ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Deep Learning Model for Context-Dependent Survival Analysis

Raphaël Langhendries[1,2] and Jérôme Lacaille[2]

1- SAMM - EA 4543
Université Panthéon Sorbonne - France

2- Safran Aircraft Engines - DataLab
Villaroche - France

Abstract. In this article, we introduce a deep learning model (denoted thereafter *DCM: Deep Contextual Model*) for survival analysis able of predicting the probability that a subject meets an *event of interest* according to its past life. The subject and the *event of interest* can be diverse depending on the field of application, thus the model can be applied in various contexts. We present an application in the aerospace field that consists in forecasting hot corrosion in turbofan.

1 Introduction

Survival analysis is a branch of statistics that aims to predict the time until an *event of interest* occurs. Various kinds of *events of interest* can be considered depending on the field of application. In the context of predictive maintenance, it is often the time to failure (see for example [1] in the aeronautical field).

When it is relevant, many survival analysis methods allow to model the *time before event* according to some explanatory variable X_i describing the subject i. In this paper, we introduce a model that is able to estimate the *time before an event* according to a time series $(X_t)_{t\in\mathbb{N}}$ (which represents *the context*). Indeed, in many fields where survival analysis is applied, data describing the past life of the subject are available.

In the case of predictive maintenance, many complex devices are equipped with several digital sensors which produce data throughout the device life. Although sensors may not be intended for predictive maintenance, they can provide meaningful information on the health status of the devices. Thus, these data can be used to predict the probability of an *event of interest* to occur. Especially if this *event of interest* is correlated with an abnormal operation of the device.

This paper is organized as follows. Section 2 introduces the problem of survival analysis with times series, Section 3 describes the proposed model, Section 4 gives an application in the aeronautical field and section 5 concludes the article.

2 Framework

2.1 Background on survival analysis

Like all statistical models, survival models need to be fitted on data. Survival data can be represented by a tuple (t, c) where t is a time and c a label. Survival data can be

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

- **Uncensored**. In this case, the *event of interest* occurs exactly at the instant t and we denote $c = 0$.
- **Right censored**. At the instant t, the *event of interest* did not yet occur (but it may in the future). We denote $c = -1$.
- **Left censored**. At the instant t, the *event of interest* already occurred. But we didn't know when. We denote $c = 1$.

The main goal is to approach the *survival function* (denoted thereafter $S(t)$) which gives the probability that a subject did not meet the event before t. Alternatively, we can approach the function $W(t) = 1 - S(t)$ that gives the probability that a subject meets the event before t.

2.2 Models for survival analysis

A wide range of methods have been developed, it includes non-parametric methods such as the Kaplan Meier estimator, semi-parametric methods such as Cox model, and parametric methods. For example, a parametric model widely used for materials degradation is the Weibull distribution (see [2]). Information and references concerning standard models in survival analysis can be found in [3].

Models based on machine learning have also been proposed, (see the survey [4] for an overview). In particular, deep learning models have also been proposed, they are often associated with the previously existing parametric model (see for example [5] and the model *deep survival* which extends Cox model). Models relying on recurrent networks (such as [6]) have shown promising results. Our model: *DCM* (*Deep Contextual Model*) falls into this category but differs from previous works on several points. In particular, *DCM* exploits a time series describing the past life of the subject (its *context*).

2.3 Survival analysis with time series

Usual survival machine learning models take advantage of vectors X_i describing the subject number i. It allows fitting one survival function $S_i(t)$ by subject.
In this article, we assume to have a time series $(X_t^i)_{t\in[1,n_i]}$ by subject i. Each data X_t^i is a partial description of the state of the subject i at time t. Thus, we try to estimate a survival function $S_i(n)$ that depends on the past life (the *context*) of this subject (which is the time series $(X_t^i)_{t\in[1,n]}$). Consequently, this approach is relevant only if we can assume that the *event of interest* is correlated to the past life of the subject.

We emphasize that an observation X_t^i does not necessarily contain direct information on the *event of interest*. Especially, we may not know if the event occurred before time t (it is the case if the *event of interest* requires specific inspections that can't be frequently done).

Under our framework, the subject i is described by the tuple $(t, c, (X_t^i)_{t\in[1,n_i]})$ where t is a time, c informs about the censorship and $(X_t^i)_{t\in[1,n_i]})$ is the time series describing the subject i.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

3 Model

3.1 Overview

For one tuple $(t, c, (X_t^i)_{t\in[1,n_i]})$, our model aims to compute the probability that the *event of interest* occurs before t. The computation is represented in the Figure below.

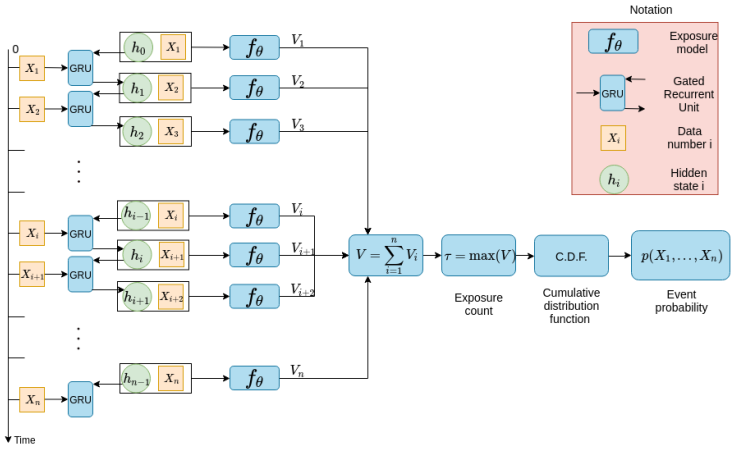

Fig. 1: Computation of the *event probability* by *DCM* (i.e probability of the event occurs before t).

DCM uses vectors X_t as inputs for the GRU block (Gated Recurrent Unit, see [7]) which is a kind of recurrent neural network related to LSTM (Long Short Term Memory). The GRU block computes a *hidden state* vector h_t which represents the health status of the subject at time t. Vector X_t and the hidden state vector h_{t-1} are concatenated. The *exposure model* (f_θ) computes damages inflicted to the subject at time t (the vector of the damage features V_t). All damage vectors V_t are summed, we keep the most critical feature $\tau = \max(V)$. Finally, we use the *total damage* τ and a cumulative distribution function to compute *the event probability*.

3.2 Exposure model and cumulative distribution function

The *exposure model* (denoted f_θ in Figure 1) computes damages inflicted at time t. Lots of choices are possible according to the structure of data X_t, in our application f_θ is a dense neural network.

The cumulative distribution must be chosen according to the problem. To this end, previously existing parametric models can be very helpful. Indeed, we can select a parametric model known to work well in similar problems. With this logic, we select the Weibull law [2] with its two interpretable parameters (see Section 4).

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

3.3 Learning

Parameters appearing in our model must be optimized. It includes parameters from the *GRU block*, parameters from the *exposition model*, and parameters from the *cumulative distribution function*. They are all learned at the same time by gradient descent. The gradient is computed from a loss function.

The likelihood is a common choice to measure the goodness of fit of a model. Our objective is to converge towards parameters maximizing the likelihood. The likelihood for samples $i \in [1, n]$ is

$$\mathbb{L}\left(\left((t, c, (X_t^i)_{t\in[1,n_i]}), S_i\right)_{i\in[1,n]}\right) = \prod_{i=1}^{n} G_i(t),$$
$$\text{with } G_i(t) = \begin{cases} \lambda_i(t) & \text{if } c = 0, \\ 1 - S_i(t)) & \text{if } c = -1, \\ S_i(t) & \text{if } c = 1. \end{cases}$$

Taking the negative log, we get

$$-\ln\left(\mathbb{L}(\left((t, c, (X_t^i)_{t\in[1,n_i]}), S_i\right)_{i\in[1,n]})\right) = -\sum_{i=1}^{n} \ln(G_i(t)).$$

Maximizing likelihood is equivalent to minimizing negative log-likelihood. It leads to the following loss function for one sample.

$$L((t, c, (X_t^i)_{t\in[1,n_i]}), \hat{p}) = \begin{cases} -\ln(\hat{p}) & \text{if } c \leq 0, \\ -\ln(1 - \hat{p}) & \text{if } c = 1, \end{cases} \tag{1}$$

where $\hat{p}$ is the *event probability* computed by the model. Indeed, data are often all censured moreover $G_i(t) = \hat{p}$ if $c = -1$ and $G_i(t) = 1 - \hat{p}$ if $c = 1$.
This loss function corresponds to the *binary cross-entropy loss*.

4 Application

4.1 Health monitoring for aircraft engine

Ensuring the correct functioning of aircraft engines is a critical task and a strict maintenance policy exists for each important component. Regular inspections of the engine are scheduled to verify each component's integrity. When an engine is inspected, a component can be either changed or not. As a precaution, maintenance policies tend to be pessimistic and flag components to be replaced even if far from failing. In this application, the *event of interest* is the moment when one specific component reaches the limit indicated by its maintenance policy.

A commercial flight is classically divided into phases: TakeOff, Cruise etc. At the beginning of each phase, sensors equipping the aircraft register data such as altitude, engine thrust etc. Furthermore, we add environmental data, such as

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

outdoor temperature and air concentration of certain pollutants at airport.
In this way, vectors X_t^i defined in Section 2.3 designed the aggregated data of the flight number t for the engine number i and the time series $(X_t^i)_{t\in[1,n_i]}$ stands for the n_i flights achieved by the engine i.

4.2 Forecasting hot corrosion

Hot corrosion is a physical phenomenon that can cause damages to different metals exposed at high temperatures [8]. Predicting hot corrosion in turbofan is a challenging issue because it involves several different physical phenomena (corrosion of type 1, 2 and oxidation are confounded). Moreover, the kinetic of these reactions depends on lots of factors: temperature, pressure, catalyst, etc.

We focus on one type of component of the aircraft engine. This component is checked when the engine is inspected (regular inspections are scheduled over the engine lifetime). If the component needs to be replaced because of hot corrosion, we will consider that our *event of interest* already occurred, thus we get a left-censored data ($c = -1$). Otherwise, it is a right-censored data ($c = 1$).

DCM is implemented in PyTorch. 32 engines are considered, the *exposure model* f_θ is a 3 layers dense neural network with 90 neurons by layer (10 for the GRU). We add a dropout layer with $p = 0.4$ and a L2 penalty to reduce over-fitting ($w\_decay = 10^{-3}$). The learning rate is equal to 10^{-3}.

We evaluate the model using a 10-folds cross-validation procedure. Each engine is used only once in the validation set. We aggregate all results obtained on validation sets and present them in the form of a ROC curve (see [9]).

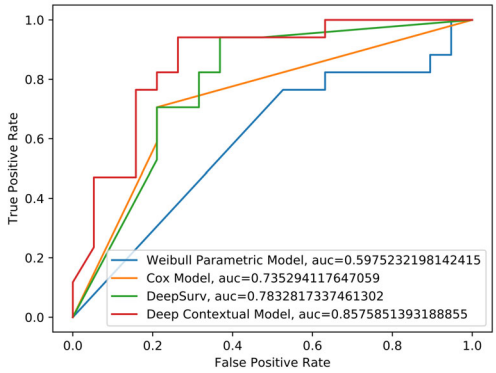

Fig. 2: ROC curves of several models

DCM is compared to several others implemented using PySurvival [10]. It includes the Cox model, the Weibull parametric model and DeepSurv (see [5]). We used the package tsfresh (see [11]) to extract a vector of features Y_i from each times series $(X_t^i)_{t\in[1,n_i]}$ describing flights achieved by the engine i. Then, vectors Y_i are used by Cox model and DeepSurv.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

It appears in Figure 2 that our model (DCM) performs better than baselines (Cox model and Weibull parametric model) and a state-of-the-art alternative solution (tsfresh features extraction + DeepSurv).

5 Conclusion

In this article, we introduced a new model for survival analysis. Its originality lies in the use of a time series describing the past life of the subject to get more relevant prediction.

In practice, the model relies on a recurrent neural network (GRU) to compute a hidden state that represents the health status of the subject at any time. Then it computes and accumulates damages inflicted to the subject at each time. Eventually, accumulated damages are used as inputs for a cumulative distribution function (that can be parametric) to compute the probability that the *event of interest* already occurred.

We implemented the model and tested it with real data. We were interested in computing the probability that a component of the aircraft engine will be renewed because of hot corrosion.

References

[1] Wim J.C. Verhagen and Lennaert W.M. De Boer. Predictive maintenance for aircraft components using proportional hazard models. *Journal of Industrial Information Integration*, 12:23–30, December 2018.

[2] Waloddi Weibull. A Statistical Distribution Function of Wide Applicability. *Journal of applied mechanics*, 18:7, 1951.

[3] David G. Kleinbaum and Mitchel Klein. *Survival Analysis*. Statistics for Biology and Health. Springer New York, New York, NY, 2012.

[4] Ping Wang, Yan Li, and Chandan K. Reddy. Machine Learning for Survival Analysis: A Survey. *ACM Computing Surveys*, 51(6):1–36, February 2019.

[5] Jared Katzman, Uri Shaham, Jonathan Bates, Alexander Cloninger, Tingting Jiang, and Yuval Kluger. DeepSurv: Personalized Treatment Recommender System Using A Cox Proportional Hazards Deep Neural Network. *BMC Medical Research Methodology*, 18, February 2018.

[6] Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Lin Qiu, and Yong Yu. Deep recurrent survival analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:4798–4805, Jul. 2019.

[7] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078 [cs, stat]*, September 2014. arXiv: 1406.1078.

[8] Fred Pettit. Hot Corrosion of Metals and Alloys. *Oxidation of Metals*, 76(1-2):1–21, August 2011.

[9] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.

[10] Stephane Fotso et al. PySurvival: Open source package for survival analysis modeling, 2019–.

[11] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing*, 307:72–77, September 2018.