Data-driven predictive maintenance and time-series applications

Kefalas, M.

Chapter 2

Preliminaries

This chapter serves as a brief introduction to the notions of time-series and artificial intelligence. We should note here that this chapter does by no means offer an exhaustive overview of the aforementioned fields, as this would lie out of the scope of the present thesis. Its aim, instead, is to acclimatize the reader to the definitions of terminology that will be recurring in the following chapters and to demonstrate the significance and purpose of the said fields.
We will start with the concept of time-series.

2.1 Time-Series

Oddly enough time-series, as the name suggests, are not series in the strict mathematical sense (i.e., summation of infinitely many quantities), but instead are a sequence of numbers that are indexed usually through time (although other indices are possible as well). These numbers can represent almost anything and the indices are, most commonly, equally spaced points in time. The purpose of time-series and their analysis is to understand or model the stochastic mechanisms of various phenomena that take place in an interval of time and predict or forecast future values.
There are ample examples of time-series that we deal with (almost) daily. A weather forecast for the next 10 days or hours, the monthly national unemployment figures and even the steps we take daily and which are recorded on our wearables (e.g., smart watches) are examples of time-series. Perhaps the reader might find more interesting that the number of daily Covid-19 cases (caused by the SARS-CoV-2 virus) since January 2020, the increasing numbers of the global temperature and the sea level in the last 100 years, and the earth's population since the initial recordings are all examples of time-series.
Time-series, of course, need not only measure a single quantity such as the temperature in the next 10 days, but also *multiple* quantities, such as the humidity, the UV index, and the wind speed during that time. When time-series hold multiple values per time-index, then we talk

about multivariate time-series. This is in contrast to univariate time-series which represent only a single quantity per time-index.
More formally a time-series T is defined as follows:

Definition 2.1 (Time-series)**.** Let T be a time-series. Then $T = \{\mathbf{x_i} \in \mathbb{R}^m : i \in \mathbb{N}\}$, where $m \in \mathbb{N}$. When $m = 1$ we refer to T as a univariate or simple time-series and when $m > 1$ we will be talking about multivariate time-series. The values of the time-series T are also called data points, instances or simply data and the time-index i, accompanying a value, is also referred to as a time-step. Finally, the variables measured by a time-series T are also most commonly referred to as features or attributes.

We should note here that in Definition 2.1 we did not bound the time-index i, but instead we let it tend to infinity. Mathematically this is sound. However, in practical applications, such as the ones presented in this thesis, whenever we refer to a time-series T it will be of *finite* length $n \in \mathbb{N}$, unless stated otherwise. One can see the usefulness of this constraint when taking into account that an "infinite" time-series would also require infinite memory in a computing machine. What is more, even though $\mathbf{x_i}$ can lie in any set of numbers, such as the complex numbers $\mathbb{C}^m$, in this thesis we will not be dealing with time-series data in that domain, but we will remain in the set of real numbers $\mathbb{R}^m$. Lastly, the interpretation of the time-index as time is in, general, unimportant from a mathematical perspective, and as such the index can lie in any set of numbers. The applications, though, that we will deal with demand a natural interpretation of the time-index as time and as such, will restrain the time-index in the set of natural numbers $\mathbb{N}$.
We will, furthermore, often talk about a part or a subsequence of a time-series T.

Definition 2.2 (Subsequence)**.** A subsequence $T_{i,k}$ of a time-series T of length $n \in \mathbb{N}$ is a *contiguous* subset of values from T of length $k \in \mathbb{N}$ starting at position with index $i \in \mathbb{N}$. $T_{i,k} = \{\mathbf{x_j} \in \mathbb{R}^m : j \in \mathbb{N}, i \leq j \leq i+k-1\}$, where $1 \leq i \leq n-k+1$ and $m \in \mathbb{N}$. A subsequence of size k is also frequently called a window of size k.

The significance of time-series is paramount and consequently the mathematical tools and the theory that have been developed are vast. We refer the interested reader to [46, 207].

2.2 Artificial Intelligence (AI)

Artificial intelligence (AI) is one of the most fresh fields in science and engineering. The name was coined back in 1956 at Dartmouth College with work being done as soon as 1943 from Warren McCulloch and Walter Pitts and after the Second World War [192]. It can, however, be traced back to the 1840s when Augusta Ada King, Countess of Lovelace and the daughter of Lord Byron (Ada Lovelace) conceived the idea of a programmable computer being intelligent, as notes to the works of Babbage and Menabrea [192, 149]. She stated that: "[. . .] the engine [the programmable computer] might compose elaborate and scientific pieces of music of any degree of complexity or extent." [149].

Nowadays AI has found itself intertwined with our daily lives. From search engines (e.g., *Google*), recommendation systems (e.g., *Netflix*) and virtual assistants (e.g., *Apple's Siri*) all the way to agriculture [148, 220, 75], healthcare [117, 186, 150, 122], law [15] and the automotive industry (e.g., the *Tesla* self-driving car), to name a few. But what exactly is AI?

The answer to that question is not an easy one. After all, what is intelligence [136]? Furthermore, defining a particular field or discipline to the satisfaction of all involved parties is an extremely challenging endeavor and lies outside the scope of this thesis. However, Russel and Norvig [192] suggested that AI be defined in terms of its goals and adopted the following definition:

Definition 2.3 (Artificial Intelligence)**.** "[. . .] we adopt the view that intelligence is concerned mainly with rational action. Ideally, an intelligent agent takes the best possible action in a situation. We study the problem of building agents that are intelligent in this sense."

To alleviate the murkiness of the aforementioned, we refer the interested reader to [136, 192, 98] for a more thorough overview of the field. We will now briefly dive into two related concepts from AI that we will encounter in the following chapters, namely machine learning and deep learning.

2.2.1 Machine Learning (ML) and Deep Learning (DL)

Machine learning (ML) as a term was first introduced by Arthur Lee Samuel an American pioneer in the field of computer gaming and artificial intelligence (AI) in his 1959 paper titled "Some Studies in Machine Learning Using the Game of Checkers" [194]. There he defined ML as:

Definition 2.4 (Machine Learning)**.** "Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed."

But what does it mean for a computer to "learn"? Tom Mitchell defined learning in his book [159] as follows:

Definition 2.5 (Learning)**.** "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Essentially, this means, that ML comprises a set of computational methods which use experience to improve performance, by having the ability to acquire their own knowledge through the extraction of patterns from (raw) data [98]. According to Ian Goodfellow, Yoshua Bengio and Aaron Courville, ML is the only viable approach to creating AI systems that can operate in complex, real-world environments [98].

A closely related concept to ML is that of deep learning (DL), which can be considered a particular type of ML [98]. Yann Lecun, Yoshua Bengio and Geoffrey Hinton defined DL as follows [132]:

Definition 2.6 (Deep Learning)**.** "Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction."

As can be seen from the definition, the essence of DL lies in its representational power. It can achieve great power and flexibility by learning to represent the data as a nested hierarchy of concepts defined from simpler concepts [98]. These multiple levels of representation transform the representation at one level into a representation at a higher and slightly more abstract level [132]. The key aspect of DL is that these layers of representations are not designed by humans, but instead they are learnt from data using a general-purpose learning procedure [132]. The adjective "deep" in DL stems exactly from this hierarchy of concepts. If we draw a graph showing how these concepts are related to each other, the graph will be *deep*, with many layers of concepts and abstractions [98].

At this point we should disambiguate between DL and another term that will often appear: artificial neural networks (ANN) or simply neural networks (NN). Historically, the earliest learning algorithms were intended to be computational models of how learning takes place in the brain. As a result of this ANN is one of the names that DL has gone by [98]. Obviously the modern area of DL goes beyond and above this neuroscientific perspective and finds applications in areas were learning is not necessarily neurally inspired. However, most researchers in the field of AI use the term DL to describe very large NN, in which case "large" points to deep. In fact, occasionally the number of layers in an ANN distinguishes an ANN from a DL approach. Even though there is no definite number that qualifies an ANN as being deep, DL and ANN are very deeply intertwined and both terms are often used interchangeably meaning the same thing.

Finally, we would like to end this chapter by defining certain terms in the context of AI that are important to the understanding of this thesis. Namely, supervised learning, regression, classification training data, test data, validation data, features, labels, hyperparameters and hyperparameter optimization.

Definition 2.7 (Supervised Learning [192])**.** Given a training set of N example input-output pairs:

$$(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N) \ ,$$

where each y_j was generated by an unknown function $y = f(x)$, the task of supervised learning is to discover a function h that approximates f (the underlying relationship between x and y).

Definition 2.8 (Classification)**.** Classification is the task in which a learning algorithm assigns a category (class) to each item (input) [161]. In other words, the output y in Definition 2.7 is one of a finite set of values [192]. It is called binary (or Boolean) classification if there are only two possible values [192].

Definition 2.9 (Regression)**.** Regression is the task in which a learning algorithm predicts a (real) value for each item (input). In other words, the output y in Definition 2.7 is a number [192].

Definition 2.10 (Training data)**.** Training data are the examples that are used to train a learning algorithm [161]. It can be related to the experience E of the algorithm (see Definition 2.5). Another name for training data is training samples. In a supervised learning setting the training data have the form as in Definition 2.7.

Definition 2.11 (Test data)**.** Test data are the examples that are used to evaluate the performance of a learning algorithm (see Definition 2.5) [161]. The test data play the role of the proxy to unseen, real-life data. Another name for test data is test samples.

Definition 2.12 (Features)**.** Features are the set of attributes associated to the data [161]. Another name for features is attributes or variables.

Definition 2.13 (Labels)**.** Labels are the values or categories (classes) assigned to the data [161].

Definition 2.14 (Hyperparameters)**.** The hyperparameters of a learning algorithm are the parameters that dictate the behavior of the learning process. These parameters cannot be learnt by the algorithm from its experience E (see Definition 2.5), but need to be set by the researcher.

Definition 2.15 (Hyperparameter Optimization)**.** Hyperparameter optimization (HPO) is the task of determining the optimal (with respect to some measure) hyperparameters of a learning algorithm (see Definition 2.14).

Definition 2.16 (Validation data)**.** Validation data are the examples that are used to tune the hyperparameters of a learning algorithm in the process of hyperparameter optimization (see Definition 2.15), when working with labeled data [161]. Validation data play the role of a proxy to the test data. Another name for validation data is validation samples.

Having defined all the tools we need we are ready to move on to the main subject of this thesis: predictive maintenance.