

Data Driven Fault Detection in Hydroelectric Power Plants based on Deep Neural Networks

Tiago Gaspar da Rosa

Department of Mechatronics and Mechanical Systems, University of São Paulo, Brazil. E-mail: tiago.gaspar@usp.br

Arthur Henrique de Andrade Melani

Department of Mechatronics and Mechanical Systems, University of São Paulo, Brazil. E-mail: melani@usp.br

Fabio Norikazu Kashiwagi

Department of Mechatronics and Mechanical Systems, University of São Paulo, Brazil. E-mail: fabiokashiwagi@yahoo.com.br

Miguel Angelo de Carvalho Michalski

Department of Mechatronics and Mechanical Systems, University of São Paulo, Brazil. E-mail: michalski@usp.br

Gilberto Francisco Martha de Souza

Department of Mechatronics and Mechanical Systems, University of São Paulo, Brazil. E-mail: gfmsouza@usp.br

Gisele Maria de Oliveira Salles

Companhia Paranaense de Energia - COPEL, Brazil. E-mail: gisele.salles@copel.com

Emerson Rigoni

Federal University of Technology – Parana, Brazil. E-mail: rigoni@utfpr.edu.br

In this paper, a set of fault detection methods that use variations of autoencoder based DNN was implemented over simulated data that emulates the behavior of a generating unit of a hydropower plant. These variations comprise the modulation of different hyperparameters, numbers, and types of layers, such as dense, long-short term memory (LSTM) and convolutional neural network (CNN). The use of advanced abnormality detections techniques for this kind of machinery, in special the deep learning related, have not been so explored if compared to the ones focused on assets of other power generating modalities. Hence, this study aims to investigate the feasibility and compare the performance of each one of the proposed methods in order to select potential candidates to be implemented in real operational scenarios.

Keywords: Fault detection, deep neural networks, hydroelectric power plants, condition-based maintenance, autoencoders, sensor fusion.

1. Introduction

Condition-based maintenance (CBM), whose primary objective is to identify upcoming equipment failure so that maintenance is proactively scheduled only when necessary, has been increasingly used in the industrial sector to improve asset's reliability, safety and increase overall system availability. Critical to the application of CBM, fault detection methods have been extensively studied, but industrial

applications in complex rotating machines are still in an early stage of development (Melani et al. 2021) However, with the increasing presence of sensors in industrial plants and the increasing ease of storing and managing monitoring data, the feasibility of applying the CBM strategy in the industrial context has risen significantly.

Recently, deep learning-based techniques for machine health monitoring and fault diagnosis have gained a lot of attention due to their

versatility and efficiency in extracting features from monitored data (Zhao et al. 2019). Deep neural networks (DNN), in particular, have been increasingly applied in fault detection due to their ability to perform sensor data fusion, i.e., to combine different monitored variables aiming at increasing accuracy over the detection results.

In this paper, a set of fault detection methods that use variations of autoencoder based DNN was implemented over simulated data that emulates the behaviour of a generating unit of a hydropower plant. These variations comprise the modulation of different hyperparameters, numbers, and types of layers, such as dense, long-short term memory (LSTM) and convolutional neural network (CNN). The use of advanced abnormality detections techniques for this kind of machinery, in special the deep learning related, have not been so explored if compared to the ones focused on assets of other power generating modalities. Hence, this study aims to investigate the feasibility and compare the performance of each one of the proposed methods in order to select potential candidates to be implemented in real operational scenarios.

The paper is organized as follows: Chapter 2 presents a very brief introduction of the autoencoder based DNNs, offering several bibliographic references on the subject; Chapter 3 presents the proposed method for using autoencoder based DNNs for data drive fault detection in Hydroelectric Power Plants (HPPs); Chapter 4 shows the results obtained by applying the method to simulated failure data from a HPP and; Chapter 5 presents the conclusions derived from this work.

2. Autoencoder based DNNs

As an extension of traditional artificial neural networks, DNNs can be seen as a stack of neural networks, or as a network composed of several layers (Das and Roy 2019). There are several types of DNN models, such as CNNs, recurrent neural networks (RNNs), LSTM and autoencoders (Subasi 2020).

Autoencoders based DNNs introduce a bottleneck on the network layers to create a compressed representation of the input data. Such compression helps the network to capture the dependencies or correlations present in the input data, contributing to the elimination of mutually dependent features (Koul and Manvi 2021).

Figure 1 presents the general structure of an autoencoder based DNN. Here, the input vector, x , is transformed by the encoder into the compressed feature vector, z , as shown in Eq. (1).

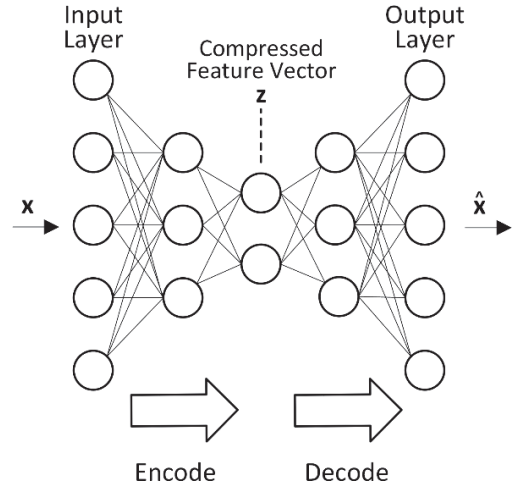


Fig. 1. Structure of a basic autoencoder based DNN.

$$z = f(W_1 x + b_1) \quad (1)$$

where f is the encoder activation function, W_1 is the weight matrix and b_1 is the bias vector. The compressed feature vector is then reconstructed back into $\hat{x}$ via the decoder, as shown in Eq. (2).

$$\hat{x} = g(W_2 z + b_2) \quad (2)$$

where, similarly to Eq. (1), g is the decoder activation function, W_2 is the weight matrix and b_2 is the bias vector.

For an autoencoder based DNN to be properly trained, it is first necessary to define a set of hyperparameters that characterize the structure of the network to be used. It has to be defined, for instance, the number and type of layers to be used in the network. Such layers can be, for example, dense, LSTM (Zhang and Qiu 2022) or CNN (Jana et al. 2022). The bottleneck size, i.e., the compressed feature vector size, is another hyperparameter to be tuned in the network, as it decides how much compression the data has to go through.

As the hyperparameters substantially impact the result obtained by the network, it is possible to test sets of different hyperparameters and

choose the one that presents the best overall performance.

Autoencoder based DNNs and its variations have been used to solve problems of fault detection and diagnosis (Wu et al. 2021; Yang, Baraldi, and Zio 2022; Liu et al. 2019; Meng et al. 2018), prognostics health management (Zhao et al. 2019) and remaining useful life (Kong et al. 2019). In this paper, they will be used for fault detection in Hydroelectric Power Plants (HPPs).

3. Proposed Method

A dataset with 26 entries that represent common variables usually monitored in a hydro generator unit provides a general panorama of their subsystems. For the current study, 18 experiments have been simulated, six for each fault type described in Table 1, recording the machine behavior after and before the symptoms of degradation are inserted. For the simulations under normal operating conditions, before the symptoms of degradation, data from a real hydro generator were taken as a basis. Correlations were defined between the monitored parameters and their operational limits. In this operational condition, the simulated data follows a random walk between the defined limits, seeking to respect both the distribution and the oscillatory pattern of the real data. After entering degradation symptoms, exponential gains (Faults 1 and 2) and an additive gain (Fault 3) are added to simulate each type of degradation. Such a solution for generating synthetic data was previously adopted by (Melani et al. 2021)

Furthermore, a simulated experiment comprises 5749 data points, 5000 in normal operating conditions, and 749 in a degraded state, such that the 5000th entry is the turning point for the abnormality. The collection interval between observations of the sensors is set to 4 hours.

Table 1. Simulated fault types.

Faults	Description	Expected Behavior
1	Generator Shaft Excessive Vibration	Exponential trend
2	Stator copper insulation degradation 0° Fault	Exponential trend
3	Temperature Sensor of Generator Combined Bearing Outlet (hot) Water Fault	Amplitude increase

The data preparation procedure consists in standardizing the data by removing the mean and scaling to unit variance and then generating a set of subsequences that will feed the deep learning models. There is an adoption of a moving window with a temporal iteration step of one to sample the subsequences of size n . As expected, the autoencoders are trained with subsequences that represent the normal operating conditions and validated using a fraction of 10 % of the preprocessed samples. The samples labelled as fault are used to test the abnormality detection capability in the sense of indicating accurately the transition for the degraded state, so an offset of 700 points before the turning point is jointed with these samples, making possible the perception of false positives alarms. The performance of the models on the referred task is measured by the detection coverage, Eq. (3), over the data points labeled as abnormal and the false-positive coverage, Eq. (4), over the ones labeled as normal. Then, in a given experiment j on the faulty condition i :

$$d_{ij} = \frac{\sum \ell_{di}}{\ell_d} * 100 \quad (3)$$

$$f_{ij} = \frac{\sum \ell_{fi}}{\ell_{of}} * 100 \quad (4)$$

Where ℓ_d is the length of the interval I_d , which the system is in the degraded state and ℓ_{of} refers to the length of the interval I_n , which the system is in the normal state. $\sum \ell_{di}$ stands for the sum of all subintervals of I_d signaled by the method as an abnormality and $\sum \ell_{fi}$ is the sum of subintervals of I_n highlighted on the same condition.

Moreover, the convergence of the autoencoders for the training and validation sets are also an assessment criterion, since overfitting and poor time series reconstruction capacity are unwanted, because it leads to either: false-positive occurrences or undetection of the fault. For the context of this work, false positives are considered all the abnormality indications linked to the indexes before the beginning of the degradation pattern.

The decision criteria to classify whether an entry is abnormal or not is based on a threshold defined by the maximum reconstruction error E over the entire sets of training and validation. That is, the autoencoder input x_i with shape $(26, n)$ and prediction x_p are compared through mean

squared error, returning a vector of length n , that sequentially organizes the pointwise errors of the feature space retrieval. The average of the components of this vector is what is understood as the reconstruction error addressed for the index of the data point that marks the end of the subsequence.

To avoid discontinuities in abnormally signaled intervals, a specified amount of consecutive entries must have E above the threshold value. It is established as $\frac{1}{4}$ of the input length. If this parameter is below 200, thus that quantity is fixed in 50.

As mentioned previously, three autoencoders with different kinds of layers have been assembled, which are convolutional unidimensional layers, stacked Long Short Term Memory, and a simple dense multilayer perceptron. A summary of the models' hyperparameters is presented in Table 2, highlighting that some of them are modulated into a grid-search experiment with the objective of an ablation study and tuning to the proposed finality. The kinds of architectures are detailed in Table 3 to 5.

Table 2. Hyperparameters specification of the analyzed autoencoders.

Global Hyperparameters	
Number of layers – Encoder only	2
Dropout rate*	0,3
Loss Function	MSE
Optimization Technique	Adam
Subsequences Size	[60, 200, 300, 400, 500, 700]
Validation data fraction	10%
1D-CNN Hyperparameters	
Strides	2
Learning Rate	0,001
Number of filter units	See Table 8
Kernel Size	See Table 10
Padding	'same'
Activation Function	<i>LeakyReLU</i>
Epochs	80
LSTM Hyperparameters	
Activation Function	tanh
Number of LSTM units	See Table 8
Epochs	100
MLP Hyperparameters	
Activation Function	<i>LeakyReLU</i>
Neurons	See Table 8
Epochs	500

Table 3. Description of the 1D-convolutional architecture. Sequential layers ordered from the input to output of the network.

Nº	Layer	Activation	Output Shape
1	Input Layer		(200,26)
2	1D-Conv.	<i>LeakyReLU</i>	(100,32)
3	Dropout		(100,32)
4	1D-Conv.	<i>LeakyReLU</i>	(50,16)
5	1D-Conv.	<i>LeakyReLU</i>	(100,16)
6	Transpose		(100,16)
7	Dropout		(100,16)
8	1D-Conv.	<i>LeakyReLU</i>	(200,32)
	Transpose		(200,26)

Table 4. Description of the LSTM architecture.

Nº	Layer	Activation	Output Shape
1	Input Layer		(200,26)
2	LSTM	tanh	(200,128)
3	LSTM	tanh	(,64)
4	Repeat Vector		(200,64)
5	LSTM	tanh	(200,64)
6	LSTM	tanh	(200,128)
7	Time Distributed (Dense)		(200,26)

Table 5. Description of the dense architecture.

Nº	Layer	Activation	Output Shape
1	Input Layer		(200,26)
2	Dense	<i>LeakyReLU</i>	(200,32)
3	Dropout		(200,32)
4	Dense	<i>LeakyReLU</i>	(200,16)
5	Dropout		(200,16)
6	Dense	<i>LeakyReLU</i>	(200,16)
7	Dropout		(200,16)
8	Dense	<i>LeakyReLU</i>	(200,32)
9	Dense	<i>LeakyReLU</i>	(200,26)

4. Results

From all the trained models on the hyperparameters search space, only a fraction could accomplish convergence that presents loss on the validation set lower than 0.4, as observed in Figure 2, which is considered the limit for the purposes required in this study. A new grid search would be necessary if the objective were to produce a fine-tuning of the models.

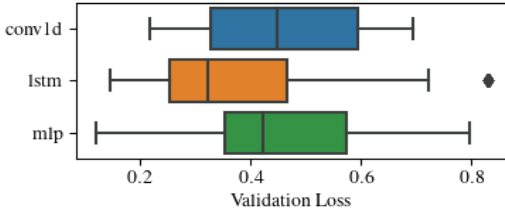


Fig. 2. Validation loss for all the trained models grouped by kind of hidden layer.

Still according to Figure 2, it has perceived that multilayer perceptron achieves the best fitting between the three kinds of layers because it has been trained with a higher number of neurons and during more epochs. Also by its simplicity, multilayer perceptron is the one that introduces the smallest among of operations on the input data and then have had less capacity to codify information in the time and feature domain. The fidelity of the reproduction allows inferring that it acts more like a common neural network than specifically an autoencoder. Although MLP performs better in terms of detection coverage, it also has the highest false-positive coverage, which could be interpreted from the Figures 3 and 4, being a fact that corroborates with the previous affirmation.

The modulation of the subsequence size has a small influence on the detection coverage in general for the convolution-based detectors, slightly raising for the third fault in Table 1 when bigger timesteps are adopted (500,700) - see Table 9. However, this hyperparameter, when increased, negatively influences the false-positive coverage, which is counterintuitive, considering the decision criteria applied and because longer subsequences tend to smooth traits that could be recognized as abnormalities. The possible explanation is that at a certain point (700) it is necessary compensate the subsequence size with a greater number of filters or neuronal units to be able to extract features properly.

The kernel size held a small influence on detection and false-positive coverage. The most prominent one is the decrease of the false positive rate for the higher tested kernel (25,15), generated by the diffusion of the irregular patterns per convolutional window. Besides this effect, the higher dimensioned kernel shows up poor average fitting in comparison with the others settings.

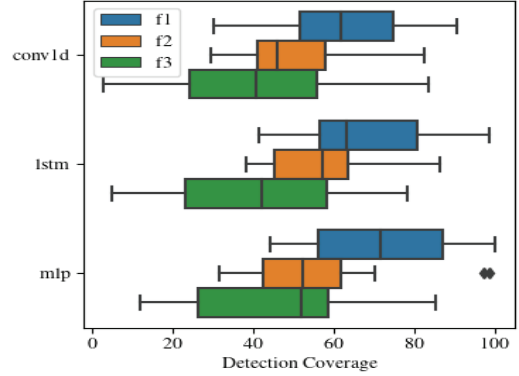


Fig. 3. Detection coverage for the tree kind of layers in each simulated fault.

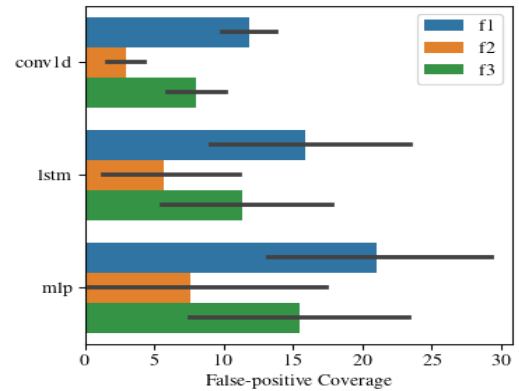


Fig. 4. False-positive coverage for the tree kind of layers in each simulated fault.

The number of units has been the argument that affected the most the fitting of the model. Its modulation allows perceiving the transition from an underfitting to an overfitting condition. On the first, there are low false-positive coverage and low detection coverage, and on the second, the two present themselves high. The same could be observed with the number of LSTM units per layer or neurons on a dense type. Additionally, the number of units determines the output space dimension, thus affecting the compressibility capacity of the autoencoder.

In general, the examined models have demonstrated better performance over the two first kinds of faults in Table 1, which have similar damage evolution behaviour. The third fault, on the other hand, has presented the worst detection coverage by almost all the combinations of hyperparameters.

Tables 6 and 7 displays the best optimized models, ordered according to the weighted average $w_a = 0.7d_m + 0.3f_m$, with d_m and f_m being the mean of metrics described in Sec. 3 for all the samples tested in a named version, considering only elements that reached $d_m > 70$ and $f_m < 30$. Notably, unidimensional convolution is the most recurrent kind on this table, but because it was the predominant type among the generated instances. The chosen kernel and number of units are in agreement with the discussed above. The subsequences length does not pronounce great impact on performance, which could be perceived also by the Table 9. For this study, both LSTM and 1d-convolutional autoencoder are equally suitable if well tuned.

Figures 5 to 7 show the progression of the reconstruction error for the best representative of each type in Table 6.

Table 6. Parameters setting for the best-optimized models according to the weighted average w_a of the performance metrics.

Model	Kind	l_t	Units	Kernels
1	conv1d	300	(32, 16)	(10, 5)
2	lstm	200	(128, 64)	
3	conv1d	200	(32, 16)	(10, 5)
4	conv1d	700	(32, 16)	(10, 10)
5	conv1d	400	(32, 16)	(10, 10)
6	conv1d	60	(32, 16)	(10, 10)
7	conv1d	500	(32, 16)	(10, 10)
8	mlp	200	(64, 32)	
9	conv1d	500	(32, 16)	(10, 5)
10	mlp	500	(64, 32)	

Table 7. Performance metrics and their weighted average w_a for the best-optimized models.

Model	d_m	f_m	Validation loss	w_a
1	76.51	21.09	0.25	77.23
2	78.46	27.18	0.15	76.77
3	76.87	24.12	0.22	76.58
4	75.79	23.02	0.29	76.15
5	71.84	15.35	0.27	75.69
6	71.74	15.37	0.28	75.61
7	75.62	25.66	0.24	75.24
8	74.91	24.2	0.24	75.18
9	74.99	24.78	0.24	75.06
10	74.07	22.89	0.25	74.98

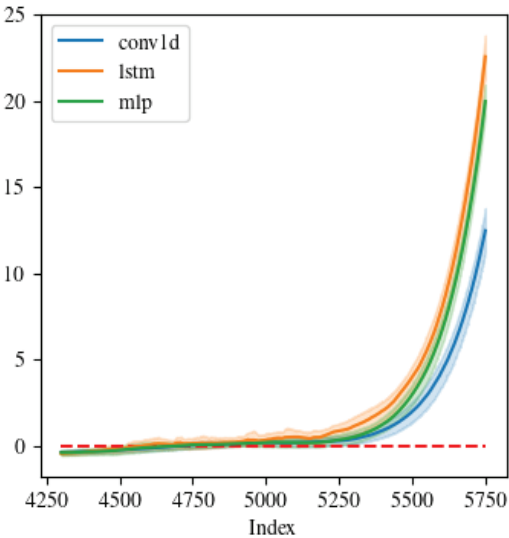


Fig. 5. Evolution of the reconstruction error through the test labelled index for the tree kinds of AE, when fault 1 related degradation is inserted at index 5000. The variable is normalized by the threshold E , such that a potential abnormality is indicated by values above 0.95 % c.i. over the five experiments.

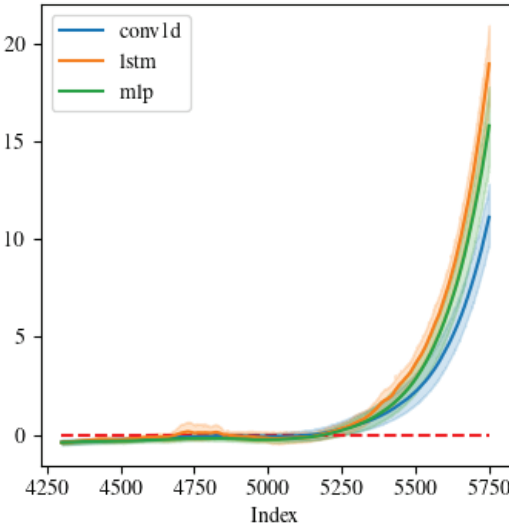


Fig. 6. Evolution of the reconstruction error, when fault 2 related degradation is inserted as in Fig. 6.

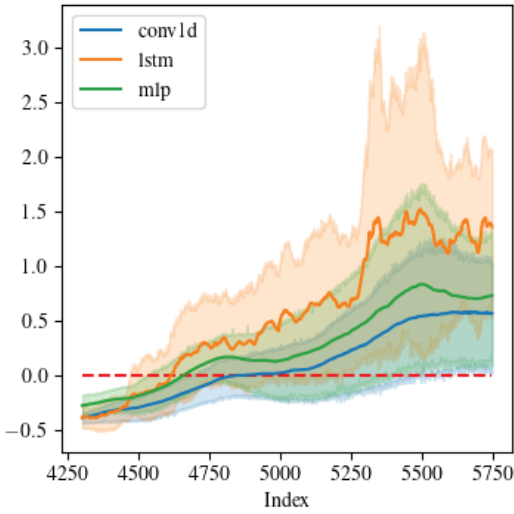


Fig. 7. Evolution of the reconstruction error, when fault 3 related degradation is inserted as in Fig. 6.

5. Conclusions

The examined autoencoders could detect abnormalities in the simulated conditions of the hydro generator, although with limitations. It is perceived that the tested architectures are, in purpose of investigation, the consolidated ones, without embracing the recent advances in this kind of network. Thereupon, there is space for performance improvements by either exploring the abnormality decision criteria, model design, or featuring engineering the data, especially with contributions enable to handle many input channels and multiple fault types, since it prevails in real industrial applications.

Acknowledgements

This paper presents part of the results obtained with the execution of the project PD-06491-0341-2014 “Methodology for asset management applied to hydro-generators based on mathematical models of reliability and maintainability” carried out by the Federal University of Technology at Parana and University of Sao Paulo to COPEL Geração e Transmissão S.A within the scope of the Electric Sector’s Research and Technological Development Program regulated by National Agency of Electrical Energy (ANEEL).

Appendix A. Evaluation metrics for the modulated parameters

Table 8. Detection, false positive coverage and standards deviation s computed by the number of units per encoder layer. As an example (32,16), means 32 first hidden layer units and 16 second hidden layer units. The decoder has an antisymmetric disposition.

1D-CNN				
Units	d_m	$s(d_m)$	f_m	$s(f_m)$
(32, 16)	63.38	32.61	14.97	23.41
(16, 8)	56.43	30.9	8.23	17.91
(16, 4)	41.76	28.25	3.43	14.3
(8, 4)	38.49	25.7	2.49	10.28
LSTM				
(128, 64)	78.46	28.23	27.18	27.92
(128, 32)	52.65	29.55	1.51	4.01
(128, 16)	50.76	24.33	3.39	9.01
(64, 32)	60.04	27.29	5.64	11.76
(64, 16)	56.83	29.75	3.35	6.76
(32, 16)	58.1	31	11.69	20.47
(32, 8)	36.35	27.4	3.6	15.22
(16, 8)	41.6	26.9	2.85	4.94
(16, 4)	32.47	22.6	0	0
(8, 4)	33.43	23.49	0.41	1.75
(64, 64)	81.06	26.38	37.23	24.79
MLP				
(128, 64)	92.55	22.9	53.8	26.99
(64, 32)	74.49	30.79	23.54	27.7
(32, 16)	54.16	30.99	8.07	18.1
(16, 4)	39.51	24.63	2.02	5.46
(8, 4)	32.86	20.63	0.17	0.71

Table 9. Detection, false positive coverage and standards deviation s computed by the subsequence size.

1D-CNN				
t_s	d_m	$s(d_m)$	f_m	$s(f_m)$
60	28.01	38.75	5.21	14.48
200	41.06	42.93	9.99	19.02
400	41.33	43.41	6.03	13.04
500	43.91	43.79	8.54	17.93
700	49.89	44.83	15.45	28.87
LSTM				
60	36.56	35.79	5.17	10.01
200	39.36	42.31	10.98	21.06
400	58.3	48.8	11.21	17.77
500	78.08	35.13	32.05	22.19
700	55.21	45.06	11.83	28.84
MLP				
60	24.91	32.48	0.66	2.27
200	47.13	44.69	20.17	28.71
400	40.3	47.7	0.95	2.33
500	56.3	45.29	22.34	28.37
700	47.73	50.38	10.4	17.8

Table 10. Detection, false positive coverage and standards deviation s computed by the kernels size of the 1D-CNN architecture.

Kernels	d_m	$s(d_m)$	f_m	$s(f_m)$
(15, 10)	42.23	42.65	7.23	15.25
(10, 5)	43.12	43.93	11.04	21.36
(25, 10)	34.62	41.27	4.27	13.07
(10, 10)	39.48	43.41	9.41	20.53

References

Das, H. S., and P. Roy. 2019. "A Deep Dive into Deep Learning Techniques for Solving Spoken Language Identification Problems." In *Intelligent Speech Signal Processing*, 81–100. Elsevier. <https://doi.org/10.1016/B978-0-12-818130-0.00005-2>.

Jana, D., J. Patil, S. Herkal, S. Nagarajaiah, and L. Duenas-Orsorio. 2022. "CNN and Convolutional Autoencoder (CAE) Based Real-Time Sensor Fault Detection, Localization, and Correction." *Mechanical Systems and Signal Processing* 169 (April). <https://doi.org/10.1016/j.ymssp.2021.108723>.

Kong, Z., Y. Cui, Z. Xia, and H. Lv. 2019. "Convolution and Long Short-Term Memory Hybrid Deep Neural Networks for Remaining Useful Life Prognostics." *Applied Sciences (Switzerland)* 9 (19). <https://doi.org/10.3390/app9194156>.

Koul, N., and S. K. S. Manvi. 2021. "Computational Intelligence Techniques for Cancer Diagnosis." In *Recent Trends in Computational Intelligence Enabled Research*, 95–110. Elsevier. <https://doi.org/10.1016/b978-0-12-822844-9.00032-3>.

Liu, X., Q. Zhou, J. Zhao, H. Shen, and X. Xiong. 2019. "Fault Diagnosis of Rotating Machinery under Noisy Environment Conditions Based on a 1-D Convolutional Autoencoder and 1-D Convolutional Neural Network." *Sensors (Switzerland)* 19 (4). <https://doi.org/10.3390/s19040972>.

Melani, A. H. A., M. A. C. Michalski, R. F. Silva, and G. F. M. Souza. 2021. "A Framework to Automate Fault Detection and Diagnosis Based on Moving Window Principal Component Analysis and Bayesian Network." *Reliability Engineering and System Safety* 215. <https://doi.org/10.1016/j.ress.2021.107837>.

Meng, Z., X. Zhan, J. Li, and Z. Pan. 2018. "An Enhancement Denoising Autoencoder for

Rolling Bearing Fault Diagnosis." *Measurement: Journal of the International Measurement Confederation* 130 (December): 448–54. <https://doi.org/10.1016/j.measurement.2018.08.010>.

Subasi, A. 2020. "Machine Learning Techniques." In *Practical Machine Learning for Data Analysis Using Python*, 91–202. Elsevier. <https://doi.org/10.1016/b978-0-12-821379-7.00003-5>.

Wu, X., Y. Zhang, C. Cheng, and Z. Peng. 2021. "A Hybrid Classification Autoencoder for Semi-Supervised Fault Diagnosis in Rotating Machinery." *Mechanical Systems and Signal Processing* 149 (February). <https://doi.org/10.1016/j.ymssp.2020.107327>.

Yang, Z., P. Baraldi, and E. Zio. 2022. "A Method for Fault Detection in Multi-Component Systems Based on Sparse Autoencoder-Based Deep Neural Networks." *Reliability Engineering and System Safety* 220 (April). <https://doi.org/10.1016/j.ress.2021.108278>.

Zhang, S., and T. Qiu. 2022. "Semi-Supervised LSTM Ladder Autoencoder for Chemical Process Fault Diagnosis and Localization." *Chemical Engineering Science* 251 (April). <https://doi.org/10.1016/j.ces.2022.117467>.

Zhao, R., R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao. 2019. "Deep Learning and Its Applications to Machine Health Monitoring." *Mechanical Systems and Signal Processing*. Academic Press. <https://doi.org/10.1016/j.ymssp.2018.05.050>.