Frigate Speed Estimation Using CODLAG Propulsion System Parameters and Multilayer Perceptron

Procjena brzine fregate pomoću parametara CODLAG pogonskog sustava i višeslojnog perceptrona

Sandi Baressi Šegota
University of Rijeka
Faculty of Engineering
e-mail: sbaressisegota@riteh.hr

Ivan Lorencin
University of Rijeka
Faculty of Engineering
e-mail: ilorencin@riteh.hr

Jelena Musulin
University of Rijeka
Faculty of Engineering
e-mail: jmusulin@riteh.hr

Daniel Štifanić
University of Rijeka
Faculty of Engineering
e-mail: dstifanic@riteh.hr

Zlatan Car
University of Rijeka
Faculty of Engineering
e-mail: car@riteh.hr

DOI 10.17818/NM/2020/2.4
UDK 004.89:629.5
Preliminary communication / *Prethodno priopćenje*
Paper accepted / *Rukopis primljen:* 13. 12. 2019.

Summary

Authors present a Multilayer Perceptron (MLP) artificial neural network (ANN) method for the purpose of estimating a speed of a frigate using a combined diesel-electric and gas (CODLAG) propulsion system. Dataset used is publicly available, as condition-based maintenance of naval propulsion plants dataset, out of which GT Compressor decay state coefficient and GT Turbine decay state coefficient are unused, while 15 features are used as input and ship speed is used as dataset output. Data set consists of 11934 data points out of which 8950 (75%) are used as a training set and 2984 (25%) are used as a testing set. 26880 MLPs, with 8960 different parameter combinations are trained using a grid search algorithm, quality of each solution being estimated with coefficient of determination (R2) and mean absolute error (MAE). Results show that a high-quality estimation can be made using an MLP, with best result having an error of just 3.4485×10^{-5} knots (absolute error of 0.00014% of the range). This result was achieved with a MLP with three hidden layers containing 100 neurons each, logistic activation function, LBFGS solver, constant learning rate of 0.1 and no L2 regularization.

KEY WORDS

artificial intelligence
artificial neural networks
CODLAG propulsion system
multilayer perceptron
speed estimation

Sažetak

Autori predstavljaju metodu višeslojnog perceptrona (MLP) umjetne neuronske mreže (ANN) u svrhu procjene brzine fregate koristeći se kombiniranim dizel-električnim i plinskim pogonskim sustavom (CODLAG). Korišteni skup podataka javno je dostupan, poput skupa podataka o održavanju po stanju brodskih pogonskih postrojenja, od kojih se ne koriste koeficijent raspadanja GT kompresora i koeficijent stanja raspada GT Turbine, dok se 15 značajki koriste kao ulazni, a brzina broda kao izlazni podatak. Skup podataka sastoji se od 11934 podatkovne točke, od čega se 8950 (75 %) koristi kao skup za vježbu, a 2984 (25 %) kao testni skup. 26880 MLP-ova s 8960 različitih kombinacija parametara uvježbava se algoritmom pretraživanja energetske mreže, a kvaliteta svakog rješenja procjenjuje se koeficijentom određivanja (R2) i prosječnom apsolutnom pogreškom (MAE). Rezultati pokazuju da se visokokvalitetna procjena može napraviti uz pomoć MLP-a, pri čemu će najbolji rezultat imati pogrešku od samo $3,4485 \times 10^{-5}$ čvorova (apsolutna pogreška raspona 0,00014 %). Za postizanje rezultata korišten je MLP s tri skrivena sloja koji sadrže po 100 neurona, logistička funkcija aktivacije, LBFGS rješavač, konstantna brzina učenja od 0,1 bez L2 regulacija.

KLJUČNE RIJEČI

umjetna inteligencija
umjetne neuronske mreže
CODLAG pogonski sustav
višeslojni perceptron
procjena brzine

1. INTRODUCTION / *Uvod*

Nowadays, a significant increase of energy efficiency requirements for the worldwide fleet can be noticed. One of approaches which offer a significant increase in ships energy efficiency is to utilize proper method for ship speed estimation [1].

The standard approach to estimation is to use mathematical models for simulating the ship performances [2] or to use Kalman filtering [3]. In some studies researchers used propulsion system parameters to estimate the ship propulsion performances. The authors of research presented in [4] have proposed the solution for marine diesel engine performances estimation. In this research, the vessel speed is used for the estimation of effective power, fuel consumption and emission of marine diesel engine. In [5], the estimation of exhaust emission of various ocean-going vessels based on the ship speed is performed. From the presented papers, it can be seen that such estimations are based on mathematical models and simulations that are often complex and time-consuming. As an alternative approach to ship speed estimation, machine learning (ML) algorithms are imposed [6] [7]. ML is one of the most propulsive branches of computer science today. More recent research studies have presented the possibility of ML utilization in various fields of science and technology, ranging from medicine [8], through robotics [9] [10] and computer vision [11], to power plants [12]. It can be noticed that high regression performances are achieved if artificial neural networks (ANN) are utilized for solving problems in the fields of power plants and propulsion systems [13] [14]. It is shown that when multilayer perceptron (MLP) is utilized for solving similar problems it offers the most stabile regression performances [15] [16].

From literature overview, it can be noticed that the majority of modern ship propulsion systems are based on various types of diesel engines [17] [18]. The data obtained from the aforementioned propulsion systems is then used for design of various numerical models that are used in simulations of diesel engines. Such simulations are performed with the aim of engines optimization [19], emissions reduction [20] [21], analysis of operational parameters [22], etc.

On the other hand, in some parts of shipping industry such as the transport of liquefied natural gas (LNG) steam propulsions systems still represent a dominant choice [23] [24]. It can be noticed that such propulsion systems are based on steam turbines [25] [26] and other components [27] which are necessary for proper performances of such propulsion system. However, the use of diesel engines in this part of the industry is rapidly increasing [28].

Together with diesel and steam propulsion systems, there are various new propulsion systems under development. The aforementioned systems are based on the combination of one or more diesel engines, steam and gas turbine [29] [30]. One of these complex propulsion systems is combined with the diesel-electric and gas (GODLAG) propulsion systems. Such propulsion system is based on the combination of electrical motor and gas turbine. Electrical power for electrical motors is produced by using diesel generators [31]. From the presented literature overview, some questions could be asked:

- is it possible to estimate the ship speed by using CODLAG propulsion system parameters,
- is MLP a suitable regression algorithm for ship speed estimation and
- which configuration of MLP hyperparameters achieves the best regression performances?

In this paper, an MLP-based method for vessel speed estimation is presented. A frigate with CODLAG propulsion system is analyzed and its speed is estimated by using propulsion system parameters. Dataset used in this research is publicly available as a part of UCI Machine Learning Repository.

2. DATASET DESCRIPTION / *Opis skupa podataka*

As mentioned in previous section, the dataset used in this research is publicly available and published online. The authors of aforementioned dataset have proposed the machine learning-based solutions for condition-based maintenance of CODLAG propulsion system component, its compressor and gas turbine [32 - 34]. As presented in [32], the dataset is created by using data obtained from numerical simulations of naval CODLAG propulsion system. It can be noticed that such propulsion system contains one gas turbine (GT) in combination with two electrical motors powered by diesel generators, as shown in Figure 1.

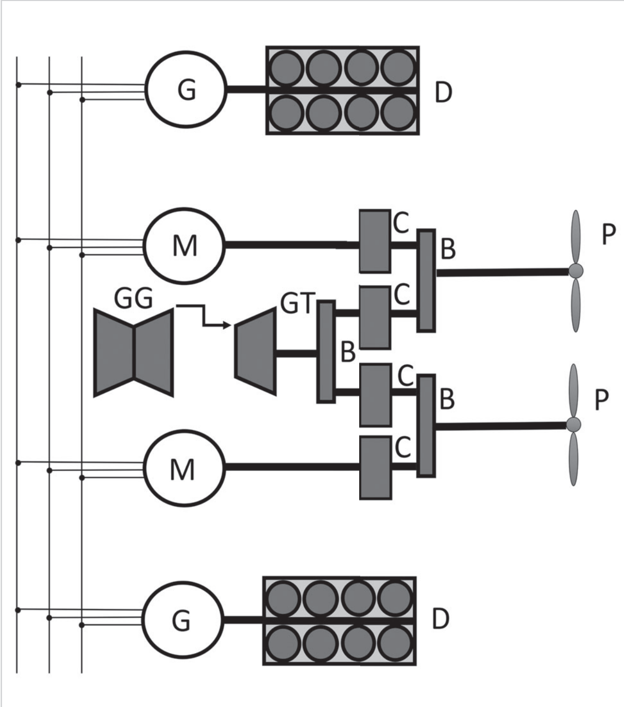


Figure 1 Scheme of Frigate CODLAG propulsion system (GT – gas turbine; M – electrical motor; G – electrical generator; D – diesel engine; B – gearbox; C – clutch; P – Frigate propeller, GG – gas generator)

Slika 1. Shema CODLAG pogonskog sustava fregate (GT - plinska turbina; M - električni motor; G - električni generator; D - dizelski motor; B - mjenjač; C - spojka; P – propeler fregate, GG - plinski generator)

Power generated with GT and electrical motors is transmitted to frigate propellers through the system that contains three gearboxes and four clutches. GT used in this propulsion system is based on two-shaft design, as shown in Figure 2.

This GT design is based on one compressor and two GTs: high pressure (HP) one and low pressure (LP) one. HP turbine is used only for compressor drive, while the LP is used for ship propulsion. Connection of HP and LP is achieved by the flue gasses only. In such configurations, HP together with combustion chamber and compressor can also be called "gas generator" [35]. CODLAG propulsion system parameters and their ranges, that are used for frigate speed estimation are presented in Table 1 together with frigate speed.

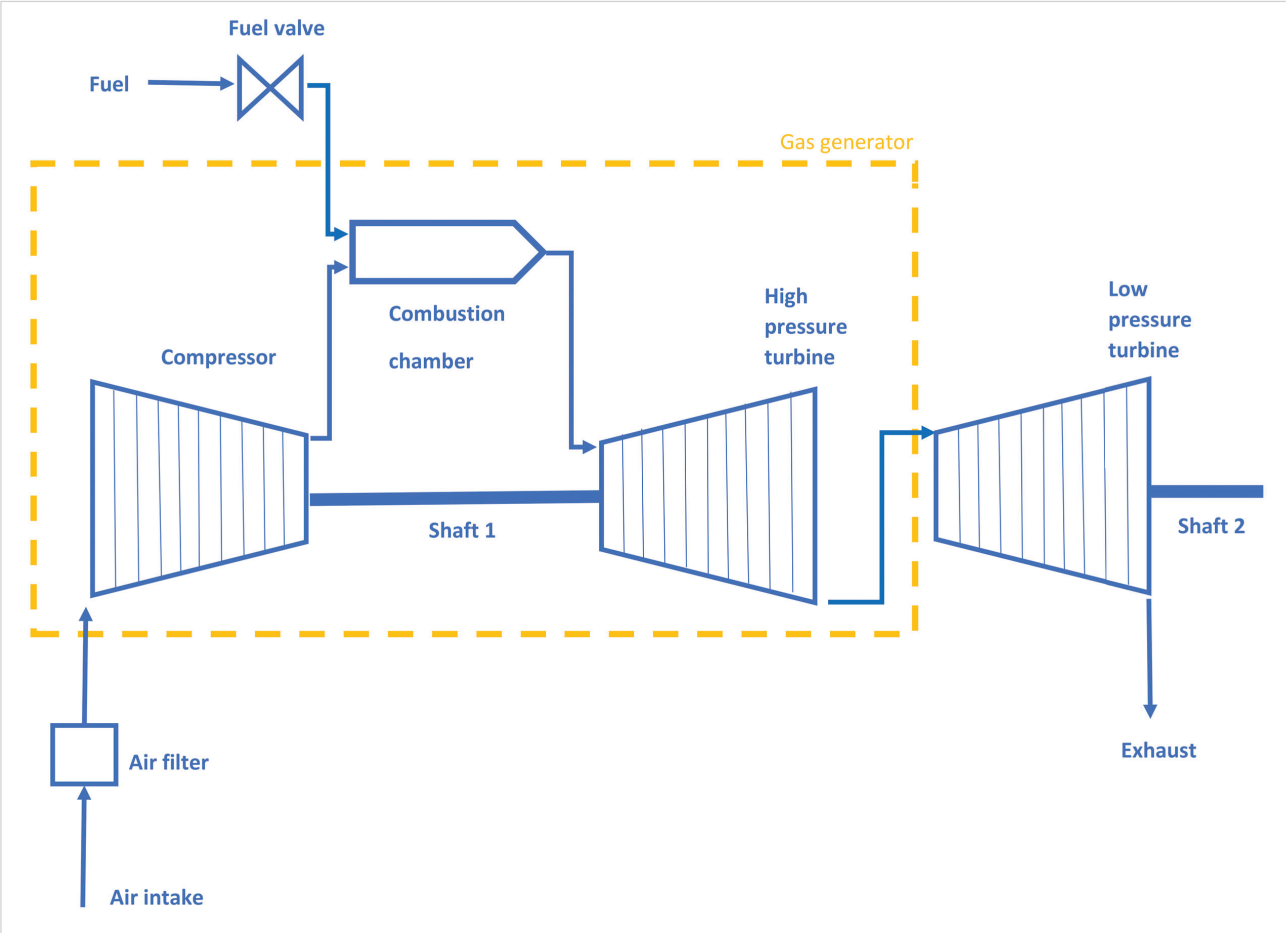


Figure 2 Scheme of gas turbine used in frigate CODLAG propulsion system
Slika 2. Shema plinske turbine korištene u CODLAG pogonskom sustavu fregate

Table 1 Dataset description
Tablica 1. Opis skupa podataka

Input parameters		
Parameter	Range	Unit
Lever position	1.138 - 9.3	-
LP turbine shaft torque (LPT)	253.547 - 72784.872	kNm
LP turbine rate of revolutions (LPn)	1308 - 3561	rpm
Gas generator rate of revolutions (GGn)	6589 - 9797	rpm
Starboard propeller torque (Ts)	5.304 - 645.249	kNm
Port propeller torque (Tp)	5.304 - 645.249	kNm
HP turbine exit temperature (T48)	442.364 - 1115.797	K
GT compressor inlet air temperature (T1)	288	K
GT compressor outlet air temperature (T2)	540.442 - 789.094	K
HP turbine exit pressure (P48)	1.093 - 4.56	bar
GT compressor inlet air pressure (P1)	0.998	bar
GT compressor outlet air pressure (P2)	5.828 - 23.14	bar
LP turbine exit pressure (Pexh)	1.019 - 1.052	bar
Combustion chamber injection control (CCIC)	0 - 92.556	%
Fuel flow (mf)	0.068 - 1.832	kg/s
Output parameter		
Parameter	Range	Unit
Ship speed (v)	3 - 27	kn

3. METHODOLOGY / *Metodologija*

3.1. Multilayer perceptron / *Višeslojni perceptron*

MLP is a type of a feed-forward artificial neural network, which can be used for regression or classification tasks [36]. In this paper MLP is used as a regressor, in an attempt to determine the value of ship speed from different variables. MLP consists of multiple layers, each containing neurons. Neurons can be thought of as a node that provides the weighted sum of the inputs. Each input of a neuron is the output of a neuron in a previous layer, multiplied with the weight of their connection. With that, each neuron will have a vector of inputs X and Θ per [37]

$$X = [x_1\ \ x_2\ \ x_3 ...\ x_n], \qquad (1)$$

and

$$\Theta_0 = [\theta_1\ \ \theta_2\ \ \theta_3 ...\ \theta_n] \qquad (2)$$

where n is the number of inputs of a given neuron. With the above, the output variable of a neuron in a neural network can be calculated using

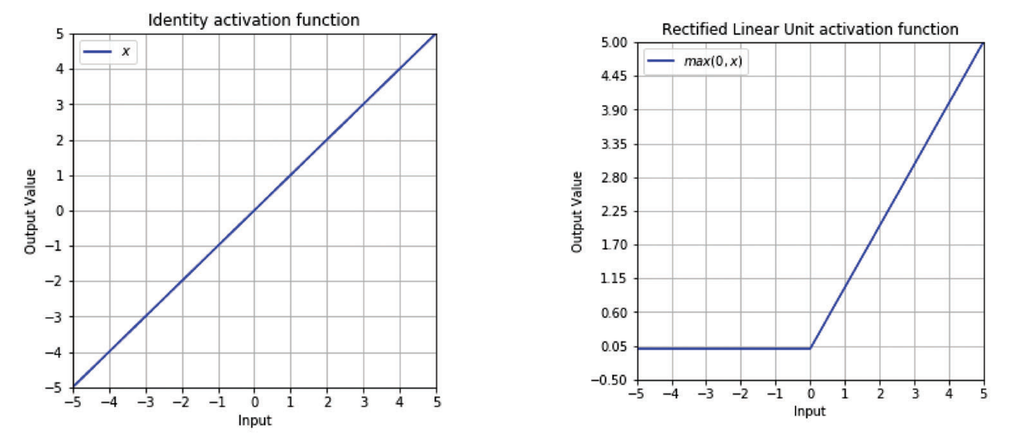
$$f(X_k) = F\,(X_k \cdot \Theta) = F\,(x_1 \cdot \theta_1 + x_2 \cdot \theta_2 + ... + x_n \cdot \theta_n) \qquad (3)$$

where $F\,(X_k \cdot \Theta)$ represents the activation function of a neuron. Activation functions are functions that transform the weighted sum of neurons inputs [38]. They are commonly used for mapping the values of a neuron to a given range (e.g. tanh or sigmoid activation functions), or to eliminate certain, unwanted values (e.g. ReLU activation function). Activation functions used in our research are: identity which maps the input directly to output ($F\,(x) = x$), ReLU which maps positive values directly, but turns negative values to zero ($F\,(x) = max\,(0, x)$), logistic activation function which maps input values into a range of [0,1] ($\mathcal{F}(x) = s = \frac{1}{1+e^{-x}}$) and Tanh which maps input values to a range of [-1,1] ($\mathcal{F}(x) = a = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$) [39, 40]. Used activation functions are shown in Figure 3.

Each MLP consists of at least three layers of neurons. First is the input layer which consists of as many neurons as there are input variables, and values of those neurons are the input variables of the data point that is being propagated through the neural network. Final layer is the output layer which, for a regressor, consists of a single neuron. The value of that neuron is the output of the neural network for the values of the input neurons. Each MLP consists of at least one or more hidden layers. The number of hidden layers and neurons inside them greatly affects the results achieved by the neural network [41]. An example of the neural network can be seen in Figure 4.

Before using MLP to regress the speed from input variables neural network needs to be trained. Training of the neural network is done using a larger part of the dataset (in this instance 75%) – from now on referred to as the "training set". Training the neural network consists of two parts – forwards propagation and backwards propagation. Before explaining these processes it is necessary to define the vector Y as

$$Y = [\hat{y}_1\ \ \hat{y}_2\ \ \hat{y}_3 ...\ \hat{y}_m]. \qquad (4)$$



(a) Plot of identity (linear) function

(b) Plot of rectified linear unit function

(c) Plot of hyperbolic tangent function

(d) Plot of logistic (sigmoid) function

Figure 3 Activation Functions used: (a) identity function, (b) relu function, (c) tanh function and (d) logistic function

Slika 3. Korištene funkcije aktivacije: (a) identitetna funkcija, (b) relu funkcija, (c) tanh funkcija i (d) logistička funkcija

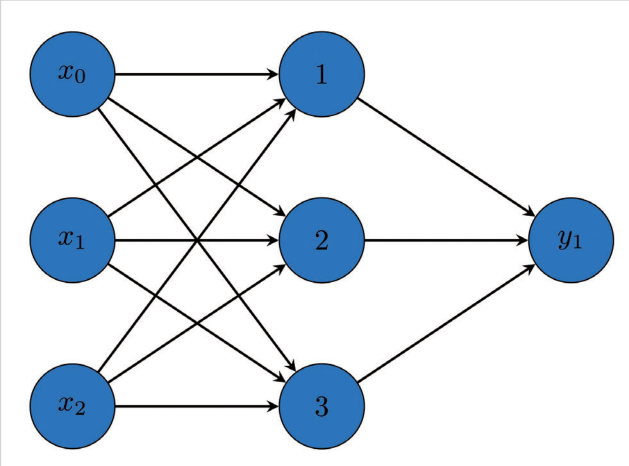


Figure 4 An example of an MLP ANN with three inputs, single output and a single hidden layer with three neurons.
Slika 4. Primjer MLP ANN-a s tri ulaza, jednim izlazom i jednim skrivenim slojem s tri neurona.

Where $\hat{y}_k$ is the result of the neural network for the k-th set of input. During the forward propagation, values of each data point in the training set are placed as values of the input layer of the neural network. At the beginning the vector *Θ* containing weights of all connections is set to random values, and the input values are propagated forward trough the MLP, which allows us to calculate the value $\hat{y}_k$ This value is then compared to the real value of the output for this dataset y_k. In our research this means that the value y_k represents the real, measured speed of the ship obtained from the dataset, while the $\hat{y}_k$ is an attempt of the neural network algorithm at correctly calculating that value given the inputs of the k-th data point. It can be expected that there will be a difference between the real and predicted value. This difference is caused by the ill-defined weights of connections between the neurons [37, 42]. To achieve a smaller error weights need to be adjusted, which is done during the backpropagation process. For the backpropagation process it is necessary to define the cost function *J(Θ)*, usually mean square error (MSE) given as [43]

$$J(\Theta) = MSE = \frac{\sum_{i=0}^{n} (y_k - \hat{y}_k)^2}{2n}. \tag{5}$$

With this value calculated the gradient of a connection is calculated as a partial derivation [44]

$$\Theta' = \frac{\partial J(\Theta)}{\partial \Theta}. \tag{6}$$

Once the gradient is calculated the existing weights need to be adjusted, depending on its value. Due to the usage of the derivation higher error will mean a greater change, while smaller error will cause smaller change in weights. New values of the weights are calculated per

$$\Theta_{new} = \Theta_{old} - \frac{\alpha}{m} \cdot \Theta', \tag{7}$$

where *m* is the number of data points in the dataset, and *α* is the learning rate [37, 45]. Higher learning rate will allow the neural network to learn faster, but if too high the backpropagation could fail to converge towards the error of zero. With a multitude of data points MLP is trained over many iterations and its weights are adjusted. The quality of the achieved result greatly depends on the parameters of neural network (so called hyperparameters).

3.2. Hyperparameter adjustment / *Podešavanje hiperparametra*

In the presented research the hyperparameters are adjusted by utilizing a grid search algorithm. Grid search takes multiple values for each hyperparameter being adjusted and trains the neural network for each of the combinations [46]. Hyperparameters that we adjusted during the training were [47]:

- Hidden Layers – the number of hidden layers and number of neurons in each layer. Presented with a tuple, in which each layer is presented with a positive integer equaling number of neurons in that layer,
- Activation function – the activation function of the neurons in the MLP,
- Solver – algorithm that will be used for calculating the weight values during backpropagation,
- α – initial learning rate for backpropagation. This is the value that adjusts the initial speed of ANN weight adjustments. Higher value of the learning rate will cause the ANN to converge to a solution faster, as weights will be changed more quickly, but setting it too high can cause the ANN to diverge instead of converge as it will skip over the weights that will cause the convergence to a solution. Setting the learning rate too low will cause the neural network too converge extremely slowly – or, more problematically, not to converge at all due to the runtime (number of iterations) being too high for realistic applications [48].
- α Adjustment – adjustment of learning rates through the training. Constant keeps the learning rate the same as starting throughout training, while adaptive and inverse scaling adjust it depending on the weight gradient value, and
- L2 – regularization parameter, larger value of which penalizes the affect of variables that have a larger influence [49, 50].

Hyperparameters that are adjusted and their possible values can be seen in the Table 2.

Table 2 Adjusted hyperparameters and their values
Tablica 2. Podešavani hiperparametri i njihove vrijednosti

Hyperparameter	Possible Values	Number
Hidden Layers	(16), (32), (50), (100), (16,16), (32,32), (16,32,16), (32,32,32), (16,16,16,16), (16,32,32,16), (32,32,32,32), (32,50,50,32), (100,100,100), (100,100,100,100)	14
Activation Function	Identity, ReLU, Tanh, Logistic	4
Solver	Adam, LBFGS	2
α	0.5, 0.1, 0.01, 0.00001	4
α Adjustment	constant, adaptive, inverse scaling	4
L2	0.1, 0.01, 0.001, 0.0001, 0.0	5

Hyperparameter choices, including solver, learning rate and others, are usually randomized, but since this adds to complexity by increasing the search space the possible values are selected based on authors previous experience with similar problems in which similar hyperparameter choices and combinations provided good results [8,11,48].

Total combinations of parameter combinations in grid search are 8960. Additionally, each combination of parameters is run 3 times, which is done to avoid poor results stemming from poorly chosen starting random connection weights. With this process a total of 26880 networks are trained. Each solution is then evaluated using two metrics.

3.3. Solution evaluation / *Procjena rješenja*

Every trained neural network is evaluated on the remaining part of the dataset (henceforward referred to as the training set). Each datapoint of the training set is used as an input, and the result that is obtained from the MLP is compared to the actual data to evaluate the quality of the regression for that network. No backpropagation is performed, as the goal is just to test the performance of the ANN, as opposed to adjusting its weights [51].

Two metrics used to compare the predicted solution vector $\hat{Y}$ and the real result vector Y are coefficient of determination (R^2) and mean absolute error (*MAE*).

Mean average error calculates the difference for each element of the two solution vectors, measures their absolute error per each element. It can be calculated using [52,53]

$$MAE = \frac{1}{n}\sum_{i}^{n} |y_i - \hat{y}_i|. \tag{8}$$

R^2 , or coefficient of determination is a value, in range [0,1] that defines how well is the variance of the real results represented by the predicted results. Higher values represent a better fitting solution. R2 can be calculated as a factor of total and residual sum of squares as [54, 55]

$$R^2 = 1 - \frac{S_{RESIDUAL}}{S_{TOTAL}} = 1 - \frac{\sum_i^m (y_i - \hat{y}_i)^2}{\sum_i^m (y_i - \bar{y})^2}, \tag{9}$$

Where $\bar{y}$ is the mean of the data results calculated per

$$\bar{y} = \frac{1}{m}\sum_{i}^{m} y_i. \tag{10}$$

Each solution is first evaluated using R^2. Coefficient of determination is given in range of [0,1], with higher values being better [56]. If the coefficient of determination value is higher than 0.99 the model is stored and evaluated again using MAE where a smaller MAE is considered better.

4. RESULTS / *Rezultati*

Out of the total 26880 solutions 7373 solutions have satisfied the condition of R^2 value larger than 0.99. Out of these 7373 solutions 4751 solutions have achieved MAE that is less than 0.026 which is 1% of the total range of speeds contained within the dataset. The worst (largest) achieved MAE is 0.67 knots or 2.81%. Best achieved solution has R^2 of 1 and MAE of 3.4485×10^{-5} knots (absolute error of 0.00014% of the range). Hyperparameters of the MLP that achieved these results are shown in the Table 3.

Table 3 Hyperparameters of the best found MLP with the MAE of 3.4485×10^{-5}.
Tablica 3. Hiperparametri najbolje pronađenog MLP-a s MAE-om od 3.4485×10^{-5}.

HIDDEN LAYER SIZES	ACTIVATION	SOLVER	α ADJUSTMENT	α	L2
(100, 100, 100)	logistic	lbfgs	constant	0.1	0

The graphs given below show how many of the best 7373 solutions each parameter has. Figure 5. shows that 5876 of the solutions that satisfied the given condition used lbfgs solver, while 1497 used the adam solver. Figure 6. shows that relu activation was used by 3523 of the networks, tanh was used by 1282, logistic by 1121 and identity activation by 1447. The scaling type for the learning rate was distributed in a way where 2552 had a constant learning rate, 2410 used adaptive and 2411 used inverse scaling learning rate, which is shown in Figure 7. Initial learning rates were distributed in a way that 1470 models had learning rate of 0.5, 1546 had a learning rate of 0.1, 1574 had a learning rate of 1574 and 2783 had a learning rate of 0.00001. This is shown in Figure 8. When it comes to L2 regularization parameter 1590 solutions used the value of 0.1, 1514 had used a value of 0.01, 1528 the value of 0.001, 1501 the value of 0.0001 and 1240 solutions used the value of 0, and this distribution is shown in Figure 9. Finally, when it comes to hidden layers most solutions had chosen a single hidden layer with the number of neurons equaling 16 (1460 solutions) and 32 (1754 solutions). There were no solutions that used the hidden layer configuration of (16,16,16,16) , (32,32,32,32) and (100,100,100,100). Full distribution is shown in Figure 10.

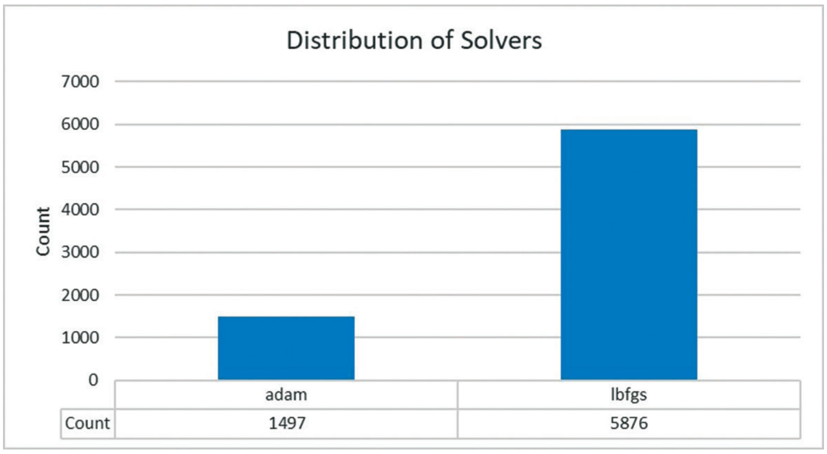


Figure 5 Distribution of solvers amongst best solutions
Slika 5. Raspodjela rješavača među najboljim rješenjima

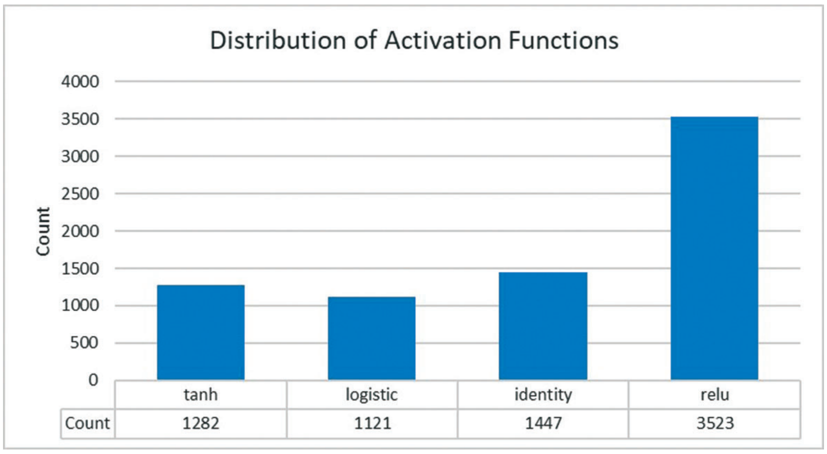


Figure 6 Distribution of Activation Functions amongst best solutions
Slika 6. Raspodjela funkcija aktivacije među najboljim rješenjima

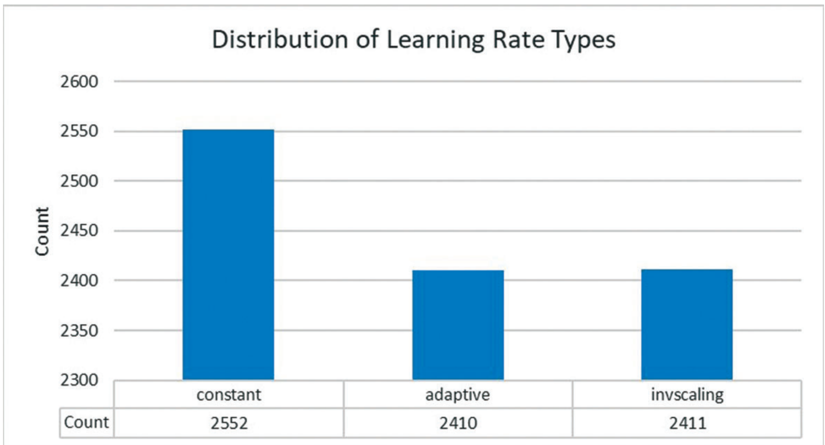


Figure 7 Distribution of learning rate types amongst best solutions
Slika 7. Raspodjela vrsta stope učenja među najboljim rješenjima

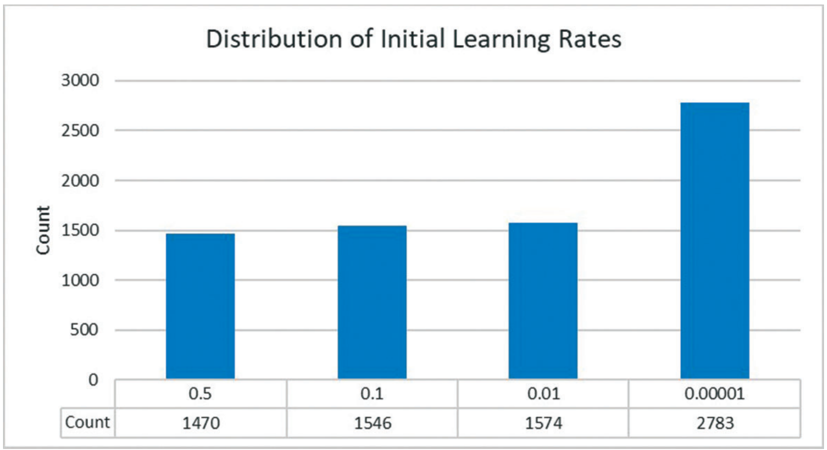


Figure 8 Distribution of Initial Learning Rates amongst best solutions
Slika 8. Raspodjela početnih stopa učenja među najboljim rješenjima

5. DISCUSSION / *Rasprava*

The results show a successful regression of the speed value from the parameters contained within the dataset. Best solution achieved has an extremely small error, and a large number of MLPs trained show the ability of regressing the value with a low (<1%) error.

When observing the solutions with the R^2 value larger than 0.99, observations can be made about the number of solutions that had certain parameter values. It can be seen that most solutions that satisfied the condition used the lbfgs solver. Most solutions preferred the relu activation function with more than double solutions using it, compared to equally distributed number of solutions which used tanh, logistic and identity activation functions. Learning rate adjustment types were equally distributed amongst all possible values (constant, adaptive and invscaling). When observing the learning rates most solutions preferred the smallest initial learning rate of 0.00001, with 2783 using that value. Number of solutions using

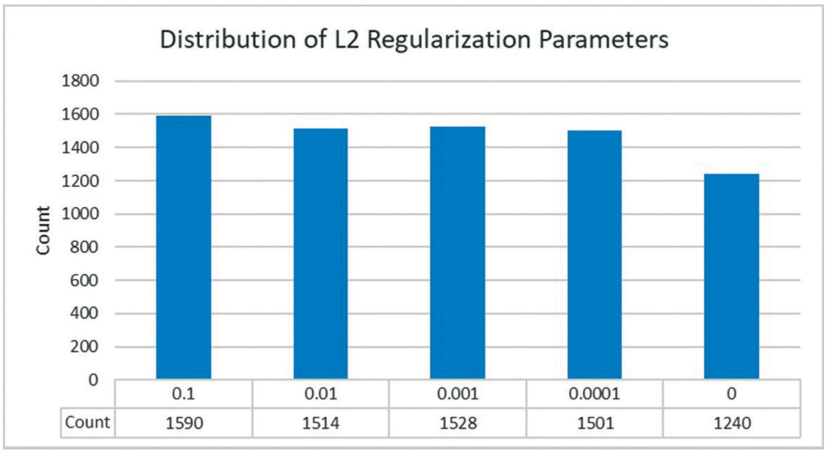


Figure 9 Distribution of regularization parameters amongst best solutions
Slika 9. Raspodjela regulacijskih parametara među najboljim rješenjima

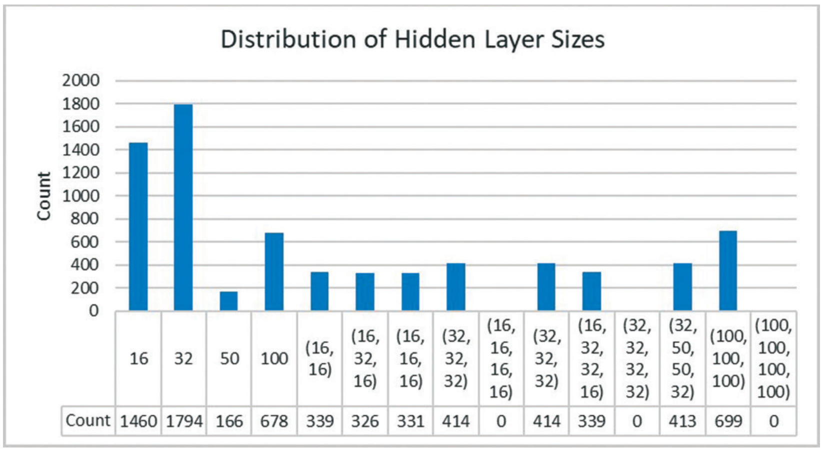


Figure 10 Distribution of Hidden Layer Sizes amongst best solutions
Slika 10. Raspodjela veličina skrivenih slojeva među najboljim rješenjima

the remaining learning rates were equally distributed. The number of solutions using L2 regularization parameter was equally distributed amongst all possible values. Equal distribution such as this points towards regularization of the input parameters not being largely important for the dataset, i.e. all variables affecting the speed outcome in a similar amount. When observing the graph for hidden layer sizes it is clear that more solutions used simpler neural network architecture. Using a single layer of 32 neurons has the largest amount of solutions, with single layer of 16 neurons being a close second. Other solutions are equally distributed, except for the most solutions using four hidden layers which had no solutions amongst the best ones.

6. CONCLUSION / *Zaključak*

The research done shows the possibility of regressing the speed of a vessel from propulsion system parameters. It is also shown that MLP is a suitable algorithm for such a task, due to low error and high coefficient of determination achieved by it. Research shows that satisfactory results can be achieved with relatively simple MLP architectures. Best regression performance is achieved by the MLP which uses three hidden layers with 100 neurons in each, a logistic activation function with lbfgs solver while using the learning rate of 0.1 throughout the training process and performing no regularization. Such a solution achieved a minimal MAE of 3.44854×10^{-5} knots (~0.00014% of the total speed range contained in the dataset).

Acknowledgment / *Zahvala*

This research has been (partly) supported by the CEEPUS network CIII-HR-0108, European Regional Development Fund under the grant KK.01.1.1.01.0009 (DATACROSS) and University of Rijeka scientific grant uniri-tehnic-18-275-1447.

REFERENCES / *Literatura*

[1] Mao, W., Rychlik, I., Wallin, J., & Storhaug, G. (2016). Statistical models for the speed prediction of a container ship. Ocean engineering, 126, 152-162. https://doi.org/10.1016/j.oceaneng.2016.08.033

[2] Yasukawa, H., & Yoshimura, Y. (2015). Introduction of MMG standard method for ship maneuvering predictions. Journal of Marine Science and Technology, 20(1), 37-52. https://doi.org/10.1007/s00773-014-0293-y

[3] Pascoal, R., Perera, L. P., & Soares, C. G. (2017). Estimation of directional sea spectra from ship motions in sea trials. Ocean Engineering, 132, 126-137. https://doi.org/10.1016/j.oceaneng.2017.01.020

[4] Borkowski, T., Kasyk, L., & Kowalak, P. (2011). Assessment of ship's engine effective power, fuel consumption and emission using the vessel speed. Journal of KONES, 18, 31-39. doi: https://doi.org/10.5957/jspd.32.2.150008

[5] Yau, P. S., Lee, S. C., Corbett, J. J., Wang, C., Cheng, Y., & Ho, K. F. (2012). Estimation of exhaust emission from ocean-going vessels in Hong Kong. Science of the total environment, 431, 299-306. https://doi.org/10.1016/j.scitotenv.2012.03.092

[6] Gan, S., Liang, S., Li, K., Deng, J., & Cheng, T. (2016, August). Ship trajectory prediction for intelligent traffic management using clustering and ANN. In 2016 UKACC 11th International Conference on Control (CONTROL) (pp. 1-6). IEEE. https://doi.org/10.1109/control.2016.7737569

[7] Gan, S., Liang, S., Li, K., Deng, J., & Cheng, T. (2016). Long-term ship speed prediction for intelligent traffic signaling. IEEE Transactions on Intelligent Transportation Systems, 18(1), 82-91. https://doi.org/10.1109/tits.2016.2560131

[8] Lorencin, I., Anđelić, N., Španjol, J., & Car, Z. (2019). Using multi-layer perceptron with Laplacian edge detector for bladder cancer diagnosis. Artificial Intelligence in Medicine, 101746. https://doi.org/10.1016/j.artmed.2019.101746

[9] Wei, Y., Hiraga, M., Ohkura, K., & Car, Z. (2019). Autonomous task allocation by artificial evolution for robotic swarms in complex tasks. Artificial Life and Robotics, 24(1), 127-134. https://doi.org/10.1007/s10015-018-0466-6

[10] Wei, Y., Nie, X., Hiraga, M., Ohkura, K., & Car, Z. (2019). Developing end-to-end control policies for robotic swarms using deep Q-learning. Journal of Advanced Computational Intelligence and Intelligent Informatics, 23(5), 920-927. https://doi.org/10.20965/jaciii.2019.p0920

[11] Lorencin, I., Anđelić, N., Mrzljak, V., & Car, Z. (2019). Marine Objects Recognition Using Convolutional Neural Networks. NAŠE MORE: znanstveno-stručni časopis za more i pomorstvo, 66(3), 112-119. https://doi.org/10.17818/nm/2019/3.3

[12] Lorencin, I., Anđelić, N., Mrzljak, V., & Car, Z. (2019). Genetic Algorithm Approach to Design of Multi-Layer Perceptron for Combined Cycle Power Plant Electrical Power Output Estimation. Energies, 12(22), 4352. https://doi.org/10.3390/en12224352

[13] Li, Y.; Tang, G.; Du, J.; Zhou, N.; Zhao, Y.; Wu, T. Multilayer Perceptron Method to Estimate Real-World Fuel Consumption Rate of Light Duty Vehicles. IEEE Access 2019, 7, 63395–63402. https://doi.org/10.1109/access.2019.2914378

[14] Kim, M.; Choi, W.; Jeon, Y.; Liu, L. A Hybrid Neural Network Model for Power Demand Forecasting. Energies 2019, 12, 931. https://doi.org/10.3390/en12050931

[15] Ghorbani, M.; Deo, R.C.; Yaseen, Z.M.; Kashani, M.H.; Mohammadi, B. Pan evaporation prediction using a hybrid multilayer perceptron-firefly algorithm (MLP-FFA) model: Case study in North Iran. Theor. Appl. Climatol. 2018, 133, 1119–1131. https://doi.org/10.1007/s00704-017-2244-0

[16] Moura, M.B.; Vidal, D.C.; Schueler, C.; de Matos, L.J.; Ferreira, T.N. Outdoor-to-Indoor Power Predictionf or 768 MHz Wireless Mobile Transmission using Multilayer Perceptron. In Proceedings of the 2018 IEEE International Joint Conference on Neural Networks (IJCNN), 2018; pp. 1–7. https://doi.org/10.1109/ijcnn.2018.8489221

[17] Feng, Y., Wang, H., Gao, R., Zhu, Y.: A Zero-Dimensional Mixing Controlled Combustion Model for Real Time Performance Simulation of Marine Two-Stroke Diesel Engines, Energies 12 (10), 2000, 2019. https://doi.org/10.3390/en12102000

[18] Mrzljak, V., Medica, V., Bukovac, O.: Simulation of a Two-Stroke Slow Speed Diesel Engine Using a Quasi-Dimensional Model, Transactions of Famena 2, p. 35-44, 2016. https://doi.org/10.21278/tof.40203

[19] Wang, G., Yu, W., Li, X., Su, Y., Yang, R., Wu, W.: Experimental and numerical study on the influence of intake swirl on fuel spray and in-cylinder combustion characteristics on large bore diesel engine, Fuel 237, p. 209–221, 2019. https://doi.org/10.1016/j.fuel.2018.09.156

[20] Wang, Z., Kuang, H., Zhang, J., Chu, L., Ji, Y.: Nitrogen Oxide Removal by Coal-Based Activated Carbon for a Marine Diesel Engine, Applied sciences 9 (8), 1656, 2019. https://doi.org/10.3390/app9081656

[21] Senčić, T., Mrzljak, V., Blecich, P., Bonefačić, I.: 2D CFD Simulation of Water Injection Strategies in a Large Marine Engine, Journal of Marine Science and Engineering 7 (9), 296, 2019. https://doi.org/10.3390/jmse7090296

[22] Mrzljak, V., Medica, V., Bukovac, O.: Volume agglomeration process in quasi-dimensional direct injection diesel engine numerical model, Energy 115, p. 658-667, 2016. https://doi.org/10.1016/j.energy.2016.09.055

[23] Koroglu, T., Sogut, O. S.: Conventional and Advanced Exergy Analyses of a Marine Steam Power Plant, Energy 163, p. 392-403, 2018. https://doi.org/10.1016/j.energy.2018.08.119

[24] Mrzljak, V., Poljak, I., Mrakovčić, T.: Energy and exergy analysis of the turbo-generators and steam turbine for the main feed water pump drive on LNG carrier, Energy Conversion and Management, 140, p. 307–323, 2017. https://doi.org/10.1016/j.enconman.2017.03.007

[25] Mrzljak, V., Blecich, P., Anđelić, N., & Lorencin, I. (2019). Energy and Exergy Analyses of Forced Draft Fan for Marine Steam Propulsion System during Load Change. Journal of Marine Science and Engineering, 7(11), 381.https://doi.org/10.3390/jmse7110381

[26] Mrzljak, V., Senčić, T., & Žarković, B. (2018). Turbogenerator steam turbine variation in developed power: Analysis of exergy efficiency and exergy destruction change. Modelling and Simulation in Engineering, 2018. https://doi.org/10.1155/2018/2945325

[27] Lorencin, I., Anđelić, N., Mrzljak, V., & Car, Z. (2019). Exergy analysis of marine steam turbine labyrinth (gland) seals. Pomorstvo, 33(1), 76-83. https://doi.org/10.31217/p.33.1.8

[28] Marques, C. H., Caprace, J.-D., Belchior, C. R. P., Martini, A.: An Approach for Predicting the Specific Fuel Consumption of Dual-Fuel Two-Stroke Marine Engines, Journal of Marine Science and Engineering 7 (2), 20, 2019. https://doi.org/10.3390/jmse7020020

[29] Fernández, I. A., Gómez, M. R., Gómez, J. R., Insua, A. A. B.:Review of propulsion systems on LNG carriers, Renewable and Sustainable Energy Reviews 67, p. 1395–1411, 2017. https://doi.org/10.1016/j.rser.2016.09.095

[30] Ammar, N. R.: Environmental and cost-effectiveness comparison of dual fuel propulsion options for emissions reduction onboard LNG carriers, Shipbuilding: Theory and Practice of Naval Architecture, Marine Engineering and Ocean Engineering 70 (3), p. 61-77, 2019. https://doi.org/10.21278/brod70304

[31] Lorencin, I., Anđelić, N., Mrzljak, V., & Car, Z. (2019). Multilayer Perceptron approach to Condition-Based Maintenance of Marine CODLAG Propulsion System Components. Pomorstvo, 33(2)https://doi.org/10.31217/p.33.2.8

[32] Coraddu, A., Oneto, L., Ghio, A., Savio, S., Anguita, D., & Figari, M. (2016). Machine learning approaches for improving condition-based maintenance of naval propulsion plants. Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment, 230(1), 136-153. https://doi.org/10.1177/1475090214540874

[33] Cipollini, F., Oneto, L., Coraddu, A., Murphy, A. J., & Anguita, D. (2018). Condition-based maintenance of naval propulsion systems with supervised data analysis. Ocean Engineering, 149, 268-278. (doi:10.1016/j.oceaneng.2017.12.002) https://doi.org/10.1016/j.oceaneng.2017.12.002

[34] Coraddu, A., Oneto, L., Ghio, A., Savio, S., Figari, M., & Anguita, D. (2015, March). Machine learning for wear forecasting of naval assets for condition-based maintenance applications. In 2015 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles (ESARS) (pp. 1-5). IEEE. https://doi.org/10.1109/esars.2015.7101499

[35] Kostyuk, A., Frolov, V.: Steam and gas turbines, Mir Publishers, Moscow, 1988.

[36] Heidari, A. A., Faris, H., Aljarah, I., & Mirjalili, S. (2019). An efficient hybrid multilayer perceptron neural network with grasshopper optimization. Soft Computing, 23(17), 7941-7958. https://doi.org/10.1007/s00500-018-3424-2

[37] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[38] Kanwar, N., Goswami, A. K., & Mishra, S. P. (2019, April). Design Issues in Artificial Neural Network (ANN). In 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) (pp. 1-4). IEEE. https://doi.org/10.1109/iot-siu.2019.8777337

[39] Zhang, H., Weng, T. W., Chen, P. Y., Hsieh, C. J., & Daniel, L. (2018). Efficient neural network robustness certification with general activation functions. In Advances in neural information processing systems (pp. 4939-4948).

[40] Karlik, B., & Olgac, A. V. (2011). Performance analysis of various activation functions in generalized MLP architectures of neural networks. International Journal of Artificial Intelligence and Expert Systems, 1(4), 111-122.

[41] Shanahan, M., Nikiforou, K., Creswell, A., Kaplanis, C., Barrett, D., & Garnelo, M. (2019). An Explicitly Relational Neural Network Architecture. arXiv preprint arXiv:1905.10307.

[42] Hastie T, T. R., & Friedman, J. H. (2003). Elements of statistical learning: data mining, inference, and prediction.

[43] Zhang, N., Shen, S. L., Zhou, A., & Xu, Y. S. (2019). Investigation on performance of neural networks using quadratic relative error cost function. IEEE Access, 7, 106642-106652. https://doi.org/10.1109/access.2019.2930520

[44] Seth, N., Ubrani, A., Mane, S., & Kazi, F. A. (2019). Comparative Analysis of Major Jacobian and Gradient Backpropagation Optimizers of ANN on SVPWM. In Soft Computing and Signal Processing (pp. 345-357). Springer, Singapore. https://doi.org/10.1007/978-981-13-3600-3_32

[45] Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv:1803.09820.

[46] Kaul, A., Jandhyala, V. K., & Fotopoulos, S. B. (2018). Parameter estimation for high dimensional change point regression models without grid search. arXiv preprint arXiv:1805.03719.

[47] Wang, B., & Gong, N. Z. (2018, May). Stealing hyperparameters in machine learning. In 2018 IEEE Symposium on Security and Privacy (SP) (pp. 36-52). IEEE. https://doi.org/10.1109/sp.2018.00038

[48] Baressi Šegota, S., Anđelić, N., Kudláček, J., & Čep, R. (2019). Artificial neural network for predicting values of residuary resistance per unit weight of displacement. Pomorski zbornik, 57(1), 9-22 .(https://doi.org/10.18048/2019.57.01.)

[49] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... & Layton, R. (2013). API design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238. https://doi.org/10.1145/2786984.2786995

[50] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830. https://doi.org/10.3389/fninf.2014.00014

[51] Ju, C., Bibaut, A., & van der Laan, M. (2018). The relative performance of ensemble methods with deep convolutional neural networks for image classification. Journal of Applied Statistics, 45(15), 2800-2818. https://doi.org/10.1080/02664763.2018.1441383

[52] Wang, W., & Lu, Y. (2018, March). Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model. In IOP Conference Series: Materials Science and Engineering (Vol. 324, No. 1, p. 012049). IOP Publishing. https://doi.org/10.1088/1757-899x/324/1/012049

[53] Maiseli, B. J. (2019). Optimum design of chamfer masks using symmetric mean absolute percentage error. EURASIP Journal on Image and Video Processing, 2019(1), 74. https://doi.org/10.1186/s13640-019-0475-y

[54] Jones, T. (2019). A Coefficient of Determination for Probabilistic Topic Models. arXiv preprint arXiv:1911.11061.

[55] Armstrong, R. A. (2019). Should Pearson's correlation coefficient be avoided?. Ophthalmic and Physiological Optics. (dot: https://doi.org/10.1111/opo.12636)

[56] Kurtela, A., Antolović, N., Glavić, N., & Kožul, V. (2019). Length-Weight Relations and Monthly Occurrence of Juvenile Fish Species from the Donji Molunat Bay, Croatia (South-East Adriatic Sea). NAŠE MORE: znanstveno-stručni časopis za more i pomorstvo, 66(3 Supplement), 1-5. https://doi.org/10.17818/nm/2019/3.7