25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

COB-2019-0221
MACHINE LEARNING APPROACH TO SIMULATE A COAL-FIRED STEAM GENERATOR: STUDY CASE OF PECEM

Natalia de Assis Brasil Weber
Mechanical Engineering Department – Federal University of Rio Grande do Sul
natalia.weber@ufrgs.br

Jonathan James Armstrong
Queen's University Belfast
jonathan.j.armstrong.br@gmail.com

Paulo Smith Schneider
Mechanical Engineering Department – Federal University of Rio Grande do Sul
pss@mecanica.ufrgs.br

Guilherme Lacerda Batista de Oliveira
EDP Energia – PECEM
guilherme.oliveira@edpenergiapecem.com.br

Abstract. *Machine learning techniques such as Artificial Neural Networks (ANN) are gaining traction due to their ability in modelling complex systems. This paper brings the development of an ANN in Python to estimate the steam generation of a real thermoelectric power plant, based on a year-long dataset. A proposed method was used to develop a robust ANN. Then, an analysis of the performance parameters MAE and RMSE showed that the model is capable of simulate the steam mass flow rate of the steam generator of the Pecem power plant with precision.*

Keywords: *Artificial neural networks, Machine Learning, Steam generator, Thermoelectric power plant operation modelling*

1. INTRODUCTION

Some of the most effective ways of improving the efficiency of large power plant operations is the use of control techniques, data analysis and continuous monitoring. This intrinsically creates another problem for the plant operator, the need to manage large quantities of data. Actual plants can provide a large quantity of operational data due to the continuous online monitoring systems linked to hundreds of sensors and data acquisition systems. The complexity of keeping track of all these parameters is no small task and thus these conditions are particularly suitable for the use of machine learning models. These systems are able to recognize patterns, create correlations and infer interrelationships in extremely large datasets (Ascione *et al.*, 2017).

One of the more simple, though nonetheless efficient, machine learning systems particularly suitable where a large domain of variables is explored are artificial neural networks, or ANNs. An ANN is a processing data system which learns the relationship between inputs and outputs by studying recorded data obtained from the original model. It is a network of computation units, called neurons, which are connected by weighted links, known as synaptic connections (or synapses), over which information is transmitted and manipulated (Ascione *et al.*, 2017).

Over the last two decades, various studies have been made to analyse the performance of neural networks in steam generators. Suresh *et al.* (2011) developed a model for a real thermoelectric plant with the aim of optimizing the operating parameters. Two scenarios were studied: with and without water feedback, presenting a R square of 0.9999 and 0.9728 respectively. This can be used to prove the ANN's ability to predict the desired output although more recent studies have shown that it is necessary to analyse more parameters that simply R square to obtain a correct picture of how well the network is running. Strušnik *et al.* (2015) also presented a study with real data of operation of an energy generator. The ANN aimed to estimate steam generation and the efficiency of the steam generator of a real plant. For this, it was necessary to develop two neural networks to model the steam generation process. At the end of the study, the authors concluded that the actual efficiency of the plant was 90.7 % and the efficiency of the model, 90.0 %.

In this context, there is a wide application of artificial neural networks in problems involving steam generators and the like. Since these studies are related to thermoelectric power plants, most of them have a control centre that allows the collection of real data, which are used for the development of studies in this area. Regarding the development of the present network, literature indicates that the ANN architecture is test-based and there are no rules for ANN creation. In all the studies, the number of intermediate layers and their neurons were tested by “trial and error“ (Ascione *et al.*, 2017).

2. MACHINE LEARNING APPROACH TO MODEL STEAM-GENERATOR

Machine Learning is the science of programming computers so they can learn from data. Machine learning algorithms build a mathematical model of sample data, known as "training data", in order to make predictions or decisions. Neural networks are one of the many supervised algorithms in the machine learning family that can be used for classification and regression. As the objective of the model of this study case is to predict a target number, such as the steam flow generated by the steam generator, we used a Neural Network algorithm with regression purpose.

2.1 Basic concepts of neural networks computing

Neural networks are information processing systems composed of simple processing elements (nodes) linked by weighted connections (Jiang *et al.*, 2010). In its simplest form, the multilayered feed-forward neural network consists of three layers of processing elements: the input, the hidden and the output layers. Haykin *et al.* (2009) defined three basic elements a neuron: synapses, summing and activation function, as schematically presented in Fig. 1. The signal is processed forward from the input to the output layer. Each node collects the output values, weighted by the connection weights, from all the nodes of the preceding layer, processes this information through an activation function and then delivers the result towards all the nodes of the successive layer. The end of this connection is given by the synapses between the neurons.

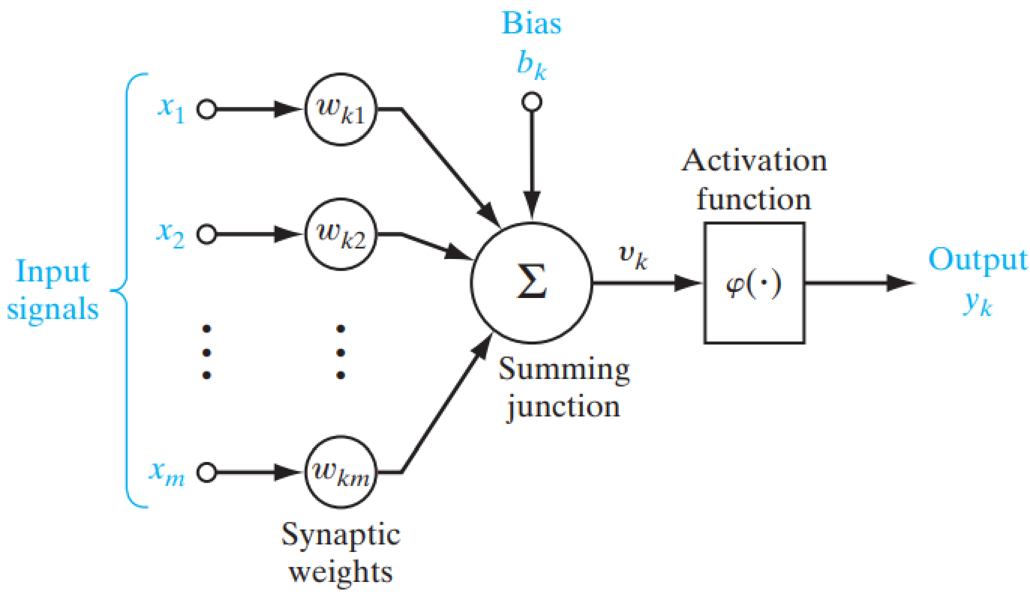

Figure 1. Artificial neuron
Source: (Haykin *et al.*, 2009)

A signal, x_j enters the synapse j, and is connected to a neuron k, which is multiplied by the synaptic weight, wk_j, (where k refers to the neuron in question and j to the input of the synapse to which the weight is linked). Unlike the weight of a biological synapse, the synaptic weight of an artificial neuron may be in a range of both negative and positive values. The adder is responsible for summing the input signals, weighted by the synaptic forces of each neuron. The activation function limits the output amplitude of a neuron (Haykin *et al.*, 2009).

It is also possible to describe mathematically the functioning of a neuron, expressed by Eq. (1) and (2).

$$u_k = \sum_{m}^{j} w_k j x_j \tag{1}$$

$$y_k = \varphi(u_k + b_k) \tag{2}$$

where the input parameters are represented by x_j and the synaptic weights of the neurons are represented by wk_j. The multiplication of these parameters results in the output of the linear combiner due to the input signals, u_k (not shown in Fig. 5). The second equation includes the neuron output signal, y_k, which is a function in the activation function φ (bias) and is defined by b_k, and u_k. It should be noted that bias is the polarizing parameter, which increases or decreases the input, the effect of applying an affine transformation to the output u_k of the linear combiner in the model of Fig. 5, as shown by Eq. (3)

$$v_k = (u_k + b_k) \tag{3}$$

2.2 Multi-layer perceptron

The perceptron network is a network model based on the layering of input and output neurons. By adding intermediate layers, the model is called multilayer perceptron (MLP). MLP's architecture contains an input layer, an output layer, and one or more intermediate layers termed hidden layers (Gevrey *et al.*, 2003). A hidden node in the MLP network is the basic computational unit that first calculates the weighted sum of all its inputs. The resulting signal is processed by a nonlinear transfer function such as the logistic or Tansig to compute the node's output. The nonlinear function's approximation ability of an MLP neural network arises from the usage of such nonlinear transfer functions for computing outputs of the hidden layer nodes. In order for the MLP neural network to capture (learn) the relationship between example data inputs and outputs, it needs to be trained so that a pre-specified error function is minimized.

This training procedure essentially aims at obtaining an optimal set of network connection weights that minimizes the error function (Gevrey *et al.*, 2003). The widely utilized error function is known as the root-mean-squared error (RMSE) and the commonly employed minimization technique is known as the error-back-propagation (EBP) algorithm.

One of the important factors, which results in large errors in the MLP-predicted outputs, is known as model over-fitting which occurs due to over-training of the network and selection of more than necessary hidden nodes (Ghugare *et al.*, 2014).

It is also necessary to determine whether the predicted output meets the accuracy requirements. If the error of the predicted output and actual output does not meet the precision requirement and does not reach the maximum training time, then it will enter the error back-propagation phase. This occurs when the error is transferred layer by layer in some form through the hidden layer to the input layer, and is apportioned to each layer neural network (Rumelhart *et al.*, 1988). The error signal of each neuron has then the right to amend the value of each neuron. Weight adjustment process is the network training process of learning. This process continuously loops until the network output error is reduced to the required accuracy or to a pre-set maximum number of times (Deshpande *et al.*, 2012).

2.3 Activation function

An activation function sets the output behaviour of each neuron in an artificial neural network. This output is then used as input for the next neuron and so on until a desired solution to the original problem is found.

Activation functions are basic architectures of artificial neural networks (ANN), since they introduce non-linear properties to the network (Haykin *et al.*, 2009). This allows the ANN to make sense of, and learn from complicated, non-linear mappings between inputs and response variables. Without these functions, nodal activation could only be a linear process, which would exponentially increase the processing power and time needed to solve problems.

While there are many variations of these functions, most network frameworks begin by computing the weighted sum of the inputs. This total net input is most often then transformed in some fashion by applying a squashing function. For example, a step function is the most basic squashing function. In this case, if the total net input is less than a certain threshold, say 0, then the output of that node is 0. If more than 0, then the output is 1. For multi-layer neural networks the main activation functions are: Logistic (sigmoid), Hyperbolic tangent, Exponential linear unit (Elu), Rectifier (Relu), Scaled Exponential Linear Unit (Selu), Rectifier (Softplus) (Chollet, 2017).

3. METHODOLOGY

The present section aims to describe a step by step sequence of methods applied to build the model to simulate the steam mass flow rate of the steam generator of the Pecem power plant.

3.1 Research object

The Pecem power plant is set to produce 720 MW of electricity and it is equipped with two coal-fired superheated steam generators (SSG), where each one designed to produce 360 MW with 1200 t/h of superheated steam at 540 °C and 18 MPa and overall plant efficiency of 35 % on average.

In order to chose the the inputs and outputs parameters an interview with the SSG operator expert was made first.

This preview was very import to guide the data acquisition, because the SSG has more than 58 measured data used by the system operator in its control.

Figure 2 shows the main parameters chosen to be the inputs and outputs of the ANN model:

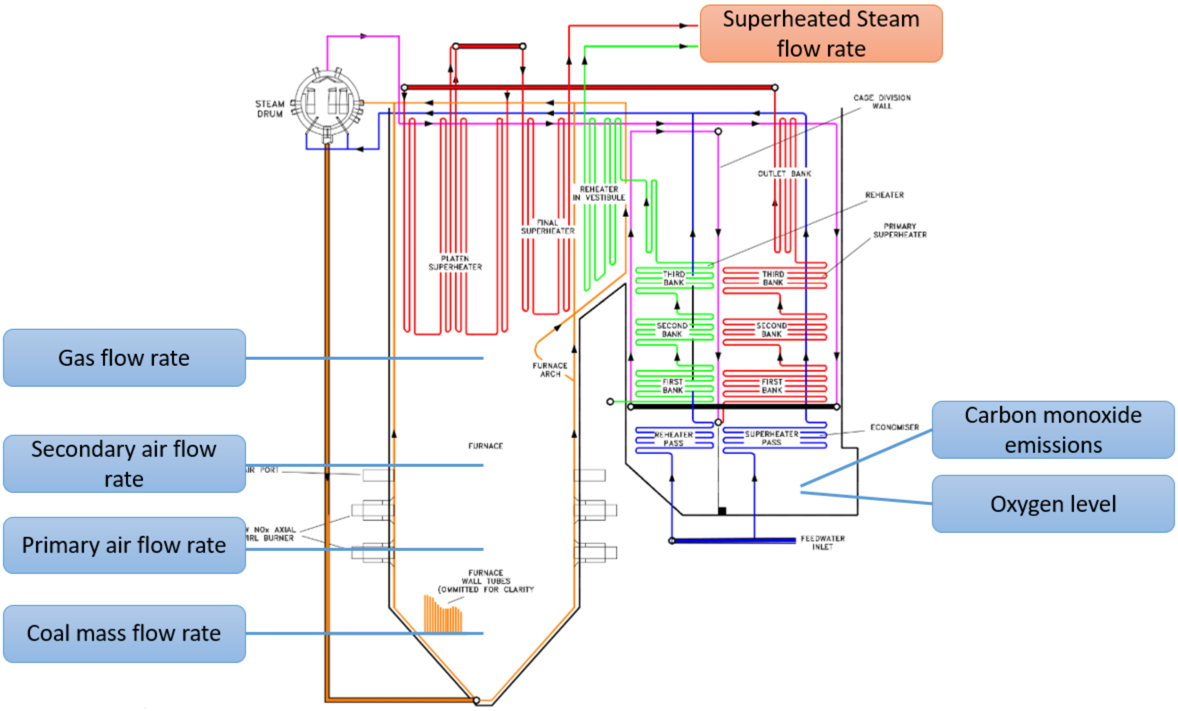

Figure 2. SSG input parameters (blue tags) and output (orange tag)
Source:personal archives.

The Tab. 1 shows the reference values for the parameters acquired for this model.

Table 1. SSG parameters description

Parameter	Measured unit	Minimun	Maximum	Mean
Steam pressure	kg/s	825.7	1203.2	1169.3
Coal mass flow rate	t/h	122.3	132.9	144.6
Primary air flow	kg/s	67.5	97.1	75.2
Secondary air flow	kg/s	192.9	276.7	240.7
CO emissions	-	4.9	632.5	201.9
Oxygen level	%	0.18	2.44	1.64
Gas flow rate	kg/s	358.5	406.0	385.9

3.2 Methods

The method applied was developed in four blocks due to the problem‘s complexity, as shown in Fig. 3. Block 1 refers to all the process that involve the data acquisition. Block 2 show all the steps that involve the data pre-training. Block 3 resume all the process that involve the model fine-tuning. Finally, the block 4 refers to the performance analysis of the RNA. After all those Blocks were finalized and evaluated carefully, the model can be deployed.

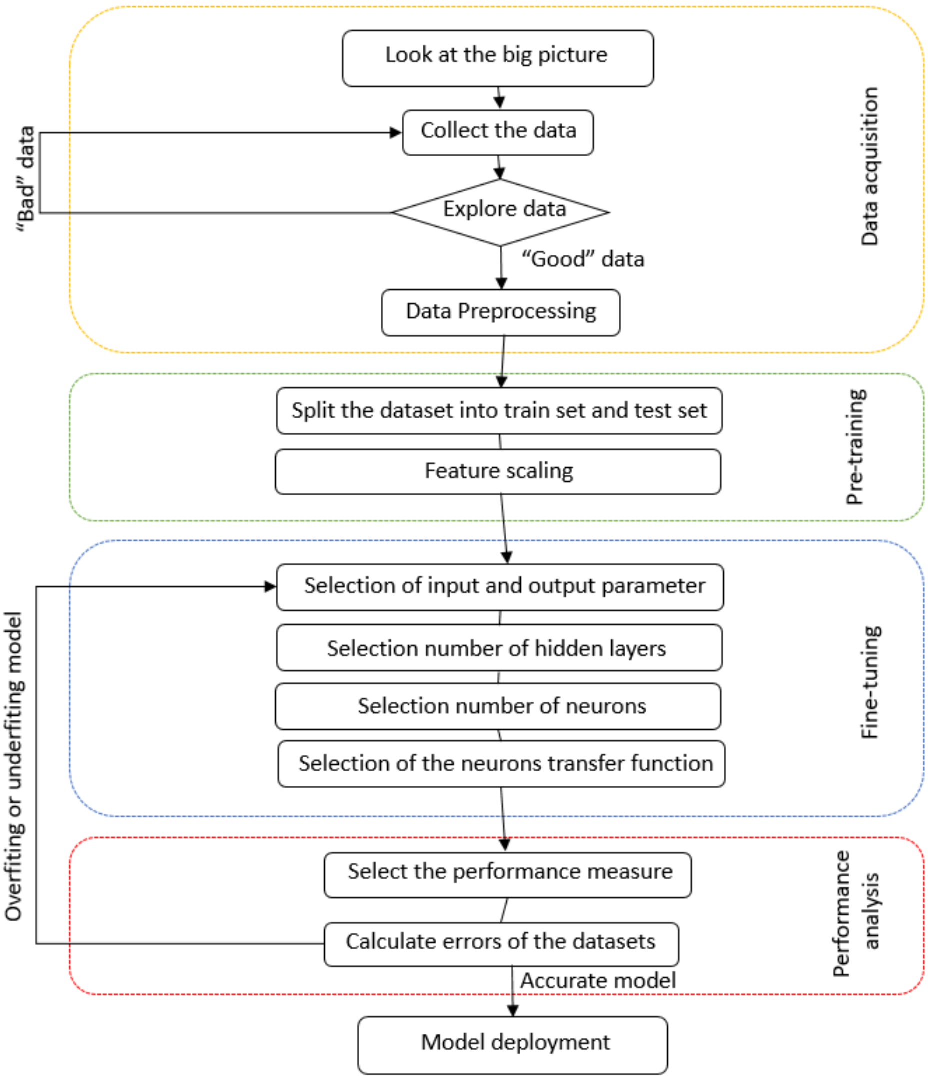

Figure 3. Flowchart of the proposed method
Source:personal archives.

3.3 Data acquisition

Composition of the data sets for training is crucial for successful development of neural models. The inputs parameters selection were made from one year long data set that represented a variety of correct (non-fault) operating states of the steam generator. Then, the data was separated into two separate power plant electric output regimes at 240 MW and 360 MW. So, the final collected data were filtered for the range of 350 MW to 360 MW. Pointing out that these data are hourly averages acquired directly from the control program of the Pecem power plant.

Subsequently, a statistical analysis of the data set were performed. Considering that not all data is normal or normal enough to treat it as being drawn from a Gaussian distribution. First, it is fundamental to check if each parameter has a normalized distribution or not. Therefore, an Anderson-Darling test was made (Scholz and Stephens, 1987). This test define the outlier removal method of the data. If the distribution of values in the sample is Gaussian or Gaussian-like, we can use the standard deviation of the sample as a cut-off for identifying outliers. Otherwise, a good statistic for summarizing a non-Gaussian distribution sample of data is the Interquartile Range, or IQR for short.

In order to maintain the data that incorporate the actual behavior of the steam generator, we opted for two outliers removal options. The first was to withdraw only those values that were well above their respective standard deviations and then withdraw values from repeated samples.

The final objective is to have a dataset that represents the real steam generator behavior so making it possible to train an ANN model with the required generalization capabilities.

3.4 Pre-training

In short, the main task to build a machine learning model is to select a learning algorithm and train it with some data. The only way to know how well a model will generalize to new cases is to actually try it out on new cases. The most recommended way to do that is to split the data into two sets: the training set and the test set. The first one is used to train the model and the other is to test the trained model with new data. The error rate on new cases tells us how well the model will perform on instances it has never seen before. It is common sense to use 80 % of the data for training and hold out 20 % for testing. Althought, Strušnik and Avsec (2015) point out that a proper selection of data representing a complete range of operating conditions can lead to the successful training of ANN with a smaller number of data sets even for a complex system.

3.5 Fine-tuning

Determining the optimal (hyperparameter) settings for each model is crucial for the robustness of the model. Hyperparameters determine how exactly an algorithm works and how they have an influence on the final outcome. On machine-learning the hyperparameters of the algorithms need to be tuned to achieve optimal performances (Duarte and Wainer, 2017; Schratz *et al.*, 2018). For hyperparameter tuning at first, n hyperparameter settings are randomly chosen from a user-defined search space. Next, they are evaluated on the chosen tuning strategy, based on these evaluations the regression model is fitted. This is continued until a termination criterion is reached.

To find an appropriate model size in this work we started as recommended by Chollet (2015) with relatively few layers and parameters, then began increasing the size of the layers or adding new layers until the validation loss was diminishing. To do so, we create a baseline model and subsequently create smaller and larger versions, and compare them. Table 2 summarizes the hyperparameters and the three different models selected to find the best model to predict the steam flow. Taking into account that the real challenge is generalization, not fitting the model (Chollet, 2015).

Table 2. Selected hyperparameters for fine-tuning and the models versions

	Neurons	Layers	Activation function
Smaller	2	3	Elu
Baseline	12	4	Relu
Bigger	24	5	Selu

3.6 Performance analysis

To judge the performance of the ANN model some metrics are necessary. The most common performance metrics used to analyse the overall predictive accuracy of an ANN are, Mean Square Error (MSE), Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). In sum, the MSE takes two arguments (y_j expected and $\hat{y_j}$ predicted output) and allows to compute a total error value over the whole dataset and the RMSE is simple the square root of MSE. This kind of measure gives an idea of how much error the system typically makes in its predictions, thus with a higher weight on for large errors. MSE and RMSE function are calculated by Eq. (4 and 5),

$$MSE = \frac{1}{n} \sum_{i} (\hat{y}_j - y_j)^2 \tag{4}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i} (\hat{y}_j - y_j)^2} \tag{5}$$

MAE, in its turn, measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight. MAE it is defined in Eq. (6):

$$MAE = \frac{1}{n} \sum_{i} (|\hat{y}_j - y_j|) \tag{6}$$

The final model is finally ready when each of the 4 blocks has achieved the desired results.

4. RESULTS

Based on the developed method the next procedures were followed in order to obtain the ANN model to predict the steam flow of the steam generator of Pecem. To obtain the data set, the first step was to obtain the real data from the Pecem power plant supervisory, for the following period: September 21st of 2017 to September 20st of 2018. The data set contained 6682 samples. Then, the outliers and values below 350 MW were withdrawn and the data stayed with 4412 samples. The following process in the data acquisition process was to remove data that were repeated. So, the final data set contained 4174 samples of data representing four seasons of the year and for the 360 MW power electric output regime. In the sequence data set was split into two sets, one with 80 % of the data to train the ANN model and the rest, 20 %, reserved to be used later as a test set.

With a robust data set the next step was to train the network. To chose the best model for our problem we have created a base model and versions of this model with different network architectures and activation function. To evaluate which one was better we used the RMSE and the MAE as indicators. The first hyperparameter tested was the number of neurons. The baseline was trained with 12 neurons, 4 layers, and with the Relu activation function. The smaller model was trained with 2 neurons, 4 layers, and with the Relu activation function. The bigger model was trained with 24 neurons, 4 layers, and with the Relu activation function. The RMSE and MAe results for each model were: 9.9 and 1148544 for the smaller model; 5.6 and 1513 for the baseline model; and, 3.6 and 580 for the bigger model, respectively.

The next hyperparameter setting to train the models was the number of layers. To set the same number of neurons it was chosen the best results of the previous test, 24 neurons. Then, all the models was trained with different numbers of layers and with with 24 neurons in each layer. So, the baseline was trained with 24 neurons, 4 layers, and with the Relu activation function. The smaller model was trained with 24 neurons, 3 layers, and with the Relu activation function. The bigger model was trained with 24 neurons, 5 layers, and with the Relu activation function. The RMSE and MAE results for each model were: 9.6 and 1148544 for the smaller model; 1.4 and 1513 for the baseline model; and, 1.2 and 580 for the bigger model, respectively.

In sequence and keeping the model configuration with lower error, three new ANN were trained with different activation functions. The baseline was trained with 24 neurons, 5 layers, and with the Relu activation function. The smaller model was trained with 24 neurons, 5 layers, and with the Elu activation function. The bigger model was trained with 24 neurons, 5 layers, and with the Selu activation function. The RMSE and MAE results for each model were: 0.9 and 190 for the smaller model; 1.2 and 497 for the baseline model; and, 1.0 and 264 for the bigger model, respectively. The results show that the smaller model presented the lowest error.

Table 3 resume the results of those indicators for each model and tested parameters.

Table 3. Results for each evaluated fine-tunig parameter and model

	Neurons		Layers		Activation	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Smaller	1304796	9.9	1148544	9.6	190	0.9
Baseline	169479	5.6	1513	1.4	497	1.2
Bigger	39482	3.6	580	1.2	264	1.0

The results showed, in the Tab. 3, that the Bigger model presented the lowest MAE and RMSE for the number of neurons in the hidden layer and for the number off layers, but the smaller model was better for the activation functions tested. So the best architecture for the ANN is with 5 layers, with 6 neurons in the input layer, 24 neurons in each hidden layers and one neuron in the output layer, Fig. 4 show the ANN final architecture.

From that network architecture the final model was trained with a multilayer feed forward perceptron neural network, with the Elu activation function, and for one hundred epochs. The results of the training and the test for each epoch of this model are shown in Fig. 5, 6.

The Tab. 4 resume the results obtained for RMSE and MAE square for the whole trained model.

Table 4. Final model evaluation performance

Data set	MAE	RMSE
Train dataset	223	1.0
Test dataset	292	1.1

Finally, it as able to plot and analyse visually the performance of the ANN model, Fig. 7 presents the real values given to the model to train (in light green) and the values the model predicted (green).

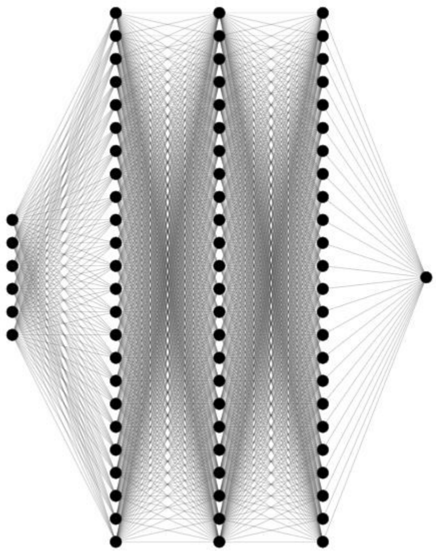

Figure 4. ANN architecture used to train the final model
Source:personal archives.

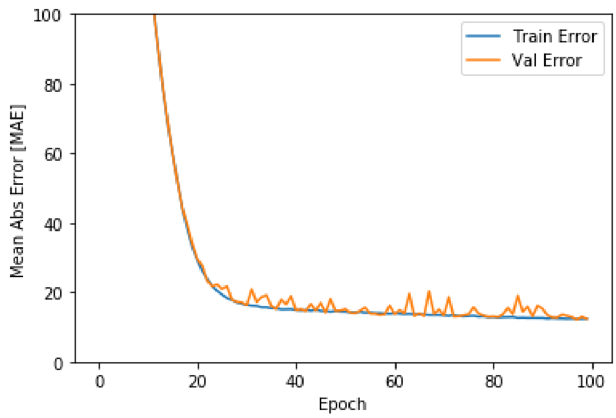

Figure 5. MAE results of the train and test set of the final ANN model
Source:personal archives.

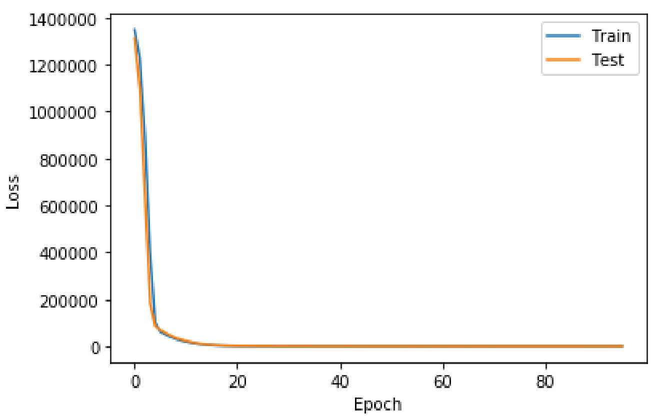

Figure 6. Loss (MSE) results of the train and test set of the final ANN model
Source:personal archives.

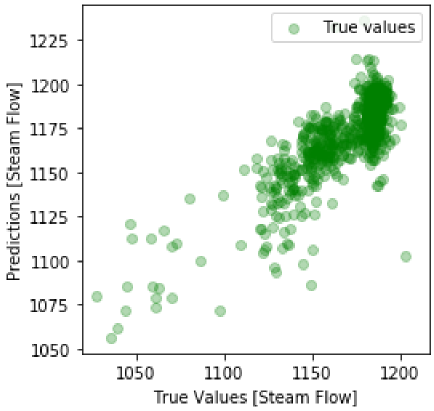

Figure 7. ANN evaluation of trained (true) and predicted values of the steam flow
Source:personal archives.

As evidenced by the performance parameters presented above the final ANN model has a robust performance and is able to replicate efficiently the workings of the steam generator. This result proves its effectiveness in estimating the steam generation, even though there were a few errors.

5. CONCLUSIONS

The aim of this study was to develop an ANN model capable of estimating the steam mass flow rate based on real data from a steam generator of a coal-fired power plant. In order to find the best MLP neural network model a different combinations of hyperparameters and architectures were set to be evaluated.

Each analysis carried out was aimed at developing a highly accurate ANN model. Between the three ANN models tested the Bigger model presented the best performance parameters. So, the final model was built with those hyperparameters: 24 neurons, 5 layers, and Elu Activation function. As a result the best RMSE and MAE values were obtained, 223 and 1.0, respectively for the train dataset, and 292 and 1.1 for the test dataset.

So, the performance parameters indicated a favourable comparison between the predicted and actual values observed in the real power plant. Which indicates that the model is capable of generalizing what it learned from the data set to new operational situations of the coal-fired steam generator in question.

6. ACKNOWLEDGEMENTS

We acknowledge Energy of Portugal EDP for the financial and technical support to this project; Smith Schneider acknowledges CNPq for his research grant (PQ305357/2013-1).

7. REFERENCES

Ascione, F., Bianco, N., De Stasio, C., Mauro, G.M. and Vanoli, G.P., 2017. "Artificial neural networks to predict energy performance and retrofit scenarios for any member of a building category: A novel approach". *Energy*, Vol. 118, pp. 999–1017.

Chollet, F., 2015. "Explore overfitting and underfitting". Tensorflow tutorials. 19 Jul. 2019 <http://www.tensorflow.org/tutorials/keras/overfit$_{and_u}nderfit/$ > .

Chollet, F., 2017. "Activation functions for artificial neural networks". Tensorflow tutorials. 19 Jul. 2019 <http://www.keras.io/activations/>.

Deshpande, P., Warke, N., Khandare, P. and Deshpande, V., 2012. "Thermal power plant analysis using artificial neural network". In *2012 Nirma University International Conference on Engineering (NUiCONE)*. IEEE, pp. 1–6.

Duarte, E. and Wainer, J., 2017. "Empirical comparison of cross-validation and internal metrics for tuning svm hyperparameters". *Pattern Recognition Letters*, Vol. 88, pp. 6–11.

Gevrey, M., Dimopoulos, I. and Lek, S., 2003. "Review and comparison of methods to study the contribution of variables in artificial neural network models". *Ecological modelling*, Vol. 160, No. 3, pp. 249–264.

Ghugare, S., Tiwary, S., Elangovan, V. and Tambe, S., 2014. "Prediction of higher heating value of solid biomass fuels using artificial intelligence formalisms". *BioEnergy Research*, Vol. 7, No. 2, pp. 681–692.

Haykin, S.S., Haykin, S.S., Haykin, S.S., Elektroingenieur, K. and Haykin, S.S., 2009. *Neural networks and learning* *machines*, Vol. 3. Pearson Upper Saddle River.

Jiang, J., Trundle, P. and Ren, J., 2010. "Medical image analysis with artificial neural networks". *Computerized Medical Imaging and Graphics*, Vol. 34, No. 8, pp. 617–631.

Rumelhart, D.E., Hinton, G.E., Williams, R.J. *et al.*, 1988. "Learning representations by back-propagating errors". *Cognitive modeling*, Vol. 5, No. 3, p. 1.

Scholz, F.W. and Stephens, M.A., 1987. "K-sample anderson–darling tests". *Journal of the American Statistical Association*, Vol. 82, No. 399, pp. 918–924.

Schratz, P., Muenchow, J., Iturritxa, E., Richter, J. and Brenning, A., 2018. "Performance evaluation and hyperparameter tuning of statistical and machine-learning models using spatial data". *arXiv preprint arXiv:1803.11266*.

Strušnik, D. and Avsec, J., 2015. "Artificial neural networking and fuzzy logic exergy controlling model of combined heat and power system in thermal power plant". *Energy*, Vol. 80, pp. 318–330.

Strušnik, D., Golob, M. and Avsec, J., 2015. "Artificial neural networking model for the prediction of high efficiency boiler steam generation and distribution". *Simulation Modelling Practice and Theory*, Vol. 57, pp. 58–70.

Suresh, M., Reddy, K. and Kolar, A.K., 2011. "Ann-ga based optimization of a high ash coal-fired supercritical power plant". *Applied energy*, Vol. 88, No. 12, pp. 4867–4873.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.