Evaluating time series encoding techniques for Predictive Maintenance

---

Title: Evaluating time series encoding techniques for Predictive Maintenance

Authors: Aniello De Santo[1], Antonino Ferraro[2], Antonio Galli[2], Vincenzo Moscato[2], Giancarlo Sperlì[2]

Affiliation: [1] Department of Linguistics of the University of Utah (Salt Lake City, UT 84112, USA) [2] Department of Electrical Engineering and Information Technology (DIETI), University of Naples "Federico II", Via Claudio 21, Naples, Italy

Email: aniello.desanto@utah.edu,antonino.ferraro@unina.it, antonio.galli@unina.it, vincenzo.moscato@unina.it, giancarlo.sperli@unina.it

Corresponding Author Giancarlo Sperlì, Department of Electrical Engineering and Information Technology (DIETI), University of Naples "Federico II", Via Claudio 21, Naples, Italy (email: giancarlo.sperli@unina.it Phone: (+39) 081-76883606).

Evaluating time series encoding techniques for Predictive MaintenanceEvaluating time series encoding techniques for Predictive Maintenance

Abstract

Predictive Maintenance has become an important component in modern industrial scenarios, as a way to minimize down-times and fault rate for different equipment. In this sense, while machine learning and deep learning approaches are promising due to their accurate predictive abilities, their data-heavy requirements make them significantly limited in real world applications. Since one of the main issues to overcome is lack of consistent training data, recent work has explored the possibility of adapting well-known deep-learning models for image recognition, by exploiting techniques to encode time series as images. In this paper, we propose a framework for evaluating some of the best known time series encoding techniques, together with *Convolutional Neural Network*-based image classifiers applied to predictive maintenance tasks. We conduct an extensive empirical evaluation of these approaches for the failure prediction task on two real-world datasets (*PAKDD2020 Alibaba AI OPS Competition* and *NASA bearings*), also comparing their performances with respect to the state-of-the-art approaches. We further discuss advantages and limitation of the exploited models when coupled with proper data augmentation techniques.

Keywords: Predictive maintenance, Time series Encoding techniques, Failure Prediction, Deep Learning

1. Introduction

Fast changes to the landscape of digital technologies have been significantly transforming industrial processes, due to the deep integration between physical

and digital systems of production environments. Nowadays, it is possible to collect vast amounts of data about the way different equipment operates while, at the same time, allowing for targeted exchanges of information among people, products, and machines.

The increasing transformative speed of technical innovations has led some experts to label this new phase of development the "Fourth Industrial Revolution" (or *Industry 4.0*) — associating it to high-connectivity, the availability of rich data sources, and of technologies able to explore the high dimensionality of such sources due to increase in power and storage capacity (Schwab (2017)). For instance, the variety of events occurring at every moment along an industrial production line can now be analyzed in real-time, correlating real-time data to past events to detect and prevent possible structural failures, thus avoiding long down-times.

More in general, rich, high-dimensional real-time data can bring out valuable insights about the internal dynamics of complex industrial systems. In this sense, the application of data analytic techniques to the industrial pipe-line has shown incredible potential towards a variety of domains: maintenance cost reduction, machine fault reduction, repair stop reduction, spare parts inventory reduction, increased spare part life, increased overall production, improvement in operator safety, repair verification, overall profit, just to name a few (Zhang et al. (2019a); Alizadeh & Ma (2021)). Notably, most of these issues are tied to the timely deployment of efficient and effective maintenance procedures.

A first field of interest is surely represented by the condition monitoring and diagnostics of mechanical components within Avionic or Automotive industries such as gearboxes, ball bearings, and rotating shafts Souza et al. (2021). A sudden break in production line has costs of missing product that abundantly exceed the costs of the component itself. Another interesting domain of application in which maintenance procedures are also crucial is Information technology (IT) Infrastructures management, and in particular for hard disk failures prediction within large-scale data centers. Hard disk disruptions in this kind of data centers directly affect the reliability of the entire infrastructure, thus negatively

impacting the business Service Level Agreement (Su & Huang (2018)).

To address these issues, *Predictive Maintenance* techniques have become essential in guaranteeing strong business improvements, exploiting the technological changes of *Industry 4.0* to minimize down-times and fault rate for different equipment in heterogeneous contexts (Zonta et al. (2020a); Carvalho et al. (2019); Cañas et al. (2021); Geng & Wang (2022); Nakagawa et al. (2021)).

As in many other areas concerned with huge amount of complex data, approaches exploiting machine learning and deep learning tools appear to be most promising among the diverse array of modern predictive maintenance techniques (Carvalho et al. (2019); Ran et al. (2019); Rieger et al. (2019)). Such approaches usually leverage historical datasets, structured as labeled time series about equipment operations, to train a variety of regression/classification models which can then be used to predict possible failures in terms of *Remaining Useful Life* (RUL) estimation. It follows immediately that the performance of such approaches is fundamentally tied to the availability of extensive and reliable training data.

Since this is not always the case in real world scenarios, deep learning models have gained attention as a way to overcome potential imitations due to missing of data (Dalzochio et al. (2020a)). Given that the majority of reliable deep learning architectures were developed with a focus on image analysis, the last few years have seen a rise in studies aiming to encode time series as images, and re-frame RUL as an image classification task (Krishna & Kalluri (2019)).

In this paper, we propose a framework for evaluating the performance on predictive maintenance tasks of some of the most diffused time series encoding techniques (i.e., Recurrence Plot, Gramian Angular Field, Markovian Transition Filed, Wavelet Transform) together with image classifiers based on *Convolutional Neural Network*s (CNNs). The CNN models are then compared with three benchmarking deep learning models on the PAKDD2020 Alibaba AI Ops Competition — which provides data on hard disk status within a data center — and other two state-of-the-art models on the *NASA bearing* datasets — that is composed by vibration signal of bearing. In particular, the experimental eval-

uation underlines that the use of CNN, whose input is generetad by enconding techniques, achieves high effectiveness performances with respect to the majority of the state-of-the-art models, also being the best model in terms of Memory Occupation parameter. We discuss the advantage of proper data augmentation processes, also based on Generative Adversarial Network (GAN), and show results highlighting the advantages and disadvantages of different modelling and training choices. In particular, we show that while using a GAN helps in the training process by providing a slight increase in performance, this small advantage has to be balanced with heavier demands on training time and memory resources. To the best of our knowledge, the present work is one of the first studies reporting a complete, systematic benchmark of a variety of vastly used time series encoding techniques as valid models for predictive maintenance tasks.

Summarizing, the main novelties of the proposed approach concern:

- the design of a general framework for evaluating the performance for predictive maintenance tasks of some of the most diffused time series encoding techniques;
- the use of two different CNN-based models for equipment failure prediction, whose input is generated by different encoding techniques;
- the comparison of the encoding-based techniques with respect to different state-of-the-art approaches on two real-world dataset (*PAKDD2020 Alibaba AI Ops Competition* and *NASA bearing*);
- a performance analysis adopting GANs as data augmentation strategy.

The paper is organized as follows. Section 2 surveys related work on predictive maintenance techniques, focusing on recent machine approaches. Section 3 provides a theoretical background on time series encoding techniques. Sections 4 and 5 describe in detail the definition of a predictive maintenance task, and of the evaluation framework adopted in this paper. Section 6 reports on the

experiments we conducted to evaluate the different encoding techniques introduced throughout the paper. Finally, Section 8 concludes with a summary of the results and discussion of potential future work.

2. Related Work

Scheduling maintenance decisions is a critical task to avoid unexpected shutdowns of mechanical equipment with the aim to increase their reliability Ran et al. (2019). For this reason, predictive maintenance has increasingly played a key role in Industry to jointly improve equipment's efficiency and reduce operating costs by using machine learning models, analyzing large amount of their operational data (see Ran et al. (2019); Zhang et al. (2019b); Zonta et al. (2020a); Dalzochio et al. (2020b) for more details).

Clearly, while predictive maintenance remains a challenging task in the domain of machine health status in general (Dalzochio et al. (2020b); Zonta et al. (2020b); Fink et al. (2020)), advances in "Artificial Intelligence" models have been crucial in improving the reliability of predictive approaches to failure detection (Solomon et al. (2022); Serradilla et al. (2022); Giordano et al. (2022)). Notoriously, deep learning techniques rely heavily on labelled datasets, incorporating information about equipment operation trajectories (usually encoded as time series), to train models then used to estimate RUL (we refer the reader to (Carvalho et al. (2019); Ran et al. (2019); Rieger et al. (2019); Schwendemann et al. (2021)) for recent surveys of machine learning approaches to predictive maintenance). Just to mention a few recent results in this sense, Zhang et al. (2018a) successfully exploit an autoregressive model (together with a regularized particle-filter algorithm, AR-RPF) to predict the RUL of lithium-ion batteries. Similarly, a least squares support vector machine (SVM) model has been used for fault detection and diagnosis of chillers (Han et al. (2019)). Other approaches have been further proposed to predict RUL of bearings, the most common mechanical components used in different equipment that deteriorates over time due to the harsh working conditions, with the aim to reduce costly unplanned main-

tenance and increasing machine reliability, availability, and safety (Wang et al. (2018); Liu & Zhang (2020)). First approaches (Siegel et al. (2012); Loutas et al. (2013)) mainly relied on regression strategies for predicting Remaining Useful Life (RUL) of bearings. In particular, the former perform a classical machine learning pipeline, composed by feature extraction, selection and regression-based prediction method, while the latter uses a wavelet transform to extract statistical features from time, frequency and time-scale domain, that are fed as input to a *Support Vector Regression* (SVR). Other approaches (Liu et al. (2021a); Qin et al. (2021)) are mainly focused on deep learning models for RUL estimation on the basis on vibration signals of bearings. Other approaches (Anantharaman et al. (2018); Basak et al. (2019); Lima et al. (2018); De Santo et al. (2020)) aimed to exploit the fine-granularity of information offered by S.M.A.R.T. attributes for predicting Hard Disk Drive (HDD) RUL. Anantharaman et al. (2018) evaluate two different models — Random Forest and Long-Short Term Memory (LSTM) networks — on the task of predicting predict RUL, crucially characterized as a multi-classification task. A similar comparative approach is taken by Lima et al. (2018), who contrast LSTM and CNN models on a failure prediction task. In (Basak et al. (2019)), the authors propose a feature selection strategy based on correlation values, that are successively used to train an LSTM network to predict disk failure within a ten day window. Somewhat relatedly, De Santo et al. (2020) experiment with a LSTM-based model that combines S.M.A.R.T. attributes and temporal analysis for predicting HDD health status 45 days before failure.

Chen et al. (2021b) developed a LSTM model based on attention mechanism to predict machine's RUL prediction learning sequential features from raw sensory data. A further deep learning-based method using attention mechanism has been proposed by Song et al. (2021) for RUL prediction analyzing data from different industrial sensors. Ragab et al. (2021) proposed another attention-based mechanism, called *ATS2S*, whose aim is to jointly optimize reconstruction and RUL prediction loss to minimize estimation error.

The performance of such methodologies is obviously strongly correlated with

the availability of extensive, consistent, and reliable training data. As shown in Zhao et al. (2021), estimation of equipment's RUL strongly depends on the extraction method of performance degradation features. Because data might often been missing in real world scenarios, some past work has investigated the possibility of adopting deep learning models (Dalzochio et al. (2020a)). Given that extensive work on deep learning models has taken place in the domain of image analysis, recent work has experimented with encoding time series in the form of images. Deep networks can then be applied to remaining life estimation, now recast as a special instance of a more general image classification task (Guillaume et al. (2020)).

In this sense, a variety of techniques have been developed in order to encode the temporal correlation of different features as images, that can be used for training well-known deep learning architectures in support of a diverse array of applications. Four of these encoding methods seem to be particularly prominent in the literature.

First of all, some approaches have exploited the *Gramian Angular Field* encoding method, to transform one-dimensional time series data into images, then fed as training to a Convolutional LSTM applied to Solar Irradiation Forecasting (Hong et al. (2020)), visual deep learning models for knowledge distillation (Liu et al. (2021b)) or activity recognition (Qin et al. (2020)), or even financial forecasting (Barra et al. (2020)). In (Kiangala & Wang (2020), the authors designed a further CNN-based model, whose input is generated by GAF method, for predictive maintenance about Conveyor Motors in an Industry 4.0 environment.

Among the few attempts in this direction, (Ferraro et al. (2020)) proposed an approach to predictive maintenance by encoding sequences of S.M.A.R.T. attributes over time through a Gramian Angular Field (GAF) to generate images for training a CNN. Furthermore, GAF-based methodologies has been applied to generate input to be fed as input to a CNN-based model for predicting electricity consumption (Chan et al. (2019)) and stock market prediction (Chen et al. (2021a)) based on multivariate time series data.

Other approaches have used a Discrete Wavelet Transform (DWT). For instance, DWT has been used over time series in order to generate the input of a CNN and identify fault conditions of gearboxes (Chen et al. (2019); Liang et al. (2019)). It has also been combined with a 1-dimensional hexadecimal local pattern technique for arrhythmia detection based on ECG signals (Tuncer et al. (2019)). Furthermore, a two-stage predictive algorithm based on DWT and Echo State Network (ESN) has been implemented by (Gao et al. (2021)) for time series forecasting.

Another popular method found in the literature is the Markov Transition Field (MTF). For instance, (Bugueño et al. (2021)) leverage it to encode raw signals in 2-D images. In that paper, light curves are encoded as 2-D images via the MFT, and used as input to a CNN in order to classify candidate transients. Similarly, (Vandith Sreenivas et al. (2021)) use this method to generate 2-D images from ECG signals for Arrhythmia classification. Other approaches based on MTF encoding methods and CNN has been designed by (Fahim et al. (2020)) and (Zhang et al. (2018b)) aiming to identify anomalous energy consumption and online fraud, respectively.

Instead, the work of Yang et al. (2020), proposed a framework to perform sensor classification using multivariate sensor time series data as input, encoded through two types of encoding, GAF and MTF, to perform classification with a CNN.

Finally, the *Recurrent Plot* method has been used to encode data from 3-axis accelerometer as images and train a CNN model for human activity recognition (Lu & Tong (2019a)). A similar approach has been used to analyze handwriting dynamics signals for the diagnosis of Parkinson's disease (Afonso et al. (2019)). (Zhang et al. (2021)) designed an Inception Architectural Networks whose input are encoded multi-scale time series through RP method to deal with classification tasks.

Table 1 provides summary of recent state-of-the-art contributions, using the different encoding methods discussed so far. It is important to observe that the majority of these techniques rely on a one-dimensional time series analysis and

CNN-based models.

While past work seems to show that these encoding approaches have been successful applied across to time series a variety of application domains (see Table 1) — from activity recognition to disease identification — there seems to be a gap in the literature in terms of applying them to the task of equipment failure prediction, and in particular to HDD and bearing health status prediction, which constitute the mos diffused and studied case studies for predictive maintenance. In particular, our aim is to investigate the use of encoding strategies to deal with predictive maintenance task in order to decrease the required resource in terms of memory and training time while achieving effectiveness performance at least similar to the state-of-the-art approaches. Therefore, the basic idea is to encode time-series to leverage the last advances in supervised learningSuaboot et al. (2020); Sundararajan & Woodard (2018) for unveiling local patterns that would otherwise be spread over time. As it is easy to note in Table 1, the proposed methodology aims to investigate multidimensional time-series, which are typically generated in industrial settings, by different encoding techniques while the majority of the proposed approaches are based on the analysis of one-dimensional signals, mainly using only one type of encoding strategies.

In the rest of the paper, we outline a framework for evaluating the four time series encoding techniques discussed here over predictive maintenance tasks, coupling them with CNN-based image classifiers. We focus our evaluation on the *Alibaba* and *NASA* bearing datasets, and compare the performance of the CNN models to a few alternative machine learning architectures.

3. Background

This section provides some mathematical background on the four main encoding techniques evaluated in this paper — namely, *Recurrence Plot*, *Gramian Angular Field*, *Markovian Transition Field*, and *Wawelet Transform* — in order to highlight the fundamental formal differences at the core of these approaches.

Paper	Model	Time Series	Encoding	Application
Hong et al. (2020)	Convolutional LSTM	One-dimensional	GAF	Solar Irradiation Forecasting
Liu et al. (2021b)	CNN	One-dimensional	GAF	Knowledge distillation
Qin et al. (2020)	CNN	One-dimensional	GAF	Activity Recognition
Barra et al. (2020)	CNN	One-dimensional	GAF	Financial forecasting
Kiangala & Wang (2020)	CNN	One-dimensional	GAF	Conveyor Motor maintenance
Ferraro et al. (2020)	CNN	Multi-dimensional	GAF	Hard drives health status
Yang et al. (2020)	CNN	Multi-dimensional	GAF, MTF	Sensor classification
Chan et al. (2019)	CNN+SVM	Multi-dimensional	GAF	Electricity consumption forecasting
Chen et al. (2021a)	CNN	Multi-dimensional	GAF	Stock Market Forecasting
Chen et al. (2019)	CNN	One-dimensional	DWT	Gearboxes fault diagnosis
Liang et al. (2019)	CNN	One-dimensional	DWT	Gearboxes fault diagnosis
Gao et al. (2021)	ESN	One-dimensional	DWT	Time-series forecasting
Tuncer et al. (2019)	NCA+1NN classifier	One-dimensional	DWT	Arrhythmia detection
Bugueño et al. (2021)	CNN	One-dimensional	MTF	Light curves classification
Vandith Sreenivas et al. (2021)	CNN	One-dimensional	MTF	Arrhythmia classification
Fahim et al. (2020)	CNN	One-dimensional	MTF	Anomalous energy consumption
Zhang et al. (2018b)	CNN	One-dimensional	MTF	Online Fraud Detection
Lu & Tong (2019a)	CNN	Multi-dimensional	RP	Activity Recognition
Afonso et al. (2019)	CNN	Multi-dimensional	RP	Disease identification
Zhang et al. (2021)	CNN	One-dimensional	RP	Classification task

Table 1: State-of-the art approaches classified on the basis of the neural network model adopted, the encoding method, and their application domain. Encoding methods are: Gramian Angular Field (GAF), Recurrence Plot (RP), Markovian Transition Field (MTF), and Wavelet Transform (WT). NCA and 1NN stand respectively for Neighborhood Component Analysis and 1-Nearest Neighborhood.

3.1. Recurrence Plot

A Recurrence Plot ((RP; Eckmann et al., 1995; Marwan et al., 2007)) is a visualization tool to explore an m-dimensional phase space trajectory through a 2-dimensional representation of its recurrences. The core idea is to reveal in which points some trajectories return to a previous state. Mathematically, this concept can be formulated as:

$$R_{i,j} = \theta(\epsilon - ||\vec{s_i} - \vec{s_j}||), \quad \vec{s}(.) \in R^m, \quad i, j = 1, ..., K \tag{1}$$

where K is the number of states $\vec{s}$, ϵ is a threshold distance, $||.||$ is the norm and θ is the Heaviside function. As a result R is a matrix. Fading to the upper left and lower right corners represents a trend, while vertical and horizontal lines indicate that some states do not change or change slowly.

3.2. Gramian Angular Field

A Gramian Angular Field ((GAF; Wang & Oates, 2015)) encoding produces an image representing a time series in a polar coordinate system rather than the typical Cartesian coordinates. Let $Y = \{y_1, y_2, \ldots, y_n\}$ be a time series having observation values scaled within the $[-1, 1]$ interval. Then, the scaled time series is represented in polar coordinates by encoding each value as the angular cosine, and the time stamp as the radius using the equation below:

$$\begin{cases} \phi = arccos(\tilde{x_i}) & -1 \leq \tilde{x_i} \leq 1, \quad \tilde{x_i} \in \tilde{X} \\ r = \frac{t_i}{N} & t_i \in N \end{cases} \tag{2}$$

where t_i is the time stamp and N is a constant factor to regularize the span of the polar coordinate system. This transformation has three core properties: i) it is bijective with rescaled $[0, 1]$ time series data and it produces one and only map; ii) it is surjective with rescaled $[-1, 1]$ data and it produces one map, as the inverse image is not unique because of the ambiguity of $cos(\phi)$ when ϕ is in $[0, 2\pi]$; iii) unlike Cartesian coordinates, polar coordinates preserve absolute temporal relations. A GAF provides a different information granularity for

classification tasks. After transforming the time series into polar coordinates, a GAF constructs a map by calculating the trigonometric sum (GASF), or the difference (GADF), between each point. GASF and GADF are calculated as follows:

$$\begin{cases} GASF = cos(\phi_i + \phi_j) \\ GADF = sin(\phi_i - \phi_j) \end{cases} \tag{3}$$

The GAF is defined as follow:

$$G = \begin{bmatrix} cos(\phi_1 + \phi_1) & \cdots & cos(\phi_1 + \phi_n) \\ cos(\phi_2 + \phi_1) & \cdots & cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ cos(\phi_n + \phi_1) & \cdots & cos(\phi_n + \phi_n) \end{bmatrix} \tag{4}$$

As mentioned, the use of this encoding preserves temporal dependence and temporal correlations.

3.3. Markovian Transition Field

In a Markovian Transition Field ((MTF; Wang & Oates, 2015)), the encoding starts with a time series X and identifies its Q quartile bins by assigning to each x_i its corresponding bin q_j ($j \in [1, Q]$). After that, the adjacency matrix $W = Q \times Q$ is constructed, where each element $w_{i,j}$ represents the frequency with which a point in q_j is followed by a point in q_i. W is called the Markov transition matrix. Importantly, this step potentially leads to a loss of temporal information. In order to overcome this issue, the matrix is distributed. An MTF is thus defined as:

$$\begin{bmatrix} w_{ij|x_1 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_1 \in q_i, x_n \in q_j} \\ w_{ij|x_2 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_2 \in q_i, x_n \in q_j} \\ \cdots & \cdots & \cdots \\ w_{ij|x_n \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_n \in q_i, x_n \in q_j} \end{bmatrix}$$

where each element $w_{i,j}$ represents the transition probability from quantile q_i to quantile q_j and the main diagonal is the special case of the self-transition probability from each quantile to itself. The sum of the elements of a row must have a value equal to 1. Like the GAF, the MTF is surjective and, starting from a time series X and fixing quantile bins Q, produces a single map. Note that the inverse image of the MTF is not unique.

3.4. Wavelet Transform

The Wavelet Transform ((WT; Akansu et al., 2001; Addison, 2005, a.o.)) is an alternative to the more classic Fourier transform, decomposing a function into a set of wavelets. It provides high resolution in both the time and frequency domains, and it is thus suitable for analysing dynamic signals. An important property is that a wavelet exists for a finite duration.

There are two types of WT:

- **Discrete Wavelet Transform (DWT)**: The frequencies of the original signals are decomposed into *approximate coefficients* and *detail coefficients* (also called *wavelet coefficients*). Detail coefficients with larger amplitudes are considered significant, while those with smaller amplitudes are noise. A DWT used in combination with threshold denoising is a low-pass filter: it removes high-frequency noise and it is suitable for removing transient signals.

- **Continuous Wavelet Transform (CWT)**: It is based on the *mother wavelet*. One type of application requires a different mother, because each of them has a characteristic frequency band. The equivalent frequency is defined as:

$$F_{eq} = \frac{C_f}{s\delta t} \tag{5}$$

  where C_f represents the center frequency, s is the wavelet scale and δt is the sampling interval. The output of a CWT are coefficients that are function of scale, frequency and time: the higher the number of scales considered, the finer is the scale discretization.

The DWT is often used for denoising and compression of signals because it can represent them with few coefficients. The CWT is instead often used in time-frequency analysis and filtering of time-localised frequency components.

4. Task definition

Having put some formal preliminaries in place, we can now turn to the definition of the most essential component of this paper: the evaluation task. This task concerns failure detection for an equipment, in order to jointly increase their *RUL* and optimize maintenance operations.

Definition 1. *Let* $(\mathcal{TS}_i)_{i\in\mathbb{Z}_N}$ *be a set of* N *time series concerning different physical attributes of the considered equipment* EQ*, the predictive task can be seen as a function aiming to assess the health status* $y = f((\mathcal{TS}_i)_{i\in\mathbb{Z}_N}) \in [0,1]$ *of* EQ *within a time interval* $[t_s, t_f]$*, where* t_s *and* t_f *are, respectively, the initial and final instants of analysis.*

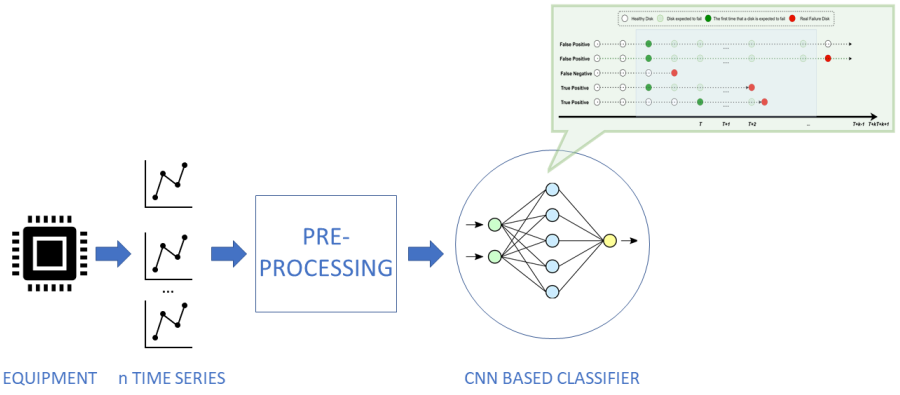

Figure 1: Task definition - This task concerns the analysis of n time series, representing equipment behavior over time. These time series are investigated by the pre-processing stage in order to extract the main features to fed as input to the classification module for predicting the health state of an equipment.

Figure 1 illustrates the behaviour of the prediction system in regard to the equipment health status over a set observation window. Once an equipment status and an observation window are defined, several traditional classification scenarios can occur: if an equipment does not fail (or does fail, but outside of the prediction window) even though it was predicted to do so, it is a False Positive; if a failure occurs without being predicted, it is a False Negative; if an equipment fails within the observation window predicted by the model, it is True Positive.

Remember, then, that we are interested in analyzing equipment attributes over time, since taking temporal trends into account should improve overall performance in the prediction phase. Thus, as mentioned multiple times before, we compare CNN models trained over multivariate time series $TS = \{ts_1, \cdots, ts_n\}$ encoded a set of feature maps $F = \{f_1, \cdots, f_m\}$, computed via four different image encoding methods — *Recurrence Plots*, *Gramian Angular Field*, *Markovian Transition Field* and *Wavelet Transform*. The goal of the trained models is to predict equipment failures based on the extracted features.

5. Framework

An overview of our training and evaluation framework is shown in Figure 2. The aim is to provide a benchmark of different techniques for encoding time-series into in images, focusing on performance over an equipment failure prediction task.

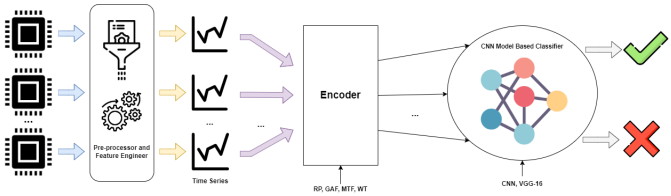

Figure 2: Architectural overview of the proposed framework , that is composed by different phases. The initial step is devoted to pre-processing and feature engineering, the second step is to create the time series sequences and convert them into images, choosing the technique to be used in the encoding module.

First, we conduct a series of pre-processing and feature engineering operations on the dataset. These consist of eliminating attributes with missing values, or columns that do not affect equipment failure prediction (e.g. its capacity or manufacture), causing a reduction of the overall number of features. We also apply a series of rebalancing and feature transformation techniques, through a variety of different methodologies (see Section 6.1.2). In a second phase, we build time-series sequences and feed them to one of the image encoding techniques described in Section 3, thus producing the input to the subsequent CNN for the time-window under consideration. We then train two types of CNN models to deal with the predictive maintenance task. Additionally, we specifically investigate whether using a Generative Adversarial Network (GAN) affects overall performance.

5.1. CNN-based Classifiers

The time series naturally extracted from the dataset can be of different length, so it is first necessary to generate time series sequences based on fixed time windows (40 steps). These can then be fed to one of the encoding techniques discussed in Section 6.1.2, in order to generate image encodings that can be used as input to a CNN model. We consider two alternative CNNs.

A first model is composed of three convolutional layers with *leaky ReLU* as the activation function, each followed by a max pooling layer with filter size and, on top, a fully connected layer and softmax activation (Model 1; Figure 3).

Secondly, we consider a CNN based on the VGG-16 architecture (Simonyan & Zisserman (2014)), a pre-trained model that takes in input $(224, 224)$ RGB images. It consists of 2 convolutional layers, with 64 and 128 filters respectively, followed by the max pooling layer. A third block has 3 convolutional layers, with 256 filters and a max pooling layer. At the top of this architecture, there are two fully connected layers, each with 2048 neurons, and softmax activation (VGG-like; Figure 4). The choice of a shallower CNN allows us to better deal with smaller inputs (40×40).

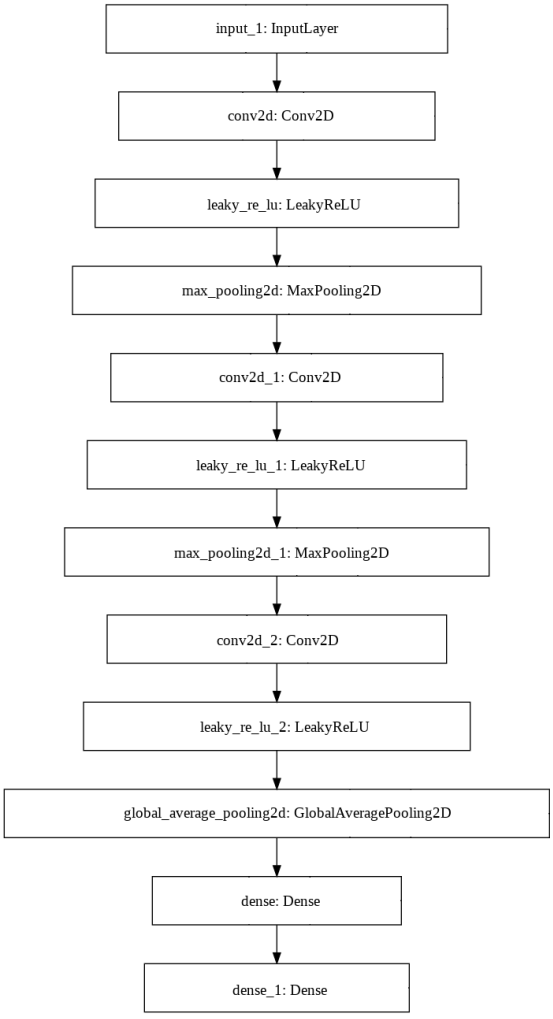

Figure 3: First CNN architecture

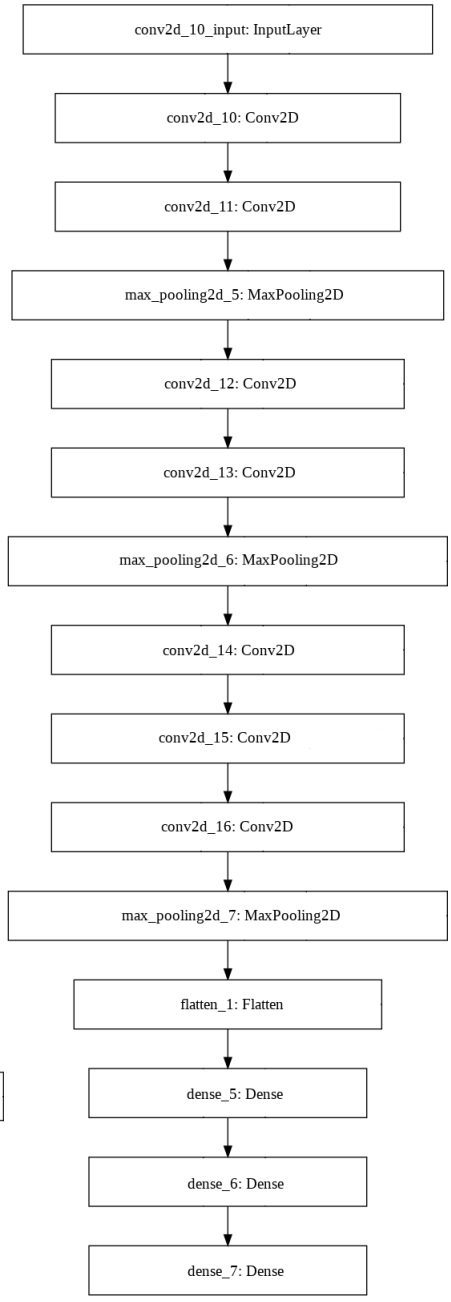

Figure 4: VGG-like architecture

Finally, we choose the **log-loss** (also called **cross-entropy loss**) loss function, which returns a probability between 0 and 1 for the two classes of the task according to equation 6.

$$L_i = -(y \cdot log(p) + (1 - y) \cdot log(1 - p)) \tag{6}$$

where y is the correct label, p is the probability for the correct label and L_i represents the loss for the i-th element that classifier is predicting. Furthermore, we adopt the *Adam* optimizer (Kingma & Ba (2015)), since it guarantees smoother gradient descent, avoiding local optima. The Adam optimizer introduces two additional parameters, called the first and second moment: the former is a velocity term, representing a combination of its history and the current value; while the latter is an energy term of recent movements.

6. Experiments

With all the technical infrastructure in place, we can finally focus on the core of the evaluation task. We evaluate performance of different methodologies with the goal of maximizing effectiveness, thus reducing false positives and false negatives as much as possible, and efficiency. Therefore, we evaluate different combinations of encoding methods and architectures on *efficacy*, as characterized by the metrics described in Section 6.2, and *efficiency* measured in terms of each model's memory usage and training time.

6.1. Experimental protocol

As a base for evaluation, we chose two different datasets for investigating the use of encoding methods in different industrial applications. Since both datasets are related to a challenge, we selected the approaches that yielded the highest effectiveness values.

The HDD dataset released by Alibaba in the PAKDD 2020 AIOps Competition[1]. After a pre-processing and feature engineering phase (see Section 6.1.2), the dataset size is reduced to about 420 MB, corresponding to more than $150,000$

Parameters	Values
Windows size	(1,3,5,7,10,15,20,30)
Epoch	[20 - 300]
Learning Rate	0.1,0.01,0.001
% Fake	25%,30%
% GAN module	Yes,No

Table 2: Hyper-parameters optimization phase.

tensors of size 40×76, where 40 and 76 are, respectively, the window's size and the number of features. In particular, we increased the number of features by adding to raw S.M.A.R.T. attributes some of the generated ones computed on the basis of five methods (Raw, Normalized, Shift, Absolute and Relative), whose description has been provided in Section 6.1.2. The dataset was divided so that 60% of it was used as the training set, 20% as the validation set, and 20% as the testing set. The number of healthy and failed disks was balanced as described in Section 6.1.2.

In turn, the *NASA Bearing* dataset[2] is made up of $19,680,000$, divided into 984 files containing 20K samples After the pre-processing phase (see Section 6.1.2), we applied a window, having size equal to five, to each file for generating encoded images. Therefore, downstream of this elaboration process we will have a total of 984 images that portray the state of health of the bearing at a precise moment during its operation. The dataset has been divided into 3 parts by using *Stratified* approach: 60% and 30% for training and test set and 10% for evaluating the generalization error.

6.1.1. Dataset

As briefly mentioned before, predictive maintenance of equipment's health status is crucial in large-scale industrial infrastructures, since equipment failure can significantly affect the reliability of infrastructure as a whole. Thus, we focus our evaluation of the different methodologies described so far, on the task

Bearing	**No.of Samples**	**No. of raw features**	**Conditions**
Bearing 1	984	20480	Outer race failure
Bearing 2	984	20480	No defect
Bearing 3	984	20480	No defect
Bearing 4	984	20480	No defect

Table 3: NASA Bearing dataset

of predicting potential failures of equipment within a pre-set time window (e.g. 30 days), based on the analysis of time series analysis.

To this aim, we chosen two different datasets for investigating predictive maintenance task in different industrial environments, mainly concerning bearings, being typically undergone to a high speed and very high pressure, and HDD, one of the main crucial point affecting the reliability of infrastructure in large-scale data centers. In particular, we leveraged a dataset from the *PAKDD2020 Alibaba AI OPS*[1] competition, with approximately 40 GB of samples collected from *07/2017* until to *07/2018*. Features in the dataset correspond to S.M.A.R.T. attributes, providing both a raw and a normalised value for each disk per day, as well as a label and the time of failure.

Furthermore, we used the *NASA Bearing* dataset[2], whose details are shown in Table 3. In particular, we are focused on the analysis of the vibration signals only using the accelerometers along the X-axis. These vibration signals were recorded using a time window with a duration of 1 second, at an interval of 10 minutes. The sampling rate for data collection is 20KHz. Then, we perform a noise reduction and data normalization by applying a moving average technique and *Min-max Normalization*, respectively.

[1] `https://tianchi.aliyun.com/competition/entrance/231775/introduction`

[2] `https://ti.arc.nasa.gov/c/3/`

6.1.2. Pre-processing and Feature engineering

Pre-processing of the Alibaba dataset happened in two consecutive phases. First, we carved down a set of relevant features, pruning from over 500 down to 32. Specifically, we started by removing attributes with a percentage of missing values greater than 10% and standard deviation equal to 0, also fully dropping columns that are not critical in failure prediction. The remaining missing values were assigned by a moving average with a window of 5 steps back and 5 steps forward. Furthermore, the dataset overall ratio of healthy :: unhealthy disks was rebalanced — from 1% to 50% — by reducing the total number of healthy disks. In turn, NASA bearing dataset has been sampled from 20KHz to 4KHz by using a windows size equals to five. Since the dataset is not labelled we proceed to perform this operation through visual analysis to perceive the moment in which the failure occurs, following the experimental procedure shown in Markiewicz et al. (2019). In particular, we classified vibration data into three classes according to Markiewicz et al. (2019): a *HIGH-RISK* label for samples near the breakdown, *LOW-RISK* label for the samples representing the normal behaviour and *MEDIUM RISK* that it represents a state of hypothetical medium risk of failure. Summarizing, we classify samples in the *NASA Bearing dataset* as follows:

- Samples from 2004-02-12 10:32:39 to 2004-02-17 10:52:39 are labeled as LOW-RISK
- Samples from 2004-02-17 10:52:39 to 2004-02-18 13:52:39 are labeled as MEDIUM-RISK
- Samples from 2004-02-18 13:52:39 to 2004-02-19 06:22:39 are labeled as HIGH-RISK

We, further, designed the prediction task as a binary problem, in which we consider the union of the classes *HIGH-RISK* and *MEDIUM-RISK* in a single one.

Then, raw features were transformed into normalized ones, using the following methods:

- **Shift features** (Shift): we shifted the original raw features ($V(n)$) by N days ($V(n-N)$), considering different values for $N = \{1, 3, 5, 7, 10, 15, 20, 30\}$.

- **Relative comparison features** (Diff): we computed the difference between a raw feature ($V(n)$) and its corresponding shifted feature ($V(n-N)$):

$$Relative(n)[N] = V(n) - V(n-N) \tag{7}$$

- **Absolute comparison features** (Sum): we calculated the sum between a raw feature ($V(n)$) and its corresponding shifted feature ($V(n-N)$):

$$Absolute(n)[N] = V(n) + V(n-N) \tag{8}$$

- **Exponential moving average features** (Exp): we applied this transformation on *S.M.A.R.T.* raw features for each disk according to equation 9:

$$\begin{aligned} history(n) &= 0.9 \times history(n-1) + 0.1 \times raw(n) \\ history(-1) &= 0 \end{aligned} \tag{9}$$

  where $raw(n)$ is the current value of $S.M.A.R.T._{raw}$ at the $n-th$ step and *history* is the cumulative weighted sum of historic data.

- **Division features** (Div): they represent the ratio between raw and normalized features, as shown in equation 10:

$$Division(n) = \frac{S.M.A.R.T._{raw(n)}}{S.M.A.R.T._{Normalized(n)} + \epsilon} \tag{10}$$

  where ϵ is a constant used to avoid division by zero.

It is important to note that the number inside the parenthesis in the next tables corresponds to the shifted feature in terms of number of days.

A grid search has been performed on the basis of hyper-parameters shown in Table 2, for identifying the optimal ones used for training our models. For

	Model 1			VGG-like		
P	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**
15	39.64±0.31	33.04±1.78	31.07±2.34	28.44±0.79	32.23±0.19	27.67±2.41
30	**59.24± 0.39**	**61.15±3.18**	**57.62±2.61**	**31.59±1.25**	**29.58±0.22**	**34.03±3.13**
45	47.94±1.41	42.77±3.47	48.08±2.89	46.67±2.09	43.01±1.74	48.13±3.85

Table 4: Evaluation of the both networks varying the P parameter.

statistical validation of our results, we ran a 10-cross validation Duin (1996); Benavoli et al. (2017) reporting the mean and standard deviation of each experiment outcome, also performing a stratified sampling strategy for splitting the dataset in training and test set.

Our evaluation framework was deployed on Google Colaboratory[3] using TensorFlow V2[4] and Keras[5] to build deep learning models and using pyts[6] to perform the pre-processing operations and run the time series classification algorithms.

6.2. Evaluation metrics

In this section we describe several metrics used to evaluate the efficiency of the proposed framework, that is defined as the ability to assess the equipment health status within a 30 day interval. Specifically, we define a P-window (setting to 30 days - further details in table 4) as a fixed-size sliding window starting from the first moment in which a disk is predicted to fail.

Precision for P-window: the fraction of records that actually failed (TP) and the fraction expected overall ($TP + FP$):

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

[3] `https://colab.research.google.com/`
[4] `https://www.tensorflow.org/`
[5] `https://keras.io/`
[6] `https://pyts.readthedocs.io/`

where TP and FP are respectively true and false positives.

Recall for R-window: the fraction of predicted failed disks that actually failed (TP) over the overall number of failed disks ($TP + FN$):

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

where TP and FN are respectively true and false negatives.

F1-score is defined according to equation 13:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{13}$$

7. Results

In this section, we discuss the obtained results on the *Alibaba HDD* (Section 7.1) and *NASA bearing* (Section 7.2) datasets. Finally, we investigated in Section 7.3 how the performance of the discussed methodology varies adopting a Generative adversarial Networks (GAN).

7.1. Results on Alibaba HDD

We, firstly, compare the two CNN models described in Section 5.1. Our aim is to understand how each model performs, based on the best fit between different encoding techniques — (RP, GAF, MTF, WT) — and pre-processing approaches (see Section 6.1.2). Summarizing, we compare two different CNN networks (one custom and another one pretrained) in order to effectively exploit the images generated by using the encoding strategies. In this way, our goal is to compare two different strategies (pretrained vs custom) varying the different encoding methods. Table 5 shows the performance of both models in terms of memory usage and overall training time, where Model 1 achieves best results independently of encoding method. This result is due to the larger number of parameters to be optimize within the VGG-like model, resulting in a larger increase in network training time.

Table 6 shows how different combination of encoding techniques and feature engineering approaches affect the overall performance of the CNN we dubbed

Model	Memory(kB)	Training time(secs/epoch)
Model 1	**470**	**13.7±0.1**
VGG-like	73.000	37.4±0.2

Table 5: Memory usage and training time for both models (independently of encoding method)

Technique	**F1-score**	**Precision**	**Recall**
RP + Sum(1)	22.96±1.34	16.52±0.56	37.61±4.34
MTF + Diff(1)	21.64±0.55	15.74±0.22	34.82±2.22
GADF + Diff(7)	**32.50±1.44**	24.98±0.64	**46.41±3.76**
GASF + Exp(15)	31.04±2.09	**28.03±0.46**	35.01±4.84
WV + Exp(30)	26.67±1.46	21.73±0.31	35.01±5.11

Table 6: Performance of Model 1, based on the different image encoding techniques and preprocessing approaches used to generate its input. It is important to note that the number inside the parenthesis in the next tables corresponds to the shifted feature in terms of number of days.

as Model 1. Evidently, *GASF+Exp(15)* achieves the best results in terms of precision, but results in large number of True Negatives (see Table 7). In turn, *GADF+Diff(7)* shows the best results in terms of F1-Score, but with a large number of False Positives (see Table 7).

Moving on to our second CNN model, Table 8 shows the results achieved by the VGG-16-like architecture for different combinations of feature engineering methods and encoding techniques. It should be easy to note that the best results in terms of F1-score and Precision have been achieved by *GASF+Exp(7)* — although it identifies a large number of True Positives — while *RP+Sum(1)* reaches best Recall score, but returns a large number of False Positives (Table 9).

Finally, we investigated the performances of both models on six different types of faults according to the tag field into the *PAKDD2020 Alibaba AI Ops Competition*[1], whose results have been shown in Table 10 and 11. In particular, we can see that *GAF*, using difference and exponential features over 7 and 15

Technique	TP	FP	FN	TN
RP + Sum(1)	30	155	49	127
MTF + Diff(1)	28	148	52	133
GADF + Diff(7)	50	150	60	101
GASF + Exp(15)	30	75	57	199
WV + Exp(30)	27	100	54	180

Table 7: Model 1: Confusion matrices (median values over repeated tests)

Technique	F1-score	Precision	Recall
RP + Sum(1)	22.17±1.52	15.15±0.70	**41.68±5.46**
MTF + Diff(1)	20.87±0.82	14.04±0.28	41.29±4.40
GADF + Diff(7)	29.63±1.17	23.53±0.23	40.23±3.62
GASF + Exp(15)	**31.59±1.25**	**29.58±0.22**	34.03±3.13
WV + Exp(30)	26.29±1.85	22.32±0.27	32.38±5.04

Table 8: Performance of the VGG-like model, based on the different image encoding techniques and pre-processing approaches used to generate its input.

Technique	TP	FP	FN	TN
RP + Sum(1)	34	182	42	103
MTF + Diff(1)	29	183	42	107
GADF + Diff(7)	45	146	66	104
GASF + Exp(15)	31	74	61	195
WV + Exp(30)	27	95	58	181

Table 9: VGG-like architecture: Confusion matrices (median values over repeated tests)

days respectively, achieves highest results; this is related to how the encoding method handles features distribution over the time from different point of views (GADF and GASF), representing time series data in multi-channel images.

Table 11 shows the performance metrics of Model 1 on the six available fault types ($[0, 5]$), based on the different coding techniques and pre-processing approaches.

The results show that Model 1 far outperforms VGG-like, this was expected since the latter was pre-trained, instead Model 1 was optimized for the type of tasks to be performed.

7.1.1. Comparison with other models (LSTM,GRU,XGBoost, ResNet-50, DenseNet-121 and VGG-16)

Having assessed the overall performance of our CNN models, we now compare our best performing model with respect to alternative NN approaches, which have been chosen due to their highest effectiveness outcome: an LSTM and Gated Recurrent Unit (GRU) based model, XGBoost, ResNet-50, DenseNet-121 and VGG-16. The XGBoost is a machine learning algorithm based on decision trees using a gradient boosting framework implemented via the *XGBoost* library.[7]

The LSTM and GRU model has 2 layers, each composed of 64 units, and a final activation layer (softmax). These models were chosen as they have been reported to perform well on the task. ResNet-50 is a variant of the ResNet model (He et al. (2016)) with 48 convolution levels and 1 MaxPool level and 1 Average Pool level. In a DenseNet architecture (Huang et al. (2017)) , each layer is directly connected with every other layer; specifically the DenseNet-121 version has 120 convolutions, 4 AvgPools and 1 fully connected layer. Finally VGG16 is based on (Simonyan & Zisserman (2014)), where 16 refers to the number of layers with weights, in detail there are 13 convolutional layers, 5 Max Pooling layers, and 3 dense layers. In particular XGBoost was the best

[7] `https://xgboost.readthedocs.io/en/latest/`

Fault	Metrics	Technique				
		RP + Sum(1)	**MTF + Diff(1)**	**GADF + Diff(7)**	**GASF + Exp(15)**	**WV + Exp(30)**
0	**F1-score**	22.76±1.33	20.77±0.51	32.45±1.41	31.02±2.08	26.65±1.49
	Precision	16.26±0.55	15.82±0.19	24.98±0.62	28.09±0.43	21.71±0.32
	Recall	37.61±4.37	34.91±2.19	46.33±3.75	34.97±4.84	34.98±5.09
1	**F1-score**	23.67±1.35	22.69±0.59	32.52±1.47	31.06±8.10	26.76±1.45
	Precision	16.97±0.57	15.85±0.22	25.02±0.63	27.98±0.49	21.75±0.34
	Recall	37.51±4.30	34.67±2.25	46.57±3.79	35.08±4.86	35.07±5.13
2	**F1-score**	22.45±1.36	21.47±0.53	32.63±1.45	31.11±8.09	26.67±1.46
	Precision	16.65±0.58	15.62±0.23	24.68±0.64	28.01±0.48	21.76±0.33
	Recall	37.21±4.34	35.06±2.24	46.21±3.78	35.01±4.82	35.02±5.12
3	**F1-score**	22.75±1.32	21.76±0.56	32.42±1.43	31.38±2.11	26.59±1.46
	Precision	16.54±0.56	15.61±0.24	25.08±0.62	28.03±0.45	21.74±0.29
	Recall	37.72±4.33	34.62±2.22	46.63±3.74	35.02±4.83	35.01±5.10
4	**F1-score**	23.63±1.37	21.70±0.54	32.47±1.42	30.9±2.07	26.63±1.47
	Precision	16.84±0.53	15.87±0.21	25.03±0.66	28.02±0.45	21.69±0.31
	Recall	37.86±4.37	34.89±2.21	46.37±3.74	35.03±4.81	35.04±5.12
5	**F1-score**	22.44±1.31	21.45±0.57	32.51±1.46	30.77±2.09	26.72±1.43
	Precision	15.86±0.57	15.67±0.23	25.13±0.67	28.05±0.46	21.73±0.27
	Recall	37.75±4.33	34.77±2.21	46.35±3.76	34.95±4.88	34.94±5.10

Table 10: Performance of Model 1 according to six different HDD fault types, based on the different images technique and pre-processing approaches used to generate its input.

Fault	Metrics	Technique				
		RP + Sum(1)	**MTF + Diff(1)**	**GADF + Diff(7)**	**GASF + Exp(15)**	**WV + Exp(30)**
0	**F1-score**	22.13±1.56	20.86±0.66	29.60±1.15	31.57±1.25	26.22±1.85
	Precision	15.13±0.66	13.95±0.28	23.65±0.23	29.65±0.21	22.31±0.26
	Recall	41.54±5.44	41.26±4.15	40.26±3.65	34.02±3.12	32.33±5.02
1	**F1-score**	22.18±1.61	20.89±0.67	29.65±1.14	31.60±1.24	26.34±1.84
	Precision	15.18±0.67	13.97±0.31	23.67±0.21	29.67±0.22	22.36±0.28
	Recall	41.67±5.41	41.27±4.46	40.24±3.58	34.04±3.16	32.42±5.05
2	**F1-score**	22.21±1.48	20.87±0.71	29.67±1.22	31.56±1.23	26.24±1.83
	Precision	15.13±0.71	14.05±0.26	23.54±0.26	29.54±0.23	22.31±0.27
	Recall	41.79±5.61	41.29±4.55	40.22±3.74	34.02±3.14	32.42±5.07
3	**F1-score**	22.11±1.51	20.86±0.72	29.63±1.11	31.58±1.26	26.58±1.86
	Precision	15.14±0.72	14.09±0.27	23.52±0.27	29.62±0.20	22.35±0.29
	Recall	41.87±5.43	41.31±4.45	40.25±3.75	34.02±3.11	32.37±5.03
4	**F1-score**	22.22±1.53	20.91±0.69	29.61±1.22	31.61±1.24	26.17±1.84
	Precision	15.17±0.69	14.17±0.29	23.39±0.22	29.49±0.23	22.33±0.25
	Recall	41.66±5.45	41.36±4.37	40.21±3.56	34.01±3.12	32.36±5.04
5	**F1-score**	22.17±1.43	20.83±0.71	29.62±1.18	31.62±1.28	26.19±1.83
	Precision	15.15±0.71	14.01±0.27	23.41±0.25	29.51±0.23	22.32±0.27
	Recall	41.55±5.42	41.25±4.42	40.20±3.44	34.07±3.13	32.38±5.03

Table 11: Performance of VGG-like according to six different HDD fault types, based on the different images technique and pre-processing approaches used to generate its input.

performing model in the competition organised by Alibaba.

The original dataset features (S.M.A.R.T. raw and normalised) and some of the generated features (Shift, Relative and Absolute) were used to maximise performance. We contrast the performance of the LSTM, GRU and XGBoost models with our CNN-based Model 1 coupled with GASF as its image coding method — as this turned out to be our best performing combination (Table 6).

Model	**F1-score**	**Precision**	**Recall**
XGBoost	40.19±0.60	30.03±0.41	60.85±1.02
LSTM	52.51±1.32	42.87±1.73	**67.79±2.24**
GRU	51.73±1.81	41.47±1.85	66.81±2.45
VGG-16	52.23±1.74	42.17±1.81	67.21±2.25
ResNet-50	51.91±1.71	41.38±1.75	66.92±2.32
DenseNet-121	51.22±1.83	41.16±1.87	66.24±2.47
CNN Model 1	**59.24±0.39**	**61.15±3.18**	57.62±2.61

Table 12: Performances of the CNN Model 1 with respect to six state-of-the-art ones.

Model	**TP**	**FP**	**FN**	**TN**
XGBoost	83	136	51	161
LSTM	96	127	44	166
GRU	89	122	50	170
VGG-16	90	121	51	169
ResNet-50	88	121	55	167
DenseNet-121	87	123	53	168
CNN Model 1	103	67	71	190

Table 13: Confusion matrices (median values over repeated tests)

Table 12 compares our best model against our chosen benchmark models. It can be seen that our Model 1 performs better in terms of F1-score and Precision, while the LSTM model is the best with respect to Recall. Furthermore, the

Model	Memory usage	Training time (seconds)
XGBoost	7 MB	780 (2000 estimators)
LSTM	850 kB	6 s/epoch (best at 10-th epoch)
GRU	767 kB	**5 s/epoch** (best at 15-th epoch)
VGG-16	8 MB	12 s/epoch (best at 21-th epoch)
ResNet-50	11 MB	14 s/epoch (best at 19-th epoch)
DenseNet-121	15 MB	8 s/epoch (best at 26-th epoch)
CNN Model 1	**540 kB**	91 s/epoch (best at 25-th epoch)

Table 14: Memory usage and training time

number of true positives and true negatives for our CNN model are higher than those predicted by the XGBoost, GRU, LSTM, VGG-16, ResNet-50 and DenseNet-121 models (Table 13).

Finally, we compared all models in terms of memory usage and training time (Table 14). In terms of memory usage, we found that our model is better than both the benchmarking models. However, the best overall model in terms of training time is the GRU one.

Summarizing, Model 1 achieves highest performances in terms of efficacy and efficiency w.r.t. the VGG-based network, because the latter is a pre-trained network.

7.2. Results on NASA Bearing

In this section we discuss about the experimental results by using encoding techniques to generate images fed in input to CNN classification, whose results are computed for both labeling procedures (binary and three classes).

7.2.1. Three Classes Classification Results

We, firstly, analyze the results regarding the prediction of bearing health status in three classes using CNN, whose output represents the probability that a sample belongs to one of the three classes. As we can see in Table 15 the

Encoding Techniques	Accuracy	Precision	Recall	F1-Score
GAF+Diff(7)	0.80±0.02	0.84±0.01	0.70±0.02	0.75±0.02
MTF+Exp(7)	0.75±0.01	0.74±0.02	0.74±0.01	0.74±0.02
RP+ Exp(15)	**0.87±0.02**	**0.86±0.01**	**0.88±0.01**	**0.83±0.01**
WV+Exp(30)	0.79±0.02	0.76±0.02	0.73±0.01	0.75±0.02

Table 15: Performances — 3-class classification

technique that achieves the best performances in all evaluated metrics is the *Recurrence Plot*.

Furthermore, the outcome in Table 15 is supported by the analysis of confusion matrices computed on the predictions made by the model using each encoding technique on the test set (see Figure 5, 6 and 7).

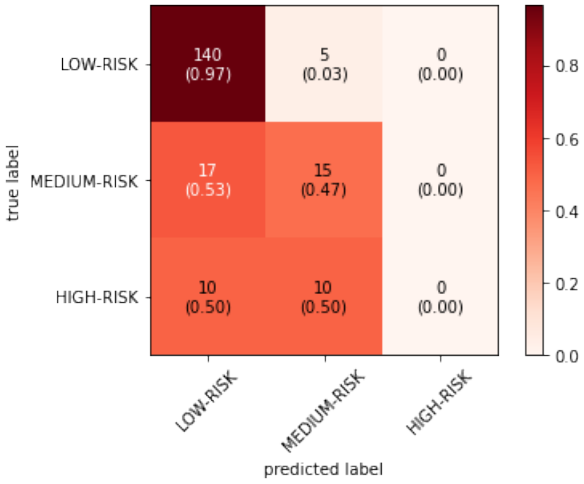

Figure 5: GAF Confusion Matrix — 3-class classification.

In particular, it is clear that the first class (LOW-RISK) is perfectly recognized by the network when we use *RP* is used as an encoding technique, while the other two classes (MEDIUM-RISK and HIGH-RISK) are confused with each other. This result is caused by the complexity of recognize the difference between these two classes which are similar from the point of view of vibration

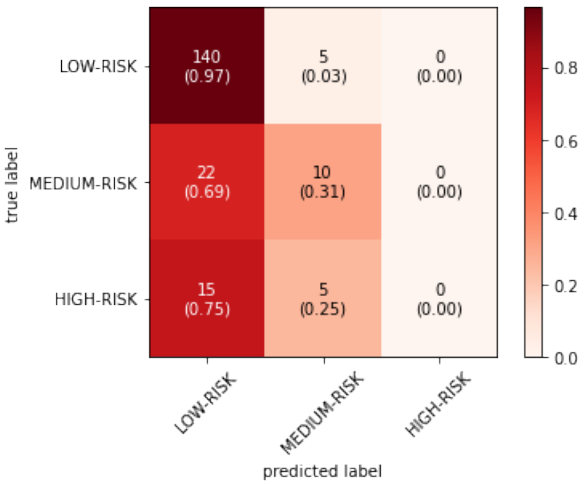

Figure 6: MTF Confusion Matrix — 3-class classification.

signals.

As is possible to see from the Figure 8 the loss curve has a normal shape with a plateau that stops around the value 0.4 for the validation subset. This indicates that the network has been trained in the correct way but that the predictions it makes do not have a high percentage factor.

7.2.2. Two Classes Classification Results

In this section, we analyze the results about the prediction of bearing health status in two classes using CNN by combining the classes MEDIUM-RISK and HIGH-RISK into one building a 2-Band classification model. Table 16 shows that the best performance in all metrics has been achieved by **Recurrence Plot**, also in this task.

Furthermore, the outcome in Table 16 is supported by the analysis of confusion matrices calculated on the predictions made by the model on the test set (see Table 17).

It is easy to note that the designed network achieves high performance using only two classes, improving model prediction assurance; in fact, the confusion matrices shows that the numbers of *False Positive* and *False Negative* are very

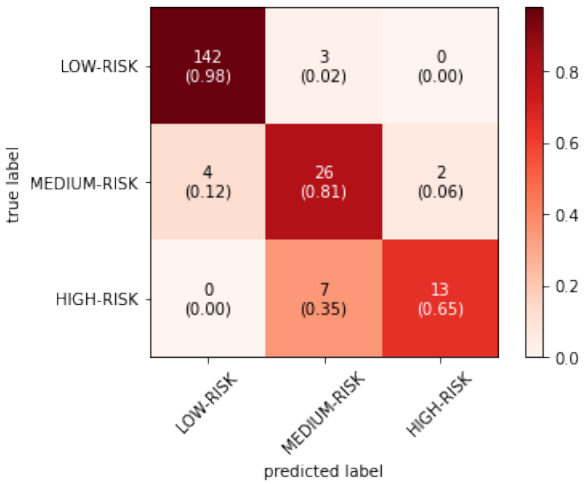

Figure 7: Recurrence Plot Confusion Matrix — 3-class classification.

Encoding Techniques	Accuracy	Precision	Recall	F1-Score
GAF+Diff(7)	0.85±0.02	0.84±0.01	0.83±0.02	0.84±0.02
MTF+Exp(7)	0.81±0.01	0.80±0.02	0.79±0.01	0.80±0.02
RP+Exp(15)	**0.96±0.02**	**0.95±0.01**	**0.95±0.01**	**0.95±0.01**
WV+Exp(30)	0.83±0.01	0.82±0.02	0.81±0.01	0.81±0.01

Table 16: Performances — 2-class classification

small.

As is possible to see from the Figure 9 also in this case the loss curve has a normal shape with a plateau that stops around the value 0.2 for the validation subset. This indicates that the network has been trained in the correct way and that the predictions it makes have a *better percentage factor* than the case with 3 classes.

7.2.3. Comparison with different baselines

In this section, we compared the designed model with respect to those achieved from two reference models (see Table 18) on the binary classification task: the first one is a LSTM, typically used to classify time series for predictive

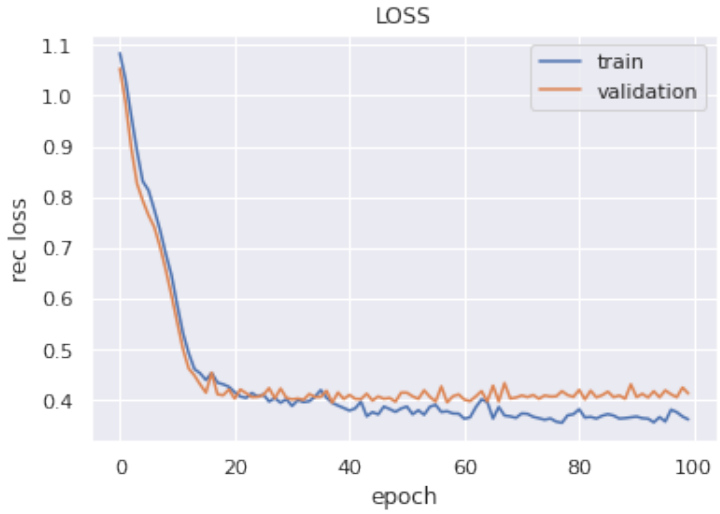

Figure 8: Model loss with RP — 3-class classification.

Technique	TP	FP	FN	TN
GAF+Diff(7)	131	14	19	33
MTF+Exp(7)	138	7	17	35
RP+Exp(15)	140	5	6	46
WV+Exp(30)	139	8	16	34

Table 17: Confusion matrices (median values over repeated tests) — 2-class classification

maintenance task - whose structure is made of 2 layers, each composed of 64 units, and a final layer with softmax activation function - and the one described in (Roy et al. (2018)), whose classifier achieved the best performance on the classification task using the NASA bearing dataset.

It is worth to note that the proposed network does not exceed the performance of the (Roy et al. (2018)) while it achieved better result than LSTM network in term of *Accuracy* and *F1-score*. Table 19 shows the efficiency performance of the designed model with respect to the two examined in Table 18 in terms of the size of the model in memory and the mean training time required.

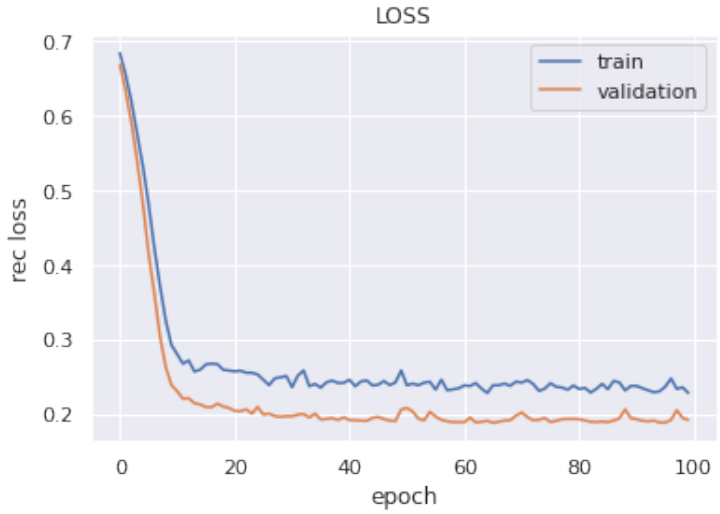

Figure 9: Model loss with RP — 2-class classification

Model	Accuracy	F1-score
Roy et al. (2018)	**0.98±0.01**	**0.97±0.01**
LSTM	0.90±0.02	0.91±0.02
Proposed CNN	0.96±0.02	0.95±0.01

Table 18: Performances of the three compared models.

It is easy to note that the proposed network is the best model in terms of *Memory Occupation* parameter whilst it is faster than Roy et al. (2018) but it is slower than the LSTM model in terms of *Training Time* parameter although the latter achieves worst performance.

To summarize, the proposed model achieves almost similar performance while it obtains better Training Time and Memory Occupation w.r.t. Roy et al. (2018). The advantage of this approach is twofold: on one hand, it can be used for supporting different learning strategies with the aim to increase classification effectiveness (e.g. online learning, active learning) and, on other hand, it can optimize resource allocation, requirements and power consumption.

Model	Memory usage	Training time (seconds)
Roy et al. (2018)	5 MB	60 s/epoch (b. at 60-th ep)
LSTM	850 kB	**10 s/epoch** (b. at 20-th ep)
Proposed CNN	**380 kB**	20 s/epoch (b. at 50-th ep)

Table 19: Memory usage and training time

7.3. Benefits of GAN

Despite several strategies have been proposed for data augmentation purpose, some of them (i.e., rotating and flipping) in the encoded image domain will distort the time domain signal, which is obviously unreasonable (see (Lu & Tong (2019b)) for some examples). For this reason, we analyze how the performance of the discussed methodology varies adopting a Generative adversarial Networks. In particular, Table 20 shows that while using a GAN helps in the training process by providing a slight increase in performance, this small advantage has to be balanced with heavier demands on training time and memory resources.

To deal with the potential drawback failure labels are still within the minority class, we developed a GAN to be used to increase the number of samples in the minority class (see Figure 10). The GAN uses a CNN (**discriminator**) to distinguish real images from false ones generated by another CNN (**generator**), that takes random samples from a Gaussian distribution.

The GAN model is based on jointly training the discriminator and the generator model, whose architectural designed are shown in Figure 11 and 12: the former is trained on a batch composed by half fake and half real samples and the latter is updated on the loss of the discriminator when frozen. Then the discriminator model has to predict the probability of a given input image to be assigned a label of class '0' (fake) and '1' (real). The generator aims to maximise the probability of the discriminator predictions of "truthfulness" for the artificially generated images. If the discriminator predicts a low average probability of truthfulness for the artificially generated images, this will result in a

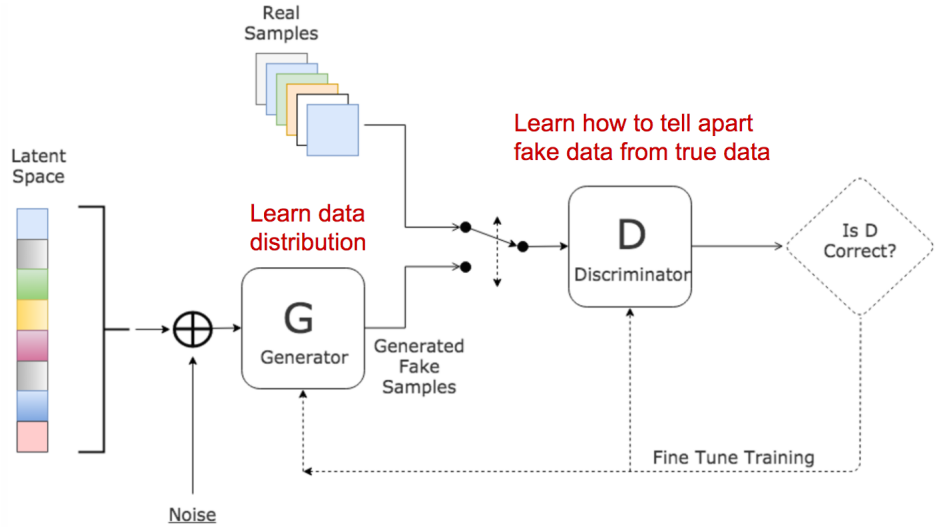

Figure 10: GAN architecture - It consists of two sub-models, Generator and Discriminator. The former is responsible for generating new plausible examples from the problem domain. The second one is used to classify examples as real (from the domain) or false (generated).

large back-propagated error signal in the generator. Consequently, this error will bring a relatively large feedback to the generator to improve its ability to generate "good" false samples in the next batch.

We further justified our assumption observing the generator's loss behavior. For the sake of simplicity, we show the loss about HDD dataset in Figure 13, where it is easy to note that the generator's loss becomes almost constant between 2000 and 3000 (number of batches) suggesting that the generator is behaving positively.

7.4. Combination of Encoding strategies

Due to the advances proposed in the recent literature (Ahmad et al. (2021); Ahmad & Khan (2021)), an ensemble of encoding techniques has been applied by combining various strategies, described in Section 3, in order to improve the performance and reliability of the model. In fact, for each sample, a single three channel input volume ($40 \times 40 \times 3$) is obtained through three different images generated by applying three different encoding strategies (GAF, MTF, RP),

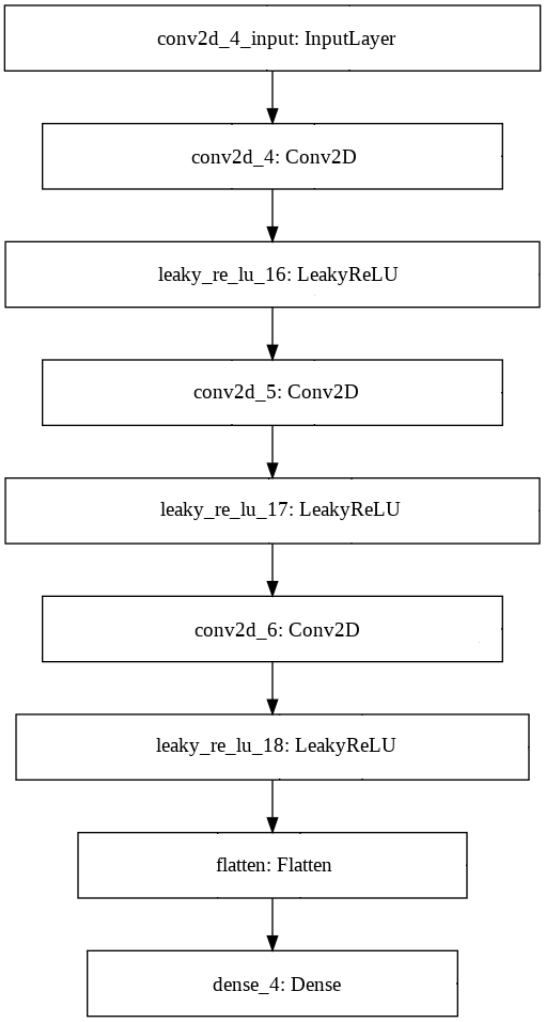

Figure 11: Discriminative network

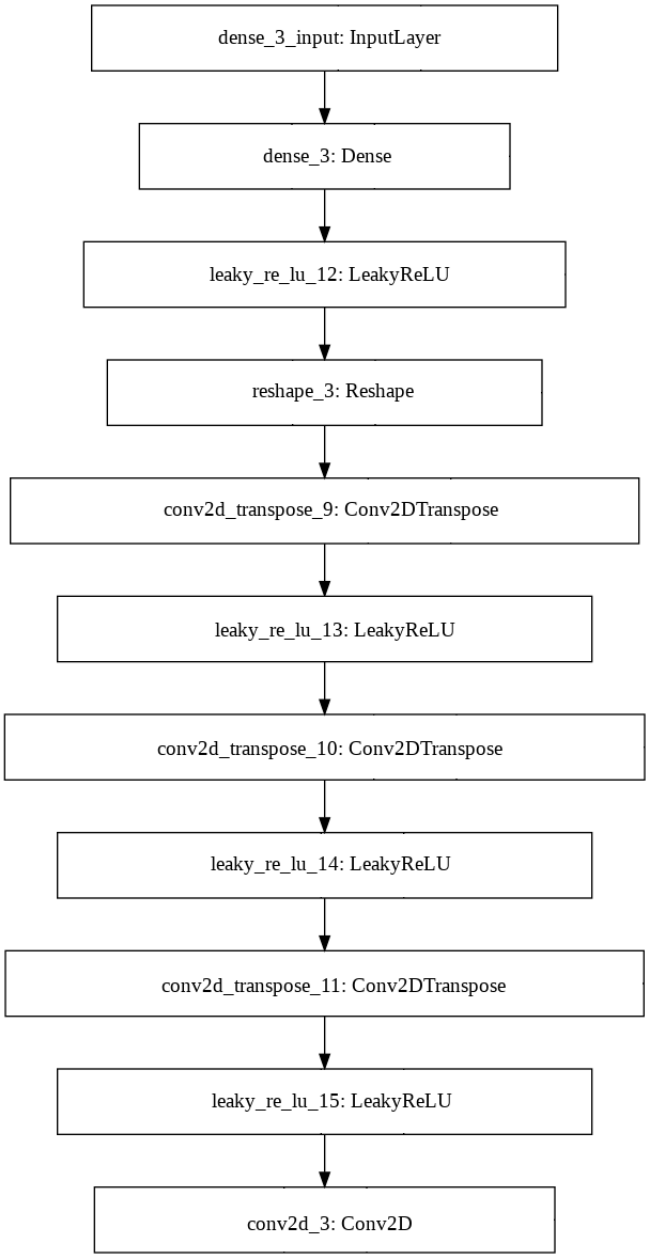

Figure 12: Generative network

	Technique	F1-Score	Precision	Recall
Alibaba	Without GAN (GASF + Exp(15))	31.59±1.25	29.58±0.22	34.03±3.13
	With GAN (+25% fake tensors)	**34.47±2.13**	**32.46±0.22**	37.02±4.96
	With GAN (+50% fake tensors)	32.52±1.24	27.43±0.66	**40.16±3.91**
Bearing	Without GAN (RP + Exp(15))	0.95±0.01	0.95±0.01	0.95±0.01
	With GAN (+25% fake tensors)	**0.97±0.02**	**0.97±0.03**	0.96±0.02
	With GAN (+50% fake tensors)	0.96±0.01	0.96±0.02	**0.97±0.01**

Table 20: Results of data augmentation with GAN on Alibaba HDD and NASA Bearing datasets.

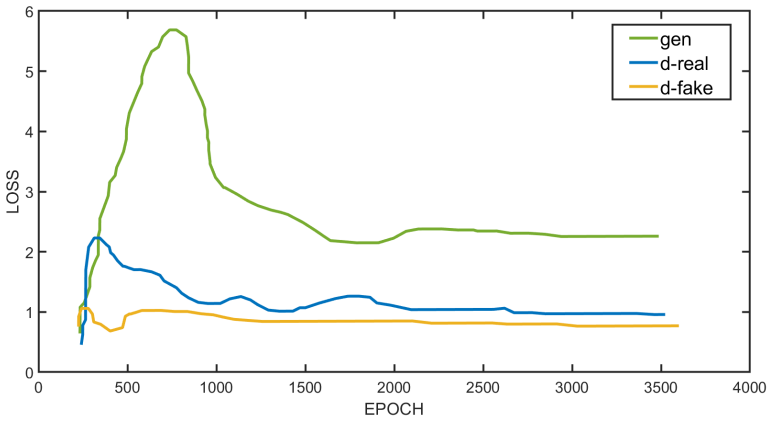

Figure 13: Loss plot for real and fake samples, and the generator

respectively. We used the same network parameters and the related training from previous experiments, only modifying the network input according to the new input. In Table 21 the experimental results compared the performance of the 1 and 3 channel networks. They show a slight increase in the confidence interval of the combination strategy, being the loss statistically smaller in the second network, although the performance in terms of accuracy and F1 is not statistically different.

Encoding Type	**Accuracy**	**F1-score**	**Loss**
1-Channel	0.96±0.02	0.95±0.01	0.06±0.01
3-Channel	0.96±0.01	0.94±0.01	**0.01±0.01**

Table 21: Performance of 3-channel encoding

8. Discussion & Conclusions

The increasing internal complexity of industrial systems has made preventive maintenance and effective monitoring techniques a fundamental necessity. In this sense, well-done predictive maintenance has been shown to bring several advantages to a variety of businesses and industrial settings. For instance, in the context of large data centers, being able to correctly predict the exact moment in time in which an HDD will become faulty can preventing costly data losses and unexpected service down-times. Furthermore, monitoring and diagnostics of mechanical components is a need for every maintenance center in Industry.

Importantly, the digital advances of the last decade make it so that huge amount of data about the inner workings of industrial systems at each level can be made available in real time. What seems to be essential, then, is to develop sound and reliable techniques that can effectively exploit this richness of information.

With this in mind, in this paper we offered an evaluation framework to

benchmark on predictive maintenance tasks some of the most diffused time series encoding techniques together with *Convolutional Neural Network* (CNN) image classifiers. Image classifiers have been shown to handle extremely well some of the most prominent shortcoming of the data available for predictive maintenance (e.g., missing data or data sparsity). Thus, it seems important to explore the performance of these models when combined with techniques to convert time series data from industrial processes into image encoding.

We considered four types of encoding methods, and evaluated two different CNN models on the *PAKDD2020 Alibaba AI Ops Competition* and the *NASA Bearing* datasets, containing information about HDD health status in big data centers and vibration signal of bearing recorded using a time window with a duration of 1 second, respectively. Additionally, we explored the hypothesis that adopting a GAN could further improve a model's performance. As it turns out, while the addition of a GAN to our training pipeline did slightly increase overall prediction performance, this was at the cost of significant additional training time and computing resources. Thus, we suggested that the overall performance increase is not enough to justify the additional computing costs.

In a second evaluation step, we compared our best performing combination of CNN model and encoding technique with respect to three and two benchmarking neural network models on the *Alibaba* and *NASA bearing* datasets, respectively As a baseline, we considered an LSTM model and XGBoost — which had achieved the top scores at the *PAKDD2020 Alibaba AI Ops Competition* in 2020 — and another LSTM model and another described in Roy et al. (2018) — whose classifier achieved the best performance on the classification task using the *NASA bearing dataset*. We extensively discussed the trade-offs between computing resources and general performance of our model compared with these two benchmarking approaches, across a variety of evaluation metrics.

While the CNN model trained on image encoding of time series performed well when compared to the other models, its increased performance has to be balanced with heavier resource commitments — for instance in terms of time. In particular, we can note that the use of deep-learning based model, whose

input is generated by encoding techniques, enables a more easily training process using well-known encoding techniques, also reducing the tendency to exhibit vanishing gradients.

Overall, in the novel context of predictive maintenance tasks, our results support the combined use of image encoding techniques with neural network models like CNNs. Moreover, the work in this paper shows the importance of conducting extensive cross-model evaluations across a variety of tasks.

Summarizing, we can see that the proposed approach achieves results similar or better than the state of the art for both datasets, also achieving highest performance in terms of efficiency.

To address some of the shortcomings highlighted by our results, future work will explore the combination of encoding techniques with a **tiled CNNs** (Ngiam et al. (2010)), which have been shown to be computationally more efficient than standard CNNs. Moreover, additional focus on developing a more effective GAN could be beneficial. Finally, in this paper we tested performances of a CNN model coupled with a single image encoding technique. In this sense, exploring the possibility of adopting an ensemble model could lead to further increases in classification performance, as well as investigating XAI approaches for explaining mis-classification in order to support practitioners in their job.

References

Addison, P. S. (2005). Wavelet transforms and the ECG: a review. *Physiological measurement*, *26*, R155–R199. doi:10.1088/0967-3334/26/5/r01.

Afonso, L. C., Rosa, G. H., Pereira, C. R., Weber, S. A., Hook, C., Albuquerque, V. H. C., & Papa, J. P. (2019). A recurrence plot-based approach for parkinson's disease identification. *Future Generation Computer Systems*, *94*, 282–292. doi:https://doi.org/10.1016/j.future.2018.11.054.

Ahmad, Z., & Khan, N. (2021). Inertial sensor data to image encoding for human action recognition. *IEEE Sensors Journal*, *21*, 10978–10988. doi:10.1109/JSEN.2021.3062261.

Ahmad, Z., Tabassum, A., Guan, L., & Khan, N. M. (2021). Ecg heartbeat classification using multimodal fusion. *IEEE Access*, *9*, 100615–100626. doi:10.1109/ACCESS.2021.3097614.

Akansu, A. N., Haddad, R. A., Haddad, P. A., & Haddad, P. R. (2001). *Multiresolution signal decomposition: transforms, subbands, and wavelets*. Academic press.

Alizadeh, M., & Ma, J. (2021). A comparative study of series hybrid approaches to model and predict the vehicle operating states. *Computers & Industrial Engineering*, *162*, 107770. doi:https://doi.org/10.1016/j.cie.2021.107770.

Anantharaman, P., Qiao, M., & Jadav, D. (2018). Large scale predictive analytics for hard disk remaining useful life estimation. In *2018 IEEE International Congress on Big Data (BigData Congress)* (pp. 251–254). doi:10.1109/BigDataCongress.2018.00044.

Barra, S., Carta, S. M., Corriga, A., Podda, A. S., & Recupero, D. R. (2020). Deep learning and time series-to-image encoding for financial forecasting. *IEEE/CAA Journal of Automatica Sinica*, *7*, 683–692. doi:10.1109/JAS.2020.1003132.

Basak, S., Sengupta, S., & Dubey, A. (2019). Mechanisms for integrated feature normalization and remaining useful life estimation using lstms applied to hard-disks. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 208–216). doi:10.1109/SMARTCOMP.2019.00055.

Benavoli, A., Corani, G., Demšar, J., & Zaffalon, M. (2017). Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. *The Journal of Machine Learning Research*, *18*, 2653–2688.

Bugueño, M., Molina, G., Mena, F., Olivares, P., & Araya, M. (2021). Harnessing the power of cnns for unevenly-sampled light-curves using markov

transition field. *Astronomy and Computing*, *35*, 100461. doi:https://doi.org/10.1016/j.ascom.2021.100461.

Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., da P. Francisco, R., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, *137*, 106024. doi:https://doi.org/10.1016/j.cie.2019.106024.

Cañas, H., Mula, J., Díaz-Madroñero, M., & Campuzano-Bolarín, F. (2021). Implementing industry 4.0 principles. *Computers & Industrial Engineering*, *158*, 107379. doi:https://doi.org/10.1016/j.cie.2021.107379.

Chan, S., Oktavianti, I., & Puspita, V. (2019). A deep learning cnn and ai-tuned svm for electricity consumption forecasting: Multivariate time series data. In *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (pp. 0488–0494). doi:10.1109/IEMCON.2019.8936260.

Chen, R., Huang, X., Yang, L., Xu, X., Zhang, X., & Zhang, Y. (2019). Intelligent fault diagnosis method of planetary gearboxes based on convolution neural network and discrete wavelet transform. *Computers in Industry*, *106*, 48–59. doi:https://doi.org/10.1016/j.compind.2018.11.003.

Chen, W., Jiang, M., Zhang, W.-G., & Chen, Z. (2021a). A novel graph convolutional feature based convolutional neural network for stock trend prediction. *Information Sciences*, *556*, 67–94. URL: https://www.sciencedirect.com/science/article/pii/S0020025520312342. doi:https://doi.org/10.1016/j.ins.2020.12.068.

Chen, Z., Wu, M., Zhao, R., Guretno, F., Yan, R., & Li, X. (2021b). Machine remaining useful life prediction via an attention-based deep learning approach. *IEEE Transactions on Industrial Electronics*, *68*, 2521–2531. doi:10.1109/TIE.2020.2972443.

Dalzochio, J., Kunst, R., Pignaton, E., Binotto, A., Sanyal, S., Favilla, J., & Barbosa, J. (2020a). Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges. *Computers in Industry*, *123*, 103298. doi:https://doi.org/10.1016/j.compind.2020.103298.

Dalzochio, J., Kunst, R., Pignaton, E., Binotto, A., Sanyal, S., Favilla, J., & Barbosa, J. (2020b). Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges. *Computers in Industry*, *123*, 103298. doi:https://doi.org/10.1016/j.compind.2020.103298.

De Santo, A., Galli, A., Gravina, M., Moscato, V., & Sperli, G. (2020). Deep learning for hdd health assessment: an application based on lstm. *IEEE Transactions on Computers*, (pp. 1–1). doi:10.1109/TC.2020.3042053.

Duin, R. P. (1996). A note on comparing classifiers. *Pattern Recognition Letters*, *17*, 529–536. doi:https://doi.org/10.1016/0167-8655(95)00113-1.

Eckmann, J.-P., Kamphorst, S. O., Ruelle, D. et al. (1995). Recurrence plots of dynamical systems. *World Scientific Series on Nonlinear Science Series A*, *16*, 441–446.

Fahim, M., Fraz, K., & Sillitti, A. (2020). Tsi: Time series to imaging based model for detecting anomalous energy consumption in smart buildings. *Information Sciences*, *523*, 1–13. doi:https://doi.org/10.1016/j.ins.2020.02.069.

Ferraro, A., Galli, A., Moscato, V., & Sperlí, G. (2020). A novel approach for predictive maintenance combining gaf encoding strategies and deep networks. In *2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application (DependSys)* (pp. 127–132). doi:10.1109/DependSys51298.2020.00027.

Fink, O., Wang, Q., Svensén, M., Dersin, P., Lee, W.-J., & Ducoffe, M. (2020). Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial*

Intelligence, *92*, 103678. doi:https://doi.org/10.1016/j.engappai.2020.103678.

Gao, R., Du, L., Duru, O., & Yuen, K. F. (2021). Time series forecasting based on echo state network and empirical wavelet transformation. *Applied Soft Computing*, *102*, 107111. doi:https://doi.org/10.1016/j.asoc.2021.107111.

Geng, S., & Wang, X. (2022). Predictive maintenance scheduling for multiple power equipment based on data-driven fault prediction. *Computers & Industrial Engineering*, *164*, 107898. doi:https://doi.org/10.1016/j.cie.2021.107898.

Giordano, D., Giobergia, F., Pastor, E., La Macchia, A., Cerquitelli, T., Baralis, E., Mellia, M., & Tricarico, D. (2022). Data-driven strategies for predictive maintenance: Lesson learned from an automotive use case. *Computers in Industry*, *134*, 103554. doi:https://doi.org/10.1016/j.compind.2021.103554.

Guillaume, A., Vrain, C., & Wael, E. (2020). Time series classification for predictive maintenance on event logs. *arXiv preprint arXiv:2011.10996*, .

Han, H., Cui, X., Fan, Y., & Qing, H. (2019). Least squares support vector machine (ls-svm)-based chiller fault diagnosis using fault indicative features. *Applied Thermal Engineering*, *154*, 540–547. doi:https://doi.org/10.1016/j.applthermaleng.2019.03.111.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). doi:10.1109/CVPR.2016.90.

Hong, Y.-Y., Martinez, J. J. F., & Fajardo, A. C. (2020). Day-ahead solar irradiation forecasting utilizing gramian angular field and convolutional long short-term memory. *IEEE Access*, *8*, 18741–18753. doi:10.1109/ACCESS.2020.2967900.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2261–2269). doi:10.1109/CVPR.2017.243.

Kiangala, K. S., & Wang, Z. (2020). An effective predictive maintenance framework for conveyor motors using dual time-series imaging and convolutional neural network in an industry 4.0 environment. *IEEE Access*, *8*, 121033–121049. doi:10.1109/ACCESS.2020.3006788.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio, & Y. LeCun (Eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. URL: http://arxiv.org/abs/1412.6980.

Krishna, S. T., & Kalluri, H. K. (2019). Deep learning and transfer learning approaches for image classification. *International Journal of Recent Technology and Engineering (IJRTE)*, *7*, 427–432.

Liang, P., Deng, C., Wu, J., Yang, Z., Zhu, J., & Zhang, Z. (2019). Compound fault diagnosis of gearboxes via multi-label convolutional neural network and wavelet transform. *Computers in Industry*, *113*, 103132. doi:https://doi.org/10.1016/j.compind.2019.103132.

Lima, F. D. S., Pereira, F. L. F., Leite, L. G. M., Gomes, J. P. P., & Machado, J. C. (2018). Remaining useful life estimation of hard disk drives based on deep neural networks. In *2018 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–7). doi:10.1109/IJCNN.2018.8489120.

Liu, J., Pan, C., Lei, F., Hu, D., & Zuo, H. (2021a). Fault prediction of bearings based on lstm and statistical process analysis. *Reliability Engineering & System Safety*, *214*, 107646. doi:https://doi.org/10.1016/j.ress.2021.107646.

Liu, Y., Wang, K., Li, G., & Lin, L. (2021b). Semantics-aware adaptive knowledge distillation for sensor-to-vision action recognition. *IEEE Transactions on Image Processing*, *30*, 5573–5588. doi:10.1109/TIP.2021.3086590.

Liu, Z., & Zhang, L. (2020). A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings. *Measurement*, *149*, 107002. doi:https://doi.org/10.1016/j.measurement.2019.107002.

Loutas, T. H., Roulias, D., & Georgoulas, G. (2013). Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic e-support vectors regression. *IEEE Transactions on Reliability*, *62*, 821–832. doi:10.1109/TR.2013.2285318.

Lu, J., & Tong, K.-Y. (2019a). Robust single accelerometer-based activity recognition using modified recurrence plot. *IEEE Sensors Journal*, *19*, 6317–6324. doi:10.1109/JSEN.2019.2911204.

Lu, J., & Tong, K.-Y. (2019b). Robust single accelerometer-based activity recognition using modified recurrence plot. *IEEE Sensors Journal*, *19*, 6317–6324.

Markiewicz, M., Wielgosz, M., Bocheński, M., Tabaczyński, W., Konieczny, T., & Kowalczyk, L. (2019). Predictive maintenance of induction motors using ultra-low power wireless sensors and compressed recurrent neural networks. *IEEE Access*, *7*, 178891–178902. doi:10.1109/ACCESS.2019.2953019.

Marwan, N., Carmen Romano, M., Thiel, M., & Kurths, J. (2007). Recurrence plots for the analysis of complex systems. *Physics Reports*, *438*, 237–329. URL: https://www.sciencedirect.com/science/article/pii/S0370157306004066. doi:https://doi.org/10.1016/j.physrep.2006.11.001.

Nakagawa, E. Y., Antonino, P. O., Schnicke, F., Capilla, R., Kuhn, T., & Liggesmeyer, P. (2021). Industry 4.0 reference architectures: State of the art and future trends. *Computers & Industrial Engineering*, *156*, 107241. doi:https://doi.org/10.1016/j.cie.2021.107241.

Ngiam, J., Chen, Z., Chia, D., Koh, P., Le, Q., & Ng, A. (2010). Tiled convolutional neural networks. *Advances in neural information processing systems*, *23*.

Qin, Y., Chen, D., Xiang, S., & Zhu, C. (2021). Gated dual attention unit neural networks for remaining useful life prediction of rolling bearings. *IEEE Transactions on Industrial Informatics*, *17*, 6438–6447. doi:10.1109/TII.2020.2999442.

Qin, Z., Zhang, Y., Meng, S., Qin, Z., & Choo, K.-K. R. (2020). Imaging and fusing time series for wearable sensor-based human activity recognition. *Information Fusion*, *53*, 80–87. doi:https://doi.org/10.1016/j.inffus.2019.06.014.

Ragab, M., Chen, Z., Wu, M., Kwoh, C.-K., Yan, R., & Li, X. (2021). Attention-based sequence to sequence model for machine remaining useful life prediction. *Neurocomputing*, *466*, 58–68. doi:https://doi.org/10.1016/j.neucom.2021.09.022.

Ran, Y., Zhou, X., Lin, P., Wen, Y., & Deng, R. (2019). A survey of predictive maintenance: Systems, purposes and approaches. *arXiv preprint arXiv:1912.07383*, .

Rieger, T., Regier, S., Stengel, I., & Clarke, N. L. (2019). Fast predictive maintenance in industrial internet of things (iiot) with deep learning (dl): A review. In *CERC* (pp. 69–80).

Roy, M., Bose, S. K., Kar, B., Gopalakrishnan, P. K., & Basu, A. (2018). A stacked autoencoder neural network based automated feature extraction method for anomaly detection in on-line condition monitoring. doi:10.1109/SSCI.2018.8628810.

Schwab, K. (2017). *The fourth industrial revolution*. Currency.

Schwendemann, S., Amjad, Z., & Sikora, A. (2021). A survey of machine-learning techniques for condition monitoring and predictive maintenance

of bearings in grinding machines. *Computers in Industry*, *125*, 103380. doi:https://doi.org/10.1016/j.compind.2020.103380.

Serradilla, O., Zugasti, E., Rodriguez, J., & Zurutuza, U. (2022). Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects. *Applied Intelligence*, (pp. 1–31). doi:https://doi.org/10.1007/s10489-021-03004-y.

Siegel, D., Ly, C., & Lee, J. (2012). Methodology and framework for predicting helicopter rolling element bearing failure. *IEEE Transactions on Reliability*, *61*, 846–857. doi:10.1109/TR.2012.2220697.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, .

Solomon, A., Kertis, M., Shapira, B., & Rokach, L. (2022). A deep learning framework for predicting burglaries based on multiple contextual factors. *Expert Systems with Applications*, *199*, 117042. doi:https://doi.org/10.1016/j.eswa.2022.117042.

Song, Y., Gao, S., Li, Y., Jia, L., Li, Q., & Pang, F. (2021). Distributed attention-based temporal convolutional network for remaining useful life prediction. *IEEE Internet of Things Journal*, *8*, 9594–9602. doi:10.1109/JIOT.2020.3004452.

Souza, R. M., Nascimento, E. G., Miranda, U. A., Silva, W. J., & Lepikson, H. A. (2021). Deep learning for diagnosis and classification of faults in industrial rotating machinery. *Computers & Industrial Engineering*, *153*, 107060. doi:https://doi.org/10.1016/j.cie.2020.107060.

Su, C.-J., & Huang, S.-F. (2018). Real-time big data analytics for hard disk drive predictive maintenance. *Computers & Electrical Engineering*, *71*, 93–101. doi:https://doi.org/10.1016/j.compeleceng.2018.07.025.

Suaboot, J., Fahad, A., Tari, Z., Grundy, J., Mahmood, A. N., Almalawi, A., Zomaya, A. Y., & Drira, K. (2020). A taxonomy of supervised learning for idss in scada environments. *ACM Comput. Surv.*, *53*. doi:10.1145/3379499.

Sundararajan, K., & Woodard, D. L. (2018). Deep learning for biometrics: A survey. *ACM Comput. Surv.*, *51*. doi:10.1145/3190618.

Tuncer, T., Dogan, S., Pławiak, P., & Rajendra Acharya, U. (2019). Automated arrhythmia detection using novel hexadecimal local pattern and multilevel wavelet transform with ecg signals. *Knowledge-Based Systems*, *186*, 104923. doi:https://doi.org/10.1016/j.knosys.2019.104923.

Vandith Sreenivas, K., Ganesan, M., & Lavanya, R. (2021). Classification of arrhythmia in time series ecg signals using image encoding and convolutional neural networks. In *2021 Seventh International conference on Bio Signals, Images, and Instrumentation (ICBSII)* (pp. 1–6). doi:10.1109/ICBSII51839.2021.9445177.

Wang, D., Tsui, K.-L., & Miao, Q. (2018). Prognostics and health management: A review of vibration based bearing and gear health indicators. *IEEE Access*, *6*, 665–676. doi:10.1109/ACCESS.2017.2774261.

Wang, Z., & Oates, T. (2015). Imaging time-series to improve classification and imputation. In *Proceedings of the 24th International Conference on Artificial Intelligence* IJCAI'15 (p. 3939–3945). AAAI Press.

Yang, C.-L., Chen, Z.-X., & Yang, C.-Y. (2020). Sensor classification using convolutional neural network by encoding multivariate time series as two-dimensional colored images. *Sensors*, *20*. URL: https://www.mdpi.com/1424-8220/20/1/168. doi:10.3390/s20010168.

Zhang, L., Mu, Z., & Sun, C. (2018a). Remaining useful life prediction for lithium-ion batteries based on exponential model and particle filter. *IEEE Access*, *6*, 17729–17740. doi:10.1109/ACCESS.2018.2816684.

Zhang, R., Zheng, F., & Min, W. (2018b). Sequential behavioral data processing using deep learning and the markov transition field in online fraud detection. *arXiv preprint arXiv:1808.05329*, .

Zhang, W., Yang, D., & Wang, H. (2019a). Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, *13*, 2213–2227. doi:10.1109/JSYST.2019.2905565.

Zhang, W., Yang, D., & Wang, H. (2019b). Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, *13*, 2213–2227. doi:10.1109/JSYST.2019.2905565.

Zhang, Y., Hou, Y., OuYang, K., & Zhou, S. (2021). Multi-scale signed recurrence plot based time series classification using inception architectural networks. *Pattern Recognition*, (p. 108385). doi:https://doi.org/10.1016/j.patcog.2021.108385.

Zhao, H., Liu, H., Jin, Y., Dang, X., & Deng, W. (2021). Feature extraction for data-driven remaining useful life prediction of rolling bearings. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–10. doi:10.1109/TIM.2021.3059500.

Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020a). Predictive maintenance in the industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, *150*, 106889. doi:https://doi.org/10.1016/j.cie.2020.106889.

Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020b). Predictive maintenance in the industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, *150*, 106889. doi:https://doi.org/10.1016/j.cie.2020.106889.