minimizing the KL divergence between the variational approximation and the prior distribution of z . With that being said, the focus is mainly placed on the encoder as its goal is to map the input data into a lower dimensional space, specifically to a two-dimensional latent space dominated by the mean and the variance of the approximated distribution.

To deal with the time dependence of the monitoring data, instead of using convolutional layers, as vanilla VAEs do, the encoder is implemented with RNNs. Bidirectional LSTMs are chosen because of their ability to run from past to future but also from future to past, thus preserving information from both periods. This is very valuable due to the fact that the network is aware of how the data may look like in its future stages, so it can help to understand what kind of information to predict. The code of the model can be found in <https://github.com/NahuelCostaCortez/RVAE>.

There is an additional problem, as mentioned above, VAEs are mainly oriented to enhance the generative process, and this causes the regularization of the latent space to lead the encoder to project the data as compressed as possible, resulting in obvious overlaps. This is a barrier to our objectives because these overlaps make it difficult to correctly differentiate the different stages of degradation. First visually: although similar instances will be close in the latent space, they will not be clearly differentiated from those that are far away. Then, because any model built on top of this will be guided by this representation and will most likely result in prediction failures. Therefore, the training of a vanilla VAE does not meet the stated needs so the use of variational inference must be adapted. The image on the left in Figure 3.11 represents why this is not suitable. It corresponds to the latent space the encoder learns for one of the aircraft datasets after training the model without any restrictions thence the regularization of the latent space for the generation of new data is prioritized. This causes the input data to be placed in areas where instances whose features are not similar are not clearly differentiated or even overlap.

Instead, the proposed modification includes omitting the decoder, as it is not used and may wrongly interfere in the training process and focus the learning on obtaining an interpretable latent space. Thereby, the main difference with respect to a VAE is that the decoder is replaced by a classifier or a regressor model, depending on the needs of the problem, so the proposed model is trained to minimize a loss function composed of two objectives:

$$L_x = -D_{\text{KL}}(q_{\phi}(z|x)||p_{\theta}(z)) + L_{\text{classification/regression}} \quad (4)$$

The first objective corresponds to the regularization of the latent space through variational inference, as explained before in Eq. (3) and the second can be the root mean square error (RMSE) in regression problems or the categorical cross-entropy in classification problems. The inclusion of the optimization of any of these models in the loss function adds a constraint to the autoencoder, as it will strive not only for a continuous latent space but also a space in which the different aging sequences are sufficiently separated to be differentiated so that the rate of deterioration can be observed over the life cycles of the system to be monitored. The code of this modification is available at <https://github.com/NahuelCostaCortez/Remaining-Useful-Life-Estimation-Variational>. The right part of Figure 3.11 demonstrates the effectiveness of training the model in this way for the aircraft engines problem. In this case, the aim is to estimate the RUL, so a simple regression model is used: on top of the encoder base, a fully connected layer with a tanh activation function and another layer with a single neuron containing the RUL prediction are added.

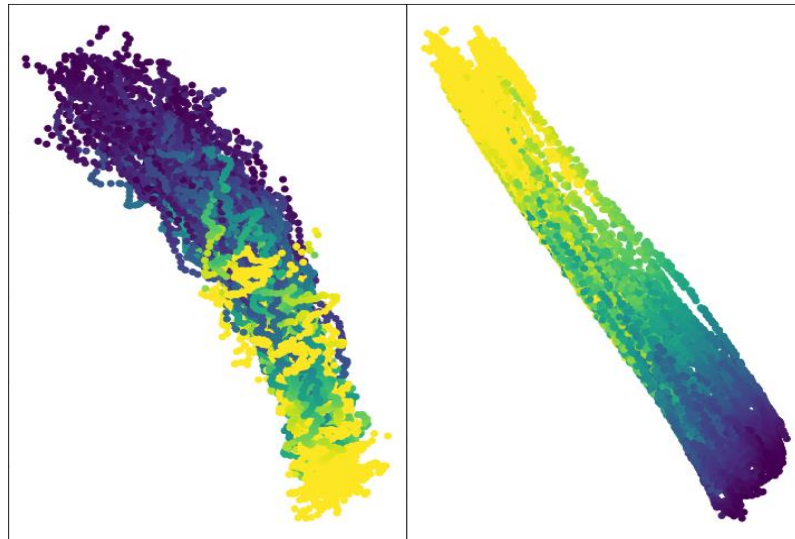


Figure 3.11.- Latent representations learned by the encoder for one of the aircraft datasets. The figure on the left shows the regular training of a VAE, while the figure on the right shows the result with the proposed model, which does not include the decoder but a regression model that adds a penalty for wrong predictions.

The latent space can be interpreted as a diagnostic map from which to understand the nature of the data. This is one of the main objectives for the achievement of an explanatory and interpretable model and will be further explained in section 3.3 EXPLAINABILITY.

3.2.2. Generative Adversarial Networks (GAN)

As explained previously, generative models are a hot topic in Deep Learning. Together with VAEs, GAN networks [80] have been a major breakthrough in this regard with applications already widely integrated into different fields such as video game [88] and fashion industry [89], image editing [90], [91], or medical imaging [92].

The architecture consists of two neural networks that are able to learn based on the feedback received by each other. On the one hand, there is the generator network, which receives as input a vector of completely random numbers or noise, from which it tries to generate new data.

On the other hand, there is the discriminator network, whose task is to identify whether the data generated by the generator belongs to the same distribution of the input data or not.

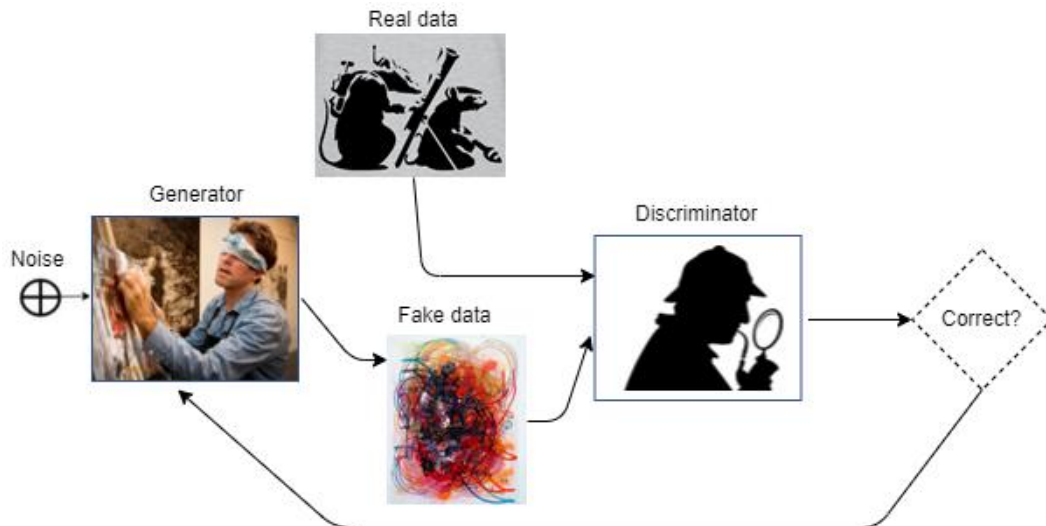


Figure 3.12.- Workflow of GAN networks.

Figure 3.12 illustrates how the model works. The discriminator has access to the data on which the model is trained, and its task is to learn the probability distribution that better describes it. The objective of the generator is to generate data that the discriminator considers to be from that distribution. At first, the generations will be completely random, and the discriminator will not be able to identify with a high level of confidence whether they are

realistic, so both networks adjust their weights to improve on their objectives until they reach a point where the generator is able to fool the discriminator and the discriminator learns the distribution of the data.

As with the decoder in the VAE, although the generator is the most used component of these models, for condition monitoring problems the discriminator is of greater interest due to its ability to faithfully learn the distribution of the data. If a GAN is trained with simulated data generated under certain parameters with a digital twin, the resulting discriminator will be able to recognize only that type of data. Therefore, if a set of GANs is trained with data covering a wide range of degradation types, a set of discriminators will be obtained that will be activated only if the data fed to the model match situations similar to those of the data with which they were trained (Figure 3.13).

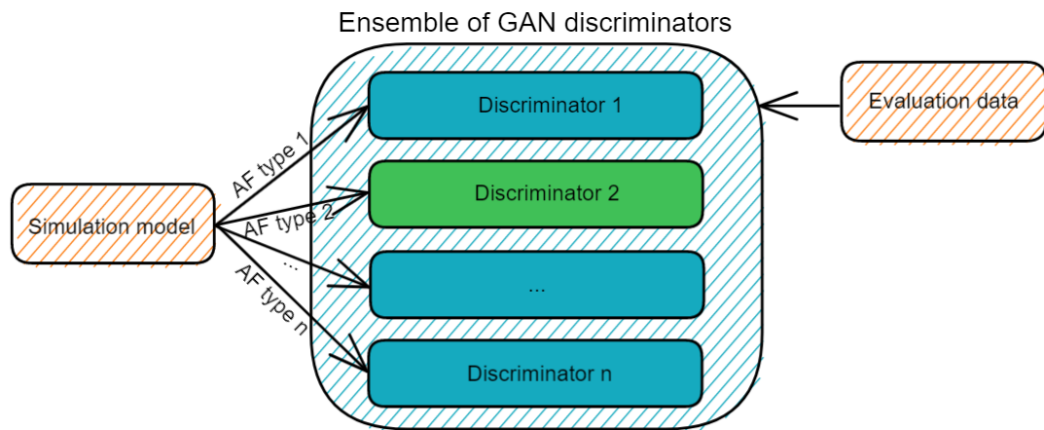


Figure 3.13.- Ensemble of GAN discriminators, each trained with different classes of AF. When feeding evaluation data to the ensemble at least one of the discriminators is expected to be activated.

GAN research is largely oriented to convolutional networks for computer vision problems and hardly to sequential data, so the proposed model is customized with recurrent networks in order to deal with the temporality of the data. The code can be found in <https://github.com/NahuelCostaCortez/FA-GAN>. The following subsection describes the different parts of the model.

3.2.2.1 Discriminator network

The discriminator network is implemented with a recurrent network, specifically with LSTM cells. The input receives the data to be evaluated and as output produces a state vector with the information process by the LSTM units (Figure 3.14).

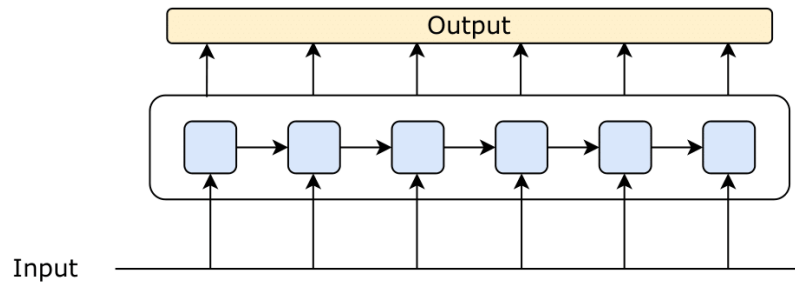


Figure 3.14.- Discriminator network structure.

First, the training data is fed to the model and then data produced by the generator. The objective is to classify the sequences coming from the training set as true and those coming from the generator network as false, therefore, the output of the network will establish the veracity of what it receives as input.

For each sequence that is received (consisting of a fixed number of samples), there is an output for each sample in the sequence. Each output is a number indicating the probability of that sample being part of a real sequence. The hidden state of the last block of the RNN is normally used since it contains the relevant information of the previous elements of the sequence and then it is passed through a function or linear layer to predict an output. On the contrary, it is proposed to use all the outputs of the RNN (passed through a linear layer) in order to measure for each time interval whether the sequence to be evaluated is sufficiently realistic; this is important to then adjust the generator parameters, which may have learned to generate correctly specific parts of the sequence, but not others.

As this architecture is applied to the AF estimation, which is a classification problem, each network is trained to minimize the average cross entropy between the predictions and the real values for each time step. Thus, the loss function can be defined as follows:

$$D_{\text{loss}}(X, y) = \text{CE}(D_o(X), y), \quad (5)$$

where X are the sequences fed to the network, y are the real values of the sequences: it will be a vector of 1s for real sequences or a vector of 0s for synthetic sequences and CE is the cross-entropy between two values: D_o , the output generated by the discriminator from the inputs and y .

Since in each iteration the network "visualizes" both real and synthetic data, the loss is calculated for each type of data and the final loss value will be the sum of both losses, the one obtained with real data and the one obtained with synthetic data:

$$D_{\text{loss_final}} = D_{\text{loss_real}} + D_{\text{loss_synthetic}} \quad (6)$$

3.2.2.2 Generator network

In this case, although the generator is not used after training, it is key for obtaining an accurate discriminator. It is another RNN, formed by a LSTM structure similar to that of the discriminator. It receives an input tensor and generates a sequence (Figure 3.15).

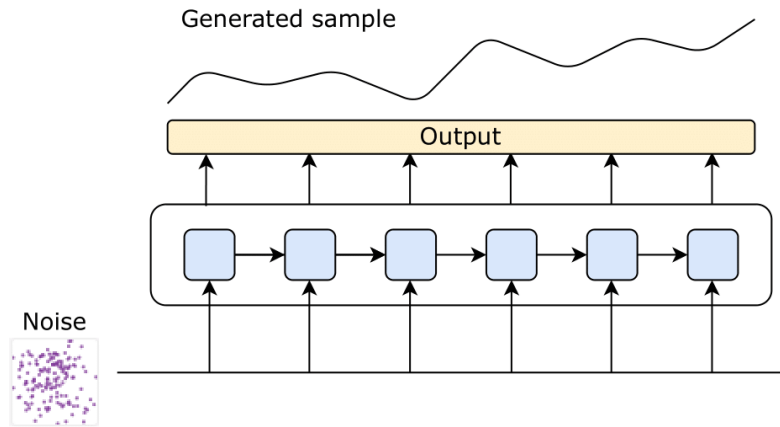


Figure 3.15.- Generating network structure.

The input that the network receives is noise, in order to produce sequences that resemble as closely as possible the data of the real distribution. This noise is produced randomly from a normal distribution with mean 0 and variance 1. All the outputs of the network are also used here, since they will form the synthetic sequences that will then be passed as input to the discriminator.

The goal of the generator is to fool the classifier in its task of sequence classification, so that it classifies what it generates as true. This translates into minimizing the average cross-entropy between the predictions made by the discriminating network on the data passed to it by the generator and the target predictions. The target predictions will be a vector of 1s since the prediction for synthetic sequences should be as close to 1 as possible, which indicates that the sequence is evaluated as real.

The loss function of the generator, therefore, is as follows:

$$G_{\text{loss}}(Z) = D_{\text{loss}}(\text{Do}(\text{Go}(Z)), 1) = \text{CE}(\text{Do}(\text{Go}(Z)), 1), \quad (7)$$

where Z represents the noise, G_0 the output of the generator, and D_0 the output of the discriminator.

3.2.3. Convolutional Neural Networks

In the case of CNNs, no modification in the architecture has been proposed but it has been adapted to the problem to be solved. CNNs consist of multiple layers of neurons, the structure of the proposed model for Li-ion battery degradation modes identification is depicted in Figure 3.16. The detailed description of each layer is as follows:

- **Masking layer:** this layer is used to mask data to be omitted by the next layer. This is particularly useful for the DTW images, where there are areas of the image that do not provide information and can therefore be omitted.
- **Convolutional layers (Conv1 to Conv4):** these layers are composed each of a conv2D layer (light orange) and a Max-Pooling layer (dark orange). The conv2D layers consist of multiple filters, which are applied to the image to highlight certain features that make the image unique such as the direction of the lines or their shape. The resulting images are known as feature maps. 64 filters are applied in each of the first two layers to obtain the features maps that mainly characterize the image, while in the last two layers more filters are needed (128 each) to capture finer details like color intensity or brightness. The Max-Pooling layer reduces the spatial size of the feature maps and learns to ignore irrelevant and redundant information, that is why the dimension of the blocks is reduced in each layer.
- **Flatten layer:** after the convolution and max pooling flow, the shape of the matrices is flattened to a single vector containing all the information needed for predictions.
- **Dense layer:** this layer applies a sigmoid activation function to obtain a value between 0 and 1 representing the percentage prediction of each of the degradation modes.

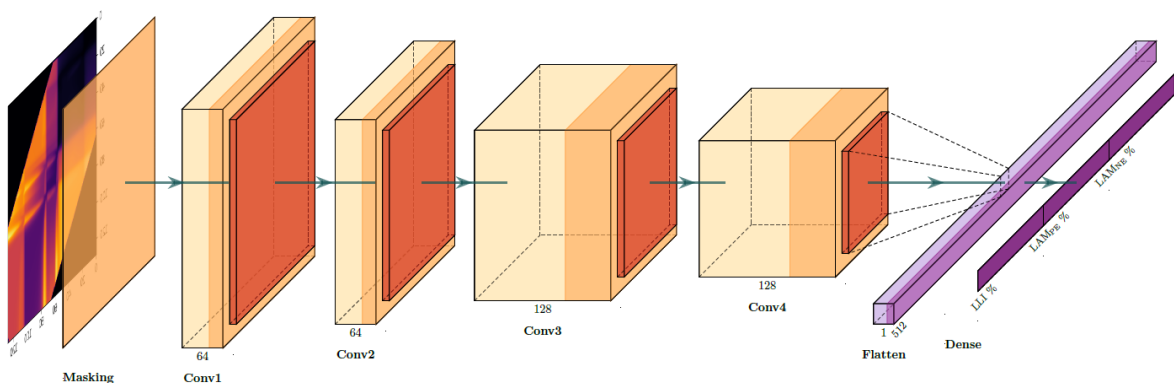


Figure 3.16.- Model architecture for Li-ion degradation modes diagnosis. Conv1 to Conv4 represent the convolution layers followed by the max pooling layers. The features extracted are condensed in a flatten layer from which the 3 degradation modes are predicted.

3.3. EXPLAINABILITY

The evolution of AI systems has reached a point where human intervention is barely required for their design and deployment. In this context, when the decisions derived from these systems end up directly affecting business processes or even the lives of human beings, there is an emerging need for understanding how such decisions are made [93].

Traditional AI systems such as decision trees, decision rules and linear regression are easily interpretable, however, the empirical success of Deep Learning models such as neural networks have made this process difficult. These networks usually have a huge parameter space comprising hundreds of layers and millions of parameters, which makes them to be considered as complex black-box models [94].

As black-box DL models are increasingly used to make important predictions in critical contexts, the demand for transparency is increasing from the various AI stakeholders. The danger lies in creating and using decisions that are not justifiable, legitimate, or simply do not allow for detailed explanations of their behavior. Explanations that support the output of a model are crucial, especially in condition monitoring, where experts require much more information from the model than a simple numerical prediction.

The methods developed during the course of this thesis have sought to contribute to this direction in order to provide a better understanding of how the models learn as well as the nature and relationship between the data.

One of the common approaches to understand the models' decisions is to examine the activation of internal parameters such as the neurons in different layers. Other models or techniques can be built over this to extract further information. Thus, the mechanisms developed in this thesis have used this approach to build visual explanatory tools.

The use of digital twins offers the possibility to adjust the simulation parameters. This is an advantage because when evaluating real data, it is possible to identify within this set of parameters those that best explain the situation and therefore the criticality of the system to be monitored.

3.3.1. GAN ensembles

It is recalled that GAN networks were used not for data generation but to exploit the discriminator's ability to learn the distribution of the input data. Particularly, an ensemble of GAN networks was trained with data reflecting AF behavior in such a way that a set of discriminators was obtained, each being capable of detecting a simulated AF class.

When patient data is fed to the ensemble, each of the discriminators, which is a binary classifier, will provide a numerical response corresponding to the degree of confidence with which the classifier recognizes the given records. Each classifier has a sigmoid activation in the last layer that will determine the output in the range $[0,1]$, so it is expected that the classifier with the closest output to 1 will be the one that has been trained with the data that best describes the patient's situation. This data is simulated by the digital twin and therefore its properties are known, from which the properties of the actual patient are inferred.

This numerical information is fundamental because, although it is of interest to classify the type of arrhythmia, it is also important to know the similarity to other types of arrhythmias in order to deduce the speed of its evolution. Thus, it is proposed to organize the activations of the last layer of each discriminator in a graphic map.

Figure 3.17 presents two maps to illustrate the method. The horizontal axis is labeled β , which is the inverse of the parameter λ_{NA} , and can be understood as the expected number of days between two AF episodes. The vertical axis is denoted α and measures the speed of arrhythmia progression. The lower the α value, the faster the progression to permanent AF. The color code is shown in the bar at the right. Red areas are the highest activations, and blue areas the lowest.

The ensemble outputs belong to six types of AF (remember Table 3.1) that are within a given set of simulation parameters. Working with only six values would provide a map that is not very explanatory because the activations would be concentrated in one area. To solve this problem, it was proposed to use a Kriging interpolator [95]. Kriging is a method that uses a limited set of sampled data points to estimate the value of a variable over a continuous spatial field. It is typically used in geostatistics but in this case it was applied to simulate that the map had activations of many more discriminators.

The first map shows the output of the set of discriminators when the input is a synthetic sequence of AF recordings, generated by the Markov model, with $\lambda_{NA} = 1/90$ and $\alpha = 0.999$. It is expected that only one of the detectors (or only a few) will react to this artificial sequence. The map obtained corresponds to what is expected, obtaining the predicted parameters from the most intense colored area in the map (red zone), with the estimation of these parameters corresponding to $\beta = 90$ and $\alpha = 0.999$, i.e., the model prediction corresponds exactly to the type of arrhythmias fed to it.

On the right side of the same figure is a second map with a projection of an AF sequence of events taken by a real pacemaker. Here there is not a clear identification as before, possibly because the records do not exactly follow the Markov model. However, the projection of the sequence in the given parameter space gives a decent insight on the evolution of the patient. In particular, the dark red area at the bottom of the map is compatible with a value of $\beta \approx 120$ and with a rapid progression to permanent AF, $\alpha = 0.998$.

The major difficulty of the model lies in the number of records for each patient. A large set of AF episodes is not compatible with an early diagnosis: if many captured records were treated, the information would surely be sufficient to know the patient's condition accurately. The interest of the study is focused on being able to get an insight of the patient's condition with as few records as possible in order to act prematurely and avoid potential future interventions. For this reason, it should be pointed out that the achievement of these maps is a step forward obtaining explanatory estimates with imprecise information.

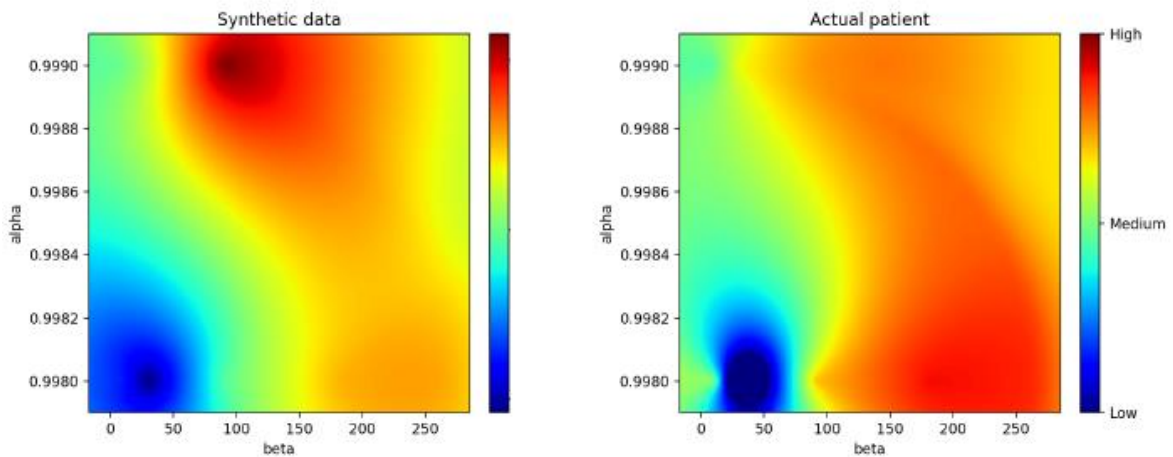


Figure 3.17.- Left: projection of AF events using the Markov model, with $\beta = 90$ and $\alpha = 0.999$. Right: projection of a sequence of AF events from a real pacemaker. The map is consistent with a value of $\beta \approx 200$ and a rapid progression toward permanent AF, $\alpha = 0.998$.

3.3.2. Variational encoding

The compression the proposed encoder does on the latent space is projected in two dimensions, which belong to the vector of means and variances of the learned data distribution. These vectors can be used to construct a two-dimensional map showing the location of the training data, whose diagnosis is known, so that when feeding data from a real system, the degradation it may be suffering can be immediately seen. This idea has been tested on intracardiac and aviation data and in both cases the proposed tool was able to show the deterioration rate of the system to be monitored.

For the case of AF, the latent space is displayed as a map showing the actual state of the patient and the rate of change from paroxysmal to permanent AF. Once the encoder is trained with the data from the Markov model, a topological map is obtained in the latent space from which the groups corresponding to the different types of arrhythmias are identified, as can be seen in Figure 3.18 (left). As the map is organized, it is evident that the values of β are located from left to right from highest to lowest (180, 30, 10), which is equal to an organization from lowest to highest criticality as low values of α indicate short times between different episodes. On the other hand, the values of α are organized from top to bottom (999,

998), from less to more critical. This information can be used to facilitate a better interpretation of the map. The upper right zone denotes the less critical arrhythmias, while the lower right zone shows those arrhythmias that represent a very advanced stage of the disease. At the same time, the rest of the parameters of the simulation model during the generation of the training set have been varied randomly, which slightly influences the condition of the arrhythmias, therefore this property can give rise to the interpretation of arrhythmias between two clusters as an interpolation between the parameters of two classes.

When real data is used as input to the model, i.e., a patient's intracardiac data, the encoder will place them according to their characteristics, in an area of the latent space that will provide information about the type of arrhythmias the patient is suffering from. First, the parameters that best fit the state of the disease will be known according to the group into which the patient's intracardiac activity data fall. Second, according to the proximity to other groups of arrhythmias trained with different parameters, the most probable evolution of the disease will be known, providing medical specialists with an insight of how the disease could evolve if action is not taken prematurely. As an example, in the right part of Figure 3.18 it is shown a projection of a randomly selected patient. In this case, it is more likely that the average time between arrhythmias of this patient, β , will occur at least every 30 days, but due to its proximity to the lower-left group ($\beta = 180$), it can be understood that its evolution is on the way to reach 30, possibly a value between 180 and 30. The most critical parameter, α , corresponds to a value of 0.998, which means that the evolution is closer to a permanent arrhythmia. This is not the most critical case, but it may need medical intervention in order to prevent future complications.

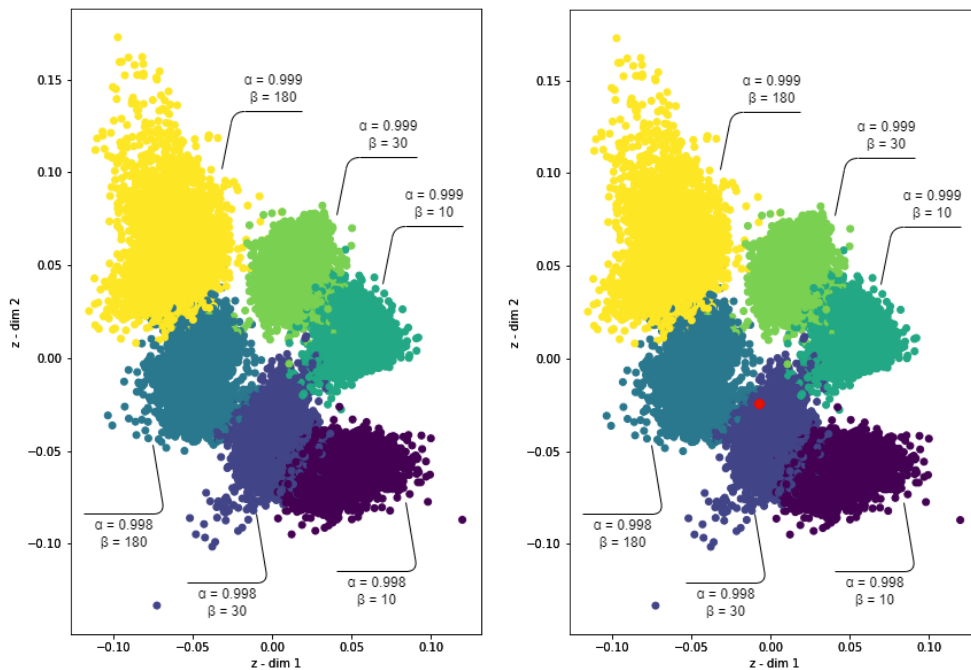


Figure 3.18.- Learned representation of simulated AF events (left). Latent projection of a real patient (right).

For the case of aircraft engines, the workflow is the same: the diagnostic tool is built with the activations of the latent space showing the actual state of the engine and also the rate of change from healthy to deteriorated. Each point in the map represents the status of an engine associated with a window of events during its flight history so that points of degraded aircraft are grouped in nearby areas and, on the contrary, points belonging to healthy aircraft are located in more distant areas. As the actual health status of the training aircraft will be known, since the CMAPSS simulation system was used, a color scale can be established to clearly differentiate healthy aircraft from deteriorated or totally deteriorated aircraft, coloring each point according to its corresponding RUL.

An example of one of the maps produced by this algorithm is shown in Figure 3.19, left side. Aircraft with high RUL values are painted in yellow while aircraft with low RUL values are painted in dark purple. It can be observed that there is a clear progression in the colors along the map since events with no or low deterioration are located in the upper part of the map (high RUL values) while the most deteriorated ones are located in the lower part (low RUL values). The red trail corresponds to the evolution of the health status of a simulated engine. The compressed representation of the first thirty cycles corresponds to the first upper left red dot, while the compressed representation of the last thirty would be the last lower right red dot. Its latent representation begins in the upper left zone and, as it starts to degrade, this location moves to the right until the registers of the last cycles are located in the lower rightmost area, indicating that the engine is totally degraded (low RUL).

This map is considered explainable, since the method's decisions are based entirely on the learned representations and can therefore be justified; and interpretable, because a simple glance at the map gives insight into the status of each engine unit. Other Deep Learning methods can also reveal interpretable information in intermediate layers; however, extra processing is needed in order to find the most suitable layers or to transform the content of these layers into human readable information. An example of this is the embedding projector of tensorflow [96], which applies different dimensionality reduction methods such as UMAP, T-SNE or PCA to provide a visualization of an embedding layer. In contrast to this, this method provides a direct 2-D compression, which does not need any further processing.

Finally, another example is shown in Figure 3.19, right side, for a float of real engines. Six airplanes have been chosen to project their state into the latent space in two different time steps: $t = 0$, which corresponds to feeding the network with the data corresponding to the cycles from 0 to windows length and $t = 1000$, starting from data corresponding to the cycle 1000. The evolution of each engine between the two time periods is marked with an arrow. Fixing the latent projection obtained after training gives some insight into the progression of the health status of these units. The latent projection of engine e1, e2, e3 and e4 during the time steps shown remain over the upper left quadrant, next to other aircraft with similar characteristics: RUL around two hundred cycles, with no signs of near degradation. On the contrary, there is a clear progression in samples e5 and e6, which move clearly downward,

being placed together with engine units close to their end of life (low values of RUL), thus obtaining an accurate and explainable diagnosis beyond a possible label indicating the predicted health. In <https://github.com/NahuelCostaCortez/Remaining-Useful-Life-Estimation-Variational/tree/main/images/gifs> there are some videos available showing the evolution of different engines in the map according to their life cycle history. Also, a demo of the model is available at <https://huggingface.co/spaces/NahuelCosta/RUL-Variational>.

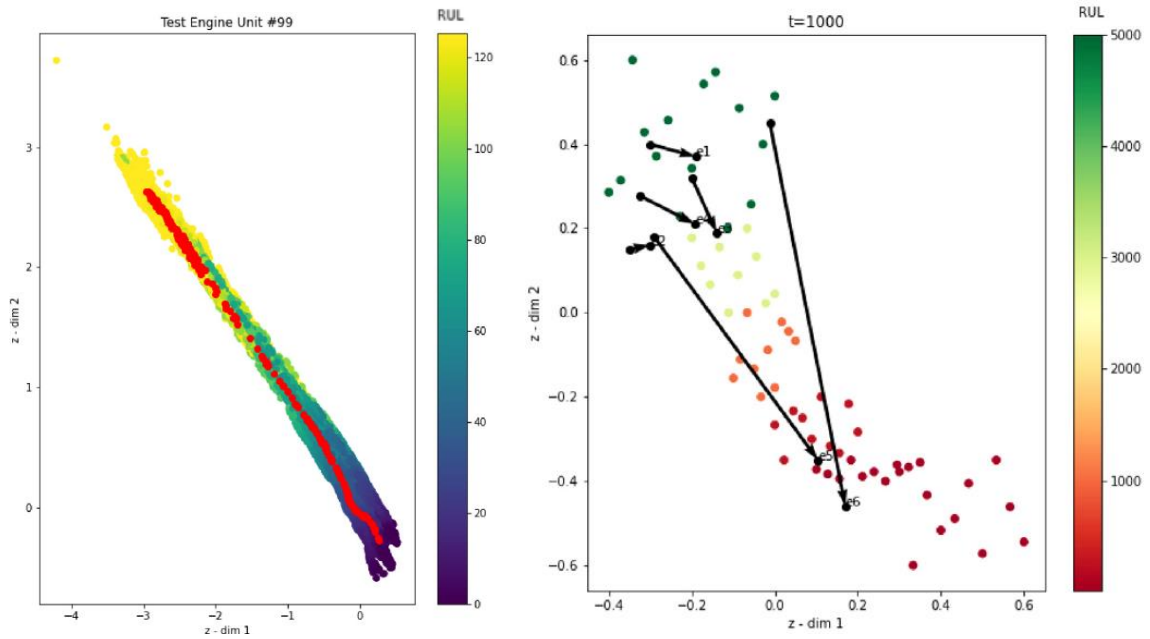


Figure 3.19.- Left: Latent projection of one of the CMAPSS training sets. The red trail corresponds to the RUL evolution of a testing engine. Right: RUL evolution of six real engines in two different time steps, cycle 0 and cycle 1000.

3.4. NUMERICAL RESULTS

After presenting the models and methodologies developed, this section presents a summary of the numerical results achieved, including comparisons with state-of-the-art methods in each topic. The section is organized according to the papers presented in this thesis.

3.4.1. Graphical analysis of the progression of atrial arrhythmia using recurrent neural networks

Costa, N., Fernández Cortés, J., Couso Blanco, I., & Sánchez Ramos, L. (2020). Graphical analysis of the progression of atrial arrhythmia using recurrent neural networks. *International Journal of Computational Intelligence Systems*, 13 (1). DOI: [10.2991/ijcis.d.200926.001](https://doi.org/10.2991/ijcis.d.200926.001); JCR Impact Factor 1.838 (Q2).

Different methods were included in this study for the diagnosis of AF. RNNs were compared together with two standard non-Deep Learning classification methods: Multilayer Perceptron (MLP) and Random Forest. Table 3.4 collects the performance of the different models for each class in terms of accuracy, i.e., each entry in the table is the number of times a series simulated with the parameters indicated in each class was recognized as such. Also, to illustrate the performance of each method, the ranking computed by Friedmans method for each class and the averaged resulting ranking was added.

	Accuracy				
	MLP	Random Forest	GRU	LSTM	GAN Ensemble
998na10	0.9921 (3)	0.9918 (4)	0.9964 (1)	0.9943 (2)	0.9782 (5)
998na30	0.9654 (5)	0.9857 (3)	0.9911 (1)	0.9875 (2)	0.9686 (4)
998na180	0.9371 (5)	0.9800 (3)	0.9879 (1)	0.9946 (2)	0.9596 (4)
999na10	0.9739 (5)	0.9943 (3)	1.0000 (1.5)	1.0000 (1.5)	0.9803 (4)
999na30	0.9368 (5)	0.9979 (3)	0.9996 (1.5)	0.9996 (1.5)	0.9911 (4)
999na180	0.9911 (5)	0.9946 (3)	0.9982 (1)	0.9957 (2)	0.9796 (5)
Summary Results					
Accuracy	0.9661	0.9907	0.9955	0.9953	0.9762
Average rank	4.6666	3.1666	1.1666	1.8333	4.3333

Note: AF, atrial fibrillation; MLP, multilayer perceptron; GRU, gated recurrent unit; LSTM, long short-term memory; GAN, generative adversarial network.

Table 3.4.- Accuracy of the different classifiers, six types of AF.

In all cases RNNs improve the results of MLP and Random Forest. In terms of accuracy, GRU is the RNN that better exploits the incomplete information in truncated AF event series.

It is better than MLP, Random Forest, and GAN with a p-value lower than 0.012, followed by LSTM, although the difference is not statistically significant.

The GAN ensemble approach does not achieve the best numerical results, although in all cases the accuracy is close to one, therefore it is not the most important metric. Instead, the focus is placed on the specificity of the classifiers, as it has a higher impact in the visual coherence of the map. Note that a misclassification may have a great impact, e.g., if a patient whose AF episodes occur every 180 days is assigned 90, 30, or 10 days.

In order to keep the perceptual coherence, the cost of misclassifying arrhythmias must not be uniform. In this respect, Figure 3.20 contains the confusion matrices of the GAN ensemble (left) and Random Forest (right) for the six AF types. Although the average number of correctly classified series is better for Random Forest, this does not imply that the model is better as the specificity is not correct. Mind for instance the pair 998na10-999na10 (15). This means that the model classified wrongly the rate of evolution for the same time between episodes; or the pair 998na30-999na180 (15), the model got wrong the rate of evolution and the initial time between episodes. These are errors that cannot be accepted from a medical diagnosis point of view.

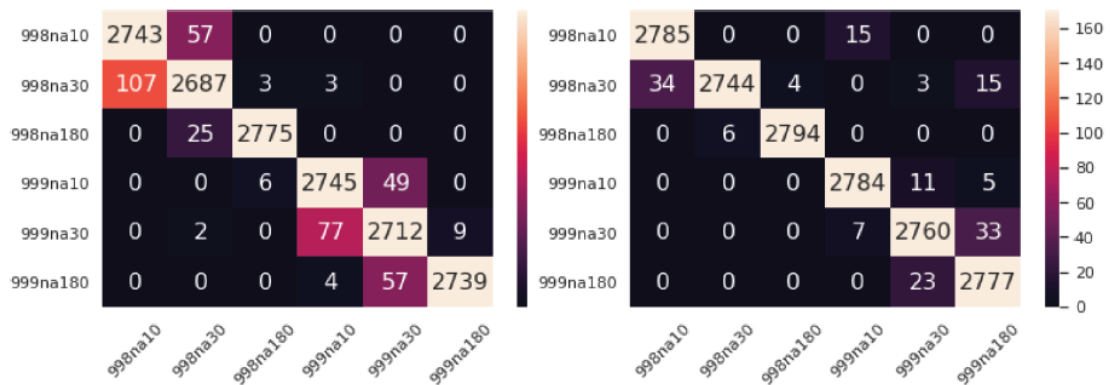


Figure 3.20.- Left: GAN ensemble confusion matrix. Right: Random Forest confusion matrix. Similar classes should be nearby on the map, thus classification errors should be close to the diagonal.

In contrast, the set of GAN discriminators provides an accurate diagnosis, since the classification errors are produced due to the fact that the patient's records have a high similarity with those of the nearby classes and precisely this can be used to build the graphical tool introduced in section 3.3 EXPLAINABILITY.

3.4.2. Semi-supervised recurrent variational autoencoder approach for visual diagnosis of atrial fibrillation

Costa, N., Sanchez, L., & Couso, I. (2021). Semi-supervised recurrent variational autoencoder approach for visual diagnosis of atrial fibrillation. *IEEE Access*, 9, 40227-40239. DOI: 10.1109/ACCESS.2021.3064854; JCR Impact Factor 3.367 (Q1).

In this study, the VAE framework was compared with state-of-the-art classifiers for time series in the AF problem. It is important to note that the simulation model used was updated, so the dataset is not exactly the same as the one used in the paper on the previous subsection.

Table 3.5 shows the performance of the different models for each class in terms of accuracy. It can be quickly seen that the best classifier was Resnet, followed by the proposed solution, labelled as RVAE.

	Accuracy				
	Resnet	FCN	Encoder	TWIESN	RVAE
998na10	0.9681(2)	0.9867 (1)	0.9361(5)	0.9517(4)	0.9543(3)
998na30	0.9664(1)	0.8788(5)	0.9456(3)	0.9438(4)	0.9553(2)
998na180	0.9846(1)	0.9719(4)	0.9729(3)	0.9611(5)	0.9779(2)
999na10	0.9849(1)	0.9849(2)	0.9364(5)	0.9505(4)	0.9770(3)
999na30	0.9879(1)	0.9778(3)	0.9826(2)	0.9791(4)	0.9733(5)
999na180	0.9904(1)	0.9886(4)	0.9895(3)	0.9786(5)	0.9895(2)
Summary Results					
Accuracy	0.9803	0.9647	0.9603	0.9607	0.9712
Average rank	1.166	3.166	3.500	4.333	2.833

Table 3.5.- Accuracy of the different classifiers, six types of AF.

Post-hoc tests were carried out to detect significant differences in pairs between all the classifiers. Table 3.6 shows the family of hypotheses formulated to compare the classifiers ordered by the corresponding p-values. If the significance test yields a p-value lower than a predefined threshold (usually 0.05), then the difference is considered significant, therefore one model is declared superior to another. In this case only Resnet was significantly higher than the other models, which are FCN, Encoder and TWIESN if a significance level of 0.05 is considered since the p-values are below this threshold. The only solution to which it does not significantly exceed is ours. If the Bonferroni correction is considered, in which the number of comparisons is taken into account, the threshold which would have to be set is 0.05 divided by the number of comparisons, i.e., $0.05/6 = 0.0083$. Taking this value, Resnet would only be significantly higher than TWIESN. This is important to note because only

TWIESN and the proposed solution use RNNs, so it can be stated that RVAE outperforms the best state of the art RNN classifier.

i	hypothesis	$z = (R_0 - R_i)/SE$	p
1	Resnet vs TWIESN	3.465	0.0005
2	Resnet vs Encoder	2.556	0.0106
3	Resnet vs FCN	2.191	0.0285
4	Resnet vs RVAE	1.826	0.0679
5	RVAE vs TWIESN	1.640	0.1010
6	FCN vs TWIESN	1.275	0.2023
7	Encoder vs TWIESN	0.912	0.3618
8	RVAE vs Encoder	0.731	0.4648
9	FCN vs Encoder	0.366	0.7144
10	FCN vs RVAE	0.365	0.7151

Table 3.6.- Family of hypotheses ordered by p-value.

As a conclusion of this comparative study, it can be stated that the proposed framework can compete with the best time series classifiers. Besides, the misclassification errors of the model correspond to arrhythmias that are organized in the latent space between classes similar to the one that really belongs, remember Figure 3.18, right side: the classifier learns from that representation, so it can be assumed that failures are most likely due to the overlap of instances of a similar nature, which can also be interpreted as an estimate of the class of arrhythmia that most resembles its parameters or even as the possible future evolution that they will have.

3.4.3. Variational encoding approach for interpretable assessment of remaining useful life estimation

Costa, N., & Sánchez, L. (2022). Variational encoding approach for interpretable assessment of remaining useful life estimation. *Reliability Engineering & System Safety*, 222, 108353. DOI: 10.1016/j.ress.2022.108353; JCR Impact Factor 6.188 (Q1).

The modification introduced in the recurrent VAE (explained in section Variational Autoencoder (VAE), where the decoder was discarded in the training process was tested on the aircraft engines problem.

The comparison results of the proposed framework with other popular approaches on the test sets are listed in Table 3.7 where the selected metrics of all methods, included the proposed

approach, labeled as RVE (Recurrent Variational Encoder), are listed for every dataset. Results in which RVE outperforms the others are highlighted in bold. It can be quickly noted that with datasets FD001 and FD003, although the metrics are considered good, they are not the best. However, the interest lies mostly in FD002 and FD004 as the increasing number of operating conditions and failure modes make these two datasets contain more complicated multiscale degradation features. RVE significantly improves prediction accuracy in these two for both metrics, due to its good feature extraction capability in the face of these complex fault prediction problems.

The comparison also includes a row labeled "VAE+RNN", which corresponds to the same approach including the decoder. This is just to highlight the superiority of this modification in the architecture. Although both use variational inference, the numerical differences are explained by the different latent spaces obtained: one dispersed and the other one continuous (recall Figure 3.11), allowing the latter to improve the predictive capabilities of the model.

	FD001		FD002		FD003		FD004	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
MLP [17]	37.56	18 000	80.03	7 800 000	37.39	17 400	77.37	5 620 000
SVR [17]	20.96	1380	42.00	590 000	21.05	1600	45.35	371 000
RVR [17]	23.80	1500	31.30	17 400	22.37	1430	34.34	26 500
CNN [17]	18.45	1299	30.29	13 600	19.82	1600	29.16	7890
Deep LSTM [17]	16.14	338	24.49	4450	16.18	852	28.17	5550
Semi-supervised [23]	12.56	231	22.73	3366	12.10	251	22.66	2840
DCNN [55]	12.61	273.7	22.36	10 412	12.64	284.1	23.31	12466
MS-DCNN [55]	11.44	196.22	19.35	3747	11.67	241.89	22.22	4844
VAE+RNN	15.81	326	24.12	4183	14.88	722	26.54	5634
RVE	13.42	323.82	14.92	1379.17	12.51	256.36	16.37	1845.99

Table 3.7.- Evaluation metrics of different approaches for RUL estimation on C-MAPSS datasets.

In conclusion, it is demonstrated that, besides providing a visual assessment of the rate of degradation in aircraft engines, the proposed method can also accurately estimate the RUL, outperforming current state-of-the-art methods on the popular C-MAPSS dataset.

3.4.4. Li-ion battery degradation modes diagnosis via Convolutional Neural Networks

Costa, N., Sanchez, L., Anseán, D., & Dubarry, M. (2022). Li-ion battery degradation modes diagnosis via Convolutional Neural Networks. *Journal of Energy Storage*, 55, 105558. DOI: 10.1016/j.est.2022.105558; JCR Impact Factor 8.907 (Q1).

In this study, the DTW approach was used to train a CNN to accurately estimate the degradation modes in Li-ion batteries. The experimental validation was compared to state-of-the-art methods using batteries with different cell configurations on three different chemistries: LFP, NCA and NMC.

Results for degradation mode quantification for all methods are shown in Table 3.8 and Table 3.9 for the LFP cells. It should be noted that only the proposed approach uses DTW images while the other approaches use the raw IC curves. Table 3.8 lists the diagnosis accuracy (by the means of RMSE values) for the quantification of the three degradation modes at six different cycles (10, 50, 100, 200, 400 and 1000) for three different LFP cell configurations. The best predictions are highlighted in bold.

The approach presented in this work, labeled as DTW-CNN, clearly outperforms the others with an average error of 2.00% (see Table 3.9). Yet, there are certain cycles where other methods perform slightly better. This may be due to some bias during training that may lead to unbalanced predictions and, consequently, to reasonable performance in one degradation mode but not in the others. For instance, the predictions of "1DConv" for cycle 400 in C1. Numerically in LAM_{PE} it has a better result than the proposed approach (3.38% vs 3.59%), however, for LLI (1.68% vs 1.31%) and especially for LAM_{NE} (2.83% vs 1.93%) the performance is considerably worse. This is quickly identified in the standard deviation, where DTW-CNN, with a value of 1.96, shows a lower dispersion compared to the other models.

The results for the NMC cells are similar with an average error of 2.03%, compared to errors from 2.56 to 7.27% for the other methods. The approach seems to perform better for NCA cells with an average error of 1.11%, compared to errors from 1.31 to 7.01%.

The main reason behind the consistent estimations in our approach is that the representation of degradations in the images is largely preserved between different cell configurations, something that is not the case in pure IC curve processing, where peaks, despite having similar morphologies, suffer from shifts that can cause models to misleading predictions.

		LLI						LAM _{PE}						LAM _{NE}											
		C1	C2	C3	10	50	100	200	400	1000	10	50	100	200	400	1000	10	50	100	200	400	1000			
FNN [53]	C1	1.89	1.93	2.00	1.82	1.67	3.91	2.53	2.90	3.09	3.28	3.58	11.11	2.30	2.32	2.32	2.10	2.15	6.31						
	C2	2.06	2.16	2.23	1.81	1.55	3.67	3.30	2.94	2.94	2.73	3.46	11.32	3.41	3.29	3.28	2.77	2.35	6.19						
	C3	1.45	1.93	1.88	1.68	1.73	4.02	2.27	3.14	3.40	3.52	3.78	11.31	3.04	3.06	2.98	2.64	2.44	6.37						
RF [52]	C1	6.32	5.69	4.94	3.62	3.23	9.21	5.89	5.13	4.26	3.15	5.16	9.13	7.00	6.06	5.02	3.82	6.24	11.83						
	C2	6.32	5.69	4.94	3.64	3.14	9.22	5.89	5.13	4.26	3.17	4.97	9.79	7.00	6.06	5.02	3.82	6.38	11.55						
	C3	6.32	5.69	4.94	3.62	3.20	9.13	5.89	5.13	4.26	3.15	5.07	9.37	7.00	6.06	5.02	3.82	6.18	11.66						
1DConv [40]	C1	1.18	0.95	0.73	1.06	1.68	3.21	1.90	1.23	1.80	2.80	3.38	10.73	1.18	1.33	1.27	1.71	2.83	6.60						
	C2	0.63	0.59	0.86	1.11	1.62	3.15	0.41	1.28	2.76	2.62	3.50	10.85	2.05	1.83	2.03	2.36	2.86	6.58						
	C3	1.95	0.89	0.60	0.96	1.75	3.35	2.08	1.15	2.01	2.95	3.44	10.86	2.86	1.97	1.59	2.07	2.93	6.61						
DTW-CNN	C1	0.14	0.53	0.72	1.16	1.31	2.47	0.96	0.98	1.82	2.67	3.59	8.64	0.17	0.70	1.40	1.98	1.93	3.86						
	C2	0.44	0.84	0.91	1.18	1.32	2.15	0.78	2.06	2.76	3.22	3.92	8.89	0.21	0.57	0.80	1.11	1.41	4.01						
	C3	0.80	0.56	0.56	0.95	1.12	2.58	2.30	1.32	2.03	2.72	3.67	8.63	0.59	0.55	1.00	1.43	1.64	3.94						

Table 3.8.- RMSE results for each degradation mode and cycle for the LFP test set.

	FNN	RF	1DConv	DTW-CNN
Mean \pm std	3.32 \pm 2.21	5.87 \pm 2.23	2.64 \pm 2.42	2.00 \pm 1.96

Table 3.9.- RMSE results summary for the LFP test set calculated as the average and the standard deviation of predictions in all cycles for all cells.

In conclusion, the performance of the method was shown to be superior to state-of-the-art methods for degradation modes quantification, with RMSPE errors around 2% in average for 1000 duty cycles compared to between 2.64 to 7.27% for the other tested methods. The successful performance of the model is largely due to its adaptive nature to different cell configurations. Also, the model was tested on real cells, where the diagnosis corresponded to a large extent with previously existing studies on the same cells. This opens up new opportunities for collaboration between AI and battery research.

Chapter 4

Conclusions

In this thesis, several solutions to condition monitoring problems have been presented. In particular, the applicability of various processing methods to deal with inaccuracies in monitoring data have been studied and different Deep Learning approaches have been developed to provide an explainable diagnosis and prognosis of the systems to be monitored.

The techniques presented have been validated in 3 fields of study: atrial fibrillation, aircraft engines and lithium-ion batteries. Since there is a lack of data in these fields, simulation models have been used or created to generate the necessary training data for the algorithms developed. Furthermore, different techniques have been presented to reduce the gap between simulation models and real data and precisely all models have been tested using real monitoring data.

In light of the proposed objectives, solutions strongly oriented to the explainability of the models and the interpretability of the data have been developed. Generative models such as GANs and VAE were chosen and adapted to sequential data given that their latent properties allow an excellent internal organization of the data, which can be later used for classification, regression and visualization tasks. In this sense, visual tools have been proposed to understand the decisions of the models and the interpretability of the results, which can be extremely useful for subsequent decision-making.

The aim of the solutions proposed was not only to diagnose the systems to be monitored but also to reflect their future evolution in order to anticipate potential problems. This includes the identification of the rate of change between normal and anomalous operation, which can easily be associated with problems that do not involve a direct disturbance in the system to be monitored but can severely affect it in the long term.

On the other hand, the models developed have proven to be competitive and even superior to the state-of-the-art in the problems to which they have been applied.

Finally, all the code, both for the tools developed and for the reproducibility of results, as well as different demos associated with them have been made public. This will facilitate the dissemination of results and promote the use of these techniques to other fields in which they can easily be adapted.

4.1. FUTURE WORK

Although the work carried out during the completion of this thesis has successfully contributed to the proposed goals, there are still some limitations that can be further explored.

To begin with, the recurrent models used in these studies are LSTM networks or slight related modifications. This was also the way to go in NLP problems until the appearance of Transformers in 2017 [97], a new architecture based on attention. Attention is a mechanism first introduced in [98] to represent the most relevant information in a vector.

The advantages that transformers have over RNNs are the following:

1. They have more memory, therefore they do not suffer from short-term memory.
2. They can be parallelized because data is not processed sequentially.

Since its publication, advances in NLP have been and still are oriented almost exclusively to Transformers and they are also beginning to stand out in Computer Vision [99]. However, although it is also ideal for time series, research it is still limited, with some recent work tailored to multi-step forecasting [100], [101]. This may be due to the use of pre-trained models such as BERT [102]. Transformers have an encoder-decoder architecture, but it is the encoder that learns the underlying representations from which other tasks can be done. Under this premise, BERT is a Transformer encoder pre-train with data from many different text sources and it can be leveraged to fine-tune it for various problems such as sentiment analysis, text prediction or translation tasks.

In NLP or Computer Vision it is feasible to offer a generic model from which to apply transfer learning to a particular problem because text and images have the same nature regardless of its origin, but it is not the same for condition monitoring data. The physical composition of each problem can vary completely and that makes it difficult to pre-train a model with data from several different sources.

However, this assumption is not absolute, and some works have applied transfer learning for time series [103]. Precisely, this is a potential line of research. Being able to develop a pre-trained framework from which the knowledge generated can be leveraged for almost any problem, would greatly facilitate the application of different novel techniques in monitoring data research.

On the other hand, after hundreds of publications claiming that "attention is all we need" (referred to Transformers), new ideas are emerging that question this assertion. The attention layer, although better than RNNs is still a bottleneck for network efficiency. That is why Google researchers replaced BERT's attention layers with Fourier transforms and report reaching 92% accuracy of BERT but 7 times faster [104]. Also, in [105] the authors propose

an MLP-based architecture that improves Transformers in some NLP and vision tasks. It is clear that Deep Learning research is following an exponential pace and these recent advances can mean significant breakthroughs for condition monitoring as well.

There is another trend that is achieving significant results in Computer Vision, known as Contrastive Learning [106]. The main idea of Contrastive Learning is to learn, in an unsupervised way, representations so that in the latent space similar samples stay close to each other, while different samples are distant, something quite similar to what it was achieved in the works presented in this thesis with VAEs. The main motivation for this mode of learning comes from the learning patterns humans have. We recognize objects without remembering all the small details, for instance, we find it easy to look at a picture and find a chair in it based on color, shape and some other features. Roughly speaking, some kind of representation is created in our mind and then we use it to recognize new objects. It would be of interest to apply this learning pattern to condition monitoring as well to provide a better understanding of the data or to use it in problems where labels are missing.

Finally, a hot topic in Deep Learning lately is diffusion models, which are behind major breakthroughs in image generation from text descriptions such as Dall-e 2 [107], Stable diffusion [81] or Imagen [108]. Similar to Denoising Autoencoders, diffusion models work by repeatedly adding Gaussian noise to training data, and then learning how to get the data back by reversing this process. They offer substantial sample diversity and accurate mode coverage of the learned data distribution, which means that are suitable for learning models with lots of different and complicated data. This is barely explored in condition monitoring problems, with only a few recent contributions [109], [110], and could be a major breakthrough for the development of the field.

Chapter 5

Publications

This section includes the research work published during the thesis as well as another publication that is still under review.

4.2. JOURNAL PUBLICATIONS

1. Costa, N., Fernández Cortés, J., Couso Blanco, I., & Sánchez Ramos, L. (2020). Graphical analysis of the progression of atrial arrhythmia using recurrent neural networks. *International Journal of Computational Intelligence Systems*, 13 (1). DOI: [10.2991/ijcis.d.200926.001](https://doi.org/10.2991/ijcis.d.200926.001); JCR Impact Factor 1.838 (Q2).
 2. Costa, N., Sanchez, L., & Couso, I. (2021). Semi-supervised recurrent variational autoencoder approach for visual diagnosis of atrial fibrillation. *IEEE Access*, 9, 40227-40239. DOI: [10.1109/ACCESS.2021.3064854](https://doi.org/10.1109/ACCESS.2021.3064854); JCR Impact Factor 3.367 (Q1).
 3. Costa, N., & Sánchez, L. (2022). Variational encoding approach for interpretable assessment of remaining useful life estimation. *Reliability Engineering & System Safety*, 222, 108353. DOI: [10.1016/j.ress.2022.108353](https://doi.org/10.1016/j.ress.2022.108353); JCR Impact Factor 6.188 (Q1).
 4. Costa, N., & Sánchez, L. (2022). RUL-RVE: Interpretable assesment of Remaining Useful Life. *Software Impacts*, 100321. DOI: [10.1016/j.simpa.2022.100321](https://doi.org/10.1016/j.simpa.2022.100321); JCR Not indexed.
 5. Costa, N., Sanchez, L., Anseán, D., & Dubarry, M. (2022). Li-ion battery degradation modes diagnosis via Convolutional Neural Networks. *Journal of Energy Storage*, 55, 105558. DOI: [10.1016/j.est.2022.105558](https://doi.org/10.1016/j.est.2022.105558); JCR Impact Factor 8.907 (Q1).
-

4.3. CONFERENCE PROCEEDINGS

1. N. Costa, N., Fernández, J., Couso Blanco, I., & Sánchez Ramos, L. (2019). Graphical analysis of the progression of atrial arrhythmia through an ensemble of Generative Adversarial Network Discriminators. In *Proceedings of the 11th Conference of the European Society for Fuzzy Logic and Technology, EUSFLAT*. Atlantis Press.
2. Costa, N., & Sánchez, L. (2021, September). Remaining useful life estimation using a recurrent variational autoencoder. In *International Conference on Hybrid Artificial Intelligence Systems* (pp. 53-64). Springer, Cham.
3. Sánchez, L., Costa, N., Anseán, D., & Couso, I. (2022). Informed Weak Supervision for Battery Deterioration Level Labeling. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (pp. 748-760). Springer, Cham.

4.4. UNDER REVIEW

1. Costa, N., & Sánchez, L. Remaining Useful Life Estimation Using a Recurrent Variational Autoencoder. *Logic Journal of the IGPL*. JCR Impact Factor 0.931 (Q1).
-

REFERENCES

- [1] Vanraj, D. Goyal, A. Saini, S. S. Dhama, and B. S. Pabla, "Intelligent predictive maintenance of dynamic systems using condition monitoring and signal processing techniques — A review," in *2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring)*, 2016, pp. 1–6, doi: 10.1109/ICACCA.2016.7578870.
 - [2] Y. Kubota, H. Nakamoto, S. Egawa, and T. Kawamata, "Continuous EEG monitoring in ICU," *J. Intensive Care*, vol. 6, no. 1, p. 39, Dec. 2018, doi: 10.1186/s40560-018-0310-z.
 - [3] W. Jiang, "Applications of deep learning in stock market prediction: Recent progress," *Expert Syst. Appl.*, vol. 184, p. 115537, Dec. 2021, doi: 10.1016/j.eswa.2021.115537.
 - [4] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, p. 9, Dec. 2016, doi: 10.1186/s40537-016-0043-6.
 - [5] A. Barredo Arrieta *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.
 - [6] World Health Organization, "WHO - The top 10 causes of death," *24 Maggio*, 2018. [Online]. Available: <http://www.who.int/en/news-room/fact-sheets/detail/the-top-10-causes-of-death>.
 - [7] B. A. Schoonderwoerd, M. D. Smit, L. Pen, and I. C. Van Gelder, "New risk factors for atrial fibrillation: causes of 'not-so-lone atrial fibrillation,'" *Europace*, vol. 10, no. 6, pp. 668–673, Apr. 2008, doi: 10.1093/europace/eun124.
 - [8] C. D. Swerdlow, G. Kalahasty, and K. A. Ellenbogen, "Implantable Cardiac Defibrillator Lead Failure and Management," *J. Am. Coll. Cardiol.*, vol. 67, no. 11, pp. 1358–1368, Mar. 2016, doi: 10.1016/j.jacc.2015.12.067.
 - [9] J. Fernández, J. Velasco, and L. Sánchez, "Detection of Cardiac Arrhythmias Through Singular Spectrum Analysis of a Time-Distorted EGM Signal," 2018, pp. 137–146.
 - [10] Y. Wang, Y. Zhao, and S. Addepalli, "Remaining Useful Life Prediction using Deep Learning Approaches: A Review," *Procedia Manuf.*, vol. 49, pp. 81–88, 2020, doi: 10.1016/j.promfg.2020.06.015.
 - [11] A. Nanduri and L. Sherry, "Anomaly detection in aircraft data using Recurrent Neural Networks (RNN)," in *2016 Integrated Communications Navigation and Surveillance (ICNS)*, 2016, pp. 5C2-1-5C2-8, doi: 10.1109/ICNSURV.2016.7486356.
-

- [12] L. Liu, D. Liu, Y. Zhang, and Y. Peng, "Effective Sensor Selection and Data Anomaly Detection for Condition Monitoring of Aircraft Engines," *Sensors*, vol. 16, no. 5, p. 623, Apr. 2016, doi: 10.3390/s16050623.
- [13] M. R. Palacín, "Understanding ageing in Li-ion batteries: a chemical issue," *Chem. Soc. Rev.*, vol. 47, no. 13, pp. 4924–4933, 2018, doi: 10.1039/C7CS00889A.
- [14] X. Lin, K. Khosravinia, X. Hu, J. Li, and W. Lu, "Lithium Plating Mechanism, Detection, and Mitigation in Lithium-Ion Batteries," *Prog. Energy Combust. Sci.*, vol. 87, p. 100953, Nov. 2021, doi: 10.1016/j.pecs.2021.100953.
- [15] M. Dubarry, G. Baure, and D. Anseán, "Perspective on State-of-Health Determination in Lithium-Ion Batteries," *J. Electrochem. Energy Convers. Storage*, vol. 17, no. 4, Nov. 2020, doi: 10.1115/1.4045008.
- [16] T. Lombardo *et al.*, "Artificial Intelligence Applied to Battery Research: Hype or Reality?," *Chem. Rev.*, vol. 122, no. 12, pp. 10899–10969, Jun. 2022, doi: 10.1021/acs.chemrev.1c00108.
- [17] P. M. Attia *et al.*, "Review—'Knees' in Lithium-Ion Battery Aging Trajectories," *J. Electrochem. Soc.*, vol. 169, no. 6, p. 060517, 2022, doi: 10.1149/1945-7111/ac6d13.
- [18] G. dos Reis, C. Strange, M. Yadav, and S. Li, "Lithium-ion battery data and where to find it," *Energy AI*, vol. 5, p. 100081, Sep. 2021, doi: 10.1016/j.egyai.2021.100081.
- [19] ISO, "Condition monitoring and diagnostics of machines — Data processing, communication and presentation. Part 1: General guidelines," *Int. Organ.*, vol. 3, p. 8, 2003.
- [20] K. Tidiri, N. Chatti, S. Verron, and T. Tiplica, "Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges," *Annu. Rev. Control*, vol. 42, pp. 63–81, 2016, doi: 10.1016/j.arcontrol.2016.09.008.
- [21] K. L. Tsui, N. Chen, Q. Zhou, Y. Hai, and W. Wang, "Prognostics and Health Management: A Review on Data Driven Approaches," *Math. Probl. Eng.*, vol. 2015, pp. 1–17, 2015, doi: 10.1155/2015/793161.
- [22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [23] "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," pp. 1–38, 2014.
- [24] C. Poultney, S. Chopra, Y. L. Cun, and others, "Efficient learning of sparse representations with an energy-based model," *Adv. Neural Inf. Process. Syst.*, pp. 1137–1144, 2006, doi: 14:1771-1800.
-

- [25] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," *Proc. 28th Int. Conf. Mach. Learn. ICML 2011*, pp. 833–840, 2011.
- [26] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," Dec. 2013.
- [27] Z.-K. Zhang, C. Liu, Y.-C. Zhang, and T. Zhou, "Handwritten Digit Recognition with a Back-Propagation Network," *Adv. Neural Inf. Process. Syst.*, pp. 396–404, 1990.
- [28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
- [29] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 2015, pp. 234–241.
- [30] D. . Pham and D. Karaboga, "Training Elman and Jordan networks for system identification using genetic algorithms," *Artif. Intell. Eng.*, vol. 13, no. 2, pp. 107–117, Apr. 1999, doi: 10.1016/S0954-1810(98)00013-2.
- [31] A. Graves, G. Wayne, and I. Danihelka, "Neural Turing Machines," Oct. 2014, doi: 1410.5401.
- [32] S. Hochreiter and J. Schmidhuber, "Long Short Term Memory," *Neural Comput.* 9, 1997.
- [33] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 2014.
- [34] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Networks*, vol. 5(2), 1994.
- [35] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997, doi: 10.1109/78.650093.
- [36] Alle, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.
- [37] C. Wu, P. Jiang, C. Ding, F. Feng, and T. Chen, "Intelligent fault diagnosis of rotating machinery based on one-dimensional convolutional neural network," *Comput. Ind.*, vol. 108, pp. 53–61, Jun. 2019, doi: 10.1016/j.compind.2018.12.001.
- [38] M. Sakurada and T. Yairi, "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction," in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis - MLSDA'14*, 2014, pp. 4–11, doi: 10.1145/2689746.2689747.
- [39] C. Zhou and R. C. Paffenroth, "Anomaly Detection with Robust Deep Autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 665–674, doi: 10.1145/3097983.3098052.
-

- [40] X. Guo, X. Liu, E. Zhu, and J. Yin, "Deep Clustering with Convolutional Autoencoders," 2017, pp. 373–382.
- [41] C. Li, R.-V. Sánchez, G. Zurita, M. Cerrada, and D. Cabrera, "Fault Diagnosis for Rotating Machinery Using Vibration Measurement Deep Statistical Feature Learning," *Sensors*, vol. 16, no. 6, p. 895, Jun. 2016, doi: 10.3390/s16060895.
- [42] J. Li, X. Li, D. He, and Y. Qu, "A Novel Method for Early Gear Pitting Fault Diagnosis Using Stacked SAE and GBRBM," *Sensors*, vol. 19, no. 4, p. 758, Feb. 2019, doi: 10.3390/s19040758.
- [43] J. Ma, H. Su, W. Zhao, and B. Liu, "Predicting the Remaining Useful Life of an Aircraft Engine Using a Stacked Sparse Autoencoder with Multilayer Self-Learning," *Complexity*, vol. 2018, pp. 1–13, Jul. 2018, doi: 10.1155/2018/3813029.
- [44] A. Listou Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture," *Reliab. Eng. Syst. Saf.*, vol. 183, pp. 240–251, Mar. 2019, doi: 10.1016/j.ress.2018.11.027.
- [45] L. Zhang, J. Lin, B. Liu, Z. Zhang, X. Yan, and M. Wei, "A Review on Deep Learning Applications in Prognostics and Health Management," *IEEE Access*, vol. 7, pp. 162415–162438, 2019, doi: 10.1109/ACCESS.2019.2950985.
- [46] G. Pang, C. Shen, L. Cao, and A. Van Den Hengel, "Deep Learning for Anomaly Detection," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2022, doi: 10.1145/3439950.
- [47] I. Errandonea, S. Beltrán, and S. Arrizabalaga, "Digital Twin for maintenance: A literature review," *Comput. Ind.*, vol. 123, p. 103316, Dec. 2020, doi: 10.1016/j.compind.2020.103316.
- [48] Y. Liu, D. K. Frederick, J. A. Decastro, J. S. Litt, and W. W. Chan, "User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)," *Nasa/Tm*, vol. 2012–21743, no. March, pp. 1–40, 2012.
- [49] A. Saxena and K. Goebel, "Turbofan Engine Degradation Simulation Data Set," *ACM Ref. Format*, 2017.
- [50] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliab. Eng. Syst. Saf.*, vol. 172, pp. 1–11, Apr. 2018, doi: 10.1016/j.ress.2017.11.021.
- [51] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long Short-Term Memory Network for Remaining Useful Life estimation," in *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2017, pp. 88–95, doi: 10.1109/ICPHM.2017.7998311.
- [52] W. Yu, I. Y. Kim, and C. Mechefske, "An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme," *Reliab. Eng. Syst. Saf.*, vol. 199, p. 106926, Jul. 2020, doi: 10.1016/j.ress.2020.106926.
-

- [53] Z. Shi and A. Chehade, "A dual-LSTM framework combining change point detection and remaining useful life prediction," *Reliab. Eng. Syst. Saf.*, vol. 205, p. 107257, Jan. 2021, doi: 10.1016/j.ress.2020.107257.
- [54] M. Dubarry, C. Truchot, and B. Y. Liaw, "Synthesize battery degradation modes via a diagnostic and prognostic model," *J. Power Sources*, vol. 219, pp. 204–216, Dec. 2012, doi: 10.1016/j.jpowsour.2012.07.016.
- [55] M. Dubarry and D. Beck, "Big data training data for artificial intelligence-based Li-ion diagnosis and prognosis," *J. Power Sources*, vol. 479, p. 228806, Dec. 2020, doi: 10.1016/j.jpowsour.2020.228806.
- [56] J. S. Edge *et al.*, "Lithium ion battery degradation: what you need to know," *Phys. Chem. Chem. Phys.*, vol. 23, no. 14, pp. 8200–8221, 2021, doi: 10.1039/D1CP00359C.
- [57] K. S. Mayilvahanan, K. J. Takeuchi, E. S. Takeuchi, A. C. Marschilok, and A. C. West, "Supervised Learning of Synthetic Big Data for Li-Ion Battery Degradation Diagnosis," *Batter. Supercaps*, vol. 5, no. 1, Jan. 2022, doi: 10.1002/batt.202100166.
- [58] N. Costa, L. Sánchez, D. Anseán, and M. Dubarry, "Li-ion battery degradation modes diagnosis via Convolutional Neural Networks," *J. Energy Storage*, vol. 55, p. 105558, Nov. 2022, doi: 10.1016/j.est.2022.105558.
- [59] H. Miao, B. Li, C. Sun, and J. Liu, "Joint Learning of Degradation Assessment and RUL Prediction for Aeroengines via Dual-Task Deep LSTM Networks," *IEEE Trans. Ind. Informatics*, vol. 15, no. 9, pp. 5023–5032, Sep. 2019, doi: 10.1109/TII.2019.2900295.
- [60] J. Jiang, Y. Gao, C. Zhang, W. Zhang, and Y. Jiang, "Lifetime Rapid Evaluation Method for Lithium-Ion Battery with Li(NiMnCo)O₂ Cathode," *J. Electrochem. Soc.*, vol. 166, no. 6, pp. A1070–A1081, Apr. 2019, doi: 10.1149/2.1051904jes.
- [61] S. Torai, M. Nakagomi, S. Yoshitake, S. Yamaguchi, and N. Oyama, "State-of-health estimation of LiFePO₄/graphite batteries based on a model using differential capacity," *J. Power Sources*, vol. 306, pp. 62–69, Feb. 2016, doi: 10.1016/j.jpowsour.2015.11.070.
- [62] X. Han, M. Ouyang, L. Lu, J. Li, Y. Zheng, and Z. Li, "A comparative study of commercial lithium ion battery cycle life in electrical vehicle: Aging mechanism identification," *J. Power Sources*, vol. 251, pp. 38–54, Apr. 2014, doi: 10.1016/j.jpowsour.2013.11.029.
- [63] S. Lee and Y. Kim, "Li-ion Battery Electrode Health Diagnostics using Machine Learning," in *2020 American Control Conference (ACC)*, 2020, pp. 1137–1142, doi: 10.23919/ACC45564.2020.9147633.
- [64] S. Kim, Z. Yi, B.-R. Chen, T. R. Tanim, and E. J. Dufek, "Rapid failure mode classification and quantification in batteries: A deep learning modeling framework," *Energy Storage Mater.*,
-

- vol. 45, pp. 1002–1011, Mar. 2022, doi: 10.1016/j.ensm.2021.07.016.
- [65] H. Ruan, J. Chen, W. Ai, and B. Wu, “Generalised diagnostic framework for rapid battery degradation quantification with deep learning,” *Energy AI*, vol. 9, p. 100158, Aug. 2022, doi: 10.1016/j.egyai.2022.100158.
- [66] P. Tormene, T. Giorgino, S. Quaglini, and M. Stefanelli, “Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation,” *Artif. Intell. Med.*, vol. 45, no. 1, pp. 11–34, Jan. 2009, doi: 10.1016/j.artmed.2008.11.007.
- [67] T. Górecki and M. Łuczak, “Non-isometric transforms in time series classification using DTW,” *Knowledge-Based Syst.*, vol. 61, pp. 98–108, May 2014, doi: 10.1016/j.knosys.2014.02.011.
- [68] M. Shah, J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, “Learning DTW-Shapelets for Time-Series Classification,” in *Proceedings of the 3rd IKDD Conference on Data Science, 2016*, 2016, pp. 1–8, doi: 10.1145/2888451.2888456.
- [69] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, and E. Keogh, “Generalizing DTW to the multi-dimensional case requires an adaptive approach,” *Data Min. Knowl. Discov.*, vol. 31, no. 1, pp. 1–31, Jan. 2017, doi: 10.1007/s10618-016-0455-0.
- [70] N. Begum, L. Ulanova, J. Wang, and E. Keogh, “Accelerating Dynamic Time Warping Clustering with a Novel Admissible Pruning Strategy,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015*, pp. 49–58, doi: 10.1145/2783258.2783286.
- [71] L. Tao, C. Lu, and A. Noktehdan, “Similarity recognition of online data curves based on dynamic spatial time warping for the estimation of lithium-ion battery capacity,” *J. Power Sources*, vol. 293, pp. 751–759, Oct. 2015, doi: 10.1016/j.jpowsour.2015.05.120.
- [72] Y. Liu, C. Zhang, J. Jiang, Y. Jiang, L. Zhang, and W. Zhang, “Capacity Estimation of Serial Lithium-ion Battery Pack Using Dynamic Time Warping Algorithm,” *IEEE Access*, vol. 7, pp. 174687–174698, 2019, doi: 10.1109/ACCESS.2019.2956326.
- [73] P. Hu, G. Ma, Y. Zhang, C. Cheng, B. Zhou, and Y. Yuan, “State of health estimation for lithium-ion batteries with dynamic time warping and deep kernel learning model,” in *2020 European Control Conference (ECC)*, 2020, pp. 602–607, doi: 10.23919/ECC51009.2020.9143757.
- [74] S. Kim, N. H. Kim, and J.-H. Choi, “Prediction of remaining useful life by data augmentation technique based on dynamic time warping,” *Mech. Syst. Signal Process.*, vol. 136, p. 106486, Feb. 2020, doi: 10.1016/j.ymssp.2019.106486.
- [75] T. Waldmann, B.-I. Hogg, and M. Wohlfahrt-Mehrens, “Li plating as unwanted side reaction in commercial Li-ion cells – A review,” *J. Power Sources*, vol. 384, pp. 107–124, Apr. 2018,
-

doi: 10.1016/j.jpowsour.2018.02.063.




- [76] D. Beck, P. Dechent, M. Junker, D. U. Sauer, and M. Dubarry, "Inhomogeneities and Cell-to-Cell Variations in Lithium-Ion Batteries, a Review," *Energies*, vol. 14, no. 11, p. 3276, Jun. 2021, doi: 10.3390/en14113276.
 - [77] A. Oussidi and A. Elhassouny, "Deep generative models: Survey," in *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, 2018, pp. 1–8, doi: 10.1109/ISACV.2018.8354080.
 - [78] A. Tewari *et al.*, "State of the Art on Neural Rendering," *Comput. Graph. Forum*, vol. 39, no. 2, pp. 701–727, May 2020, doi: 10.1111/cgf.14022.
 - [79] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models," Mar. 2021, doi: 10.1109/TPAMI.2021.3116668.
 - [80] I. Goodfellow *et al.*, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.
 - [81] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," Dec. 2021.
 - [82] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep Unsupervised Learning using Nonequilibrium Thermodynamics," Mar. 2015.
 - [83] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Spec. Lect. IE*, vol. 2, no. 1, pp. 1–18, 2015.
 - [84] J. Sun, X. Wang, N. Xiong, and J. Shao, "Learning Sparse Representation With Variational Auto-Encoder for Anomaly Detection," *IEEE Access*, vol. 6, pp. 33353–33361, 2018, doi: 10.1109/ACCESS.2018.2848210.
 - [85] H. Xu *et al.*, "Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, 2018, pp. 187–196, doi: 10.1145/3178876.3185996.
 - [86] D. Park, Y. Hoshi, and C. C. Kemp, "A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018, doi: 10.1109/LRA.2018.2801475.
 - [87] S. Suh, D. H. Chae, H.-G. Kang, and S. Choi, "Echo-state conditional variational autoencoder for anomaly detection," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 1015–1022, doi: 10.1109/IJCNN.2016.7727309.
 - [88] Y. Jin, J. Zhang, M. Li, Y. Tian, H. Zhu, and Z. Fang, "Towards the Automatic Anime Characters Creation with Generative Adversarial Networks," Aug. 2017.
-

- [89] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, “Generative Visual Manipulation on the Natural Image Manifold,” Sep. 2016.
- [90] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” Mar. 2017.
- [91] H. Zhang *et al.*, “StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks,” Dec. 2016.
- [92] X. Yi, E. Walia, and P. Babyn, “Generative adversarial network in medical imaging: A review,” *Med. Image Anal.*, vol. 58, p. 101552, Dec. 2019, doi: 10.1016/j.media.2019.101552.
- [93] B. Goodman and S. Flaxman, “European Union Regulations on Algorithmic Decision-Making and a ‘Right to Explanation,’” *AI Mag.*, vol. 38, no. 3, pp. 50–57, Oct. 2017, doi: 10.1609/aimag.v38i3.2741.
- [94] D. Castelvechi, “Can we open the black box of AI?,” *Nature*, vol. 538, no. 7623, pp. 20–23, Oct. 2016, doi: 10.1038/538020a.
- [95] Z. Kebaili Bargaoui and A. Chebbi, “Comparison of two kriging interpolation methods applied to spatiotemporal rainfall,” *J. Hydrol.*, vol. 365, no. 1–2, pp. 56–73, Feb. 2009, doi: 10.1016/j.jhydrol.2008.11.025.
- [96] Google Ltd., “Embedding Projector - TensorFlow,” 2021.
- [97] A. Vaswani *et al.*, “Attention Is All You Need,” *アジア経済*, Jun. 2017.
- [98] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” Sep. 2014.
- [99] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” Oct. 2020.
- [100] H. Zhou *et al.*, “Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting,” *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, pp. 11106–11115, May 2021, doi: 10.1609/aaai.v35i12.17325.
- [101] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, “Temporal Fusion Transformers for interpretable multi-horizon time series forecasting,” *Int. J. Forecast.*, vol. 37, no. 4, pp. 1748–1764, Oct. 2021, doi: 10.1016/j.ijforecast.2021.03.012.
- [102] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Oct. 2018, doi: 1810.04805.
- [103] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, “Transfer learning for time series classification,” in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 1367–1376, doi: 10.1109/BigData.2018.8621990.
- [104] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon, “FNet: Mixing Tokens with Fourier
-

- Transforms,” May 2021, doi: 2105.03824.
- [105] H. Liu, Z. Dai, D. R. So, and Q. V. Le, “Pay Attention to MLPs,” May 2021, doi: 2105.08050.
- [106] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, “A Survey on Contrastive Self-Supervised Learning,” *Technologies*, vol. 9, no. 1, p. 2, Dec. 2020, doi: 10.3390/technologies9010002.
- [107] J. Haworth and P. Vincent, “Hierarchical Text-Conditional Image Generation with CLIP Latents,” *Adv. Geogr. Geogr. Learn.*, vol. 6, no. 2, pp. 113–116, 1974.
- [108] C. Saharia *et al.*, “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding,” May 2022.
- [109] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, “Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting,” Jan. 2021.
- [110] Y. Tashiro, J. Song, Y. Song, and S. Ermon, “CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation,” *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 24804–24816, 2021.
-

Research Article

Graphical Analysis of the Progression of Atrial Arrhythmia Using Recurrent Neural Networks

Nahuel Costa^{1, }, Jesús Fernández², Inés Couso^{3, }, Luciano Sánchez^{1,*, }

¹Computer Science Department, University of Oviedo, Gijón, Asturias, Spain

²Medtronic, S.A. Gijón, Asturias, Spain

³Statistics Department, University of Oviedo, Gijón, Asturias, Spain

ARTICLE INFO

Article History

Received 01 May 2020

Accepted 11 Sep 2020

Keywords

Heart disease
 Graphical analysis
 Generative networks
 Recurrent neural networks
 Time series

ABSTRACT

Pacemaker logs are used to predict the progression of paroxysmal cardiac arrhythmia to permanent atrial fibrillation by means of different deep learning algorithms. Recurrent Neural Networks are trained on data produced by a generative model. The activations of the different nets are displayed in a graphical map that helps the specialist to gain insight into the cardiac condition. Particular attention was paid to Generative Adversarial Networks (GANs), whose discriminative elements are suited for detecting highly specific sets of arrhythmias. The performance of the map is validated with simulated data with known properties and tested with intracardiac electrograms obtained from pacemakers and defibrillator systems.

© 2020 The Authors. Published by Atlantis Press B.V.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

1. INTRODUCTION

Atrial fibrillation (AF) is an abnormal heartbeat, common in the elderly, that sometimes progresses from paroxysmal arrhythmia (episodes of arrhythmia that end spontaneously) to persistent arrhythmia (episodes that last more than seven days and do not end without external intervention) or permanent arrhythmia (uninterrupted episodes). It is common for paroxysmal arrhythmia to progress to persistent or permanent arrhythmia [1]. There are numerous risk factors that influence the progress [2], and an early diagnosis is beneficial for optimal treatment.

Surface electrocardiograms (ECGs) are a potential source of information about the evolution of the arrhythmia [3]. Recent advances in connected and pervasive healthcare allow for continuous monitoring of the ECG signal, that is helpful for detecting pathological signatures and arrhythmias [4]. Portable ECG monitors are most helpful with patients in the latter stages of permanent AF [5]. The health risks for patients in the early states of paroxysmal arrhythmia are minor and the drawbacks of carrying this kind of medical equipment at all times outweigh the advantages. This situation may change in the near future, as recent ECG sensors are small enough to be embedded in smartwatches. The Apple Heart Study [6] has shown that different AF types can be detected with wearable sensors, but the battery consumption of ECG sensors is still high and that prevents that the sensor is always on. Detection and timing of short AF episodes remains an open problem.

The treatment of AF often involves the use of pacemakers or Implantable Cardiac Defibrillators (ICDs) [8]. These devices keep a record of the dates and lengths of the episodes and are a source of data that, to the best of our knowledge, has not been used in the past for assessing the evolution of AF. In addition to dates and episode lengths, short intracardiac electrocardiograms (iECGs) spanning a few seconds before and after the detection of each episode are stored in the device memory (see Figure 1). These iECGs are not intended for medical diagnosis, but for adjusting the operational parameters of the ICD. The amount of information that an iECG carries is reduced: the morphology of the heartbeat in iECGs is lost in the high-pass filtering at the ICD electrode and the only relevant information is kept in the instantaneous frequencies of atrium and ventricle.

Given that the shape of the heartbeat is not available in ICD-based iECGs [9], the most reliable source of information is given by the dates and lengths of the recorded episodes. There is an additional problem with this source patients with a long record of episodes will be in the latest stages of AF, when the diagnostic is clear. The challenge is to anticipate the future pace of the AF since the initial episodes. The patients of interest have a short history, that might not be large enough for fitting a nontrivial model (see Figure 2). This is aggravated by the fact that the data is nonstationary and it is precisely the change in the properties of this data (from paroxysmal to permanent) that we want to predict on the basis of a short sample.

There are also technical difficulties [10]. The algorithm that the ICD uses for detecting AF episodes depends on certain parameters

* Corresponding author. Email: luciano@uniovi.es

that are adjusted by the technician on the basis of the iECGs mentioned before. Safety concerns prevail, thus the rate of false positives is high. As a consequence of this, long AF episodes are often reported as clusters of short episodes and a nontrivial preprocessing is needed to remove spurious events. This kind of preprocessing shortens the lists of episodes even more.

Because of the reasons mentioned before, the progression of AF is a complex process that depends on many different factors, but each patient will be associated to only a few tens of pacemaker records. There are not many different techniques for classifying short time series [11] and, according to our own experimentation, none of them is capable of finding a reliable break point between paroxysmal and permanent AF.

The solution that is proposed in this paper consists in a generative map: a generative model produces data that is used to train a topology-preserving map, where the distances between the inputs are correlated with the distances between their projections in the map [12]. The topological map can be derived either from the activation of a single multi-class classifier or from an ensemble of binary classifiers. In the latter case, each of the binary classifiers is only exposed to arrhythmias of a certain type. When this array is fed with ICD records from a real patient, it is expected that only a few of these classifiers will react, meaning that the patient's arrhythmia is of the same type as the arrhythmia with which these classifiers were trained.

Most AI-based systems have a black-box nature that allows powerful predictions, but cannot be explained directly. For this reason explainable AI (XAI) has been gaining increasing attention recently. Layer-wise Relevance Propagation [13] is used as a proposal to understand classification decisions of nonlinear classifiers using heat maps that show the contribution of each pixel in computer vision applications. Class Activation Map (CAM) [14] has been also a popular method to generate saliency maps that highlight the most important regions in the data for making predictions, usually images. This concept has been applied in medical diagnosis [15]. Other methods rely on localization, gradients, and perturbations under the category of sensitivity [16,17]. Our method can be considered as a mixture of the latter and CAM. We project a visualization of the data using the activations of the neurons of the studied methods as a base to build these maps. The location of these activations will be arranged on the map to provide an intuitive visual diagnosis.

AF episodes are sequential data. Recurrent Neural Networks (RNNs) have been used in the literature in recent years for this type of problem and typically architectures such as Long Short-Term Memory (LSTM) [18] or Gated Recurrent Unit (GRU) [19] have proven to be good alternatives. On the other hand, a deep neural net architecture known as Generative Adversarial Network (GAN) [20] is currently breaking into Machine Learning in many fields [21–23]. Nonetheless, its research in the medical field is still limited [24,25], and their application for the diagnosis of cardiovascular diseases has not been explored yet.

Figure 3 presents a summary of the operation mode followed to give a better overall understanding:

1. A generative model(1) is used to simulate real clinical data(2).

2. The generated data is used for training different methods(3) to evaluate intracardiac records. Among these methods, further research is done to obtain a time series classifier based on adversarial training.
3. A self-explanatory graphic map(4) is obtained when the proposed methods are fed with data from real patients with AF.

The structure of this paper is as follows: in Section 2, the generative model of the AF episodes are described. In Section 3, different approaches to solve the problem are presented. Performance of the different methods is discussed in Section 4. Visual representations and assessments are reported in Section 5 while conclusions are drawn in Section 6.

2. MODEL OF THE SEQUENCE OF ICD EVENTS

The purpose of this study is to predict the progression of paroxysmal cardiac arrhythmia to permanent AF on the basis of iECGs and other data collected by ICDs. AF episodes are easily detected in surface electrograms (ECGs) but iECGs are less informative. ECGs are representations of cardiac electrical activity from two electrodes placed on the surface of the body which are located apart from the heart (recall Figure 1, upper part). With this type of derivation, all kinds of electrical activity are recorded, including noncardiac electrical activity. On the contrary, iECGs (Figure 1, lower part) are representations of the potential difference between two points in contact with the myocardium in space over time.

2.1. AMS Events

ICDs do not store a continuous stream of data, but there are certain events that trigger that data is recorded. The primary purpose of an ICD is to release an electrical current between two points to activate the cardiac cells and therefore facilitate cardiac contraction. Depending on the electrical signal that is measured through the leads, the pacemaker will respond in order to stimulate, inhibit, or change its operation mode. In particular, in the presence of cardiac arrhythmia, if a patient experiences a high intrinsic atrial heart rate the pacemaker does not try to match the ventricle to the atrial rate. Instead, the pacemaker changes its operation mode and uses a different algorithm for generating the excitation of the ventricle. This process is called Automatic Mode Switching (AMS) [26]. AMS events are stored in the pacemaker memory and are used to mark the beginning of AF episodes (Figure 2, upper part). The lengths of the AF episodes are stored along with the AMS dates in the pacemaker memory.

Although AMS is a simple concept, the mode switching depends on a large number of variables that depend on the patient. It is possible that the pacemaker algorithm prematurely concludes that the AF event has ended, only to discover past a few seconds that an AF is still taking place. In this case, a second AMS event is generated and the pacemaker mode is restored. This has not relevant consequences for the efficiency of the device, but the stored information is inaccurate, as there may be cases where a cluster of short arrhythmias is reported instead of a long event.

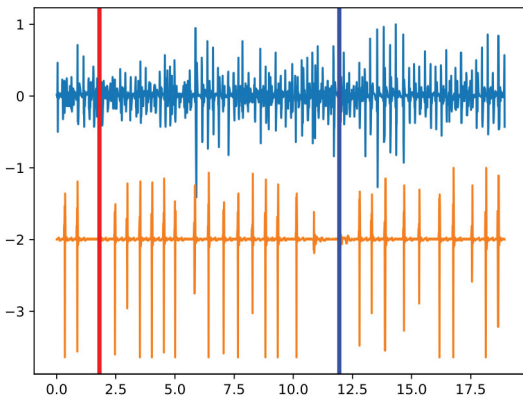
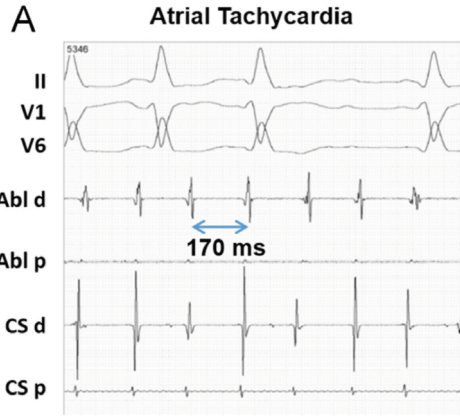


Figure 1 | Top: Surface electrocardiogram (ECG) (taken from Ref. [7]). Bottom: Intracardiac ECG. The morphology of the surface ECG is not kept in the intracardiac ECG (iECG), where there is only one peak for each heartbeat.

2.2. Markov Model

The proposed dynamical model of the operation of an ICD is depicted in Figure 4. There are three states: “Normal,” “Arrhythmia,” and “False Normal.” A patient is in “Normal” state until an AMS event is issued by the ICD and the patient transitions to state “Arrhythmia.” There are two possible paths from this state: back to “Normal” when the episode ends or a transition to “False Normal” when a spurious end of episode is issued. In this second case, the patient remains in the state “False Normal” until a new AMS event is dispatched and then goes back to “Arrhythmia.” AMS events mark either the beginning of a true AF episode or the end of a “False Normal” state. This second class of AMS events are abnormal and should be purged, but there is not a simple procedure to remove them from ICD data [26]. Given that these events will be present in actual patients, the generative model must produce these spurious events as well.

It will be assumed that the dates of the AF episodes conform an inhomogeneous Poisson process. The time between two episodes follows an exponential distribution with parameter $\lambda_{NA}(t)$. The length of an episode also follows an exponential distribution with parameter $\lambda_A(t)$. The progression from paroxysmal to permanent

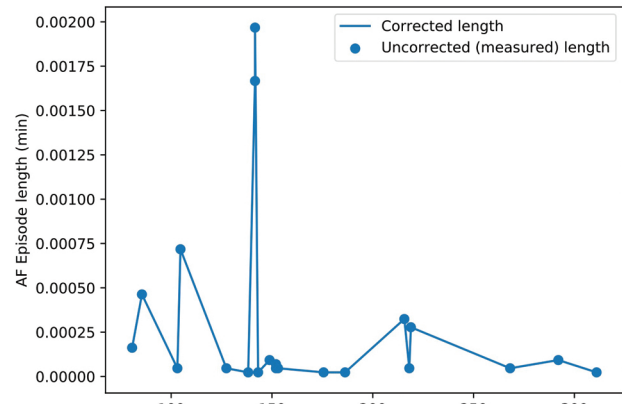
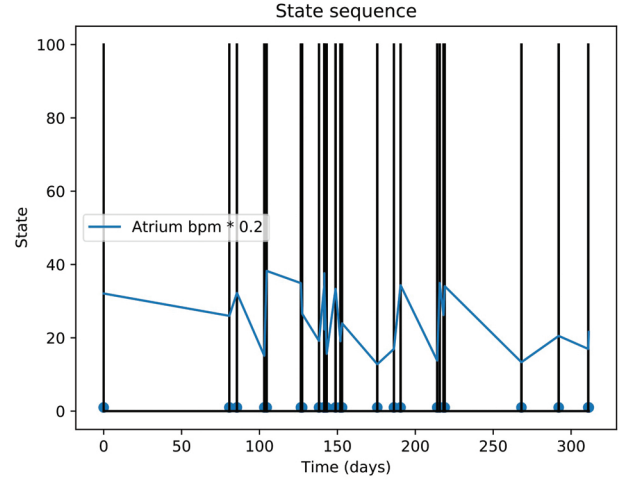


Figure 2 | Top: Dates of pacemaker mode changes during a year. Bottom: Recorded length of the atrial fibrillation (AF) episodes.

AF is measured by the speed of change in these two parameters: as the cardiac condition worsens, the time between episodes is shorter and episodes are longer. The speed of the progression is modelled by a parameter $\alpha \in [0, 1]$,

$$\lambda_{NA}(t) = \lambda_{NA}(0) \cdot \alpha^t, \quad (1)$$

$$\lambda_A(t) = \lambda_A(0) \cdot \alpha^{-t}, \quad (2)$$

where $\alpha = 1$ is a stable patient and values of α lower than 1 are patients with a quick progression to permanent arrhythmia. It will also be supposed that the transition from state “Arrhythmia” to “Normal” can happen with a probability p_{AN} . The probability of the transition from “Arrhythmia” to “False Normal” is therefore $p_{AG} = 1 - p_{AN}$. p_{AG} is the fraction of false positives, which is the probability that the AF detection algorithm in the ICD signals the end of an episode too early.

From a formal point of view, this model is a continuous-time Markov process that is characterized by a tuple of five parameters: $(\lambda_{NA}(0), \lambda_{GA}, \lambda_A(0), p_{AG}, \alpha)$. The generative model that feeds the RNNs described in Section 3 inputs a random seed and produces a list of AMS events by Monte-Carlo simulation. Each of these randomly generated lists can be regarded as an hypothetical patient, whose AF type is defined by the mentioned parameters.

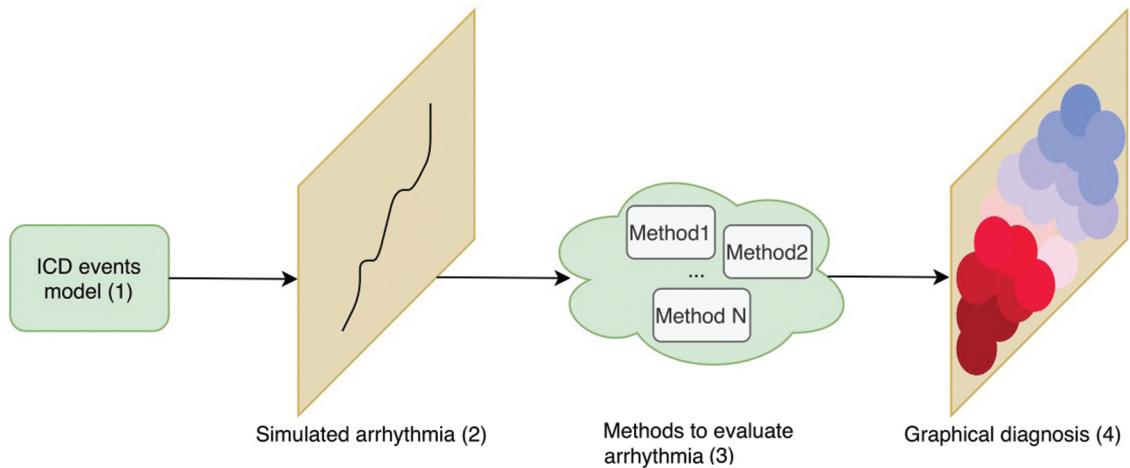


Figure 3 | Pipeline of the presented work.

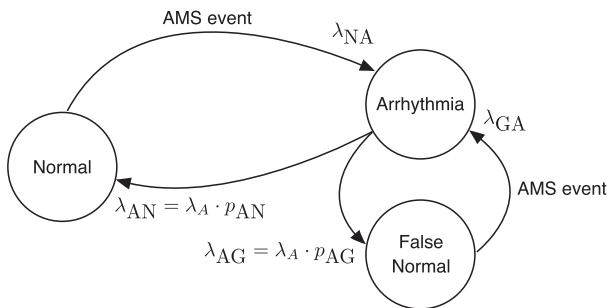


Figure 4 | State diagram of the dynamical model of the beginning of atrial fibrillation (AF) episodes.

3. GENERATIVE MAP

The diagnosis tool that is introduced in this study is a color-coded generative map that displays the actual state of the patient and the speed of change in his/her condition from paroxysmal to permanent AF. When the input is a Monte-Carlo simulation of AMS events, only a small area in the map should become active; ideally just one point. Otherwise, when actual AMS events are used, a potentially larger area could activate because ICD data will not match the output of any model in a perfect way. In other words, the activation area in the map is small when the diagnostic is clear and large when many different diagnostics are compatible with the available data. In this respect, the map can be regarded as a projection of the ICD data in an space whose coordinates are the values of λ_{NA} , λ_{GA} , λ_A , p_{AG} , and α . The values of λ_{NA} , λ_A , and α in the projection measure the condition of the patient and the progression of the AF. λ_{GA} and p_{AG} measure the chance that an AMS event in the ICD is spurious.

3.1. Uncertainty in the Data

Because of the behavior of the ICDs mentioned in the preceding section, spurious AMS events can be produced and it is possible that a long AF episode is perceived as a series of short events. There is not an easy procedure for knowing whether a non-simulated patient is in “Normal” or “False Normal” state.

In this study we will cope with this uncertainty by means of a fuzzy postprocessing that replaces the list of ICD logs by a continuous-time function that can be sampled at regular intervals. This transform consists in computing the degree of truth of the assert “*the patient was undergoing an AF episode at time t* ” [27]. Thus, this function measures the percentage of daily AF events, subsequently becoming a soft window (with Gaussian membership) that extends a few days before and after time t (see Figure 5).

3.2. LSTM and GRU Networks. Error Minimization and GAN Architecture

Networks are sought that are able to estimate the parameters of the Markov model given a truncated sample of postprocessed ICD events. RNNs are arguably the technique of choice for this application [28]. Let us remark that the difficulty of the problem at hand is learning from short time series, i.e., from incomplete information. The shorter the sample is, the more probable is that different models can produce the same sample.

Accurate and specific RNNs are sought. In our context, accuracy measures how often the net reacts to AF episodes similar to those in the training set. Specificity measures how different two models must be for the net being able to separate one from the other. The quality of the map depends on the RNN having the right amount of specificity: if the classifiers are too specific, there will be patients that are not visible in the map. If the specificity is too low, different parts of the map will be visible at the same time and the diagnosis will not be useful either.

LSTMs and GRUs are the most commonly used RNNs for classifying time series. In both cases, the input is distributed over a chain of cells and the main differences with previous RNNs are in the operations carried out within each cell, which will allow maintaining or forgetting information. LSTM cells consist of three gates: input, forget, and output gate. These multiplicative gates learn to manage the information passed so each memory cell decides what to store. GRU networks differ mainly in the number of gates: GRUs have two gates (input and forget gates are combined into a single gate) instead of three, which means lighter storage and faster training. Although LSTM has the ability to remember longer sequences, GRUs exhibit

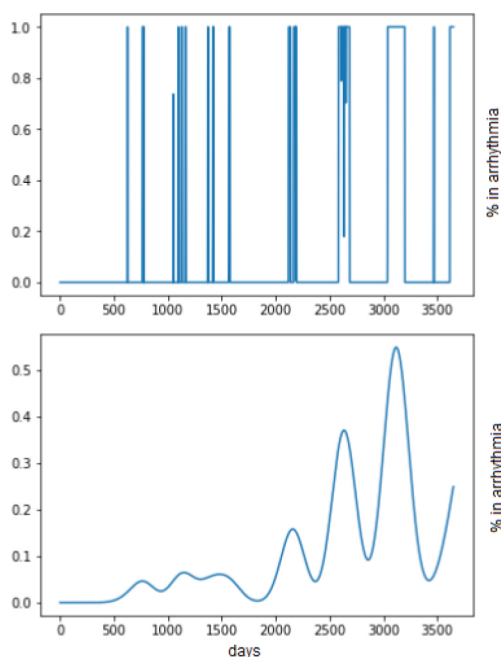


Figure 5 | Top: Synthetic sequence of episodes (simulation time: 10 years). Bottom: Continuous-time function measuring the degree of truth that the patient is undergoing an arrhythmia episode at time .

better performance on certain tasks [29,30], which makes us to consider them as an alternative for short-time series.

Training data is comprised by the postprocessed continuous-time functions defined in Section 3.1. In turn, two different methods were considered for training the RNNs:

1. **Error minimization:** the networks are trained for minimizing the squared error between the output of the net and the parameters of the Markov model. Alternatively, a set of clusters can be defined in the space of parameters of the model and the problem redefined as a multi-class classification task. The clusters in the space of parameters represent medical cases of interest, such as paroxysmal stable AF, paroxysmal AF with slow evolution to permanent AF, paroxysmal AF with quick evolution to permanent AF, permanent AF, and others. In this case, the concepts “accuracy” and “specificity” can be traced down to the confusion matrix of the classifier.
2. **GANs:** LSTMs or GRUs can be configured as GANs (see Figure 6.) GANs consist of 2 RNN: a generative net and a discriminative net. The generator net produces new data instances from noise, while the discriminator receives real data and the data from the generator and decides whether the generator’s data belongs to the same distribution as the real data. From this verdict, the parameters of both networks are adjusted to improve in the next iteration until the generator is able to produce realistic data, that is to say, sequences of arrhythmia episodes. If a GAN is trained with arrhythmias with specific features a discriminator will be obtained that separates arrhythmias of that type from any other kind of arrhythmia. It is remarked that for this particular application we are not interested in the

generative network, that is discarded after training (because the generative model introduced in Section 3.1 fulfills this function) but in the discriminative element. This process is repeated for each of the clusters in the space of parameters. A different GAN discriminator is learned for each class, and the generative map is the output of an ensemble that combines all the nets.

4. NUMERICAL RESULTS

The experimental validation of the proposed generative map has two parts. First, synthetic data with known properties is used to assess each of the presented alternatives. Second, actual patients are diagnosed, and their maps are validated by a human expert.

The experimental setup is described first. Second, the specificity of the GAN architecture is analyzed. In third place, the properties of LSTM and GRU networks are compared to that of GAN and also to non-neuronal classifiers. Fourth and last, some representative real-world cases are discussed.

4.1. Experimental Setup

The experimental setup is as follows: the code for training GAN recurrent networks for time series has been adapted from the publicly available code at <https://github.com/ratschlab/RGAN> [31].

A total of 14000 sequences have been generated for each combination of parameters chosen (60% was used for training, 20% for validation, and remaining 20% of data was used for testing). For multi-class problems, a softmax activation function is applied to the last layer of the LSTM- and GRU-based solutions in order to predict the class for the given pacemaker data. For GANs, each discriminator of the ensemble has its output passed through a sigmoid to determine whether the input belongs to the distribution data with which it was trained. Then all discriminator outputs are compared to determine which is the predicted class for the input.

4.2. Sensibility of the GAN-Based Approach

A brief study about the sensibility of the GAN-based maps has been included in Tables 1 and 2. The first table collects the results for $\alpha = 0.998$ (fast progression) and the second table contains the same experiments for $\alpha = 0.999$ (slow progression).

The meaning of the rows and columns of these tables is as follows: each column contains the fraction of correct classifications of a discriminator that has been trained with sequences produced by the generative model. The values of $\lambda_{NA}(0)$ used for computing these sequences are indicated in the column labels. The first and second rows, “Train” and “Test” are the percentage of correct detections of “True” sequences (generative models) versus “False” sequences (produced by the generator net in the GAN architecture). The rows labelled $\alpha = 0.997 \dots 0.999$ are the fraction of sequences with the same parameters as those used for training the net but a different parameter α . The remaining rows are the fraction of correct classifications when the net is fed with sequences with a different value of λ_{NA} .

These results show that the nets are highly responsive when the arrhythmia is paroxysmal (low values of λ_{NA} , thus time between

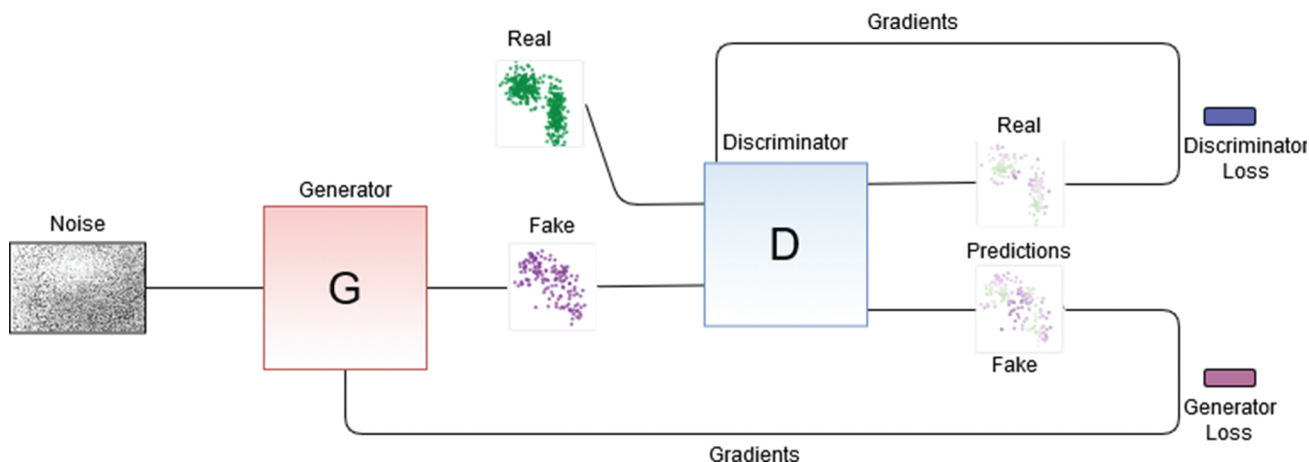


Figure 6 | Generative adversarial network (GAN) architecture for obtaining one of the discriminant elements. The red block represents the generator net which generates fake data that is passed to the discriminator (blue block). The latter decides what is true and what is false from the input data and the gradients are adjusted according to the true labels until a discriminator that knows exactly what type of arrhythmia that is being trained with is obtained.

Table 1 | Sensitivity of the discriminator for $\alpha = 0.998$.

	$\lambda_{NA} = 1.0/10$	$\lambda_{NA} = 1.0/30$	$\lambda_{NA} = 1.0/90$	$\lambda_{NA} = 1.0/180$	$\lambda_{NA} = 1.0/260$
Train	0.9794	0.9804	0.9830	0.9868	0.9800
Test	0.9779	0.97978	0.9811	0.9847	0.9797
$\alpha = 0.997$	0.5299	0.2523	0.5373	0.4324	0.4878
$\alpha = 0.999$	1.0000	1.0000	0.9979	0.3475	0.4424
$\lambda_{NA} = 1/5$	0.3333	1.0000	1.0000	1.0000	1.0000
$\lambda_{NA} = 1/10$	-	0.8162	1.0000	1.0000	1.0000
$\lambda_{NA} = 1/30$	0.9967	-	0.9505	0.9970	0.9983
$\lambda_{NA} = 1/90$	1.0000	0.9703	-	0.1369	0.1969
$\lambda_{NA} = 1/180$	1.0000	0.9994	0.0914	-	0.0312
$\lambda_{NA} = 1/260$	1.0000	1.0000	0.1494	0.0008	-

Table 2 | Sensitivity of the discriminator for $\alpha = 0.999$.

	$\lambda_{NA} = 1.0/10$	$\lambda_{NA} = 1.0/30$	$\lambda_{NA} = 1.0/90$	$\lambda_{NA} = 1.0/145$	$\lambda_{NA} = 1.0/180$
Train	0.9832	0.9823	0.9809	0.9825	0.9838
Test	0.9821	0.9818	0.9818	0.9853	0.9783
$\alpha = 0.997$	0.9986	0.9987	0.9986	0.9986	0.9997
$\alpha = 0.998$	0.9998	0.9998	0.9956	0.9485	0.9809
$\lambda_{NA} = 1/5$	0.1543	1.0000	1.0000	1.0000	1.0000
$\lambda_{NA} = 1/10$	-	1.0000	1.0000	1.0000	1.0000
$\lambda_{NA} = 1/30$	1.0000	-	0.9988	0.9997	0.9996
$\lambda_{NA} = 1/90$	1.0000	0.9978	-	0.1703	0.2002
$\lambda_{NA} = 1/120$	0.9800	0.9800	0.0012	0.0516	0.0566
$\lambda_{NA} = 1/145$	1.0000	1.0000	0.0001	-	0.0357
$\lambda_{NA} = 1/260$	1.0000	1.0000	0.0000	0.0008	0.0089

episodes is high). This is the desired result, because these are the cases with clinical interest. The net is less capable when λ_{NA} is high, however these are the cases where the patient is in a permanent arrhythmia condition at the beginning of the experiments thus the evolution of the patient is self-evident.

4.3. Compared Results

In this section, 6 of the 10 AF categories used in the preceding subsection are used. These classes are labelled 998na10, 998na30, 998na180, 999na10, 999na30, and 999na180. The class labels begin with the first three decimals of α , which is the speed of the progression of the AF (998 is slow, 999 is fast). The second number in

the class label is $1/\lambda_{NA}(0)$, which is the average time between two AF episodes, measured in days (10, 30, and 180 days). Accuracy and sensitivity of the classifier are assessed by means of a confusion matrix where the number of times that an AF was correctly diagnosed is counted, and in this last case the deviation between the prediction and the desired value is also accounted for.

The different RNNs discussed in the preceding section are compared between them and also to two other standard nondeep learning classification methods, that have been included as a baseline: Multilayer Perceptron (MLP) and Random Forest. Table 3 collects the performance of the different models for each different class in terms of accuracy, i.e., each entry in Table 3 is the number of times

Table 3 Accuracy of the different classifiers, six types of AF.

	Accuracy				
	MLP	Random Forest	GRU	LSTM	GAN Ensemble
998na10	0.9921 (3)	0.9918 (4)	0.9964 (1)	0.9943 (2)	0.9782 (5)
998na30	0.9654 (5)	0.9857 (3)	0.9911 (1)	0.9875 (2)	0.9686 (4)
998na180	0.9371 (5)	0.9800 (3)	0.9879 (1)	0.9946 (2)	0.9596 (4)
999na10	0.9739 (5)	0.9943 (3)	1.0000 (1.5)	1.0000 (1.5)	0.9803 (4)
999na30	0.9368 (5)	0.9979 (3)	0.9996 (1.5)	0.9996 (1.5)	0.9911 (4)
999na180	0.9911 (5)	0.9946 (3)	0.9982 (1)	0.9957 (2)	0.9796 (5)
Summary Results					
Accuracy	0.9661	0.9907	0.9955	0.9953	0.9762
Average rank	4.6666	3.1666	1.1666	1.8333	4.3333

Note: AF, atrial fibrillation; MLP, multilayer perceptron; GRU, gated recurrent unit; LSTM, long short-term memory; GAN, generative adversarial network.

that a series that was generated by the correct model was recognized as such. Also, to illustrate the performance of each method the ranking computed by Friedmans method for each dataset and the averaged resulting ranking is added.

Observe that in all cases RNNs improve the results of MLP and Random Forest. In terms of accuracy, GRU is the RNN that better exploits the incomplete information in truncated ICD event series. It is better than MLP, Random Forest, and GAN with a p-value lower than 0.012 (according to Bonferroni correction [32]), followed by LSTM, although the difference is not statistically significant. LSTMs in GAN configuration apparently do not improve simpler classifiers such as Random Forest but their specificity is better and this metric has a higher impact in the visual coherence of the map. This point will be made clearer in Subsection 4.4. Observe also this metric is heavily dependent on the chosen division of the AF in clusters. To illustrate this fact, in Table 4 the same experiments carried out in Table 3 were repeated for a division in 8 classes (class labels 998na90 and 999na90 were added, with 90 days between AF episodes). The new classes are not easily separated from those with 180 days and the mean accuracy of the classifiers decreases.

Observe that the visual perception is much different if, e.g., a patient whose AF episodes occur every 180 days is assigned 90, 30, or 10 days. In order to keep the perceptual coherence the cost of misclassifying arrhythmias must not be uniform. This will be illustrated too in Section 4.4. In this respect, Figure 7 contains the confusion matrices of GAN (left) and Random Forest (right) for the initial division in six AF types. Observe that the number of correctly classified series is better for Random Forest, as expected, but there are two cells with errors that cannot be accepted from the medical diagnosis point of view: the cell 998na10- 999na10 (wrong rate of evolution for the same time between episodes) and, of secondary importance the cell 998na30- 999na180 (wrong rate of evolution and the initial time between episodes in the fast case is higher).

Observe that this behavior can be corrected if a cost matrix is introduced in the problem, although the problem of choosing the best cost matrix remains. For instance, if the cost matrix

$$c_{i,j} = \sum_{i \neq j}^N |i - j|^k, \quad (3)$$

(where $N = 6$, the number of classes) is used, the weighted accuracy of the GAN method would be better for values of $k > 1.73$.

Table 4 Accuracy of LSTM and GRU, eight types of AF.

	Accuracy	
	GRU	LSTM
998na10	0.9982	0.9968
998na30	0.9764	0.9796
998na90	0.8343	0.8529
998na180	0.8754	0.8464
999na10	1.0000	0.9982
999na30	0.9989	0.9975
999na90	0.8521	0.8904
999na180	0.9471	0.9286
Summary Results		
Accuracy	0.9353	0.9363

Note: AF, atrial fibrillation; GRU, gated recurrent unit; LSTM, long short-term memory.

4.4. Graphical Representation and Discussion

Three different experiments will be carried in this section. First, maps generated with different architectures (GAN and minimal error) are compared on data generated by the model. Second, two maps with minimal error and different clusterings of the generative model parameters are compared. Third, a true patient will be diagnosed by a human expert and by means of the proposed map.

4.4.1. Random forest versus LSTM-GAN

Two maps (see Figure 8) were selected for illustrating the differences between maps comprising RNNs and maps comprising other classifiers. The left map was obtained with an LSTM in a GAN configuration. The map in the right panel of the same figure was derived from a Random Forest. The horizontal axis is labelled β , which is the inverse of the parameter λ_{NA} , and can be understood as the expected number of days between two AF episodes at time $t = 0$. The vertical axis is labelled α and measures the speed of the progression. The lower the value of α , the quickest the progression to permanent AF. The color code is shown in the bar at the right. Red areas are the highest activations, and blue areas the lowest.

Data is a random sample of the model with parameters $\alpha = 0.998$ and $\beta = 1/\lambda_{NA} = 30$. The proper diagnosis would be a red dot at coordinates (30, 0.998). Observe that the confidence of the

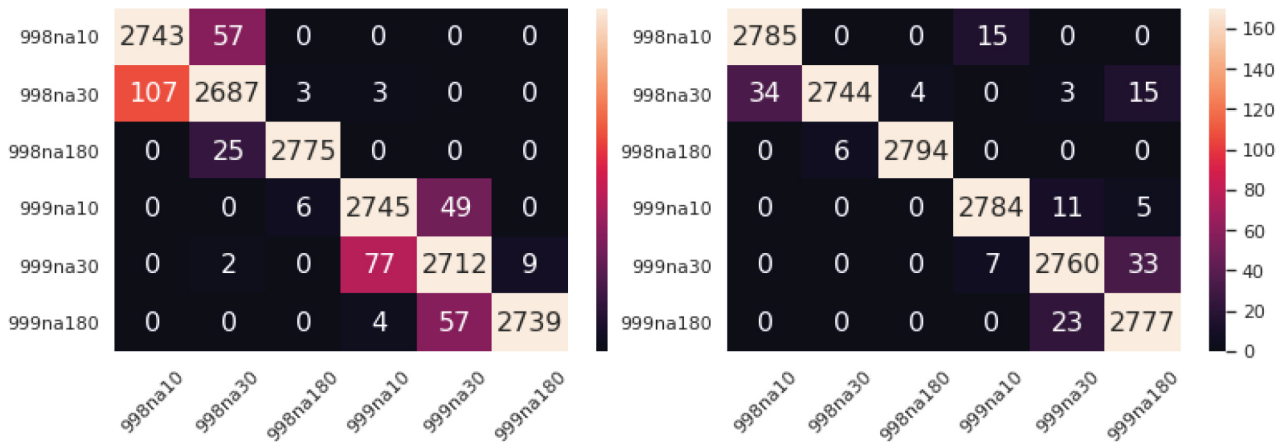


Figure 7 | Left: Generative adversarial network (GAN) ensemble confusion matrix. Right: Random Forest confusion matrix. Similar classes are nearby on the map, thus errors in prediction should be close to the diagonal.

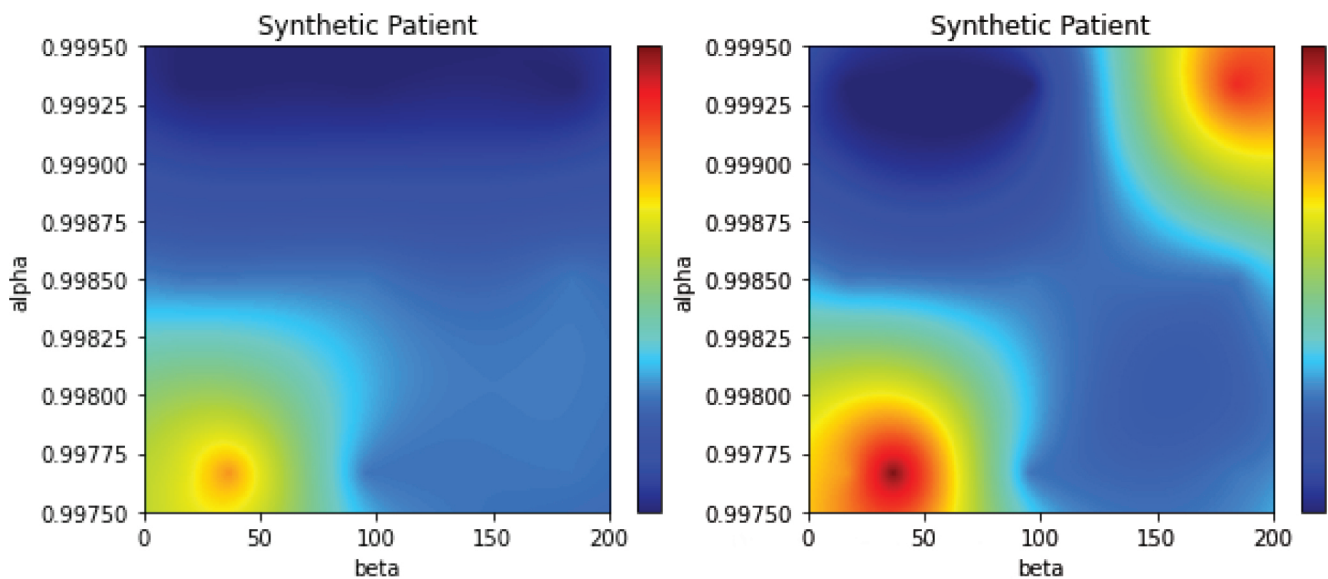


Figure 8 | Left: Generative adversarial network (GAN) map for simulated atrial fibrillation (AF) $\alpha = 0.998$, $\beta = 30$. Right: Random Forest-based map for the same data.

detectors in the correct area is higher for the map in the right, but there is also a clear red dot in the upper right corner that is an artifact of the classifier. This pair of maps illustrates the problem indicated in the preceding subsection: the presence of misclassifications that are far from the diagonal in the confusion matrix causes that abnormal regions in the map are activated, while the misclassifications near the diagonal are perceived as an increase the area around the correct diagnostic. In this respect, LSTMs and GRU produce results with a higher quality in terms of the medical diagnosis and furthermore it is not needed that a cost matrix is introduced in the classification task.

4.4.2. Effect of the different clustering in the generative model parameters

In Table 4 we shown that the division of the AF in categories influenced the accuracy of the RNNs. In Figure 9 two LSTM-based maps are compared. In the left panel, AF is divided into the six categories 998na10, 998na30, 998na180, 999na10, 999na30, and 999na180. In the right panel of the same figure the two additional categories

were added, named 998na90 and 999na90. These two categories are harder to separate and the global accuracy decreases. The resulting maps are correct (both maps have maximum activations centered at $\beta = 20$ and $\alpha = 0.9994$ but the right map has a much higher area of uncertainty).

4.4.3. Diagnosis of an actual patient

Actual data downloaded from the ICD of a patient with paroxysmal arrhythmia is displayed in Figure 10. The black spikes are clusters of events (the isolated AMS events are not visible at this time scale). About three years of data are included in the figure. Observe that the time between events is higher in the first two years and the pace increases quickly in the last part (around the mark of the day 1000).

In Figure 11 three maps are displayed with the same conventions seen in the preceding subsection. The map in the left of the upper panel has been obtained with GRU, and map in the right in the same panel is produced by a LSTM network. The map in the bottom panel was obtained with another LSTM in GAN configuration.

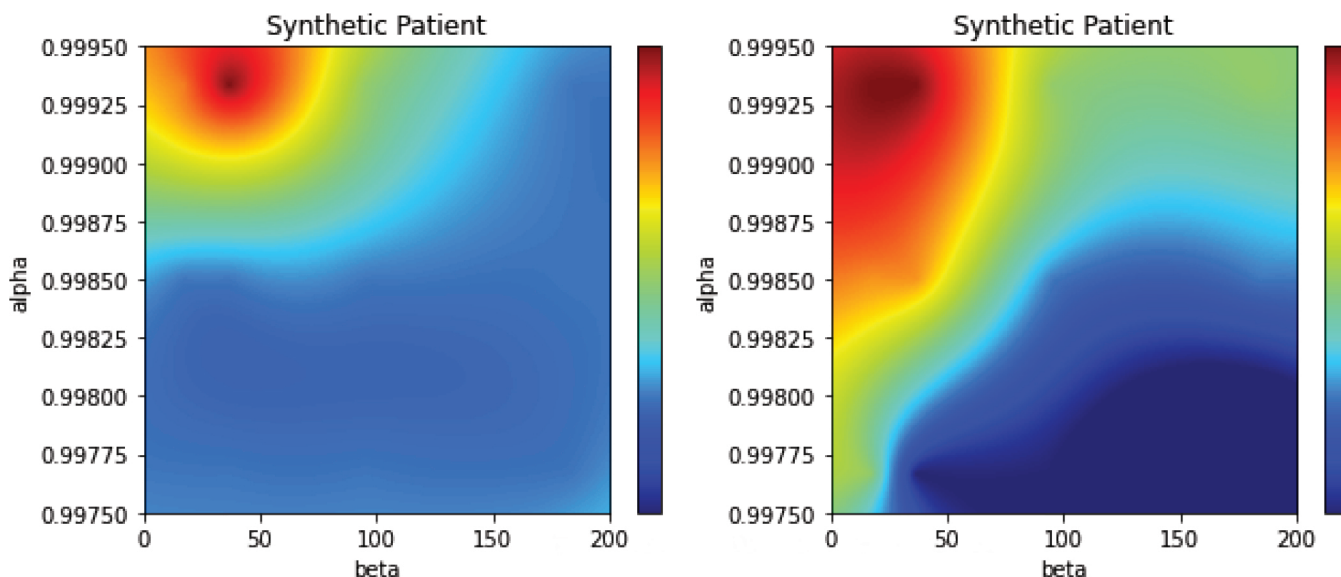


Figure 9 | Left: Long short-term memory (LSTM)-based map, 6 clusters of atrial fibrillation (AF). Generative model with. Right: Same data, 8 clusters of AF.

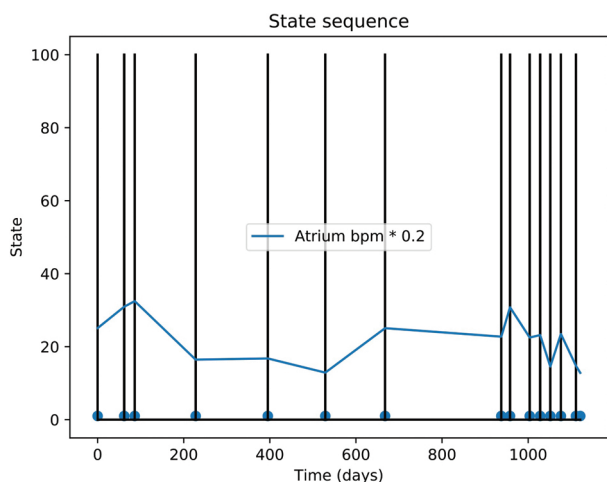


Figure 10 | Dates of the automatic mode switching (AMS) events (black lines) and atrium beats per minute (bpm * 0.2) for an actual patient.

The three maps are similar and produce coherent results. The interpretation of these maps is as follows: the red region is centered in $\alpha = 0.9994$ and $\beta = 180$. This means that the patient began suffering AF episodes every 6 months, but the evolution of the arrhythmia is moderate and is expected that the average time between episodes is multiplied by 0.77 every year.

Observe that the map for an actual patient is not as specific as the maps obtained from data from the generative model. This means that it cannot be discarded that the patient has episodes every 3–4 months and his/her evolution is faster, up to a reduction factor of 0.58 per year. If Figure 10 is recalled, the number of episodes in the first 100 days was of three, but the following three episodes happened in more than one year, thus this kind of uncertainty in the diagnosis is correct, although the most probable diagnosis is that of a slow evolution.

5. CONCLUDING REMARKS AND FUTURE WORK

We have shown that iECGs from ICDs and pacemakers can be used to a certain extent for predicting the change from paroxysmal to permanent AF. The main difficulty is with the short length of the pacemaker records, that has been addressed here by means of a graphical projection of the sequence of AMS events in the parameter space of a generative model. If the data is enough for a clear diagnosis, the map produces an estimation of the patient condition and future evolution, and in those cases where the data is insufficient the map produces a set of estimations that can be subjectively assessed in order to determine whether the evolution is positive or not. Such a diagnosis can help specialists reduce the time spent analyzing intracardiac data.

LSTM and GRU have shown remarkable results as a standalone multi-class classifier, and LSTM was adequate as a part an ensemble of GAN detectors as well. GANs have an intrinsic advantage, that is the obtention of the generator network, that may be a better generative model than the continuous Markov model used in this study. If a number of ICD records of actual patients was high enough, it would make sense to bootstrap the model with the generative model described in this paper and fine-tune the GANs with real-world data, for obtaining an improved generative model. Such a GAN-based generative model could have an application on its own, as a predictor of future AF episodes. Lastly, we are currently working in other alternatives than GANs for obtaining the diagnostic map, such as the use of Variational Autoencoders, than can also be trained on model-generated data and be applied to ICD logs to get a compact representation of the evolution of AF.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

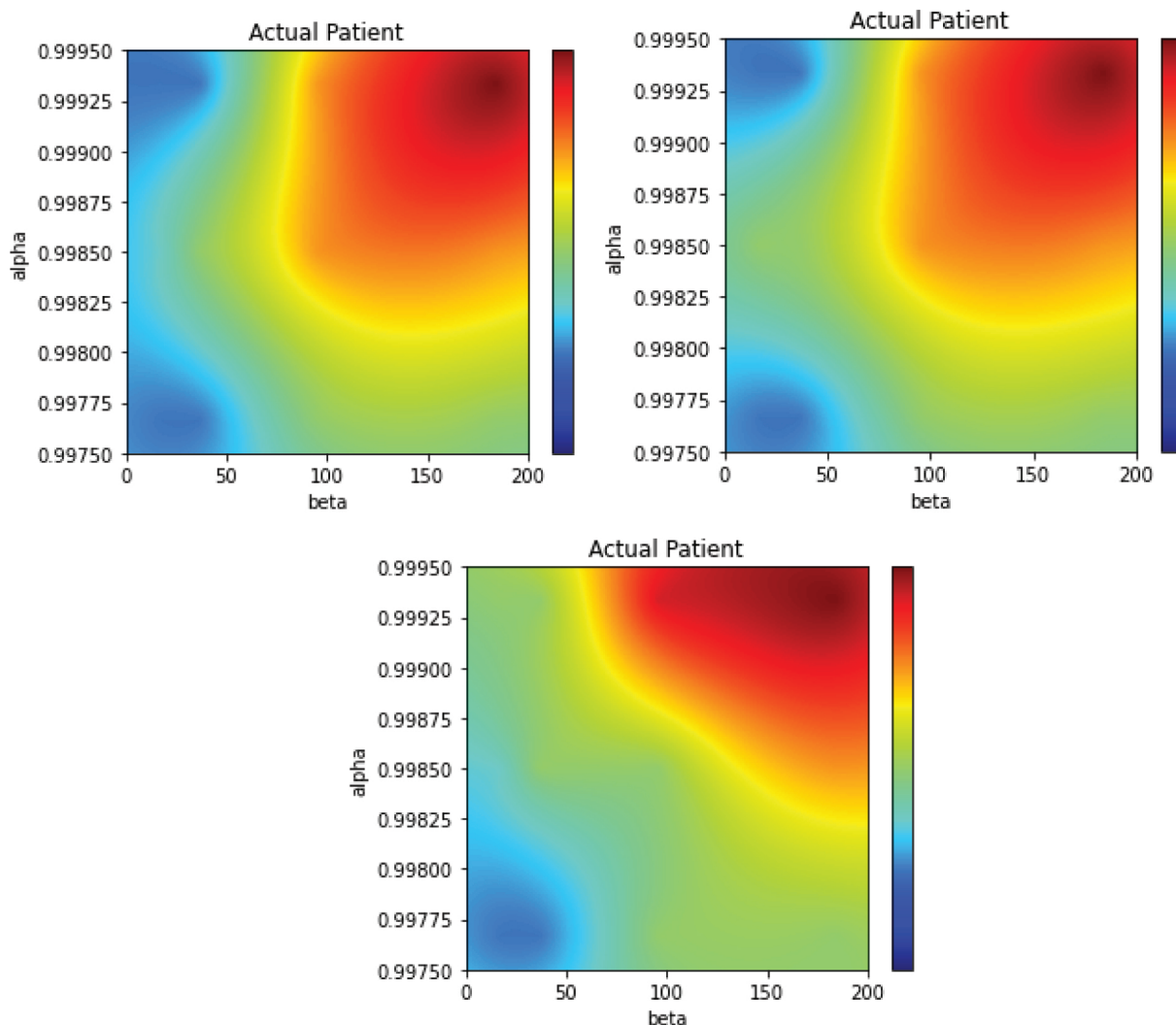


Figure 11 | Maps of the patient in Figure 10. Top panel, left: multi-class gated recurrent unit (GRU). Top panel, right: multi-class long short-term memory (LSTM). Bottom panel: LSTM in generative adversarial network (GAN) configuration.

AUTHORS' CONTRIBUTIONS

Conceptualization: Nahuel Costa, Jesús Fernández, Luciano Sánchez; Methodology: Nahuel Costa, Luciano Sánchez; Software: Nahuel Costa; Validation: Nahuel Costa, Inés Couso.

ACKNOWLEDGMENTS

This work has been partially supported by the Ministry of Economy, Industry and Competitiveness (“Ministerio de Economía, Industria y Competitividad”) from Spain/FEDER under grant TIN2017-84804-R and by the Regional Ministry of the Principality of Asturias (“Consejería de Empleo, Industria y Turismo del Principado de Asturias”) under grant GRUPIN18-226.

REFERENCES

- [1] H. Ogawa, Y. An, S. Ikeda, Y. Aono, K. Doi, M. Ishii, *et al.*, Progression from paroxysmal to sustained atrial fibrillation is associated with increased adverse events, *Stroke*. 49 (2018), 2301–2308.
- [2] G.J. Padfield, C. Steinberg, J. Swampillai, H. Qian, S.J. Connolly, P. Dorian, *et al.*, Progression of paroxysmal to persistent atrial fibrillation: 10-year follow-up in the Canadian registry of atrial fibrillation, *Heart Rhythm*. 14 (2017), 801–807.
- [3] F. Holmqvist, S. Kim, B.A. Steinberg, J.A. Reiffel, K.W. Mahaffey, B.J. Gersh, *et al.*, Heart rate is associated with progression of atrial fibrillation, independent of rhythm, *Heart*. 101 (2015), 894–899.
- [4] P.K.D. Pramanik, B.K. Upadhyaya, S. Pal, T. Pal, Chapter 1 - internet of things, smart sensors, and pervasive systems: enabling connected and pervasive healthcare, in: N. Dey, A.S. Ashour, C. Bhatt, J. Fong (Eds.), *Healthcare Data Analytics and Management, Advances in Ubiquitous Sensing Applications for Healthcare*, Academic Press, Durgapur, India, 2019, pp. 1–58.
- [5] A. Bansal, R. Joshi, Portable out-of-hospital electrocardiography: a review of current technologies, *J. Arrhythm*. 34 (2018), 129–138.
- [6] M.P. Turakhia, M. Desai, H. Hedlin, A. Rajmane, N. Talati, T. Ferris, *et al.*, Rationale and design of a large-scale, app-based study to identify cardiac arrhythmias using a smartwatch: the apple heart study, *Am. Heart J.* 207 (2019), 66–75.

- [7] D. Calvo, J. Rubín, D. Pérez, J. Jalife, Spectral analysis of electrograms in a substrate modified by radiofrequency ablation reveals similarities between organized and disorganized atrial rhythms, *Heart Rhythm*. 11 (2014), 2306–2309.
- [8] G. Neal Kay, K.A. Ellenbogen, M. Giudici, M.M. Redfield, L.S. Jenkins, M. Mianulli, B. Wilkoff, The ablate and pace trial: a prospective study of catheter ablation of the av conduction system and permanent pacemaker implantation for treatment of atrial fibrillation, *J. Interv. Card. Electr.* 2 (1998), 121–135.
- [9] F. Roosevelt Gilliam III, T-wave oversensing in implantable cardiac defibrillators is due to technical failure of device sensing, *J. Cardiovasc. Electrophysiol.* 17 (2006), 553–556.
- [10] C.D. Swerdlow, G. Kalahasty, K.A. Ellenbogen, Implantable cardiac defibrillator lead failure and management, *J. Am. Coll. Cardiol.* 67 (2016), 1358–1368.
- [11] M. González, C. Bergmeir, I. Triguero, Y. Rodríguez, J.M. Benítez, Self-labeling techniques for semi-supervised time series classification: an empirical study, *Knowl. Inf. Syst.* 55 (2018), 493–528.
- [12] V. Fortuin, M. Häser, F. Locatello, H. Strathmann, G. Rätsch, Deep self-organization: interpretable discrete representation learning on time series, in *International Conference on Learning Representations*, New Orleans, 2019. arXiv preprint arXiv:1806.02199
- [13] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, *PloS One*. 10 (2015), e0130140.
- [14] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: visual explanations from deep networks via gradient-based localization, in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 618–626.
- [15] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, *et al.*, Chexpert: a large chest radiograph dataset with uncertainty labels and expert comparison, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, 2019, vol. 3, pp. 590–597.
- [16] R.C. Fong, A. Vedaldi, Interpretable explanations of black boxes by meaningful perturbation, in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 3429–3437.
- [17] D. Alvarez-Melis, T.S. Jaakkola, A causal framework for explaining the predictions of black-box sequence-to-sequence models, arXiv preprint arXiv:1707.01943, 2017.
- [18] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (1997), 1735–1780.
- [19] K. Cho, B. Van Merriënboer, D. Bahdanau, C. Gulcehre, H. Schwenk, F. Bougares, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078, 2014.
- [20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, K.Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., Montreal, Canada, 2014, pp. 2672–2680.
- [21] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 1125–1134.
- [22] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D.N. Metaxas, Stackgan: text to photo-realistic image synthesis with stacked generative adversarial networks, in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 5907–5915.
- [23] J.-Y. Zhu, T. Park, P. Isola, A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 2242–2251.
- [24] A. Brock, J. Donahue, K. Simonyan, Large scale GAN training for high fidelity natural image synthesis, in *International Conference on Learning Representations*, New Orleans, 2019. arXiv preprint arXiv:1809.11096v2
- [25] T. Iqbal, H. Ali, Generative adversarial network for medical images (mi-gan), *J. Med. Syst.* 42 (2018), 1–11.
- [26] J. Fernández, J. Velasco, L. Sánchez, Detection of cardiac arrhythmias through singular spectrum analysis of a time-distorted egm signal, in: H.P. García, J. Alfonso-Cendón, L.S. González, H. Quintián, E. Corchado (Eds.), *International Joint Conference SOCO'17-CISIS'17-ICEUTE'17* León, Spain, Springer International Publishing, Cham, Switzerland, 2017,
- [27] M.D. Peláez-Aguilera, M. Espinilla, M.R.F. Olmo, J. Medina, Fuzzy linguistic protoforms to summarize heart rate streams of patients with ischemic heart disease, *Complexity*. 2019 (2019), 1–11.
- [28] F. Karim, S. Majumdar, H. Darabi, Insights into LSTM fully convolutional networks for time series classification, *IEEE Access*. 7 (2019), 67718–67725.
- [29] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555, 2014.
- [30] M. Ravanelli, P. Brakel, M. Omologo, Y. Bengio, Improving speech recognition by revising gated recurrent units, arXiv preprint arXiv:1710.00641, 2017.
- [31] C. Esteban, S.L. Hyland, G. Rätsch, Real-valued (medical) time series generation with recurrent conditional GANs, arXiv preprint arXiv:1706.02633, 2017.
- [32] S. Garcia, F. Herrera, An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons, *J. Mach. Learn. Res.* 9 (2008), 2677–2694. <https://www.jmlr.org/papers/v9/garcia08a.html>

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Semi-supervised recurrent Variational Autoencoder approach for visual diagnosis of Atrial Fibrillation

NAHUEL COSTA¹, LUCIANO SÁNCHEZ², AND INÉS COUSO³.

¹Computer Science Department, University of Oviedo, Gijón, Asturias, Spain (e-mail: costanahuel@uniovi.es)

²Computer Science Department, University of Oviedo, Gijón, Asturias, Spain (e-mail: luciano@uniovi.es)

³Statistics Department, University of Oviedo, Gijón, Asturias, Spain (e-mail: couso@uniovi.es)

Corresponding author: Nahuel Costa (e-mail: costanahuel@uniovi.es).

This work has been partially supported by the Ministry of Economy, Industry and Competitiveness (“Ministerio de Economía, Industria y Competitividad”) of Spain/FEDER under grants TIN2017-84804-R and PID2020-112726-RB.

ABSTRACT In this work we propose a semi-supervised framework to visually assess the progression of time series. To this end, we present a recurrent version of the VAE to exploit the generative properties that lead it to learn in an unsupervised way a continuous compressed representation of the data. We introduce a classifier in the VAE training process to control the regulation of the latent space, allowing the network to learn latent variables that set the basis for creating an explainable evaluation of the data. We use the proposed framework to address the diagnosis of Atrial Fibrillation (AF) first validating it with simulated data with known properties and subsequently testing it with intracardiac data obtained from pacemakers and defibrillator systems.

INDEX TERMS Graphical Analysis, Heart Disease, Recurrent Neural Networks, Time Series, Variational Autoencoder

I. INTRODUCTION

MOST AI-based systems have a black box nature that allows powerful predictions, but cannot be directly explained. This is especially true when it comes to time series data, where the bulk of methods stick to rawly classifying or predicting a number or a set of numbers. Unsupervised learning approaches are a possible alternative for this. Within this paradigm Autoencoders are one of the most promising methods that we can find. Autoencoders are a family of neural networks that have the ability to learn a simplified representation of the data, typically for dimensionality reduction. These networks are designed to reconstruct the input data while at the same time learn a compressed representation of it; the so-called latent space. Variations of the original model [1] [2] [3] have been developed in order to enhance classification and clustering tasks until the emergence of Variational autoencoders (VAEs), whose main purpose is the generation of new data.

Variational autoencoders are rooted in Bayesian inference [4] and are comprised of an encoder function $q_\phi(z|x)$ and a decoder function $p_\theta(x|z)$ where z is the latent encoding vector, x is the input data and ϕ and θ are parameters that initialize a probability distribution. By introducing the Kullback-

Leibler divergence into the loss function, which simply measures how much one probability distribution diverges from another, the above-mentioned parameters corresponding to the input data distribution can be learned. This, together with a reconstruction error added to the loss function, allows the model to produce a latent space in which similar data will be located close to each other and also enables new data to be sampled from points that do not belong to the original data, thus having a generative model.

The main difference between VAEs and the rest of autoencoders lies in the learned latent space: The inputs are not coded to a set of fixed vectors, but the compression depends on a probability distribution $q_\phi(z|x)$ instead, causing the data to be organised in a continuous space, i.e. two nearby points in the latent space should give similar contents when reconstructed (Figure 1). Precisely, other unsupervised techniques such as clustering algorithms lack this property. Although they do prioritize grouping data of a similar nature, the visual disposition of the clusters can often be arbitrary. On the other hand, neither can the VAE latent space be used for clustering since the encoded data tend to be overlap to prioritize the generative process.

Therefore, a methodology capable of combining the above

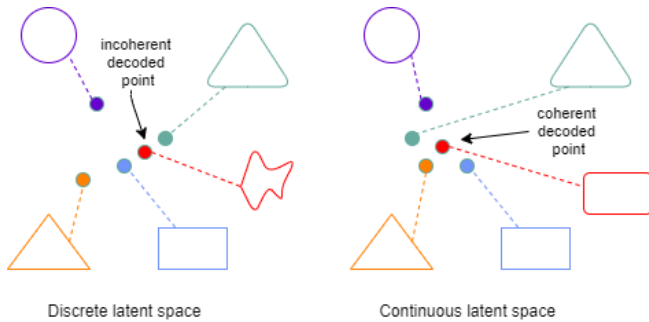


FIGURE 1: Simplified representation of the compression resulting from a vanilla Autoencoder (left) and a VAE (right). When the latent space is continuous, the organisation of the data allows decoding of a meaningful figure, in this case a cross between a rectangle and a triangle, thus favouring the generation of new data.

properties, that is, depicting the input data into clusters, while preserving a continuous representation according to its underlying complexion, would be of interest to time series data. In fact, there has been a recent interest in seeking such a model, as can be shown in [5] [6] [7] [8], which make use of Variational Autoencoders together with Gaussian Mixtures in order to achieve an interpretable clustering. Nevertheless, these approaches are not intended to be applied in time series.

Besides, although VAEs have proven to be efficient in multiple domains, mainly related to computer vision [9] [10] [11] and Natural Language Processing (NLP) [12] [13] [14], as generative frameworks as well as data compressors, there is a lack of research when it comes to time series. In [15] the authors present a VAE model that can map time series to a latent vector representation, but the model has become obsolete due to more recent advances in recurrent architectures. Other promising work has begun to emerge: In [16] LSTM networks are used to model the temporal complexion of the data, whereas in [17] the authors propose to use echo-state networks for the same objective. Despite the fact that these works combine recurrent architectures with VAEs, their goal differs from ours since they aim to detect anomalies based either on reconstruction errors or on anomaly scores, while what we are pursuing is an interpretable assessment of time series.

The solution that we propose is to introduce a recurrent version of the VAE to deal with temporary data along with the inclusion of a classifier in the training process that controls the regularisation of the latent space to prevent the resulting clusters from overlapping. In this way, a representation that can be used for displaying a graphic map that gives insight into the evolution of the time series is obtained.

The creation of such a model is motivated by the need to offer a solution to a problem in which the presence of efficient algorithms is limited: the diagnosis of Atrial Fibrillation (AF). AF is the most common type of arrhythmia in clinical practice. It is a type of heartbeat in which the atria tremble,

causing an irregular and accelerated heart rhythm.

The treatment of the disease often involves the use of pacemakers. These devices are a source of data that record the dates and lengths of the episodes of high atrial rate, comprising a historical record, that is, a time series. Effective and accurate diagnosis of this condition remains challenging these days. Also, a simple prediction may not be informative enough for specialists to examine the state of the disease. Thus, a variational-clustering approach is tailored to our needs in order to accomplish a visual diagnosis capable of assessing the evolution of AF.

The structure of this paper is organised as follows: Section II introduces the importance in the treatment of this condition and the difficulties associated with its diagnosis. A detailed description of the proposed method comprising the semi-supervised VAE framework for achieving an explainable diagnosis is described in Section III. Before reporting experimental results in Section V, an illustrative problem is presented in Section IV while conclusions are drawn in Section VI.

II. AF DIAGNOSIS

AF is an abnormal heartbeat usually presented in the elderly. The course of the disease can lead to a progression from paroxysmal arrhythmia (arrhythmia episodes that appear and disappear spontaneously) to persistent arrhythmia (episodes that last at least seven days and do not end without external intervention) or to permanent arrhythmia (uninterrupted episodes). The progression of AF is a complex process that depends on several risk factors [18], and an early diagnosis may condition the provision of optimal treatment.

Episodes of AF are easily detected on surface electrocardiograms (ECGs), obtained from non-invasive devices, but the activity recorded is over a very specific period of time, which in no case is enough to capture the evolution of the disease. Portable ECG monitors are an advantage in this respect and recent advances in healthcare facilitate continuous monitoring of intracardiac activities. This is beneficial in detecting pathological signatures and arrhythmias [19], especially in patients in the latter stages of permanent AF [20]. On the contrary, the health risks for patients in the early stages of paroxysmal arrhythmia are lower and the disadvantages of wearing these devices continuously outweigh the advantages.

It seems that the situation may change in the near future as new ECG sensors are small enough to be incorporated into wearable devices. The Apple Heart Study [21] shows that different types of AF can be detected in smartwatches, though the battery consumption is high, which prevents the sensor from being always on. To date, the detection and timing of short AF episodes remains an open problem.

In those patients in latter stages of the disease, pacemakers or IDCs (Implantable Cardiac Defibrillators) are normally used to control the heart rate [23] in order to keep common symptoms such as dizziness or chest pain under control. These devices provide heart rhythm monitoring, being able to detect episodes of arrhythmia, specifically of high atrial rate,

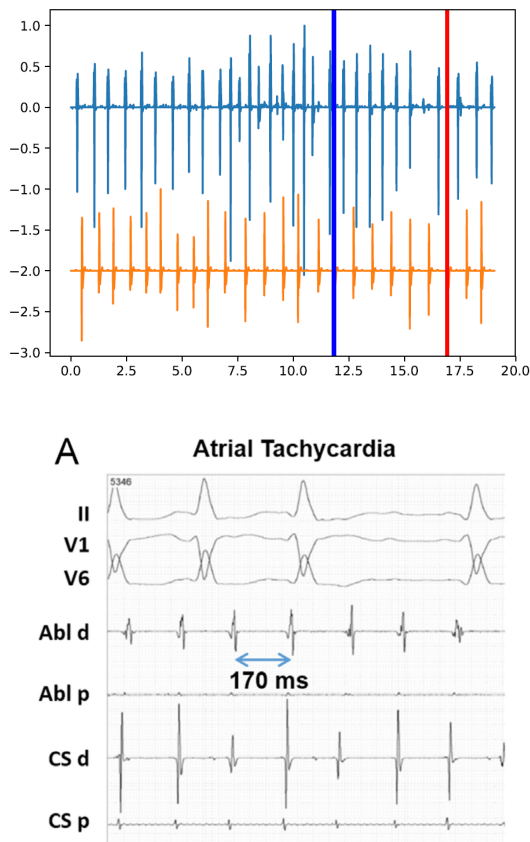


FIGURE 2: Top: Intracardiac ECG. The morphology of the surface ECG is not kept in the iECG, where there is only one peak for each heartbeat. Bottom: Surface ECG (taken from reference [22]).

which normally correspond to AF episodes. Intra-cardiac electrocardiograms (iECGs) are stored in the memory of these devices (see Figure 2 upper part) which are representations of the difference in potential between two points in contact with the myocardium in space over time. They keep a record of information seconds before and after the detection of each episode. This includes only the instantaneous frequencies of the atrium and ventricle because the morphology of the heartbeat is lost in the high-pass filtering at the IDC electrode, unlike surface ECGs (see Figure 2 lower part). Moreover, this information is not used for diagnosis, but rather to adjust the operational parameters of the device.

Given these facts, the most reliable source of information to work with, are the dates and lengths of the recorded episodes, which are also stored. This is a drawback because patients with multiple episodes will probably be in the latest stages of AF, but the interest lies in patients with a short history (initial episodes) so that the worsening rhythm of AF can be anticipated. This goal is very challenging because it is difficult to find a model that fits such a small amount of data (see Figure 3). Besides, the fact that the data are non-

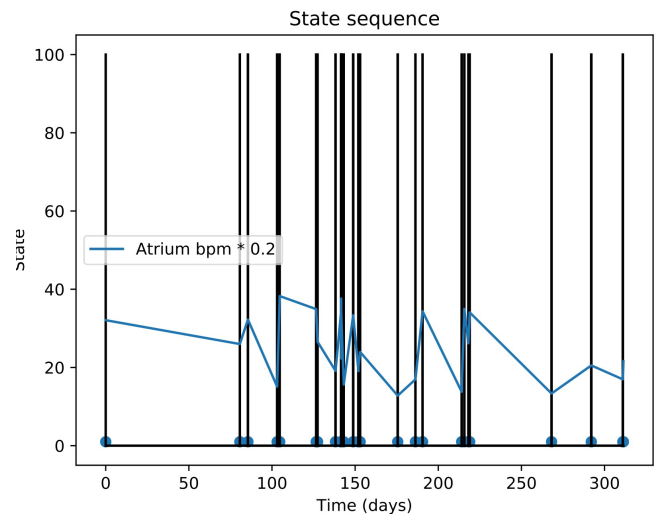


FIGURE 3: Top: Dates of pacemaker mode changes during a year Bottom: Recorded length of the AF episodes.

stationary makes the problem even more complicated, and this is exactly what we want to predict on the basis of a short sample, the transition from paroxysmal to permanent arrhythmias.

There are additional difficulties [24] because the algorithm used by the IDC to determine the duration of the episodes is not completely reliable. The device parameters are adjusted based on the iECGs mentioned above and safety concerns prevail, so the false positive rate is high. This leads to long episodes of AF that are sometimes mistakenly reported as short episode sets, so preprocessing is needed to take these spurious events into account, which in turn causes the number of episodes to be reduced even further.

All these reasons being explained, it is clear that the progression of AF is a complex process that depends on many different factors. To ease the interpretation of these factors, we are looking for a model capable of providing clinical staff with a diagnostic tool that can accurately determine the status of a patient with AF, therefore something more than a straightforward prediction is pursued. A possible path is to establish a Representation Learning approach since, unlike others, the performance of models following this approach depends directly on the internal representations, which in turn can be leveraged in favour of a better understanding of the problem itself. Typically, an algorithm capable of learning the characteristics that best represent the underlying data distribution is required, making it easier to perform other tasks such as classification or prediction. Since Principal Component Analysis (PCA) was developed, Representation Learning has been investigated to overcome the challenges of high dimensionality. Over the last decade, Deep Learning has been taking an important role in this field through supervised and unsupervised learning strategies, where it has had a great impact due to the feasibility of processing temporal/spatial

data or images more efficiently than superficial methods such as ICA, LDA or LLE.

Representation Learning has been employed in several areas of medicine for purposes such as risk factor selection, disease phenotyping, and prediction or classification of disease risks [25]. This line of research is key to developing explainable AI, where the results can be interpreted by human experts. There is a lack of work in this direction concerning the diagnosis of cardiovascular diseases where the few works that exist are focused on image processing [26] or simple classifiers are developed for time series data [27]. Regarding the diagnosis of AF, the vast majority of papers analyze ECG data from non-invasive devices [28], which are compatible with patients in the early stages of the disease or without previous pathologies, hence they are out of the scope of this work. There are also incentive contributions [29] where the authors study data from wrist-worn devices with convolutional networks. Nonetheless, pacemakers are still the devices that can provide valuable information in those patients in more advanced stages of the disease.

In our previous work [30] we tried to contribute to this path by presenting a graphical approach for analyzing the progression of AF using the output of Recurrent Neural Networks (RNN). Activations of the last layers of LSTM and GRU classifiers were used to create a topological map that provided an intuitive visual diagnosis. Although the results achieved were significant in terms of accuracy, the interpolation used to create the map obtained was highly sensitive to the differences between the neuron activations, which might provoke inconsistencies in the map. For that reason, a recurrent version of Generative Adversarial Networks (GAN) [31] was also introduced to use an ensemble formed by the discriminative part of these nets trained on different types of arrhythmia in order to learn a representation according to the complexion of the data. Nevertheless, the classification results were outperformed by the LSTM and GRU classifiers.

What we propose in this work is to exploit the representation learning potential of VAEs in order to provide a visual early diagnosis of the evolution of AF. VAEs have the ability to condense data from a high dimensionality to a much smaller dimension, maintaining consistency between the distance of the data that is reduced. The influence of Bayesian Variational Inference provokes the learned latent space to depict a two-dimensional projection of the data according to its nature, which is accurate to the AF problem: this resulting latent space can be interpreted as an explainable map where the distances between the compressed input data, are correlated with the differences between the various types of arrhythmia. When a sample of a patient's data is presented to the VAE, the resulting location on the map gives insight into the state of the intracardiac activity of the patient and how the disease might evolve in a short period of time, due to proximity to other nearby points.

III. PROPOSED METHOD

This section describes the proposed framework for performing the task of evaluating the evolution of time series. Figure 4 shows the pipeline followed for applying this framework to the diagnosis of patients with AF out of their intracardiac data, where two main components are distinguished: First, a model capable of simulating the behaviour of actual AF clinical data (1) is used to generate a dataset that reflects the variety of arrhythmias that a patient may suffer. Then, a recurrent VAE (2) is trained with the generated dataset and consequently, a latent representation that serves as a basis for creating the proposed diagnostic tool is obtained.

A. AF SIMULATION MODEL

One of the main difficulties in applying Machine Learning methods to medical problems is the data availability. It is well known that the larger and more diverse the dataset with which the model is trained, the better the learning. However, medical data are often highly sensitive and there are privacy concerns. In this case, although pacemakers registers can be collected keeping the privacy of the patients, gathering enough data that reflect the different progressions the disease may have in different people is beyond our reach. Instead, we opt to use the simulation model presented in [30].

This simulation model is based on a continuous Markov model with 3 states: "Normal", "Arrhythmia" and "False Normal" (see Figure 5). The first and second states refer to the periods of time in which the patient is in a normal state (out of arrhythmia) or suffering an episode. The "False Normal" state refers to those cases, as stated in the previous section, in which the pacemaker erroneously detects the end of an episode of AF and which subsequently leads to a change in the pacemaker's operation mode to control the arrhythmia, a process also known as Automatic Mode Switching (AMS). There are AMS events in the transitions from "Normal" to "Arrhythmia", but also in the transitions from "False Normal" to Arrhythmia.

AMS events can therefore define the beginning of a true episode of AF or the false end of an episode ("False Normal" state). The latter case is not desired but there is no simple procedure to purge these events from the IDC data in real patients [32], so the proposed generative model should produce these spurious events as well.

It will be assumed that the times in the "Normal" and "False Normal" states follow an exponential distribution with parameters $\lambda_{NA}(t)$ and $\lambda_{GA}(t)$, respectively. The time in the "Arrhythmia" state follows an exponential distribution as well, with parameter $\lambda_A(t)$. The probability that the next state after "Arrhythmia" is "False Normal", where the end of an episode is signaled before time, is p_{AG} and the probability that instead of "False Normal" it is "Normal" is $p_{AN} = 1 - p_{AG}$.

Under these conditions, the parameter $\lambda_{NA}(t)$ determines the distribution of the time between two episodes and the parameter $\lambda_A(t)$ determines the duration of an episode. The progression from paroxysmal to permanent AF is measured by the rate of change in these two parameters: the time

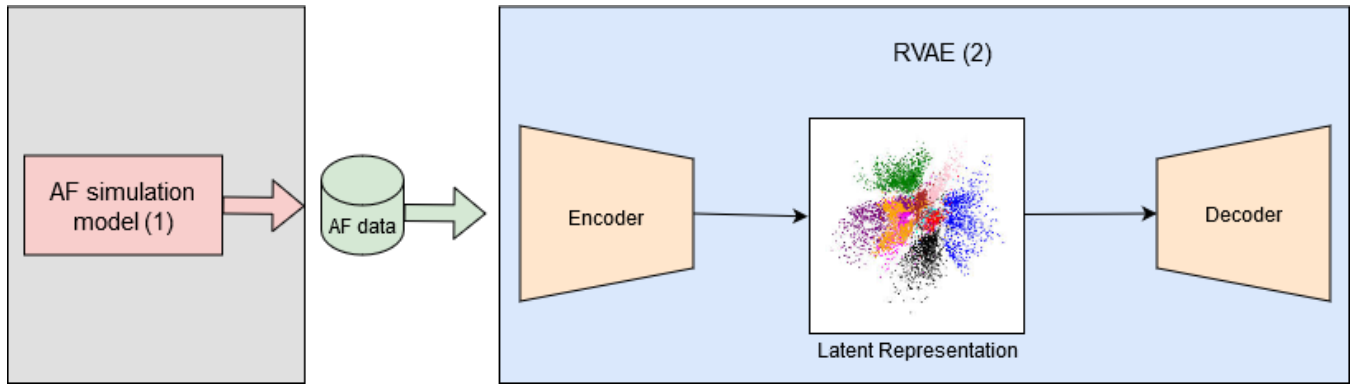


FIGURE 4: Pipeline of the proposed solution. A simulation model is used to generate synthetic arrhythmias that reflect different stages of Atrial Fibrillation. These data are fed to the proposed recurrent VAE so that it learns a representation that will later be used to evaluate the condition of new patients.

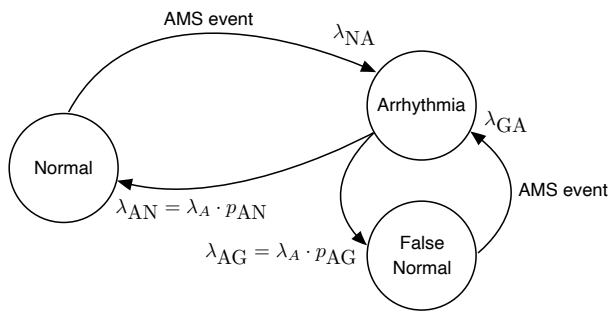


FIGURE 5: State diagram of the dynamical model of the simulation of AF episodes.

between episodes will be shorter and their duration longer as the heart condition worsens. The rate of progression is modeled by a parameter $\alpha \in [0, 1]$,

$$\lambda_{NA}(t) = \lambda_{NA}(0) \cdot \alpha^t, \quad (1)$$

$$\lambda_A(t) = \lambda_A(0) \cdot \alpha^{-t}, \quad (2)$$

where $\alpha = 1$ denotes a stable patient while α values less than 1 evidence patients with a quick progression to permanent arrhythmia.

To sum up, the proposed generative model is a Markov model in continuous time characterized by 5 parameters: $(\lambda_{NA}(0), \lambda_{GA}, \lambda_A(0), p_{AG}, \alpha)$. With this model, it is possible to produce a list of AMS events through Monte-Carlo simulation by using a random seed. Each randomly generated list can be viewed as a hypothetical patient whose type of AF is defined by the above parameters. The data generated by this model will be used to create the training set for the proposed VAE.

B. RECURRENT VAE (RVAE)

The workflow followed in this component is quite simple: a VAE is trained with the generated dataset to learn a simplified representation of the data. Thus, the learned encoder acts as

a feature extractor that describes the input data according to its properties, which are different stages of AF. This section explains how this extraction, reflected in the resulting latent space, can be leveraged to create the diagnostic map we are pursuing. It also emphasizes the recurrent architecture proposed to deal with time series as well as how the presence of a classifier built over the frozen weights of the encoder in the training process can influence the final solution.

1) Encoder as a feature extractor

In a VAE the training is regularised to avoid overfitting and to ensure that the latent space has good properties that allow the generative process. Precisely these properties contribute to the input data being mapped in the latent space in such a way that similar data are nearby and that this representation can be used as a feature extractor.

A VAE, given an input, tries to find a latent vector that is capable of describing it and at the same time has the instructions to generate it again. The process can be described as: $p(x) = \int p(x|z)p(z)dz$. Given that the integral of this formula is intractable due to the continuous domain of z , the variational inference is needed via the lower bound of the log likelihood, \mathcal{L}_{vae} ,

$$\mathcal{L}_{vae} = E_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})). \quad (3)$$

The first term is the reconstruction of \mathbf{x} that tends to make the coding-decoding scheme as efficient as possible by maximizing the log-likelihood $\log p_\theta(\mathbf{x}|\mathbf{z})$ with sampling from $q_\phi(\mathbf{z}|\mathbf{x})$, modeled by a neural network whose output are the parameters of a multivariate Gaussian: a mean and a diagonal covariance matrix. The second term tends to regularise the organisation of latent space by causing the distributions returned by the encoder to approach a standard normal. It regularises the latent variables (represented by \mathbf{z}) by minimising the KL divergence between the variational approximation and the prior distribution of \mathbf{z} . The encoder, represented by $q_\phi(\mathbf{z}|\mathbf{x})$ is the component that will be used as a feature extractor since its goal is to map the input data into a lower dimensional space.

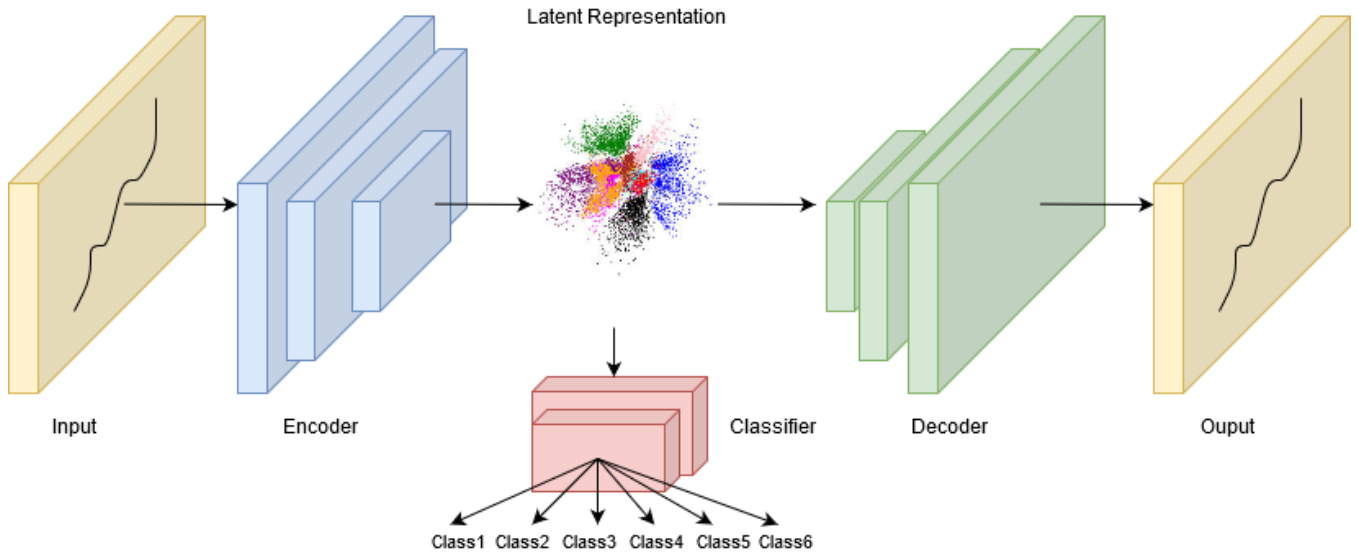


FIGURE 6: Network structure of the proposed method. The blue and green blocks are the encoder and decoder respectively and the red blocks refer to the linear classifier.

One of the advantages Autoencoders have is the flexibility provided by their architecture. The temporal nature of the data suggests that the VAE can be combined with time series modeling approaches such as RNNs. Among the different types of RNN that can be found, LSTM networks are the most outstanding ones. LSTM networks process data from back to front preserving the information from the past through the hidden states. Nevertheless, it is also possible to preserve information from the future by processing data from front to back. This property is the operating principle of Bidirectional LSTMs: they run in two directions, from past to future first, and then from future to past by preserving information from both periods. This is very valuable due to the fact that the network is aware of how the data may look like in its future stages, so it can help to understand what kind of information to predict (different progressions of AF).

With that being said, we decide to replace the encoder of a vanilla VAE with a Bidirectional LSTM network. In this way, the encoder approximates the Gaussian distribution $p_{\theta}(\mathbf{z})$ by feeding the output into two linear modules to estimate its mean and covariance. The compression of the input data results in a two-dimensional latent space dominated by the axis represented by the mean and the variance of the approximated distribution. It is expected that arrhythmias are grouped in different clusters according to their features, depicting a simpler representation of their nature.

Based on the representation learned by the encoder, the data, x , is sampled from the conditional probability distribution $p(x|\mathbf{z})$. For generative purposes, this regularisation in the latent space is very effective for easy random sampling and interpolation for the creation of new data. This is the objective of the decoder and is the most extended application of VAEs in the literature. Yet, we decide to discard this part after training the model because our efforts are focused on

the diagnosis of the input data instead of the generation of new unseen cases.

2) Diagnostic map

The diagnostic tool introduced in this study is a color-coded map that displays the actual state of the patient and the speed of change in his/her condition from paroxysmal to permanent AF. Once the VAE is trained with the Monte-Carlo simulation of pacemaker events, a topological map is obtained in the latent space from which evident clusters corresponding to different types of arrhythmias are identified, as can be seen in Figure 7 (part right). In this respect, the map can be regarded as a graphic projection of the pacemaker data in a space whose coordinates are the values of λ_{NA} , λ_{GA} , λ_A , p_{AG} and α . The values of λ_{NA} , λ_A and α in the projection measure the condition of the patient and the progression of the AF. λ_{GA} and p_{AG} measure the chance that an AMS event in the pacemaker is spurious. When actual pacemaker registers are used as input, the encoder will place them according to their features, giving information about what type of arrhythmias the patient suffers depending on which cluster they fall into. The following section provides further details on the interpretation of this map.

3) Classifier: encoder quantitative diagnosis

In order to have a clustering-like approach from the latent representation that the encoder learns we propose the inclusion of a classifier in the model training process. A similar approach was taken in [33] by Kingma et al. in what they refer to as the latent-feature discriminative model. The authors train a VAE and then feed a classifier with the outputs obtained from the resulting encoder. That is to say, a classifier is used to enhance the benefits of the VAE; however, what we propose is to enhance the latent clustering properties of VAE

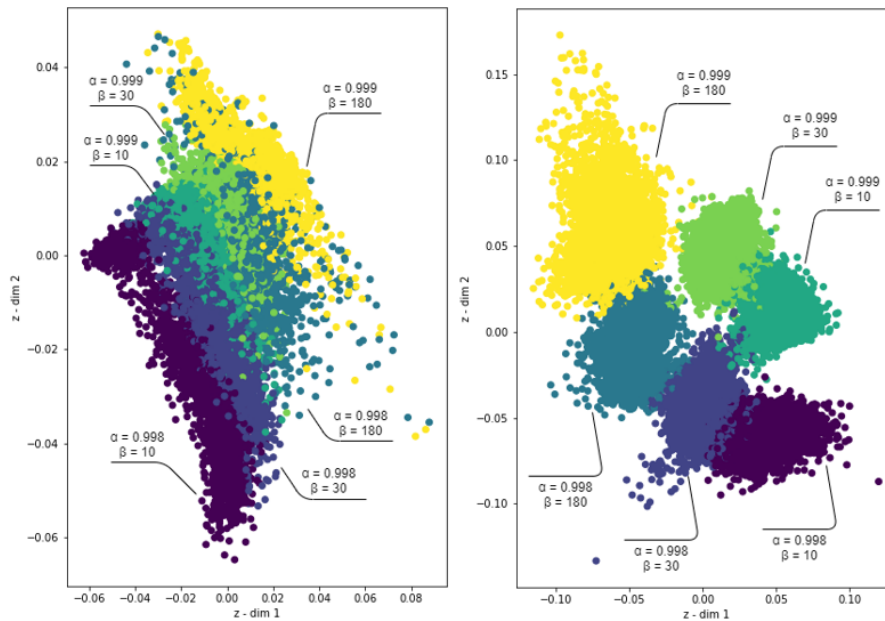


FIGURE 7: Latent representation learned by the encoder following two different training approaches. In the left figure no restriction was added to the model while in the right one a penalty for misclassifications was included.

by using a classifier, not once the VAE is trained, but while it is being trained.

The image on the left in Figure 7 represents why the first approach is not suitable for the problem at hand. It corresponds to the latent space the encoder learns after training the model without any restrictions, therefore the regularisation of the latent space for the generation of new data is prioritized. This results in the location of the input data in areas where instances that do not belong to the same group of parameters with which they were trained, are located nearby and in many cases overlap. A classifier was trained over the frozen weights of the encoder, nevertheless, this overlap severely penalizes the performance in classification.

Instead, we decided to take another path: including the optimization of the classifier in the training process. Although few previous works have taken this approach the results are very promising [34]. Therefore, unlike what is proposed in other works applied to VAEs where the problem is divided into two steps: an unsupervised pre-training step (VAE), followed by a supervised learning step (classifier), we decide to merge both. The VAE is trained to minimise a loss function composed of two objectives:

$$\mathcal{L}_{vae} + \mathcal{L}_{cross-entropy}(y, \hat{y}) \quad (4)$$

The first objective corresponds to the VAE objective itself and the second one is the categorical cross-entropy between the labels and predictions for measuring the performance of the classifier. By including the optimization of the classifier in the loss function a restriction is added to the VAE because it will strive not only for a continuous latent space but also for a space where the different classes are separated enough

to be clearly differentiated, as can be seen in the right side of Figure 7. The architecture of the classifier is simple: A single Fully Connected layer and a softmax layer are added on top of the encoder base.

In addition to the visual diagnosis that can be offered in the map explained in the previous subsection, the classifier obtained can report explicitly which parameters modeled by the simulation model are the ones that best represent each arrhythmia that is fed to the model. Also, the fact of building a classifier will allow us to compare our method with other state-of-the-art classifiers as we will see in the following section.

Coming up with this result was not straightforward. Firstly, a good choice of the Learning Rate (LR) is necessary, and secondly, it must be taken into account that two optimizations are being made: the VAE objectives and the classification objective. This means that the contributions of each loss must be assessed. Nevertheless, the relevance of each objective is unbalanced and this causes the model representations to preferentially optimize the task with the highest individual loss. To solve this, we decided to use a penalty for misclassification by using different weights for each problem, thus, we can find the perfect balance for the objectives we pursue.

IV. ILLUSTRATIVE PROBLEM

Before addressing the diagnosis of arrhythmias, we present a generic problem to demonstrate the ability of our model. We aim to develop a solution that can analyze sequential data by presenting a visual interpretation of its nature. To this end, a dataset composed of sinusoidal sequences will be used as an illustrative example of what can be obtained with our framework. Thus, we have a dataset whose nature has

a periodic factor and relies on three parameters: frequency, amplitude and phase. We generate six classes by varying the frequencies in $[1.0, 4.0]$, amplitudes in $[0.125, 0.5]$, and random phases between $[-\pi, \pi]$.

The aim of training the model with these data is to obtain a representation that is capable of dividing the six classes into different groups and at the same time keeping a coherence between the distances of the different clusters. That is, if a class has frequency 4 and amplitude 0.125, it is not desired that samples belonging to this class are grouped near those belonging to the class generated with frequency 1 and amplitude 0.5 because the dissimilarities are evident and what we pursue is that data be located near those that are most similar.

Figure 8 (part right) shows the resulting latent space after training the model. At first sight, there are six clusters, each one belonging to each generated class. In terms of classification, the performance is optimal since each point is classified within the group to which it belongs and the proximity between clusters, which is the feature we will use later for the diagnosis, is understandable. The 3 most external clusters belong to examples that have the same amplitude: 0.5, but different frequency: 1, 2 and 3 from most external to most internal. The 3 innermost clusters have the same frequency: 4 and 0.125, 0.25 and 0.375 as amplitudes from innermost to outermost. This arrangement shows that the most similar classes are adjacent on the map, which may let a new point be located in the area of the map that best fits its parameters. Again, it is important to highlight the influence of Bayesian Inference in the display of the clusters since other clustering methods lack this property as can be seen in the left side of the figure where PCA was used. Sequences with frequency 4 and amplitude 0.125 and 0.25 are not clearly differentiated and most importantly, the outermost cluster belongs to sequences with amplitude 0.375 and frequency 4, however, according to the similarity between the data it should be placed between the blue and red clusters. In consequence, the importance of achieving a faithful representation according to the similarity between the data is appreciated.

To conclude this section, it should be noted that although the reconstruction is remarkable (see Figure 9), the addition of the classifier provokes some division between clusters. This penalizes the generative condition because if sequences are generated from "empty" latent zones, that is, where there are no points previously represented, the resulting reconstruction may not make sense. Nevertheless, the generative purpose of the framework does not fall within our objectives nor will it be used for any purpose.

V. EXPERIMENTS

The experimental validation of the proposed framework has two parts. First, synthetic data with known properties are used to compare our framework with other state-of-the-art classifiers. Second, actual patients are diagnosed and their maps are validated by a human expert.

We begin by describing the experimental setup and then introducing the numerical results. Finally, the diagnostic map

achieved is presented and the experimental validation is discussed.

A. EXPERIMENTAL SETUP

In the experiments carried out for both, the toy problem and the AF diagnosis, the datasets were composed of sequences of length 144 and one feature. The number of samples for training was 84000 and 16800 for test, which were completely balanced between the six classes that were used. On the other hand, as hyperparameter tuning is a very challenging task, we made use of Hyperopt [35], a specific library for hyperparameter optimization. Also, an accurate choice of the LR is particularly essential to improve the optimization process, therefore for this parameter we used an adaptive LR optimizer, Adam, and the Cyclic Learning Rate technique proposed in [36] to help to select the optimal LR with which to start the training. Also, to attain the best possible performance of our model we used callbacks in the Keras [37] Deep Learning library for all our experiments to relegate the training stop condition to the validation error instead of the number of epochs. These implementations led us to find the best results in terms of the functions to be optimized. All models and experiments were implemented in python and the source code to reproduce the experimental results is available in a public git repository: <https://github.com/NahuelCostaCortez/RVAE>.

B. NUMERICAL RESULTS

In this section, we demonstrate that our framework can compete with state-of-the-art classifiers for time series for the case at hand. It should be noted that the baseline methods we present do not include any representation of the data, but simply predict the class to which each sample belongs, which makes us appreciate the importance of Representation Learning as it provides a more illustrative information than just a numerical or categorical result, as we will see in the next subsection.

In regard to the data, six AF categories were generated using the model described in Section III. These classes are labelled 998na10, 998na30, 998na180, 999na10, 999na30 and 999na180. The class labels begin with the first three decimals of α , which is the speed of the progression of the AF (998 is fast, 999 is slow) while the second number in the class label is $1/\lambda_{NA}(0)$, from now β for simplicity, which is the average time between two AF episodes, measured in days (10, 30 and 180 days).

1) Baseline Methods

To evaluate the performance of our model, we used the tool provided in [38] to implement 5 baseline methods,

- Resnet: a deep Residual Network proposed by [39] composed of three residual blocks followed by a GAP layer and a final softmax classifier whose number of neurons is equal to the number of classes in a dataset.
- FCN: A Fully Convolutional Neural Network, with the architecture also proposed in [39], which consists of

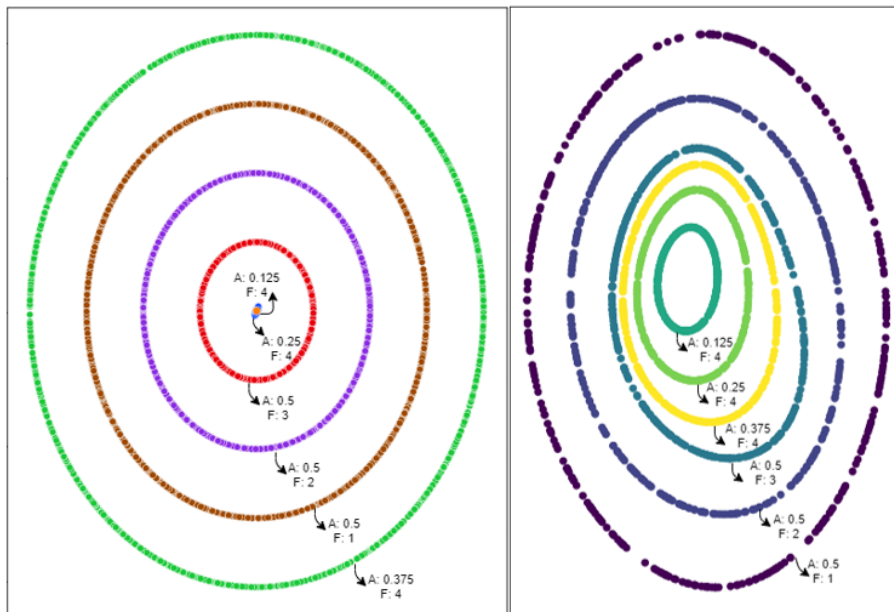


FIGURE 8: Simplified representations of sine wave sequences by PCA (left) and our model (right). It is shown that in our solution the organisation of the clusters is consistent according to the data while in PCA the organisation may not fit the nature of the data.

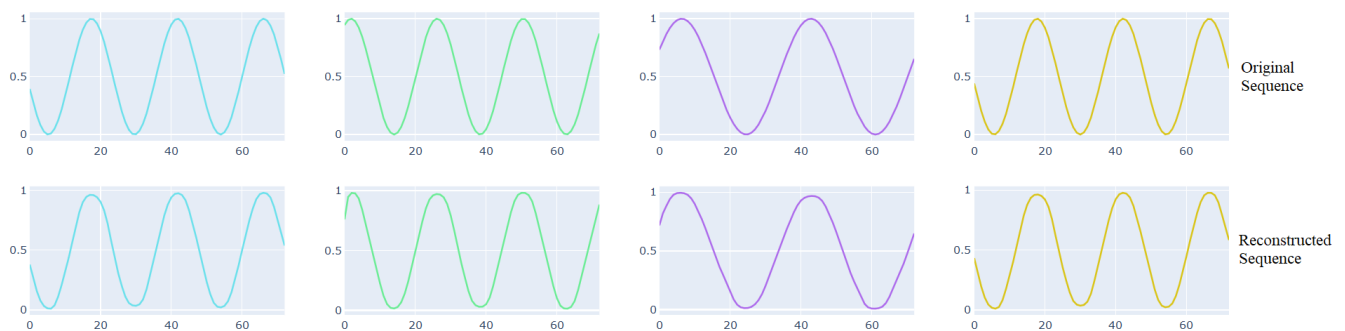


FIGURE 9: Comparison of reconstructed and original senoid samples.

three convolutional blocks whose result is averaged over the entire time dimension that corresponds to the GAP layer. Finally, a traditional softmax classifier is completely connected to the output of the GAP layer.

- Encoder: Originally proposed by [40], Encoder is a hybrid deep CNN whose architecture is inspired by FCN (Wang et al. 2017b) with a main difference where the GAP layer is replaced with an attention layer.
- TWIESN: Time Warping Invariant Echo State Network, a variant of the Echo State Networks (ESN) proposed by [41] in which each timestep is projected in a space whose dimensions are inferred from the size of a reservoir. Then for each element, a Ridge classifier is trained to predict the class of each element in the time series.

It should be noted that in the previous section the importance of RNN for time series processing was highlighted

whereas in this study TWIESN is presented as the only RNN to be compared. RNNs are generally applied for time series forecasting, however, when it comes to classification there are some drawbacks that emerge:

- This type of architecture is primarily designed to predict an output for each element in the time series [42].
- RNNs often suffer from the Vanishing Gradient problem due to long time series training [43].
- RNNs are considered difficult to train and parallelize, which leads to the avoidance of their use for computational reasons [44].

In our case the main objective is not classification but the treatment of the evolution in the series that represent the arrhythmias, which is why the application of other architectures was not considered.

Table 1 shows the performance of the different models for

each class in terms of accuracy. Each entry in the table is the number of times an arrhythmia in a class was recognized by each model for the appropriate class. In addition, to illustrate the performance of each method, the ranking calculated by the Friedman method (ranking by range) for each dataset and the resulting averaged ranking are included.

It can be seen that the best classifier is Resnet, followed by our solution, labelled as RVAE. To extend the comparison between the different methods, post-hoc tests were carried out to detect significant differences in pairs between all the classifiers as recommended in [45]. Table 2 shows the family of hypotheses formulated to compare the classifiers ordered by the corresponding p-values. If the significance test yields a p-value lower than a predefined threshold (usually 0.05), then the difference is considered significant, therefore one model is declared superior to another. In this case only Resnet is significantly higher than the other models, which are FCN, Encoder and TWIESN if a significance level of 0.05 is considered since the p-values are below this threshold. The only solution to which it does not significantly exceed is ours. If the Bonferroni correction is considered, in which the number of comparisons is taken into account, the threshold which would have to be set is 0.05 divided by the number of comparisons, i.e. $0.05/6 = 0.0083$. Taking this value, Resnet would only be significantly higher than TWIESN. This is important to note because only TWIESN and our solution use RNNs, so it can be stated that our solution outperforms the best state of the art RNN classifier.

As a conclusion of this comparative study, it can be stated that our framework is capable of competing with the best time series classifiers reported up to 2019. Besides, the misclassification errors of our model correspond to arrhythmias that by their characteristics are located between classes similar to the one that really belongs, see Figure 7 right side: the classifier learns from that representation, so it can be assumed that failures are most likely due to the overlap of instances of a similar nature, which can also be interpreted as an estimate of the class of arrhythmia that most resembles its parameters or even as the possible future evolution that they will have, as we will address next.

C. VISUAL DIAGNOSIS

As discussed in previous sections, the latent properties of the VAE were prioritized to obtain a latent space whose characteristics were suitable for a simplified representation of the data. Figure 7 (right side) is the result of the latent representations obtained by the encoder for the training data and it can be understood as a projection of the 5 parameters that govern the arrhythmias simulation model. There are six clusters, which correspond to the six classes with which the VAE was trained, labelled according to the two most relevant parameters of the simulated arrhythmias: α and β . The representations are organised according to the criticality of these two parameters.

The clinical interest lies in being able to project a real patient's data onto the latent space to find out which param-

eters of the model fit best. The procedure is quite intuitive; the arrhythmias are fed to the encoder, which predicts an output that will be the mean and variance of each one adapted to the distribution learned during the training. These two parameters are the axes that govern the latent space therefore, their codification in this space corresponds to a point with coordinates $X = \text{mean}$ and $Y = \text{variance}$. In short, each point represents an arrhythmia from the training set. By projecting these dimensions on the learned map, their location on the clusters of arrhythmias that are present will give insight about the parameters that best define the patient's condition.

Figure 10 shows a projection (red dots) of two randomly selected patients from their pacemaker records. On the left side, it can be seen that the patient's projection falls into the group belonging to arrhythmias that have parameters $\alpha = 0.999$ and $\beta = 180$. Remember, α measures the speed of progression of arrhythmias and values of α close to 0.999 indicate a slow progression of AF. β indicates the average time between arrhythmias, in this case, it is more likely that those of this patient will occur at least every 180 days, so it is estimated that this is a patient that progresses positively without involving much risk.

As the map is organised, it is evident that the values of β are located from left to right from highest to lowest (180, 30, 10), which is equal to an organisation from lowest to highest criticality as low values of β indicate short times between different episodes. On the other hand, the values of α are organised from top to bottom (999, 998), from less to more critical. This information can be used to facilitate a better interpretation of the map. The upper right zone denotes the less critical arrhythmias, while the lower right zone shows those arrhythmias that represent a very advanced stage of the disease. At the same time, the rest of the parameters of the simulation model during the generation of the training set have been varied randomly, which slightly influences the condition of the arrhythmias, therefore this property can give rise to the interpretation of arrhythmias between two clusters as an interpolation between the parameters of two classes.

This fact can be seen in the patient depicted on the right side of the same figure. This second case is located in the cluster with parameters $\alpha = 0.998$ and $\beta = 30$. Firstly, the parameter β is closer to 30, but due to its proximity to the lower-left group ($\beta = 180$), it can be understood that its evolution is on the way to reach 30, possibly a value between 180 and 30. Secondly, the most critical parameter, α , corresponds to a value of 0.998, which means that the evolution is closer to a permanent arrhythmia. This is not the most critical case, but it may need medical intervention in order to prevent future complications.

The organisation of the latent space reveals that the model is capable of setting apart the different values of α and β , allowing us to know if the condition of a certain patient evolves dangerously towards permanent AF. It is important to highlight the latent organisation obtained and its interpretability. As mentioned previously, the most dangerous arrhythmias are located on the lower right and those that do not suggest

	Accuracy				
	Resnet	FCN	Encoder	TWIESN	RVAE
998na10	0.9681(2)	0.9867 (1)	0.9361(5)	0.9517(4)	0.9543(3)
998na30	0.9664(1)	0.8788(5)	0.9456(3)	0.9438(4)	0.9553(2)
998na180	0.9846(1)	0.9719(4)	0.9729(3)	0.9611(5)	0.9779(2)
999na10	0.9849(1)	0.9849(2)	0.9364(5)	0.9505(4)	0.9770(3)
999na30	0.9879(1)	0.9778(3)	0.9826(2)	0.9791(4)	0.9733(5)
999na180	0.9904(1)	0.9886(4)	0.9895(3)	0.9786(5)	0.9895(2)
Summary Results					
Accuracy	0.9803	0.9647	0.9603	0.9607	0.9712
Average rank	1.166	3.166	3.500	4.333	2.833

TABLE 1: Accuracy of the different classifiers, 6 types of AF.

i	hypothesis	$z = (R_0 - R_i)/SE$	p
1	Resnet vs TWIESN	3.465	0.0005
2	Resnet vs Encoder	2.556	0.0106
3	Resnet vs FCN	2.191	0.0285
4	Resnet vs RVAE	1.826	0.0679
5	RVAE vs TWIESN	1.640	0.1010
6	FCN vs TWIESN	1.275	0.2023
7	Encoder vs TWIESN	0.912	0.3618
8	RVAE vs Encoder	0.731	0.4648
9	FCN vs Encoder	0.366	0.7144
10	FCN vs RVAE	0.365	0.7151

TABLE 2: Family of hypotheses ordered by p-value.

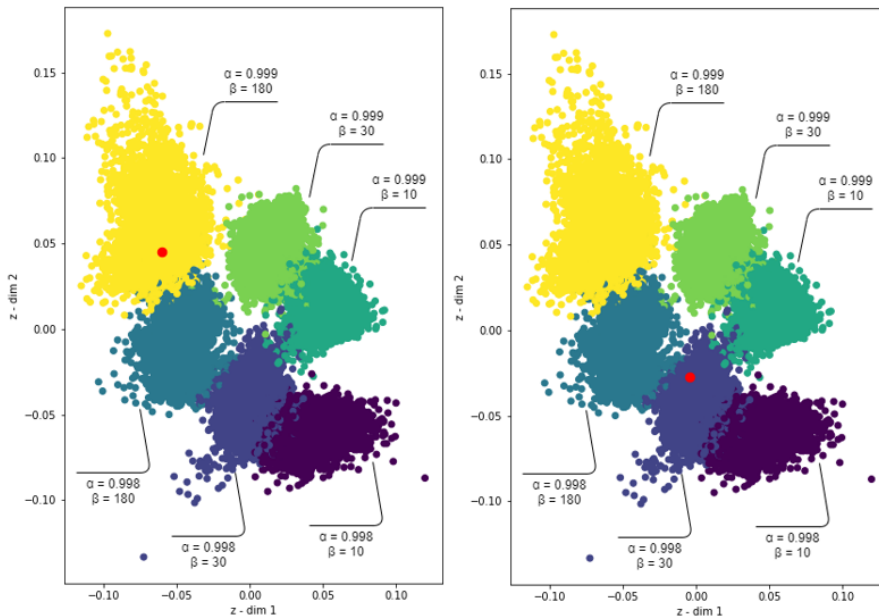


FIGURE 10: Projection of arrhythmias of actual patients.

too much danger on the upper left. This evolution from one corner to the other can be interpreted as an interpolation of the parameters used to offer a diagnosis onto the latent space: α on the Y-axis and β on the X-axis. Figure 11 reinforces this idea: new simulated arrhythmias are projected onto the latent space by varying the parameters of the simulation model, but unlike the process followed to generate the training data where the parameters λ_{GA} , $\lambda_A(t)$ and p_{AG} were randomly altered within a certain range, on this occasion they were left fixed. As a result, instances are represented as crosses and labelled according to the parameter β with which they were generated. The parameter α is omitted since the membership towards 0.998 or to 0.999 is evident. The projection on the map shows that the interpretation of the parameters of a given arrhythmia can be established according to the proximity to a specific cluster. That is, despite the fact that the first group characterizes those arrhythmias with parameters $\alpha = 0.999$ and $\beta = 10$, if an arrhythmia is located in the limit between this group and the one on its left it is very likely that it has an intermediate parameter β between both, (e.g. 20), or if an arrhythmia is located between a superior and an inferior group it would mean that the parameter α evolves dangerously towards values of 0.998. In this way, it is possible to know how the progression of a given arrhythmia could evolve.

VI. CONCLUDING REMARKS AND FUTURE WORK

We have described, trained and evaluated a recurrent VAE architecture based on Bidirectional LSTMs to assess the progression of time series by means of a graphic projection. We introduced a classifier to regularise the formation of the latent space and thus obtain a representation according to the nature of the data. The diagnosis of AF disease has been addressed with this model using intracardiac pacemaker records from actual patients and not only was an explainable diagnosis achieved but also our method was shown to be able to compete with solutions dedicated exclusively to time series classification, outperforming three of the four methods presented in terms of accuracy.

Lastly, the flexibility of the resulting model provides an opportunity to explore other future work contributions. Latent properties can be addressed in even more detail with recent architectures [46] [47] and the decoder, which has been discarded for this work, can be used for other tasks such as the detection of anomalies in data reconstruction or the prediction of the next time steps in the analysed time series.

REFERENCES

- [1] Andrew Ng et al. Sparse autoencoder. CS294A Lecture notes, 72(2011):1–19, 2011.
- [2] Salah Rifai, Grégoire Mesnil, Pascal Vincent, Xavier Muller, Yoshua Bengio, Yann Dauphin, and Xavier Glorot. Higher order contractive auto-encoder. In Joint European conference on machine learning and knowledge discovery in databases, pages 645–660. Springer, 2011.
- [3] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. Speech enhancement based on deep denoising autoencoder. In Interspeech, volume 2013, pages 436–440, 2013.

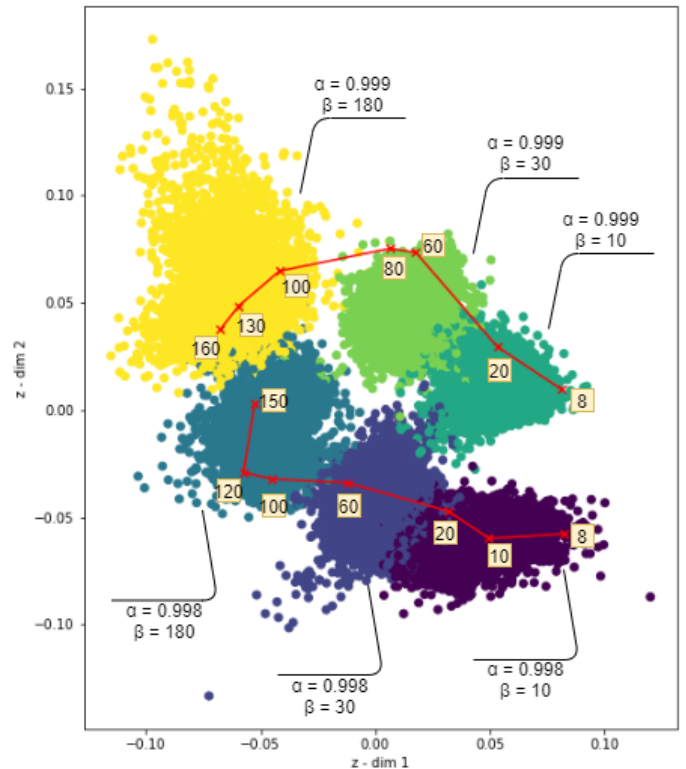


FIGURE 11: Projection of unseen simulated cases with known parameters. It can be seen that the arrhythmias are projected in the correct cluster. Besides, their location on the map is consistent according to their proximity to other clusters.

- [4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.
- [5] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. arXiv preprint arXiv:1611.02648, 2016.
- [6] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. arXiv preprint arXiv:1611.05148, 2016.
- [7] Linxiao Yang, Ngai-Man Cheung, Jiaying Li, and Jun Fang. Deep clustering by gaussian mixture variational autoencoders with graph embedding. In Proceedings of the IEEE International Conference on Computer Vision, pages 6440–6449, 2019.
- [8] Kart-Leong Lim, Xudong Jiang, and Chenyu Yi. Deep clustering with variational autoencoder. IEEE Signal Processing Letters, 27:231–235, 2020.
- [9] Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. Deep feature consistent variational autoencoder. In 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1133–1141. IEEE, 2017.
- [10] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In European Conference on Computer Vision, pages 835–851. Springer, 2016.
- [11] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In Advances in neural information processing systems, pages 658–666, 2016.
- [12] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349, 2015.
- [13] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin,

- Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [14] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. *arXiv preprint arXiv:1702.08139*, 2017.
- [15] Otto Fabius and Joost R van Amersfoort. Variational recurrent autoencoders. *arXiv preprint arXiv:1412.6581*, 2014.
- [16] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.
- [17] S. Suh, D. H. Chae, H. Kang, and S. Choi. Echo-state conditional variational autoencoder for anomaly detection. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1015–1022, 2016.
- [18] Gareth J Padfield, Christian Steinberg, Janice Swampillai, Hong Qian, Stuart J Connolly, Paul Dorian, Martin S Green, Karin H Humphries, George J Klein, Robert Sheldon, et al. Progression of paroxysmal to persistent atrial fibrillation: 10-year follow-up in the Canadian registry of atrial fibrillation. *Heart rhythm*, 14(6):801–807, 2017.
- [19] Pijush Kanti Dutta Pramanik, Bijoy Kumar Upadhyaya, Saurabh Pal, and Tanmoy Pal. Chapter 1 - internet of things, smart sensors, and pervasive systems: Enabling connected and pervasive healthcare. In Nilanjan Dey, Amira S. Ashour, Chintan Bhatt, and Simon [James Fong], editors, *Healthcare Data Analytics and Management, Advances in ubiquitous sensing applications for healthcare*, pages 1 – 58. Academic Press, 2019.
- [20] Agam Bansal and Rajnish Joshi. Portable out-of-hospital electrocardiography: A review of current technologies. *Journal of Arrhythmia*, 34(2):129–138, 2018.
- [21] Mintu P. Turakhia, Manisha Desai, Haley Hedlin, Amol Rajmane, Nisha Talati, Todd Ferris, Sumbul Desai, Divya Nag, Mithun Patel, Peter Kowey, John S. Rumsfeld, Andrea M. Russo, Mellanie True Hills, Christopher B. Granger, Kenneth W. Mahaffey, and Marco V. Perez. Rationale and design of a large-scale, app-based study to identify cardiac arrhythmias using a smartwatch: The apple heart study. *American Heart Journal*, 207:66 – 75, 2019.
- [22] David Calvo, José Rubín, Diego Pérez, and José Jalife. Spectral analysis of electrograms in a substrate modified by radiofrequency ablation reveals similarities between organized and disorganized atrial rhythms. *Heart rhythm*, 11(12):2306–2309, 2014.
- [23] G Neal Kay, Kenneth A Ellenbogen, Michael Giudici, Margaret M Redfield, Louise S Jenkins, Marcus Mianulli, and Bruce Wilkoff. The ablate and pace trial: a prospective study of catheter ablation of the av conduction system and permanent pacemaker implantation for treatment of atrial fibrillation. *Journal of Interventional Cardiac Electrophysiology*, 2(2):121–135, 1998.
- [24] Charles D Swerdlow, Gautham Kalahasty, and Kenneth A Ellenbogen. Implantable cardiac defibrillator lead failure and management. *Journal of the American College of Cardiology*, 67(11):1358–1368, 2016.
- [25] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.
- [26] MR Avendi, Arash Kheradvar, and Hamid Jafarkhani. A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac mri. *Medical image analysis*, 30:108–119, 2016.
- [27] Jae-Hong Eom, Sung-Chun Kim, and Byoung-Tak Zhang. Aptacdss-e: A classifier ensemble-based clinical decision support system for cardiovascular disease level prediction. *Expert Systems with Applications*, 34(4):2465–2479, 2008.
- [28] Bahareh Pourbabaee, Mehrsan Javan Roshtkhari, and Khashayar Khorasani. Deep convolutional neural networks and learning ecg features for screening paroxysmal atrial fibrillation patients. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(12):2095–2104, 2018.
- [29] Supreeth Prajwal Shashikumar, Amit J Shah, Qiao Li, Gari D Clifford, and Shamim Nemati. A deep learning approach to monitoring and detecting atrial fibrillation using wearable technology. In *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 141–144. IEEE, 2017.
- [30] Nahuel Costa, Jesús Fernández, Inés Couso, and Luciano Sánchez. Graphical analysis of the progression of atrial arrhythmia using recurrent neural networks. *International Journal of Computational Intelligence Systems*, 13(1):1567–1577, 2020.
- [31] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [32] Jesús Fernández, Julián Velasco, and Luciano Sánchez. Detection of cardiac arrhythmias through singular spectrum analysis of a time-distorted egm signal. In Hilde Pérez García, Javier Alfonso-Cendón, Lidia Sánchez González, Héctor Quintián, and Emilio Corchado, editors, *International Joint Conference SOCO’17-CISIS’17-ICEUTE’17 León, Spain, September 6–8, 2017, Proceeding*, pages 137–146. Cham, 2018. Springer International Publishing.
- [33] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.
- [34] Felix Berkhahn, Richard Keys, Wajih Ouertani, Nikhil Shetty, and Dominik Geißler. Augmenting variational autoencoders with sparse labels: A unified framework for unsupervised, semi-(un) supervised, and supervised learning. *arXiv preprint arXiv:1908.03015*, 2019.
- [35] James Bergstra, Dan Yamins, and David D Cox. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference*, volume 13, page 20. Citeseer, 2013.
- [36] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017.
- [37] François Chollet et al. keras, 2015.
- [38] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.
- [39] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.
- [40] Joan Serrà, Santiago Pascual, and Alexandros Karatzoglou. Towards a universal neural network encoder for time series. In *CCIA*, pages 120–129, 2018.
- [41] Pattreeya Tanisaro and Gunther Heidemann. Time series classification using time warping invariant echo state networks. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 831–836. IEEE, 2016.
- [42] Martin Långkvist, Lars Karlsson, and Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.
- [43] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *CoRR*, abs/1211.5063, 2:417, 2012.
- [44] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.
- [45] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.
- [46] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.
- [47] Karol Gregor, George Papamakarios, Frederic Besse, Lars Buesing, and Theophane Weber. Temporal difference variational auto-encoder. *arXiv preprint arXiv:1806.03107*, 2018.



NAHUEL COSTA received the B.Sc. degree in Computer Engineering in Information Technology from the University of Oviedo, Spain, in 2019, and the M.S.c. degree in Computer Science from the University of Oviedo, Spain, in 2021. He is currently working as a researcher and pursuing a Ph.D. in Artificial Intelligence at the Computer Science Department of the University of Oviedo.



LUCIANO SÁNCHEZ (M'07-SM'15) received M.Sc. and Ph.D. degrees in Electronic Engineering from the University of Oviedo, Spain, in 1991 and 1994, respectively. He is currently a full professor at the Department of Computer Science, University of Oviedo. He is the author of more than 80 international journal and more than 100 conference papers and book chapters. His research goals include the theoretical study of algorithms for mathematical modelling and intelligent data

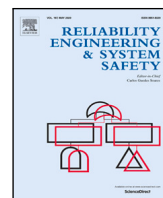
analysis, and the application of these techniques to practical problems of industrial modelling, signal processing and condition monitoring, with special interest in the study of low quality data and fuzzy information. IEEE Outstanding Paper Award in 2013 IEEE International Conference on Fuzzy Systems (Hyderabad, India). 2013 Rolls-Royce Deutschland Engineering Innovationspreis (Berlin, Germany).



INÉS COUSO (M'87) received M.Sc. and Ph.D. degrees in Mathematics from the University of Oviedo, Spain, in 1995 and 1999, respectively. She is currently a full professor in the area of Statistics and Operations Research at the same university. She has been a visiting professor at several institutes and laboratories, such as the Institut de Recherche en Informatique de Toulouse (Université Paul Sabatier, Toulouse, France) and the "Laboratoire de Informatique, de Robotique et

de Microélectronique de Montpellier" (Université Montpellier II). In 2008 she was "Professeuse invitée 1er classe" at the CNRS (France) and in 2011 she was "Professeuse invitée" at the Université Montpellier 2 (France). In 2015 she was named Excellence Advanced Researcher by the Centre International de Mathématique et d'Informatique de Toulouse (CIMI). She has written 80 articles in peer-reviewed journals and more than 100 book chapters and communications to national and international conferences. She is currently Senior Editor of the International Journal of Approximate Reasoning and Area Editor of the journal Fuzzy Sets and Systems.

...



Variational encoding approach for interpretable assessment of remaining useful life estimation

Nahuel Costa^{*,1}, Luciano Sánchez¹

Computer Science Department, University of Oviedo, Gijón, 33202, Asturias, Spain

ARTICLE INFO

MSC:
0000
1111

Keywords:

Remaining useful life
Prognostics and health management
Interpretability
Variational inference
Recurrent neural networks

ABSTRACT

A new method for evaluating aircraft engine monitoring data is proposed. Commonly, prognostics and health management systems use knowledge of the degradation processes of certain engine components together with professional expert opinion to predict the Remaining Useful Life (RUL). New data-driven approaches have emerged to provide accurate diagnostics without relying on such costly processes. However, most of them lack an explanatory component to understand model learning and/or the nature of the data. To overcome this gap we propose a novel approach based on variational encoding. The model consists of a recurrent encoder and a regression model: the encoder learns to compress the input data to a latent space that serves as a basis to build a self-explanatory map that can visually evaluate the rate of deterioration of aircraft engines. Obtaining such a latent space is regularized by a new cost function guided by variational inference and a term that penalizes prediction errors. Consequently, not only an interpretable assessment is achieved but also a remarkable prognostic accuracy, outperforming most of the state-of-the-art approaches on the popular simulation dataset C-MAPSS from NASA. In addition, we demonstrate the application of our method in a real-world scenario with data from actual Turbofan engines.

1. Introduction

Prognostic technologies are crucial in any physical system. In aircraft engines this is a must since throughout their life cycle they are subjected to different conditions that cause degradation and ultimately lead to failure. For this reason, data is routinely collected from various built-in sensors to monitor performance and avoid operating in undesirable conditions. Over the years, the amount of information collected has increased and this has paved the way for making more complex analyses in favor of maintenance that extends the useful life of these systems. However, traditional strategies such as scheduled preventive maintenance or corrective maintenance of failures [1] are increasingly unable to meet growing industrial demand in terms of efficiency and reliability. In this regard, Prognostics and Health Management (PHM) technologies are proving to have promising capabilities for application in industries [2]. As a result, metrics like remaining useful life (RUL) of systems have been established as key elements to improve maintenance schedules and avoid engineering, safety and reliability failures. Consequently, this would make it possible to determine engine deterioration, increase engine flight time and reduce maintenance costs.

1.1. Literature review

In the last decade, several techniques have been proposed to model the degradation of these complex systems, from which two currents arise: model-based approaches and data-driven approaches. Among the former, works such as [3,4] stand out, although these techniques require extensive prior knowledge about the physical systems, information that is often not available in practice. Precisely for this reason, data-driven approaches have become so popular in recent years, as they are able to model degradation features based purely on historical records from which the underlying causalities and correlations can be modeled. That is, knowledge can be inferred from sensor data to predict valuable system information such as RUL [5].

Especially, the use of Machine Learning models has had a great impact given that they are able to model highly nonlinear, complex and multi-dimensional systems with little prior prognostic experience. If we focus on RUL estimation, initial work was oriented towards the application of multi-layer perceptrons (MLP) as in [6], where the authors reported higher prediction results than model-based approaches. In [7,8] both diagnostics and prognostics were approached with PCA and hidden Markov models. Over the years, other techniques have

* Corresponding author.

E-mail address: costanahuel@uniovi.es (N. Costa).

¹ All authors have participated equally in all tasks.

been also explored: some researchers have integrated fuzzy logic to capture more information for Engine Health Monitoring (EHM) [9,10], others applied Support Vectors [11] or Gradient Boosting trees [12]. Nevertheless, despite being all of them considered relevant work for the sake of RUL estimation, the greater impact has undoubtedly been produced by the use of Deep Learning models [13]. This is due to the fact that the raw data obtained from machine health monitoring share a high dimensionality, similar to that of other problems in which these models have had a significant impact and are known to perform remarkably well, especially in Computer Vision and Natural Language Processing (NLP).

Certainly, RUL estimation is a hot-topic, partly thanks to the application of these new deep models where two trends clearly stand out: the use of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). High-impact work can be easily found in both directions, calling for the use of CNNs for feature extraction [14–16] and recurrent networks for modeling the temporal nature of the data [17–19]. From there, promising modifications have been proposed [20–22], where both architectures are combined for better prediction capabilities. In addition, approaches that go beyond RUL estimation are beginning to emerge such as [23] where a semi-supervised method is developed to avoid relying on data labeling or [24] where the authors present a model to mine different levels of degradation trends.

However, there is a clear gap between all Deep Learning oriented approaches: although they do achieve remarkable results, models are treated as black boxes where inputs are used to obtain some output, in this case, the RUL. It is challenging to find algorithms that go beyond providing good numerical performance and this is vitally important. Despite the fact that the current is to dispense with prior knowledge about the system to be monitored, in the end, these models are designed to be used by people outside academia. Therefore, it is of great interest to be able to provide a tool that gives certain interpretability of the models' decisions as well as some insights about the nature of the data. In fact, these are attributes of particular interest, if not demanded, for decision making in safety-critical applications [25].

1.2. Suitable approaches and limitations

To meet the goals stated one can think of unsupervised learning techniques as a possible way to approach this. Especially, when it comes to reveal insights about the nature of the data, Representation Learning approaches such as autoencoders come in handy. Autoencoders are models designed to reconstruct the input data while learning a compressed representation of it, the so-called latent space. Their applications are quite widespread in anomaly diagnosis, being the most common case that in which the probability distribution of non-anomalous data is learned in order to detect, through reconstruction errors, patterns that do not correspond to that distribution [26–28].

The key element of these models is that their performance is based directly on internal representations, which in turn can be used to better understand the problem itself. Accordingly, they have been used to identify anomalous elements within a set of systems with similar characteristics, such as fleets of vehicles or aircraft engines [29,30]. In these cases, the compression capacity of the autoencoder is exploited, thus enhancing the interpretability of the latent space: assuming that anomalies are infrequent, those points in the latent space furthest away from the most populated clusters can be identified as such. However, the main limitation is that, unlike other dimensionality reduction methods like PCA [31,32], the relative distances between the input patterns are not necessarily preserved in the projection of the encoder, therefore this cluster analysis is not always possible. This problem was solved with Variational Autoencoders (VAEs). In VAEs, variational inference is added to the error function through the Kullback–Leibler term, which guarantees that data with similar patterns will also be encoded nearby in the latent space. VAEs are a recent but well-known alternative with numerous applications in anomaly analysis [33–35].

The problem of determining the RUL of a system, on the other hand, has been studied in less depth. This problem has many points in common with the diagnosis of anomalies and has also been solved by using autoencoders [36,37]. Despite these similarities, both problems have a fundamental difference: in anomaly diagnosis, the aim is to look for individuals in unlikely areas of the latent space. In RUL prediction, on the contrary, the objective for a complete and interpretable diagnosis should be to project the evolution of the system in the latent space over time in order to know how fast it is moving towards anomalous zones. The presence of anomalies is indeed correlated with the RUL since anomalous latent states usually correspond to low RUL values. However, two systems can be in the same initial condition but have a different temporal evolution, so that the successive states of the system cannot be studied independently as is done in anomaly detection. Instead, RUL estimation must be linked to the temporal analysis of complete state trajectories in the latent space.

In this line of research, we have recently proposed a new VAE architecture where the input and output layers are recurrent [38], as VAE applications are mainly oriented to the image domain and not so much to time series data, which is the case of RUL estimation. This architecture allows obtaining projections of state sequences and solves to some extent the problem of applying VAEs to RUL estimation since variational inference guarantees that systems with similar degradation patterns are going to be projected in close areas of the latent space. The recurrent VAE thus allows differentiation of systems with anomalous trajectories, however, this method is not a complete solution to the RUL estimation problem, mainly for two reasons:

1. It does not produce a numerical estimation of the system lifetime. It only separates low RUL systems from high RUL systems, but does not quantify what the RUL value is at each time step.
2. There is no guarantee that the time evolution of the trajectory projections are correctly separated (see Fig. 1), so it does not provide a solution to the problems of fleet health prognosis.

Concerning the second reason, it should be noted that RUL estimation, in real-world cases, is an online process: the useful life of each system is continuously updated as new data is received. For this reason, it is not enough that the new points are located in the vicinity of the previous ones: the successive projections of each system in the latent space, as time progresses, must form a continuous trajectory, which can be extrapolated into the future. In this way, it will be possible to diagnose continuous degradations over time (such as wear, efficiency losses, etc.) that affect the RUL, but which do not correspond to occasional events and therefore cannot be identified by anomaly detection analysis.

In this study, we solve the two open problems mentioned above by the combined use of a new neural architecture based on a recurrent variational encoder and a fresh way of regularizing training. To this end, we propose a new cost function related to the association of the Kullback–Leibler term with a second term that favors that the projections of successive states of the engines in the latent space constitute a continuous trajectory. This second term, as will be further explained, penalizes the successive RUL prediction errors over time, having a positive influence both on the ability of the new network to predict the lifetime of the engines and on the quality of the latent space. Thus, we take full advantage of the use of novel recurrent network architectures without giving up Representation Learning properties due to the construction of a latent space with suitable properties to provide a visual, hence explainable and interpretable diagnosis. The method is first validated with the popular C-MAPSS dataset from NASA and subsequently tested on a real environment.

The structure of this paper is organized as follows: Section 2 introduces the settings carried out to approach this problem. A detailed description of the proposed method for achieving an explainable diagnosis of aircraft engines is described in Section 3. The experimental set-up is explained in Section 4. Experimental results concerning both a benchmark problem and a real-world problem are presented in Section 5 while conclusions are drawn in Section 6.

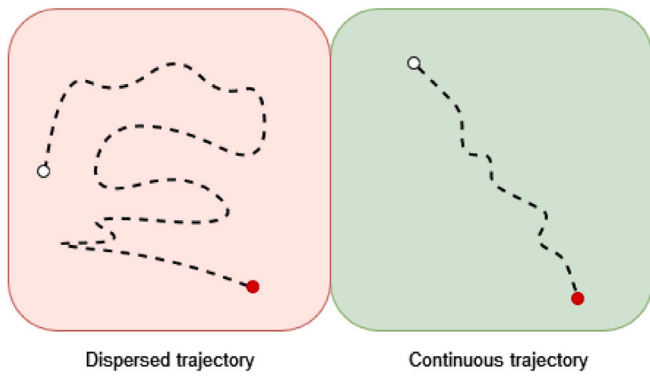


Fig. 1. In a vanilla VAE, training is regularized to prioritize generative purposes. This results in a dispersed latent projection of the system trajectory as in the figure on the left, in which there is no clear evolution between the state of the equipment at the beginning (white dot) and at the end of evaluation (red dot). On the contrary, a projection like the one in the figure on the right is what we are aiming for.

2. Problem settings

Before delving into the details of the model and results, it is of particular interest to highlight some important issues which are of great impact in achieving optimal performance and will help to better understand the problem itself.

2.1. RUL estimation

RUL stands for Remaining Useful Life and is a popular metric in prognostics, especially in aircraft monitoring [39]. Normally, sensors such as turbine pressure or compressor temperature are used to collect flight information about the engine. This data form a multi-valued series. The j th element of the series is a vector of h elements, each of which is the reading of one of the available sensors taken at the j th time instant. Having this information for several engines, a dataset could be formed from which to train a model to estimate the number of remaining time cycles in which a new unseen aircraft works well before failure, i.e its RUL.

In this paper, the proposed method is evaluated on the popular NASA’s engine degradation dataset [40], known as C-MAPPS. Although it was published several years ago, it is still relevant today, (perhaps motivated by the fact that there are hardly any other similar datasets in the field), being the standard problem on which to test new RUL estimation models. This dataset contains simulated data of Turbofan engines produced by Commercial Modular Aero-Propulsion System Simulation (C-MAPSS), a model-based simulation program. It is composed of multivariate temporal data obtained from twenty-one sensors and is further divided into 4 sub-datasets. As can be seen in Table 1, in each sub-dataset a training set and a test set are provided, from which there is a slight difference. The training set comprises run-to-failure data. That is, although each engine unit starts with different degradation states that are unknown, these are considered healthy and as time progresses, the engine units degrade to failure, therefore the last data sample corresponds to the time cycle in which the engine unit is declared unhealthy (RUL = 0). On the contrary, sensor records in the test sets terminate at some point before system failure and the actual RUL value for these engines is provided. The aim of this problem is to estimate the RUL of each engine in the test sets. It should be noted that training on a particular sub-dataset might be not applicable on another sub-dataset because the operating and failure conditions are different. There are promising approaches such as [41] in which adaptive methods are adopted to avoid these differences between training and test sets and thus avoid offline training. However, this is out of the scope of this work. Since there are four different sub-datasets, we train our model on each training set and evaluate on the test sets as they have exactly the same conditions.

Table 1

Data sets details.	FD001	FD002	FD003	FD004
Train trajectories	100	260	100	249
Test trajectories	100	259	100	248
Operating conditions	1	6	1	6
Fault conditions	1	1	2	2

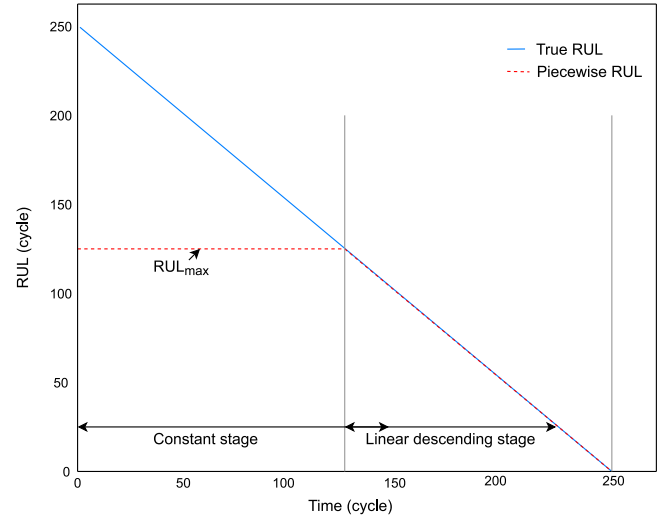


Fig. 2. RUL target function.

2.1.1. RUL target function

In prognostic problems, as the system always tends to deteriorate, it is quite usual to assume degradation behavior. Thus, a target RUL can be constructed based on these assumptions to guide the model training and enhance its predictions in a supervised manner. The most naive approach would be to assume that RUL decreases linearly over time, however, when analyzing the sensor signals, there is a common pattern: many sensors seem rather constant at the beginning until a breakpoint occurs that makes the engine degrade linearly with usage. The piecewise linear degradation model proposed in [42] follows this idea and is the most extended target function used in the literature. It simply limits the maximum value of the RUL function as illustrated in 2. We use this degradation model to obtain the RUL label with respect to each training sample at each time-step. The maximum RUL is set at 125 cycles. This is used to make fair comparisons with respect to other models that used the same methodology, but it should be noted that this is just a guideline value. Different equipment in the system has different lifetimes and different degradation trajectories, therefore this value may be too high or too low for different individuals. In [43] the authors propose a new methodology to construct the target RUL for each individual in order not to rely on a single value. However, there is still no consensus on the best way to teach the algorithm the behavior of the system. Precisely, this can be considered a bottleneck and that is why it is desirable to provide learning that does not depend exclusively on this function. In this work, we learn the nature of the data in an unsupervised manner with variational inference and fine tune it with the labels to improve predictions. Thereby, what the model learns is guided more by the nature of the data than by the labels themselves.

2.2. Metrics

In order to establish a fair comparison with the rest of the approaches the same metrics used in most similar works are chosen. On the one hand, there is the original metric proposed by NASA in PHM 2008 Data Challenge, which is described in Eq. (1), where N is the

number of engines in the test set, S is the computed score, and $d =$ (Estimated RUL - True RUL).

$$s = \sum_{i=1}^N s_i, \quad (1)$$

$$s_i = \begin{cases} e^{-\frac{d_i}{15}} - 1, & \text{for } d_i < 0 \\ e^{\frac{d_i}{10}} - 1, & \text{for } d_i \geq 0 \end{cases}$$

The main objective of this function is to differentiate late predictions from early predictions. The former are more penalized because it is understood that it is too late to perform maintenance while early predictions are not a major problem. Although maintenance resources could be wasted, priority is given to penalizing false negatives. This has some drawbacks since, if there is an outlier leading to a late prediction, this would dominate the overall performance score, thus masking the true overall accuracy of the algorithm. In addition, the level of confidence with which the algorithm is able to estimate the RUL value before the failure point is also not taken into account.

Due to these shortcomings, the use of RMSE is also proposed as it gives equal weight to early and late predictions:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (d_i)^2} \quad (2)$$

The use of RMSE together with the scoring function (Eq. (1)) would avoid favoring an algorithm that artificially lowers the score by underestimating, which is quite likely due to the reasons exposed, by resulting in a higher RMSE. In summary, both metrics complement each other by providing more information about the accuracy of the model.

3. Model

The proposed model consists of three components: an encoder network, a regression model and a latent space. The encoder learns to compress the data into the latent space so that it is described by the parameters that initialize the probability distribution to which the data belongs. Variational inference is added to the loss function through the Kullback–Leibler divergence, which measures how much one probability distribution diverges from another, to learn the above-mentioned parameters. A second term is also added to penalize wrong estimations of the regression model. In the end, all this allows a latent space to be learned in which similar data is located in nearby areas from which to efficiently perform other tasks.

The workflow followed for this problem is depicted in Fig. 3. The model is trained with data from aircraft engines to learn a simplified representation of their trajectories. Thus, the resulting encoder acts as a feature extractor compressing into the latent space the data according to their properties, which are different stages of degradation in the engines. The latent space contains the compressed representation of the aircraft, particularly in 2 dimensions. This representation will be used after training is completed to visually evaluate the degradation patterns of the engines. Finally, a numerical prediction of the RUL that best represents each engine that is fed to the model is provided by training a regression model directly with the features learned in the latent space. This section explains the main differences between our model and a VAE and how the encoder output can be leveraged to create the visual diagnosis we propose. Emphasis is placed on the implementation of the recurrent architecture for dealing with time series, as well as how latent features lead to perform RUL estimation.

3.1. Variational encoding

Variational encoding refers to the process of compressing input data based on variational inference, a key element in our research, as stated in the introduction. This process is the basis for the operation of VAEs,

therefore it is important to know how they work in order to clarify the differences with respect to our model. In a VAE the training process is regularized to avoid overfitting and to ensure that the latent space has the necessary properties that enable the generative process. To obtain them, the encoder must map the data in the latent space in such a way that similar data is close to each other. This allows the decoder not only to reconstruct the data efficiently but also to generate new instances from points in the latent space that do not correspond to the encoding of any training sample.

VAEs compress the input data into a latent vector, which is a simplified representation, described as $p(x) = p(x|z)p(z)dz$, where the domain of z is continuous and therefore intractable. For this reason variational inference is used since this intractability can be solved via the lower bound of the log-likelihood [44], \mathcal{L}_{vae} ,

$$\mathcal{L}_{vae} = E_{q_\phi(\mathbf{x}|\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) \quad (3)$$

The first term is the reconstruction of \mathbf{x} that tends to make the coding–decoding scheme as efficient as possible by maximizing the log-likelihood $\log p_\theta(\mathbf{x}|\mathbf{z})$ with sampling from $q_\phi(\mathbf{z}|\mathbf{x})$, modeled by the encoder, whose output is the parameters of a multivariate Gaussian: a mean and a diagonal covariance matrix. In other words, the main goal of the encoder is to map the input data into a lower-dimensional space, acting as a feature extractor. The second term tends to regularize the organization of the latent space by causing the distributions returned by the encoder to approach a standard normal. It regularizes the latent variables (represented by \mathbf{z}) by minimizing the KL divergence between the variational approximation and the prior distribution of \mathbf{z} .

The data is reconstructed from the conditional probability distribution $p(\mathbf{x}|\mathbf{z})$, learned by the encoder. For generative purposes, the regularization produced in the latent space facilitates random sampling and interpolation for the creation of new data. This is why VAEs are understood as generative models and their use is widespread as such.

Nevertheless, we do not strive to generate new aircraft data, but to diagnose it by making use of latent representations. VAEs latent space, in contrast, is not usually used for clustering or visualization despite it has promising properties for this. In fact, there are works in which this has been taken advantage of, as in [45] in what they refer to as the latent-feature discriminative model. The authors trained a VAE and then fed a classifier with the outputs obtained from the resulting encoder. Still, this is not further explored in the literature since VAEs are mainly oriented to generative tasks and this causes the regularization of the latent space to lead the encoder to project the data as compressed as possible, resulting in obvious overlaps.

This is a barrier to our objectives because these overlaps make it difficult to estimate the RUL. First visually: although aircraft with similar RUL values will be close on the map, they will not be clearly differentiated from those that are far away. Then, because any model built on top of this will be guided by this representation and will most likely result in prediction failures. Therefore, a vanilla VAE does not meet our needs and we must adapt the use of variational inference for our problem: what we want is to enhance the latent organizational properties of variational encoding by using a regressor, not a classifier, and not once the model is trained, but while it is being trained.

The image on the left in Fig. 4 represents why the approach mentioned in [45] is not suitable for the problem at hand. It corresponds to the latent space the encoder learns after training the model without any restrictions thence the regularization of the latent space for the generation of new data is prioritized. This causes the input data to be placed in areas where instances whose features are not similar (different RUL values) are not clearly differentiated or even overlap. As this approach suggests, a simple regressor was trained on the frozen encoder weights, however, this overlap severely penalizes the performance in RUL estimation, making it unable to compete with other state-of-the-art methods. There are promising works that propose to solve this by including regression errors in the training process as in [46], although

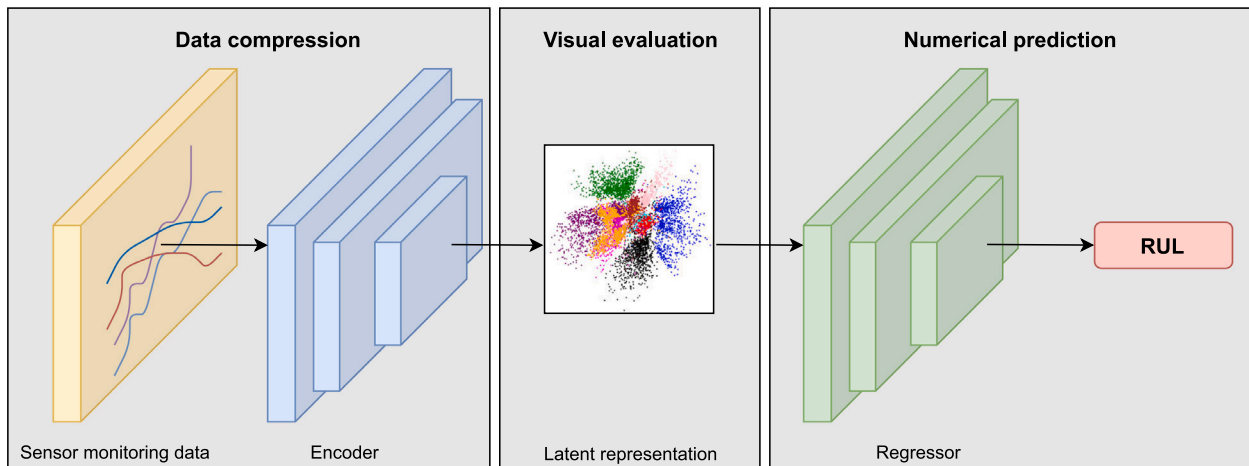


Fig. 3. Workflow followed for the proposed approach: aircraft data is fed into the encoder, which learns a latent representation based on deterioration patterns in order to build a graphical map reflecting the evolution of their trajectories. The regressor directly influences obtaining such a latent space and allows to report numerically the RUL of each engine.

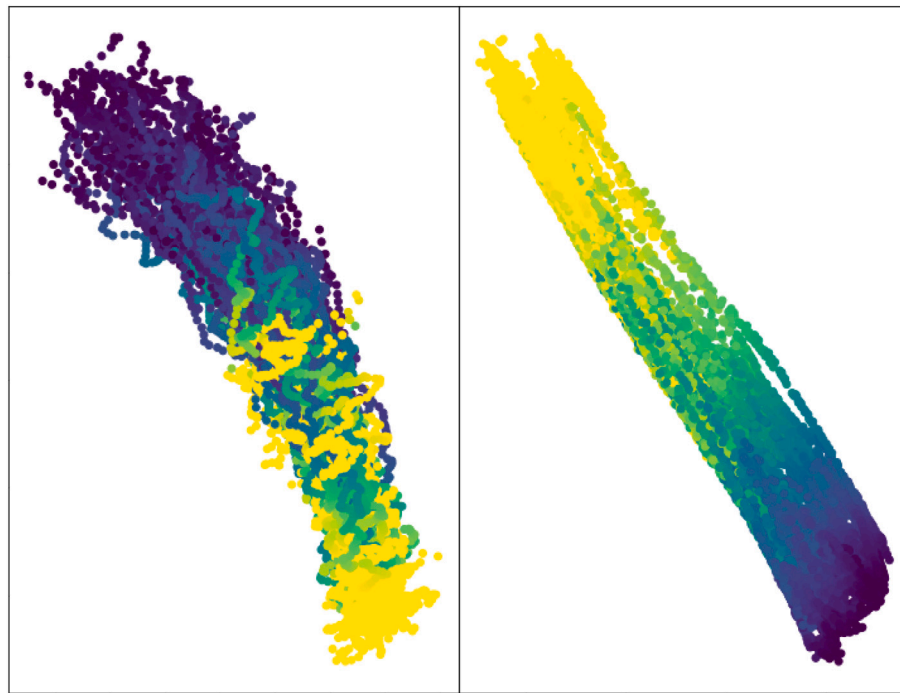


Fig. 4. Latent representations learned by the encoder for FD001. The figure on the left shows the regular training of a VAE, while the figure on the right shows the result with our model, which does not include the decoder but a regression model that adds a penalty for wrong predictions.

the decoder is still used, which may wrongly interfere with our ultimate goal: the diagnosis of the aviation history of the engines.

Instead, the path we decide to take includes the omission of the decoder to focus learning on obtaining an interpretable latent space. Thereby, the main difference with respect to a VAE is that we replace the decoder with a regression model, as shown in Fig. 3, and the training is done differently. Our proposed model is trained to minimize a loss function composed of two objectives:

$$\mathcal{L}_x = -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) + RMSE \tag{4}$$

The first objective corresponds to the regularization of the latent space through variational inference, as explained before in Eq. (3) and the second one is the Root Mean Square Error (RMSE) between the known RULs and the RULs predicted by the model. Including the regressor optimization in the loss function adds a constraint to the

model, as it will strive not only for a continuous latent space, but also for a space in which different types of trajectories, and so with different associated RUL values, are sufficiently separated to be clearly differentiated so that the evolution of degradation in an aircraft can be observed. The right part of Fig. 4 demonstrates the effectiveness of training the model in this way. The architecture of the regressor is simple: on top of the encoder base, a fully connected layer with a tanh activation function and another layer with a single neuron containing the RUL prediction are added.

As for the encoder, we decide to implement it with recurrent networks given that most recent studies make use of them to model the time complexity of historical aircraft data [47,48]. Among the different types of RNNs that can be found, LSTM networks are the most popular. These networks process data from backward to forward conserving information from the past through hidden states. However,

Bidirectional LSTM networks are in high demand because they provide not only information about the past but also about the future: data is first processed from past to future and then from future to past, thus preserving the information from both periods. This is quite helpful because the network is aware of what the data may look like in its future stages, which helps it to understand what kind of information to predict (different stages of engine deterioration). All in all, we decide to implement the model with this type of network.

In summary, the encoder, built with bidirectional LSTMs, approximates the Gaussian distribution $p_{\theta}(\mathbf{z})$ by feeding the output into two linear modules to estimate its mean and covariance. This means that the compression of the engine data by the encoder results in a two-dimensional latent space whose axes would be the mean and covariance of the approximate distribution. Consequently, the learned latent space is expected to group engine trajectories into different clusters according to their underlying nature, illustrating a simplified representation. Furthermore, the regressor influences directly over the organization of the latent space and can report explicitly which RUL value is the one that best represents the cycles belonging to each engine unit that is fed to the model.

3.2. Interpretable diagnosis

The diagnostic tool introduced in this work is a map that shows the actual state of the engine and also the rate of change from healthy to deteriorated. What we pursue is a map in which each point represents the status of an engine associated with a window of events during its flight history so that points of degraded aircraft are grouped in nearby areas and, on the contrary, points belonging to healthy aircraft are located in more distant areas. As the actual health status of the training aircraft will be known, since we used the RUL target function, a color scale can be established to clearly differentiate healthy aircraft from deteriorated or totally deteriorated aircraft, coloring each point according to its corresponding RUL. Thinking about how variational inference works, this can be easily put into practice: once the model has been trained with the engine data, each input can be encoded into the latent space, being represented in terms of the mean and variance of the approximate distribution learned. This means that the data can be projected into the latent space and each point will be clustered near those with a similar degradation pattern. An example of one of the maps produced by this algorithm is shown in Fig. 4, right side. Aircraft with high RUL values are painted in yellow while aircraft with low RUL values are painted in dark purple. It can be observed that there is a clear progression in the colors along the map since events with no or low deterioration are located in the upper part of the map (high RUL values) while the most deteriorated ones are located in the lower part (low RUL values). This representation can be used later: when new unseen engine events are used as inputs, the encoder will place them according to their characteristics, giving information about their RUL depending on the proximity to other nearby points whose diagnosis is known. This is why it is considered explainable, since the method's decisions are based entirely on the learned representation and can therefore be justified; and interpretable, because a simple glance at the map gives insight into the status of each engine unit. Other Deep Learning methods can also reveal interpretable information in intermediate layers, however, extra processing is needed in order to find the most suitable layers or to transform the content of these layers into human readable information. An example of this is the embedding projector of tensorflow [49], which applies different dimensionality reduction methods such as UMAP, T-SNE or PCA to provide a visualization of the embedding layer. In contrast to this, our method provides a direct 2-D compression, which does not need any further processing. More details on the interpretation of this map are given in the following section.

4. Experimental design

Table 1 shows the different levels of difficulty of the datasets according to the last two rows. Each dataset can operate under different operating conditions and the system failure can be caused by two components: the turbine and the compressor. Thus, FD001 and FD003 operate under the same conditions although FD003 includes engines whose failure could be caused by either of the two mentioned components. Then, FD002 operates under 6 operating conditions as does FD004, while in FD004, as in FD003, the failure conditions cover both turbine and compressor failure. In this sense, it is believed that according to their characteristics, the level of difficulty of the datasets, in increasing order, is: FD001, FD003, FD002, FD004. Some studies focus on a particular dataset [50,51] and others explore in detail the impact of different hyperparameters such as the number of sensors to use or the upper limit for the target function for each dataset [43]. Still, this is a benchmark problem and the interest lies in finding a pre-processing procedure that can be applied to similar problems, rather than finding the ideal series of steps for a particular dataset.

For this reason, the decision we make, since the different failure conditions do not have a major impact on pre-processing, is to focus on those samples where the operating conditions are different. In those cases, even a simple exploratory analysis would yield little or no information concerning the signals because apparently operating conditions change between cycles, which makes analyzing and predicting RUL much more complex. It is important to take this into account when normalizing data, although it is something that seems to be overlooked in other papers since min-max normalization is usually used [15]. Instead, we take another path by using a condition-based standardization. With this approach, all records of the same operating condition are grouped together and scaled using a standard scaler. The application of this type of scaling will bring the average of the grouped operating conditions to zero. As this technique is applied for each operating condition separately, all signals will receive an average of zero, making them comparable [52].

On the other hand, although sensor data have a general trend, it is known that they are subject to local oscillations, mainly caused by high-frequency sensors, which lead to noise [23,43]. To ease the processing of the series, an exponential weighted moving average is carried out. It takes the current value and the previous filtered value into account when calculating the filtered value:

$$X'_j = \alpha * X_j + (1 - \alpha) * X'_{j-1}$$

where X'_j is the filtered value of X_j and α the strength of the filter. Lower values for α will have a stronger smoothing effect and consequently, stationarity is lost. Nevertheless, stronger smoothings lead to better model performance. It is important to note that what we intend to model is not the detection of failure points, but the changes in the degradation rate, i.e. those breakpoints where after some time, the engine parts deteriorate at a different rate than they did before. For this reason, the smoothing we apply does not adversely affect the data. Furthermore, the sole purpose of the filter is to reduce oscillations in the sensor measurements, therefore in no case is smoothing applied that would compromise the trend of the data.

In time series problems it is quite recurrent to split the data into sequences for better prediction performance. That is, multivariate series are not processed for each engine but are sliced into fixed-size windows as shown in Fig. 5. At each time step, data is picked from sensors within the time window to form a high-dimensional feature vector used as inputs to the network to predict the RUL. Thus, each input sample in our network contains thirty single-cycle data which is extracted from the following six sensors: T30, T50, P30, EPRA, PS30, phi and the aim is to find patterns in those time-windows that can lead to an adequate RUL estimation. There may be cases in which the partitioning of the sequences for a particular engine in the last few cycles may not have enough data to complete the length of the window. In those cases a

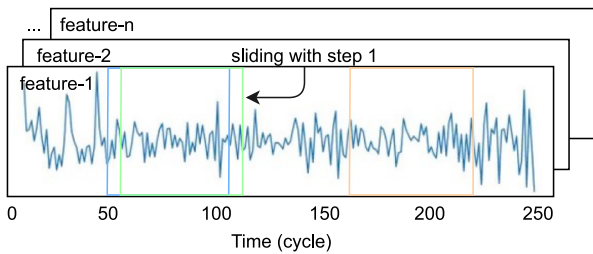


Fig. 5. Time window framing.

masked value is used and will be treated in the first layer of the model by simply ignoring those values. In this way, as much information as possible is used.

In the experiments performed, although there are not so many hyperparameters to adjust, the impact of those that are present is very noticeable in the final performance of the model. Thus, the time window length, the intensity of smoothing or the internal number of neurons of the recurrent layers are key elements. Hyperparameter tuning is an arduous task that in this case was driven by our own experience along with the use of the Hyperopt Bayesian optimization library [53] to find the final configuration. On the other hand, an accurate choice of the LR is particularly essential to improve the optimization process, therefore for this parameter an adaptive LR optimizer, Adam, and the Cyclic Learning Rate technique proposed in [54] were used to help to select the optimal LR with which to start the training. In addition, to achieve the best possible performance of our model, we used callbacks to customize our experiments in terms of relegating the training stop condition to the validation error instead of the number of epochs. These tweaks led us to find the best results for the functions to be optimized.

The choice of the sensors is not arbitrary, we only use the following six: T30, T50, P30, EPRA, PS30 and phi, which are precisely the ones available in the real problem we introduce later on. Note that in both datasets the engines are Turbofan aircraft engines. Surprisingly, we found that out of the twenty-one provided sensors, of which most of them are used in similar works, using only these not only reduces computational costs but also gives sufficient information to predict the RUL efficiently. Moreover, for datasets FD001 and FD003 the EPRA sensor is not necessary since it measures the engine thrust under different operating conditions while FD001 and FD003 operate under the same condition and so this sensor does not provide relevant information, the captured values simply remain constant.

Finally, 20% of the training data was used for validation, resulting in 17692 training samples and 4128 validation samples for FD001 and FD003, 37432 training samples and 8787 validation samples for FD002 and finally 43523 training samples and 10505 validation samples for FD004. All models and experiments were implemented in TensorFlow. Further details regarding the experimental setup and the source code to reproduce the experimental results are available in the following public git repository: <https://github.com/NahuelCostaCortez/Remaining-Useful-Life-Estimation-Variational>

5. Experimental results

The experimental validation of the proposed framework has two parts. First, C-MAPSS datasets are used to compare our framework with other state-of-the-art approaches for RUL estimation. Secondly, C-MAPSS engines as well as actual engines from the real problem we present are diagnosed based on their projections in the latent space. We begin by introducing the numerical results, the diagnostic map achieved is then presented and finally, the experimental validation is discussed.

5.1. Results on C-MAPSS

In this section we demonstrate that our framework can compete with state-of-the-art methods for RUL estimation. Both the training and test sets used are the same for all methods, since both sets are provided in the original dataset, as stated in Table 1. It is worth mentioning that the baseline methods we present, which collect the most impactful approaches to date, do not provide any representation of the data, but merely predict the RUL corresponding to the next time step. This makes us appreciate the importance of Representation Learning as it provides a piece of more illustrative information than a simple numerical or categorical result.

The comparison results of the proposed framework with other popular approaches on the test sets are listed in Table 2 where the selected metrics of all methods, included ours, labeled as RVE (Recurrent Variational Encoder), are listed for every dataset. Results in which our method outperforms the others are highlighted in bold. It can be quickly noted that with datasets FD001 and FD003, although the metrics are considered good, they are not the best. However, the interest lies mostly in FD002 and FD004 as the increasing number of operating conditions and failure modes make these two datasets contain more complicated multiscale degradation features. RVE significantly improves prediction accuracy in these two for both Score and RMSE, due to its good feature extraction capability in the face of these complex fault prediction problems. The comparison also includes a row labeled “VAE+RNN”, which corresponds to the adaptation of a recurrent VAE to this problem. The superiority of our model can be clearly seen. Although both use variational inference, the numerical differences are explained by the different latent spaces obtained: one dispersed and the other one continuous (recall Figs. 1 and 4), allowing the latter to improve the predictive capabilities of the model.

RUL estimations for the life-time of some testing engine units corresponding to the different datasets are shown in Fig. 6. It is very common to see figures like these in papers working with C-MAPSS, exhibited to obtain an understanding of the model’s performance. The RUL constructed from the piece-wise function is represented in orange, of which C-MAPSS provides the RUL corresponding to the last cycle. RUL values predicted at each time instant by our method are presented in blue. It is clearly seen that the network is able to model this degradation to, finally, accurately predict the real RUL of the engine. However, this is not enough to explain the performance of the model and this is where we differ from other methods.

These kind of figures seem very clear and promising but despite being good predictions, there is a gap when it comes to explainability of the model’s decisions and internal representations. A gap that can be filled with techniques such as the one we propose. As explained in Section 3, the latent space build by the encoder serves as a basis for creating a map that allows us to understand the evolution of the data over time and Fig. 7 is a sample of this. Each map represents the latent space obtained for the set of cycles traveled by each aircraft shown in Fig. 6. That is, for example, for plane #7 the compressed representation of the first thirty cycles corresponds to the first upper left red dot, while the compressed representation of the last thirty would be the last lower right red dot. The remaining points correspond to the representations of each data sample seen during training. The encoder learns to locate in the latent space each data window passed to it according to its characteristics. Thus, in all the exposed maps, in which the RUL is labeled in the color bar, it can be seen that when the airplane is operating in favorable conditions (high RUL values), its latent representation is located in the upper left zone and, as it begins to degrade, this location moves to the right until the data indicating that the airplane is degraded (low RUL) are located in the lower rightmost area.

In this way, a model that can be fed with data from the trajectories of an aircraft that has flown at least thirty cycles is achieved. From there, the model can be fed each time a new data sample is available

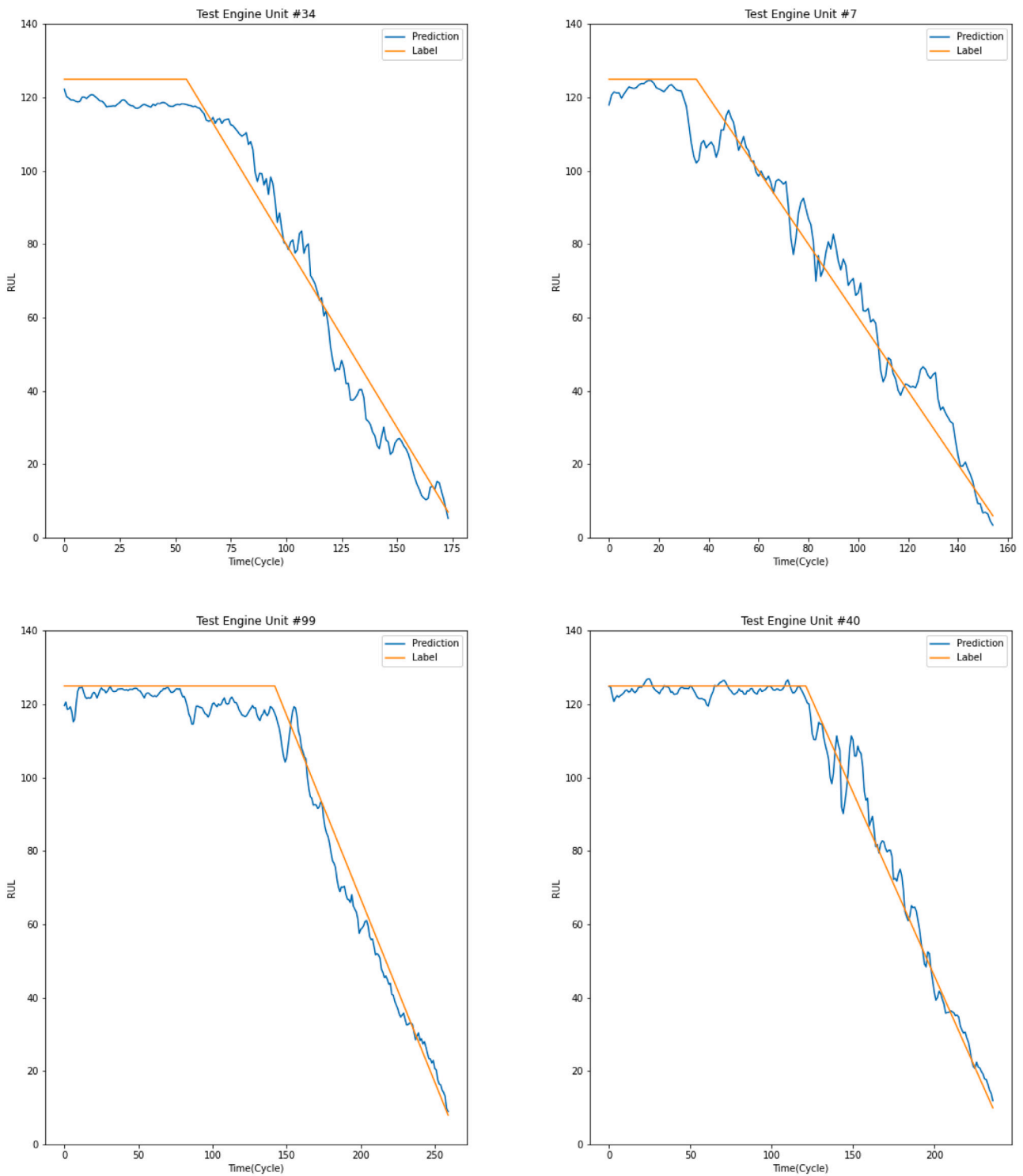


Fig. 6. Four examples of life-time RUL predictions for testing units: #34 corresponding to FD001, #7 corresponding to FD002, #99 corresponding to FD003 and #40 corresponding to FD004.

in order to obtain the two diagnostics we are looking for. First, a visual diagnosis is presented in which, based on the proximity to samples whose condition is known, the state of the aircraft at that particular moment is perceived. In <https://github.com/NahuelCostaCortez/Remaining-Useful-Life-Estimation-Variational/tree/main/images/gifs> there are some gifs available corresponding to the engines from Figs. 6 and 7 in which the speed of deterioration suffered by these airplanes along the cycles can be appreciated. Second, a quantitative diagnosis

is obtained that explicitly reports the RUL value that determines the remaining life time of the aircraft.

5.2. Results on a real-world problem

To illustrate how the proposed model may work in a more realistic context, an example for actual engines is presented below. The data is sampled on Turbofan engines under actual conditions of use. Unfortunately, for confidentiality reasons, we are unable to disclose the

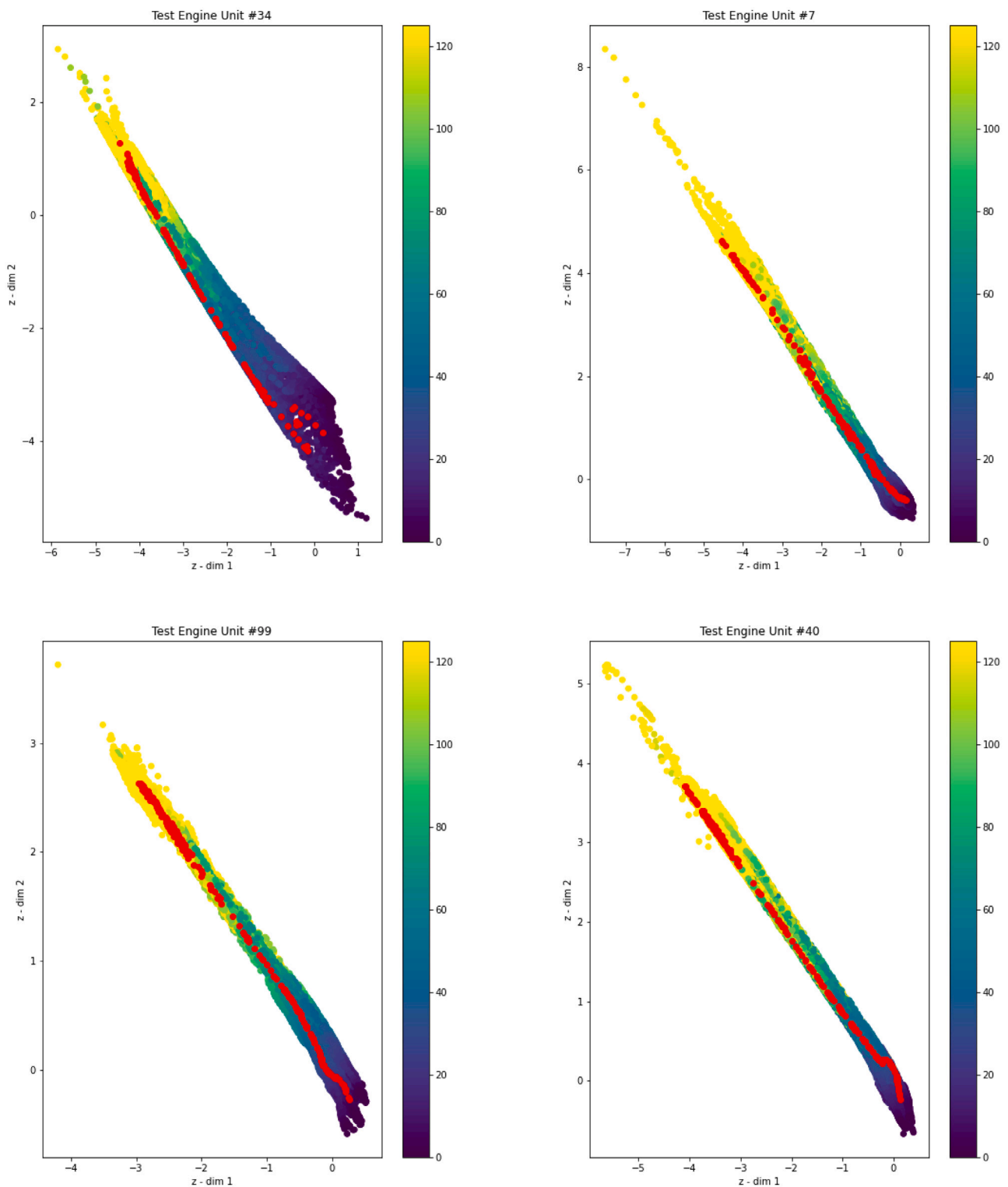


Fig. 7. Latent predictions for every time-step of the samples presented in 6.

name of the company or make the data set public. Nevertheless, a brief description of the engines is provided in Table 3.

The pre-processing applied is the same as the one explained in Section 4. The objective is to convert the data from each engine into inputs that can be processed by the network. During training, as in the NASA’s dataset, the model learns different deterioration patterns which leads the encoder to project the engine units into the latent space according to their degradation, maintaining coherence in the

distances between healthy and compromised engines. This projection is again used as a basis to find out, given undiagnosed units, how their degradation evolves as the number of flights increases. Fig. 8, as the above-mentioned gifs, pictures this idea: six airplanes have been chosen to project their state into the latent space in two different time steps: $t = 0$ would correspond to feeding the network with the data corresponding to the cycles from 0 to windows length and so it is the same with $t = 1000$, starting from data corresponding to the cycle 1000.

Table 2
Evaluation metrics of different approaches for RUL estimation on C-MAPSS datasets.

	FD001		FD002		FD003		FD004	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
MLP [17]	37.56	18 000	80.03	7 800 000	37.39	17 400	77.37	5 620 000
SVR [17]	20.96	1380	42.00	590 000	21.05	1600	45.35	371 000
RVR [17]	23.80	1500	31.30	17 400	22.37	1430	34.34	26 500
CNN [17]	18.45	1299	30.29	13 600	19.82	1600	29.16	7890
Deep LSTM [17]	16.14	338	24.49	4450	16.18	852	28.17	5550
Semi-supervised [23]	12.56	231	22.73	3366	12.10	251	22.66	2840
DCNN [55]	12.61	273.7	22.36	10 412	12.64	284.1	23.31	12466
MS-DCNN [55]	11.44	196.22	19.35	3747	11.67	241.89	22.22	4844
VAE+RNN	15.81	326	24.12	4183	14.88	722	26.54	5634
RVE	13.42	323.82	14.92	1379.17	12.51	256.36	16.37	1845.99

Table 3
Summary of the main properties of the engines provided by the manufacturer.

Model/properties	Type I engine	Type II engine
Thrust ratings	between 14,750 and 15,000 lb	between 14,000 and 21,500 lb
Flying hours	11m+	11m+
N° fans	1	2
Fan diameter	48 in.	58 in.
Two-shaft, high-bypass-ratio engine	Yes	Yes

The RUL provided by the model is shown in the colorbar. Fixing the latent projection obtained after training gives us some insight into the progression of the health status of these units: The latent projection of engine e1, e2, e3 and e4 during the time steps shown remain over the upper left quadrant, next to other aircraft with similar characteristics: RUL around two hundred cycles, with no signs of near degradation. On the contrary, there is a clear progression in samples e5 and e6, which move clearly downward, being placed together with engine units close to their end of life (low values of RUL), thus obtaining an accurate and explainable diagnosis beyond a possible label indicating the predicted health.

In the figure presented only two time steps have been selected to show the update of the health status of the engines according to the data from their sensors. However, it is noteworthy that once enough data is available to be fed into the network in each subsequent trip this update can be performed because we are using recurrent networks. This is where the interest really lies because this update allow us coping with the non-stationarity of the data distribution and in the end this can be used as a diagnostic tool. As mentioned in the introduction, this is an online process, being the useful life of each system continuously monitored. Particularly in the company, these motors have periodic maintenance cycles and also have parallel systems that warn in case of detecting any anomalous operation. The fact of having a diagnostic system of these characteristics, however, represents an invaluable economic saving for the company. This is because the aim of the method is to prevent such anomalies from occurring. To this end, the degradation speed of the engines is modeled, so that the acceleration in the normal degradation speed of an engine can be easily detected. In this way, the probability of the aircraft having an unexpected event is highly reduced.

An example of interpretation of the method is as follows: Arrows were used in Fig. 8 to depict the evolution of each sample, but as demonstrated in the previous sub-section, a footprint of every step is recorded (that is, a latent projection is available for $t = 1$, $t = 2$, $t = 3$ and so on) so there is a clear evolution over time and the rate of these updates may trigger alerts in a real-world scenario: as long as the engine projection remains in the healthy range, its evolution will be considered positive; on the contrary, if the projection moves towards the red zone rapidly, it may be a clear sign of deterioration, information that will be used by the mechanics to make a decision regarding its follow-up, either to make it more exhaustive or to take

the aircraft to the workshop for a more complete overhaul, to name some alternatives. This translates into a prolongation of the useful life of these engines by being able to anticipate the breaking point at which severe deterioration may occur.

6. Concluding remarks and future work

We have proposed a novel architecture based on variational encoding with a new way of regularizing latent representations to address aircraft engine diagnostics. These are obtained through variational inference and are shaped by a term in the cost function that penalizes erroneous RUL estimates. The result is a latent space capable of projecting the history of engines trajectories continuously and without abrupt jumps, like other models such as VAEs. As a consequence, the latent space learned by the encoder is used as a diagnostic tool. It learns a two-dimensional representation of engine data with different deterioration stages to, given an unseen engine, project its encoding near engines with similar degradation patterns. Thus, prevailing an explainable diagnosis.

We have demonstrated that, besides providing a visual assessment of the rate of degradation in aircraft engines, our method can also accurately estimate the RUL. To this end, we used the popular C-MAPSS simulation dataset on which we outperformed most of the current state-of-the-art methods. We have shown that the learned latent space can comprehensively model aircraft degradation history and consequently improve prediction capabilities. Furthermore, we include a report of its application to data belonging to actual engines to illustrate its performance in a real-world scenario.

Lastly, in future works we aim to explore the suitability of the model in other areas related to condition monitoring and predictive maintenance. Additionally, it would be of interest to motivate the model to learn latent features that, beyond differentiating the stages of degradation, can also explain the different causes of failure.

CRedit authorship contribution statement

Nahuel Costa: Writing – original draft, Validation, Supervision, Software, Methodology, Investigation. **Luciano Sánchez:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been partially supported by the Ministry of Economy, Industry and Competitiveness (“Ministerio de Economía, Industria y Competitividad”) from Spain/FEDER under grant PID2020-112726-RB-100 and by Principado de Asturias, grant SV-PA-21-AYUD/2021/50994.

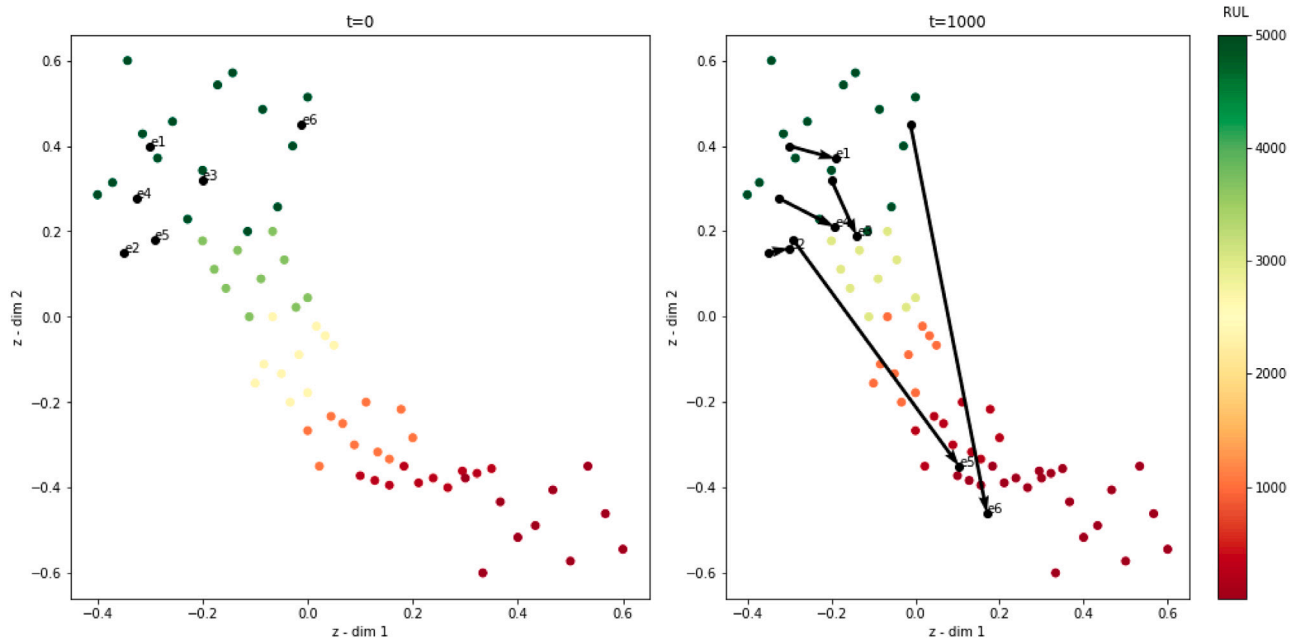


Fig. 8. RUL evolution of six selected engines. As the cycles progress, the aircraft are placed in areas close to other aircraft with similar degradation patterns whose diagnosis is known, thus making it easy to identify those that degrade more rapidly, as in the case of e5 and e6.

References

[1] Azadeh A, Asadzadeh S, Salehi N, Firoozi M. Condition-based maintenance effectiveness for series-parallel power generation system—A combined Markovian simulation model. *Reliab Eng Syst Saf* 2015;142:357–68.

[2] Zhao Z, Liang B, Wang X, Lu W. Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Reliab Eng Syst Saf* 2017;164:74–83.

[3] Ali JB, Chebel-Morello B, Saidi L, Malinowski S, Fnaiech F. Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network. *Mech Syst Signal Process* 2015;56:150–72.

[4] Jouin M, Gouriveau R, Hissel D, Péra M-C, Zerhouni N. Degradations analysis and aging modeling for health assessment and prognostics of PEMFC. *Reliab Eng Syst Saf* 2016;148:78–95.

[5] Si X-S, Wang W, Hu C-H, Zhou D-H. Remaining useful life estimation—a review on the statistical data driven approaches. *European J Oper Res* 2011;213(1):1–14.

[6] Tian Z. An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring. *J Intell Manuf* 2012;23(2):227–37.

[7] Kwan C, Zhang X, Xu R, Haynes L. A novel approach to fault diagnostics and prognostics. In: 2003 IEEE international conference on robotics and automation (Cat. No. 03CH37422), Vol. 1. IEEE; 2003, p. 604–9.

[8] Zhang X, Xu R, Kwan C, Liang SY, Xie Q, Haynes L. An integrated approach to bearing fault diagnostics and prognostics. In: Proceedings of the 2005, American control conference, 2005. IEEE; 2005, p. 2750–5.

[9] Khawaja T, Vachtsevanos G, Wu B. Reasoning about uncertainty in prognosis: a confidence prediction neural network approach. In: NAFIPS 2005-2005 annual meeting of the North American fuzzy information processing society. IEEE; 2005, p. 7–12.

[10] Zio E, Di Maio F. A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. *Reliab Eng Syst Saf* 2010;95(1):49–57.

[11] Khelif R, Chebel-Morello B, Malinowski S, Laajili E, Fnaiech F, Zerhouni N. Direct remaining useful life estimation based on support vector regression. *IEEE Trans Ind Electron* 2016;64(3):2276–85.

[12] Singh SK, Kumar S, Dwivedi J. A novel soft computing method for engine RUL prediction. *Multimedia Tools Appl* 2019;78(4):4065–87.

[13] Xu Z, Saleh JH. Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. *Reliab Eng Syst Saf* 2021;107530.

[14] Babu GS, Zhao P, Li X-L. Deep convolutional neural network based regression approach for estimation of remaining useful life. In: International conference on database systems for advanced applications. Springer; 2016, p. 214–28.

[15] Li X, Ding Q, Sun J-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliab Eng Syst Saf* 2018;172:1–11.

[16] Li X, Zhang W, Ma H, Luo Z, Li X. Data alignments in machinery remaining useful life prediction using deep adversarial neural networks. *Knowl-Based Syst* 2020;197:105843.

[17] Zheng S, Ristovski K, Farahat A, Gupta C. Long short-term memory network for remaining useful life estimation. In: 2017 IEEE international conference on prognostics and health management (ICPHM). IEEE; 2017, p. 88–95.

[18] Yu W, Kim IY, Mechefske C. An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme. *Reliab Eng Syst Saf* 2020;199:106926.

[19] Shi Z, Chehade A. A dual-LSTM framework combining change point detection and remaining useful life prediction. *Reliab Eng Syst Saf* 2021;205:107257.

[20] Li J, Li X, He D. A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction. *IEEE Access* 2019;7:75464–75.

[21] Xia T, Song Y, Zheng Y, Pan E, Xi L. An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation. *Comput Ind* 2020;115:103182.

[22] Canizo M, Triguero I, Conde A, Onieva E. Multi-head CNN-RNN for multi-time series anomaly detection: An industrial case study. *Neurocomputing* 2019;363:246–60.

[23] Ellefsen AL, Bjørlykhaug E, Æsøy V, Ushakov S, Zhang H. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliab Eng Syst Saf* 2019;183:240–51.

[24] Xiang S, Qin Y, Luo J, Pu H, Tang B. Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction. *Reliab Eng Syst Saf* 2021;216:107927.

[25] Zio E. Prognostics and health management (PHM): Where are we and where do we (need to) go in theory and practice. *Reliab Eng Syst Saf* 2022;218:108119.

[26] Sakurada M, Yairi T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis. 2014, p. 4–11.

[27] Zhou C, Paffenroth RC. Anomaly detection with robust deep autoencoders. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. 2017, p. 665–74.

[28] Zhao Y, Deng B, Shen C, Liu Y, Lu H, Hua X-S. Spatio-temporal autoencoder for video anomaly detection. In: Proceedings of the 25th ACM international conference on multimedia. 2017, p. 1933–41.

[29] Guo X, Liu X, Zhu E, Yin J. Deep clustering with convolutional autoencoders. In: International conference on neural information processing. Springer; 2017, p. 373–82.

[30] Martínez-García M, Zhang Y, Wan J, McGinty J. Visually interpretable profile extraction with an autoencoder for health monitoring of industrial systems. In: 2019 IEEE 4th international conference on advanced robotics and mechatronics (ICARM). IEEE; 2019, p. 649–54.

[31] Camacho J, Pérez-Villegas A, García-Teodoro P, Maciá-Fernández G. PCA-based multivariate statistical network monitoring for anomaly detection. *Comput Secur* 2016;59:118–37.

[32] Sanchez-Fernández A, Fuente MJ, Sainz-Palmero G. Fault detection in wastewater treatment plants using distributed PCA methods. In: 2015 IEEE 20th conference on emerging technologies & factory automation (ETFA). IEEE; 2015, p. 1–7.

[33] An J, Cho S. Variational autoencoder based anomaly detection using reconstruction probability. *Spec Lect IE* 2015;2(1):1–18.

- [34] Sun J, Wang X, Xiong N, Shao J. Learning sparse representation with variational auto-encoder for anomaly detection. *IEEE Access* 2018;6:33353–61.
- [35] Xu H, Chen W, Zhao N, Li Z, Bu J, Li Z, Liu Y, Zhao Y, Pei D, Feng Y, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In: *Proceedings of the 2018 world wide web conference*. 2018, p. 187–96.
- [36] Ren L, Sun Y, Cui J, Zhang L. Bearing remaining useful life prediction based on deep autoencoder and deep neural networks. *J Manuf Syst* 2018;48:71–7.
- [37] Yu W, Kim IY, Mechefske C. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. *Mech Syst Signal Process* 2019;129:764–80.
- [38] Costa N, Sánchez L. Remaining useful life estimation using a recurrent variational autoencoder. In: *International conference on hybrid artificial intelligence systems*. Springer; 2021, p. 53–64.
- [39] Xiongzi C, Jinsong Y, Diyin T, Yingxun W. Remaining useful life prognostic estimation for aircraft subsystems or components: A review. In: *IEEE 2011 10th international conference on electronic measurement & instruments, Vol. 2*. IEEE; 2011, p. 94–8.
- [40] Saxena A, Goebel K. Phm08 challenge data set. In: *NASA Ames prognostics data repository*. NASA Ames Research Center; 2008.
- [41] Ayhan B, Kwan C, Liang SY. Adaptive remaining useful life prediction algorithm for bearings. In: *2018 IEEE international conference on prognostics and health management (ICPHM)*. IEEE; 2018, p. 1–8.
- [42] Heimes FO. Recurrent neural networks for remaining useful life estimation. In: *2008 international conference on prognostics and health management*. IEEE; 2008, p. 1–6.
- [43] Miao H, Li B, Sun C, Liu J. Joint learning of degradation assessment and RUL prediction for aeroengines via dual-task deep LSTM networks. *IEEE Trans Ind Inf* 2019;15(9):5023–32.
- [44] Kingma DP, Welling M. An introduction to variational autoencoders. 2019, arXiv preprint arXiv:1906.02691.
- [45] Kingma DP, Mohamed S, Rezende DJ, Welling M. Semi-supervised learning with deep generative models. In: *Advances in neural information processing systems*. 2014, p. 3581–9.
- [46] Zhao Q, Adeli E, Honnorat N, Leng T, Pohl KM. Variational autoencoder for regression: Application to brain aging analysis. In: *International conference on medical image computing and computer-assisted intervention*. Springer; 2019, p. 823–31.
- [47] Wu Y, Yuan M, Dong S, Lin L, Liu Y. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing* 2018;275:167–79.
- [48] Elsheikh A, Yacout S, Ouali M-S. Bidirectional handshaking LSTM for remaining useful life prediction. *Neurocomputing* 2019;323:148–56.
- [49] Tensorflow, Embedding projector. URL <https://projector.tensorflow.org/>.
- [50] Nguyen H, Tran KP, Thomassey S, Hamad M. Forecasting and anomaly detection approaches using LSTM and LSTM autoencoder techniques with the applications in supply chain management. *Int J Inf Manage* 2021;57:102282.
- [51] Aydemir G, Acar B. Anomaly monitoring improves remaining useful life estimation of industrial machinery. *J Manuf Syst* 2020;56:463–9.
- [52] Pasa GD, de Medeiros IP, Yoneyama T. Operating condition-invariant neural network-based prognostics methods applied on turbofan aircraft engines. In: *Proceedings of the annual conference of the PHM society, Vol. 11*. 2019.
- [53] Bergstra J, Yamins D, Cox DD. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In: *Proceedings of the 12th Python in science conference, Vol. 13*. Citeseer; 2013, p. 20.
- [54] Smith LN. Cyclical learning rates for training neural networks. In: *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE; 2017, p. 464–72.
- [55] Li H, Zhao W, Zhang Y, Zio E. Remaining useful life prediction using multi-scale deep convolutional neural network. *Appl Soft Comput* 2020;89:106113.



Original software publication

RUL-RVE: Interpretable assessment of Remaining Useful Life

Nahuel Costa *, Luciano Sánchez

Computer Science Department, University of Oviedo, Gijón, 33202, Asturias, Spain



ARTICLE INFO

Keywords:

Remaining useful life
Prognostics and health management
Interpretability
Variational inference
Recurrent neural networks
Python

ABSTRACT

This paper presents RUL-RVE, a Python tool for the assessment of Remaining Useful Life (RUL). Physical systems are normally subject to degradations that ultimately lead to failure, therefore prognostic technologies are crucial to estimate the lifetime of the system to be monitored. The problem with most existing data-driven approaches is that they lack an explanatory component to understand model learning and/or the nature of the data. RUL-RVE is a framework based on recurrent neural networks and variational inference that can achieve remarkable forecast accuracy while providing an interpretable assessment, which is highly valuable in real-world environments.

Code metadata

Current code version	v0.1
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2022-58
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/4781584/tree/v1
Legal Code License	MIT License
Code versioning system used	git
Software code languages, tools, and services used	python
Compilation requirements, operating environments & dependencies	tensorflow>= 2.3.0, matplotlib>= 3.3.4, pandas>= 1.1.5, scikit-learn>= 1.0.2
If available Link to developer documentation/manual	
Support email for questions	costanahuel@uniovi.com

1. Introduction

Remaining useful life (RUL) is an estimate of the length of time an item, component, or system is estimated to be able to function according to its intended purpose before requiring repair or replacement. It is considered a key metric in prognosis that helps improve maintenance schedules and avoid engineering, safety, and reliability failures [1]. This has many real-world applications such as monitoring machining tools, batteries, turbofan engines, and rotating bearings [2].

Many techniques have been proposed to model the degradation of these complex systems, from which two currents arise: model-based approaches and data-driven approaches [3]. The former techniques usually require extensive prior knowledge about the physical systems, information that is often not available in practice. On the contrary, data-driven approaches have become popular in recent years, as they are able to model degradation features based purely on historical

records from which the underlying causalities and correlations can be modeled.

The greater impact has undoubtedly been produced by the use of Deep Learning models [4] given that the high dimensionality of raw data obtained from machine health monitoring can be modeled by methods that are known to perform remarkably well, especially in Computer Vision and Natural Language Processing (NLP).

Nevertheless, there is a clear gap between most Deep Learning approaches: although they do achieve very accurate results, models are usually treated as black boxes where it is not trivial to obtain explanations of the decisions that led the model to predict such outputs [5]. Although the current practice is to dispense with prior knowledge about the system to be monitored, in the end, these models are designed to be used by people outside academia. Consequently, it is essential to provide tools that offer some interpretability of the models' decisions,

The code (and data) in this article has been certified as Reproducible by Code Ocean: (<https://codeocean.com/>). More information on the Reproducibility Badge Initiative is available at <https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals>.

* Corresponding author.

E-mail address: costanahuel@uniovi.es (N. Costa).

<https://doi.org/10.1016/j.simpa.2022.100321>

Received 27 April 2022; Received in revised form 16 May 2022; Accepted 19 May 2022

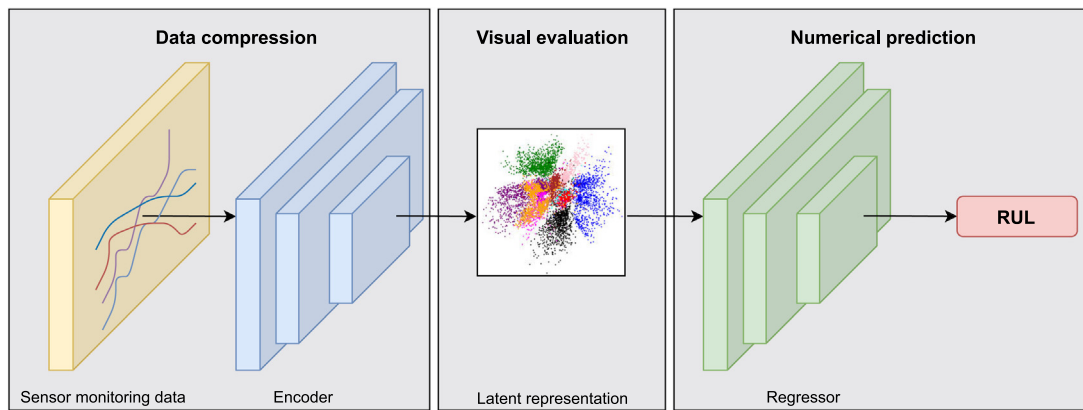


Fig. 1. Workflow followed in the framework: monitoring data is fed into the encoder, which learns a latent representation based on deterioration patterns in order to build a graphical map reflecting the evolution of the samples. The regressor learns from such latent space to report numerically the RUL of each sample.

as well as some insights into the nature of the data. In fact, these are attributes of particular interest, if not demanded, for decision making in safety-critical applications [6].

RUL-RVE [7] is a framework for the task of accurately estimating RUL while providing an explainable and interpretable diagnosis. Its performance was validated on the popular C-MAPSS dataset from NASA [8]. This dataset contains simulated data of Turbofan engines produced by Commercial Modular Aero-Propulsion System Simulation (C-MAPSS), a model-based simulation program. It is composed of multivariate temporal data obtained from twenty-one sensors and is further divided into 4 sub-datasets that differ in operating and fault conditions. RUL-RVE demonstrated that, besides providing a visual assessment of the rate of degradation in aircraft engines, it can also accurately estimate the RUL, where it outperformed most state-of-the-art approaches in terms of RMSE. In addition, its application to data belonging to actual Turbofan engines was also tested to illustrate its performance in a real-world scenario. The model's ability to continuously monitor the useful life of each system makes it possible to detect potential future anomalies, which translated into significant economic savings for the company studied.

2. Description

2.1. Workflow

Fig. 1 illustrates the workflow followed to estimate numerically and visually the RUL. The framework is based on variational inference and is composed of two neural networks: a recurrent encoder and a regression model. The encoder learns to compress the sensor monitoring data to a latent space led by the mean and the variance of an approximated Gaussian distribution, thus resulting in a 2-D representation in which input samples are organized based on their deterioration patterns. The regressor directly influences the training process for obtaining such representation and can also report explicitly which RUL value is the one that best represents each sample that is fed to the model.

In short, the user will have two prediction elements: a numerical estimation of the RUL and a visual map. This allows to visually assess the evolution of the input data, as the samples will be placed in areas close to other samples with similar degradation patterns whose diagnosis is known, thus making it easy to identify those that degrade more rapidly. The fact of having a diagnostic system of these characteristics is crucial for reducing unexpected events to happen because the degradation speed of the components is modeled so that the acceleration in the normal degradation speed of a component can be easily detected.

2.2. Implementation

The RUL-RVE model was implemented in Tensorflow 2 [9]. To introduce variational inference into the loss function, a custom model was created in which the training and testing steps were overridden. Also, a regression penalty was included to cause systems with similar degradation patterns to project into nearby areas of the latent space. Bidirectional LSTM layers were chosen for the encoder as they provide not only information about the past but also about the future, allowing the network to be aware of what the data may look like in its future stages, which helps it to understand what kind of information to predict (different stages of deterioration). For the regressor, a simple Feed Forward network with an intermediate layer with a tanh activation was implemented.

Additional pre-processing methods are included to deal with the most common input data, i.e., multivariate time-series, including scaling, smoothing, window framing and data splitting.

2.3. Usage example

In this section, we provide an example of use of the RUL-RVE to train and test on a given dataset. First, we recommend setting up a new python environment with packages matching the requirements.txt file included in the attached Github repository. It can be easily done with anaconda [10]: `conda create -name -file requirements.txt`. Another alternative is to run exactly the same environment under which this project was made with Docker [11]. A Docker file is provided, which contains the set of instructions for creating a container with all the necessary packages and dependencies. The fastest way to set it up is to download the project from GitHub, open Visual Studio Code editor, and from the command palette select "Remote-containers: Open folder in Container". Once the environment is configured, within a few lines of code (Fig. 2) the framework can be easily used for training and evaluation.

2.4. Impact overview

Artificial Intelligence research has taken a path in recent years in which the main focus is established on neural networks applied mostly to Natural Language Processing (NLP) and Computer Vision, while the application of some of these powerful algorithms for time series is not yet fully exploited. In addition, the scalability of these models makes the computational requirements increasingly higher, which is a major barrier for most research and industry groups. This is exacerbated by the fact that despite being incredibly good at some tasks, most AI models behave as black boxes that are based on feeding an input to an algorithm that outputs some number or class. However, there are


```

import model
import utils

# ----- DATA -----
x_train, y_train, x_val, y_val, x_test, y_test = # load your dataset
# -----

# ----- MODEL -----
RVE = model.create_model(timesteps = x_train.shape[1], input_dim = x_train.shape[2],
                        intermediate_dim = 300, batch_size= 128, latent_dim = 2,
                        epochs = 10000, optimizer = 'adam')
# Callbacks for training
model_callbacks = utils.get_callbacks(RVAE, x_train, y_train)
# -----

# ----- TRAINING -----
results = RVE.fit(x_train, y_train, shuffle=True, epochs=epochs,
                batch_size=batch_size, validation_data= (x_val, y_val),
                callbacks=model_callbacks, verbose=2)
# -----

# ----- EVALUATION -----
RVAE.load_weights('./checkpoints/checkpoint')
# Visual map
test_mu = utils.viz_latent_space(RVE.encoder, x_test, y_test)
# Evaluate RUL estimation
utils.evaluate(y_test, RVE.regressor.predict(test_mu), 'test')
# -----
    
```

Fig. 2. Based on a given dataset, the framework is trained and evaluated.

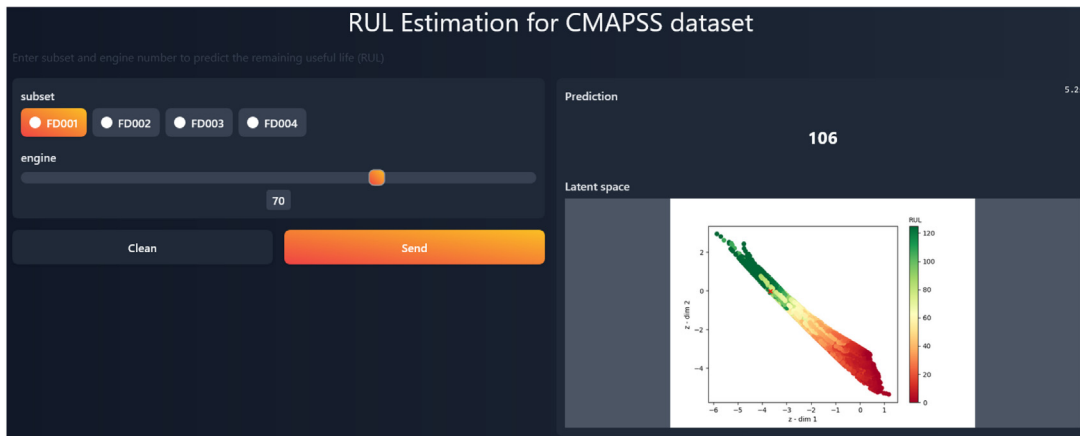


Fig. 3. Gradio demo of the model presented in [7].

many fields where this is not enough and this is where the importance of explainability comes into play. It is crucial to know in some way how these models work internally in order to bring transparency to research and provide certain interpretability of the models' decisions so that they can be easily used by people of any area outside AI.

RUL-RVE emerges as a solution to these problems: it proposes the use of a powerful yet lightweight Deep Learning model: an encoder, implemented with recurrent networks to deal with the temporality of the data, that provides an interpretable evaluation of the component to be monitored. The framework is still young and has been recently published [7] so it has not yet been used in any other existing publications, however, it can be leveraged both in industry and research.

The tool has already been tested in a real industrial test case for RUL prediction of Turbofan engines. This opens up new opportunities

to apply the model to other industrial problems of similar nature. Precisely, RUL prediction is present in a wide variety of domains such as manufacturing, power generation, automotive, or transportation. RUL-RVE focuses on the interpretability part to facilitate its use by practitioners in these sectors. The model allows updating the learned patterns as more data becomes available, which makes the successive projections of each system on the visual map form an easily guessable trajectory into the future, something much more valuable than a simple numerical prediction. Also, the lightness of the model in terms of memory facilitates its implementation on any hardware with limited computational capabilities, which is of interest for online monitoring.

On the other hand, since the main value of the tool is to monitor a system in order to detect possible future failures early enough to make decisions to expand its lifetime, RUL-RVE can extend its application to

other contexts. In this sense, the tool is made available to academic researchers so that they can develop and adapt the model to different domains such as health or economics, since the system to be monitored is not limited only to aircraft engines but can also be applied to the stock market or a pacemaker, to name a few examples.

2.5. Illustrative examples

A demo of the model presented in [7] is available at <https://huggingface.co/spaces/NahuelCosta/RUL-Variational>. The user is presented with the 4 subsets of the CMAPSS dataset, from which they can choose which engine of the test set they want to know the RUL and its representation in the latent space. An illustrative example is shown in Fig. 3. The engine #70 of the FD001 subset is chosen to predict its RUL and visualize its location in the latent space. The RUL estimate is 106 and the location of the engine (marked with an x) is at the top of the map, along with other engines with similar RUL values. The area appears safe, indicating that the engine shows no signs of concern for maintenance at this time.

Declaration of competing interest

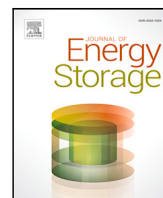
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been partially supported by the Ministry of Economy, Industry and Competitiveness (“Ministerio de Economía, Industria y Competitividad”) from Spain/FEDER under grant PID2020-112726-RB-I00 and by Principado de Asturias, grant SV-PA-21-AYUD/2021/50994.

References

- [1] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, Dong-Hua Zhou, Remaining useful life estimation—a review on the statistical data driven approaches, *European J. Oper. Res.* 213 (1) (2011) 1–14.
- [2] Liangwei Zhang, Jing Lin, Bin Liu, Zhicong Zhang, Xiaohui Yan, Muheng Wei, A review on deep learning applications in prognostics and health management, *Ieee Access* 7 (2019) 162415–162438.
- [3] Giduthuri Sateesh Babu, Peilin Zhao, Xiao-Li Li, Deep convolutional neural network based regression approach for estimation of remaining useful life, in: *International Conference on Database Systems for Advanced Applications*, Springer, 2016, pp. 214–228.
- [4] Zhaoyi Xu, Joseph Homer Saleh, Machine learning for reliability engineering and safety applications: Review of current status and future opportunities, *Reliab. Eng. Syst. Saf.* (2021) 107530.
- [5] Davide Castelvechi, Can we open the black box of AI? *Nat. News* 538 (7623) (2016) 20.
- [6] Enrico Zio, Prognostics and health management (PHM): Where are we and where do we (need to) go in theory and practice, *Reliab. Eng. Syst. Saf.* 218 (2022) 108119.
- [7] Nahuel Costa, Luciano Sánchez, Variational encoding approach for interpretable assessment of remaining useful life estimation, *Reliab. Eng. Syst. Saf.* 222 (2022) 108353.
- [8] A. Saxena, K. Goebel, Phm08 Challenge Data Set, NASA Ames Prognostics Data Repository, NASA Ames Research Center, 2008.
- [9] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, Software available from tensorflow.org.
- [10] Anaconda software distribution, in: *Anaconda Documentation*, Anaconda Inc., 2020.
- [11] Dirk Merkel, Docker: lightweight linux containers for consistent development and deployment, *Linux J.* 2014 (239) (2014) 2.



Research papers

Li-ion battery degradation modes diagnosis via Convolutional Neural Networks

N. Costa ^{a,*}, L. Sánchez ^a, D. Anseán ^b, M. Dubarry ^c^a Computer Science Department, Polytechnic School of Engineering, University of Oviedo, Gijón, 33202, Asturias, Spain^b Department of Electrical Engineering, Polytechnic School of Engineering, University of Oviedo, Gijón, 33204, Asturias, Spain^c Hawaii Natural Energy Institute, SOEST, University of Hawaii at Manoa, 1680 East-West Road, Honolulu, POST 109, HI 96822, USA

ARTICLE INFO

Dataset link: <http://dx.doi.org/10.17632/bs2j56pn7y.3>, <http://dx.doi.org/10.17632/6s6ph9n8zg.3>, <http://dx.doi.org/10.17632/2h8cpszy26.1>, <http://dx.doi.org/10.17632/pb5xpv8z5r.1>

Keywords:

Battery diagnosis
Degradation modes
Deep learning
Convolutional Neural Networks

ABSTRACT

Lithium-ion batteries are ubiquitous in modern society with a presence in storage systems, electric cars, portable electronics, and many more applications. Consequently, to enable safe and reliable use of LIB systems, diagnosis and prognosis have become critical. Within the field of Artificial Intelligence, Deep Learning algorithms have achieved significant impacts for image or object recognition, yet their application for battery diagnosis is still at an early developing stage. In this paper, we propose a novel approach for battery degradation diagnosis based on the representation of battery data as images, in order to leverage the use of well-established Convolutional Neural Networks. Accuracy for diagnosis, via the quantification of degradation modes was tested on synthetic data. Our approach was shown to be more accurate than current methodologies with Root Mean Squared Errors around 2% on average for 1000 duty cycles compared to between 2.64 to 7.27% for other state-of-the-art algorithms. We also show that the proposed methodology adapts to various cell chemistries and constructive configurations, and validate its applicability to a real-world scenario with experimental data from commercial LIBs.

1. Introduction

Since their commercialization in the early 1990s, Lithium-ion batteries (LIBs) have been widely used in key commercial and industrial applications, ranging from portable electronic and transportation to storage systems [1]. Unfortunately, the performance of LIBs declines with operation because of parasitic reactions taking place at the positive and negative electrodes (PE and NE respectively) as well as in the electrolyte [2,3]. In addition, specific side reactions such as lithium plating may create safety hazards [4,5]. Both performance decline and safety issues present a major concern for deployed LIB systems, particularly where long-lasting reliable applications are critical. To assess LIB performance and to overall ensure safety and reliability, the determination of the state of health (SOH) and/or state of charge (SOC) is required [6], and numerous methodologies have been proposed in the literature [7–9]. These methodologies can be based on testing (both invasive and non-invasive), physics-based models, data-driven approaches, and hybrid methods [7,10]. Each method provides a set of key advantages, drawbacks, and range of applicability [11]. Model-based techniques tend to be more accurate, although they require extensive prior knowledge and often invasive tests while only non-invasive techniques must be considered for application-oriented

approaches. For this reason, data-driven methods have become popular in recent years, as they can model degradation features based solely on past records from which underlying causalities and correlations can be modeled [12]. Specifically, new methodologies based upon AI for the SOH [13,14] and SOC [15] have emerged thanks to the latest improvements in processor capabilities [16], communications [17,18], novel devices [19] and Artificial Intelligence (AI) [20,21]. It is expected that AI and Machine Learning (ML) approaches to compute SOH will have a profound influence on shaping the future LIB systems diagnosis and prognosis [22]. However, these methodologies are still in their early days [23–25] and critical issues remain to be addressed.

AI and Deep Learning (DL) have been exponentially applied to fields such as health [26], biology [27] or art [28]. Expansion of AI tools to such a wide range of fields has been possible because the problems to be solved can be highly abstracted from the field's domain. In battery research, however, this is not as easy as most problematics are technical and require an extensive knowledge of chemistry and physics [29].

Another barrier to the application of DL algorithms is the nature of the data. DL algorithms are typically oriented to work with 3 major categories: images, text and time series. For batteries, current, voltage and temperature records are usually obtained through measurements

* Corresponding author.

E-mail address: costanahuel@uniovi.es (N. Costa).

over charge/discharge cycles or a mix of cycling time and calendar aging time. Accordingly, most Deep Learning-battery related papers take the available variables and apply time series-oriented neural networks, typically, Recurrent Neural Networks (RNNs), in order to predict capacity loss [30–36]. Unfortunately, this approach often does not allow satisfactory prognosis because of the possible nonlinearity of the capacity loss [37]. A SOH tracking method is only useful if it can predict or accommodate batteries with nonlinear degradations. This could be done by investigating variations of capacity vs. voltage curves or their derivatives (electrochemical voltage spectroscopies, EVS). EVS were proven extremely successful for prognosis of nonlinear degradations with the tracking of degradation modes [38,39] but, unfortunately, they do not use time series and thus do not fit into any of the 3 categories mentioned above. EVS are typically requiring constant current cycle to be applied properly. For most applications, this is done during a reference performance test which could be considered independent of the rest of the aging experiment. This makes EVS data similar to images because they provide an independent representation of the variations of capacity vs. voltage. Therefore, analyzing this type of data as images would allow the use of Convolutional Neural Networks (CNNs), which are powerful models that have been applied in many fields with remarkable results since they are able to automatically find distinctive patterns within images without the need for expert knowledge.

In addition to the growing sophistication of the needed algorithms, the amount of data needed for training and validation is also critical as battery data generation is challenging and time-consuming [40]. The reality is that existing datasets, while providing invaluable information, are scarce and only provide data for a few cells under limited testing conditions [40]. This is a major barrier to the application of DL algorithms, where large amounts of data are required for the training process. Furthermore, models trained on these datasets can lead to a false sense of confidence in their performance, as the capacity loss decays linearly in most cases and tests are usually carried on a low variety of duty cycles which are quite often disconnected from real applications (e.g., constant current cycling). Real data will be much more sporadic and sub and supra-linear degradation will be common. Moreover, since cells are different among datasets, the data cannot be compounded and knowledge from one dataset cannot be transferred to another [41].

Recent initiatives like battery archive [42] or battery data genome [29] should make data more available in the near future. In addition, the apparition of synthetic cycles [29,43–45] in the public domain could alleviate the shortage of data, and in particular the lack of variety in the duty cycles, as they can deliver data under an infinity of different degradation scenarios. A dataset consisting of millions of voltage vs. capacity curves with a complete spectrum of degradation for three major battery chemistries: LiFePO₄, Nickel Aluminum Cobalt Oxide, and Nickel Manganese Cobalt Oxide 811 was recently released [44] and will be used in this work. An important difference with respect to previous datasets is that this synthetic data not only provides information about capacity loss but also about degradation modes. This enables diagnosability [29] and opens the gate for informed prognosisability [10].

Herein, we propose solutions to improve the application of DL algorithms to battery data. To this end, we sought a new representation of battery degradation data that would allow us to apply DL algorithms that have already been validated in other domains. Our representation consists of an image highlighting the differences between the EVS curves of a pristine and aged battery. Subsequently, we exploit the use of the HNEI's synthetic dataset [43] to train a Convolutional Neural Network (CNN) that predicts the battery health state based on its degradation mechanisms and not just capacity fade. This should enable the prediction of accelerated degradations. Finally, the adaptability of the method to deal with different cell configurations was validated with new synthetic data and subsequently on real cells.

2. Degradation mechanisms and degradation modes

Degradation in LIBs is the result of a complex interplay between physical and chemical mechanisms within the cell, that leads to capacity and power fade. Degradation is path-dependent and different usages (e.g., temperatures, load currents, duty cycles, depth of discharges, cut-off voltages, etc.) might inhibit or exacerbate specific degradation mechanisms [46,47]. Degradation mechanisms include solid electrolyte growth and decomposition, binder decomposition, graphite exfoliation, or grain isolation to name just a few [2,3,48]. Regardless of their origins and nature, the degradation mechanisms can only have a limited number of impacts on the electrochemical response [6,49]. These wide-ranging degradation mechanisms can be clustered into degradation modes, which are the loss of lithium inventory (LLI), the loss of active material (LAM) on the negative and positive electrodes (NE and PE respectively) and kinetic alterations.

Although degradation modes have been extensively reported in the literature [2,3,48], including experimental proofs [49], the underlying outcomes from the degradation modes on full cell effect are not always straightforward. For instance, LLI is generally the main source of degradation, is caused by parasitic reactions that consume lithium, is nearly always responsible for the entirety of capacity fade [49,50] and it can be modeled in half-cell configuration as NE “slippage” [51,52]. In contrast, LAM needs to be decomposed at the electrode or blend component level and is caused by changes in the availability of active mass for (de)intercalation. LAMs can be modeled as an individual “shrinkage” of the affected electrode or electrode components [50]. LAMs usually do not lead to a straight capacity loss in graphite-based batteries, hence that they may be referred to as “silent” or “hidden” modes. That is because LIBs yield an excess of relative capacity for each electrode outside the voltage window of the full cell. For the PE, that excess is the result of the NE lost during the SEI formation (i.e., the slippage) and of LLI. For the NE, that excess is there by design to protect against plating and it is also increased by LLI. Hence, most LAMs initially do not produce direct capacity loss, even if cell degradation is occurring. If LAMs eventually start to play a role in capacity loss, a second stage of accelerated aging arises [37]. The tracking and extrapolation of degradation modes were proven successful to forecast knees in the capacity loss [37].

Currently, several non-invasive testing methodologies to characterize battery degradation are available, including direct capacity testing characterization [53,54], high precision Coulomb counting [55, 56], electrochemical impedance spectroscopy [57,58] and pulse power tests [59,60], or EVS [38,39]. In particular, the incremental capacity (IC) technique has been proven extremely successful [38,39] for quantifying degradation modes. EVS detects gradual changes in cell behavior with great sensitivity by studying the evolution of minute changes of the voltage response with cycling.

The relation between battery degradation and changes in the voltage response can be explained by changes in the electrode matching, i.e., how the PE and NE relate to each other and modeled using a mechanistic model. These models can be used to establish degradation mapping [61,62] that allows to select Features of Interest (FOIs) [61, 63,64] which correspond to section of the signature especially sensitive to a specific degradation mode. Typical diagnosis methods must track FOIs and deconvolute their variations in detail to enable quantification of the degradation modes.

In most studies, this manual FOI tracking requires both an exhaustive analysis and expert knowledge. Data-driven methods could alleviate this issue and allow faster diagnosis by identifying patterns associated with degradation. However, data-driven methods are not always easy to implement and must be carefully designed to ensure the results have a physical meaning. For this reason, both expert knowledge and data-driven knowledge must evolve hand in hand. The first example of FOI analysis of large synthetic dataset was presented in [61]. Looking at data-driven methods, in [65] the dataset from [43] was used

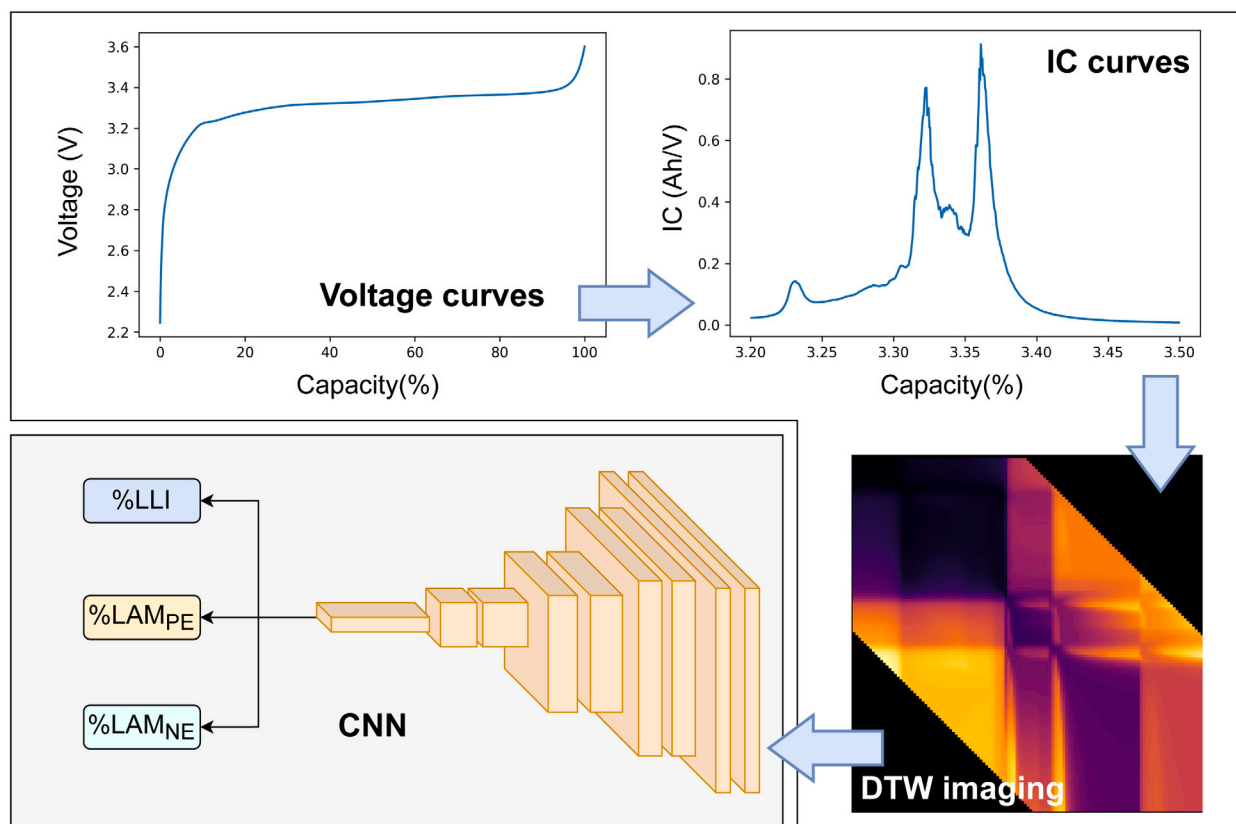


Fig. 1. Pipeline of the proposed solution. In the preprocessing step the IC curves are fed to the proposed algorithm to compute their representation as an image. Subsequently, the processed IC images are treated by a CNN that numerically quantifies the percentage of each degradation mode.

to train well-known Machine Learning methods such as Decision Trees and Random Forest regressors. In [45] the authors proposed a neural network composed of 1D convolutions for automatic classification and quantification of battery-aging modes. [66] proposed a method based on a simple multilayer feed-forward network for electrode-level Li-ion battery degradation diagnostics using EVS. The main limitation in these works is that the knowledge generated by the models cannot be extrapolated to new cell configurations. The method we propose aims to fill these gaps by transforming the voltage changes into images that reflect the degradation regardless of the cell configuration. This will create patterns that can be analyzed by traditional CNNs.

3. Deep learning approach

This section describes the proposed framework for material-based diagnosis in intercalation batteries (Li-ion and Na-ion). The process is summarized in Fig. 1 and consists of two separate steps. First, a pre-processing step, where charge data from the HNEI diagnosis dataset [43] is selected, converted to IC curves, and transformed into images. Second, a treatment step where the resulting images are used as inputs for a CNN trained to numerically identify and quantify the degradation modes.

3.1. Battery data to images — dynamic time warping

As stated in the introduction, one of the main difficulties towards AI to battery research is the type of data. Herein, this is circumvented by representing cell information as images. This opens up new opportunities for a consistent application of Deep Learning algorithms such as CNNs.

Dynamic Time Warping (DTW) [67] is an algorithm used to measure the similarity between two sequences. First, the Euclidean distance

between each pair of points between the two sequences is calculated in a matrix. Among these distances different warping paths can be found, that is, possible deformations that a sequence should follow in order to be as similar as possible to the other. The method quantifies the similarity between the sequences by finding the best warping path, which corresponds to the one with the smallest accumulated distance. Fig. 2a presents the example of the application of DTW to two sine waves, referred to as Sin #1, located in the left part of the grid and Sin #2, located in the upper part of the grid, which shows a small deformation in the second period. The best path found in the matrix is marked in blue and indicates that for the Sin #2 to be the same as Sin #1, the deformation to follow is to slightly raise the values between 15 and 20. The similarity between the two sequences can be quantified with the resulting distance, i.e., the accumulated Euclidean distances of the path, which is 0.1946. At the lower left and upper right corners, the values are marked as inf (infinite) because there are no deformation paths that extend that far, so they are not calculated in order to reduce computation time. The method developed originally for speech recognition, and it is widely used for classification and clustering tasks [68–72].

DTW was already applied to the estimation of Li-ion battery capacity [73–75] as well as for augmenting the data obtained from different operating conditions [76]. However, these works make use of the similarities found in the best warping paths, rather than the full matrix representation. In this work, we propose for the first time to use the full matrix, represented as a set of pixels (see Fig. 2b) and thus as an image. Instead of sine functions, IC curves will be used as sequences, one pristine and one aged. IC curves were chosen over straight voltage vs. capacity curves because the derivation enhances small voltage variations and as a result will provide images with more details. An image can be generated for each sample in the dataset and each image will thus represent the similarity between the corresponding IC curve

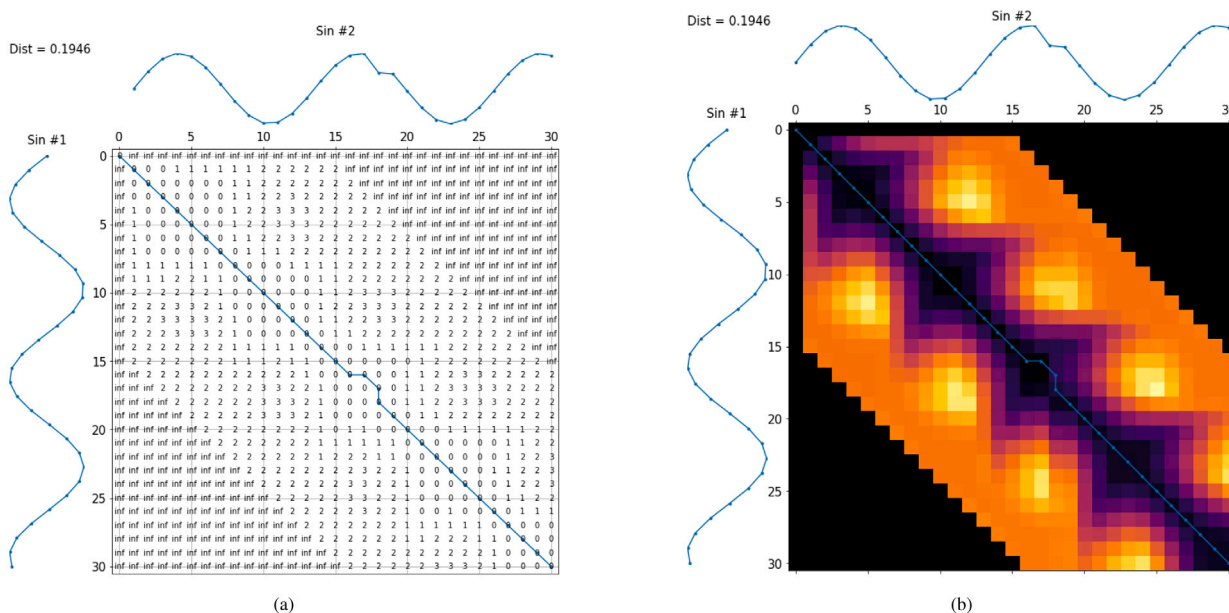


Fig. 2. Euclidean distance between each pair of points of the two sequences displayed on a grid (a). Every warping path represented as a set of pixels (b), note that the resolution depends directly on the length of the sequences, so the resulting image has the same resolution as the length of the sequences in (a), i.e., 30x30. In both images the optimal warping path is marked in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and the pristine one. Since each degradation path leads to a unique voltage response, it will also result in a unique image. As an example, Fig. 3 depicts the IC curves corresponding to 20% of each of the three degradation modes considered in the dataset: LLI, LAM_{PE} and LAM_{NE} (dashed lines) with the reference IC curve (solid lines) and their resulting images, labeled with the final DTW distance. This is to showcase that, just as the IC curves after different degradation are unique, the images are too. Straight IC curve plots cannot be used directly because they do not contain enough pixels with information. In our images, changes are reflected in shape, symmetry and colors. Note in the first degradation, LLI (Fig. 3a), the main peak located at 3.37 V is lost while in LAM_{PE} the peak that disappears is the minor one, located at 3.23 V. The images associated with these degradations also change, specifically in the intensity of the purple color, as well as in the symmetry, which is mainly lost in the first image, and consequently, the distance is greater, 0.77 vs. 0.31. In the LAM_{NE} degradation, the appearance of the peak at 3.45 V represents a sign of lithium plating in LFP cells [4]; On our image, this translates to the appearance of a lighter color band that coincides exactly with the position of the peak. The changes in this degradation are much more significant, and accordingly, the final calculated distance is greater: 1.53. In the end, just as with studying FOI variations, the degradation modes are decipherable from these unique images and so image processing algorithms such as CNNs can be undertaken.

A key property of these images is that they preserve the representation of the degradation modes regardless of the cell configuration. While the images were gathered from a dataset composed of synthetic curves, the differences between the pristine and aged IC curves should be similar for cells with slightly different cell configurations. In the mechanistic approach, a cell is defined by its active materials and two additional parameters, the loading ratio (LR), which corresponds to the electrode capacity ratio and the offset (OFS), which corresponds to their slippage compared to one another. Based on cell-to-cell variations studies [77], variations of LR by +/-0.2 and ΔOFS by +/- 2% were estimated possible within a batch. As an example, images associated with different cell configurations for the same degradation (20% of LLI) are presented in Fig. 4, with varied parameters to simulate cells from the same batch with slightly different properties (+/-0.01 for LR,

+/-1% for OFS). Visually, the three images are almost identical and this is confirmed by the final DTW distance that were 0.65, 0.66, 0.62, respectively to be compared to the 0.77, 0.31, 1.53 for LLI, LAM_{PE} and LAM_{NE} degradations on Fig. 3. This is a key factor when applying the procedure to batteries with different operating modes or cell configurations, especially since batteries from the same batch have some cell-to-cell variations and batteries from different manufacturer might not use the same materials, additives or loading. This differentiates our method from other models trained on synthetic data that might not be applicable to real data.

It is noteworthy mentioning that the resolution of the data used in this work is of 1001 points over the voltage window. To reduce computation time calculating the images the resolution was downscaled to a point every 2.3 mV per IC curve using a 1-D monotonic cubic interpolation with the Scipy Pchip Interpolator [78]. This kept the main features of interest intact while limiting the file size. As a result, the generated images, Figs. 3 and 4 included, are of 128 by 128 pixels. The dtadistance package [79] was used to compute the DTW matrix.

3.2. Model

With the new approach for the generation of high-quality training data established, attention can be set to the DL model. DL methods are sophisticated ML approaches that can handle high-dimensional data and are capable of automatically capturing underlying features to make accurate predictions. Convolutional Neural Networks (CNN) are a subset of DL models that are particularly well-suited for image recognition tasks and with multiple derived architectures such as AlexNet [80], U-NET [81] or the recent vision transformers [82].

CNNs consist of multiple layers of neurons. The structure of the proposed model is depicted in Fig. 5. The detailed description of each layer is as follows:

- Masking layer: this layer is used to mask data to be omitted by the next layer. In the DTW matrix, paths farther away from the diagonal lose importance (the inf values) and can be omitted.
- Convolutional layers (Conv1 to Conv4): these layers are composed each of a conv2D layer (light orange) and a Max-Pooling

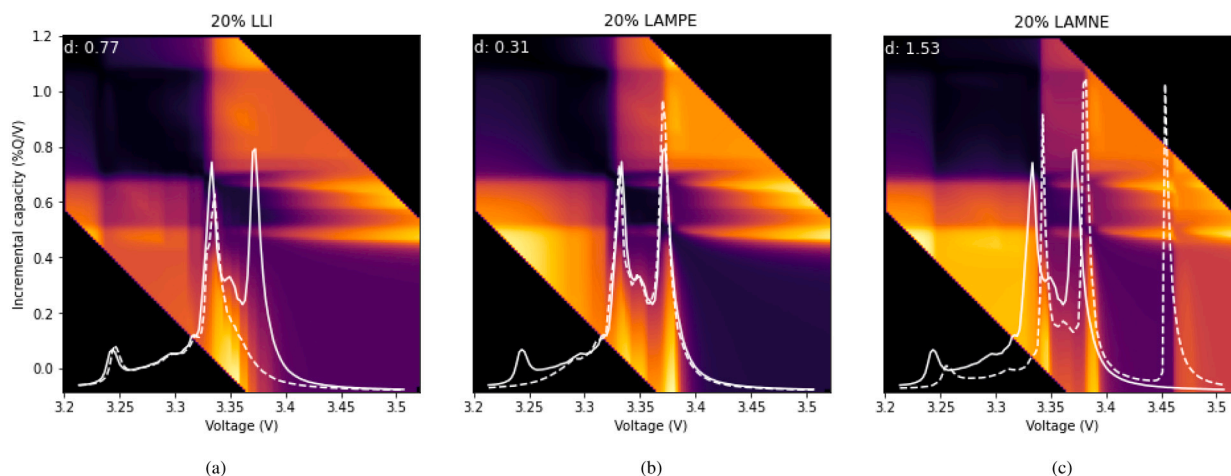


Fig. 3. IC signatures from the initial state (solid line) for each degradation in the dataset: LLI (a), LAMPE (b) and LLI(c) at 20% degradation (dashed line) together with the associated DTW image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

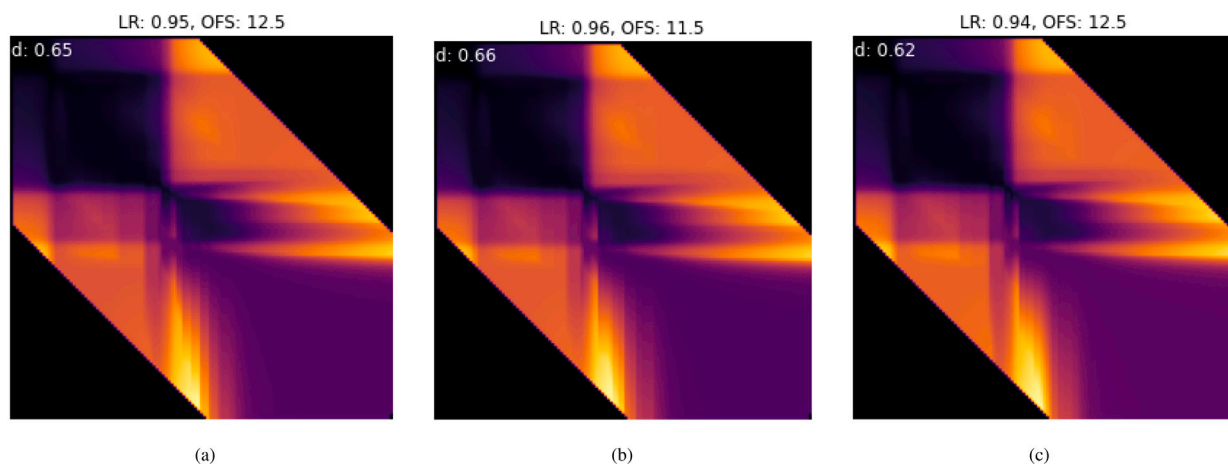


Fig. 4. DTW images for 20% LLI degradation for three different cell configurations.

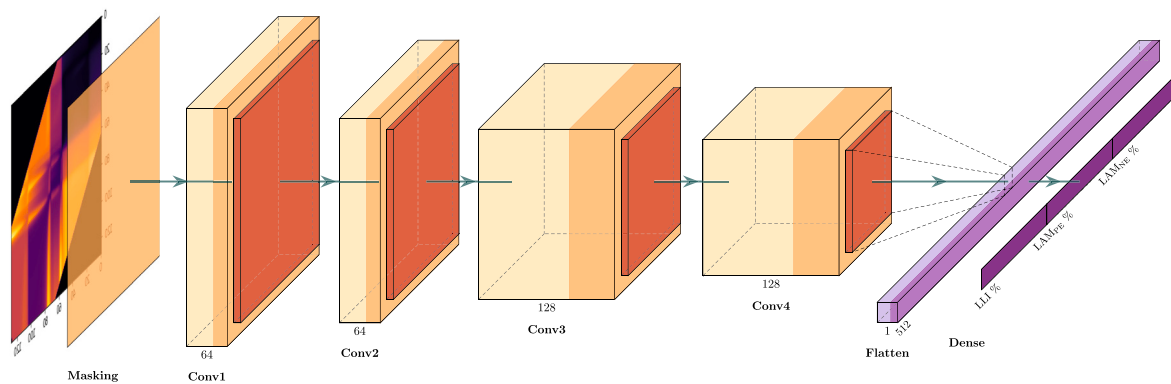


Fig. 5. Model architecture. Conv1 to Conv4 represent the convolution layers followed by the max pooling layers. The features extracted are condensed in a flattened layer from which the 3 degradation modes are predicted. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

layer (dark orange). The conv2D layers consist of multiple filters, which are applied to the image to highlight certain features that make the image unique such as the direction of the lines or their shape. The resulting images are known as feature maps. 64 filters are applied in each of the first two layers to obtain the features maps that mainly characterize the image, while in the last two layers more filters are needed (128 each) to capture finer details

like color intensity or brightness. The Max-Pooling layer reduces the spatial size of the feature maps and learns to ignore irrelevant and redundant information, that is why the dimension of the blocks is reduced in each layer.

- Flatten layer: after the convolution and max pooling flow, the shape of the matrices is flattened to a single vector containing all the information needed for predictions.

- Dense layer: this layer applies a sigmoid activation function to obtain a value between 0 and 1 representing the percentage prediction of each of the degradation modes.

The activation function used after every convolution is ReLU and also a dropout layer was included to randomly set input units to 0, which can generally help to prevent overfitting. Nevertheless, we found that including this layer led to a premature regularization and as a result to a sub-optimal model, therefore it was not used in the final model.

The WandB framework [83] was used to find the optimal configuration of filters and layers. In addition, to achieve the best possible performance of our model, callbacks to relegate the training stop condition to the validation error were used instead of the number of epochs.

In summary, each sample in the dataset consists of an image reflecting the differences between an aged IC curve and the pristine one. These images, along with the associated diagnosis, are fed to the model which learns the features that characterize each type of degradation and then compress all the knowledge in the last layer to predict the percentages of each degradation mode.

4. Experimental design

The training data used in this work is publicly available [44] and is composed of more than 700,000 unique voltage vs capacity C/25 charge curves each for LFP, NMC, and NCA. They were generated with different combinations of LLI, LAM_{PE} and LAM_{NE} at a resolution of at least 0.85% between 0 and 85%, encapsulating the full spectrum of degradation. The resulting capacity losses were capped to 85%.

The end of life (EoL) of a battery is usually driven by the application, however, usage after 40% capacity loss is rarely allowed. In practice, the benefit of diagnosing a battery lies in predicting its remaining useful life or, if it is partially deteriorated, knowing whether it can be put to a second use. Since our goal is to provide a methodology to detect subtle evidence to forecast durability, data above 40% capacity loss was not used for training.

The choice of the testing data was carefully made. To statistically validate the performance of a ML algorithm, it is common to divide the data into two independent parts: the first is used for training and the second for testing. A possible approach would be to consider one resolution for the training set and another for the test set to test the interpolation capability of the model. The main drawback of such a setting is that the test set is a sparser subset of the same initial data, therefore training is not as complete as it could have been and test sets are not independent. As a consequence, the model accuracy may be optimistic because samples in the test set are close to those in the training set, a well-known problem in ML called overfitting [84].

To avoid this problem and to differentiate whether the model has actually learned the degradations we elected to calculate new synthetic data sets for each of the chemistries with slightly different configurations than the cells in the training sets. This allows having completely independent training and validation sets to provide a benchmark for the fair and equal evaluation of different ML models. Details about the test sets, each consisting of 1000 duty cycles, can be seen in Table 1. Fig. 9(a) in Appendix A shows the capacity losses associated with the duty cycles calculated in [44]. As prognosis is the ultimate goal, we decided to select a 1000 subset of these duty cycles to provide a test set that can be used both for diagnosis (as is the case of this work) and for prognosis (for future works). Combinations of the [LLI, LAM_{PE} , LAM_{NE}] degradations were selected to generate the duty cycles for the new cell configurations to show a wide range of both sublinear and supralinear degradations. Emphasis is placed on the latter due to their interest in identifying knees in capacity loss, correlated with the so-called silent modes. Calculations are done for the following cycles: 10, 50, 100, 200, 400 and 1000 (Fig. 9(c)) with capacity losses up to 40% (Fig. 9(d)).

Table 1

Details about test sets. Three cells, labeled as C1, C2, C3, were generated using the 'alawa toolbox [50] for each chemistry. The values of LR (Loading Ratio) and OFS (offset) with which they were generated are included. Parameters used for the training data are also added to highlight the differences with respect to the test sets.

	Training data		C1		C2		C3	
	LR	OFS	LR	OFS	LR	OFS	LR	OFS
LFP	0.95	12.5	0.96	11.5	0.94	12.5	0.95	11.5
NCA	1.05	1.5	1.06	0.5	1.04	2.5	1.05	0.5
NMC	0.90	10	0.91	9	0.89	11	0.90	9

5. Results

The experimental validation of the proposed framework will be performed first on synthetic data, then on real data. In both cases, our framework will be compared to the state-of-the-art for degradation modes quantification. The metric chosen for evaluation, defined in Appendix B, is the Root Mean Squared Percentage Error (RMSPE).

5.1. Validation on synthetic cycles

In this section, the performance of our method was compared to state-of-the-art methods applied to the same synthetic dataset with different cell configurations.

Results for degradation mode quantification for all methods are presented in Tables 2 and 3 for the LFP cells. Results for NCA and NMC are included in Appendix C.1. Among the tested methods were the works described at the end of Section 2 and the one described in this work, labeled in the tables as "RF" for the Random forest regressor [65], "1DConv" for the 1D convolutional neural network [45], "FNN" for the Feed-forward neural network [66] and "DTW-CNN" for our Dynamic Time Warping-convolutional neural network approach. It should be noted that only our method uses DTW images while the other approaches use the IC curves directly. In addition, a CNN is also used in [45], but 1D convolutions are applied, which are not suitable for images. These methods did not provide any public code implementation, consequently, the steps described in their corresponding papers were followed to reproduce their models adapted to these test sets (see Appendix C.2 and <https://github.com/NahuelCostaCortez/DTW-Li-ion-Diagnosis> for details). Table 2 lists the diagnosis accuracy (by the means of RMSPE values) for the quantification of the three degradation modes at six different cycles (10, 50, 100, 200, 400 and 1000) for the three different LFP cell configurations (C1 to C3 in Table 1). The best predictions are highlighted in bold. Overall, the approach presented in this work clearly outperforms the others with an average error of 2.00% (see Table 3). Yet, there are certain cycles where other methods perform slightly better. This may be due to some bias during training that may lead to unbalanced predictions and, as a consequence, to reasonable performance in one degradation mode but not in the others. For instance, the predictions of "1DConv" for cycle 400 in C1. Numerically in LAM_{PE} it has a better result than our approach (3.38% vs 3.59%), however for LLI (1.68% vs 1.31%) and especially for LAM_{NE} (2.83% vs 1.93%) the performance is considerably worse. This is quickly identified in the standard deviation, where our model with a value of 1.96 shows a lower dispersion compared to the other models. Tables 4 and 6 in Appendix C.1 present the same analysis for the NCA and NMC cells, respectively. The results are similar with an average error of 2.03% (see Table 7) for NMC, compared to errors from 2.56 to 7.27% for the other methods. The approach seems to perform better for NCA cells with an average error of 1.11% (see Table 5), compared to errors from 1.31 to 7.01% for the other methods.

The main reason behind the consistent estimations in our approach is that the representation of degradations in the images is largely preserved between different cell configurations, something that is not the case in pure IC curve processing, where peaks, despite having similar

Table 2
RMSPE results for each degradation mode and cycle for the LFP test set.

		LLI					LAMPE					LAMNE							
FNN [66]	C1	1.89	1.93	2.00	1.82	1.67	3.91	2.53	2.90	3.09	3.28	3.58	11.11	2.30	2.32	2.32	2.10	2.15	6.31
	C2	2.06	2.16	2.23	1.81	1.55	3.67	3.30	2.94	2.94	2.73	3.46	11.32	3.41	3.29	3.28	2.77	2.35	6.19
	C3	1.45	1.93	1.88	1.68	1.73	4.02	2.27	3.14	3.40	3.52	3.78	11.31	3.04	3.06	2.98	2.64	2.44	6.37
RF [65]	C1	6.32	5.69	4.94	3.62	3.23	9.21	5.89	5.13	4.26	3.15	5.16	9.13	7.00	6.06	5.02	3.82	6.24	11.83
	C2	6.32	5.69	4.94	3.64	3.14	9.22	5.89	5.13	4.26	3.17	4.97	9.79	7.00	6.06	5.02	3.82	6.38	11.55
	C3	6.32	5.69	4.94	3.62	3.20	9.13	5.89	5.13	4.26	3.15	5.07	9.37	7.00	6.06	5.02	3.82	6.18	11.66
1DConv [45]	C1	1.18	0.95	0.73	1.06	1.68	3.21	1.90	1.23	1.80	2.80	3.38	10.73	1.18	1.33	1.27	1.71	2.83	6.60
	C2	0.63	0.59	0.86	1.11	1.62	3.15	0.41	1.28	2.76	2.62	3.50	10.85	2.05	1.83	2.03	2.36	2.86	6.58
	C3	1.95	0.89	0.60	0.96	1.75	3.35	2.08	1.15	2.01	2.95	3.44	10.86	2.86	1.97	1.59	2.07	2.93	6.61
DTW-CNN	C1	0.14	0.53	0.72	1.16	1.31	2.47	0.96	0.98	1.82	2.67	3.59	8.64	0.17	0.70	1.40	1.98	1.93	3.86
	C2	0.44	0.84	0.91	1.18	1.32	2.15	0.78	2.06	2.76	3.22	3.92	8.89	0.21	0.57	0.80	1.11	1.41	4.01
	C3	0.80	0.56	0.56	0.95	1.12	2.58	2.30	1.32	2.03	2.72	3.67	8.63	0.59	0.55	1.00	1.43	1.64	3.94
		10	50	100	200	400	1000	10	50	100	200	400	1000	10	50	100	200	400	1000

Table 3
RMSPE results summary for the LFP test set calculated as the average and the standard deviation of predictions in all cycles for all cells.

	FNN	RF	1DConv	DTW-CNN
Mean \pm std	3.32 \pm 2.21	5.87 \pm 2.23	2.64 \pm 2.42	2.00 \pm 1.96

morphologies, suffer from shifts that can cause models to misleading predictions.

The method performs remarkably well and surpassed the tested state-of-the-art approaches; however, it still has room for improvement. For instance, note the large errors in later cycles (400 and 1000), which correspond to degradations around 40% of capacity loss. Although these errors are still much lower than in the other methods the estimations for these cycles could be further improved. Some other comparative tests could also be added to the discussion; however, the main objective of this work was to enable the use of images to exploit the potential of CNNs. We have developed and used one of the many architectures that can be found in the literature, but predictions could be improved by other newer and more robust models. Furthermore, the key factor, and in fact, one of the essential features of Deep Learning, is precisely reusability. The knowledge of large models trained on a specific task can be transferred to a new, similarly related task. This is known as Transfer Learning [85] and is especially useful when little data is available, instead of training a model from scratch it can leverage the knowledge generated by a pre-trained model for fine-tuning on the available data. This technique is mainly focused on images, therefore the preprocessing we propose, besides providing an adaptive method, also allows the application of this technique: we have pre-trained a large model on the training set (DTW-CNN), so its knowledge can be now used by other models on the test sets or on new data. This path, as well as other complementary ones such as the explainability of the models or the choice of the CNN architecture, will be explored in future work.

Finally, to demonstrate the performance of the model in a more realistic application context, we provide a demo in <https://huggingface.co/spaces/NahuelCosta/DTW-CNN>. The cycles associated with the three LFP test cells can be selected to display their IC curves, the corresponding DTW image and the final diagnosis given as the percentage of each predicted degradation mode.

5.2. Validation on real battery data

As demonstrated above, one of the strengths of the model is its applicability to cells with configurations other than those seen during training. This also includes real cells so in this section our model was tested on cycling data from two commercial high-power graphite//LFP cells manufactured by A123 Systems (ANR26650M1, 2.3 Ah) that have been previously studied. The cells will be referred as CReal#1

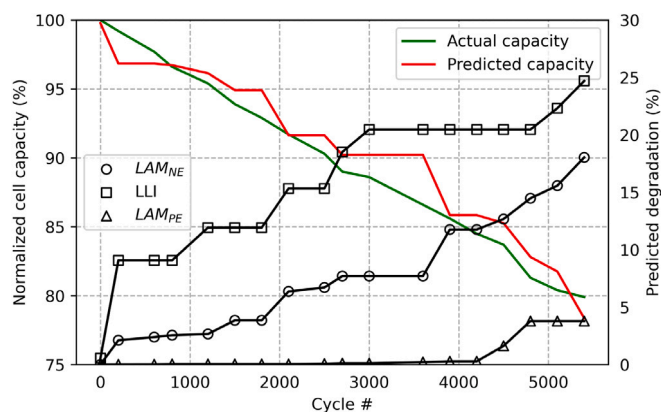


Fig. 6. Model predictions for Cell#1 for every available cycle.

and CReal#2 for simplicity; CReal#1 was tested under multistage fast charging [86] while CReal#2 was tested under dynamic stress test (DST) driving schedule [38]. In these studies, the degradation modes were quantified using the 'alawa toolbox [50]. It should be noted that the toolbox uses the same mechanistic model than the one that generated the training data used in this study. Therefore, in the end, the predictions of our model are an automatic way of making the same diagnosis without relying on prior knowledge in the field.

CReal#1 was cycled to simulate fast charge and discharge conditions. Every 300 cycles, a reference performance test (RPT) was done to determine the state of health (SOH) of each cell. Fig. 6 shows the model predictions for each of the voltage curves of the available cycles. The diagnosis established by our model in terms of degradation mode quantification and capacity loss estimation matched the experimental observations. The capacity estimation adjusted remarkably well to the evolution along the cycles and for the degradation modes, despite some fluctuations, they tended to follow a linear degradation with LAM_{NE} and LLI being the main actors, while the effect of LAM_{PE} is almost negligible. These predictions meet the results reported in [86], where the degradation was concluded to be caused by a linear loss of LLI of 0,0032% per cycle followed by a linear loss of LAM_{NE} of 0,0022%.

CReal#2 was cycled differently to study the impact of fast charge with an EV type discharge rather than constant current. The degradation path was quite different than of CReal#1 and significant Li plating occurred. Plating is considered one of the most detrimental phenomena in lithium-ion batteries, as it increases cell degradation and might lead to safety issues. RPTs were again performed every 300 full DST cycles. Model predictions are presented in Fig. 7 together with the diagnosis reported in [38]. Despite the few cycles available, the capacity estimation is quite correct. Looking at the degradation modes, their evolution is more complex than of CReal#1. LAM_{NE} is calculated

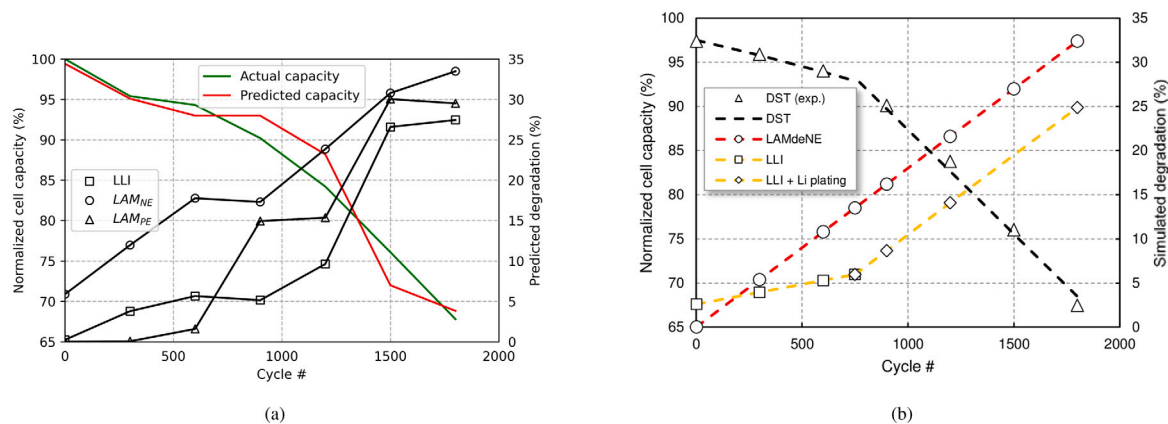


Fig. 7. Model predictions for Cell#2 for every available cycle (a). Diagnosis estimated in [38] (b).

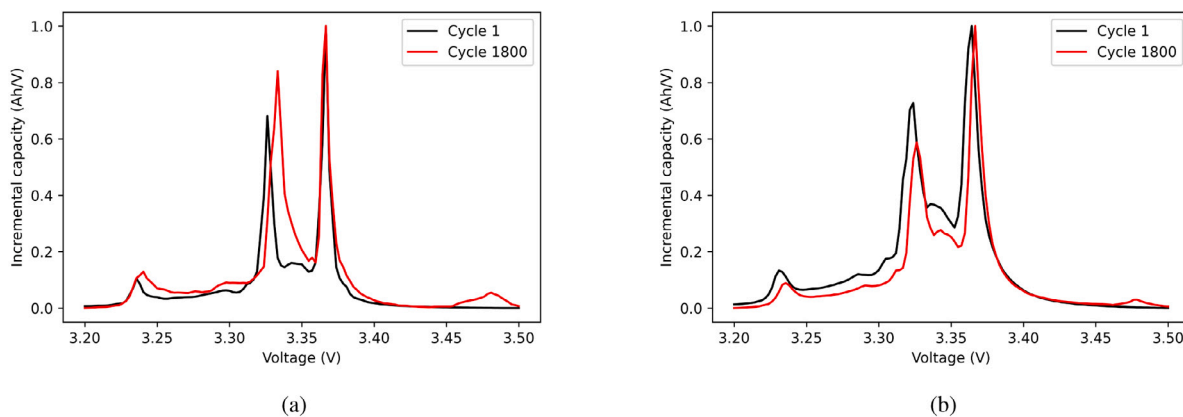


Fig. 8. IC curves for cycles 1 and 1800 in the real cell (a) and in the synthetic cell generated for the predicted degradation percentages (b).

Table 4

RMSPE results for each degradation mode and cycle for the NCA test set.

		LLI					LAMPE					LAMNE							
FNN [66]	C1	0.13	0.69	1.05	1.17	0.89	1.19	1.27	1.66	1.55	1.67	1.78	1.41	0.21	1.10	2.01	2.37	1.82	2.48
	C2	0.11	0.51	0.67	0.81	0.82	1.30	0.52	0.64	0.76	0.87	0.84	1.09	0.19	1.21	2.18	2.96	2.56	2.87
	C3	0.16	0.62	0.90	1.04	0.77	1.19	1.98	1.51	1.36	1.45	1.52	1.19	0.19	1.20	2.28	3.07	2.47	2.72
RF [65]	C1	5.58	8.58	8.04	8.65	8.90	9.35	5.08	7.77	7.10	7.15	6.55	11.58	5.64	5.82	4.82	4.63	7.16	13.34
	C2	4.58	4.41	5.67	6.81	7.87	9.23	4.16	3.93	4.95	5.61	6.41	11.68	5.40	4.56	4.07	3.8	7.35	13.75
	C3	5.43	7.53	6.97	8.05	8.71	9.29	4.95	6.76	6.13	6.60	6.73	11.58	5.60	5.42	4.50	4.12	6.82	13.48
1DConv [45]	C1	0.32	0.33	0.56	0.77	0.80	1.5	2.31	2.07	2.05	2.01	1.90	1.43	0.37	0.74	1.36	1.99	1.95	3.06
	C2	0.33	0.42	0.44	0.53	0.69	1.10	0.94	0.79	0.83	0.82	0.79	0.87	0.22	1.09	1.69	2.56	2.77	3.44
	C3	0.34	0.36	0.47	0.68	0.71	1.36	2.08	1.74	1.72	1.67	1.57	1.18	0.20	1.01	1.75	2.69	2.68	3.37
DTW-CNN	C1	0.35	0.34	0.56	0.99	1.43	2.36	0.47	0.64	0.67	0.96	1.41	2.06	1.79	1.50	1.40	1.41	1.52	2.57
	C2	0.69	0.93	1.03	1.24	1.80	1.93	0.21	0.43	0.63	0.73	0.94	1.43	0.27	0.69	0.78	1.06	1.37	2.68
	C3	0.12	0.44	0.72	0.97	1.28	2.05	0.60	0.40	0.65	1.03	1.40	2.00	0.40	0.59	0.69	1.00	1.39	3.16
		10	50	100	200	400	1000	10	50	100	200	400	1000	10	50	100	200	400	1000

as linear with a higher slope than of CReal#1. LLI started linear and then accelerated after 700 cycles. LAM_{PE} evolution fluctuates a lot but seems rather linear with a much higher slope than of CReal#1. The overall evolution of the degradation modes matched quite well the analysis performed in [38] where the high pace of LAM_{NE} induced some Li plating of which most passivated in LLI. The main difference is the LAM_{PE} estimation. LAM_{PE} is extremely difficult to quantify for LFP type cells as was considered in [38] as it was inducing neither capacity loss nor was at the origin of the knee.

In contrast, our model took every possible degradation into account during training and predicted some LAM_{PE}. The true extend of it could not be verified on the electrochemical data as no post-mortem study was carried on the aged cell. Fig. 8 shows the IC curves for cycles 1

Table 5

RMSPE results summary for the NCA test set calculated as the average and the standard deviation of predictions in all cycles for all cells.

	FNN	RF	1DCov	DTW-CNN
Mean ± std	1.31 ± 0.77	7.01 ± 2.51	1.32 ± 0.86	1.11 ± 0.67

and 1800 in the real cell and in the synthetic cell generated for the predicted degradation percentages. It is expected that the curves will not be exactly the same due to the differences between the simulation and the real data, the interest lies in the peak appearing at around 3.47 V, which is known to imply some reversible plating in the cell.

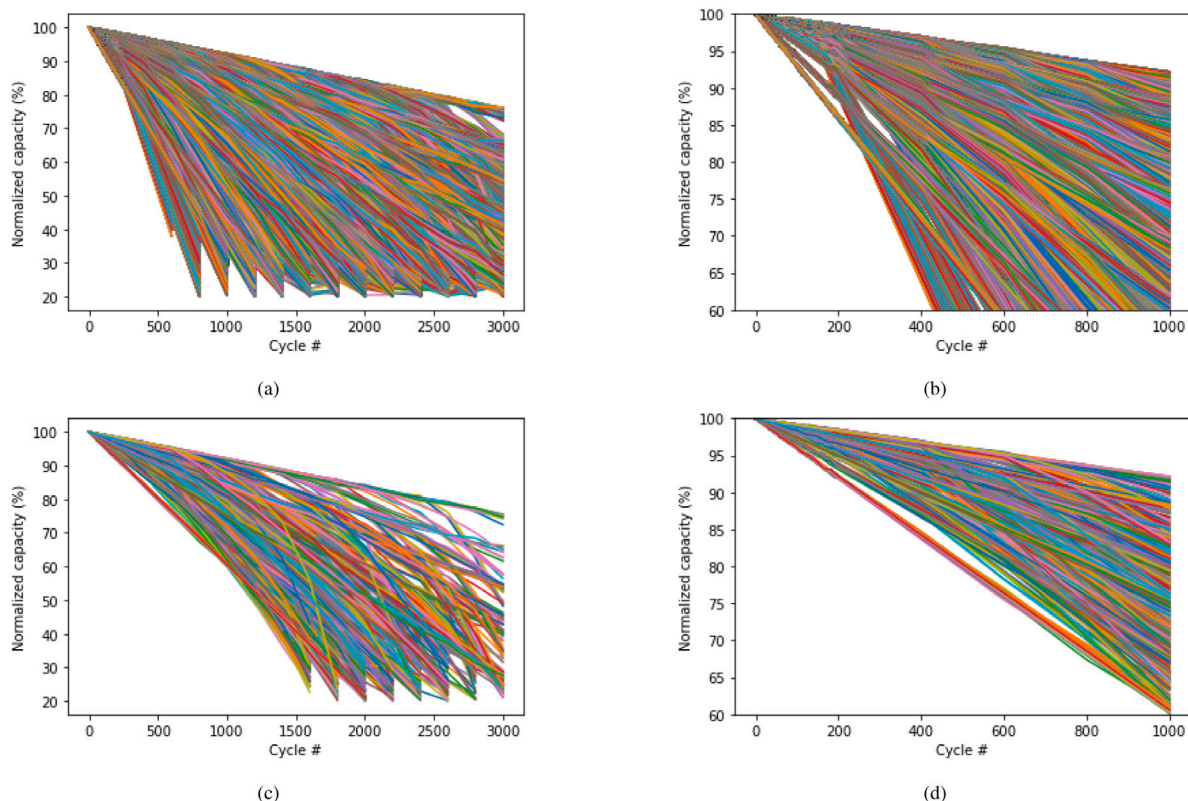


Fig. 9. Evolution of capacity loss over the 127662 duty cycles presented in [44] for the LFP cell (a). Same plot as in (a) for the first 1000 cycles and at 60% of capacity (b). Evolution of capacity loss over the 1000 duty cycles selected for test (c). Same plot as in (c) for the first 1000 cycles and at 60% of capacity (d).

Table 6

RMSPE results for each degradation mode and cycle for the NMC test set.

		LLI					LAMPE					LAMNE							
FNN [66]	C1	0.39	0.48	0.84	1.18	2.00	2.12	3.02	3.57	3.54	3.28	2.98	3.08	0.26	1.16	2.27	3.68	4.91	7.18
	C2	2.71	2.60	2.48	2.25	1.91	2.34	1.29	1.24	1.51	1.30	1.21	2.41	5.43	5.30	5.12	4.77	5.07	8.23
	C3	0.29	0.30	0.60	0.93	1.91	1.99	2.69	2.90	3.09	2.95	2.69	2.97	0.17	1.11	2.20	3.43	4.51	6.76
RF [65]	C1	8.30	7.52	6.57	4.83	3.79	11.41	6.91	6.16	5.29	3.96	5.96	9.42	9.25	8.37	7.35	5.72	5.62	14.37
	C2	8.30	7.52	6.57	4.83	3.49	11.68	6.91	6.16	5.29	3.96	5.36	8.96	9.25	8.37	7.35	5.72	5.63	15.26
	C3	8.30	7.52	6.57	4.83	3.73	11.41	6.91	6.16	5.29	3.96	6.11	9.54	9.25	8.37	7.35	5.72	5.67	14.25
1DConv [45]	C1	0.47	0.40	0.52	0.73	1.23	2.68	4.01	4.02	3.95	3.83	3.91	3.60	0.29	1.18	2.11	3.02	3.75	7.82
	C2	1.96	2.00	1.94	1.72	1.59	2.47	0.57	0.38	0.41	0.48	1.14	1.80	4.54	4.35	4.33	4.39	4.84	7.91
	C3	0.47	0.35	0.54	0.70	1.27	2.76	3.90	3.74	3.61	3.55	3.54	3.34	0.19	1.03	1.80	2.43	3.17	7.36
DTW-CNN	C1	0.65	0.67	0.72	0.64	0.96	4.32	1.70	1.92	1.99	1.92	2.29	3.91	0.44	0.48	1.06	2.18	3.55	7.58
	C2	1.00	1.00	0.82	0.74	1.23	3.00	0.80	0.59	0.52	0.86	2.36	4.54	2.19	1.94	1.91	2.91	7.48	
	C3	0.76	0.90	0.82	0.64	1.17	4.47	1.86	2.03	2.00	1.85	2.05	3.78	0.77	0.79	0.93	1.82	3.28	7.37
		10	50	100	200	400	1000	10	50	100	200	400	1000	10	50	100	200	400	1000

Table 7

RMSPE results summary for the NMC test set calculated as the average and the standard deviation of predictions in all cycles for all cells.

	FNN	RF	1DConv	DTW-CNN
Mean ± std	2.68 ± 1.81	7.27 ± 2.66	2.56 ± 1.90	2.03 ± 1.72

Hence, model predictions also suggest that the occurring degradation mechanism is irreversible lithium plating.

All things considered, it appears that the model adapted well to conditions different from those seen during training, with the predictions meeting to a large extent the previous diagnosis. Since different combinations of degradation modes can lead to the same capacity loss, which gives room to confusion in results interpretation, the model needs not only to estimate a concrete percentage of the degradation modes to offer a possible range of degradations with a certain degree

of confidence. This only can enable prognosis [44], which will be the topic of future work.

6. Concluding remarks and future work

Data-driven methods are a promising avenue for lithium-ion battery diagnostics and prognostics. Thus, efforts to use AI for state estimation and lifetime prediction have emerged in recent years. However, the application of modern AI algorithms is still at an early stage. In this work, we proposed a novel method for battery degradation diagnosis, that represents battery data as images, with the aim of enabling the use of powerful AI models in this domain. Especially, the IC curves from HNEI’s synthetic datasets were used to train a CNN that successfully predicts the main degradation mechanisms on several commercial cells of different chemistries and with different characteristics. The performance of the model was compared to other state-of-the-art methods, where the superiority of our approach was clearly demonstrated, with

Table 8

Configurations of the models used. In [66] and [45] “layers config” refers to the number of neurons per layer. Also, in [45], the authors built a 1D CNN to quantify LLI only while here we used the same model to predict the three degradation modes.

Method	Hyperparameters		
FNN [66]	<i>num_layers</i> 3	<i>layers</i> Fully Connected layers	<i>layers config</i> 64 × 32 × 3
RF [65]	<i>max_depth</i> 10	<i>random_state</i> 42	<i>n_estimators</i> 100
1DConv [45]	<i>num_layers</i> 5	<i>layers</i> 2 1D CNN layers and 3 Fully Connected layers	<i>layers config</i> 32 × 32 × 128 × 64 × 3

RMSPE errors around 2% in average for 1000 duty cycles compared to between 2.64 to 7.27% for the other tested methods. The successful performance of the model is largely due to its adaptive nature to different cell configurations. To validate this claim, the model was also tested on real cells, where the diagnosis corresponded to a large extent with previously existing studies on the subject. This opens up new opportunities for collaboration between AI and battery research.

In future works we aim to explore the use of Transfer Learning as well as the suitability of the approach for prognosis, evaluating the evolution of the peaks throughout the cycles, rather than independent cycle diagnosis. In addition, for this study the data used were only from charge cycles, however, considering discharge data is also of great interest for a more complete diagnosis. Lastly, our interests are aligned with the extension of the study of degradation modes, a key subject to contribute to the electrochemical understanding of cell deterioration.

CRedit authorship contribution statement

N. Costa: Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **L. Sánchez:** Writing – original draft, Writing – review & editing. **D. Anseán:** Writing – original draft, Writing – review & editing. **M. Dubarry:** Conception and design of study, Acquisition of data, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used in this study is available for download. LFP: <http://dx.doi.org/10.17632/bs2j56pn7y.3> and <http://dx.doi.org/10.17632/6s6ph9n8zg.3>, NCA: <http://dx.doi.org/10.17632/2h8cpszy26.1>, NMC: <http://dx.doi.org/10.17632/pb5xpv8z5r.1>.

Acknowledgments

This work has been partially supported by the Ministry of Economy, Industry and Competitiveness (“Ministerio de Economía, Industria y Competitividad”), Spain from Spain/FEDER under grant PID2020-112726-RB-I00 and by Principado de Asturias, grant SV-PA-21-AYUD/2021/50994. M. D. is thankful to the Office of Naval Research, USA APRISES grants N00014-18-1-2127 and N00014-19-1-2159 for funding. All authors approved the version of the manuscript to be published.

Appendix A. Duty cycles selection

See Fig. 9.

Appendix B. Model evaluation

In regression problems, the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics, or their versions expressed as a percentage MAPE and RMSPE are commonly used. MAE measures the mean magnitude of errors in a set of model estimates while RMSE is the root of the averaged quadratic score. In RMSE the errors are squared before averaged, whereas in MAE all individual differences are equally weighted in the mean. This makes RMSE more sensitive to large errors. Consequently, RMSE is considered more effective for testing model performance, especially when large errors are undesirable. For simplicity in the interpretation of results, RMSE expressed as a percentage (RMSPE) is chosen:

$$RMSPE = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2} \times 100 \quad (1)$$

However, using this metric requires dividing the prediction error by the actual value. In the dataset used in this problem, there are combinations where the actual value (the degradation mode) is simply 0, so the calculation would be invalid. Because of this, the denominator is replaced by the nominal capacity of the cell understood as the initial capacity expressed in percentage, i.e., 100%. Predicted values are already given in percentages (for the degradation modes) therefore the definitive metric used is:

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

Appendix C. Supplementary tables

C.1. RMSPE results for NCA and nmc

See Tables 4–7.

C.2. Summary of model configurations used

See Table 8.

Code availability

All models and experiments were implemented in TensorFlow [87]. Further details regarding the experimental setup and the source code to reproduce the experimental results are available in the following public git repository: <https://github.com/NahuelCostaCortez/DTW-Lion-Diagnosis>.

References

- [1] G.E. Blomgren, The development and future of lithium ion batteries, *J. Electrochem. Soc.* 164 (1) (2016) A5019.
- [2] M.R. Palacin, Understanding ageing in Li-ion batteries: A chemical issue, *Chem. Soc. Rev.* 47 (13) (2018) 4924–4933.
- [3] X. Han, L. Lu, Y. Zheng, X. Feng, Z. Li, J. Li, M. Ouyang, A review on the key issues of the lithium ion battery degradation among the whole life cycle, *ETransportation* 1 (2019) 100005.
- [4] T. Waldmann, B.-I. Hogg, M. Wohlfahrt-Mehrens, Li plating as unwanted side reaction in commercial Li-ion cells—A review, *J. Power Sources* 384 (2018) 107–124.
- [5] X. Lin, K. Khosravinia, X. Hu, J. Li, W. Lu, Lithium plating mechanism, detection, and mitigation in lithium-ion batteries, *Prog. Energy Combust. Sci.* 87 (2021) 100953.
- [6] M. Dubarry, G. Baure, D. Anseán, Perspective on state-of-health determination in lithium-ion batteries, *J. Electrochem. Energy Convers. Storage* 17 (4) (2020).
- [7] S. Yang, C. Zhang, J. Jiang, W. Zhang, L. Zhang, Y. Wang, Review on state-of-health of lithium-ion batteries: Characterizations, estimations and applications, *J. Cleaner Prod.* 314 (2021) 128015.
- [8] A. Barai, K. Uddin, M. Dubarry, L. Somerville, A. McGordon, P. Jennings, I. Bloom, A comparison of methodologies for the non-invasive characterisation of commercial Li-ion cells, *Prog. Energy Combust. Sci.* 72 (2019) 1–31.
- [9] M. Dubarry, V. Svoboda, R. Hwu, B.Y. Liaw, Incremental capacity analysis and close-to-equilibrium OCV measurements to quantify capacity fade in commercial rechargeable lithium batteries, *Electrochem. Solid-State Lett.* 9 (10) (2006) A454.
- [10] X. Hu, L. Xu, X. Lin, M. Pecht, Battery lifetime prognostics, *Joule* 4 (2) (2020) 310–346.
- [11] H. Rauf, M. Khalid, N. Arshad, Machine learning in state of health and remaining useful life estimation: Theoretical and technological development in battery degradation modelling, *Renew. Sustain. Energy Rev.* 156 (2022) 111903.
- [12] D.N. How, M. Hannan, M.H. Lipu, P.J. Ker, State of charge estimation for lithium-ion batteries using model-based and data-driven methods: A review, *Ieee Access* 7 (2019) 136116–136136.
- [13] Q. Li, D. Li, K. Zhao, L. Wang, K. Wang, State of health estimation of lithium-ion battery based on improved ant lion optimization and support vector regression, *J. Energy Storage* 50 (2022) 104215.
- [14] H. Sun, J. Sun, K. Zhao, L. Wang, K. Wang, Data-driven ICA-bi-LSTM-combined lithium battery SOH estimation, *Math. Probl. Eng.* 2022 (2022).
- [15] Z. Cui, L. Wang, Q. Li, K. Wang, A comprehensive review on the state of charge estimation for lithium-ion battery based on neural network, *Int. J. Energy Res.* 46 (5) (2022) 5423–5440.
- [16] T. Baji, Evolution of the GPU device widely used in AI and massive parallel processing, in: 2018 IEEE 2nd Electron Devices Technology and Manufacturing Conference, EDTM, IEEE, 2018, pp. 7–9.
- [17] W.Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, A. Ahmed, Edge computing: A survey, *Future Gener. Comput. Syst.* 97 (2019) 219–235.
- [18] A. Dogra, R.K. Jha, S. Jain, A survey on beyond 5G network with the advent of 6G: Architecture and emerging technologies, *Ieee Access* 9 (2020) 67512–67547.
- [19] D. Li, S. Li, S. Zhang, J. Sun, L. Wang, K. Wang, Aging state prediction for supercapacitors based on heuristic Kalman filter optimization extreme learning machine, *Energy* 250 (2022) 123773.
- [20] Y. Lu, Artificial intelligence: A survey on evolution, models, applications and future trends, *J. Manag. Anal.* 6 (1) (2019) 1–29.
- [21] C. Zhang, Y. Lu, Study on artificial intelligence: The state of the art and future prospects, *J. Ind. Inform. Integr.* 23 (2021) 100224.
- [22] X. Shu, S. Shen, J. Shen, Y. Zhang, G. Li, Z. Chen, Y. Liu, State of health prediction of lithium-ion batteries based on machine learning: Advances and perspectives, *Iscience* 24 (11) (2021) 103265.
- [23] L. Zhang, J. Lin, B. Liu, Z. Zhang, X. Yan, M. Wei, A review on deep learning applications in prognostics and health management, *Ieee Access* 7 (2019) 162415–162438.
- [24] X. Sui, S. He, S.B. Vilsen, J. Meng, R. Teodorescu, D.-I. Stroe, A review of non-probabilistic machine learning-based state of health estimation techniques for lithium-ion battery, *Appl. Energy* 300 (2021) 117346.
- [25] T. Lombardo, M. Duquesnoy, H. El-Bouysidy, F. Áren, A. Gallo-Bueno, P.B. Jørgensen, A. Bhowmik, A. Demortière, E. Ayerbe, F. Alcaide, et al., Artificial intelligence applied to battery research: Hype or reality? *Chem. Rev.* (2021).
- [26] F. Wang, A. Preininger, AI in health: State of the art, challenges, and future directions, *Yearb. Med. Inform.* 28 (01) (2019) 016–026.
- [27] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Židek, A. Potapenko, et al., Highly accurate protein structure prediction with AlphaFold, *Nature* 596 (7873) (2021) 583–589.
- [28] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134.
- [29] L. Ward, S. Babinec, E.J. Dufek, D.A. Howey, V. Viswanathan, M. Aykol, D.A. Beck, B. Blaiszik, B.-R. Chen, G. Crabtree, et al., Principles of the battery data genome, 2021, arXiv preprint arXiv:2109.07278.
- [30] Y. Zhang, R. Xiong, H. He, M.G. Pecht, Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries, *IEEE Trans. Veh. Technol.* 67 (7) (2018) 5695–5705.
- [31] X. Li, L. Zhang, Z. Wang, P. Dong, Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and elman neural networks, *J. Energy Storage* 21 (2019) 510–518.
- [32] W. Zhang, X. Li, X. Li, Deep learning-based prognostic approach for lithium-ion batteries with adaptive time-series prediction and on-line validation, *Measurement* 164 (2020) 108052.
- [33] Y. Fan, F. Xiao, C. Li, G. Yang, X. Tang, A novel deep learning framework for state of health estimation of lithium-ion battery, *J. Energy Storage* 32 (2020) 101741.
- [34] K. Liu, Y. Shang, Q. Ouyang, W.D. Widanage, A data-driven approach with uncertainty quantification for predicting future capacities and remaining useful life of lithium-ion battery, *IEEE Trans. Ind. Electron.* 68 (4) (2020) 3170–3180.
- [35] S. Li, C. Ju, J. Li, R. Fang, Z. Tao, B. Li, T. Zhang, State-of-charge estimation of lithium-ion batteries in the battery degradation process based on recurrent neural network, *Energies* 14 (2) (2021) 306.
- [36] S. Cui, I. Joe, A dynamic spatial-temporal attention-based GRU model with healthy features for state-of-health estimation of lithium-ion batteries, *Ieee Access* 9 (2021) 27374–27388.
- [37] P.M. Attia, A. Bills, F.B. Planella, P. Dechent, G. Dos Reis, M. Dubarry, P. Gasper, R. Gilchrist, S. Greenbank, D. Howey, et al., “Knees” in lithium-ion battery aging trajectories, *J. Electrochem. Soc.* 169 (6) (2022) 060517.
- [38] D. Anseán, M. Dubarry, A. Devie, B. Liaw, V. García, J. Viera, M. González, Operando lithium plating quantification and early detection of a commercial LiFePO₄ cell cycled under dynamic driving schedule, *J. Power Sources* 356 (2017) 36–46.
- [39] G. Baure, M. Dubarry, Synthetic vs. real driving cycles: A comparison of electric vehicle battery degradation, *Batteries* 5 (2) (2019) 42.
- [40] G. Dos Reis, C. Strange, M. Yadav, S. Li, Lithium-ion battery data and where to find it, *Energy AI* (2021) 100081.
- [41] S. Wang, S. Jin, D. Bai, Y. Fan, H. Shi, C. Fernandez, A critical review of improved deep learning methods for the remaining useful life prediction of lithium-ion batteries, *Energy Rep.* 7 (2021) 5562–5574.
- [42] V.D. Angelis, Battery archive, 2022, URL www.batteryarchive.org.
- [43] M. Dubarry, D. Beck, Big data training data for artificial intelligence-based Li-ion diagnosis and prognosis, *J. Power Sources* 479 (2020) 228806.
- [44] M. Dubarry, D. Beck, Analysis of synthetic voltage vs. Capacity datasets for big data Li-ion diagnosis and prognosis, *Energies* 14 (9) (2021) 2371.
- [45] S. Kim, Z. Yi, B.-R. Chen, T.R. Tanim, E.J. Dufek, Rapid failure mode classification and quantification in batteries: A deep learning modeling framework, *Energy Storage Mater.* 45 (2022) 1002–1011.
- [46] M. Dubarry, G. Baure, A. Devie, Durability and reliability of EV batteries under electric utility grid operations: Path dependence of battery degradation, *J. Electrochem. Soc.* 165 (5) (2018) A773.
- [47] M. Dubarry, A. Devie, K. Stein, M. Tun, M. Matsuura, R. Rocheleau, Battery energy storage system battery durability and reliability under electric utility grid operations: Analysis of 3 years of real usage, *J. Power Sources* 338 (2017) 65–73.
- [48] J. Vetter, P. Novák, M.R. Wagner, C. Veit, K.-C. Möller, J. Besenhard, M. Winter, M. Wohlfahrt-Mehrens, C. Vogler, A. Hammouche, Ageing mechanisms in lithium-ion batteries, *J. Power Sources* 147 (1–2) (2005) 269–281.
- [49] C.R. Birkl, M.R. Roberts, E. McTurk, P.G. Bruce, D.A. Howey, Degradation diagnostics for lithium ion cells, *J. Power Sources* 341 (2017) 373–386.
- [50] M. Dubarry, C. Truchot, B.Y. Liaw, Synthesize battery degradation modes via a diagnostic and prognostic model, *J. Power Sources* 219 (2012) 204–216.
- [51] H.M. Dahn, A. Smith, J. Burns, D. Stevens, J. Dahn, User-friendly differential voltage analysis freeware for the analysis of degradation mechanisms in Li-ion batteries, *J. Electrochem. Soc.* 159 (9) (2012) A1405.
- [52] A. Smith, H.M. Dahn, J. Burns, J. Dahn, Long-term low-rate cycling of LiCoO₂/graphite Li-ion cells at 55 C, *J. Electrochem. Soc.* 159 (6) (2012) A705.
- [53] M. Ecker, J.B. Gerschler, J. Vogel, S. Käbitz, F. Hust, P. Dechent, D.U. Sauer, Development of a lifetime prediction model for lithium-ion batteries based on extended accelerated aging test data, *J. Power Sources* 215 (2012) 248–257.
- [54] A. Barai, K. Uddin, W. Widanage, A. McGordon, P. Jennings, The effect of average cycling current on total energy of lithium-ion batteries for electric vehicles, *J. Power Sources* 303 (2016) 81–85.
- [55] A. Smith, J. Burns, D. Xiong, J. Dahn, Interpreting high precision coulometry results on Li-ion cells, *J. Electrochem. Soc.* 158 (10) (2011) A1136.
- [56] F. Yang, D. Wang, Y. Zhao, K.-L. Tsui, S.J. Bae, A study of the relationship between coulombic efficiency and capacity degradation of commercial lithium-ion batteries, *Energy* 145 (2018) 486–495.
- [57] P. Iurilli, C. Brivio, V. Wood, On the use of electrochemical impedance spectroscopy to characterize and model the aging phenomena of lithium-ion batteries: A critical review, *J. Power Sources* 505 (2021) 229860.
- [58] D. Li, L. Wang, C. Duan, Q. Li, K. Wang, Temperature prediction of lithium-ion batteries based on electrochemical impedance spectrum: A review, *Int. J. Energy Res.* (2022).

- [59] G. Piłatowicz, A. Marongiu, J. Drillkens, P. Sinhuber, D.U. Sauer, A critical overview of definitions and determination techniques of the internal resistance using lithium-ion, lead-acid, nickel metal-hydrate batteries and electrochemical double-layer capacitors as examples, *J. Power Sources* 296 (2015) 365–376.
- [60] A. Barai, K. Uddin, W. Widanage, A. McGordon, P. Jennings, A study of the influence of measurement timescale on internal resistance characterisation methodologies for lithium-ion cells, *Sci. Rep.* 8 (1) (2018) 1–13.
- [61] M. Dubarry, M. Bercibar, A. Devie, D. Anseán, N. Omar, I. Villarreal, State of health battery estimator enabling degradation diagnosis: Model and algorithm description, *J. Power Sources* 360 (2017) 59–69.
- [62] M. Dubarry, A. Devie, Battery durability and reliability under electric utility grid operations: Representative usage aging and calendar aging, *J. Energy Storage* 18 (2018) 185–195.
- [63] M. Dubarry, C. Pastor-Fernández, G. Baure, T.F. Yu, W.D. Widanage, J. Marco, Battery energy storage system modeling: Investigation of intrinsic cell-to-cell variations, *J. Energy Storage* 23 (2019) 19–28.
- [64] D. Anseán, G. Baure, M. González, I. Cameán, A. García, M. Dubarry, Mechanistic investigation of silicon-graphite/LiNiO₂.8MnO₂.1CoO₂.1o₂ commercial cells for non-intrusive diagnosis and prognosis, *J. Power Sources* 459 (2020) 227882.
- [65] K.S. Mayilvahanan, K.J. Takeuchi, E.S. Takeuchi, A.C. Marschilok, A.C. West, Supervised learning of synthetic big data for Li-ion battery degradation diagnosis, *Batteries Supercaps* (2021).
- [66] S. Lee, Y. Kim, Li-ion battery electrode health diagnostics using machine learning, in: 2020 American Control Conference, ACC, IEEE, 2020, pp. 1137–1142.
- [67] P. Tormene, T. Giorgino, S. Quaglini, M. Stefanelli, Matching incomplete time series with dynamic time warping: An algorithm and an application to post-stroke rehabilitation, *Artif. Intell. Med.* 45 (1) (2009) 11–34.
- [68] T. Górecki, M. Luczak, Non-isometric transforms in time series classification using DTW, *Knowl.-Based Syst.* 61 (2014) 98–108.
- [69] M. Shah, J. Grabocka, N. Schilling, M. Wistuba, L. Schmidt-Thieme, Learning DTW-shapelets for time-series classification, in: Proceedings of the 3rd IKDD Conference on Data Science, 2016, 2016, pp. 1–8.
- [70] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, E. Keogh, Generalizing DTW to the multi-dimensional case requires an adaptive approach, *Data Min. Knowl. Discov.* 31 (1) (2017) 1–31.
- [71] M. Cuturi, M. Blondel, Soft-DTW: A differentiable loss function for time-series, in: International Conference on Machine Learning, PMLR, 2017, pp. 894–903.
- [72] N. Begum, L. Ulanova, J. Wang, E. Keogh, Accelerating dynamic time warping clustering with a novel admissible pruning strategy, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 49–58.
- [73] L. Tao, C. Lu, A. Noktehdan, Similarity recognition of online data curves based on dynamic spatial time warping for the estimation of lithium-ion battery capacity, *J. Power Sources* 293 (2015) 751–759.
- [74] Y. Liu, C. Zhang, J. Jiang, Y. Jiang, L. Zhang, W. Zhang, Capacity estimation of serial lithium-ion battery pack using dynamic time warping algorithm, *IEEE Access* 7 (2019) 174687–174698.
- [75] P. Hu, G. Ma, Y. Zhang, C. Cheng, B. Zhou, Y. Yuan, State of health estimation for lithium-ion batteries with dynamic time warping and deep kernel learning model, in: 2020 European Control Conference, ECC, IEEE, 2020, pp. 602–607.
- [76] S. Kim, N.H. Kim, J.-H. Choi, Prediction of remaining useful life by data augmentation technique based on dynamic time warping, *Mech. Syst. Signal Process.* 136 (2020) 106486.
- [77] D. Beck, P. Dechent, M. Junker, D.U. Sauer, M. Dubarry, Inhomogeneities and cell-to-cell variations in lithium-ion batteries, a review, *Energies* 14 (11) (2021) 3276.
- [78] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, Í. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental algorithms for scientific computing in python, *Nature Methods* 17 (2020) 261–272, <http://dx.doi.org/10.1038/s41592-019-0686-2>.
- [79] W. Meert, K. Hendrickx, T. Graenendonck, Wannism/dtaidistance v2. 0.0, 2020, Zenodo.
- [80] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, *Adv. Neural Inf. Process. Syst.* 25 (2012) 1097–1105.
- [81] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.
- [82] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929).
- [83] L. Biewald, Experiment tracking with weights and biases, 2020, URL <https://www.wandb.com/>, Software available from wandb.com.
- [84] X. Ying, An overview of overfitting and its solutions, *J. Phys.: Conf. Ser.* 1168 (2019) 022022.
- [85] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, *Proc. IEEE* 109 (1) (2020) 43–76.
- [86] D. Anseán, M. Dubarry, A. Devie, B. Liaw, V. García, J. Viera, M. González, Fast charging technique for high power LiFePO₄ batteries: A mechanistic analysis of aging, *J. Power Sources* 321 (2016) 201–209.
- [87] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, URL <https://www.tensorflow.org/>, Software available from tensorflow.org.