

PAPER • OPEN ACCESS

Virtual sensors to generate turbine runner blade strains from indirect measurements

To cite this article: Martin Gagnon *et al* 2022 *IOP Conf. Ser.: Earth Environ. Sci.* **1079** 012067

View the [article online](#) for updates and enhancements.

You may also like

- [Analysis and Treatment of the Cracks on the Runner Blade of a Francis Turbine](#)
Yuanlin Long
- [Optimizing runner blade profile of Francis turbine to minimize sediment erosion](#)
B S Thapa, B Thapa, M Eltvik *et al.*
- [Modeling the dynamic behavior of turbine runner blades during transients using indirect measurements](#)
I Diagne, M Gagnon and A Tahan



ECS
The
Electrochemical
Society
Advancing solid state &
electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

Virtual sensors to generate turbine runner blade strains from indirect measurements

Martin Gagnon, Luc Vouligny, Luc Cauchon and Anne-Marie Giroux

Institut de recherche d'Hydro-Québec, Varennes, Québec, Canada

gagnon.martin11@hydroquebec.com

Abstract. Strain measurements on turbine blades are difficult and costly tasks. Such measurements, when carried out, generally only happen during the runner commissioning. This gives rise to two problems. The first is that some of the sensors often stop functioning properly during the measurement campaign, which leads to distorted data, and the second is that runner blade strains are not available for long-term monitoring after the measurement campaign. To alleviate the consequences of distorted or missing values, we propose the use of neural networks to automate the imputations of missing values in measurement campaign data using virtual sensors. Three types of network architecture are proposed: Long Short-Term Memory (LSTM) in different multi-stage/multi-layer configurations in Nonlinear Auto-Regressive Neural Networks with exogenous input (NARXNN), injector multi-scale attention network (Injector MA-Net), and a combined architecture using both. The performance of these architectures will be compared in four situations: the loss of strain gauge rosette branches; the loss of a complete strain gauge rosette; the loss of data on a complete blade; and the absence of strain data, which is related to the problem of identifying which sensors could be used for long-term monitoring. The performance of the proposed algorithms will be evaluated on real case scenarios from a measurement campaign during a recent unit commissioning.

1. Introduction

To assess either the fatigue reliability or remaining life of equipment, we need a damage model and a good knowledge of the stress cycle history, the initial conditions and a limit state [1, 2]. If the goal is structural health monitoring, the stress cycles need to be estimated continuously over the life of the structure. This can be done with different strategies. First, there are physical numerical models, such as finite elements, with the associated difficulties in terms of validation and uncertainty quantification [3, 4]. Next, there is direct measurement [5] during dedicated measurement campaigns, often done during unit commissioning. Then, there is indirect measurement [6], which is more suitable for online monitoring since one of the main limitations of direct measurement relates to the feasibility of permanent instrumentation of runner blades with strain gauges. Currently, strain gauges are laborious to install, have a limited life expectancy and become highly expensive as more detailed spatial information is required. These limitations are easier to deal with during short dedicated measurement campaigns. However, during those campaigns, there is always the possibility of sensor loss and, at the end, the data need to be correlated to the historical data from the online monitoring system to obtain a continuous estimate of strain cycles for structural health assessment.



In order of complexity, the cases encountered are:

- Case 1: Unusable data from a strain gauge rosette branch
- Case 2: Unusable data from a complete strain gauge rosette
- Case 3: Unusable data on a complete blade
- Case 4: Absence of strain data for long-term monitoring

To solve these four cases where missing values need to be imputed, we propose the use of virtual sensors that will be evaluated using measurement data obtained during the commissioning of a runner. The virtual sensors can be used not only for the imputation of missing data but also for the detection of anomalies if the measured values deviate significantly from their estimates. In this study, our goal is to impute the missing values using only the information available in the measurements using artificial neural networks. From the literature, we notice that Presas et al. [7] worked on a similar idea for Francis turbines condition monitoring by estimating features like mean and peak-to-peak stress amplitudes using stationary sensors. However, this is not specific to Francis turbines and required in several structural health monitoring applications. In this study from Gulgec et al. [8], a neural network methodology to estimate strains from acceleration data is proposed for civil structure. In our case, we chose to evaluate two types of neural networks: Long Short-Term Memory (LSTM) in different multi-stage/multi-layer configurations in Nonlinear Auto-Regressive Neural Networks with exogenous input (NARXNN), which we previously used for a similar task with online monitoring data [9, 10]; and injector multi-scale attention network (Injector MA-Net), which has an architecture typically used for images that we adapted for timeseries [11]. First, both types of networks will be evaluated individually. Then, a combined architecture of both will be evaluated to understand the gain that could be obtained with the model complexity increase.

Furthermore, we have the secondary goal of building in-house benchmark study cases to gain a better understanding of the performance of approaches based on neural networks. In time, we believe that such in-house study cases will facilitate the development of publicly available ones for the scientific community.

The paper is structured as follows. First, an overview of the study cases is given, followed by a presentation of the three types of neural network architecture used. Then, the results are presented and discussed.

2. Study cases

The typical setup used in our measurement campaign [5] is presented in Figure 1. Note that multiple acquisition systems are used. This generates synchronization-related issues and may render difficult correlation with the runner blade sensors, as the data may be obtained on independent acquisition systems. The acquisition systems in our study have a common signal that can be used to align the time signals, but each acquisition system still has its own independent clock speed.

The measurement campaign comprises multiple values at different locations on the runner measured simultaneously. Here is an overview of the available data:

- Runner blade strains: 4 rosettes per blade on 2 blades (24 strain signals)
- Shaft strain (torsion and flexion)
- Shaft displacements
- Runner labyrinth gaps
- Pressures (penstock, distributor, draft tube)
- Water levels
- Power, wicket gate opening, rotating speed, etc.

From the available data, we chose to use a limited subset of inputs related to the shaft displacements and the operating conditions. The inputs are: 2 generator guide bearing displacements, 2 turbine guide bearing displacements, 2 labyrinth displacements and the wicket gate opening, along with the runner blade strain gauges. This subset of inputs was chosen as they are readily available in most online monitoring systems, except for the runner blade strains.

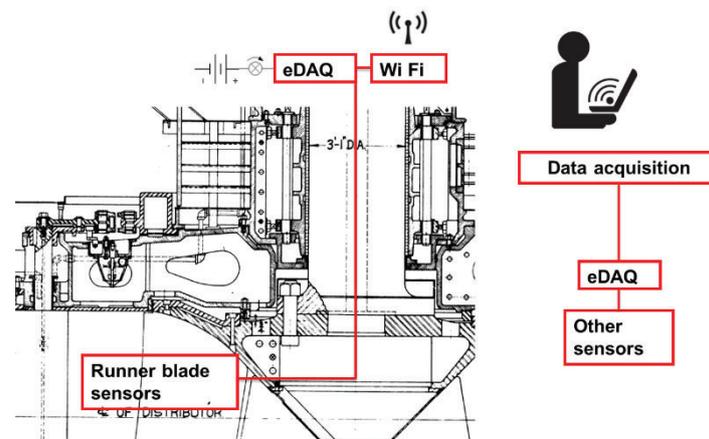


Figure 1. Typical data acquisition setup.

For Cases 1, 2 and 3, the strain signals in the imputation task are highly correlated with the other strain signals on the runner. Notice that with a rosette, the same location is measured in different directions hence the 3 outputs are highly correlated. In Case 4, the choice of the other inputs is critical since they need to be correlated with the runner blade strains to be estimated during the imputation task. Our assumption is that the virtual sensor performance should degrade going from Case 1 to 4, given that the inputs are less correlated with the output as we move from one case to the other.

The study focuses on steady state regimes. The operating conditions subset chosen as a function of maximum power output (P_{max}) is: Speed-no-load (SNL), 39% P_{max} , 73% P_{max} , 85% P_{max} and P_{max} . These operating conditions cover the main types of dynamic behaviors and span the complete range of steady state regimes. However, the subset is intentionally too sparse to properly model the complete operating range. Our goal is to highlight two situations: when the operating condition of interest is available in the algorithm training dataset, and when the operating condition of interest is not available. For the latter, the case when only SNL is available for training represents an extreme case for virtual sensors since this steady state operating condition is usually the first to be measured during a campaign and has a dynamic behavior that is far from representative of the whole operating range.

3. Proposed artificial neural network architectures

We chose three architectures: NARXNN, Injector MA-Net and a combined architecture using both. NARXNN is simpler compared to Injector MA-Net, while the combined architecture is even more complex. These three architectures will help us understand the performance to be expected from each while we explore the possibilities offered by combining them.

3.1. Nonlinear Auto-Regressive Neural Networks with exogenous input (NARXNN)

We have used NARXNN architecture in the past to generate virtual sensors for stator temperature monitoring data and shaft displacements [9]. The architecture from that work was used for the imputation task in this study. Recurrent neural network (RNN), Long Short-Term Memory (LSTM) and Nonlinear Autoregressive Exogenous (NARX) architectures are well suited to estimate time series behavior. Here, we used a combination of NARX and LSTM, inspired by the work of Massoudi [10]. Such a model feeds back its own previous estimates and uses other past and current measurements to establish the value of the current estimate, as shown in Figure 2. In order to speed up the process, we used the NARX in open loop. A first estimate is done in a first stage and those estimates are used back as input measurements to further improve the performance of the model from one LSTM layer to the next. Notice that the second stage is a bidirectional LSTM for performance improvement. Each LSTM layer has 768 units with tanh activation, a dropout rate of 0.5 and a dense layer output with linear activation.

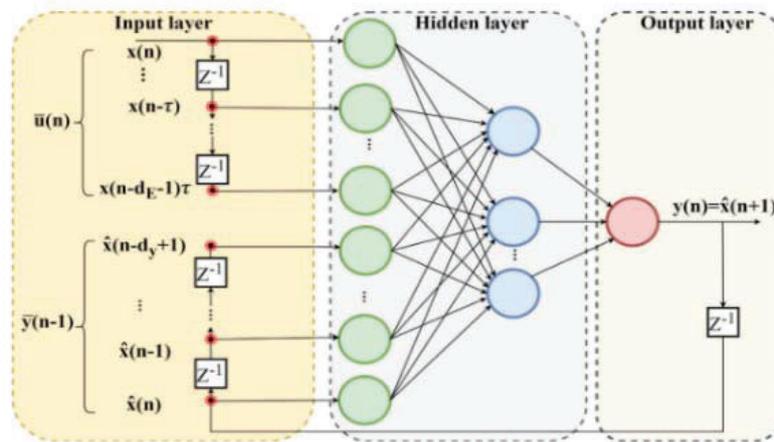


Figure 2. NARX architecture. [10]

3.2. Injector multi-scale attention network (Injector MA-Net)

For this study, we used 1D inputs divided in time windows of 128 values to estimate windows of 64 values for each runner blade strain gauge. First, the 1-D data windows are transformed to a 2-D representation using the Gramian Summation Angular Field (GASF) as described in [11]. Then, the data is fed to the Injector MA-Net that learns the relation between the 2D representation and the strain signal as shown in Figure 3.

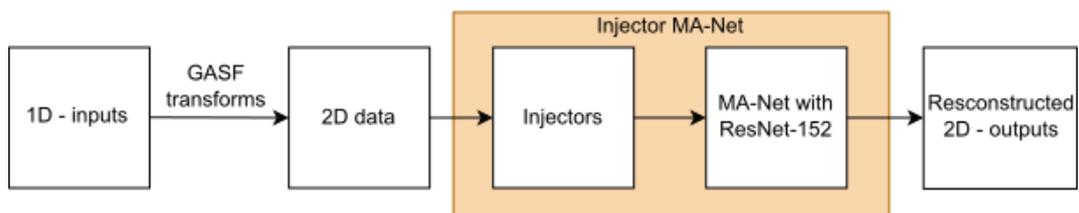


Figure 3. Injector MA-Net pipeline.

The Injector MA-Net is built using MA-Net [14] and a ResNet-152 [15] as its backbone. Prior to this step, injectors are used to downsample the input data to the appropriate dimension since there is a 2:1 dimensionality gap between inputs and the target. Each input has its own injector. We concatenate the injectors on the channel axis in order to feed MA-Net.

Finally, instead of directly using a Mean Square Error (MSE) as the loss function to drive the learning process, we use a 4 components weighted harmonic mean. The weighted harmonic mean is defined as:

$$\bar{x}_H = \frac{\sum_{i=1}^n \omega_i}{\sum_{i=1}^n \frac{\omega_i}{x_i}} \quad (1)$$

Each predicted signal ($\text{diag}(\hat{y})$) is extracted from an associated 2D output tensor (\hat{y}). The structure of a 2D output tensor is shown in Figure 4.

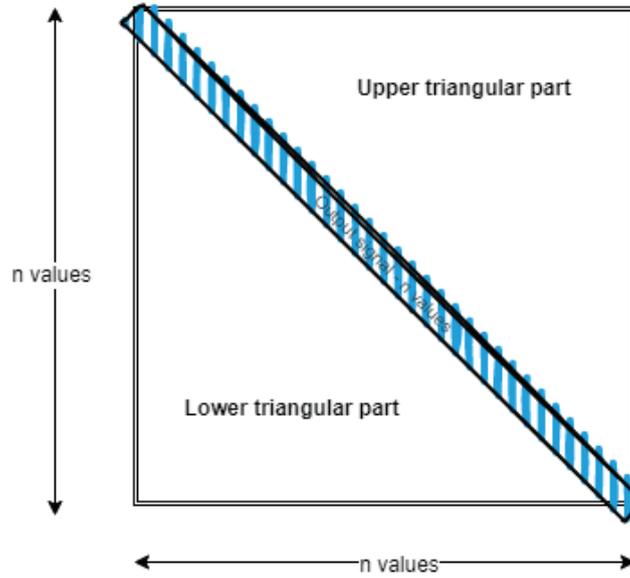


Figure 4. Structure of an output 2D tensor for a given predicted signal. The predicted signal corresponds to the diagonal

The first component in the loss is the signal reconstruction, L_1 , defined as (where n represents the number of predicted signals):

$$L_1 = \left(\frac{1}{n} \sum_{i=1}^n (\text{diag}(y_i) - \text{diag}(\hat{y}_i))^2 \right)^{1/2} \quad (2)$$

The second component present in the loss function is standard deviation of signal, L_2 :

$$L_2 = \left(\frac{1}{n} \sum_{i=1}^n (\sigma(\text{diag}(y_i)) - \sigma(\text{diag}(\hat{y}_i)))^2 \right)^{1/2} \quad (3)$$

Third and fourth components are the error predicting the upper and lower parts of the 2D output tensor, excluding the diagonal:

$$L_3 = \left(\frac{1}{n} \sum_{i=1}^n (\sigma(\text{triu}(y_i)) - \sigma(\text{triu}(\hat{y}_i)))^2 \right)^{1/2} \quad (4)$$

$$L_4 = \left(\frac{1}{n} \sum_{i=1}^n (\sigma(\text{tril}(y_i)) - \sigma(\text{tril}(\hat{y}_i)))^2 \right)^{1/2} \quad (5)$$

Applying the following weights respectively, $w_1=1$, $w_2=0.5$, $w_3=0.025$ and $w_4=0.025$, we obtain the following loss function to optimize:

$$L = 1.55 \left(\frac{1}{L_1} + \frac{0.5}{L_2} + \frac{0.025}{L_3} + \frac{0.025}{L_4} \right)^{-1} \quad (6)$$

3.3. NARXNN combined with Injector MA-Net

In order to combine both architectures, the Injector MA-Net estimate is simply used as an additional input to the LSTM layers at each stage of the NARXNN. The assumption is that the NARXNN will improve the results from the injector MA-Net.

4. Results

In Case 1, when only the value of one branch of a rosette is needed in Case 1, the results are almost perfect with the three architectures if the operating condition is present in the training dataset. The Case 1 results for NARXNN are shown in Figure 5. Notice that the values have been normalized and that it is almost impossible to distinguish the estimate from the measured signal.

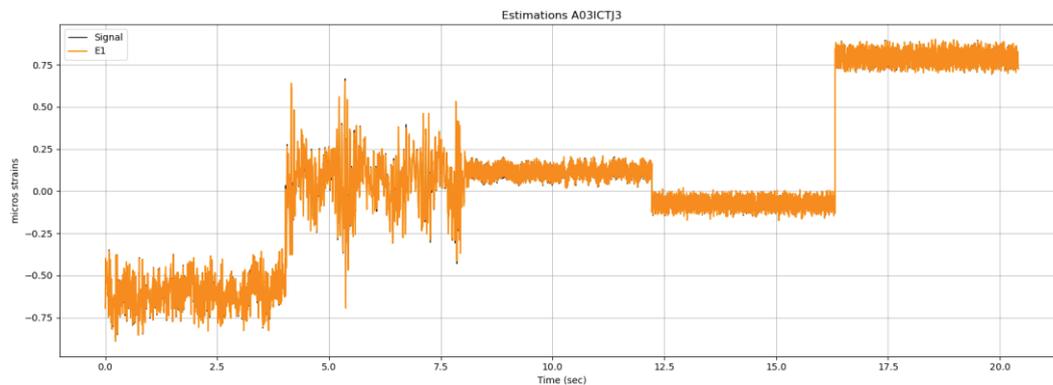


Figure 5. NARXNN Case 1 results for SNL, 39% Pmax, 73% Pmax, 85% Pmax and Pmax with the reference signal in black and the estimated in orange

Results remain good for Case 2 when the strains from other locations on the runner are available. However, in Case 3, if the strain data comes from another blade, NARXNN performance degrades. We observe that this primarily in the part load region at SNL and at 39% Pmax, as shown in Figure 6.

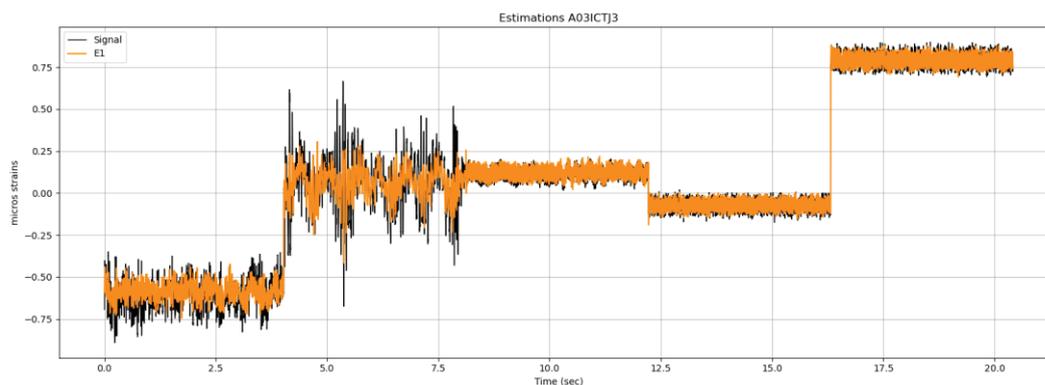


Figure 6. NARXNN Case 3 results for SNL, 39% Pmax, 73% Pmax, 85% Pmax and Pmax with the reference signal in black and the estimated in orange.

In Case 4, when no inputs from the runner blades are available, we observe that only the mean stress and a small part of the dynamic behavior can be recovered by the NARXNN architecture, as shown in Figure 7.

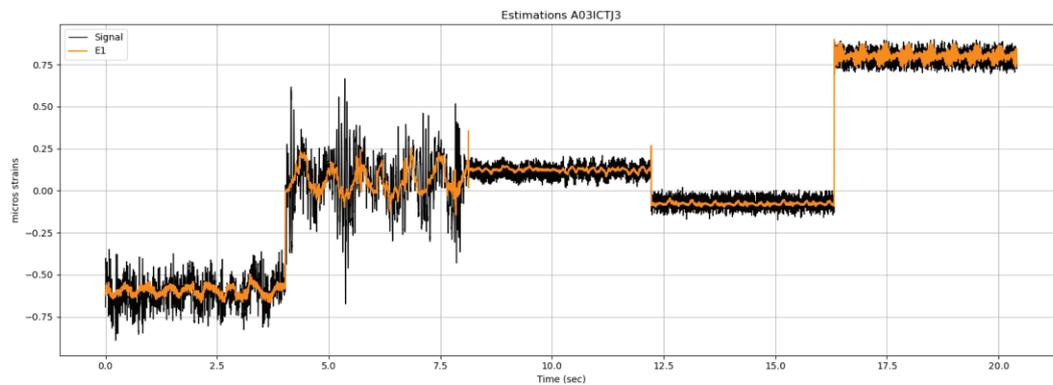


Figure 7. NARXNN Case 4 results for SNL, 39% Pmax, 73% Pmax, 85% Pmax and Pmax with the reference signal in black and the estimated in orange.

With the Injector MA-Net, the results are significantly better for Case 4, as shown in Figure 8. We observe that we can capture most of the dynamic behavior in the normal operating range (73% Pmax, 85% Pmax and Pmax). However, although the results improve, the part load dynamic behavior (SNL and 39% Pmax) still remains difficult to estimate properly. Furthermore, no correlations are enforced between the time windows, which generates some mismatch between the end of one window and the start of the next. Given the small window size this mismatch is usually negligible, but sometimes large discrepancies are generated, as we can see at 85% Pmax. We hoped this can be solved by combining it with the NARXNN LSTM layers.

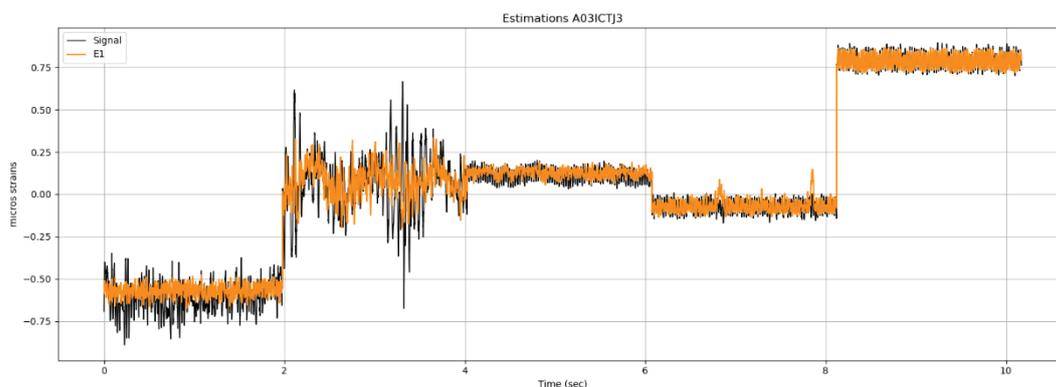


Figure 8. Injector MA-Net Case 4 results for SNL, 39% Pmax, 73% Pmax, 85% Pmax and Pmax with the reference signal in black and the estimated in orange.

The results from the NARXNN combined with Injector MA-Net architecture are shown in Figure 9. We observe that the results are better for 73% Pmax, 85% Pmax and Pmax, which represent the typical normal operating range. However, they seem to deteriorate for SNL and 39% Pmax. Notice that the sudden departures of the signal at 85% Pmax in Figure 8 are not present in Figure 9.

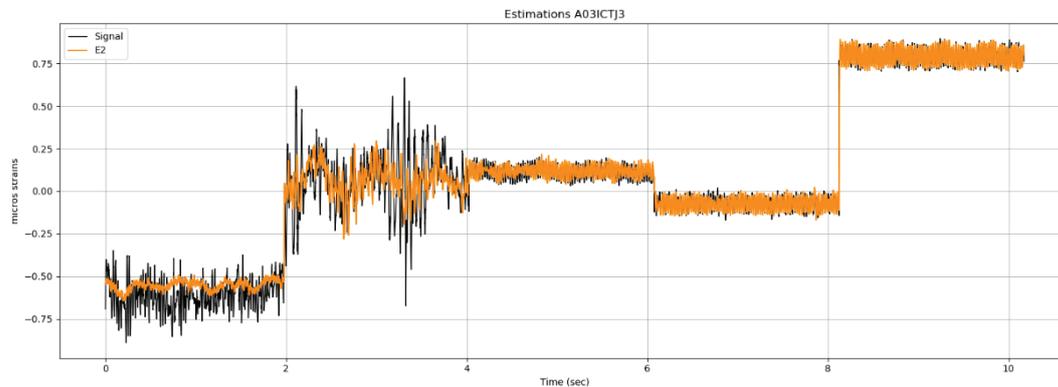


Figure 9. NARXNN combined with Injector MA-Net for SNL, 39% Pmax, 73% Pmax, 85% Pmax and Pmax with the reference signal in black and the estimated in orange.

Another difficulty arises when the estimated operating condition is not in the training dataset. In such a case, the results are good when we are close to the operating conditions in the training dataset but they deteriorate as we distance ourselves. We might even observe aberrant dynamic behavior if the type of dynamic response changes. As an example, the results are relatively good if only SNL is used for training until the apparition of the part load rope. Hence the importance of having, during training, a subset of operating conditions which cover all the expected dynamic behaviors close to the operating condition in which we want to impute values to obtain relevant results.

5. Discussion

Cases 1, 2 and 3 went quite smoothly for all the architectures even if we observed a decrease in performance for Case 3 (strain data from another blade). This points toward some variations between blades that are difficult to capture properly and highlights the need for uncertainty quantification and explainability. Our architectures have no built-in uncertainty quantification or explainability, which makes it hard to understand the observed decrease in performance. This needs to be addressed in the future.

The difficulties to imputing values are more visible in the part load operating conditions. This is in part due to the fact that in these conditions a significant component of the blade loading comes from draft tube instabilities. Since these instabilities might not completely propagate across the structure and influence the shaft displacements, correlation might not be possible with the input subset used. We also noticed that adding uncorrelated inputs tends to cause a deterioration of the performance. So, adding input from the draft tube pressure measurements might improve results at part load but deteriorate results in the other operating conditions. This should also be addressed in future studies.

Another thing to consider is that the Injector MA-Net imputes values using independent time windows. There is no enforced continuity between each window so the results exhibit discontinuity between windows that can be large, as shown in Figure 10. The goal of combining the Injector MA-Net with the NARXNN architecture was to prevent this. Figure 11 demonstrates that these have been somewhat corrected by the combined architecture, but the result is not perfect. In this case, we obtain a signal that is representative but not similar. This might limit the extent to which such a signal could be used. However, the training dataset and the inputs were significantly reduced in our study for the specific purpose of hindering the performance of the algorithms and architectures we are evaluating. We should expect better performance with the complete dataset and larger input subset.

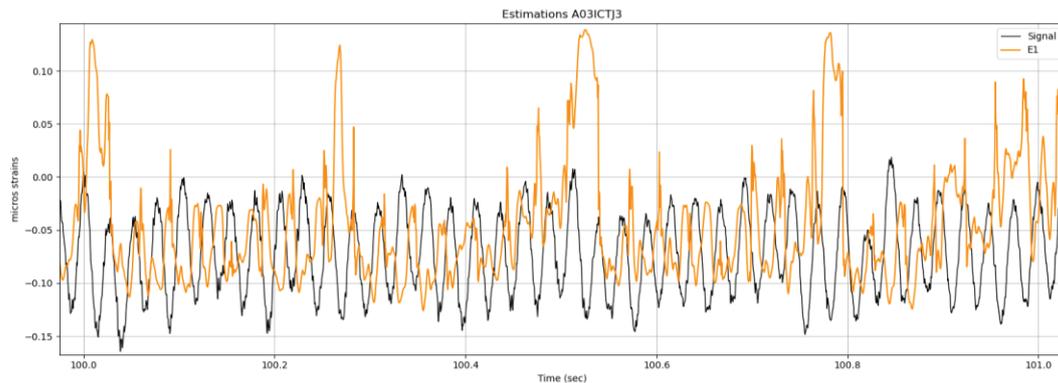


Figure 10. Injector MA-Net generated discontinuity with the reference signal in black and the estimated in orange.

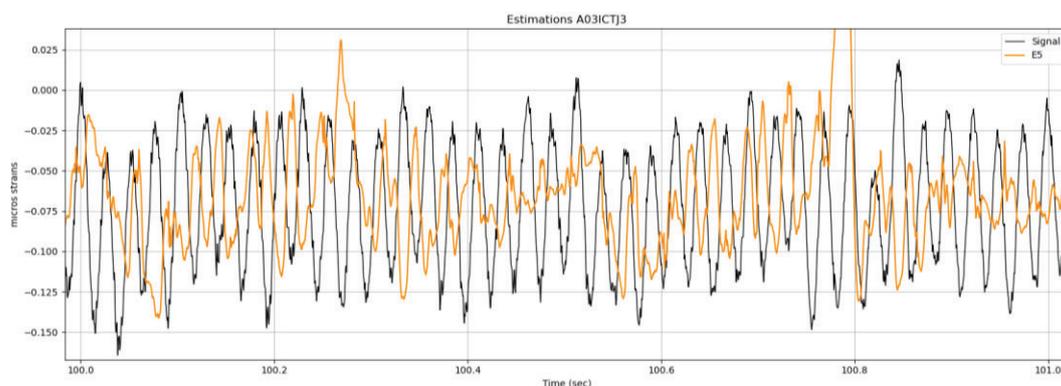


Figure 11. NARXNN combined with Injector MA-Net discontinuity correction with the reference signal in black and the estimated in orange.

6. Conclusions

In this paper, we demonstrated the performance of three types of neural network architectures in a virtual sensor setting where the goal was to impute missing data using the dataset from a dedicated measurement campaign made during the commissioning of a Francis turbine runner. The neural network architectures were NARXNN, Injector MA-Net and a combined architecture of both. Four cases were studied. For cases 1, 2 and 3, the results are good with all neural network architectures since the strain gages in the inputs are always highly correlated with the desired outputs. However, in Case 4, inputs from the guide bearings and labyrinth displacements—which are not as obviously correlated with the blade strains, and which are present in almost all monitoring systems—are used and only the Injector MA-Net is able to capture the dynamic behavior. The combined architecture further improve the results by correcting discontinuities generated by the Injector MA-Net. This opens the door for indirect measurement using neural networks in the context of online monitoring. While only a small subset of the available inputs have been used for this study, we noticed that adding uncorrelated input might deteriorate the performance, hence the inputs needed may not be the same for each operating condition. This needs to be addressed with proper uncertainty quantification and explainability. Such functionalities are readily available in some architectures from the literature, such as Time Fusion Transformer (TFT) [16], which have demonstrated good performance on general forecasting tasks. At this time, it is not clear which architecture would perform best in our context since there is a compromise to be achieved between the capacity to properly quantify uncertainty, accuracy and explainability. Furthermore, we only looked at steady state operating conditions. The performance of neural networks for transient regimes still needs

to be explored. Hence, we believe that more work needs to be done to understand the available neural network architectures on a wide range of realistic study cases before it becomes possible to propose a fully automated industrial solution for hydroelectric turbines.

References

- [1] Gagnon M, Tahan A, Bocher P and Thibault D 2013 A probabilistic model for the onset of High Cycle Fatigue (HCF) crack propagation: Application to hydroelectric turbine runner *International Journal of Fatigue* **47** 300–307
- [2] Gagnon M, Tahan A, Bocher P and Thibault D 2014 Influence of load spectrum assumptions on the expected reliability of hydroelectric turbines: A case study *Structural Safety* **50** 1–8
- [3] Chatenet Q, Tahan A, Gagnon M and Chamberland-Lauzon J 2016 Numerical model validation using experimental data: Application of the area metric on a Francis runner *28th IAHR symposium on hydraulic machinery and systems*, 4–8 of July 2016, Grenoble, France
- [4] Gagnon M, Immarigeon A, Nicolle J and Morissette JF 2018 Correlation between numerical simulations and measurements to assess uncertainties: a case study on a hydroelectric runner *29th IAHR symposium on hydraulic machinery and systems*, September 17–21, 2018, Kyoto, Japan
- [5] Marcouiller L and Thibault D 2015 Obtaining stress measurements on runners as a key contribution to reducing their degradation and improving the reliability of hydroelectric production units *HYDRO 2015*, 26–28 October 2015, Bordeaux, France
- [6] Diagne I, Gagnon M and Tahan A 2016 Modeling the dynamic behavior of turbine runner blades during transients using indirect measurements *28th IAHR symposium on hydraulic machinery and systems*, 4–8 of July 2016, Grenoble, France
- [7] Presas A, Valentin D, Zhao W, Egusquiza M, Valero C and Egusquize E 2021 On the use of neural networks for dynamic stress prediction in Francis turbines by means of stationary sensors *Renewable Energy* **170** 652–660
- [8] Gulgec NS, Takak M and Pakzad SN 2020 Structural sensing with deep learning: Strain estimation from acceleration data for fatigue assessment *Computer-Aided Civil and Infrastructure Engineering* **35** 1349–1364
- [9] Vouligny L, Gagnon M, Cauchon L and Giroux AM 2021 Anomaly detection and value imputation using virtual sensors in hydroelectric facility monitoring data *Mechanistic Machine Learning and Digital Twins for Computational Science, Engineering & Technology*, 26–29 September 2021, San Diego, CA, USA
- [10] Massaoudi M, Chihi I, Sidhom L, Trabelsi M, Refaat SS, Abu-Rub H and Oueslati FS 2021 An Effective Hybrid NARX-LSTM Model for Point and Interval PV Power Forecasting *IEEE Access* **9** 36571–36588
- [11] Wang Z and Oates T 2015 Imaging Time-Series to Improve Classification and Imputation *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI'15*, Buenos Aires, Argentina **7** 3939–3945
- [12] Kabadayi S, Pridgen A and Julien C 2006 Virtual sensors: abstracting data from physical sensors *International Symposium on a World of Wireless, Mobile and Multimedia Networks* 0-7695-2593-8
- [13] Cristaldi L, Ferrero A, Macchi M, Mehrafshan A and Arpaia P 2020 Virtual Sensors: a Tool to Improve Reliability *IEEE International Workshop on Metrology for Industry 4.0 & IoT* 978-1-7281-4893-9
- [14] Fan T, Wang G, Li Y and Wang H 2020 MA-Net: A Multi-Scale Attention Network for Liver and Tumor Segmentation *IEEE Access*, vol. 8, pp. 179656–179665
- [15] He K, Zhang X, Ren S and Sun J, Deep Residual Learning for Image Recognition 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778
- [16] Lim B, Arik S A, Loeff N and Pfister T 2021 Temporal Fusion Transformers for interpretable multi-horizon time series forecasting *International Journal of Forecasting* **37**