Filière *Sciences des données et de la décision*

Final Report

Deep Learning For Aircraft Health Management

Author:
M. Sergi ROCAMORA-ARDÈVOL

Supervisors:
Pr. Joël JEZEGOU
M. Fabrice JIMENEZ

August 20, 2018

Acknowledgements

I would like to express my gratitude to my Airbus supervisor Fabrice Jiménez for his dedicated support, useful remarks and for providing me with the opportunity to take part in internal workshops, discussions and meetings as well as his *célèbres* Techno-Cafés. His proactive, outreaching attitude has been inspiring for me all along this internship.

I would like to thank the data scientist team, Meriem Ghrib, Claude Fendzi and Pierre-Yves Koening as well as Vincent Chérière for the insights, discussions and support they eagerly offered during this work. Thanks, too, to the rest of health engineers, system experts and programmers of the 0-AOG platform. Also, a shout-out to my peer interns Léo Le Lonquer, Pierre Dubouloz and Mayma El Moussaoui for sharing their interesting ideas.

Finally, I would like to thank my friends in France, as well as those back in Spain for their support and patience during my two years abroad, among whom there are some very dear people. Last and foremost, thanks to my parents Lourdes and Sergi, as their unconditional backing made this work possible.

Contents

1 Introduction **1**

1.1 Context 1

1.1.1 The company 1

1.1.2 Predictive Health Management 1

1.1.3 The 0-AOG plateau 2

1.1.4 Available data 3

1.1.5 The Innovation Stream 3

1.2 Objective 4

2 Deep Learning for Time Series Analysis **5**

2.1 Anomaly Detection 5

2.1.1 Current Industrial Approach 6

2.1.2 Emerging Trends: Machine Learning 6

2.2 Principles of Deep Learning 7

2.2.1 Neural Networks 7

2.2.2 Good Practises for Training Neural Networks 9

2.2.3 Deep Auto-Encoders 10

2.3 Considered Architectures 11

2.3.1 Long Short-Term Memory Auto-Encoder 11

2.3.2 Sequence-to-Sequence Auto-Encoder 11

2.3.3 Convolutional Neural Network Auto-Encoder 12

2.3.4 Variational Auto-Encoder 13

2.4 Approaches for Anomaly Detection 13

2.4.1 Feature generation for anomaly isolation 13
2.4.2 Time series reconstruction for anomaly detection 15
2.4.3 Feature extraction for fault characterization 15
2.5 Challenges 16

3 Case Studies 19

3.1 Times Series Reconstruction through RNN Auto-Encoders 19
3.1.1 Case Study: ATA 28 19
3.1.2 Model definition 20
3.1.3 Results 22
3.2 Spectral Analysis through CNN Auto-Encoders 24
3.2.1 Case Study: ATA 28 24
3.2.2 Architecture 25
3.2.3 Results 28
3.3 Feature Extraction for Fault Prediction and Characterisation 29
3.3.1 Case Study: Degradation Analysis in ATA 24 29
3.3.2 Model definition 29
3.3.3 Results 33

4 Conclusion 35

Acronyms

0-AOG	Zero Aircraft On Ground
A/C	Aircraft
ACMS	Aircraft Condition Monitoring System
AE	Auto-Encoder
AIRTHM	Aircraft Real-Time Health Management
CWT	Continuous Wavelet Transformation
DL	Deep Learning
ECAM	Electronic Centralised Aircraft Monitor
FFT	Fast Fourier Transformation
LSTM	Long Short-Term Memory
MSE	Mean Squared Error
NN	Neural Network
OI	Operational Interruption
PFR	Post Flight Report
RNN	Recurrent Neural Network
SAR	Smart ACMS Recording
VAE	Variational Auto-Encoder
VFG	Variable Frequency Generator

Chapter 1

Introduction

1.1 Context

1.1.1 The company

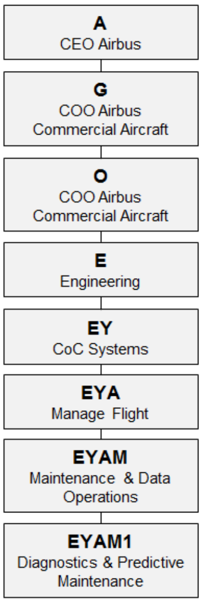
Airbus is a major actor in the global aerospace and aircraft industry, conceiving, designing, producing and delivering products and services in the field of commercial aircraft, helicopters, space and defence for a global market. It is divided into three entities, one for each main domain: Airbus Commercial Aircraft, Airbus Helicopter and Airbus Defence and Space. The present *Projet Final d'Études* was done in Airbus Commercial Aircraft, a company of more than 54.000 employees that designs and produces a family of airplanes for commercial exploitation, and offers maintenance and retrofit services.

This project took place in the EYAM1 working group (Diagnostics and Predictive Maintenance), which is part of the Engineering division of Airbus Commercial Aircraft. The detailed hierarchy structure is shown in figure 1.1. The EYAM1 team is responsible for the design and implementation of on-board maintenance systems, which monitor aircraft sensors and send fault and warning codes to the Electronic Centralised Aircraft Monitor (ECAM), in charge of informing the cabin crew, and to the Aircraft Condition Monitoring System (ACMS), which also records the sensor data along the flight and is retrieved after landing.

1.1.2 Predictive Health Management

Traditionally, the main approach to Prognosis and Health Management (PHM) for aircraft maintenance has been focused on periodical piece testing and replacement based on physical degradation models, which can offer a Remaining Useful Life (RUL) indicator, and reactive corrections after fault detections. Unexpected critical failures that led to opera-

Figure 1.1: EYAM1 upstream hierarchy.


A
CEO Airbus
G
COO Airbus Commercial Aircraft
O
COO Airbus Commercial Aircraft
E
Engineering
EY
CoC Systems
EYA
Manage Flight
EYAM
Maintenance & Data Operations
EYAM1
Diagnostics & Predictive Maintenance

tional interruptions (OI), as well as ahead-of-time piece replacements, carry non-quality costs for the airline companies.

The increased accessibility to massive data storage systems and the availability of affordable computing power have opened the doors to almost-real-time monitoring of fleet health, allowing for a much more effective PHM supervision through the ingestion and analysis of great volumes of A/C data. This opens a door to new and better methods for tackling unplanned OIs, as well as correctly estimating the RUL of the different system components, generating value for airlines and business opportunities for Airbus.

1.1.3 The 0-AOG plateau

In order to respond to exploit the business potential of these new trends, Airbus created PHM products such as the AiRTHM platform, which is referred to in the next chapter, through its SBO6 team (A/C Diagnostic and Predictive Maintenance) of the Airbus Commercial customer services division.

Recently, in an initiative to integrate expertise from both services and engineering departments, the 0-AOG plateau has been created. This transversal unit brings together experts in data science, informatics, health management and A/C systems in order to define health indicators issued from domain expertise and historical data analysis for the different programs, integrate them in a PHM pipeline, supervise fleet health and manage customer communications.

1.1.4 Available data

The platform can access the following data sources in order to observe the condition of the air fleet:

- **Post Flight Report (PFR):** List of warning and fault codes registered by the Aircraft Condition Monitoring System during a flight.
- **Smart ACMS Recording (SAR):** They are retrieved on the ground after landing, and contain high frequency data from various system sensors during certain periods of the flight. Multiple SARs are recorded during a single flight, each one identified with a number according to the recorded sensors, which usually belong to a same system.
- **Logbook:** Contains a list of the issues registered by the cabin crew on a plane and all the actions carried by the ground technicians to solve them.
- **Events:** List of Operational Interruptions (OI) of an aircraft.

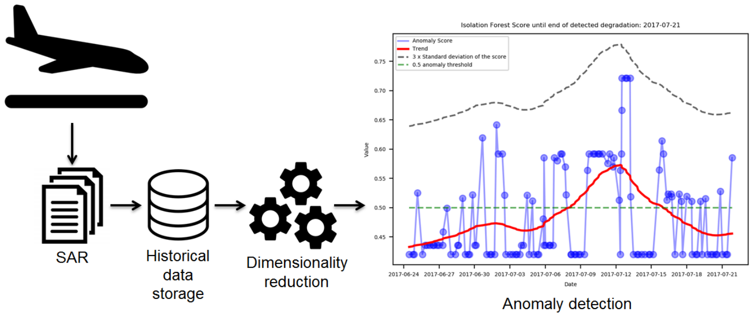
1.1.5 The Innovation Stream

Flight data consists of large sets of heterogeneous, multivariate readings produced by the embedded sensors during A/C operation. This data, most of which represents multi-modal, multi-scale physical magnitudes from different subsystems, requires systems expertise and hand-crafting in order to be usable for aircraft monitoring.

As an effort to keep up with the state of the art of the methods for anomaly detection, as well as exploring new tools for multi-discipline knowledge management, the 0-AOG plateau created the Innovation Stream.

This group, led by Philippe Chantal, has provided with various machine learning tools and demonstrators for this purpose. One of the research approaches led by Fabrice Jiménez, this internship's supervisor, has been the distributed computation of time-series features for a posterior unsupervised analysis through distance-based isolation techniques. As of now, these features consist mainly of statistical descriptors and functional decomposition coefficients (i.e. FFT and DWT). Enriching these features through machine-learning methods is one of the main motivations of this project.

Figure 1.2: Pipeline scheme of the anomaly isolation approach for SAR data.



1.2 Objective

The aim of this project is to explore Deep Learning techniques for A/C anomaly detection and analysis, as well as, if feasible, to propose and implement solutions taking into account the domain-specific needs of the aircraft maintenance activity of the 0-AOG plateau.

There is no current anomaly detection strategy or previous study in the team involving similar methods, therefore the author has been given freedom to explore a broad number of approaches and use-cases.

Chapter 2

Deep Learning for Time Series Analysis

In this chapter, the challenges of aircraft predictive health maintenance (PHM) are explained along with the interest for anomaly detection in sensor data, as well as the current industrial approaches and tendencies for this task (section 2.1).

Section 2.2 introduces the concept of deep learning and its theoretical foundations, while explaining its potential for anomaly detection. It also introduces and lists some mainstream practical pre-processing and training strategies for neural networks, which have been used for this work's experiments.

Section 2.3 contains a bibliographic review of the families of deep auto-encoders that can potentially be used for this study, commenting on their advantages and drawbacks. Inspired in these architectures, three approaches for A/C time-series analysis have been proposed in 2.4; two of these have been considered feasible and implemented.

Finally, some deep learning drawbacks which have had an impact on this work's results have been presented in section 2.5.

2.1 Anomaly Detection

A common definition for anomaly refers to any unexpected or unforeseen pattern in time series. This deviation from the expected behaviour may be originated from a system degradation or malfunction. Reciprocally, a confirmed system fault may be preceded by an anomaly in one or more of its associated sensor readings, in which case such signature becomes of interest for PHM purposes as a mean to predict a malfunction before it occurs.

2.1.1 Current Industrial Approach

Aircraft health supervision is a real-time challenge with a direct impact on an essential group of Airbus shareholders: airlines. Their A/C exploitation costs depend strongly on their reliability, as well as the number of unplanned (and therefore more expensive) maintenance activities.

As part of its services for the A350 and A380 families, the company created the Airbus Real Time Health Monitoring (AIRTHM) platform, which processes SAR data as it is generated and compute health indicators, developed by system experts, from sensor precursors for a range of A/C subsystems in order to reach the fleet operators in a streamlined fashion.

These aircraft systems experts, having a deep knowledge and experience on the analysed components and their intricate dependencies with other systems, are currently handling the task of identifying precursor anomalous patterns in sensor readings and fault root causes. This experience-based approach, however, is ill-posed for cases when highly multi-variate signals must be analysed and requires a lot of manual labour in a pool of ever-growing massive data.

2.1.2 Emerging Trends: Machine Learning

As a way to support system experts and automate certain processes in fault diagnosis through signal analysis, machine learning algorithms offer approaches that can, in a system-agnostic way, make use of latent, non-obvious information in the dataset.

A typical use-case for machine learning in anomaly detection would take a supervised approach: training a classifier on a labelled, sufficiently balanced dataset with normal and anomalous instances. Except for cases presenting a slow degradation, which have been studied in this work but are a minority, this supervised approach is not suitable for A/C anomaly detection: pathological instances are notably scarce, therefore it is impossible to construct a representative dataset.

Furthermore, flight data contains perturbations coming from the numerous, intertwined systems that form the A/C, environmental factors such as weather, ageing, component replacements and flight duration. These factors are a cause of variability on the sensor data, and create non-relevant anomalies.

Another machine learning approach on this task comes from an unsupervised analysis: the goal is not to classify flight instances as normal or anomalous, but to find the underlying common structure in the dataset through clustering or dimensional reduction among other techniques.

2.2 Principles of Deep Learning

The term deep learning (DL) comprises a family of machine learning techniques that employ neural network architectures formed by multiple layers of artificial neurons, each one of which increases its learning capability.

Deep neural networks gained notoriety in the machine learning community around 2012, when deep models demonstrated a greater performance than the contemporary state-of-the-art approaches on complex tasks such as speech recognition and image labelling.

The main interest behind DL algorithms is their capacity to automate the extraction of abstract representations (i.e. features) from the dataset. Models and NN architectures based on shallow learning such as support vector machines, decision trees and vanilla neural networks may be unable to extract useful information from complex, high-dimensional structures and relationships in the input data. In contrast, DL architectures can infer more intricate learning patterns.

This ability becomes intuitive considering that abstract representations can be decomposed into the combination of numerous simpler ones. If the DL network is interpreted a set of consecutive non-linear transformations, then the outer-most layers are hierarchical combinations of simpler, low-level patterns computed by the lower layers. This allows for a proper representations of the main factors of variation in data.

2.2.1 Neural Networks

Artificial neural networks (NN) are mathematical functions inspired by a (very) simplified interpretation of the processes occurring in the biological brain. They are defined as unidirectional graphs composed by a set of $l \in [1...L]$ layers, each one grouping a certain number i of neurons $n_i^{(l)} \in N^{(l)}$ (nodes), some of which are connected by an edge $e_{ij}^{(l)} = (n_i^{(l)}, n_i^{(l+1)})$ with an associated weight $w_{ij}^{(l)} \in W^{(l)}$. Each layer also has a trainable bias $b^{(l)}$ connected to the downstream layers.

The training of the NN on a dataset of input/output pairs and for a given error metric is done through the back-propagation algorithm (see Algorithm 1), which calculates the gradient weight-wise in order to allow for an adapted stochastic gradient descent (SGD) method application as defined in [14] (see Algorithm 2).

Algorithm 1 BackPropagation

```
Let $a_j^{(l)}$ be the activation value for a given cell $j$ in the layer $l$ for a certain input:
  procedure BACKPROPAGATION($\vec{x}$, $\vec{y}$)
      Compute activations for all neurons for the given $\vec{x}$
      for $n_{i,L}$ neuron outputs in the last layer $L$ do
          $\partial_i^{(L)} \leftarrow \frac{\partial}{\partial_{z_i}^{(L)}} \frac{1}{2} \|h_{W,b}(x) - y\|^2 = -\left(y_i - a_i^{(L)}\right) f' \left(z_i^{(L)}\right)$
      for $l = [(L-1)...1]$ do
          $\partial_i^{(l)} \leftarrow \left(\sum_{j=1}^{J_{l+1}} W_{ji}^{(l)} \partial_j^{(l+1)}\right) f'(z_i^{(l)})$
      return
          $\frac{\partial}{\partial W_{ij}^{(l)}} E(W, b; x, y) = a_j^{(l)} \partial_i^{(l+1)}$
          $\frac{\partial}{\partial b_i^{(l)}} E(W, b; x, y) = \partial_i^{(l+1)}$
```

Algorithm 2 StochasticGradientDescent

```
  procedure GRADIENTDESCENT(TrainingDataSet)
      for $j$ in $[0 : numIterations]$ do
          $TrainingBatch \leftarrow randomSubset(TrainingDataset)$
          $\Delta W^{(l)} \leftarrow 0, \Delta b^{(l)} \leftarrow 0$
          for $(x_i, y_i)$ in TrainingBatch do
              Calculate $\nabla_W^l E(W, b; y_i, \hat{y}_i)$, $\nabla_b^l E(W, b; y_i, \hat{y}_i)$ through BackPropagation
              $\Delta W^{(l)} \leftarrow \Delta W^{(l)} + \nabla_W^l E(W, b; y_i, \hat{y}_i)$
              $\Delta W^{(b)} \leftarrow \Delta W^{(b)} + \nabla_W^b E(W, b; y_i, \hat{y}_i)$
          Update network weights:
              $W^{(l)} = W^{(l)} - \alpha \left(\left(\frac{1}{m} \Delta W^{(l)}\right) + \gamma W^{(l)}\right)$
              $b^{(l)} = b^{(l)} - \alpha \left(\frac{1}{m} \Delta b^{(l)}\right)$
```

2.2.2 Good Practises for Training Neural Networks

Training a NN is a non-convex optimization problem:

$$\min_{w \in \mathbb{R}^k} f(w) = \frac{1}{M} \sum_{i=1}^{M} f_i(w) \tag{2.1}$$

where w are the NN weights (to be optimised) and f_i is the error function that calculates the deviation between the ground truth output for an input value $x_i \in \mathbb{R}^n$ and the NN output. Although there are few results guaranteeing the convergence and performance proprieties of NN for the general case, good empirical results have been achieved both in real and benchmark cases, given the adequate architecture and hyper-parameter tuning [7].

Some rules of thumb for model convergence and validation are commonly taken into account:

Model Validation

In the present case, which concerns only deep auto-encoders, the mean squared error (MSE) metric defines a similarity measure between the input and the output. In order to ensure the proper generalization of the AE model within the limits of the available data, a random holdout of 20% to 33% of the dataset has been reserved for validation[11].

This holdout is used during training in order to detect over-fitting[1]. In order to avoid information leakage, only the training dataset is used for calculating the normalization transformation.

Data Normalization

It is done with the aim of bringing all the entry variables to a similar scale and range of values, avoiding an ill-defined problem and guaranteeing a stable convergence of weights. In this work two variants have been explored:

- Standardization: The data is centred as if generated by a unitary Gaussian, $x = (x - \hat{\mu})/\sqrt{\hat{\sigma}}$.
- $[-1, 1]$-Normalization: The data is scaled to the [-1, 1] value interval. Usually, the process consists in an affine transformation where the minimum value is brought to

[1] When a statistical or machine learning model learns to classify or reproduce the training dataset too well while losing performance on the rest of the cases (therefore losing generalization properties), it is said that such model is over-fitted. In order to control this phenomenon, part of the training dataset is left out and used to test the model's generalization capacity (validation).

-1 and the maximum to 1. Since in the studied case there are no defined upper or lower bounds, the values corresponding to $3\hat{\sigma}$ of the empirical distribution of each variable have been used as maximum and minimum for the affine transformation, and the outliers have been clipped into this interval.

In order to avoid information leakage, only the training dataset is used for calculating the normalization transformation.

Batch Training

The back-propagation training of the NN is commonly based on a Stochastic Gradient Descent (SDG) algorithm or one of its variants; specifically, the ADAM method has been the tool of choice for this task. As in any SDG algorithm, the size of the data batch becomes a hyper-parameter to be defined.

There is empirical evidence that a mini-batch size of between 32 and 512 data-points is a good choice for most cases; greater values, on the other hand, tend to produce models that do not generalize as well [4].

2.2.3 Deep Auto-Encoders

Deep auto-encoders (AE) are a family of NN architectures interesting to the machine learning community for their dimensionality reduction proprieties: unlike PCA or Negative Matrix Factorization, they can represent non-linear, intricate relations between the input variables.

They consist of a multiple-layer neural network trained to approximate the identity function while being subject to a certain regularization, and can be decomposed into two sub-networks: an encoder that maps an input to a representation in a space of a generally lower dimension, and a decoder that reconstructs a vector in the input space from a vector belonging to the latent or hidden space [13]. In the usual case where the latent space is of a lower dimension than the input space, the composition of these two functions will be a projection to a hyper-surface or manifold of reconstructible inputs.

This ability to learn in an unsupervised fashion is what makes deep AE interesting for anomaly detection, where classification is not feasible as datasets are unbalanced, and the final user does not necessarily know which is the signature of the anomaly.

2.3 Considered Architectures

The following Deep AE architectures have been considered as potentially suitable for the treatment of time-series in terms of dimensionality reduction, straight-forward anomaly detection, or both.

2.3.1 Long Short-Term Memory Auto-Encoder

Recurrent Neural Networks (RNN) are a family of NN adapted to the processing of sequential data. Instead of dedicating one neuron to each input value (as in the fully connected case), each RNN cell processes and emits an activation value for every time-step. The calculation of the output is a function of the input and a hidden-state value (memory) calculated during the last step. These two values are also used for updating the mentioned hidden-state.

This temporal dependence makes RNNs able to produce very deep models with very strong representational capabilities for time-series. This comes at the cost of a computationally greedy training, as well as gradient explosion and vanishing issues [9].

In order to handle the gradient problems, the Long Short-Term Memory (LSTM) and the Gated Recurrent Unit (GRU) architectures were studied. Since benchmark tests on the difference in performance between these are inconclusive [3], the author has opted for he more mainstream LSTM. The details of the LSTM architecture will not be explored in this document; more information on it can be found in the original paper [8].

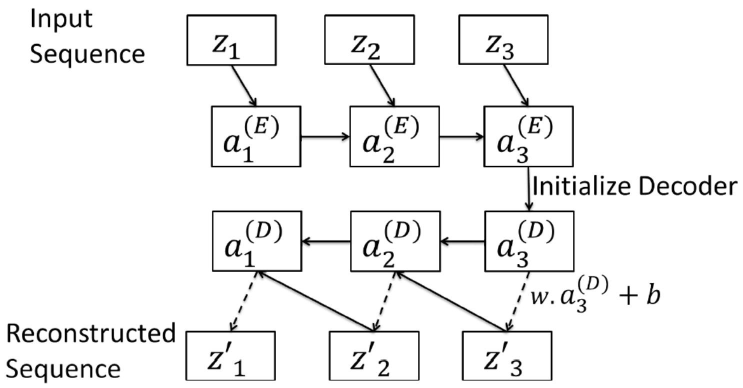
Although an LSTM AE will not be capable to reduce the temporal dimension of an input, as all RNN-inspired networks return one output time-step for each input time-step, it can do it parameter-wise; for instance, for a dataset containing temporal signals composed of n channels or parameters, an AE composed by stacked LSTM should be capable to compress the information parameter-wise and output a latent signal composed by m channels, with $m < n$. Following the auto-encoder principle, the input signal can be approximately reconstructed from this embedding, using the error as an anomaly indicator. This dimensional bottleneck will act as a regulariser.

Some works that use LSTM for anomaly reconstruction are [1][14], although they use a windowed approach. This will be referred to in the next section.

2.3.2 Sequence-to-Sequence Auto-Encoder

Most commonly known as Seq2seq, this recurrent AE was conceived for translation tasks, providing outstanding results [16]. It is composed by an LSTM-based encoder and decoder, with the particularity that the resulting encoding is the LSTM memory state at the last

Figure 2.1: Seq2seq architecture for time series encoding as proposed in [13].



time-step of the encoder. This embedding is used as an initial memory state for the decoder LSTM, which is trained to reconstruct the output in inverse order.

This architecture solves the need for temporal dimensional reduction, as it can transform time-series of any length to a fixed-length vector of abstract features. Its main drawback is its very high training cost, especially for long series, which made it impossible to apply in this work, as training a Seq2seq AE capable to model a whole flight (up to 60.000 time-steps or more) was not feasible due to computational limitations.

2.3.3 Convolutional Neural Network Auto-Encoder

This family of architectures (CNN) is the foundation stone of most of the image treatment algorithms based on neural networks.

CNN are composed by pairs of convolutional and pooling layers: the former learns discrete filters that extract the most relevant patterns on the input image, while the latter down-samples the convoluted outcome through local averages or maximal values.

Reinterpreting time-series as images of width equal to the number of samples, unitary height and number of channels equal to the number of parameters, CNNs can be proposed as alternatives to RNNs with reasonable results [2].

This approach has the advantage over LSTM of a considerable faster training time, as well as the possibility of temporal-wise dimensional reduction, allowing for latent feature space learning when implemented as an AE. Its downside is that it requires all input data to have the same dimension, requiring padding or truncation depending on the case.

2.3.4 Variational Auto-Encoder

Variational auto-encoders (VAE) are a Bayesian reinterpretation of the deterministic AE. While both types learn a lower-dimension representation of the input data, VAEs do not define a mapping to the hidden space; instead they infer a normal distribution over each of the embedding variables. These distributions can be sampled generating synthetic elements of the input space (generative network) and provide a regularizing effect (hidden values in a local vicinity will correspond to similar outputs).

As well as deterministic AEs, they can be trained through SGD thanks to the definition of a lower bound of the Kullback-Liebler difference between the inferred and the real posterior, as well as a re-parametrization trick, both presented in the original paper [10].

A recent training strategy for the VAE, the β-VAE, has been successfully used for the construction of disentangled interpretable latent spaces, where each hidden variable corresponds to one of the major sources of variation of the dataset [12].

A specific sub-architecture of VAEs called Variational Recurrent Auto-Encoder (VRAE), the Bayesian counterpart to Seq2seq AEs, has been proposed in [5]. As with Seq2seq AEs, it has not been used in this work due to its computational greediness.

VAEs and their variants have not been considered but not used for this work, as they do not present any advantage to their deterministic alternatives for the present study and do not overcome their drawbacks, which are related to data scarcity and computational limitations.

2.4 Approaches for Anomaly Detection

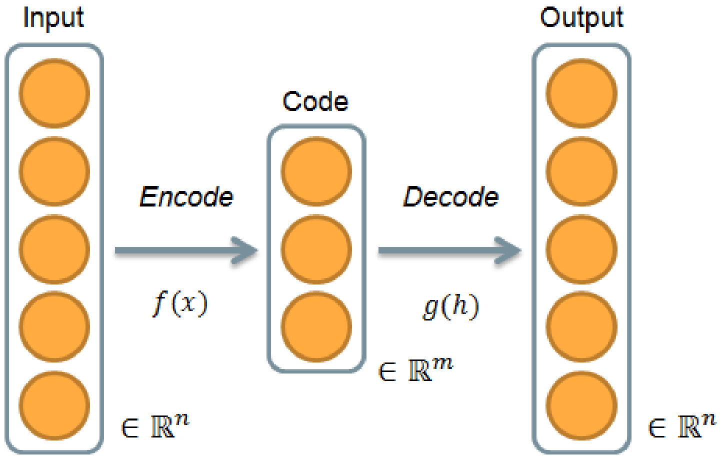
Given the architectures studied in the previous section, the nature of the studied data and the needs of the 0-AOG plateau, three different approaches have been considered for anomaly detection, of which two have been deemed feasible. About the discarded idea, the reasoning behind its proposition and dismissal is presented in this section:

2.4.1 Feature generation for anomaly isolation

As a part of its mission to provide the 0-AOG plateau with tools for aircraft system diagnosis and prognosis, the Health Anomaly Detection with Evolutionary Strategy (HADES) was developed by Fabrice Jiménez, the supervisor of this project. It is programmed to detect anomalies in a set of flights from their features, which have been previously extracted.

As a way to enrich HADES through the automatic feature generation capabilities of DL methods, the idea of training a deep auto-encoder for flight reconstruction in order

Figure 2.2: Auto-encoders are essentially compositions of non isomorphic functions.



to use the learnt features for anomaly detection was proposed. Deep learning methods, nonetheless, have been found to be ill-adapted for this task.

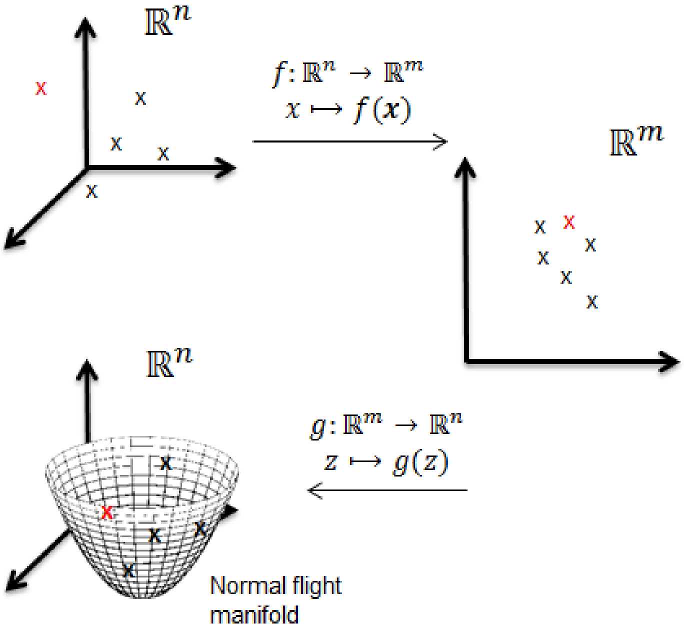
The problem comes from the fact that auto-encoders are, as commented before and shown in figure 2.2, compositions of an encoding and a decoding function. The encoder, which takes as input an element of the original space (in this case, flight data), maps it to a lower-dimension space, therefore defining a non-injective function. This means that certain families of flights will correspond to the same embedding. On the other side, the decoder defines a mapping from a lower to a higher dimension, which implies that the function will not be exhaustive.

The composition of both encoder and decoder, therefore, becomes a projection: all flights will be mapped into $\Phi = \{f \circ g(x), \forall x \in \mathbf{R}^n\}$, with $\Phi \subsetneq \mathbf{R}^n$ the manifold defined by its image. Let's see what its shape looks like.

This manifold is learnt during the training of the auto-encoder. Since the model is forced to give priority to the most explicative traits of the dataset [6], if this dataset is composed only by normal flights or has a very low ratio of anomalous flights (which is the case in this study), the projection image will be, ideally, the manifold of normal flights. Therefore, any anomaly in the data will be overlooked as an outlier and the generated latent features will not represent the variability outside the norm. A schematic of this phenomenon is shown in figure 2.3.

Due to the presented reasons, this approach has been left out of the study, as these automatically generated features will not provide relevant information for the isolation algorithm.

Figure 2.3: An auto-encoder trained on a mostly normal dataset will not reconstruct anomalous signatures; instead it will return a normalized instance of the flight. Features generated by the encoder will not encode anomalous or rare behaviours.



2.4.2 Time series reconstruction for anomaly detection

Another proposed approach is to use the auto-encoder reconstruction error for a given input (such as the mean squared error metric) in order to detect the "deviation" of a given flight from the normality. This is a common approach for unsupervised anomaly detection using neural networks [13][14][15].

The studied literature proposes the use of LSTM auto-encoders for reconstruction learning, applying the analysis on rolling windows. Nonetheless, the windowed approach has a drawback for the studied use-case: A/C data has a strong non-periodic component that depends on the flight phase and the operational context in general, which would be lost on an isolated window as it does not take into account the long-term temporal dependencies expected to be found in flight data. The AE will be trained on the whole flight instead.

2.4.3 Feature extraction for fault characterization

Given a sufficiently large number of well identified anomalous cases preceding faults on a certain system, there is interest in seeing various failure modes. This can be done through training an auto-encoder with a balanced dataset (that is to say, using a similar number of flights for each class) and extracting a flight embedding that represents the main factors of variability.

These embeddings can be studied in order to correlate the different behaviours with the fault causes.

2.5 Challenges

Neural networks in general, and DL methods in particular, have certain characteristics that make them fit for some machine learning tasks, but may pose inconveniences in other cases. This section presents some considerations regarding the particularities of both the deep learning approach and the available data for this study.

Unlabeled data

Labelling flights as anomalous or normal is not an evident task, since being anomalous depends on the rest of observed flights. Furthermore, only those anomalies that can be identified as precursors to system faults are interesting for PHM purposes, while many can correspond to unrelated perturbations.

For this study, certain system faults have been identified through Event messages and confirmed by mature algorithms currently used in the 0-AOG platform. These instances, while not numerous, will be the only way to validate the studied unsupervised approaches.

Data quality

Deep learning models are, at best, as good as the dataset on which they are trained. Inferring good predictors is only possible when the training data presents all the variability to be expected in real-life applications in a balanced way; this is far from granted in the aircraft industry, where each flight data instance is dimensionally voluminous with high variability, and instances are costly to process, which is a problem if computational capacity is not abundant. All this causes a lack of representativeness.

Interpretability

Artificial features issued from a deep encoder may not be interpretable or mappable to their associated behaviour or even physical origin, unlike in probabilistic or functional decomposition-based features. This black-box behaviour is undesirable from a diagnostics point of view, as there is a need for identifying the subsystem or component that originates the anomalous data.

Computational cost

Another constraint is computational greediness: deep learning algorithms are costly to train, with training times that depend on the dataset size, the number of weights to be trained, the architecture (recurrent, feed-forward or convolutional) and the number of necessary gradient descent iterations, which depend on the variability of the data.

For the present work, an Intel© Core^{TM} with a i5-6300U CPU was employed, which supposed a bottleneck in terms of computation and made many analysis infeasible. Due to data privacy policies, cloud computing on commercial platforms such as AWS or Google Cloud, which provide GPU instances, can not be used.

For flight data extraction and pre-processing from the AIRTHM data lake, the author has had access to the platform's on-premises Hadoop cluster, which is well-dimensioned for the task on hand.

Chapter 3

Case Studies

3.1 Times Series Reconstruction through RNN Auto-Encoders

3.1.1 Case Study: ATA 28

The ATA 28[1] contains all subsystems related to fuel containment, management and feeding.

In this analysis, the turbine fuel feeding pumps have been studied as a validation case for the LSTM time-series reconstruction approach. Each one of these components is powered by a three-phase electrical current, which is generated by a power electronics subsystem, one for each pump. These electronic boards are subject to material degradation leading to failure, leaving the fuel pump out of service. In this scenario, a secondary pump is activated in its place.

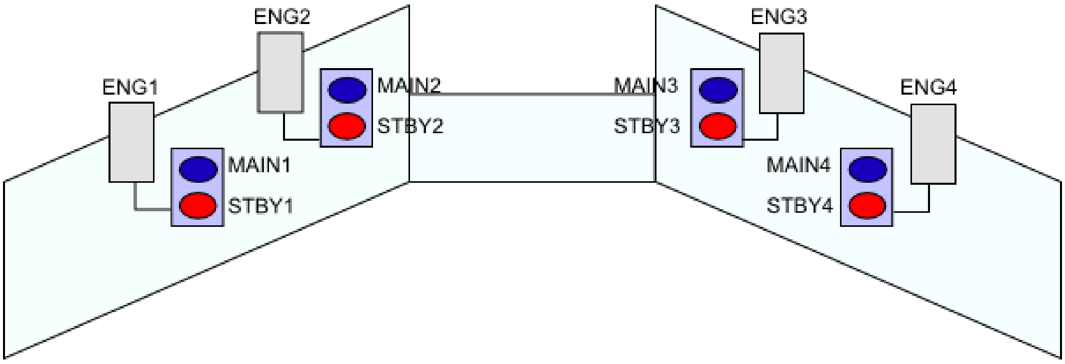
While such faults do not necessarily cause an OI, as A/C are allowed to fly with one faulty main pump as long as the secondary one is operative, there is an interest in predicting such faults in order to improve maintenance planning as well as A/C reliability.

With the aid of system experts, the key signals for this analysis were identified. For the studied anomaly, the historic SAR recordings of the electrical current intensities for each pump have been retrieved and used, as they have been confirmed to show precursor signatures by other detection algorithms.

Around a thousand flights from all available A/C have been retrieved for training. The number of retrieved samples had to be limited to this scale of values due to the difficulties

[1] ATA 28 is one of the chapters of the ATA 100 standard for numerical designation of the main systems for commercial aircraft, as defined by the Air Transportation Association of America (ATA). It is shared by the main aircraft makers such as Airbus, Boeing or Bombardier. Chapter ATA 21, for instance, refers to air conditioning, and ATA 27 to flight controls.

Figure 3.1: Fuel pump distribution on the A380.



in training for greater datasets. For the same reason, the signal (originally at 8Hz) had to be down-sampled to 1Hz, giving a final signal of between 20.000 and 60.000 data-points per channel. The implication of these decisions will be further commented.

With the support of the Health Engineering team, a list of confirmed A/C faults has been compiled with vessels from various companies in order to validate the method.

3.1.2 Model definition

The first approach for TS reconstruction in this work are stacked LSTMs. Even if, as recurrent networks, they are unable to compress data along the temporal dimension, they can in fact compress it parametrically-wise, providing with data regularization in the multivariate case (see figure 3.2).

Due to the inherent computational complexity of RNN training, the study had to be limited to a low-depth auto-encoder: for a given computation time, there was a trade-off between the amount of flights in the training dataset and the complexity of the model. Given the high variability of the data, it was considered preferable to use shallow models along with richer, more representative datasets.

The chosen architecture consisted of a 9-7-5-7-9 auto-encoder as shown in the table 3.1. *Tanh* was chosen for the internal and final activation functions, as unbounded alternatives such as the linear and rectified linear units cause gradient explosion during training. Due to this activation function, the training dataset was rescaled to the interval $[-1, 1]$ using the normalization process explained in section 2.2.2.

A random selection of 20% of training flights were left for validation. At epoch 12, validation error started platooning and the training was stopped at 85 iterations; epoch 54 was found to minimize the error.

Figure 3.2: Schematics of a (shallow) stacked RNN auto-encoder. Along the time axis, the p-dimensional input is embedded into a q-dimensional latent space, with $p > q$.

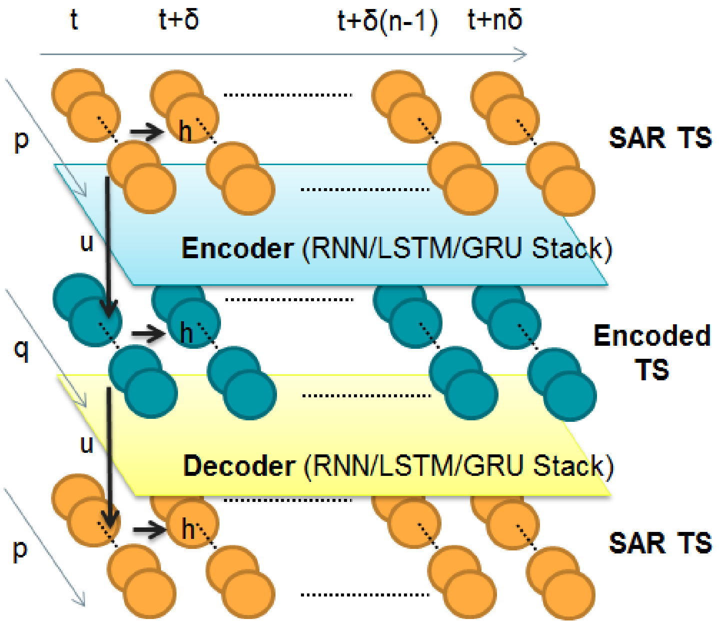
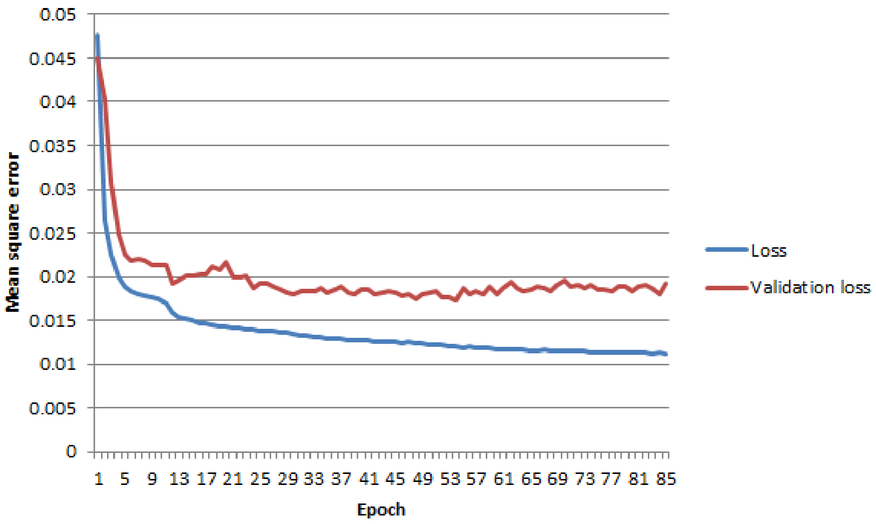


Table 3.1: Implementation detail of the LSTM auto-encoder.

Layer type	Output dimension	Activation function	Dropout
LSTM	7	Tanh	5%
LSTM	5	Tanh	5%
LSTM	7	Tanh	5%
LSTM	9	Tanh	0%

Figure 3.3: Training loss for the LSTM auto-encoder. Epoch 54 corresponds to the lowest validation MSE value.



3.1.3 Results

In order to test the model, three real fault cases outside the training time interval have been extracted and manually verified. For each confirmed fault, the reconstruction error for the two previous months of flights is computed and the evolution of the error along time is analysed for each of the channels. Intensity readings associated to the faulty feed pump are expected to raise over normality.

Results are shown in this section. Plane 1, with a fault in pump 3, presents an anomalous behaviour during all instances before degradation (figure 3.4). On the other hand, planes 2 (figure 3.5) and 3 (figure 3.6), both with a faulty pump 1, do not present a higher MSE for this signal relative to the rest of pumps. There is no consistency between malfunctions and signal error, and no apparent increase in abnormality before fault.

These negative results can be due to different causes:

- There are precursor signatures but those are masked by more apparent factors of variation. In such case unsupervised approaches will not be able to detect interesting anomalies; an alternative approach such as a model-based analysis should be taken.

- There are no precursors in the analysed signal, as they are lost to down-sampling or they are found in correlations with non analysed signals.

Figure 3.4: Reconstruction error for fuel pump currents in plane 1. Flights previous to fault at pump 3.

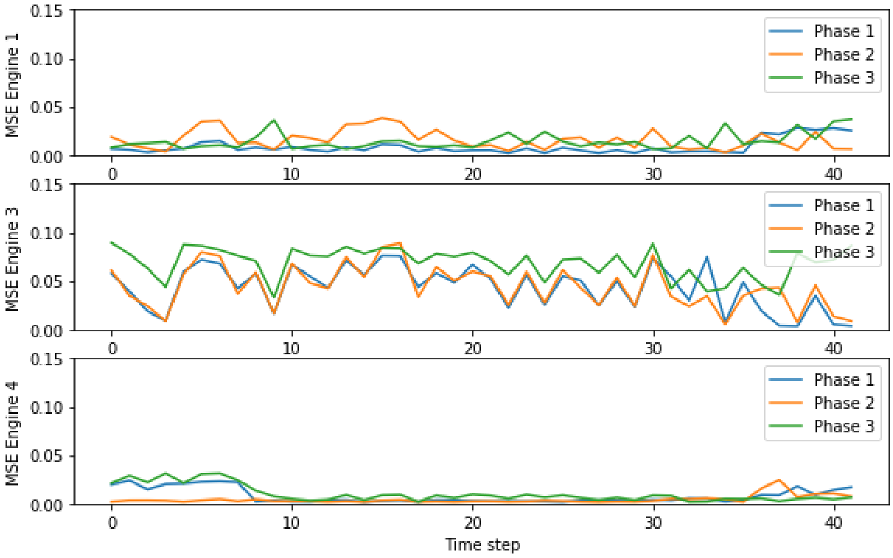


Figure 3.5: Reconstruction error for fuel pump currents in plane 2. Flights previous to fault at pump 1.

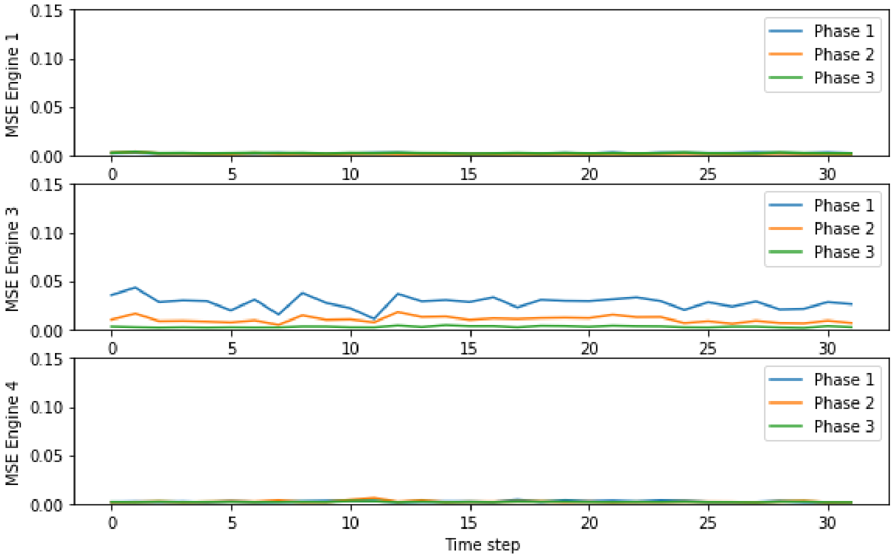
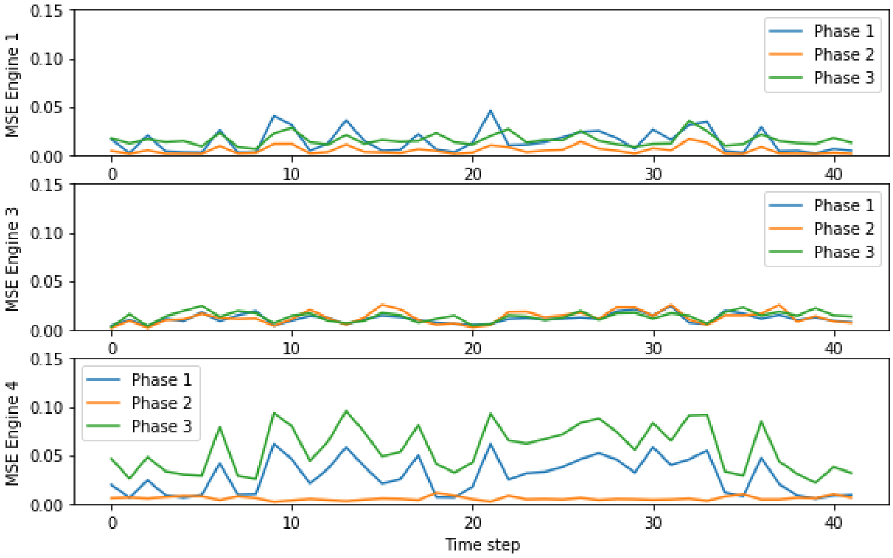


Figure 3.6: Reconstruction error for fuel pump currents in plane 3. Flights previous to fault at pump 1.



3.2 Spectral Analysis through CNN Auto-Encoders

3.2.1 Case Study: ATA 28

In the previous study, the signal was down-sampled to 1Hz from 8Hz due to computational constraints. Because of this down-sampling there is a loss of information in the frequency domain due to the Nyquist-Shannon theorem; in the case where fault precursors are found in the high frequency spectrum, relevant anomalies for the task in hand may be lost.

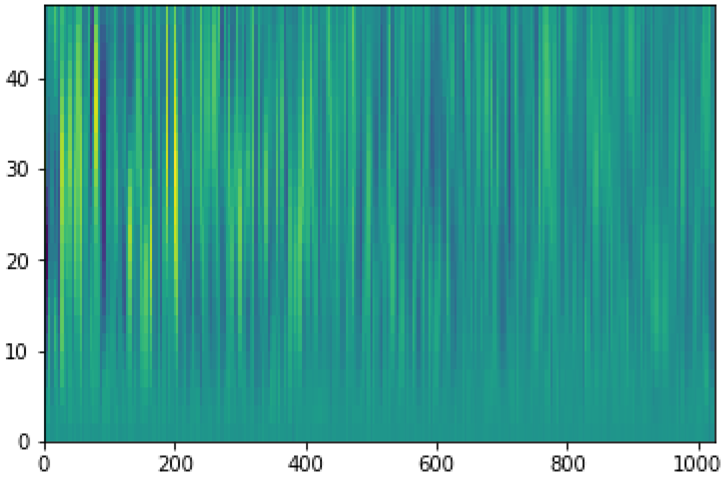
Given that the observed signal corresponds physically to an alternate current that may possibly show behavioural changes previous to an anomaly, an approach to specifically analyse anomalies in the signal frequency spectrum has been proposed:

Using a continuous wavelet transformation on the full signal (without down-sampling), a scaleogram[2] for each channel has been generated and used in order to extract the instantaneous frequency information for all along the flight. The analysed frequencies range from 8 Hz (the SAR sampling ratio) to 0,13 Hz; lower frequencies have not been explored as they have been considered to be taken into account by the previous analysis.

The resulting image had to be down-sampled in the time axis to the same size for all signals, as the used architecture required all inputs to have the same dimension. The final image dimensions are 48 wavelength amplitudes and 1024 time steps.

[2] A scaleogram is a visual representation of the spectrum of frequencies of a signal along time given a wavelet transformation.

Figure 3.7: Scaleogram of the continuous wavelet decomposition of a sample pump current signal. It has been down-sampled along the time axis to 1024 values.



3.2.2 Architecture

Based on the success of Deep Learning on image analysis and treatment, this tackled this problem through the same approaches used for the visual domain: the convolutional neural network family.

A CNN auto-encoder has been chosen in order to learn a low-dimensionality embedding of the normal spectrogram representation. Once the model has learnt to reconstruct the input, the reconstruction error is to be used as a tool for determining the anomaly of the flight in the frequency domain.

The implemented architecture consists of various consecutive pairs of convolutive and pooling layers (as in figures 3.9 and 3.10) ending with a fully connected layer mapping to a 512-dimensional latent space. The decoder consists of the same number of layers of the encoder: a first fully connected layer that reverts the last encoder mapping, and a series of alternate deconvolutive and upsampling layers. The deconvolution is just a convolutive layer with shifted entry and output channel dimensions, and the latter repeats the input in order to revert the dimensionality reduction done by the pooling layer. The combination of both sub-nets yield an auto-encoder with an architecture similar to figure 3.8.

The detailed architecture is shown in table 3.2; visual schematics of the convolution and pooling procedures are presented in figures 3.9 and 3.10.

Figure 3.8: Illustrative schematics of a CNN auto-encoder architecture.

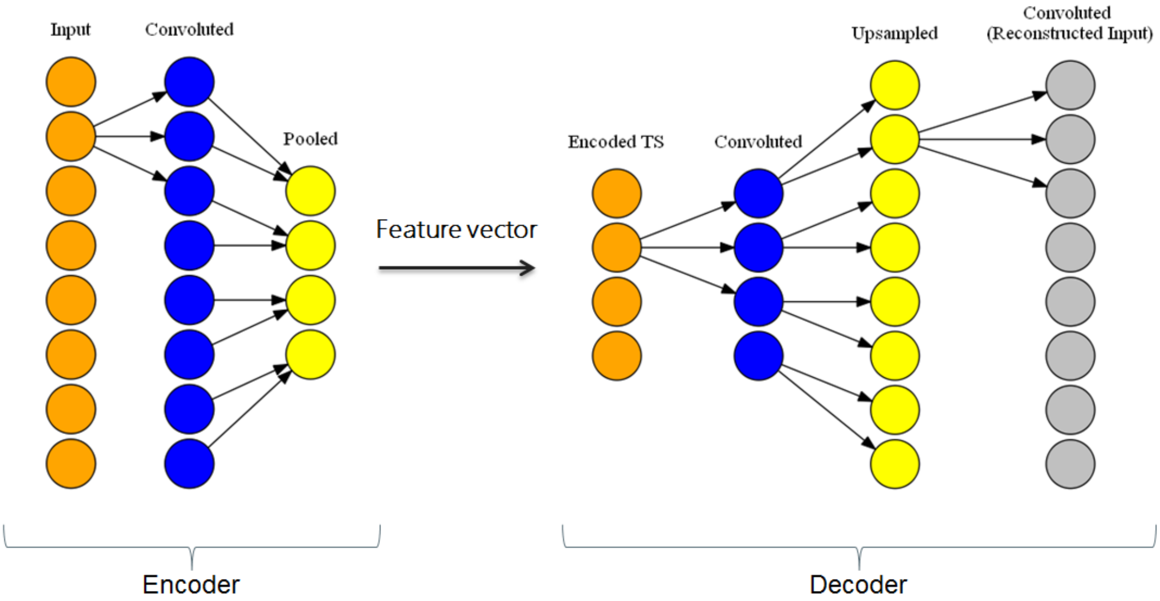


Figure 3.9: Convolution step in a CNN.

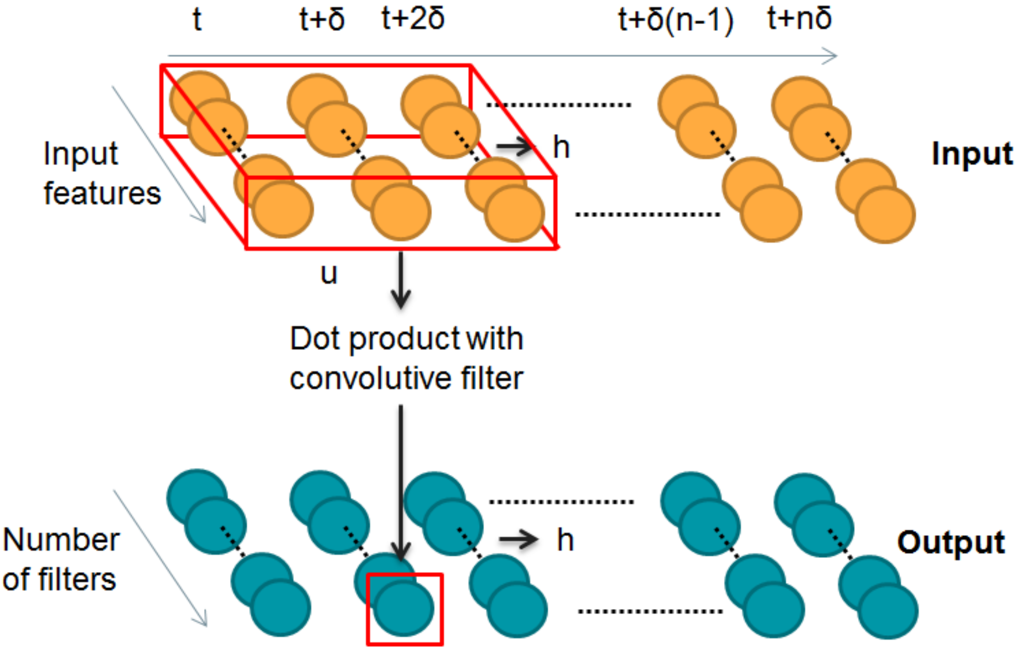
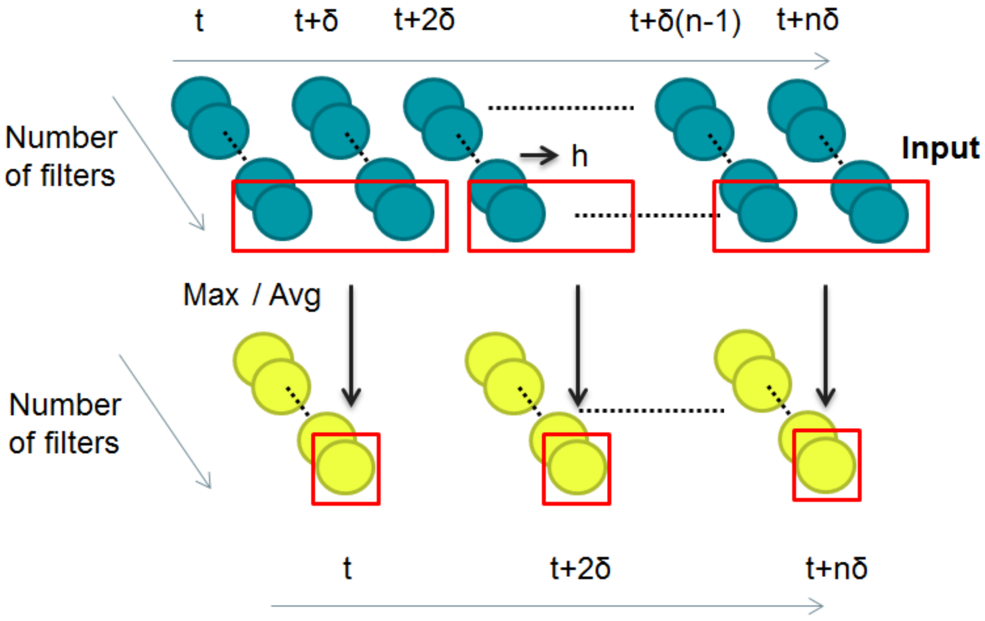


Table 3.2: Implementation detail of the CNN auto-encoder for spectrogram reconstruction.

Layer	Hidden dimension (or # of filters)	Kernel size	Activation function	Dropout
Convolution	32	(3, 5)	Relu	5%
Max Pooling	-	(2, 2)	Linear	0%
Convolution	64	(3, 5)	Relu	5%
Max Pooling	-	(2, 2)	Linear	0%
Convolution	128	(3, 5)	Relu	5%
Max Pooling	-	(2, 2)	Linear	0%
Convolution	256	(3, 5)	Relu	5%
Max Pooling	-	(2, 2)	Linear	0%
Convolution	256	(3, 5)	Relu	5%
Flatten	-	-	Linear	0%
Dense	512	-	Relu	0%
Dense	49152	-	Relu	0%
Reshape	(256, 3, 64)	-	Relu	0%
Convolution	256	(3, 5)	Relu	5%
Upsampling	-	(2, 2)	Linear	0%
Convolution	128	(3, 5)	Relu	5%
Upsampling	-	(2, 2)	Linear	0%
Convolution	64	(3, 5)	Relu	5%
Upsampling	-	(2, 2)	Linear	0%
Convolution	32	(3, 5)	Relu	5%
Upsampling	-	(2, 2)	Linear	0%
Convolution	1	(3, 5)	Relu	5%

Figure 3.10: Pooling step in a CNN.



3.2.3 Results

The method did not converge for the proposed architecture: the training dataset variability was too high for the model to detect significant patterns to reconstruct the scaleogram.

Two approaches are proposed in order to overcome this problem: on one hand, increasing the network complexity (i.e. amount of layers and number of filters per layer) would allow the model to learn more complex and intricate relationships; on the other, the down-sampling applied to the scaleogram matrices eliminates local frequency patterns that might be of interest, therefore less down-sampled transformed signals might be easier to reproduce. For that, a greater computational power would be required.

As a final observation, nonetheless, it must be noted that, as in the previous case, there is no guarantee that the anomalies detected by the reconstruction would be correlated to a system fault.

3.3 Feature Extraction for Fault Prediction and Characterisation

3.3.1 Case Study: Degradation Analysis in ATA 24

The ATA 24 chapter contains all commercial A/C subsystems related to electrical power including alternate and direct current generation and load distribution among other functions.

In the case of the A380, this group contains the variable frequency generator (VFG), an electromechanical component in charge of harvesting mechanical power directly from the turbines in order to generate electrical power. There are a total of four VFGs in the A380, one for each power plant.

Certain VFG components suffer slow degradations leading to failures (with a time-span in the order of up to 10 weeks), and such degradations can be observed through certain signals in the SAR files.

Returning to the idea of generating significant features through Deep Auto-Encoders, the reason that made it impossible to apply such an approach was the skewed training dataset, were anomalies are too scarce to be learnt by the model. Nonetheless, in this case the opposite is true: since the degradation is slow, there is a considerable amount of flights presenting a different dynamic before and after the fault (and corresponding component replacement), allowing for the construction of a balanced dataset.

Additionally, some faults are originated from different degradation processes. In this work the possibility of identifying the underlying fault causes through the latent space will be explored, as well as the possibility of using the extracted information to construct a health indicator based on a classification model.

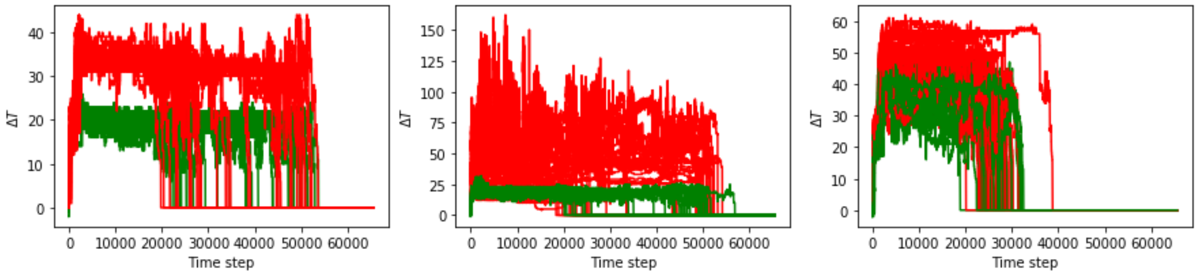
According to the advice of system experts, the analysed signal corresponds to the temperature differences between the Non-Drive End (NDE) of the pump and the oil (monovariate signal).

Three planes with well-identified faults were used to elaborate the dataset: two of them had a mechanical degradation issue; the other one had a malfunction related to the embedded computer collecting the sensor data. Two months of flights were labelled as pre and post replacement. A visual comparison of the different planes is shown in figure 3.11.

3.3.2 Model definition

As commented in section 2.3.1, RNN-inspired networks can not reduce the input dimensionality time-wise, as they output one value for each time step in the input. Therefore,

Figure 3.11: For each plane, flights before component replacement in red, after replacement in green. Leftmost and rightmost planes present mechanical degradation in the VFG, while the one in the centre suffers from embedded computer malfunction. The plane in the right is left for model validation.



the two candidate architectures to consider are Seq2seq AE and CNN AE.

Sequence to sequence AEs, while having an ideal behaviour for the task in hand since they map time series of arbitrary length into a single vector, have been dismissed as they are incapable to reconstruct the input for time-series with lengths of the order of those in this use case, given the available computational power.

Convolutional neural networks, on the other hand, are computationally less intensive than recurrent networks and, with minor reinterpretations as those presented in 2.3.3, allow for processing time series just as if they were images, therefore being proposed as the method of choice for the task in hand.

Their main drawback is their inflexibility regarding sequence length, forcing the use of zero padding in order to achieve an uniform time-series longitude, which contributes to a less precise reconstruction. This fact, while undesirable, is not necessarily important as, in this case, the goal is to generate a representative latent space rather than achieving a precise reconstruction.

As shown in figure 3.12, the main idea behind the proposed architecture is roughly the same as with the previous study, with the difference that, in the current case, the hidden layer will be used directly for classification purposes. The detailed architecture is shown in table 3.3.

In this case, rectified linear units were chosen as activation functions, which allowed the data to be unconstrained; therefore, in this case data was normalized through standardization as shown in section 2.2.2. 20% of the flights were left for validation, and the training results are shown in figure 3.13.

Table 3.3: Implementation detail of the CNN auto-encoder for VFG signal encoding.

Layer	Hidden dimension (or # of filters)	Kernel size	Activation function	Dropout
Convolution	32	(1, 5)	Relu	0%
Max Pooling	-	(1, 4)	Linear	0%
Convolution	64	(1, 5)	Relu	5%
Max Pooling	-	(1, 4)	Linear	0%
Convolution	64	(1, 5)	Relu	5%
Max Pooling	-	(1, 4)	Linear	0%
Convolution	128	(1, 5)	Relu	5%
Max Pooling	-	(1, 2)	Linear	0%
Convolution	128	(1, 5)	Relu	5%
Max Pooling	-	(1, 2)	Linear	0%
Convolution	256	(1, 5)	Relu	5%
Max Pooling	-	(1, 2)	Linear	0%
Convolution	256	(1, 5)	Relu	5%
Flatten	-	-	Linear	0%
Dense	512	-	Relu	0%
Dense	16384	-	Relu	0%
Reshape	(256, 1, 64)	-	Linear	0%
Convolution	256	(1, 5)	Relu	5%
Upsampling	-	(1, 2)	Linear	0%
Convolution	128	(1, 5)	Relu	5%
Upsampling	-	(1, 2)	Linear	0%
Convolution	128	(1, 5)	Relu	5%
Upsampling	-	(1, 2)	Linear	0%
Convolution	64	(1, 5)	Relu	5%
Upsampling	-	(1, 4)	Linear	0%
Convolution	64	(1, 5)	Relu	5%
Upsampling	-	(1, 4)	Linear	0%
Convolution	32	(1, 5)	Relu	5%
Upsampling	-	(1, 4)	Linear	0%
Convolution	1	(1, 5)	Linear	0%

Figure 3.12: For the current case, the interest in generating a feature space is to obtain a low-dimensional representation of the data and use the new representations to train a classifier on these features.

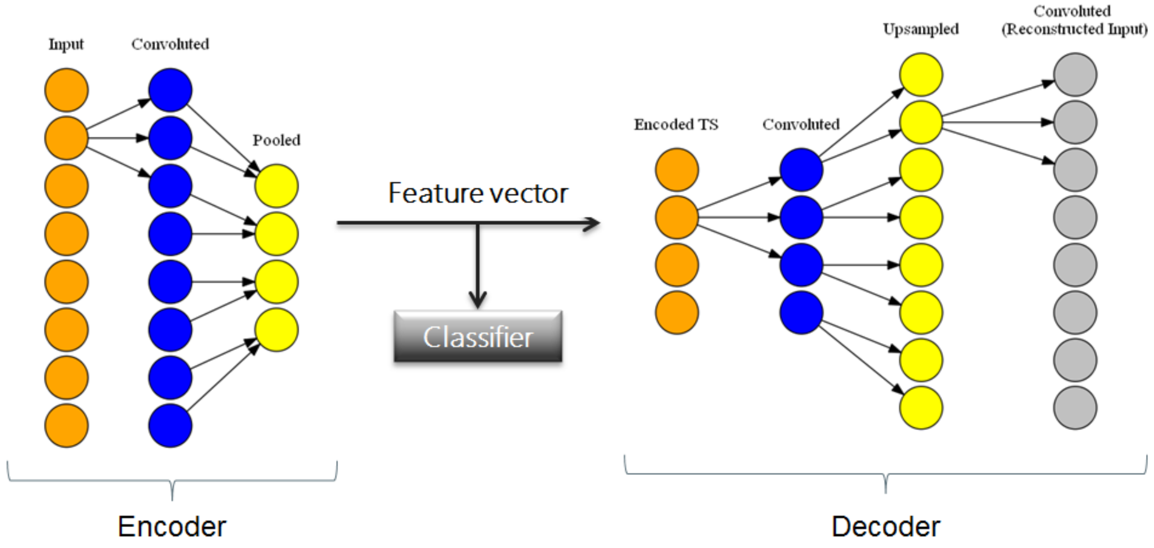


Figure 3.13: Training loss for the CNN auto-encoder. Training was interrupted before overfitting became apparent. The epoch that minimizes the error is the 40th.

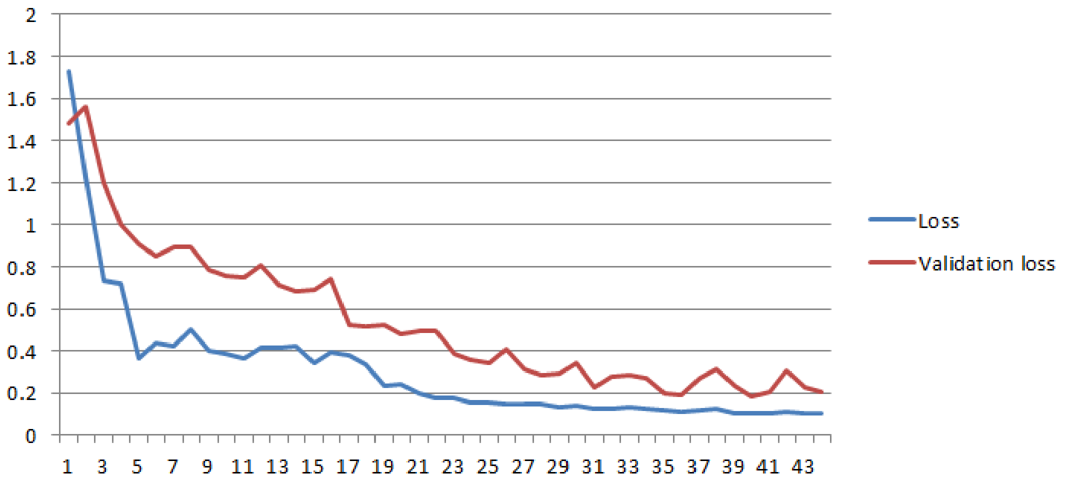
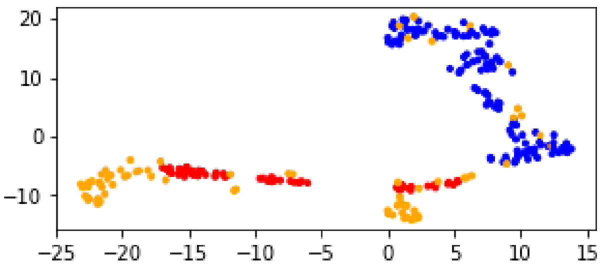


Figure 3.14: t-SNE plot for the flights in the AE latent space. Normal flights in blue, flights with mechanical degradation in red and embedded computer degradation in yellow.



3.3.3 Results

In order to see whether the latent features did effectively separate the different anomalous cases and the normal ones, the generated latent space was observed in search of structures that reflect the variance among the different flight classes.

A t-SNE analysis is shown in figure 3.14. A very apparent separation between anomalous and normal flights can be seen, as well as between both fault types. Furthermore, the three principal components of the flight features (see figure 3.15) show very clear clusters for each flight category.

Given these results, an ADABoost model is trained to classify the different types of flight, which is validated on the remaining flight, which suffers of a mechanical fault. In figure 3.16 the results of the test on the plane left for validation are shown: flights before piece replacement are correctly classified; nonetheless, after the repair the classifier labels them as suffering of a mechanical fault. This is understandable given that, due to inherent factors to this plane that can not be identified, its average temperature for normal operation is well above the normality (40°C instead of 20°C) as seen in figure 3.11. Therefore, even in normal conditions, on a visual analysis such flights would appear anomalous.

As expected, due to the lack of a sufficiently large and representative training dataset it is impossible to train a deep learning algorithm capable of generalizing for all cases. Nonetheless, with this analysis it has been shown that the feature space learnt by auto-encoders can in certain cases identify relevant signal characteristics given a balanced dataset.

Figure 3.15: Plots for the three main components of the PCA transformation. Normal flights in blue, flights with mechanical degradation in red and embedded computer degradation in yellow.

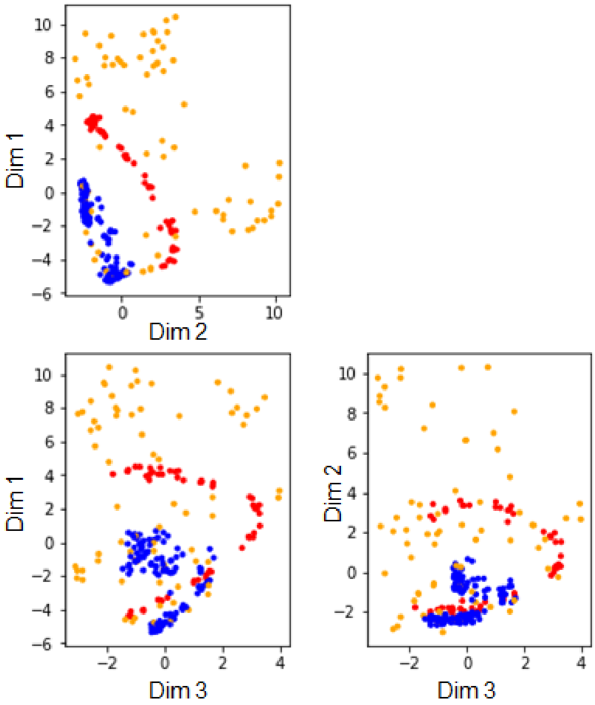
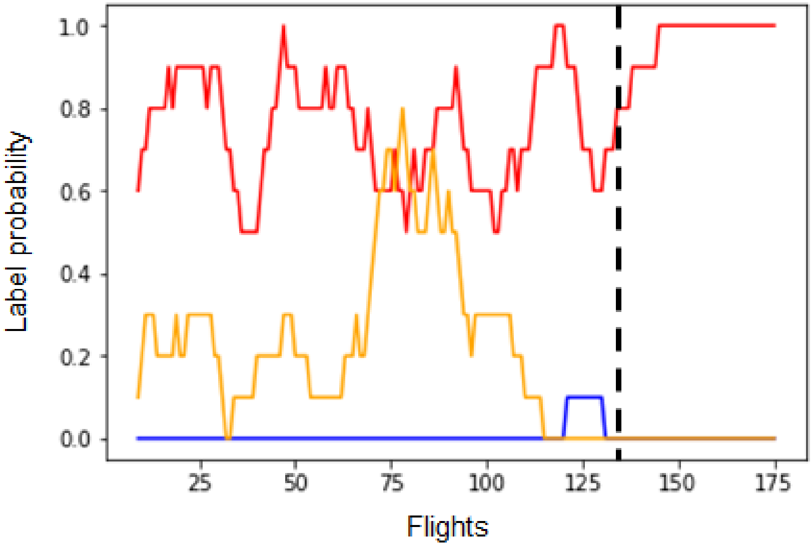


Figure 3.16: Probabilities calculated by the model. The vertical bar corresponds to the component replacement date for this A/C. Normal flights in blue, flights with mechanical degradation in red and embedded computer degradation in yellow.



Chapter 4

Conclusion

During the last years there has been a growing interest in the industry and the academic community regarding the application of deep learning models for anomaly and fault detection in time-series. The goal of this work has been to study their viability and efficacy on commercial aircraft predictive health monitoring given the particularities of flight sensor data and the needs of the 0-AOG plateau, explained in chapter 1.

A survey on auto-encoder architectures for time series has been presented in chapter 2, where three different approaches have been proposed for flight data analysis:

- Feature extraction for anomaly isolation
- Time series reconstruction for anomaly detection
- Feature extraction for fault characterisation

of which only the two last ones have been considered viable and tested in this work for the reasons explained in the same section. Also, some relevant limitations that have had an impact of this work's findings and scope have been pointed out, namely the scarcity of faulty instances, the high variability and volume of each flight's data and the restricted computational capacity available.

In chapter 3, different deep auto-encoder architectures have been tested on A/C data for fault prediction. In the first case, a stacked LSTM AE has been trained on ATA 28 sensor data and used for anomaly detection through signal reconstruction. The results have not show any relationship between the reconstruction error value and the imminence of the component's malfunction.

In the second case, the same subsystem was analysed with a focus on the signal's frequency behaviour: a CNN auto-encoder was trained on scaleogram images in order to be used for reconstruction error analysis. In this case the model training did not converge due to the high variability of the data.

Finally, a CNN auto-encoder was used on series data in order to generate a low-dimension latent space for flight data of the ATA 24, where many anomalous instances were available before the component replacement, which along the normal flights after replacement formed balanced datasets that allowed for a supervised approach. Even though the few available cases did not allow for generalization, the auto-encoder generated embedding spaces where different types of anomalies formed clusters in a non-supervised way.

Regarding the viability of deep learning on commercial A/C predictive maintenance, during this work it has become evident that the benefits that come from the learning capability of this family of methods are lost if a balanced, varied and representative training dataset is not provided. This is the case with aircraft data, where faults for a given system are scarce, systems are complex and tightly intertwined and a great amount of variability comes from environmental factors and ageing.

Furthermore, even if unsupervised approaches can learn the most relevant variation factors of the dataset and detect anomalies through reconstruction error, there is no way to assure that the detected signature is of interest for fault detection, or in case it exists, that it is not masked by a more prominent factor of variation.

Bibliography

[1] Majid alDosari. Unsupervised anomaly detection in sequences using long short term memory recurrent neural. 2016.

[2] Shaojie Bai, J. Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018.

[3] Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[4] Nitish Shirish Keskar et al. On large-batch training for deep learning: Generalization gap and sharp minima. *2017 International Conference on Learning Representations (ICLR)*, 2017. preprint, `http://arxiv.org/pdf/1609.04836.pdf`.

[5] Otto Fabius, Joost Amersfoort, and Diederik Kingma. Variational recurrent auto-encoders. *CoRR*, abs/1412.6581, 2014.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[7] Ian Goodfellow and Oriol Vinyals. Qualitatively characterizing neural network optimization problems. *CoRR*, abs/1412.6544, 2014.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[9] Rafal Józefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *ICML*, 2015.

[10] Diederik Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[11] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, 1995.

[12] Yang Li, Quan Pan, Suhang Wang, Haiyun Peng, Tao Yang, and Erik Cambria. Disentangled variational auto-encoder for semi-supervised learning. *CoRR*, abs/1709.05047, 2017.

[13] Pankaj Malhotra, Vishnu Tv, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder. 08 2016.

[14] Anvardh Nanduri and Lance Sherry. Anomaly detection in aircraft data using recurrent neural networks (rnn). *2016 Integrated Communications Navigation and Surveillance (ICNS)*, pages 5C2–1–5C2–8, 2016.

[15] Kishore K. Reddy, Soumalya Sarkar, Vivek Venugopalan, and Michael Giering. Anomaly detection and fault disambiguation in large flight data: A multi-modal deep auto-encoder approach. *Annual Conference of the Prognostics and Health Management Society 2016*, 2016.

[16] Ilya Sutskever, Oriol Vinyals, and Quoc Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.