ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20210204092154
Original scientific paper

Prediction of Robot Grasp Robustness using Artificial Intelligence Algorithms

Sandi BARESSI ŠEGOTA*, Nikola ANĐELIĆ, Zlatan CAR, Mario ŠERCER

Abstract: Predicting the quality of the robot end-effector grasp quality during an industrial robot manipulator operation can be an extremely complex task. As is often the case with such complex tasks, Artificial Intelligence methods may be applied to attempt the creation of a model - if sufficient data exists. The presented dataset uses a publicly available dataset, consisting of 992632 measurements of position, torque, and velocity - for each of the three joints of three fingers of the simulated end-effector. The dataset is first analyzed and pre-processed to prepare it for model training. The duplicate values are removed from the dataset, as well as the statistical outliers. Then, a multilayer perceptron (MLP) machine learning algorithm is applied to 80% of the data contained in the dataset, using the Grid Search algorithm to determine the best combination of MLP hyperparameters. As the dataset consists of torque, velocity, and speed measurements for separate joints and fingers of the tested end-effector the testing is performed to see if a subset of the inputs may be used to regress the robustness of the given grip. The normalization of the dataset is also applied, and its effect on the regression quality is tested. The results, evaluated with the coefficient of determination, show that while the best model is achieved using all the possible inputs, a satisfactory result can be obtained using only velocity and torque.The results also show that the normalization of the dataset improves the regression quality in all the observed cases.

Keywords: artificial intelligence, multilayer perceptron, regression, robot grasp robustness, shadow smart grasping system

1 INTRODUCTION

End-effectors mounted on robotic manipulators are a key part of any robotic system, as they allow for interaction with the parts present in the manufacturing lines and the fulfillment of tasks that engineering and manufacturing staff wants to accomplish through the use of the robotic manipulator [1]. Many types of end-effectors exist, some of which serve for the fulfillment of a specific purpose (such as a soldering iron or a welding torch which is attached to the last joint of the manipulator), but one of the most popular types of end-effectors are graspers. These end-effectors do not serve an individual purpose but are instead designed to be capable of grasping, lifting, and moving a wide variety of differently shaped objects. General graspers, which are not designed to grasp a single shape of an object are extremely popular due to a large number of popular applications - but the high variety of tasks they can perform comes at the price [2]. General graspers may not be capable of holding objects as robustly as ones designed for a specific purpose, especially when a dynamic environment is considered, with the velocity of movement exhibiting dynamic effects, such as forces and torsion on the grasper parts [3]. These values are dependant on the manipulator paths but are hard to model using existing tools. A common problem that may be experienced by manufacturing engineers is the loss of traction on the object within the graspers hold, due to low pressure being applied onto the part - or too large of a pressure being placed on the part, possibly causing damage to it, the grasper or both. Determining the robustness of the grip, in terms of the total force exhibited on the part may be complex and will require extensive mathematical modeling. The question arises - is it possible to apply Artificial Intelligence (AI) algorithms in the solving of this problem? Previous research has shown that many hard-to-solve issues can be addressed with AI, not only in the field of robotics but others as well - such as marine engineering [4] and medicine [5]. AI has already been applied in many fields of robotics with great results. Baressi Šegota et al. (2020) [3] demonstrate the use of an evolutionary algorithm to optimize the path of a 6-DOF industrial robotic manipulator, to lower the torsion exhibited on the joints during the transversal of the trajectory. Van Vuuren et al. (2020) [6] demonstrate a machine learning method for grasp selection, by applying a three-step process consisting of a convolutional neural network for sampling, grasp evaluation, and a final learning algorithm for grasp selection. The proposed solution is shown to be capable of generating a viable grasp, on previously unseen objects in 1.3 seconds. De Coninck et al. (2020) [7] show the application of a deep learning environment through demonstration, for application in robotic manipulators that cooperate with humans. Authors demonstrate the application of the algorithm on a Franka Panda collaborative robot, with a 90% average success rate of grasp. Song et al. (2020) [8] show the application of region proposal networks in the single-stage robotic grasp detection method. The proposed algorithm uses oriented anchors and is shown to significantly lower the computational complexity in comparison to standard grasp detection algorithms.

The goal of this paper is the application of a multilayer perceptron (MLP) algorithm to regress the values of grip robustness from measurements of torque, velocity, and position contained in a publicly available robot grasp dataset. The following questions are posed and researched during the presented work:

- Can the MLP algorithm be applied to the problem of grasp robustness regression from the values of individual graspers' joint torque, velocity, and position?
- Is it possible to do the same, but without using one type of the measured physical values, allowing for possible savings, through the removal of unnecessary sensors?
- If the above is possible, which are the optimal architectures of the MLP networks that provide the highest results?
- For such a problem, is there a visible change in results when dataset normalization is applied?

In the paper, first, the used research methodology will be given, with a brief description of the used algorithms.

Then, the results will be presented and discussed, with

answers to the posed questions being given in the conclusion of the manuscript.

2 METHODOLOGY

In this section, the dataset used in the research is described, along with a short description of utilized methods. Finally, the evaluation of results is described.

2.1 Dataset Description

The dataset is publicly available and obtained at [9]. Dataset is generated by dataset authors using Shadow Robot's Smart Grasping Sandbox. Authors of the dataset used Shadow Smart Grasping system mounted on Universal Robot UR3 robotic manipulator, depicted in Fig. 1, simulating the grasping of a circular object.

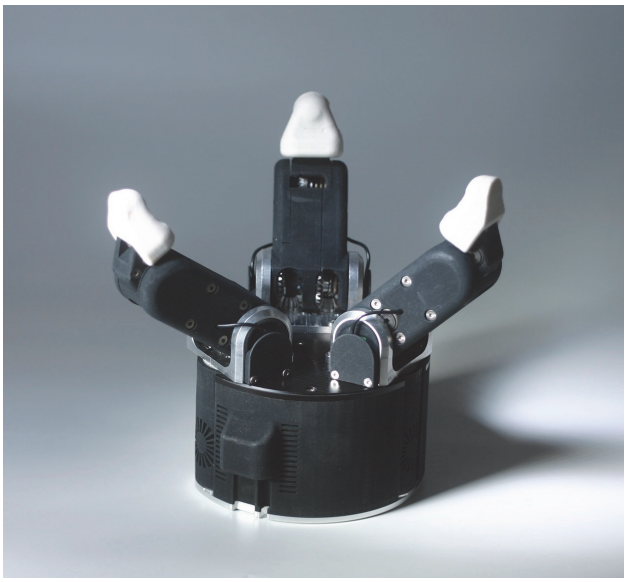

Figure 1 The image of the grasper is used within the model for dataset creation [10]

The grasping Shadow Smart Grasping System consists of three fingers, situated circularly and equidistant to the neighboring finger. Each finger consists of three joints, for 9 joints in total. The Dataset is shaped as:

$$\begin{bmatrix} P_1^1, V_1^1, T_1^1 & P_2^1, V_2^1, T_2^1 & P_3^1, V_3^1, T_3^1 \\ \vdots & \vdots & \vdots \\ P_1^n, V_1^n, T_1^n & P_2^n, V_2^n, T_2^n & P_3^n, V_3^n, T_3^n \end{bmatrix} \tag{1}$$

where the index represents the finger of the grasper *P*, *V*, *T* represents the position, velocity, and torque vectors for each of the fingers, and *n* the number of data points in the dataset. Each of the points P_i^j, V_i^j, T_i^j, represents the *j*-th measurement for the *i*-th finger ($i \in \{1,2,3\}$), If the joint of the particular grasper finger is given as *k* in $P_{i,k}^j, V_{i,k}^j, T_{i,k}^j$, then each of the above may be defined as:

$$P_i^j = \left[P_{i,1}^j, P_{i,2}^j, P_{i,3}^j \right] \tag{2}$$

$$V_i^j = \left[V_{i,1}^j, V_{i,2}^j, V_{i,3}^j \right] \tag{3}$$

and

$$T_i^j = \left[T_{i,1}^j, T_{i,2}^j, T_{i,3}^j \right] \tag{4}$$

In each simulation, the measurements are performed for each joint of the grasper, with three values being measured - the angle of the joint, velocity of the joint, and the torque on the joint. The robustness of the grasp is given as a distance between the palm and the given object [9].

2.2 Data Preprocessing

The used dataset consists of a total of 995446 measurements. Before the training of the models, the dataset is normalized and filtered. The normalization is performed on all the elements of the dataset. Within each of the individual measurements (dataset columns - M_j), the maximum and minimum values are determined. Then, the following equation is used to normalize all values from their standard range ($\left[\min M_j, \max M_j \right]$) to the range of [0, 1]. Normalization of the measurement values is done using [11]:

$$m_i^{j'} = \frac{m_i^j}{\max M_j - \min M_j} \forall m_i^j \in M_j \tag{5}$$

where m_i^j is the *i*-th element of individual measurement M_j. This is a common practice in machine learning, as it allows AI methods to more quickly converge to a desirable solution. A non-normalized set is also used in training, separately. This is done to compare the results, as normalization can sometimes have negative results [12].

Pre-processing and statistical analysis show that a dataset is highly unbalanced when it comes to the robustness output. As datasets with a high number of outliers can cause issues with the model convergence, as the outliers can cause high errors [13].

The first step is the removal of duplicate values, as such values can cause a miscalculation of scores. If the same value is found in the testing and training datasets, the value will be predicted perfectly, as the same data point was used in training. If both data points are found in the training set, the algorithm will use both values for the model training, meaning no improvement will be found. Dataset analysis shows that 2814 duplicate data points exist, and these have been removed.

Observing the dataset shows that a large number of output values lay within a small range, with a lower number of values within higher ranges. This is shown in Fig. 2.

It can be assumed that these outputs are not part of the normal operation, and are caused by inaccuracies during measurements. As such these values can be removed. To determine the outliers first the quartiles are determined. Quartiles are such values as Q_1, Q_2 and Q_3 which split dataset *D* in such a way that each of the ranges is

$\left[\min(D), Q_3(D)\right]$, $Q_3(D), Q_2(D)$, $Q_2(D), Q_1(D)$, $Q_1(D), \max(D)]$. The internal quartile range IQR is calculated as:

$$IQR = Q_3 - Q_1 \tag{6}$$

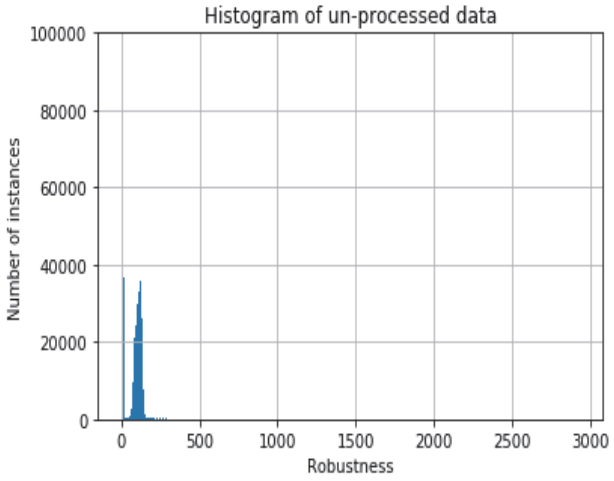

Figure 2 Histogram of Robustness values from the dataset, before dataset pre-processing

Observing Fig. 2, we can see that the lower range does not possess a high amount of outliers, as the values are grouped, so only the higher bound for outliers needs to be calculated. To confirm this we can calculate the lower bound using [13]:

$$B_L = Q_3 - 1.5 \cdot IQR \tag{7}$$

As this value comes out as negative, it can be concluded that lower bounded outliers do not exist. Upper bound is calculated similarly with:

$$B_U = Q_1 + 1.5 \cdot IQR \tag{8}$$

The histogram of the output with outliers removed is given in Fig. 3. It can be seen that the data points are now more equally distributed over the dataset range.

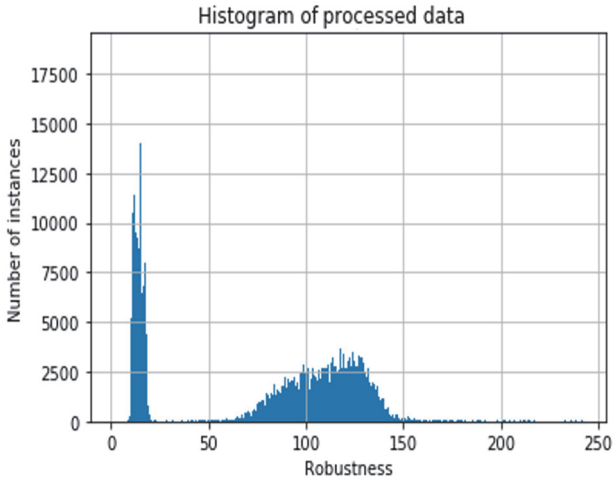

Figure 3 Histogram of Robustness values from the dataset, after the dataset pre-processing

Descriptive statistical values of the dataset before and after pre-processing are given in Tab. 1. It can be noticed how a significant number of the outlying data has been removed. Another piece of evidence signifying these values as outliers is that the mean value of the data has only slightly changed between the outlier removal.

Table 1 The minimum, maximum and mean values of the robustness in the dataset in the original and processed dataset, along with the numbers of data points in both sets

Statistic	Original data	Processed Data
Minimum Output Value	0.000001	0.000001
Maximum Output Value	2933.584361	241.7245377
Mean Output Value	77.2238	78.7887
Number of Datapoints	992632	962118

The dataset is then split into 9 subsets. One of the subsets contains all measurements. Three of the subsets consist of only a single dimension measurement - either the joint position, velocity, or torque. Finally, the remaining three are the combination of joint position and velocity, joint position and torque, and joint velocity and torque. In this manner, all the combinations of measured dimensions are used to determine if the vales can be regressed with a subset of measurements. This is done because the method used for regression can only model a single output value, so the training needs to be repeated for each desired output model. The description of the used algorithm follows.

The dataset authors have already attempted the AI modeling of the presented problem, through the classification process. Dataset authors have utilized multiple AI methods and achieved an accuracy of 78.7% [10].

2.3 MLP Description

MLP is an artificial neural network, consisting of an input layer, output layer, and one or more hidden layers. The neurons in each layer are connected using weighted connections, which connect each neuron in the current layer to all neurons in the subsequent layer. Each of the neurons serves to sum values of neurons in the previous layer, multiplied by each neuron's connection weight and activated using the current neurons activation function [14]. The activation function is a mathematical function that serves to normalize values to a given range (namely, sigmoid and tanh activation functions) or to eliminate unwanted values (such as the ReLU activation function) [15]. The model is trained using a combination of processes called forward and backward propagation. Forward propagation refers to the process of placing the values contained in the dataset onto the input neurons. The number of input neurons equals the number of variables in each data point amongst input values. The forward propagation calculates the output value of the neural network, by repeating the summation and multiplication process for each neuron of each layer, until the value of the singular neuron in the output layer [16]. The output value is determined by input values, the architecture of the model, and the connection weights. The connection weights need to be adjusted, which is done using the backpropagation process. The value acquired from the

forward propagation may be marked with $\widehat{y_i}$, while the real measured value, contained in the dataset may be marked as y. If we use n, as the number of points in the dataset, the error of the neural network, also known as the loss function, is defined as [17]:

$$J(\Theta) = \frac{\sum_{i=0}^{i=n}\left(y_k - \widehat{y_k}\right)^2}{2n} \tag{9}$$

The weights of layer k, given as $\Theta_k = [\theta_1, \theta_2, \ldots, \theta_m]$, are then adjusted depending on the value of the loss function, using the equation [18]:

$$\Theta_k^{'} = \Theta_k - \frac{\alpha}{n} \cdot \frac{\partial J(\Theta_k)}{\partial \Theta_k} \tag{10}$$

where α is the learning rate of the neural network - a value that adjusts the adjustment of weights over the training iterations. While this value cannot be set too low, as it would prevent the neural network from converging, setting it too high would cause the neural network to diverge from the solution, so careful tuning is necessary [19].

The other factor in MLP performance is the used architecture, defined by the hyperparameters [20]. The hyperparameters define the number of hidden layers and the number of neurons in each layer (given as tuple), the activation function of the neurons within the network, the solver - the algorithm used for backward propagation, learning rate α, as well as the manner of its change through iterations and L2 regularization parameter which curbs the influence of the more highly correlated values to achieve more robust models [21, 22]. The tested hyperparameters of the neural network are given in Tab. 2.

Table 2 The values of hyperparameters used in the Grid Search for the training of MLP models, with the name of the hyperparameter given in the first column, possible values of the hyperparameter in the second, and the third column representing the total number of possible values

Hyperparameter	Possible Values	Count
Hidden Layer Values	(3, 3, 3), (3, 3, 3, 3), (9, 9, 9), (9, 9, 9, 9), (27, 27, 27), (27, 27, 27, 27), (108, 108, 108), (108, 108, 108, 108), (108, 108, 108, 108, 108)	9
Learning Rate	0.5, 0.1, 0.01, 0.001	4
Learning Rate Type	Adaptive, Constant, Inverse Scaling	4
Activation Function	ReLU, Sigmoid, Identity, TanH	4
Solver	Adam, LBFGS	4
L2 Regularization Parameter	0.1, 0.01, 0.001, 0.0001	2

The grid search algorithm is used, which means that all the possible combinations of discrete hyperparameter values given in Tab. 2 have been tested. The total number of tested architectures is determined as the product of total possible values: 4808. Then, the training process is repeated for each of the combinations, and the models are separately evaluated to determine the best hyperparameter combination [5, 23]. The mode of evaluation is given in the following section.

2.3 Result Evaluation

The dataset is split into training and testing sets. The training set consists of 80% of the entire dataset (769694 data points), while the testing set consists of the remaining 20% (192424 data points). The training set is used for the previously mentioned forward and backward propagation process. Once the training is performed, the testing set is used to determine the quality of the model [24]. The testing set is unseen by the model until the evaluation and is only used for the forward propagation [25, 26]. The errors are then compared to the real values in the set, and the quality of the model is determined based on the amount of error the model achieves on the testing set [27].

Results are evaluated using the coefficient of determination. The coefficient of determination is a measurement that determines the quality of regression by determining the amount of variance of the original output data, compared to the predicted values. The value of the coefficient of determination ranges from 0 to 1, with higher values meaning that less of the variance was left unexplained [28]. This means that the higher values signify that a higher quality regression model was achieved. The coefficient of determination is marked as R^2 and given by [29, 30]:

$$R^2 = 1 - \frac{\sum_{i=0}^{i=n}\left(y_i - \widehat{y_i}\right)^2}{\sum_{i=0}^{i=n}\left(y_i - \frac{1}{n}\sum_{i=0}^{i=n} y_i\right)^2} \tag{11}$$

To define the quality of the regression for the given problem, given the high number of datapoints the R^2 score of at least 0.90 should be achieved for the regression quality to be deemed satisfactory, with an R^2 score of at least 0.95 that signifies a high-quality regression on the provided dataset [15, 30]. These values are the ones that will be used for the evaluation of the achieved results.

3 RESULTS AND DISCUSSION

The best-achieved results are shown in Tab. 3, as well as displayed in Fig. 4. For the result display, P marks "position" input, V marks "velocity" input and T marks the "torque" input. The combination of inputs is marked using the combinations of the above-listed letter, e.g. PV marks the combination of position and velocity inputs.

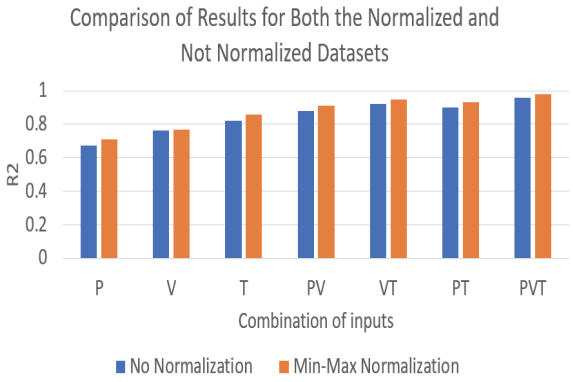

Figure 4 The comparison of results achieved for the normalized and not-normalized datasets, across all used input combinations

From Fig. 4 it can be seen that the normalization of the dataset improves the achieved scores across all the input cases, with average improvement to the R^2 score across all inputs being 0.0286, obtained from comparing values in Tab. 3. Observing Tab. 4, it can also be seen that utilization of normalization was allowed for the model which achieved the highest regression quality to use a significantly lower number of hidden neurons, making it less computationally intensive to train and use.

As shown in Fig. 4 and Tab. 3, the best results are achieved when all input combinations are used, indicating that all of the inputs - position, velocity, and torque have an influence on, or in other words a direct correlation with, the robustness of the grip. When a single input is used, the best result is reached with joint torques used as the model input. Still, it should be noted that all the single input models failed to achieve the score needed to be deemed at least satisfactory. When a combination of two inputs is used the regression quality is greatly improved, with those combinations that use torque as an input showing better results in comparison to the model which uses Joint position and velocity as inputs. When normalization is applied all three combinations reach scores above 0.90, while the position-torque and velocity-torque reach those scores even without normalization being applied. It should be noted that the combination of velocity and torque inputs, with dataset values normalized, achieves the score of 0.95 signifying a high-quality regression. When the entire dataset (position, velocity, and torque measurements) are used as inputs the highest scores are achieved - both being above 0.95, with the models trained on the not normalized data achieving the maximum R^2 of 0.96 and the models trained on the normalized data achieving the maximum R^2 of 0.98, which is above the value stated to indicate a high-quality regression.

Table 3 The maximum scores were achieved for each of the tested input combinations, with and without the application of normalization on the dataset

Normalization	Input Combination	R^2
No Normalization	*P*	0.67
	V	0.76
	T	0.82
	PV	0.88
	VT	0.92
	PT	0.90
	PVT	0.96
Min-Max Normalization	*P*	0.71
	V	0.77
	T	0.86
	PV	0.91
	VT	0.95
	PT	0.93
	PVT	0.98

The architectures of the best models are given in Tab. 4. For brevity, only the architectures of the models which achieved R^2 scores above 0.95 are given. It can be seen that all the models used the ReLU activation function and Adaptive learning rate. The model using a normalized dataset with velocity and torque inputs uses the largest possible architecture. The learning rate is in the middle of the possible value range, while the used solver is Adam. The L2 regularization parameter is also relatively high, indicating that one of the input parameters had a high influence on the output that needed to be curbed. For the model trained with the not normalized dataset and all inputs, we can see that the largest architecture was also selected, with other hyperparameters also being similar to the ones in the previously discussed model. An exception to this is the learning rate which was significantly higher for the not normalized PVT model. Finally, by observing the model with the highest score using all inputs of the normalized dataset we can see that a significantly smaller model was determined as the best, as previously mentioned. A relatively high learning rate was used, with the adaptive learning rate type selected. Other notable differences from previous models include the use of the LBFGS solver and a lower L2 regularization parameter - indicating a more balanced influence of the input parameters in the normalized dataset.

Table 4 The hyperparameters of the highest-scoring models were determined using the Grid Search algorithm

Model (R^2 score)	Hyperparameter	Value in the best model
Normalized *VT* ($R^2 = 0.95$)	Hidden Layer Values	(108, 108, 108, 108, 108)
	Learning Rate	0.01
	Learning Rate Type	Adaptive
	Activation Function	ReLU
	Solver	Adam
	L2 Regularization	0.1
Not Normalized *PVT* ($R^2 = 0.96$)	Hidden Layer Values	(108, 108, 108, 108, 108)
	Learning Rate	0.5
	Learning Rate Type	Adaptive
	Activation Function	ReLU
	Solver	Adam
	L2 Regularization	0.1
Normalized *PVT* ($R^2 = 0.98$)	Hidden Layer Values	(27, 27, 27, 27)
	Learning Rate	0.1
	Learning Rate Type	Adaptive
	Activation Function	ReLU
	Solver	LBFGS
	L2 Regularization	0.01

4 CONCLUSION

It can be concluded that the goals of the paper were successfully achieved, with high precision models being found to regress the value of end-effector grip robustness. The questions posed in the introduction may be answered as follows, based on the findings of the research:

- The MLP algorithm can be applied to successfully regress the value of the grip robustness from the used dataset, using measured values of position, velocity, and torque.
- The same can be done without the position input data being used, under the condition that the dataset normalization has been applied.
- The best architectures tend towards the larger end of the tested values, except when the dataset normalization is applied - significantly lowering the number of hidden neurons in the best model.
- Quantifiable change in the resulting quality is seen with the normalization being applied, as well as the change in the size of the model needed to regress the robustness of the robotic grasper grip on the used dataset.

Acknowledgments

This research has been (partly) supported by the CEEPUS network CIII-HR-0108, European Regional

Development Fund under the grant K.01.1.1.01.0009 (DATACROSS), project CEKOM under the grant KK.01.2.2.03.0004, CEI project "COVIDAi" (305.6019-20), project Metalska jezgra Čakovec (KK.01.1.1.02.0023) and University of Rijeka scientific grant uniri-tehnic-18-275-1447.

5 REFERENCES

[1] Welleweerd, M. K., Siepel, F. J., Groenhuis, V., Veltman, J., & Stramigioli, S. (2020). Design of an end-effector for robot-assisted ultrasound-guided breast biopsies. *International journal of computer assisted radiology and surgery*, *15*(4), 681-690. https://doi.org/10.1007/s11548-020-02122-1

[2] Mucchiani, C. & Yim, M. (2020, May). A Novel Underactuated End-Effector for Planar Sequential Grasping of Multiple Objects. *2020 11 International Conference on Robotics and Automation (ICRA)*, *11*, 8935-8941. https://doi.org/10.1109/icra40945.2020.9197380

[3] 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*, *17*(2). https://doi.org/10.1177/1729881420908076

[4] Baressi Šegota, S., Lorencin, I., Anđelić, N., Mrzljak, V., & Car, Z. (2020). Improvement of Marine Steam Turbine Conventional Exergy Analysis by Neural Network Application. *Journal of Marine Science and Engineering*, *8*(11), 884. https://doi.org/10.3390/jmse8110884

[5] Lorencin, I., Anđelić, N., Šegota, S. B., Musulin, J., Štifanić, D., Mrzljak, V., Car, Z. et al. (2020). Edge Detector-Based Hybrid Artificial Neural Network Models for Urinary Bladder Cancer Diagnosis. *Enabling AI Applications in Data Science*, 225-245. https://doi.org/10.1007/978-3-030-52067-0_10

[6] Van Vuuren, J. J., Tang, L., Al-Bahadly, I., & Arif, K. M. (2020). A 3-Stage Machine Learning-Based Novel Object Grasping Methodology. *11 Access*, *8*, 74216-74236. https://doi.org/10.1109/access.2020.2987341

[7] De Coninck, E., Verbelen, T., Van Molle, P., Simoens, P., & Dhoedt, B. (2020). Learning robots to grasp by demonstration. *Robotics and Autonomous Systems*, *127*, 103474. https://doi.org/10.1109/iros40897.2019.8967638

[8] Song, Y., Gao, L., Li, X., & Shen, W. (2020). A novel robotic grasp detection method based on region proposal networks. *Robotics and Computer-Integrated Manufacturing*, *65*, 101963. https://doi.org/10.1016/j.rcim.2020.101963

[9] Cupcic, U. (2017). *Grasping Dataset*. Kaggle.

[10] Shadow Robot Company (2016). *The Shadow Smart Grasping System.*

[11] Reddy, G. T., Bhattacharya, S., Ramakrishnan, S. S., Chowdhary, C. L., Hakak, S., Kaluri, R., & Reddy, M. P. K. (2020, February). An ensemble based machine learning model for diabetic retinopathy classification. *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)*, *11*, 1-6. https://doi.org/10.21203/rs.3.rs-47495/v1

[12] Shao, J., Hu, K., Wang, C., Xue, X., & Raj, B. (2020). Is normalization indispensable for training deep neural network? *Advances in Neural Information Processing Systems*, *33*. https://doi.org/10.24843/lkjiti.2020.v11.i03.p01

[13] Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning*. New York: Springer series in statistics. https://doi.org/10.1007/b94608

[14] Liu, Y., Xu, W. J., Ma, J., Alzahrani, F., & Hobiny, A. (2020). A new photosensitive neuron model and its dynamics. *Frontiers of Information Technology & Electronic Engineering*, *21*, 1387-1396. https://doi.org/10.1631/fitee.1900606

[15] Chaudhary, A. S. & Chaturvedi, D. K. (2020). Effects of Activation Function and Input Function of ANN for Solar Power Forecasting. *Advances in Data and Information Sciences*, 329-342. https://doi.org/10.1007/978-981-15-0694-9_31

[16] Kružić, S., Musić, J., Kamnik, R., & Papić, V. (2020). Estimating Robot Manipulator End-effector Forces using Deep Learning. *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, *11*, 1163-1168.

[17] Car, Z., Baressi Šegota, S., Anđelić, N., Lorencin, I., & Mrzljak, V. (2020). Modeling the spread of COVID-19 infection using a multilayer perceptron. *Computational and mathematical methods in medicine*, *2020*. https://doi.org/10.1155/2020/5714714

[18] Nimier-David, M., Speierer, S., Ruiz, B., & Jakob, W. (2020). Radiative backpropagation: an adjoint method for lightning-fast differentiable rendering. *ACM Transactions on Graphics (TOG)*, *39*(4), 146-1. https://doi.org/10.1145/3386569.3392406

[19] Ede, J. M. & Beanland, R. (2020). Adaptive learning rate clipping stabilizes learning. *Machine Learning: Science and Technology*, *1*(1), 015011. https://doi.org/10.1088/2632-2153/ab81e2

[20] Martinez-Ledezma, J. A., Barron-Zambrano, J. H., Diaz-Manriquez, A., Elizondo-Leal, J. C., Saldivar-Alonso, V. P., & Rostro-Gonzalez, H. (2020). Versatile implementation of a hardware-software architecture for development and testing of brain-computer interfaces. *International Journal of Advanced Robotic Systems*, *17*(6), 1729881420980256. https://doi.org/10.1177/1729881420980256

[21] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Duchesnay, E. et al. (2011). Scikit-learn: Machine learning in Python. *The Journal of machine learning research*, *12*, 2825-2830.

[22] Itano, F., de Sousa, M. A. D. A., & Del-Moral-Hernandez, E. (2018, July). Extending MLP ANN hyper-parameters Optimization by using Genetic Algorithm. *2018 International Joint Conference on Neural Networks (IJCNN)*, 1-8. https://doi.org/10.1109/IJCNN.2018.8489520

[23] Massaoudi, M., Refaat, S. S., Chihi, I., Trabelsi, M., Oueslati, F. S., & Abu-Rub, H. (2020). A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for Short-Term Load Forecasting. *Energy*, *214*, 118874. https://doi.org/10.1016/j.energy.2020.118874

[24] Awad, Z., Akel, R., Maalouf, N., & Elhajj, I. H. (2020, October). Terrain Classification for Bipedal Robots: A Comparative Study. *2020 10th Institute of Electrical and Electronics Engineers International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 61-66. https://doi.org/10.1109/CYBER50695.2020.9279169

[25] Ugartemendia, A., Rosquete, D., Gil, J. J., Diaz, I., & Borro, D. (2020). Machine Learning for Active Gravity Compensation in Robotics: Application to Neurological Rehabilitation Systems. *11 Robotics & Automation Magazine*, *27*(2), 78-86. https://doi.org/10.1109/MRA.2020.2978484

[26] Lorencin, I., Baressi Šegota, S., Anđelić, N., Blagojević, A., Šušteršić, T., Protić, A., Car, Z. et al. (2021). Automatic Evaluation of the Lung Condition of COVID-19 Patients Using X-ray Images and Convolutional Neural Networks. *Journal of Personalized Medicine*, *11*(1), 28. https://doi.org/10.3390/jpm11010028

[27] Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., & Levine, S. (2021). How to train your robot with deep reinforcement learning: lessons we have learned. *The*

International Journal of Robotics Research, 0278364920987859. https://doi.org/10.1177/0278364920987859

[28] Zhang, H., Zhan, B., Pan, F., & Luo, W. (2020). Determination of soluble solids content in oranges using visible and near infrared full transmittance hyperspectral imaging with comparative analysis of models. *Postharvest Biology and Technology*, *163*, 111148. https://doi.org/10.1016/j.postharvbio.2020.111148

[29] Anđelić, N., Baressi Šegota, S., Lorencin, I., Jurilj, Z., Šušteršič, T., Blagojević, A., Car, Z. et al. (2021). Estimation of COVID-19 Epidemiology Curve of the United States Using Genetic Programming Algorithm. *International Journal of Environmental Research and Public Health*, *18*(3), 959. https://doi.org/10.3390/ijerph18030959

[30] Nagelkerke, N. J. (1991). A note on a general definition of the coefficient of determination. *Biometrika*, *78*(3), 691-692. https://doi.org/10.1093/biomet/78.3.691

Contact Information:

Sandi BARESSI ŠEGOTA, mag. ing. comp.
(Corresponding Author)
Faculty of Engineering, University of Rijeka,
Vukovarska 58, 51000 Rijeka
E-mail: sbaressisegota@riteh.hr

Nikola ANĐELIĆ, mag. ing. mech.
Faculty of Engineering, University of Rijeka,
Vukovarska 58, 51000 Rijeka
E-mail: nandelic@riteh.hr

dr. sc. Zlatan CAR, dipl. ing.
Faculty of Engineering, University of Rijeka,
Vukovarska 58, 51000 Rijeka
E-mail: car@riteh.hr

dr. sc. Mario Šercer, dipl. ing.
Razvojno - edukacijski centar za metalsku industriju Metalska jezgra Čakovec,
Bana Josipa Jelačića 22 D, 40 000 Čakovec
E-mail: mario.sercer@mev.hr