ISSN 0554-6397
UDK: 004.032.26
Preliminary communications
Received: 23rd March 2020

Sandi Baressi Šegota
E-mail: sbaressisegota@riteh.hr
University of Rijeka, Faculty of Engineering, Vukovarska 58, 51000 Rijeka, Croatia
Daniel Štifanić
E-mail: dstifanic@riteh.hr
University of Rijeka, Faculty of Engineering, Vukovarska 58, 51000 Rijeka, Croatia
Kazuhiro Ohkura
E-mail: kohkura@hiroshima-u.ac.jp
Hiroshiuma University, 1-4-1, Kagamiyama, Higashi-Hiroshima,
Hiroshima, 739-8527, Japan
Zlatan Car
E-mail: car@riteh.hr
University of Rijeka, Faculty of Engineering, Vukovarska 58, 51000 Rijeka, Croatia

Use of Artificial Neural Network for Estimation of Propeller Torque Values in a CODLAG Propulsion System

Abstract

An artificial neural network (ANN) approach is proposed to the problem of estimating the propeller torques of a frigate using combined diesel, electric and gas (CODLAG) propulsion system. The authors use a multilayer perceptron (MLP) feed-forward ANN trained with data from a dataset which describes the decay state coefficients as outputs and system parameters as inputs – with a goal of determining the propeller torques, removing the decay state coefficients and using the torque values of the starboard and port propellers as outputs. A total of 53760 ANNs are trained – 26880 for each of the propellers, with a total 8960 parameter combinations. The results are evaluated using mean absolute error (MAE) and coefficient of determination (R^2). Best results for the starboard propeller are MAE of 2.68 [Nm], and MAE of 2.58 [Nm] for the port propeller with following ANN configurations respectively: 2 hidden layers with 32 neurons and identity activation and 3 hidden layers with 16, 32 and 16 neurons and identity activation function. Both configurations achieve R^2 value higher than 0.99.

Keywords: artificial neural network, machine learning, CODLAG, propeller torque estimation, propulsion systems

1. Introduction

Propeller torque has a great effect on the thrust produced by the frigates' propulsion system. Propeller torque can also affect fuel consumption, operating safety and reliability. This paper attempts to determine the torque values of starboard and port propellers of a frigate. With this goal in mind, the authors propose the use of a MLP ANN, trained using the data contained in the *"Condition Based Maintenance of Naval Propulsion Plants Data Set"* [1]. This dataset was designed to estimate the need for maintenance based on the frigate CODLAG propulsion system condition; but it contains many measured parameter values. Previous work has shown that methods such as machine learning [2, 3] and evolutionary computing algorithms [4] can be used to a great effect in maritime applications, such as optimization of labyrinth seal leakage [5], marine propulsion systems [6 - 8] and operation dynamics [9 - 12] as well as other fields such as medicine [13, 14], energetics [15] and robotics [16]. As opposed to the use of the planned output (decay coefficients), two of the planned input parameters are used for the output – starboard and port propeller torque.

1.1. Research questions

Through research performed in this paper authors want to answer the following questions:

- can torque value of the frigate propellers be estimated from the dataset collected for the purpose of decay coefficient measurement and determination and
- can the same ANN architecture be used for both port and starboard propellers torque, due to similarities in used inputs and configuration?

1.2. State of the Art

Tan et al. (2019) show the use of a single class support vector machine to assess the need for the maintenance based on condition of CODLAG propulsion system. Their research demonstrates the ability to use a machine learning algorithm to determine the need for maintenance of such a system [17]. Lorencin et al. (2019) attempt to determine the decay coefficients of the CODLAG propulsion system in the same manner using a MLP. This approach shows a high regression quality with lowest mean relative error achieved being 1.094 % [18]. Li et al. (2019) demonstrate the use of the MLP ANN to estimate the real fuel consumption rate of a light-duty vehicle; through the use of data containing external environmental factors, the manipulation of vehicle companies, and the drivers' driving habits. After training the MLP with 2,424,379 samples satisfactory results are shown, but authors conclude the difference between real world and modeled data exists [19]. Fang et al. (2019) show the use of Convolutional Neural Networks and Generative Adversarial Networks, along with deconvolution, with the goal of fuel

consumption prediction. Authors conclude that by using their proposed FuelNet ANN fuel consumption can be reduced by up to 12.8% [20]. Do et al. (2020) demonstrate the use of heuristic algorithms such as cuckoo search, gravitational search algorithm, particle swarm optimization and genetic algorithm to adjust parameters of an ANN with the goal of predicting performance and exhaust emissions of a diesel engine. Authors found that ANNs with parameters optimized using the above heuristic algorithms provide satisfactory models for the estimation of performance and exhaust emissions [21].

2. Methodology

In this chapter, the methodology of the research is presented. First, the dataset used in research is described, followed by the description of MLP ANN, and model evaluation used in the paper.

2.1. Dataset Description

Dataset consists of 16 attributes, along with 2 decay coefficient values intended by authors to be used as output. It describes the simulated data of a CODLAG propulsion plant, schematic of which is shown in Figure 1 [1, 22]. Dataset contains 11934 instances.

While originally intended for the use of regressive machine learning techniques to predict the values of decay coefficients, and as such determine the need of maintenance based on the condition of the naval propulsion plant in this paper the same dataset is used for a different task. The compressor and turbine decay coefficients are removed from the dataset, and in their stead port and starboard propeller torques are used as outputs, with the remaining 14 measured parameters being used as inputs. These parameters, along with their symbols, value ranges and units are shown in Table 1.

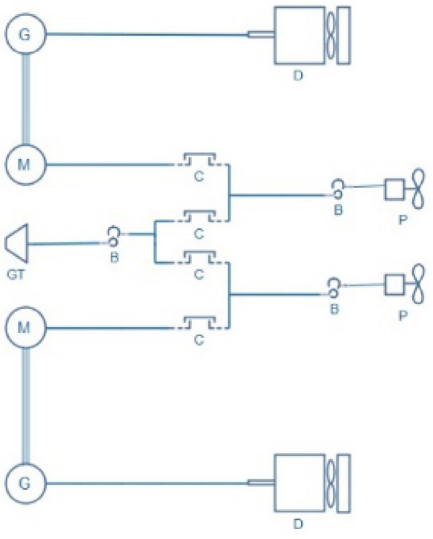

Figure 1: Schematic of CODLAG propulsion system.
(D - diesel engine; G - electric generator; M – electric motor;
GT – gas turbine; B – gearbox; C – clutch; P – frigate propeller) [1]

Table 1: List of dataset parameters, with each parameters symbol, value range and unit listed; split into parameters used as input and output to the ANN.

Parameter	Range	Unit
Lever position	1.138 - 9.3	-
Ship speed	3 - 27	kn
LP turbine shaft torque	253.547 - 72784.872	kNm
LP turbine rate of revolutions	1307.675 - 3560.741	rpm
Gas generator rate of revolutions	6589.002 - 9797.103	rpm
HP turbine exit temperature	442.364 - 1115.797	K
GT compressor inlet air temperature	288	K
GT compressor outlet air temperature	540.442 - 789.094	K
HP turbine exit pressure	1.093 - 4.56	bar
GT compressor inlet air pressure	0.998	bar
GT compressor outlet air pressure	5.828 - 23.14	bar
LP turbine exit pressure	1.019 - 1.052	bar
Combustion chamber injection control	0 - 92.556	%
Fuel Flow	0.068 - 1.832	kg/s
Starboard propeller torque	5.304 – 645.249	kNm
Port propeller torque	5.304 – 645.249	kNm

The dataset used in this research is freely available and published online.

2.2. MLP

MLP is a fully connected feed-forward ANN, which consists of neurons arranged in multiple layers – one input, one output and one or more hidden layers. Output layer consists of a single neuron, value of which is the regressed output value of the ANN. Input layer consists of the number of neurons equaling the number of input parameters of the ANN (in the given case 14). The number of hidden layers and number of neurons they consist of is one of the most important factors determining the quality of the model described within the ANN. Each neuron in the preceding layer is connected to all the neurons of the hidden layer. Value of neurons in the input layer equals the values of input parameters, while the neuron values of hidden and output layer are calculated as the activated weighted sum of inputs of neurons from the previous layer connected to the given neuron [23, 24]:

$$f\left(X_k\right) = \mathcal{F}\left(X_k \cdot \Theta\right) = \mathcal{F}\left(x_1 \cdot \theta_1 + x_2 \cdot \theta_2 + \cdots + x_n \cdot \theta_n\right) \tag{1}$$

Where $X_k = \begin{bmatrix} x_1 & x_2 & x_3 \cdots x_n \end{bmatrix}$ represents the kth vector of input values, consisting of individual values x_i; $\Theta = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \cdots \theta_n \end{bmatrix}^T$, is the vector of connection weights, consisting of individual connection weights θ_i, and $\mathcal{F}$ represents the activation function of the given neuron. Connection weights Θ are assigned to each output value of a neuron from a preceding layer. To achieve a quality model these values, need to be adjusted. The adjustment is made in the so-called "training" stage. First, a larger part of the input dataset – in the research presented in this paper case 75% (8,950) data points are separated into a training set. The remaining 25% (2,984) are placed into the testing set, to be used in the later stage [25]. For each of the data points in the training set, two steps are executed – forward propagation and backwards propagation. First, the weights of the ANN connections are randomized, and a vector of inputs consisting a single data point is used as the input value of the neural network. These values are propagated forward through the ANN, until the output neuron is reached, where output value is calculated. This is the predicted value of the ANN $\hat{y}_k$. This value is then compared to the real value y_k. The square of differences is defined as the cost function of the ANN, with m being the total number of data points used so far in the training [26]:

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

The value $J(\Theta)$ is called the cost function, and we want to minimize it towards 0 ($J(\Theta) \rightarrow 0$)). To achieve this the gradient of the weights is calculated according to [27]:

$$\Theta' = \frac{\partial J\left(\Theta\right)}{\partial \Theta}. \tag{3}$$

Use of the gradient allows as to adjust the speed of the change. If value of cost function is close to zero the weights will not get adjusted as fast as they would be if the cost function value was large. To update the weights algorithms called solvers are utilized. While the way of weight updating differs from solver to solver the basic way of updating weights is given by [27]:

$$\Theta_{new} = \Theta_{old} - \frac{\alpha}{n} \cdot \Theta^{'}, \tag{4}$$

where the value α represents the learning rate of the ANN – this parameter allows fine-tuning of the convergence speed of the ANN. With parameter α set too high the ANN will diverge, and with the value set too low the ANN will converge too slowly for practical purposes. Parameter α is one of the hyperparameters of the ANN, which are presented more closely in the following section.

Once the training stage is completed, the testing stage begins. In the testing stage the data points of the testing set are propagated forward through the ANN, but no backward propagation is used – meaning the weights are not adjusted. This stage serves to evaluate the ANN output value $\hat{y}_k$ with the real values, to determine the quality of the achieved model. Techniques used to evaluate model in the presented research are given in the section 2.2.2. Model Evaluation.

2.2.1. Hyperparameters

Hyperparameters are values that describe the architecture of the ANN used for modeling the regression task. One of the most important parameters are the hidden layers [28]. Hidden layers are defined through the tuple which describes the number of neurons in each layer. Number of layers is equal to the number of integers in the tuple, while each of the integers describes the number of neurons in the given layer. The next hyperparameter that is defined is the activation function. Activation function is a mathematical function that maps the input values to the neuron (weighted sum of inputs) to the neurons output. They can be defined to map the input of the neuron to a known range of values, such as sigmoid or Tanh activation functions, eliminate unwanted values such as Rectified Linear Unit (ReLU) or transform the input value directly, as is the case with identity activation function [29]. Functions and ranges of the activation functions used in presented research are given in Table 2.

Table 2: Activation functions used in this paper, along with their equations and ranges of equations the input is mapped to when using the given activation function.

Activation Function	Equation	Range
Identity	$\mathcal{F}(x) = x$	$[-\infty, +\infty]$
Sigmoid	$\mathcal{F}(x) = s = \frac{1}{1+e^{-x}}$	$[0,1]$
Tanh	$\mathcal{F}(x) = a = tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	$[-1,1]$
ReLU	$\mathcal{F}(x) = max(0,x)$	$[0,+\infty]$

The following hyperparameter that needs to be defined is the learning rate . As mentioned previously, this value defines the speed at which the weights are updated [30]. If this value is too large, the weight adjustment will not be capable of finding the point at which the cost function equals zero, but if set to a value that is too low the ANN will not be capable of converging within a viable execution time [31]. The adjustment of the learning rate through the training is also one of the used hyperparameters. It can be set to be constant for which the value of learning rate does not change through the execution, adaptive for which the value of learning rate adjusts itself depending on the current value of the cost function, and inverse scaling which lowers the learning rate as the training elapses, to avoid problems with divergence once the weights reach the values for which the cost function is near zero [32]. After these values the regularization parameter L2 needs to be set. Regularization is a technique in which automatically lowers the weights which are significantly higher than others. The larger values of L2 parameter have a greater effect at lowering the higher weights [33]. Regularization helps with avoiding too large of an influence of certain input parameters, but can sometimes have negative effects, such as eliminating the parameters which realistically have a large influence on the output. Finally, the solver is defined [34]. Solver is an algorithm which defines calculates the gradient and adjusts the weights during the training phase [35, 36]. Possible values of hyperparameters are shown in Table 3. In Table 3 first column is the name of the hyperparameter, second shows the possible values the hyperparameter can be set to and finally, the third column "count" represents the total number of possible variations for each hyperparameter.

Finding the right combination of hyperparameters are important part of achieving a high-quality model. In the presented research the grid search algorithm is used. Grid search is one of the simplest algorithms for finding good hyperparameter combinations. It begins by defining all the possible hyperparameter combinations – in our case that is a total of 8960 combinations [37, 38]. Then, a separate neural network is trained and evaluated for each of those combinations, with the most successful ANNs stored. To

avoid the problem of initial, randomized, weights influencing the final models' quality, multiple trainings are performed. In the work shown, six neural networks are trained for each hyperparameter combination – three ANNs are trained using port propeller torque as output; and three ANNs are trained for the starboard propeller torque modeling. This gives a total of 53760 ANNs trained and evaluated.

Table 3: List of Hyperparameters adjusted in the research and their possible values.

Hyperparameter	Possible values of a hyperparameter	Count
Hidden Layers Tuple	(16) (32) (50) (100) (16, 16) (32, 32) (16, 32, 32) (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 of ANN	Identity ReLU Tanh Logistic	4
α	0.5 0.1 0.01 0.00001	5
α change type	Constant Adaptive Inverse Scaling	3
L2 regularization parameter	0.1 0.01 0.001 0.00001 0.0	5
Solver	Adam LBFGS	2

2.2.2. Model Evaluation

Models are evaluated using two metrics – mean absolute error (MAE) and coefficient of determination (R^2). If we define the real value contained in the dataset as $Y = [y_1 \ y_2 \ ... \ y_n]$, with each member y_i being a single real output value from the dataset; and $\hat{Y} = [\hat{y}_1 \ \hat{y}_2 \ ... \ \hat{y}_n]$ as a solution vector calculated by the ANN, with each member $\hat{y}_i$ being the single output value corresponding to the set of inputs associated with real output value y_i, then we can define MAE using [39, 40]:

$$MAE = \frac{1}{n}\sum_{i}^{n}\left|y_i - \hat{y}_i\right|. \tag{5}$$

R^2 provides information on how well the variance is contained in the set Y (real results) represented by the predicted results $\hat{Y}$ [41]. The closer the value is to 1, higher the regression quality of the evaluated model [42]. By using the previously defined variables R^2 can be defined with [43]:

$$R^2 = 1 - \frac{S_{RESIDUAL}}{S_{TOTAL}} = 1 - \frac{\sum_i^n (y_i - \hat{y}_i)^2}{\sum_i^n \left(y_i - \frac{1}{n}\sum_i^n y_i\right)^2}. \tag{6}$$

In this paper, the first evaluation is done using R^2 score, in which a model needs to have a value of at least 0.9 to be considered for further evaluation. Then, the remaining models are evaluated using MAE, with those that have a lower MAE being considered the higher quality models.

3. Results

A total of 7200 results are collected for starboard and 7237 for port propeller which satisfies the condition of R^2 being larger than 0.9. Distribution of results achieved are given in Figure 2 for R^2 score, and in figure 3 for MAE. For better visibility a log scale is used on y-axis. It can be seen that a large number of models have a MAE value larger than 10, which is comparatively a weak result, but a high number of models have a high R2 value. The reason for this is that, while R2 analyses the variance between two sets of data it does not necessarily mean that a model with high R2 value will have similar values, just that the both sets of data follow a similar trend [40, 42]. This is why, although smaller error and high R2 score have a certain degree of correlation, R2 has to be used in conjunction with an error measurement, especially when high number of quality models are generated such as was the case in this research [41].

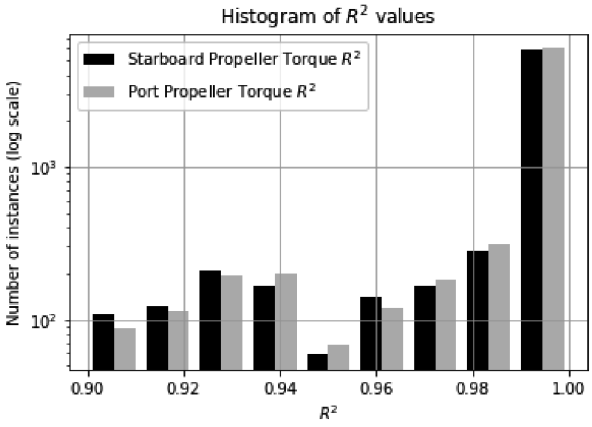

Figure 2: Histogram showing distribution of R^2 values, for collected solutions. On the x-axis bins of R^2 values are shown, and the number of solutions in each bin is shown on y-axis (logarithmical scale).

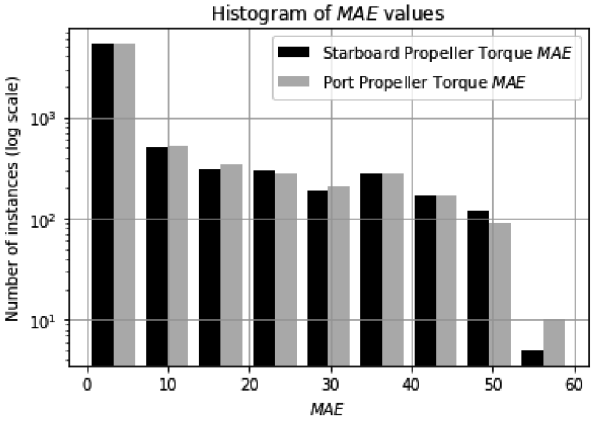

Figure 3: Histogram showing distribution of MAE values, for collected solutions. On the x-axis bins of MAE values are shown, and the number of solutions in each bin is shown on y-axis (logarithmical scale).

The best result achieved for port propeller torque has R^2 value of 0.9999, MAE 0.00257737 [kNm], which – when equals 0.00040275 % of the total range of the output torque parameter. Best result achieved for the starboard propeller torque has R^2 value of 0.9999, MAE 0.00268098 [kNm], which – when equals 0.00041894 % of the total range of the output torque parameter. These results are presented in Table 4.

Table 4: Regression scores and hyperparameters of ANNs that achieved the best scores for port and starboard propeller torque values.

Output	R^2	MAE [kNm]	MAE [%]	Hidden Layers	Activation Function	Solver	α	α adjustment	L2
Port Propeller Torque	0.9999	0.00257737	0.00040275	(32, 32)	Identity	lbfgs	$1x10^{-5}$	Constant	0.1
Starboard Propeller Torque	0.9999	0.00268098	0.00041894	(16, 32, 16)	Identity	lbfgs	$1x10^{-1}$	Adaptive	0.01

To answer the question of what the result of the same ANN would be being applied to the regression modeling of the opposite propellers torques, we can look at the results of ANNs with same architectures, but for the opposite propeller torque. Best results presented by the same architecture as the best – but for the opposite propeller are shown in Table 5.

Table 5: Best scores of models for the starboard and port propeller torques using the best model for regressing the opposite (port architecture shown in Table 4 applied on starboard propeller torque output, and vice-versa) model torque

Output	R^2	MAE [kNm]	MAE [%]
Starboard propeller torque	0.9999	0.17495312	0.02733877
Port propeller torque	0.9999	0.01873803	0.00292807

4. Conclusion

Results show that it is possible to achieve a successful regression using ANN on both the models for starboard and port propeller torque values. Both models achieve extremely low errors, with MAE below 3 [Nm]. When considering the possible range of torque values, which is approximately 640 [kNm] as shown in Table 1 the error achieved is below 0.001 [%] of the total possible range. Regression quality of models is high, as evidenced by high R2 scores. This points towards the possibility of using ANN modeling in predictions of various frigate propulsion system values. In this way values which are complex to calculate and predict can be determined using measured data and well-trained AI systems. Tests performed on the application of models of opposite propellers show the possibility of use of models created for one propeller on the other, with minor loss in accuracy. This research demonstrates the possibility of creating regression models of values which were originally intended to be input parameters. In practical terms this means that any dataset collected can be used not only for regressing the originally measured output, but input parameters as well. This implies a wider usage of existing datasets and datasets which will be collected in the future.

5. Acknowledgment

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), project CEKOM under the grant KK.01.2.2.03.0004 and University of Rijeka scientific grant uniri-tehnic-18-275-1447.

6. References

1. 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. (doi: 10.1177/1475090214540874)
2. 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. (doi: 10.18048/2019.57.01.)
3. 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. (doi: 10.17818/NM/2019/3.3)
4. Baressi Šegota, S., Lorencin, I., Ohkura, K., & Car, Z. (2019). On the Traveling Salesman Problem in Nautical Environments: an Evolutionary Computing Approach to Optimization of Tourist Route Paths in Medulin, Croatia. *Pomorski zbornik*, *57*(1), 71-87. (doi: 10.18048/2019.57.05.)
5. Lorencin, I., Anđelić, N., Mrzljak, V., & Car, Z. (2019). Exergy analysis of marine steam turbine labyrinth (gland) seals. *Pomorstvo*, *33*(1), 76-83. (doi: 10.31217/p.33.1.8)
6. Bukovac, O., Medica, V., & Mrzljak, V. (2015). Steady state performances analysis of modern marine two-stroke low speed diesel engine using MLP neural network model. *Brodogradnja: Teorija i praksa brodogradnje i pomorske tehnike*, *66*(4), 57-70.
7. Mrzljak, V., Poljak, I., & Prpić-Oršić, J. (2019). Exergy analysis of the main propulsion steam turbine from marine propulsion plant. *Brodogradnja: Teorija i praksa brodogradnje i pomorske tehnike*, *70*(1), 59-77. (doi: 10.21278/brod70105)
8. Poljak, I., Orović, J., Mrzljak, V., & Bernečić, D. (2020). Energy and Exergy Evaluation of a Two-Stage Axial Vapour Compressor on the LNG Carrier. *Entropy*, *22*(1), 115. (doi: doi.org/10.3390/e22010115)
9. Mrzljak, V., Anđelić, N., Poljak, I., & Orović, J. (2019). Thermodynamic analysis of marine steam power plant pressure reduction valves. *Pomorski zbornik*, *56*(1), 9-30. (doi: 10.18048/2019.56.01)
10. Orović, J., Mrzljak, V., & Poljak, I. (2018). Efficiency and losses analysis of steam air heater from marine steam propulsion plant. *Energies*, *11*(11), 3019. (doi: 10.3390/en11113019)
11. Mrzljak, V., Orović, J., Poljak, I., & Lorencin, I. (2019, January). Exergy analysis of steam turbine governing valve from a super critical thermal power plant. In *XXVII INTERNATIONAL SCIENTIFIC CONFERENCE trans & MOTAUTO'19*
12. Poljak, I., Orović, J., & Mrzljak, V. (2018). Energy and exergy analysis of the condensate pump during internal leakage from the marine steam propulsion system. *Pomorstvo*, *32*(2), 268-280. (doi: 10.31217/p.31.2.12)
13. Bogović, K., Lorencin, I., Anđelić, N., Blažević, S., Smolčić, K., Španjol, J., & Car, Z. (2018, January). Artificial intelligence-based method for urinary bladder cancer diagnostic. In *International Conference on Innovative Technologies, IN-TECH 2018*.
14. Lorencin, I., Anđelić, N., Španjol, J., & Car, Z. (2020). Using multi-layer perceptron with Laplacian edge detector for bladder cancer diagnosis. *Artificial Intelligence in Medicine*, *102*, 101746. (doi: 10.1016/j.artmed.2019.101746)
15. 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. (doi: 10.3390/en12224352)

16. Baressi Šegota, S., Anđelić, N., Lorencin, I., Saga, M., & Car, Z. (2020). Path planning optimization of six-degree-of-freedom robotic manipulators using evolutionary algorithms. *International Journal of Advanced Robotic Systems*. (doi: 10.1177/1729881420908076)
17. Tan, Y., Niu, C., Tian, H., Hou, L., & Zhang, J. (2019). A one-class SVM based approach for condition-based maintenance of a naval propulsion plant with limited labeled data. *Ocean Engineering*, *193*, 106592. (doi: 10.1016/j.oceaneng.2019.106592)
18. 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), 181-190. (doi: 10.31217/p.33.2.8)
19. Li, Y., Tang, G., Du, J., Zhou, N., Zhao, Y., & Wu, T. (2019). Multilayer perceptron method to estimate real-world fuel consumption rate of light duty vehicles. *IEEE Access*, *7*, 63395-63402. (doi: 10.1109/access.2019.2914378)
20. Fang, C., Song, S., Chen, Z., & Gui, A. (2019, November). Fine-Grained Fuel Consumption Prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 2783-2791). (doi: 10.1145/3357384.3357836)
21. Do, Q. H., Tuan, T. T., Ha, L. T. T., Doan, T. T. H., & Nguyen, T. V. A. (2020). Development of Artificial Neural Networks Trained by Heuristic Algorithms for Prediction of Exhaust Emissions and Performance of a Diesel Engine Fuelled with Biodiesel Blends. In *Applied Nature-Inspired Computing: Algorithms and Case Studies* (pp. 253-275). Springer, Singapore. (doi: 10.1007/978-981-13-9263-4_11)
22. Altosole, M., Benvenuto, G., Figari, M., & Campora, U. (2009). Real-time simulation of a COGAG naval ship propulsion system. *Proceedings of the institution of mechanical engineers, part M: journal of engineering for the maritime environment*, *223*(1), 47-62. (doi: 10.1243/14750902JEME121)
23. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media. (doi: 10.1007/bf02985802)
24. Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer. (doi: 10.1016/c2009-0-22409-3)
25. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press. (doi: 10.1007/s10710-017-9314-z)
26. [26] Kumari, P. A., & Geethanjali, P. (2020). Artificial Neural Network-Based Smart Energy Meter Monitoring and Control Using Global System for Mobile Communication Module. In *Soft Computing for Problem Solving* (pp. 1-8). Springer, Singapore. (doi: 10.1007/978-981-15-0184-5_1)
27. Bishop, C. M. (1999). Pattern recognition and feed-forward networks. In *The MIT encyclopedia of the cognitive sciences* (Vol. 13, No. 2). MIT Press. (doi: 10.1017/cbo9780511812651.006)
28. Tsai, C. W., Hsia, C. H., Yang, S. J., Liu, S. J., & Fang, Z. Y. (2020). Optimizing hyperparameters of deep learning in predicting bus passengers based on simulated annealing. *Applied Soft Computing*, 106068. (doi: 10.1016/j.asoc.2020.106068)
29. Goyal, M., Goyal, R., Reddy, P. V., & Lall, B. (2020). Activation Functions. In *Deep Learning: Algorithms and Applications* (pp. 1-30). Springer, Cham. (doi: 10.1007/978-3-030-31760-7_1)
30. Pramita, D., Nusantara, T., & Negara, H. R. P. (2020, January). Analysis of accuracy parameters of ANN backpropagation algorithm through training and testing of hydro-climatology data based on GUI MATLAB. In *IOP Conference Series: Earth and Environmental Science* (Vol. 413, No. 1, p. 012008). IOP Publishing. (doi: 10.1088/1755-1315/413/1/012008)
31. Norris, D. J. (2020). Practical deep learning ANN demonstrations. In *Machine Learning with the Raspberry Pi* (pp. 279-334). Apress, Berkeley, CA. (doi: 10.1007/978-1-4842-5174-4_5)
32. Haidong, S., Junsheng, C., Hongkai, J., Yu, Y., & Zhantao, W. (2020). Enhanced deep gated recurrent unit and complex wavelet packet energy moment entropy for early fault prognosis of bearing. *Knowledge-Based Systems*, *188*, 105022. (doi: 10.1016/j.knosys.2019.105022)
33. Jahangir, H., Golkar, M. A., Alhameli, F., Mazouz, A., Ahmadian, A., & Elkamel, A. (2020). Short-term wind speed forecasting framework based on stacked denoising auto-encoders with rough ANN. *Sustainable Energy Technologies and Assessments*, *38*, 100601. (doi: 10.1016/j.seta.2019.100601)
34. Perrin, D. R. (2020). Improving the Prediction Accuracy of Species Distribution Models using Artificial Neural Networks. *TNHC-Publications*.

35. Polak, A. G., Wysoczański, D., & Mroczka, J. (2019, June). Solving the inverse problem in spirometry with the methods of global and local estimation. In *2019 IEEE International Symposium on Medical Measurements and Applications (MeMeA)* (pp. 1-6). IEEE. (doi: 10.1109/memea.2019.8802169)
36. Xu, Y., Zhang, H., Li, Y., Zhou, K., Liu, Q., & Kurths, J. (2020). Solving Fokker-Planck equation using deep learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *30*(1), 013133. (doi: 10.1063/1.5132840)
37. Pillai, N., Schwartz, S. L., Ho, T., Dokoumetzidis, A., Bies, R., & Freedman, I. (2019). Estimating parameters of nonlinear dynamic systems in pharmacology using chaos synchronization and grid search. *Journal of pharmacokinetics and pharmacodynamics*, *46*(2), 193-210. (doi: 10.1007/s10928-019-09629-4)
38. Fu, W., Nair, V., & Menzies, T. (2016). Why is differential evolution better than grid search for tuning defect predictors?. *arXiv preprint (arXiv:1609.02613*.)
39. Turnbull, A. M., Crawford, G. J., & Barrett, G. D. (2020). Methods for Intraocular Lens Power Calculation in Cataract Surgery after Radial Keratotomy. *Ophthalmology*, *127*(1), 45-51. (doi: 10.1016/j.ophtha.2019.08.019)
40. Hall, G., Sanchez-Pinto, L., Floh, A., Abrams, D., & Roebuck, N. (2020). 465: Comparison of Machine Learning and Conventional Equations for Energy Estimation In The PICU. *Critical Care Medicine*, *48*(1), 213. (doi: 10.1097/01.ccm.0000620204.63922.d1)
41. Uysal, R. S., Acar-Soykut, E., & Boyaci, I. H. (2020). Determination of yolk: white ratio of egg using SDS-PAGE. *Food Science and Biotechnology*, *29*(2), 179-186. (doi: 10.1007/s10068-019-00650-4)
42. Ali, M. M., Hashim, N., & Hamid, A. S. A. (2020). Combination of laser-light backscattering imaging and computer vision for rapid determination of oil palm fresh fruit bunches maturity. *Computers and Electronics in Agriculture*, *169*, 105235. (doi: 10.1016/j.compag.2020.105235)
43. Sotgia, S., Fois, A. G., Sotgiu, E., Zinellu, A., Paliogiannis, P., Mangoni, A. A., & Carru, C. (2020). Micellar electrokinetic capillary chromatographic determination of pirfenidone and 5-carboxy-pirfenidone by direct injection of plasma from patients receiving treatment for idiopathic pulmonary fibrosis (IPF). *Microchemical Journal*, *154*, 104536. (doi: 10.1016/j.microc.2019.104536)