Proceedings

Machine Learning Models Applied to Predictive Maintenance in Automotive Engine Components †

Iron Tessaro [1,*], Viviana Cocco Mariani [2,4] and Leandro dos Santos Coelho [3,4]

[1] Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR), Curitiba 80215-901, Brazil; irontessaro@gmail.com

[2] Mechanical Engineering Graduate Program (PPGEM), Pontifical Catholic University of Parana (PUCPR), Curitiba 80215-901, Brazil; viviana.mariani@pucpr.br

[3] Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR), Curitiba 80215-901, Brazil; leandro.coelho@pucpr.br

[4] Department of Electrical Engineering, Federal University of Parana (UFPR), Curitiba 80060-000, Brazil

* Correspondence: iron_lukas@hotmail.com

† Presented at the First International Electronic Conference on Actuator Technology: Materials, Devices and Applications, 23–27 November 2020; Available online: https://iecat2020.sciforum.net/.

Published: 21 November 2020

Abstract: Fault detection on automotive engine components is an important feature that motivates research from different engineering areas due to the interest of automakers in its potential to increase safety, reliability, and lifespan and to reduce pollutant emissions, fuel consumption, and maintenance costs. The fault detection can be applied to several types of maintenance strategies, ranging from finding the faults that generated a component failure to finding them before the failure occurs. This work is focused on predictive maintenance, which aims to constantly monitor the target component to detect a fault at the beginning, thus facilitating the prevention of target component failures. It presents the results of different machine learning methods implemented as classification predictors for fault detection tasks, including Random Forest (RF), Support Vector Machines (SVM), Artificial Neural Networks (ANN) variants, and Gaussian Processes (GP). The data used for training were generated by a simulation testbed for fault diagnosis in turbocharged petrol engine systems, whereby its operation was modeled using industrial-standard driving cycles, such as the Worldwide Harmonized Light Vehicle Test Procedure (WLTP), New European Driving Cycle (NEDC), Extra-Urban Driving Cycle (EUDC), and the United States Environmental Protection Agency Federal Test Procedure (FTP-75).

Keywords: machine learning; random forest; Gaussian processes; artificial neural network; support vector machine; fault detection; automotive engine; simulation; predictive maintenance

1. Introduction

Due to production costs, it is not possible to have sensors installed in all engine components, which makes it difficult to apply predictive maintenance for all of them [1]. One way to work around that problem is to use predictors based on machine learning paradigms that can use signals from indirect sensors and predict a fault [2]. To accomplish that, it is necessary to acquire data that contain both normal and faulty behaviors from the target component to train a machine learning method to recognize the defective behavior before embedding it into the software of an engine electronic control unit [3].

Data acquisition can be an expensive process as well, as it may require several rounds of destructive testing for different driving cycles, which must be performed in real time on instrumented vehicles in a dynamometer [4]. Since machine learning methods are capable of handling a certain amount of noise, the data to train them can be generated by simulating the model of the respective engine in which the target component is installed. That process does not require real-time executions and vehicle instrumentation in a dynamometer lab, which decreases the cost of the data acquisition as a whole [5]. Simulated data-driven training of machine learning methods has been used in different applications [6–9], showing that it is a valid approach to problems where it is possible to entirely or partially model the system where it is going to be applied.

The purpose of this work is to show the feasibility of using different machine learning approaches as predictors of fault diagnosis, for predictive maintenance purposes, by using training and testing datasets of different standardized driving cycles, generated by a simulation testbed for fault diagnosis in turbocharged petrol engine systems.

The remainder of this article is organized as follows. The experiments are detailed in Section 2. Section 3 presents the fundamentals of machine learning approaches. After, the results analysis and discussion are presented in Sections 4 and 5, respectively. The conclusion is mentioned in Section 6.

2. Experiments

A platform for the evaluation of fault diagnosis algorithms and strategies [10], implemented in the Matlab/Simulink computational environment, was used to simulate and build all necessary data that were used to train the machine learning methods. That platform is a simulation testbed for fault diagnosis in turbocharged petrol engine systems, which allows for the selection of fault modes for different components and driving cycles. For this work, the fault mode applied was a leakage in the compressor system for all four driving cycles available, i.e., Worldwide Harmonized Light Vehicle Test Procedure (WLTP), New European Driving Cycle (NEDC), Extra-Urban Driving Cycle (EUDC), and the United States Environmental Protection Agency Federal Test Procedure (FTP-75), all with a sampling time equal to 35 milliseconds.

2.1. Dataset

The dataset was divided into two subsets: training and testing sets. The training sets were comprised of three out of four driving cycles, i.e., NEDC, EUDC, and FTP-75, whereas the WLTP was used for testing purposes. The target used by the machine learning methods was a binary error flag, resulting from the normalization of the residual value provided by the simulation. There were fourteen signals available in the simulator that could be used to feed the machine learning methods, as inputs. Five among them were selected by a brute force algorithm, which compared the best accuracy for combinations of them against the accuracy of when they were used. The selected inputs were the following:

1. Ambient pressure [Pa];
2. Compressor temperature [K];
3. Compressor pressure [Pa];
4. Intercooler temperature [K];
5. Intake manifold temperature [K].

The dataset comprised 146,606 samples, where 94,971 made up the training set, and 51,635 composed the testing set. Inputs and targets were normalized between 0 and 1 and only for the training dataset, they have shuffled afterward. The testing dataset was kept unshuffled to keep the time-series signal for plotting purposes.

2.2. Machine Learning Methods

Five machine learning algorithms were chosen and implemented by using Matlab's built-in functions for this work.

2.2.1. Single-Layer Feed-Forward Neural Network

An artificial neural network is composed of the connection of two or more mathematical elements called artificial neurons. These neurons act as functions that receive multiple inputs and produce a single output. A weight is assigned for each input of the artificial neuron as well as a bias for each neuron itself. The weighted inputs and the bias are added together, resulting in a linear output that is fed into a non-linear function, i.e., activation function, which is common to all neurons within the same layer [11]. A single-layer feed-forward neural network (SLFN) is an artificial neural network with only one hidden layer, formed by parallel artificial neurons, which connect the input neurons to the output neurons [12]. The configuration adopted in this work used 100 neurons in the hidden layer with a hyperbolic tangent activation function, along with a linear neuron in the output.

2.2.2. Random Vector Functional Link Networks

Random Vector Functional Link Networks (RVFL) is an SLFN in which the weights and biases of the hidden neurons are randomly generated within a suitable range and kept fixed while the output weights are computed via a simple closed-form solution [13,14]. Randomization-based neural networks benefit from the presence of direct links from the input layer to the output layer as in RVFL [15]. The original features are reused or propagated to the output layer via the direct links. The direct links act as a regularization for the randomization [16]. It also helps to keep the model complexity low with the RVFL being smaller and simpler compared to its other counterparts, which makes the RVFL attractive to use compared to other similar randomized neural networks [17]. The setup adopted in this work had 95 neurons in the hidden layer with hyperbolic tangent activation function, 5 enhanced neurons, and a linear neuron in the output. The total number of neurons was kept the same as in the SLFN structure (i.e., 100 neurons).

2.2.3. Support Vector Machines

Support vector machine (SVM) is a supervised learning algorithm that follows the principle of structural minimization of dimensional risk, based on the Vapnik–Chervonenkis theory [18]. Its goal is to classify a given set of data points, which are mapped to a multidimensional feature space using a kernel function by representing a decision limit as a hyperplane in a higher dimension, in the feature space [19]. One of the crucial ingredients for SVM is the so-called kernel trick which allows for the computation of scalar products in spaces of high dimension characteristics using simple functions, defined in pairs of input patterns. This trick allows for the formulation of non-linear variants for any algorithm that can be expressed in terms of scalar product; the most promising of these is SVM [20]. The kernel function used in this work was the Gaussian kernel function [21].

2.2.4. Random Forest

Random forest (RF) is an algorithm from the ensemble methods, which are methods that combine different models to obtain a single result. This feature makes these algorithms more robust and complex, leading to a higher computational cost that is usually accompanied by better results [22]. During the creation of a model, different configurations of this algorithm can be tested, thus generating different models, but, at the end of the machine learning process, only the best result is used. In an ensemble method, different models are created from an algorithm, but all the results are used instead; a result is obtained for each model and combined into a single result. For instance, the result with the highest frequency is the chosen one in classification problems [23]. An RF is made up of ensembled decision trees, which establish the rules for decision making [24]. The algorithm creates a graph structure, similar to a flowchart, with nodes where a condition is verified. Depending on the decision conditions attached to each node, the flow follows through one branch or the other, always leading to the next node, until the tree ends. With the training data, the algorithm searches for the configuration and node connections that minimize the error [25]. The number of ensembled classification trees adopted in this work was 100.

2.2.5. Gaussian Processes

A Gaussian process is a collection of random variables, indexed by time or space, fully specified by its mean and covariance functions, such that every finite collection of those random variables has a multivariate normal distribution [26]. Gaussian processes use lazy learning and a measure of the similarity between points (i.e., the kernel function) to predict the value for an unseen point from training data. The prediction is not just an estimate for that point but also has uncertainty information. For multi-output predictions, multivariate Gaussian processes are used, for which the multivariate Gaussian distribution is the marginal distribution at each point [27]. The kernel (i.e., covariance) function adopted in this work was the exponential kernel function [28].

2.3. Metrics and Statistics

To compare the performance of the five selected machine learning methods, the same training and testing datasets were used to feed all of them. The outputs were classified as true positives (TPs), false positives (FPs), true negatives (TNs), and false negatives (FNs); thus, the metric used to compare the performance was the binary accuracy:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (1)$$

which represents the rate of success of each assessed predictor [29]. All selected machine learning methods were trained and validated 90 times to generate enough data for statistical comparison [30], which was accomplished by evaluating 5 statistical values: the minimum, median, mean, maximum, and standard deviation of the accuracy. A second test performed was the application of a low-pass filter, in the form of a moving average filter [31], ensuring maximum possible accuracy but allowing some time delay before indicating a failure. The same metric and statistics from the first test (i.e., no filtering) were applied to this second part, along with the evaluation of the time delay associated with the failure detection.

3. Results

After 90 runs, the statistics for the accuracy of all five machine learning methods were evaluated, as shown by the results in Table 1.

Table 1. Accuracy statistics for selected machine learning methods with no filtering (90 runs).

Method	Minimum	Mean	Median	Maximum	Standard Deviation
SLFN	0.67917	0.74440	0.74849	0.77105	0.01842
RVFL	0.76067	0.77493	0.77503	0.78577	0.00535
SVM	0.80612	0.80612	0.80612	0.80612	0.00000 [1]
RF	0.88539 [1]	0.88749 [1]	0.88746 [1]	0.88976 [1]	0.00108
GP	0.78371	0.79245	0.79293	0.80300	0.00433

[1] Best results considering accuracy maximization.

The statistics showed that the best results were achieved by the random Forest method, since its minimum accuracy, i.e., 0.88539, was greater than the second maximum accuracy, i.e., 0.806120, achieved by the support vector machines method. Nevertheless, it is possible to increase the accuracy of all methods by low-pass filtering the outputs. A brute force algorithm was used to sweep different moving average window sizes, starting with the size of one sample, incrementing it by unit steps, until it reached a maximum mean accuracy. Along with each moving average window size, there is an associated delay, which is one sampling time (i.e., 35 milliseconds) per window size unit. The results for each method, the moving average window sizes, and delays associated with them are presented in Table 2, together with the updated statistical values.

Table 2. Accuracy statistics for selected machine learning methods with low-pass filtering (90 runs).

Method	Window Size	Delay	Minimum	Mean	Median	Maximum	Standard Deviation
SLFN	3164	110.74	0.85982	0.95563	0.97318	0.99030	0.03742
RVFL	2935	102.725	0.95630	0.97431	0.97478	0.98554	0.00732
SVM	2711	94.885	0.99041	0.99041	0.99041	0.99041	0.00000 [1]
RF	827 [1]	28.945 [1]	0.98577 [1]	0.99084 [1]	0.99205 [1]	0.99238	0.00202
GP	2166	75.81	0.98565	0.99084 [1]	0.99093	0.99262 [1]	0.00118

[1] Best results considering accuracy maximization.

The results for the low-pass filtering outputs of the machine learning methods applied differed. One important thing to notice is that all filtered methods reached the maximum accuracy during the test phase at least once, if the delay caused by the moving average window is considered. The residual error is due to the samples within the interval between the failure applied and the failure recognition, which depends directly on the time delay. Figure 1 shows the filtered output of the best run (i.e., highest maximum accuracy equal to 0.99262) for the Gaussian processes method, but the waveform was the same for the outputs of all methods.

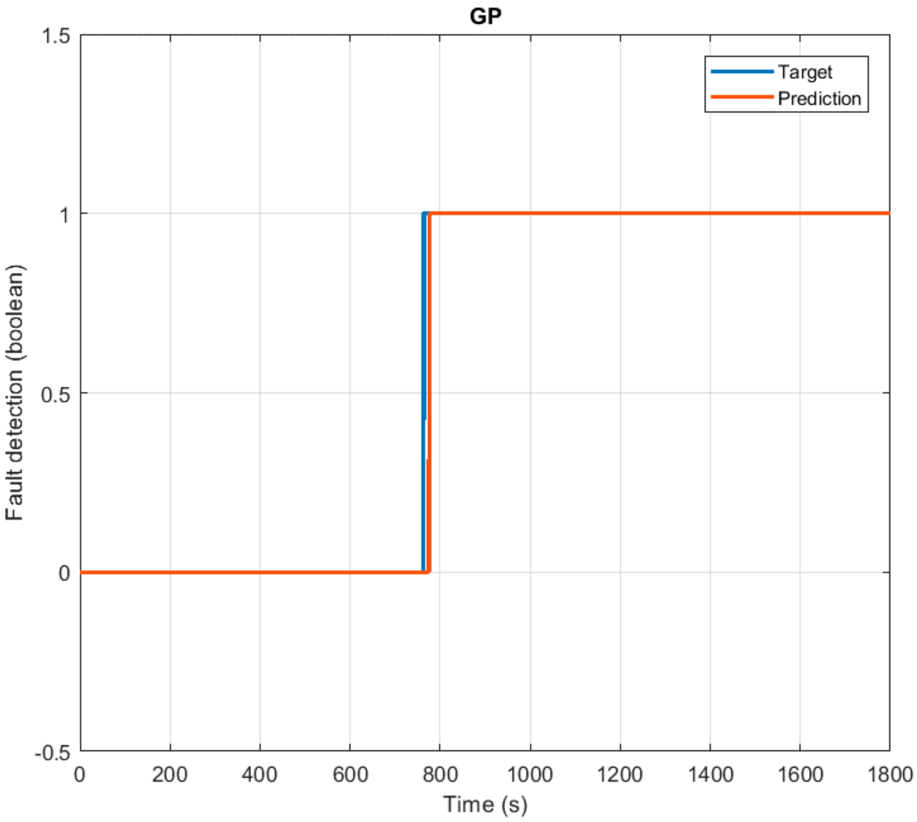

Figure 1. This is a time series of both target and Gaussian Process (GP) predictions. The target represents a fault applied to the simulated engine component (i.e., leakage in the compressor system) during a Worldwide Harmonized Light Vehicle Test Procedure (WLTP) driving cycle. The prediction identifies the fault by using a low-pass filtered output from a trained Gaussian processes method, which is accomplished after a delay caused by the filtering process.

4. Discussion

The results suggest that machine learning methods, trained with simulation data, can be used in predictive maintenance to recognize failures in automotive engine components. To increase precision, the application of low-pass filtering is necessary, leading to delays in fault detection which must be considered for each component, application, and design requirements. The computational cost is also a limiting factor for real-life applications, which may lead to unfeasibility depending on the embedded technology used on such applications. In that sense, further tests in onboard, real vehicles are necessary to validate all the methods used in this work, due to its accuracy and computational cost feasibility. Nevertheless, once it is validated, it is possible to apply the methods for different components, not limited to engine components but all vehicle components that can benefit from predictive maintenance in the form of fault diagnosis. Furthermore, the application of different approaches for fault recognition is considered for further research, such as multi-step forecasting based on mode decomposition [32–34] along with the artificial wavelet neural networks based on swarm intelligence paradigms [35].

5. Conclusions

Machine learning methods, such as random forest, support vector machines, single-layer feed-forward neural networks, random vector functional link networks, and Gaussian processes can be applied as fault predictors for predictive maintenance in automotive engine components by using generated data from simulation testbeds for fault diagnosis, whereby fault behaviors can be simulated and compared when performed in distinct driving cycles. Maximum accuracy is reachable when a moving average (i.e., low-pass) filter is applied, but a response delay must be considered before fault recognition.

Acknowledgments: The authors would like to thank the National Council of Scientific and Technologic Development of Brazil—CNPq (Grants: 307958/2019-1-PQ, 307966/2019-4-PQ, 405101/2016-3-Univ, 404659/2016-0-Univ) and Fundação Araucária (PRONEX-FA/CNPq 042/2018) for its financial support of this work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RF	Random Forest
SVM	Support Vector Machines
ANN	Artificial Neural Networks
GP	Gaussian Processes
WLTP	Worldwide Harmonized Light Vehicle Test Procedure
EUDC	Extra-Urban Driving Cycle
NEDC	New European Driving Cycle
FTP-75	United States Environmental Protection Agency Federal Test Procedure
Pa	Pascal
K	Kelvin
SLFN	Single-Layer Feed-Forward Neural Network
RVFL	Random Vector Functional Link Networks
TP	True Positive
TN	True Negative
FP	False Positive
FP	False Negative
s	Seconds

References

1. Frank, P.M. Fault detection in industrial processes. *IFAC Proc. Vol.* **1998**, *31*, 891–896, doi:10.1016/S1474-6670(17)40665-3.
2. Bode, G.; Thul, S.; Baranski, M.; Müller, D. Real-world application of machine-learning-based fault detection trained with experimental data. *Energy* **2020**, *198*, 117323, doi:10.1016/j.energy.2020.117323.
3. Abdelgayed, T.S.; Morsi, W.G.; Sidhu, T.S. Fault detection and classification based on co-training of semisupervised machine learning. *IEEE Trans. Ind. Electron.* **2018**, *65*, 1595–1605, doi:10.1109/TIE.2017.2726961.

4. Ruan, D.; Xie, H.; Song, K.; Zhang, G. Adaptive speed control based on disturbance compensation for engine-dynamometer system. *IFAC Pap.* **2019**, *52*, 642–647, doi:10.1016/j.ifacol.2019.09.102.
5. Cavalcante, I.M.; Frazzon, E.M.; Forcellini, F.A.; Ivanov, D. A supervised machine learning approach to data-driven simulation of resilient supplier selection in digital manufacturing. *Int. J. Inf. Manag.* **2019**, *49*, 86–97, doi:10.1016/j.ijinfomgt.2019.03.004.
6. Chen, Z.; Mi, C.C.; Xu, J.; Gong, X.; You, C. Energy management for a power-split plug-in hybrid electric vehicle based on dynamic programming and neural networks. *IEEE Trans. Veh. Technol.* **2014**, *63*, 1567–1580, doi:10.1109/TVT.2013.2287102.
7. Huttunen, J.M.J.; Kärkkäinen, L.; Lindholm, H. Pulse transit time estimation of aortic pulse wave velocity and blood pressure using machine learning and simulated training data. *PLoS Comput. Biol.* **2019**, *15*, e1007259, doi:10.1371/journal.pcbi.1007259.
8. Janet, J.P.; Chan, L.; Kulik, H.J. Accelerating chemical discovery with machine learning: Simulated evolution of spin crossover complexes with an artificial neural network. *J. Phys. Chem. Lett.* **2018**, *9*, 1064–1071, doi:10.1021/acs.jpclett.8b00170.
9. Li, Q.; Rajagopalan, C.; Clifford, G.D. A machine learning approach to multi-level ECG signal quality classification. *Comput. Methods Programs Biomed.* **2014**, *117*, 435–447, doi:10.1016/j.cmpb.2014.09.002.
10. Ng, K.Y.; Frisk, E.; Krysander, M.; Eriksson, L. A realistic simulation testbed of a turbocharged spark-ignited engine system: A platform for the evaluation of fault diagnosis algorithms and strategies. *IEEE Control Syst. Mag.* **2020**, *40*, 56–83, doi:10.1109/MCS.2019.2961793.
11. Nielsen, M.A. *Neural Networks and Deep Learning*, 1st ed.; Determination Press: Boston, MA, USA, 2015; pp. 1–5.
12. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*, 1st ed.; MIT Press: Boston, MA, USA, 2016; pp. 164–167.
13. Pao, Y.H.; Takefuji, Y. Functional-link net computing: Theory, system architecture, and functionalities. *IEEE Comput.* **1992**, *25*, 76–79, doi:10.1109/2.144401.
14. Pao, Y.H.; Park, G.H.; Sobajic, D.J. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing* **1994**, *6*, 163–180, doi:10.1016/0925-2312(94)90053-1.
15. Vukovi, N.; Petrovi, M.; Miljkovi, Z. A comprehensive experimental evaluation of orthogonal polynomial expanded random vector functional link neural networks for regression. *Appl. Soft Comput.* **2018**, *70*, 1083–1096, doi:10.1016/j.asoc.2017.10.010.
16. Zhang, L.; Suganthan, P.N. A comprehensive evaluation of random vector functional link networks. *Inf. Sci.* **2016**, *367–368,* 1094–1105, doi:10.1016/j.ins.2015.09.025.
17. Ren, Y.; Suganthan, P.N.; Srikanth, N.; Amaratunga, G. Random vector functional link network for short-term electricity load demand forecasting. *Inf. Sci.* **2016**, *367–368,* 1078–1093, doi:10.1016/j.ins.2015.11.039.
18. Vapnik, V.N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: New York, NY, USA, 2000; pp. 267–270.
19. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222, doi:0.1023/B:STCO.0000035301.49549.88.
20. Schölkopf, B.; Smola, A.J. *Learning with Kernels, Support Vector Machines, Regularization, Optimization and Beyond*, 1st ed.; MIT Press: Boston, MA, USA, 2002; pp. 11–15.
21. Keerthi, S.S.; Lin, C.-J. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural Comput.* **2003**, *15*, 1667–1689, doi:10.1162/089976603321891855.
22. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32, doi:10.1023/A:1010933404324.
23. Shih, Y.S. Families of splitting criteria for classification trees. *Stat. Comput.* **1999**, *9*, 309–315, doi:10.1023/A:1008920224518.
24. Meinshausen, N. Quantile regression forests. *J. Mach. Learn. Res.* **2006**, *7*, 983–999, doi:10.5555/1248547.1248582.
25. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*, 1st ed.; CRC Press: Boca Raton, FL, USA, 1984; pp. 255–259.
26. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Boston, MA, USA, 2006; pp. 37–41.
27. Bijl, H.; Wingerden, J.-W.; Verhaegen, M. Applying gaussian processes to reinforcement learning for fixed-structure controller synthesis. *IFAC Proc. Vol.* **2014**, *47*, 10391–10396, doi:10.3182/20140824-6-ZA-1003.01623.

28. Neal, R.M. *Bayesian Learning for Neural Networks*; Springer: New York, NY, USA, 1996; pp. 118–119.
29. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874, doi:10.1016/j.patrec.2005.10.010.
30. Majid, U. Research fundamentals: Study design, population, and sample size. *Undergrad. Res. In Nat. Clin. Sci. Technol. (Urncst) J.* **2018**, *2*, 1–7, doi:10.26685/urncst.16.
31. Lyandres, V.; Briskin, S. On an approach to moving-average filtering. *Signal Process.* **1993**, *34*, 163–178, doi:10.1016/0165-1684(93)90160-C.
32. Silva, R.G.; Ribeiro, M.H.D.M.; Moreno, S.R.; Mariani, V.C.; Coelho, L.S. A novel decomposition-ensemble learning framework for multi-step ahead wind energy forecasting. *Energy* **2020**, 119174, doi:10.1016/j.energy.2020.119174.
33. Ribeiro, M.H.D.M.; Mariani, V.C.; Coelho, L.S. Multi-step ahead meningitis case forecasting based on decomposition and multi-objective optimization methods. *J. Biomed. Inform.* **2020**, *111*, 103575, doi:10.1016/j.jbi.2020.103575.
34. Moreno, S.R.; Silva, R.G.; Mariani, V.C.; Coelho, L.S. Multi-step wind speed forecasting based on hybrid multi-stage decomposition model and long short-term memory neural network, *Energy Convers. Manag.* **2020**, *213*, 112869, doi:10.1016/j.enconman.2020.112869.
35. Klein, C.E.; Bittencourt, M.; Coelho, L.S. Wavenet using artificial bee colony applied to modeling of truck engine powertrain components, *Eng. Appl. Artif. Intell.* **2015**, *41*, 41–55, doi:10.1016/j.engappai.2015.01.009.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.