Floating Production Storage and Offloading Electric Power Demand Modelling using Soft Computing Techniques

Daniel C. de Araujo, Jr. * **Vitor H. Ferreira** ** **Artur A. Pessoa** *
Marcio Z. Fortes ** **Bruno S. M. C. Borba** ** **Andre A. Augusto** **
André C. Pinho ** **Angelo C. Colombini** ** **Marcos L. Ramos** ***
Gabriel R. F. Q. Mafra ***

* *Departamento de Engenharia de Produção, Universidade Federal Fluminense, RJ.*
** *Departamento de Engenharia Elétrica, Universidade Federal Fluminense, RJ.*
*** *Petrobras, RJ.*

Abstract: This work presents a case study of the application of soft computing techniques to model the load of the main electric equipments of three of Petrobras' FPSO units. The methodology proposed was used in the development of a modelling and simulation tool called FPSO Power Demand Analytics (FPDA), developed in a partnership between Universidade Federal Fluminense (UFF) and Petrobras. The applied methodology resulted in a library of models from which the median absolute error rarely exceeds the 3% mark. The median of the median absolute errors observed across platforms and test scenarios is often less than 1%. The presented results were found satisfactory by UFF and Petrobras' teams.

Keywords: FPSO; Machine Learning; Modelling; Neural Network; Artificial Intelligence.

1. INTRODUCTION

Among floating platform technologies, FPSOs (Floating Production Storage and Offloading) are very common in Brazil's Santos and Campos basins (Figueiredo, 2014). Brazil holds the largest number of FPSOs globally, with 60 units: 46 operational, 11 under construction, and 3 available (Offshore Magazine, 2021). The design of these platforms' electrical systems is conservative in nature. Adding to this, production history shows that the equipments operate under much lower load conditions than those considering during design, which creates the opportunity for expanding production at no risk of overload. In fact, an 1% increase on the production levels of a platform with a daily output of 150 thousand barrels (Petrobras, 2017) with a barrel at USD 90 (approx. BRL 450) yields an additional monthly revenue of BRL 20.25 million. Additionally, accurate predictions of load factors and intermittency can lead to savings and reduced risk of re-design needs. As such, this paper discusses some of the results obtained from a computational solution deployed at Petrobras, FPSO Power Demand Analytics (FPDA), which was developed to estimate and simulate relevant electrical equipment load factors on oil and natural gas production platforms through soft computing techniques. The methodology supports evaluating potential daily production increases for existing FPSOs and designing new FPSOs using knowledge gained from operational history analysis.

Despite the recent growth in adoption, literature on data-driven modelling using production data remains scarce. The authors of this work have not found studies proposing similar methodologies. The exception is another work by the same authors (Ferreira et al., 2022), where initial results were presented: the results for three compressors of a single FPSO unit. Here we present the results for a total of 82 electric equipment split across three FPSO units, with new data and extensive statistical metrics. Section 2 presents details of the solution and its implementation, while Section 3 presents the case study and results.

2. FPSO POWER DEMAND ANALYTICS (FPDA)

FPDA is an computational solution built on Python, leveraging its rich libraries, such as Scikit-learn (Pedregosa et al., 2011), PandaPower (Thurner et al., 2018) and Pandas (The pandas development team, 2020). Once a model is properly trained it is stored in a library of models, becoming thus available for later use in power flow simulations; being able to simulate and study existing FPSOs or even new designs.

2.1 Soft Computing Techniques

In structural terms, the relevant machine learning models can be summarized as follows. Let $\underline{x}_i \in \mathbb{R}^n$ be the vector with the i-th sample of n independent variables (inputs), and $d_i \in \mathbb{R}$ be the corresponding value of the dependent variable (desired output). ML models assume that the relationship between these quantities is given by:

$$d_i = f\left(\underline{w}, \underline{x}_i\right) + e_i \tag{1}$$

ISSN: 2175-8905 / 2177-6164

where $f(\underline{w}, \underline{x}_i) : \mathbb{R}^n \to \mathbb{R}$ represents an unknown function, but with a set of $\underline{w} \in \mathbb{R}^M$ of M parameters to be estimated; and e_i is the i-th realization of a random variable e with a probability distribution to be defined.

The difference between ML models often lies in the structure of the function $\hat{f}(\hat{\underline{w}}, \underline{x}_i)$. The techniques used vary in nature, including, for instance, simple linear combinations (Gujarati et al., 2011) sometimes aided by regularization components (Hoerl and Kennard, 1970), filters that allow the understanding of nonlinearities (He et al., 2015), specialized optimizers (Kingma and Ba, 2014), mathematical structures originally inspired by biology (Haykin, 1998), boolean tests styled in a structured form, also known as decision trees (Shalev-Shwartz and Ben-David, 2014) and even models that combine said decision trees in order to improve results (Breiman, 2001; Friedman, 2001).

Below is a list of the techniques used by FPDA:

Multivariate Regression (MVR) MVR is a traditional statistical technique for data modelling. The model output (expected value of the dependent variable) is obtained through a linear combination of independent variables, computed using estimated coefficients (weights) to optimize a specific statistical criterion. (Gujarati et al., 2011). In its traditional version, MVR uses mean squared error as an evaluation metric and employs the Ordinary Least Squares (Behrens, 1997) method for coefficient estimation.

Ridge Regression (RDG) RDG is similar to MVR, the difference being the use of a hyperparameterized regularization term through a hyperparameter α, aiming at penalizing coefficient magnitudes to avoid overfitting risk (Hoerl and Kennard, 1970).

Multilayer Perceptron (MLP) MLP is a widely used supervised learning technique for regression and classification modelling. It is built with three layers of neurons (*perceptrons*): an input layer, with a number of neurons (size) equal in number to the cardinality of the input vector, fully connected to a hidden layer of user defined size, which is in turn fully connected to the output layer, with a size equivalent to the cardinality of the desired output vector, representing the model's output. Although multiple hidden layers can be used, a MLP with a single hidden layer containing an adequate number of neurons can approximate any continuous function accurately (Haykin, 1998). Each neuron utilizes an activation function, such as the popular Rectified Linear Unit (ReLU) (He et al., 2015), to allow proper nonlinearities representation. MLPs are usually trained with optimizers such as the Adam optimizer (Kingma and Ba, 2014).

Support Vector Machine (SVM) SVMs are based on a paradigm known as statistical learning, being designed to solve problems with limited data and little or no prior knowledge, common in real-world applications (Vapnik, 1998). The were originally formulated with classification problems in mind, they were later expanded for tasks such as signal processing and regression. An interesting capability of SVMs is that in the case of regression tasks there is a tolerance range where error is not considered (Platt, 2000).

Decision Tree (DTR) DTRs are algorithms that map a sequence of boolean logical decisions (e.g. $x_i > 5$) in a tree-like structure. At the edges of the tree (*leafs*) is the output value of the trained model. Despite being developed for classification, DTRs can be used for regression as well (Shalev-Shwartz and Ben-David, 2014).

Random Forest (RFR) RFRs work by building multiple decision trees, assuming that a combination of multiple models will produce better results. It is a popular algorithm for both regression and classification tasks (Breiman, 2001).

Gradient Boosting Regression (GBR) GBRs work by building a serialized sequence of "weak learners" (simple models, typically decision trees) in such a way that the residue of the last model is the dependent variable of the next model, thus attempting to identify dynamics not understood by the previous model, improving results. (Friedman, 2001).

2.2 Data Pipeline

Figure 1 illustrates the details of the implemented data pipeline.

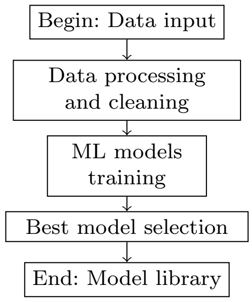

Figure 1. Detail of the methodology used to construct ML models for each electrical load

Data is preprocessed by first extracting the columns (variables) selected by the user (any number of independent variables and one dependent variable) from the dataset. The extracted columns are each cleaned of any data point (row-wise) with missing data (NaNs) and are each converted to the [-1, 1] range. Data is also automatically split between training (66%) and validation (34%). The split percentages were defined empirically by the authors with experience in machine learning and adopted as the standard.

Each model is trained in a customized two-level grid search procedure, selecting the best hyperparameters for each model and then the best model, from which the best possible option is returned. The set of hyperparameters searched are described in the appendix A.

Model performance is assessed primarily through the analysis of absolute error distribution. Since devices have different active power ranges, the *Mean Absolute Error* is taken in percentage:

$$\text{MAE\%} = 100 \times \frac{1}{n y_{max}} \sum_{i=1}^{n} |y_{pred_i} - y_{gt_i}| \qquad (2)$$

Where y_{max} represents the equipment's maximum power, y_{pred} and y_{gt} represent the prediction and ground truth vectors, respectively, and n represents the cardinality of

y_{pred} and y_{gt}: the amount of data points available to assess the models' performance. The best model selected is stored in a model library for later use, which might include simulating specific production scenarios, power flow simulations and even new designs.

Alternatively, another metric reported is the *Median Absolute Error*, also taken in percentage:

$$\text{MDAE\%} = 100 \times \frac{1}{y_{max}} Med\left(|y_{pred} - y_{gt}|\right) \quad (3)$$

3. CASE STUDY

The developed algorithms were trained and analyzed using data from three different FPSOs, referred to as P1, P2 and P3. Power demand history per device synchronized with process variables was provided. Table 1 summarizes the data of each platform, showing the total equipment and process variables, as well as the distinct sampling period for each platform. The equipments generally refer to pumping, compression or injection systems, with compressors being the most relevant for an accurate prediction.

Table 1. Time horizon of the training and validation data for each platform

	P1	P2	P3
Sampling period	1 hour	6 minutes	1 hour
Start date	03/23/21	04/13/22	06/05/22
End date	03/23/22	05/10/22	06/05/22
N. devices	35	35	36
N. variables	105	105	109
Total records per var.	8761	6481	8761

To assess the accuracy of trained models, two independent test scenarios were constructed. These scenarios aim to evaluate the FPDA algorithms' performance for two possible situations:

- Sequential data prediction (scenario 01): models are tested with data in which the beginning of its time horizon coincides with the end of the training and validation data's time horizon.
- Non-sequential data prediction (scenario 02): models are tested with data that was registered after a significant amount of time (no temporal connection).

Table 2 shows the data details for the two scenarios constructed to evaluate the models. As mentioned earlier, scenario 01 maintains the temporal continuity of the databases. For scenario 02, this continuity is lost to assess prediction quality for other future instances. Both sets provide a timespan of 168 hours (1 week) on each platform.

Table 2. Time horizon of each platform's data, for scenarios 01 and 02

	Start time	End time
P1	01:00 - 03/23/2022	23:00 - 03/30/2022
P2	01:00 - 05/10/2022	23:00 - 05/17/2022
P3	01:00 - 06/05/2022	23:00 - 06/12/2022
P1, P2, P3 (scenario 02)	01:00 - 07/15/2022	23:00 - 07/22/2022

3.1 Equipments Modelling Results

Table 3 shows the number of equipment modelled in the study. Due to the nature of machine learning techniques' data needs, not every model was able to be successfully trained. In some instances, for example, the remaining available data for the output variable was constant after cleaning (sometimes even *before* cleaning). In others, the data was missing for the length of the dataset's timespan.

Table 3. Amount of equipments modelled and assessed in the two established scenarios for the case study.

	Modelled	Equipments Amount
P1	30	35
P2	23	35
P3	29	36
Total	82	106

Table 4 shows the amount of model types chosen for each FPSO, with decision trees (34 equipment with Random Forest Regressions and 18 with Gradient Boosting Regressions) and support vector machines (30 equipments) being the most popular. These three model types have outranked the others, which were not selected. Tree-based models' high performance might be due to their ability to output discrete values (exactly zero power demand), unlike models with continuous outputs.

Table 4. Summary of selected models for each FPSO equipment

	P1	P2	P3	Total
Multilayer Perceptron	0	0	0	0
Multivariate Regression	0	0	0	0
Support-Vector Machine	5	10	15	30
Gradient Boosting Regression	10	4	4	18
Random Forest Regression	15	9	10	34
Ridge Regression	0	0	0	0
Decision Tree Regression	0	0	0	0
Total	30	23	29	82

Given that each platform has its set of equipment, and that each equipment has a MAE% and a MDAE% metric, the results are presented in summarized tables, discriminating relevant statistics, platform-wise (P1, P2 and P3) and data-wise (validation, scenario 01 and scenario 02). The statistics taken are the average, interquartile range, and several quartiles: 25th, 50th (median), 75th and 100th (maximum). Tables 5-10 show the distributions of the MAE% and MDAE% metrics, with each platform discriminated.

Table 5. Summary of MAE% for the validation set

	P1	P2	P3
Avg.	0.5249	0.6497	2.0437
IQR	0.4565	0.7181	1.6840
25%	0.1682	0.1663	0.7329
50%	0.3760	0.3172	1.4963
75%	0.6247	0.8844	2.4169
Max	3.0545	2.5718	9.5857

Table 5 summarizes the mean absolute error in percentage (MAE%) for the validation set. Results show adequate model fit, confirming algorithm choice and selection process. Model performance is confirmed in Table 6 (next week projection) and Table 7 (distant future week projection), with the third quartile's MAE% at around 10% in the worst case scenario (platform P2): all others are below the 3% mark. Attention is drawn to the clear outlier with a MAE% of 86%, likely due to missing data during training or another issue.

Table 6. Summary of MAE% for scenario 01

	P1	P2	P3
Avg.	1.2423	4.9811	1.9879
IQR	0.8196	1.6427	1.2651
25%	0.0529	0.2386	0.3134
50%	0.4193	0.9398	0.8603
75%	0.8725	1.8813	1.5785
Max	18.7205	85.1536	18.9521

Table 7. Summary of MAE% for scenario 02

	P1	P2	P3
Avg.	4.4316	11.5915	1.7343
IQR	2.5962	8.9995	2.5577
25%	0.3832	1.3996	0.2345
50%	0.8417	3.4696	0.7032
75%	2.9794	10.3991	2.7922
Max	68.5344	86.0370	7.4237

Tables 8, 9 and 10 showcase the median absolute error in percentage (MDAE%) for the validation, scenario 01 and scenario 02, respectively. Upon inspection it can be hypothesized that the median errors are generally lower than the mean errors.

Table 8. Summary of MDAE% for the validation set

	P1	P2	P3
Avg.	0.1380	0.2951	0.5493
IQR	0.2123	0.3804	0.8661
25%	0.0005	0.0159	0.0212
50%	0.0629	0.0990	0.5054
75%	0.2128	0.3963	0.8873
Max	0.6399	1.8675	1.6945

Table 9. Summary of MDAE% for scenario 01

	P1	P2	P3
Avg.	0.2893	5.2201	0.7354
IQR	0.3581	1.4583	0.7196
25%	0.0051	0.1950	0.2154
50%	0.1380	0.7015	0.4965
75%	0.3632	1.6533	0.9350
Max	2.0394	94.1706	3.5306

3.2 Subsystems modelling

The platform's structural organization arranges equipment into internally redundant subsystems for increased reliability and ease of maintenance. This justifies a subsystem modelling strategy, where a subsystem's set of independent variables are the union of its equipment's independent variables, while its dependent variable is the sum of the equipment's dependent variables. Table 11 shows the number of modelled subsystems.

Table 10. Summary of MDAE% for scenario 02

	P1	P2	P3
Avg.	4.5824	5.8623	1.2191
IQR	1.8633	2.7822	1.6848
25%	0.1161	0.0710	0.0097
50%	0.4569	1.5124	0.5536
75%	1.9794	2.8532	1.6945
Max	75.1696	86.0370	4.4373

Table 11. Modelled subsystems per platform

	Modelled	Subsystems Amount
P1	12	15
P2	13	15
P3	11	13
Total	36	43

Table 9 presents the number of models chosen for each FPSO. Once again the Support-Vector Machine, Gradient Boosting Regression and the Random Forest Regression models have outranked the others.

Table 12. Selected models for each FPSO

	P1	P2	P3	Total
Multilayer Perceptron	0	0	0	0
Multivariate Regression	0	0	0	0
Support-Vector Machine	3	6	6	15
Gradient Boosting Regression	2	2	3	7
Random Forest Regression	7	5	2	14
Ridge Regression	0	0	0	0
Decision Tree Regression	0	0	0	0
Total	12	13	11	36

Tables 13-18 show the platform-wise distributions of the MAE% and MDAE% metrics, as in section 3.1.

Table 13. MAE% summary for the validation set across the three FPSOs

	P1	P2	P3
Avg.	0.4402	0.4759	1.6949
IQR	0.3558	0.5394	1.2804
25%	0.2249	0.1341	0.9359
50%	0.4102	0.2256	1.2949
75%	0.5807	0.6735	2.2163
Max	1.0244	1.4197	3.8655

Table 14. MAE% summary for test scenario 01 across the three FPSOs

	P1	P2	P3
Avg.	0.6403	1.6402	1.2805
IQR	0.4213	1.9530	1.1527
25%	0.2582	0.3690	0.5749
50%	0.3836	0.9686	1.1273
75%	0.6795	2.3220	1.7276
Max	3.2440	6.0411	3.9559

Subsystem modelling sees a generally improved performance in the case of the average MAE%s, with the average

Table 15. MAE% summary for test scenario 02 across the three FPSOs

	P1	P2	P3
Avg.	2.0634	11.3518	2.5566
IQR	2.1476	16.9108	1.5822
25%	0.6055	1.5435	0.8441
50%	1.5601	12.0767	1.3634
75%	2.7531	18.4543	2.4263
Max	5.9722	25.1607	12.5369

Table 16. MDAE% summary for the validation set across the three FPSOs

	P1	P2	P3
Avg.	0.2077	0.2222	0.6021
IQR	0.3057	0.2531	0.6358
25%	0.0119	0.0382	0.3413
50%	0.2128	0.0899	0.4549
75%	0.3176	0.2913	0.9771
Max	0.5039	0.8982	1.3195

Table 17. MDAE% summary for test scenario 01 across the three FPSOs

	P1	P2	P3
Avg.	0.3662	1.2443	0.8620
IQR	0.2580	1.3906	1.1392
25%	0.1007	0.2671	0.2286
50%	0.2061	0.8667	0.5120
75%	0.3587	1.6577	1.3678
Max	2.1094	4.8256	2.6833

Table 18. MDAE% summary for test scenario 02 across the three FPSOs

	P1	P2	P3
Avg.	1.5745	6.1971	2.6639
IQR	1.9315	11.9417	1.3914
25%	0.2824	0.2398	0.7942
50%	0.5008	1.0204	1.1963
75%	2.2139	12.1815	2.1856
Max	6.4668	22.8319	15.6770

MAE% of platforms P1, P2 and P3 in scenario 01 dropping from 1.24%, 4.98% and 1.99% to 0.64%, 1.64% and 1.28%, respectively. There is also a significant improvement on the outliers, with P2's maximum MDAE% dropping from approximately 94 and 86 percent to 4.83 and 22.83 percent on scenarios 01 and 02. There are some counter examples, however, with the median MDAE% for platform P3 rising from 0.55% to 1.19% on scenario 02.

4. CONCLUSION

This paper presented the results of the Machine Learning algorithms used to model equipments and subsystems of FPSOs implemented on FPSO Power Demand Analytics (FPDA). The presented results were found satisfactory by both the development and the engineering teams at UFF and Petrobras. The methodology resulted in a library of models from which the median absolute error rarely exceeds the 3% mark. In fact, the median of the median absolute errors observed across platforms and test scenarios is often less than 1%.

Regarding the subsystem scheme versus the tradicional equipment scheme, one interesting dynamic which might decrease outlier likelihood on individual observations and increase overall performance on the subsystems scheme is that it leads to more information per model. Since the platforms use redundancy by design, the model of an individual equipment might be confused and not be able to tell that an equipment is turned off because *another* in the subsystem is working. To counterpoint this, the extra amount of variables could predicate the model requiring more training data and associated issues.

It should be kept in mind that the sampling size of the performance estimation of the platforms on the subsystem modelling scheme is naturally smaller than the of the equipments scheme. This might raise concerns from a statistical perspective, specially when discussing the observed performance distribution (outliers, averages, quartiles etc).

While emphasizing that both development and engineering teams found the results satisfactory, further research could be done regarding the aforementioned points. particularly in increasing the sampling size of subsystems (and equipments) by increasing the number of assessed platforms.

ACKNOWLEDGMENTS

The authors thank Universidade Federal Fluminense (UFF), Petrobras and Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP), for the opportunity and financial support of the research.

REFERENCES

Behrens, J.T. (1997). Principles and procedures of exploratory data analysis. *Psychological Methods*, 2, 131–160. doi:10.1037/1082-989X.2.2.131. Place: US Publisher: American Psychological Association.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45, 5–32. doi:10.1023/A:1010950718922.

Ferreira, V.H., Cavaliere, C.A.C., Ramos, M.L., Colombini, A.C., Augusto, A.A., Pessoa, A.A., Mafra, G.R.F.Q., Ferraz, P.E.P., De Oliveira, W.A., Borba, B.S.M.C., and D.C., A.J. (2022). FPSO Power Demand Analytics: Machine Learning and Analytics for FPSO Electric Power Systems Simulation. *Rio Oil and Gas Expo and Conference*, 22(2022), 142–143. doi:10.48072/2525-7579.rog.2022.142. URL `https://biblioteca.ibp.org.br/pt-BR/search/43772`.

Figueiredo, F.R. (2014). MARINHA DO BRASIL CENTRO DE INSTRUÇÃO ALMIRANTE GRAÇA ARANHA ESCOLA DE FORMAÇÃO DE OFICIAIS DA MARINHA MAERCANTE. 44.

Friedman, J.H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189–1232. doi:10.1214/aos/1013203451. Publisher: Institute of Mathematical Statistics.

Gujarati, D.N., Porter, D.C., Durante, D., Rosemberg, M., Rosa, M.L.G.L., Shikida, C.D., Júnior, A.F.d.A., and Salvato, M.A. (2011). *Econometria Básica.* AMGH, 5ª edição edition.

Haykin, S.O. (1998). *Neural Networks: A Comprehensive Foundation: United States Edition.* Pearson, Upper Saddle River, N.J, 2ª edição edition.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. doi:10.48550/ARXIV.1502.01852. URL `https://arxiv.org/abs/1502.01852`. Publisher: arXiv Version Number: 1.

Hoerl, A.E. and Kennard, R.W. (1970). Ridge Regression: Applications to Nonorthogonal Problems. *Technometrics*, 12(1), 69–82. doi:10.2307/1267352. URL `https://www.jstor.org/stable/1267352`. Publisher: [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality].

Kingma, D.P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. doi:10.48550/ARXIV.1412.6980. URL `https://arxiv.org/abs/1412.6980`. Publisher: arXiv Version Number: 9.

Offshore Magazine (2021). 2021 Worldwide Survey of Floating Production, Storage, and Offloading (FPSO) Units.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85), 2825–2830. URL `http://jmlr.org/papers/v12/pedregosa11a.html`.

Petrobras (2017). P-66 deixa estaleiro rumo ao campo de Lula. URL `https://petrobras.com.br/fatos-e-dados/p-66-deixa-estaleiro-rumo-ao-campo-de-lula.htm`.

Platt, J. (2000). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Adv. Large Margin Classif.*, 10.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, new edição edition.

The pandas development team (2020). pandas-dev/pandas: Pandas. doi:10.5281/zenodo.3509134. URL `https://doi.org/10.5281/zenodo.3509134`.

Thurner, L., Scheidler, A., Schafer, F., Menke, J.H., Dollichon, J., Meier, F., Meinecke, S., and Braun, M. (2018). Pandapower—An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems. *IEEE Transactions on Power Systems*, 33(6), 6510–6521. doi:10.1109/TPWRS.2018.2829021. URL `https://ieeexplore.ieee.org/document/8344496/`.

Vapnik, V.N. (1998). *Statistical Learning Theory: 2*. Wiley-Interscience, New York, 1ª edição edition.

Appendix A. GRID SEARCH PARAMETERS

Tables A.1-A.6 presents the main parameters of each model of the grid search used for each equipment or subsystem: each possible hyperparameter combination is a grid point. The set of parameters to be searched was determined iteratively (empirically), in such a manner that it did not take too long to execute the search while guaranteeing the desired level of quality.

Table A.1. Multilayer Perceptron Parameters

Hyperparameter	Values
Hidden Layer Neuron Amount	$2 \times$ Input Cardinality $+ 1$, 100
Activation	ReLU
Learning Rate	0.005, 0.001
Solver	Adam
Max Iter.	900
Batch Size	100
Tolerance	10^{-4}
N. of iterations with change < Tolerance	50

Table A.2. Support-Vector Machine Parameters

Hyperparameter	Values
Kernel	Radial Basis Function
C	0.5, 1
ϵ	0.001, 0.0001

Table A.3. Gradient Boosting Regression Parameters

Hyperparameter	Values
Number of Estimators	100, 600
Learning Rate	0.1, 0.2
Criterion	Friedman MSE

Table A.4. Random Forest Regression Parameters

Hyperparameter	Values
Number of Estimators	100, 400
Max Depth	50, 5
Criterion	Squared Error

Table A.5. Ridge Regression Parameters

Hyperparameter	Values
Alpha	0.0, 0.1, 1

Table A.6. Decision Tree Regression Parameters

Hyperparameter	Values
Max Depth	3, 5
Criterion	Abs. Error, Friedman MSE