

COB-2023-0931

EVALUATING THE IMPACTS OF A NEURAL NETWORK TOPOLOGY ON ANCHOR RADIUS' ESTIMATION IN MOORING LINES

Johne Negrão Trindade

Thiago Pontin Tancredi

Universidade Federal de Santa Catarina – Rua Dona Francisca, 8300

johnetrindade@gmail.com

thiago.tancredi@ufsc.br

Bernardo Luis Rodrigues de Andrade

Universidade de São Paulo – Rua da Reitoria, 374

beluroan@usp.br

Abstract. *The anchoring system connecting the ocean bottom to the hull of offshore platforms is crucial for limiting platform offset. This study evaluates the impact of neural network topology on estimating anchor radius in mooring lines. A dataset was generated using Exmoor software, incorporating line length, wet weight, top traction, and water line depth, while calculating top angle, horizontal force, and anchor radius. The dataset analysis guided the selection of inputs and outputs for the neural network models. Fifty-four different topologies were constructed and trained separately, with mean squared error (MSE) used to evaluate accuracy. Each model was trained five times to determine the standard deviation of the MSE, totaling 270 training processes. The results revealed high precision in predicting anchor radius. Notably, the number of hidden layers and neurons did not always directly correlate with accuracy. The optimal balance was found within a middle range, exhibiting the lowest MSE. These findings demonstrate the potential of neural networks for anchor radius estimation in mooring lines. They underscore the importance of selecting appropriate network topology and achieving a balance in hidden layers and neurons for accurate predictions among with lower computational cost, contributing to the development of reliable models for enhancing the safety and efficiency of offshore operations.*

Keywords: *Applied AI, Mooring Lines, Offshore Platform, Optimization, Topology.*

1. INTRODUCTION

The anchoring system is composed by lines that connect the bottom of the ocean to a point located on the hull of an offshore platform. It is responsible for limiting the offset of the platform around the extraction role located on the seabed. Although in the past the design of mooring lines for offshore platforms operating at water depths of up to 400 meters was heavily based on empirical methods and engineer's experience, the challenging exploration of reserves located at over 3000 meters of water depth requires the application of rational design heuristics.

Mooring lines plays a crucial function, ensuring the safety of the offshore platform's operation to extract petroleum at deep water. It secures the platform against excitations caused by various sea conditions, generated by the combination of waves, current and wind (Andrade et al., 1995). Due to the advance into greater water depths, the total weight of the mooring lines becomes an issue to the design of offshore platforms. So, in order to solve this problem, these lines can be made either by uniform material over its entire length, or made by different materials each one with a different length. Usually, chains are used at the beginning and at end of the line, interconnected by some lighter material.

As specified by Tancredi (2008), artificial neural networks are computational techniques inspired by the intelligent organism's functioning, where its most important property is from an iterative training, recognize complex patterns in data, learning and enhancing its performance progressively.

According to Vellasco (2007), with the development of the artificial intelligence's field, its success to predict a value based on input variables, obtained through a known history, showed up to be a promising approach because of its capacity of recognize, connect and generalize complex patterns. It is promisor to enhance the accuracy and reliability of anchor radius estimation in offshore platforms.

As discussed by Nielsen (2015), the topology of a neural network, defined by its architecture and connectivity, significantly influences its performance and predictive capability. Understanding the impact of neural network's topology on anchor radius estimation is crucial for the development of optimized models that accurately predict the estimated anchor radius size under various environmental conditions with less computational cost.

2. DEVELOPMENT OF THE MODELS

In order to begin with the development of the models which was used in this study, it was necessary to generate a data set for training and testing purposes of all the models. As a way to create the database, the Exmoor software was used, generating a single mooring line connected to a generic platform at the ocean. It was set as environmental conditions the absence of loads due to the action of waves, wind and current. Also, ocean depth has been established as a constant of 1000 meters and the line length was set to 1500 meters, that is 1.5 times ocean depth. A set of lines with different wet weight, break capacity and materials are provided in the software, which also calculates the minimum top traction for the line based on its weight, length and ocean's depth. The top traction is calculated from the suspended portion of the mooring line, in other words, the higher the anchor radius the more the line stretches and the portion that was before supported on the bottom of the ocean, now is being supported by the platform, generating a restoration force and keeping the platform inside the anchor radius. Therefore, starting from the approximated minimum top traction calculated by the Exmoor, the top traction was varied by increasing 10 kN and stop after collecting 58 points. The output calculated with Exmoor inputs which are top traction, depth, length and wet weight of the line were top angle, horizontal force and anchor radius. Table 1 shows a sample of 36 lines created separately.

Table 1 – Sample of the original database

Top traction [kN]	Top angle [°]	Horizontal force [kN]	Anchor radius [m]	Depth [m]	Length [m]	Wet Weight [kN/m]
1030	3.4	60.4	681.6	1000	1500	0.9958
1040	3.9	70.4	697.5	1000	1500	0.9958
1050	4.4	80.4	712.3	1000	1500	0.9958
1060	4.9	90.4	726.0	1000	1500	0.9958
1070	5.4	100.5	738.9	1000	1500	0.9958
1080	5.9	110.5	751.0	1000	1500	0.9958
1090	6.3	120.5	762.5	1000	1500	0.9958
1100	6.8	130.5	773.3	1000	1500	0.9958
1110	7.3	140.5	783.6	1000	1500	0.9958
1420	9.6	235.8	830.6	1000	1500	1.2165
1430	9.9	245.8	837.0	1000	1500	1.2165
1440	10.2	255.8	843.3	1000	1500	1.2165
1450	10.6	265.9	849.4	1000	1500	1.2165
1460	10.9	275.9	855.3	1000	1500	1.2165
1470	11.2	285.9	861.0	1000	1500	1.2165
1480	11.5	295.9	866.5	1000	1500	1.2165
1490	11.8	306.0	871.9	1000	1500	1.2165
1500	12.2	316.0	877.2	1000	1500	1.2165
1905	17.6	576.4	959.0	1000	1500	1.3656
1915	17.8	586.4	962.0	1000	1500	1.3656
1925	18.1	596.5	964.9	1000	1500	1.3656
1935	18.3	606.5	967.8	1000	1500	1.3656
1945	18.5	616.5	970.6	1000	1500	1.3656
1955	18.7	626.5	973.3	1000	1500	1.3656
1965	18.9	636.6	976.1	1000	1500	1.3656
1975	19.1	646.6	978.8	1000	1500	1.3656
1985	19.3	656.6	981.4	1000	1500	1.3656
2100	16.2	586.1	939.5	1000	1500	1.5559
2110	16.4	596.1	942.4	1000	1500	1.5559
2120	16.6	606.1	945.3	1000	1500	1.5559
2130	16.8	616.2	948.1	1000	1500	1.5559
2140	17.0	626.2	950.8	1000	1500	1.5559
2150	17.2	636.2	953.6	1000	1500	1.5559
2160	17.4	646.2	956.2	1000	1500	1.5559
2170	17.6	656.3	958.9	1000	1500	1.5559
2180	17.8	666.3	961.5	1000	1500	1.5559

The total of 252 cases of platforms anchored by single lines were create, 58 lines for each wet weight. Each line of the table represents a platform with one single mooring line attached to it. From the database generated with the help of Exmoor software, it was plotted a graphic containing the axis horizontal and vertical corresponding to top traction and anchor radius respectively, while the wet weight varies between the lines, shown in Figure 1.

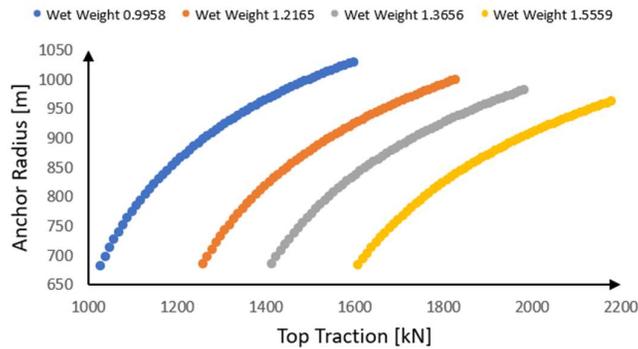


Figure 1. Variation of anchor radius in function of top traction and wet weight.

It is noted how each wet weight, given in kN/m, has a completely different shape in comparison with the other lines. This was important to evaluate the wet weight as one of the main needed characteristics that influences the anchor radius and consequently the neural network's predictive capacity. As discussed before, the top traction at the mooring line is also deeply correlated with its wet weight because the restoration force is function of the weight of the line. Furthermore, the top traction needs to sustain the weight of the suspended part of the mooring line, while the restoration force is horizontal and acts in the direction of the extraction hole, keeping the platform at a safety distance. As a result of that characteristic, the top traction and wet weight were chosen as the neural network's input, while anchor radius is the model's output.

After the create of the database to use in the training and test of all the models, it was proceeded with the generation of a table as a tool, containing the topology of each model used in this study. The table was fill in with 54 different topologies, each one corresponding to a single model, varying the number of hidden layers, number of neurons per layer, and epochs used in model's training.

Table 2 – Model's topology and training

Model	Layers	Neurons	Epochs
1	1	8	500
2	1	8	1000
3	1	8	2000
4	1	8	10000
5	1	8	50000
6	1	8	100000
7	2	8	500
8	2	8	1000
9	2	8	2000
10	2	8	10000
11	2	8	50000
12	2	8	100000
13	3	8	500
14	3	8	1000
15	3	8	2000
16	3	8	10000
17	3	8	50000
18	3	8	100000
19	1	16	500
20	1	16	1000
21	1	16	2000
22	1	16	10000
23	1	16	50000
24	1	16	100000
25	2	16	500
26	2	16	1000

27	2	16	2000
28	2	16	10000
29	2	16	50000
30	2	16	100000
31	3	16	500
32	3	16	1000
33	3	16	2000
34	3	16	10000
35	3	16	50000
36	3	16	100000
37	1	32	500
38	1	32	1000
39	1	32	2000
40	1	32	10000
41	1	32	50000
42	1	32	100000
43	2	32	500
44	2	32	1000
45	2	32	2000
46	2	32	10000
47	2	32	50000
48	2	32	100000
49	3	32	500
50	3	32	1000
51	3	32	2000
52	3	32	10000
53	3	32	50000
54	3	32	100000

As can be seen at Table 2, each line represents an architecture for the correspondent model, with its number of hidden layers varying from 1 to 2 and 3 layers, while the input and output layer aren't counted. Also, the number of neurons within each hidden layer is equal between the layers and configured with 8, 16 or 32 neurons. As two variables were chosen as input params and only one as output of the neural network, for all models, its input and output layer has two and one neuron, respectively.

In terms of training, there was 6 configurations of training, from 500 to 100,000 epochs. Each epoch represents the model has been trained through all the training set one time and it defines the number of iterations the model will accomplish with all the training set. Thus, the more is increased the number of epochs, the model is repeatedly trained with all the train data set.

3. TRAINING NEURAL NETWORKS

An algorithm was written using python to train, test and make predictions by neural networks, just using a few public libraries available, being Numpy, Pandas, TensorFlow and Sklearn. To evaluate the performance of each neural network generated, it was randomly separated the database in 80% used to train the model and 20% for its test. According Nielsen (2015), the test data is important due to its use to measure the neural network's capacity of recognize patterns after completing training.

Generally, to do this calculation is used a statistic approach (Cepowski, 2020), calculating the mean squared error (MSE) that belongs to the model, using the Eq. (1).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{pred} - y_{expt})^2, \quad (1)$$

Where the average is given in m² per point analyzed, n is the number of samples, y_{pred} is the prediction calculated by the model and y_{expt} is the real anchor radius used to train and regulate the weights of the model. It is a good choice to divide the MSE by the water depth squared in other to have a dimensionless value of reference if necessary.

Introduced by Kingma and Ba (2015), the adaptive moment estimation (ADAM) optimizer has been established as a computational efficient method and was the algorithm used for the correction of the weight involved in the neural networks. Moreover, based on previous tests, the batch size selected for all models was 5, it means the weights being

corrected per each 5 points analyzed during the model's training. An important characteristic of the process of training a neural network is that for each epoch, the set of training is shuffle, meaning that the database of training per epoch is never the same. It helps to improve the process of learning minimizing the odds of overfitting the model.

Thus, each model created to predict the anchor radius of a platform based on the inputs, which are top traction and wet weight, were trained 5 times separately using the same database than the others, shuffle during each epoch of training. In total, 270 training process were made and it's MSE calculated, with the purpose of enabling the calculate of standard deviation (STD) of each model with the Eq. (2).

$$STD = \sqrt{\frac{\sum(X_{actual} - X_{mean})^2}{n}}, \quad (2)$$

The standard deviation calculated is a metric of the set's dispersion given in meters and it shows how the sequence used to train the model per epoch influences in its final result and accuracy. As the MSE case, it is possible to obtain a dimensionless value dividing the STD by the water depth.

3.1 Input data

To maximize the learning and predictive accuracy of the developed neural network, it was made a treatment in the input data. Given that the wet weight's unity is near to 1 while top traction can fluctuate from 1030 to 2180, it is recommended to process the input data. Normalize a neural network's input has shown up to significantly increase the accuracy of the final model. Equation (3) was used in the process of normalizing all the top traction set before training each model.

$$X_{norm} = \frac{(X_{actual} - X_{mean})}{STD}, \quad (3)$$

3.2 Activation function

Previous tests were coordinated to select a single activation function that fits the most to the actual problem, improving the accuracy with the less need of computational cost. A total of 36 models were made in this previous study, with special attention to the comparison of the models with the activations function Tanh, ReLU and Softplus, respectively shown in Figures 2, 3 and 4. Arrows were plotted to identify the wet weight of each mooring line analyzed in kN/m.

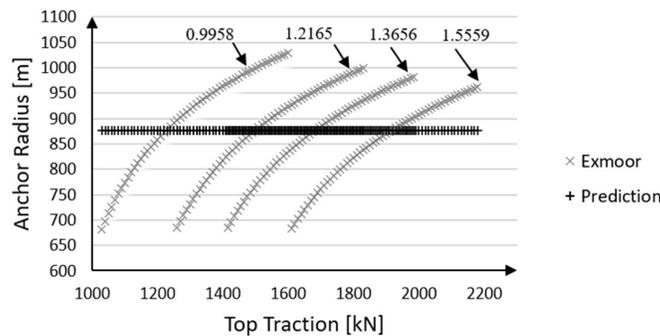


Figure 2. Activation function Tanh and 2000 epochs of training

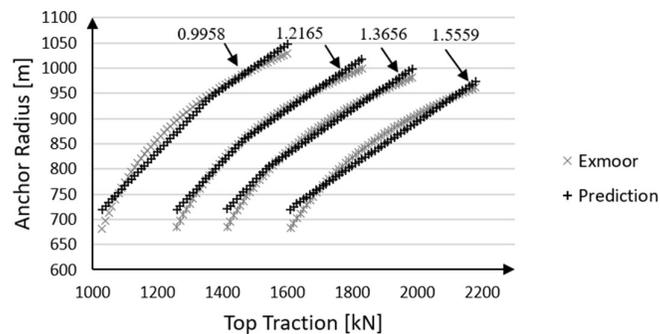


Figure 3. Activation function ReLU and 2000 epochs of training

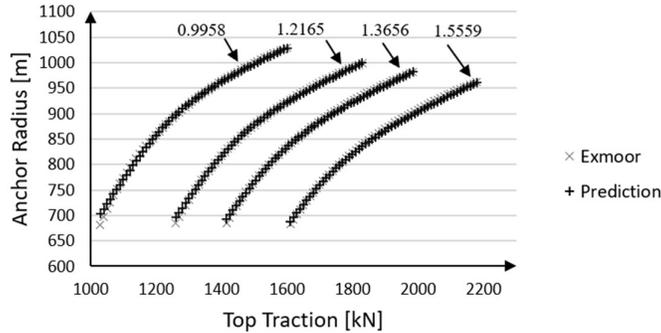


Figure 4. Activation function Softplus and 2000 epochs of training

The labels at the right portion of Figure 2, Figure 3 and Figure 4 shows the symbols used to represent the database obtained from Exmoor software and the predictions made from the neural network with the specified activation function such as Tanh, ReLU and Softplus, using as input top traction and wet weight. Moreover, each model had only one activation function used for all hidden layers. The neural network's topology was the same to all three examples, shown at Table 2 as model 27, containing 2 hidden layers, 16 neurons in each hidden layer and 2000 epochs of training. The results of MSE to the activation functions Tanh, ReLU and SoftPlus were 7083.24 m², 219.91 m² and 5.72 m² respectively.

It is important to remember that the order of the set used to train a neural network plays a crucial role in the final result, so it's also necessary to repeat the process of training of each model in order to have a statistical evaluation. Therefore, this previous study was used only to guide the selection of a first activation function to conduct a deep analysis and as a result of preliminary studies, Softplus activation function was selected.

4. RESULTS

The results shown in Table 3 were calculated based on the Eq. (1) using the database's sample of 20% that was not used to train the models. Thus, the calculations of mean squared error, shown as results, it is free of points that already has its value known by the neural network, which could miss evaluate the model's performance.

Table 3 – MSE Results and analysis for each model

Model	Iteration					Analysis		Process time [Minutes]
	MSE 1	MSE 2	MSE 3	MSE 4	MSE 5	Mean	STD	
1	13937.995	13926.301	13979.084	13933.236	13977.784	13950.880	22.807	1
2	11487.841	11660.695	11813.275	11663.476	11821.809	11689.419	122.474	1
3	6486.069	6329.650	6757.140	6805.042	6667.236	6609.028	177.154	1
4	252.853	250.604	251.988	250.745	250.756	251.389	0.887	6
5	252.267	252.098	252.220	253.692	258.944	253.844	2.616	30
6	249.337	251.995	252.614	252.751	254.127	252.165	1.576	59
7	9686.546	5891.020	8028.649	9450.169	7806.618	8172.600	1362.680	1
8	323.801	1225.571	354.742	312.390	296.934	502.688	361.938	1
9	11.256	312.614	160.821	305.295	285.552	215.108	115.850	1
10	8.543	14.739	1.105	0.317	1.934	5.328	5.538	6
11	0.398	1.287	0.106	0.185	0.365	0.468	0.423	33
12	0.104	245.544	1.085	235.595	0.482	96.562	117.624	65
13	389.867	895.746	726.444	402.190	406.824	564.214	208.639	1
14	333.668	364.766	325.266	338.293	360.760	344.551	15.498	1
15	382.810	358.442	353.174	359.450	322.080	355.191	19.452	2
16	14.821	2.758	0.654	0.936	1.702	4.174	5.373	6
17	0.294	4.092	0.120	1.943	0.482	1.386	1.500	35
18	11.918	0.050	0.666	0.237	0.050	2.584	4.672	71
19	13192.154	13195.839	13343.254	13193.430	12994.643	13183.864	110.918	1
20	9727.926	9868.447	9848.015	9959.635	9885.919	9857.988	75.150	1
21	2459.511	2549.386	2411.075	2292.792	2273.824	2397.318	103.304	1

22	254.736	253.747	252.992	255.371	245.976	252.564	3.394	6
23	256.728	253.220	256.126	254.591	253.712	254.875	1.354	30
24	257.708	251.545	251.459	253.143	252.123	253.195	2.335	59
25	3677.718	4681.294	3144.924	2024.315	1332.921	2972.234	1186.112	1
26	311.047	313.556	240.343	321.320	317.781	300.809	30.438	1
27	5.722	2.421	3.010	2.606	329.273	68.606	130.339	1
28	0.157	0.088	0.250	0.268	258.137	51.780	103.178	7
29	2.843	0.051	0.589	0.116	0.132	0.746	1.066	32
30	0.025	0.038	0.050	0.012	0.253	0.076	0.090	66
31	321.419	361.604	354.530	337.296	307.047	336.379	20.252	1
32	346.297	352.943	341.037	388.780	333.928	352.597	19.137	1
33	373.864	383.537	326.681	328.551	462.718	375.070	49.519	1
34	0.179	0.306	0.415	0.465	0.297	0.332	0.100	7
35	0.127	0.468	0.249	0.273	11.380	2.499	4.442	36
36	0.115	0.282	0.065	1.575	0.328	0.473	0.560	71
37	12081.216	12060.230	12159.282	12068.139	12087.312	12091.236	35.330	1
38	7482.555	7356.976	7190.476	7395.086	7178.807	7320.780	118.438	1
39	261.551	264.600	266.938	264.550	264.747	264.477	1.716	1
40	253.407	257.002	253.846	262.352	254.014	256.124	3.364	6
41	257.499	255.747	256.609	260.090	262.157	258.421	2.367	30
42	254.653	254.402	252.818	251.346	254.413	253.526	1.271	61
43	413.278	342.452	385.900	370.760	352.145	372.907	25.147	1
44	324.902	341.833	334.868	349.531	320.725	334.372	10.598	1
45	303.529	282.327	329.085	326.069	318.923	311.986	17.263	1
46	1.260	0.092	3.097	0.183	0.219	0.970	1.146	6
47	0.193	0.015	0.222	0.045	0.028	0.101	0.088	36
48	0.215	0.426	0.054	0.041	0.057	0.159	0.148	68
49	361.557	358.021	337.383	341.028	474.334	374.464	50.802	1
50	322.675	379.996	288.507	341.160	365.827	339.633	32.308	1
51	25.454	321.122	361.170	1.923	2.467	142.427	162.968	1
52	0.975	2.858	22.762	0.203	0.759	5.511	8.671	7
53	0.108	1.709	11.785	1.079	0.096	2.956	4.457	37
54	0.048	0.035	1.413	0.018	0.045	0.312	0.551	75

As can be seen in Table 3, in order to have a statistic evaluation of the model's performance, each model was trained independently for 5 times and its mean squared error calculated accordingly through the model and the iteration number. By the right side of the Table 3 it is shown the approximately mean value in minutes that was taken to train and evaluate each model.

It was calculated the mean of the five iteration's mean squared error to have a precisely report of the neural network's performance and its dependence with the training dataset sequence.

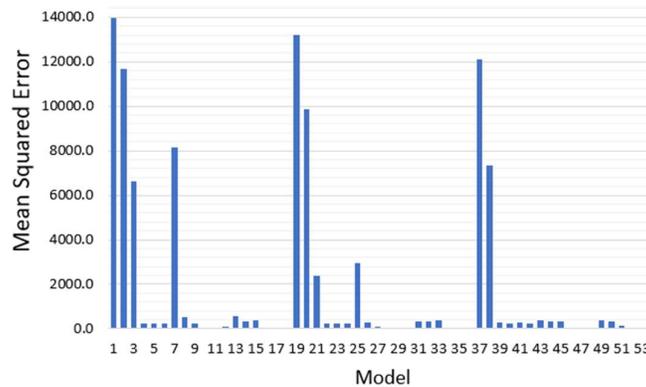


Figure 5. Mean of each model's MSE

The topology of the models presented in the horizontal axis at Figure 5 and Figure 6 are in accordance with each model's number and topology presented at Table 2. Along with this result it is notable the directly dependency between the number of epochs used to train each model and the accurately of its results, minimizing the variability of the model's performance in each training and enhancing the reliability into the success of the final model developed. As the neural network's performance increases, so do the computational cost, as can be seen in Figure 9, that represents the time of training by model.

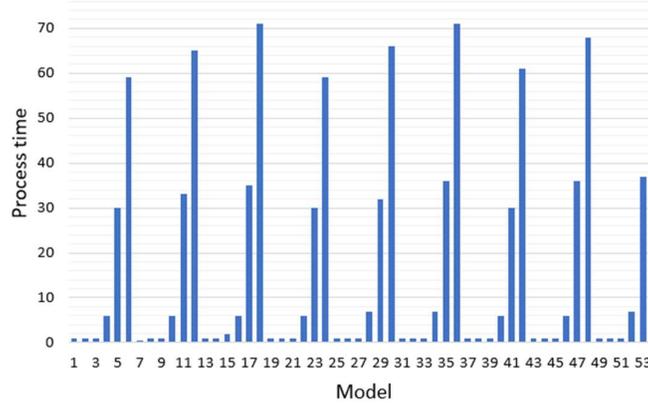


Figure 6. Process time in minutes for each model training

The standard deviation of models 7, 8 and 25 were too high in comparison with the orders models. It can be interpreted as the dependence of the models with its training dataset and the sequence of the inputs, seeing that all three models were trained with 500 epochs, which is the lowest number of training epochs used in this study.

4.1 Model's efficiency

According to Tancredi (2008), multi-objective optimization problems can lead to several good solutions, where none of which is better than the other. The pair plot depicted in Figure 7 represents the mean squared error (MSE) of the models as a function of their processing time. This curve showcases the correlations between two conflicting objectives and aids in focusing on the efficient models.

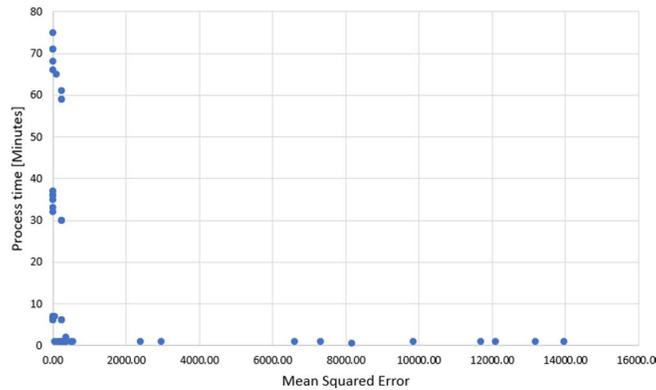


Figure 7. Process time in function of MSE

It is notable the similar behavior with a Pareto curve, by plotting the MSE against the processing time for each model, Figure 7 provides valuable insights into the trade-off between prediction accuracy and computational cost. The curve highlights the models that achieve a good balance between these two objectives, when increasing the model's accuracy, it is sacrificed the process time, and with lower computational cost, it results in less reliable networks. Besides, with the pair plot shown in Figure 7, it is possible to choose models that demonstrate superior predictive accuracy while also being computationally efficient. In Figure 8, it is shown the models with process time of under 8 minutes and a maximum MSE of 600. Indeed, the observation of models with the same training time but different mean squared error values is a notable aspect in the analysis.

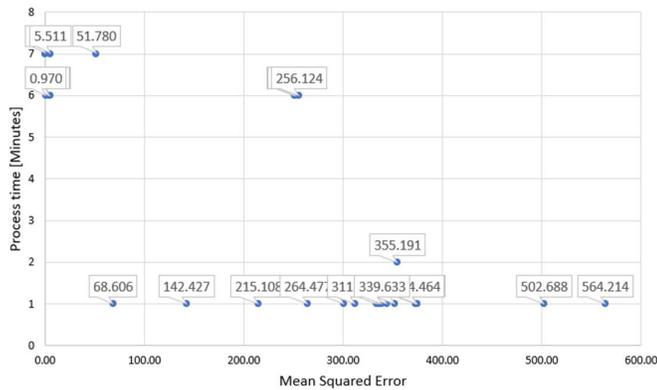


Figure 8. Limiting models to lower numbers of MSE and process time

5. CONCLUSION

It is notable the increase of neural network's accuracy based on the increase from 1 to 2 hidden layers. However, the addition of a third hidden layer did not always lead to a better performance in comparison to the models with 2, actually, in some cases it became worse. In order to evaluate this, Figure 9 and Figure 10 shows the comparisons of the model's performance with the increase of hidden layers and neurons with a fixed value of epochs in each column.

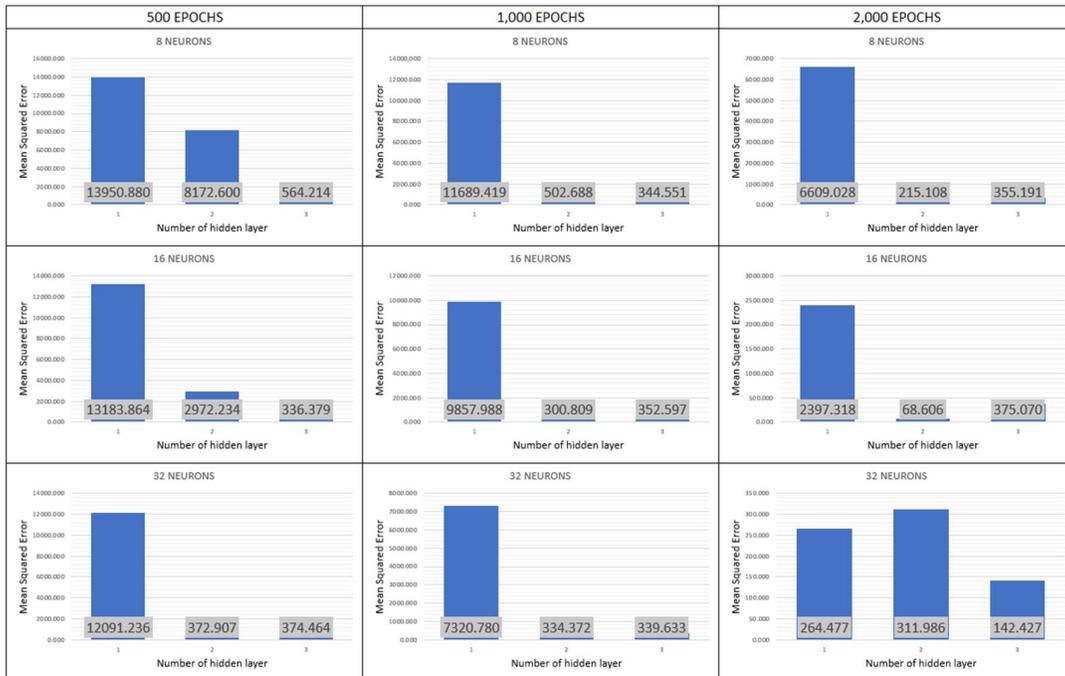


Figure 9. Results of MSE for 500 to 2,000 epochs of training

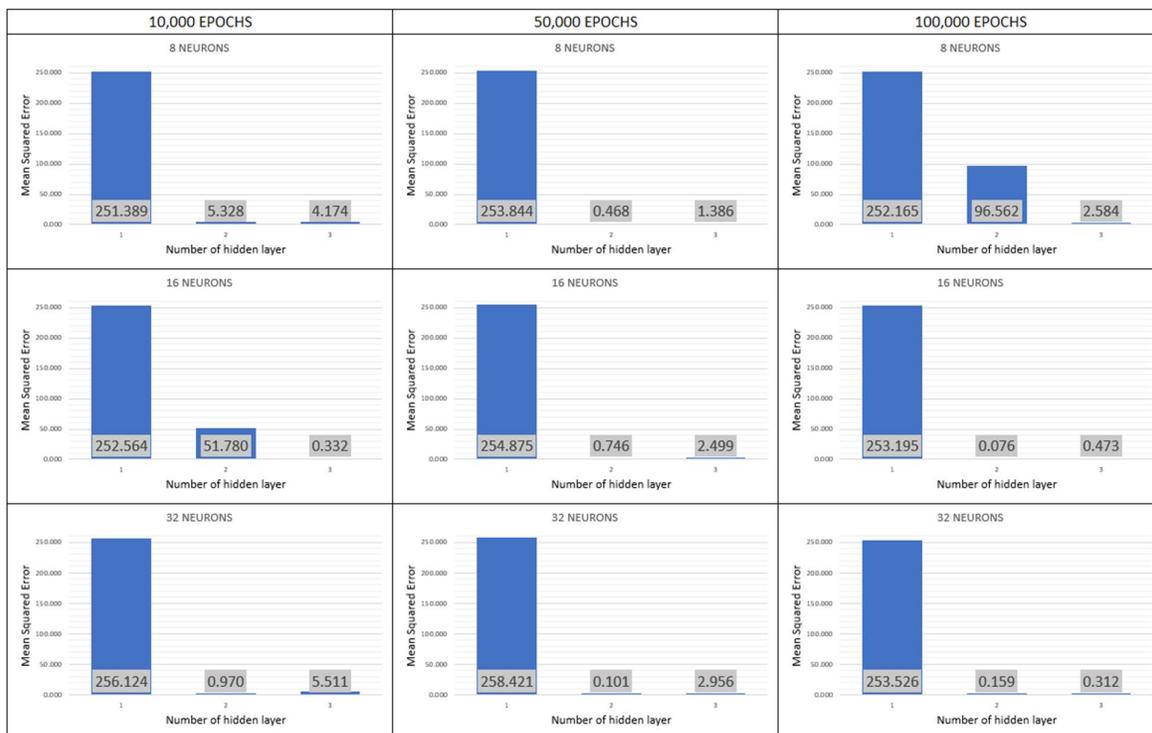


Figure 10. Results of MSE for 10,000 to 100,000 epochs of training

In this comparison it is observed that the model 30 with 2 hidden layers and 16 neurons in each, had a better performance in comparison with the model 48 with also 2 hidden layers and 32 neurons in each. As of 10,000 epochs, the increase of training in models with just 1 hidden layer had almost no impact in model's performance.

The analysis of this study's results shows that not always the increase of the hidden layers number and neurons led to a higher accuracy in the prediction of a mooring line's anchor radius. Thus, the better performance calculated, as determined by the lowest MSE, was at an intermediate point in terms of number of hidden layers and neurons.

6. ACKNOWLEDGEMENTS

We thank Petr leo Brasileiro S.A. (Petrobras), for providing the software Exmoor enabling the generate of a reliably database and support this work.

7. REFERENCES

- Andrade, B. L. R.; Brinati, H. L.; Morishita, H. M., 1995. *Anchoring and positioning system* (in Portuguese). Monography (Specialization in Naval and Oceanic Engineering) – Polytechnic School, University of S o Paulo, S o Paulo, Brasil.
- Cepowski, T., 2020. *The prediction of ship added resistance at the preliminary design stage by the use of an artificial neural network*. Ocean Engineering, Vol. 195.
- Kingma, D.P. and Ba, J.L., 2015. "Adam: A method for stochastic optimization". In *3rd International conference on learning representations – ICLR 2015*. San Diego, CA, USA.
- Nielsen, M., 2015. *Neural Networks and Deep Learning*. <https://static.latexstudio.net/article/2018/0912/neuralnetworksanddeeplearning.pdf>. Accessed 10 jun. 2023.
- Tancredi, T. P., 2008. *Distributed multidisciplinary optimization applied to engineering projects* (in Portuguese). thesis (Doctorate degree in Naval and Oceanic Engineering) – Polytechnic School, University of S o Paulo, S o Paulo, Brasil.
- Vellasco, M. M. B. R., 2007. *Artificial neural networks* (in Portuguese). *Laborat rio de Intelig ncia Computacional Aplicada*. Brasil.

8. RESPONSIBILITY NOTICE

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