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/.

Convolutional Neural Network Architecture for Classification of Aircraft Engines Flight Time Series

Delphine Bay and Clémence Bisot

Safran Aircraft Engines, 2 chemin de Viercy, 77019 Montereau-sur-le-Jard, France

Abstract. During each flight, an aircraft engine sends data to a ground system. This data corresponds to different sensors measurements (temperatures, pressures, vibrations...) collected at key moments of the flight. It constitutes rich multivariate time series used to monitor the engine's health. In this article, we used flight data to predict the main removal cause of the engine. The problem falls within the framework of time series classification. This article proposes an interpretable neural network architecture which fits with the physical understanding of the modeled phenomenon in order to address the problem on a real-world, industrial dataset.

1 Introduction

A shop-visit consists in removing an engine from the aircraft to perform a large-scale maintenance operation. Shop-visits, which happen every 10 to 15 years for each engine, are major events that need to be well understood and anticipated by the engine manufacturers. Every shop-visit event is driven by a main removal cause. Main removal causes include continuous deterioration of any major module of the engine (combustor, compressor, turbine...) or any punctual event like bird ingestion, destroying many parts of the engine at once. Each shop-visit can be associated to its main removal cause using exogenous data provided by the airlines. This process is called shop-visit classification.

Additionally, the aircraft engine sends endogenous data to a ground system at each flight. Typically this data contains temperature, pressure, rotation speed, or vibration measurements, collected by sensors along the engine at different key moments of the flight. These extremely rich multivariate time series are used to monitor the engine's health.

In this article, we evaluated the opportunity to use only endogenous flight data to ease the shop-visit cause classification process using deep learning models. Our problem is thus the classification of multivariate time series.

We are tackling a hard problem. To illustrate some of the specific difficulties related to the processing of real-world flight data, one can refer to the work of [1]. First of all, no human would be able to do this task easily, even with a lot of expertise on the engine: as mentioned, the removal cause classification is usually achieved using external data. Secondly, our time series are extremely noisy. A large part of the variance is due to changes in flight conditions, yet another is explained by evolution of the engine's health state. This last part of the variance is the one we are interested in catching.

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/.

2 Related work

In the literature the Time Series Classification (TSC) problem is mostly driven by Human Activity Recognition [2] applications: the purpose is to recognise the types of movements that a human has made based on sensor data.

Historically, signal processing methods were used to manually extract relevant features from the time series. However, those feature engineering methods require a lot of expertise and *a priori* understanding of what relevant features are. In our case, since flight data are not usually used for the task of shop-visit classification, we lack this human inferring expertise. This is where deep learning becomes a true opportunity.

In their reviews of existing deep learning approaches for end-to-end TSC tasks, [3] and [4] note that CNN architectures are usually favoured, thanks to their robustness and the relatively low training time required. Many variants of CNN are explored: for instance, the Multi-Channel CNN [5] makes convolution operations on each variable of the multivariate series separately, whereas DeepConvLSTM [6] is a hybrid model seeking to combine the advantages of CNN and LSTM by replacing the last dense layers of a classic CNN with recurrent ones.

Recurrent Neural Networks (RNN) such as LSTM [7] are another option when dealing with temporal data. However, RNN are not well suited for long time series: the bigger the time range, the slower and the harder they are to train. Moreover, if the entirety of the time series is relevant for the classification, the "loosing memory" property of RNN is not adapted.

Finally, neural networks are known to work as black boxes, which is a substantial drawback in an industrial context. To address this issue, [8] proposed to take advantage of the mathematical properties of a layer called *Global Average Pooling* (GAP) to build *Class Activation Maps* (CAM), *i.e.* heat maps that highlight which regions in the input data have contributed the most in the classification decision of the network. This methodology was initially thought to work for images, but [9] adapted the principle to temporal data. In our final architecture, we took advantage of these layers to interpret our results (see Section 4).

3 Data description and methodology

3.1 Data description

In our raw data set, we have 1367 classified shop-visits. A shop-visit can happen at any time during the life of an engine. There are 6 possible shop-visit causes (*i.e* 6 classes). Their distribution, detailed in Table 1, shows that our data is heavily unbalanced. Section 3.3 gives details on how this issue was handled.

For each sample, we have a multivariate time series of 500 time stamps, corresponding to the last 500 flights with available data before engine removal.

At each time step, we have a measurement over 10 sensors (temperatures, pressures, vibrations...) at two different flight phases: one measure at Takeoff and one at Cruise. However, Takeoff and Cruise data should not be melted too

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/.

Cause	Mod. 1	Mod. 2	Mod. 3	Mod. 4	Perf.	Ext. event
Nb of samples	721	147	140	131	129	99

Table 1: Class distribution. Mod. stands for module, an ensemble of parts; Perf. for performance, a global performance degradation of the engine; Ext. event, an external one-off event damaging the engine.

lightly, since it would not make sense physically. We are therefore looking for a network architecture aligned with this internal structure of the data. The 10×2 sensors data is completed with one last variable containing the age of the engine at each flight. Hence we have 1367 time series in a 21 dimensional space, each one labelled by the terminal event. An example of one sensor time series can been seen on Figure 3 in the CAM interpretation section.

All variables are centered and standardised. The missing data (circa 17 %) are imputed with a linear interpolation. No other pre-processing step was needed. In particular, we have glitches and outliers in the time series (see Figure 3 for an example of glitch). We rely on the network to learn how to properly handle those anomalies.

3.2 Summary of the CNN architecture

Our global methodology is presented on Figure 1 and can be summarised as follows.

First of all, we train one network per cause. We have thus $n = 6$ networks with all exactly the same architecture: each network is specialised in the recognition of one cause i against the $n - 1$ others (this architecture is later referred to as "6x Conv2D" architecture). Note that our first attempt was to train a single network directly deciding the removal cause (referred to as "1x Conv2D" architecture). This, however, gave unsatisfying results (worse than random: see Table 3 for details). Having one network per cause also eases the interpretation of the results (see Section 4).

Afterwards, we use the outputs of the n networks, *i.e.* the probabilities estimated by each network for their given removal cause, to vote for the most likely one. We tested a highest probability voting method ("6x Conv2D & Highest Probability"). The most successful strategy was however use the n outputs to train a dedicated Support Vector Machine ("6x Conv2D & SVM" method).

As for the precise architecture of each cause-specialised network: encouraged by our literature review in Section 2, we build CNNs. We make convolution operations in 2D, so as to allow convolutions along the "sensors" axis (since we have a lot of correlation between variables). We also force the convolution kernel to cover all of the parameters axis, so as to allow convolutions between all sensors. The use of padding enables to preserve information, in particular as we get close to the end of the time series, *i.e.* the engine removal.

In order to analyse Takeoff and Cruise data separately, the networks have two parallel branches: each is specialised in the creation of features for its own flight

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/.

phase. Each branch has the same architecture, detailed on Figure 2. They are composed by two steps of {convolution, batch normalisation, ReLu activation}, followed by a Global Averaging Pooling layer to allow the generation of CAMs for interpretability. Then, a dense layer in each branch makes a classification decision. At the end, the results from the two branches are concatenated and a last dense layer makes the final decision.

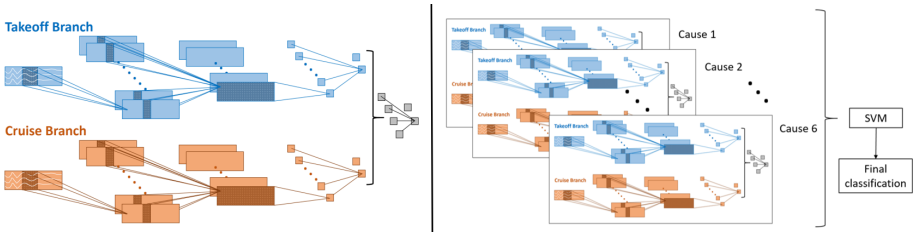


Fig. 1: Global classification algorithm architecture. Left hand part shows the architecture for one cause classification. Right hand part shows the global architecture with one network per cause and a SVM for final classification.

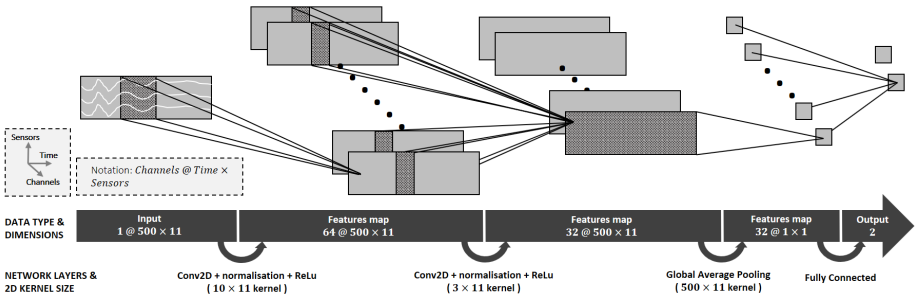


Fig. 2: Detailed CNN architecture for one phase (Takeoff or Cruise) branch. For each layer, one can read the detail of the architecture.

3.3 Training strategy

As presented in Section 3.1, the raw dataset is unbalanced. However, we chose to use a balanced testing set with 20 samples per class (120 samples). The remaining sample (about 90 % of the dataset) used for training stays unbalanced, yet, within the learning procedure, we force each batch to be balanced along causes. Each batch is bootstrapped out of the training dataset with the same pickup probability for each cause. This process is repeated 5 times for cross-validation. Here the cross-validation procedure is absolutely necessary to evaluate the precision of our results because our labeled dataset is relatively small with regard to its variability (see Table 3).

The stochastic gradient descend is optimised with an Adam algorithm with a learning rate of $1.e^{-2}$. Each of the 40 epochs consists in a succession of 20 batches.

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/.

4 Results and interpretation

As stated previously, the "6x Conv2D & SVM" methodology turns out to perform the best, with an accuracy of 42.3 %. The confusion matrix obtained is presented in Table 2. A comparison of the different methods is also recorded in Table 3.

		Predictions						
		Mod. 1	Mod. 2	Mod. 3	Mod. 4	Perf.	Ext. ev	Total
Observations	Mod. 1	10.8	1.6	2.2	2.0	2.2	1.2	20
	Mod. 3	2.2	9.0	2.4	2.2	1.6	2.6	20
	Mod. 4	5.6	2.2	4.4	3.6	1.2	3.0	20
	Mod. 3	3.6	0.8	2.2	6.8	1.6	5.0	20
	Perf.	1.6	2.2	1.4	1.2	11.4	2.2	20
	Ext. ev	1.4	3.0	2.4	3.6	1.2	8.4	20
	Total	25.2	18.8	15.0	19.4	19.2	22.4	120

Table 2: Confusion matrix for model "6x Conv2D & SVM" with 5-fold cross-validation. Lines correspond to true classes, columns to predicted ones.

Methodology	Mean acc. (%)	Min-max acc. (%)
Random baseline	16.7	–
1x Conv2D	13.3	8.3–16.7
6x Conv2D & Highest Probability	39.7	35.0–45.0
6x Conv2D & SVM	42.3	38.3–47.5

Table 3: Accuracy comparison on test set (5-fold cross-validation)

We generated CAMs to check if the networks use a somewhat physically understandable decision process. The results are mostly conclusive. Figure 3 gives an example for an engine which was removed to restore its performance:

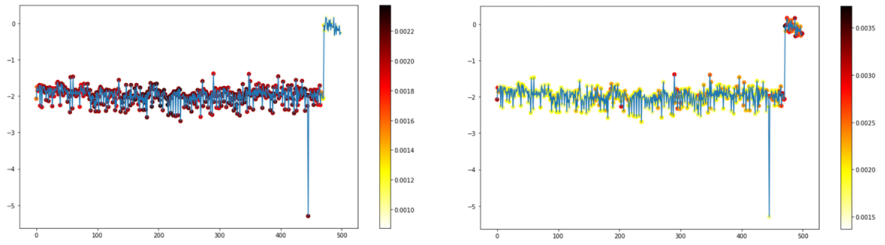
(a) The Perf. model considers the long period of time during which the performance variable has been low, which is exactly what this cause is linked to physically.

(b) The External event model only looks at the sudden parameters shift towards the end of the time period. It is consistent with what external events should correspond to.

5 Conclusion and future works

In this work we tackled a very challenging classification problem with real-world flight data. Few pre-processing steps (standardisation and missing data imputation only) were needed before feeding a well-chosen convolutional neural network to perform a classification task that humans, even with a lot of expertise on the

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/.



(a) As analysed by the Perf. model (b) As analysed by the Ext. event model

Fig. 3: Examples of Class Activation Maps for a key performance variable at Takeoff, for an engine removed for the cause Performance

engine, would not be able to do. The chosen architecture is adapted to our physical understanding of the modeled phenomena, and the resulting network is interpretable and makes sense to domain experts. Our work also corroborates with literature, by confirming that CNN architectures are well adapted to our type of data and quick to train (about 30 minutes).

Future work will rely on this experience to solve other problems with the same data using deep neural networks. The open problems include classification of maintenance efficiency and modeling of the engine deterioration speed.

References

[1] T. Rabenoro, J. Lacaille, M. Cottrell and F. Rossi, Anomaly detection based on aggregation of indicators, *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014.

[2] J. Wang, Y. Chen, S. Hao, X. Peng and L. Hu, Deep learning for sensor-based activity recognition: A Survey, *Pattern Recognition Letters*, 2019.

[3] H.I. Fawaz, G. Forestier, J. Weber, L. Idoumghar and P.-A. Muller, Deep learning for time series classification: A review, *Data Mining and Knowledge Discovery*, 2019.

[4] O. Faust, Y. Hagiwara, T.J. Hong and S. O. Lih, Deep learning for healthcare applications based on physiological signals: A review, *Computer Methods and Programs in Biomedicine*, 2018.

[5] L. Zheng, Q. Liu, E. Chen, Y. Ge and L.J. Zhao, Time series classification using multi-channels deep convolutional neural networks, *International Conference on Web-Age Information Management*, 2014.

[6] F. Ordóñez and D. Roggen, Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition, *Sensors*, 2016.

[7] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural computation*, 1997.

[8] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference of Computer Vision and Pattern Recognition*, 2016.

[9] Z. Wang, W. Yan and T. Oates, Time series classification from scratch with deep neural networks: A strong baseline, *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017.