Journal of Process Management and New Technologies
Vol. 13, Issue 1-2, 2025, pp. 81-101

OPTIMIZING MULTILAYER PERCEPTRON NEURAL NETWORK HYPERPARAMETERS

Ivana PETKOVSKI[1*]

[1]*Mathematical Institute of the Serbian Academy of Sciences and Arts, Belgrade, Serbia, ivana993@turing.mi.sanu.ac.rs*

Abstract: *Predictions across multiple disciplines rely on the efficacy of fundamental artificial neural networks, such as multilayer perceptron (MLP). Optimization is a key step in improving predictive performance of these models. In this study, a global nonlinear neural model was developed to predict the impact of using the digital technology in society, the economy and public administration on economic development. By conducting 17 experiments on the basic MLP neural network, authors investigated the effects of modifying the network architecture, learning rate and type of activation function. Standard measures of model errors and coefficient of determination were employed as criteria to prioritize configurations using the PROMETHEE II multi-criteria approach. The results reveal that models featuring two hidden layers, reduced learning speed and adequate activation functions achieve optimal performance with* $MSE_{16}=0.012$, $RMSE_{16}=0.110$, $MAPE_{16}=12.186$ *and* $R^2_{16}=0.719$. *Conversely, too complex models complicate the learning process and lead to imprecise predictions as in the case of* $MSE_5=0.019$, $RMSE_5=0.138$, $MAPE_5=17.225$ *and* $R^2_5=0.559$. *The results indicate the importance of adjusting the neural network hyperparameters to the nature of the research problem. Additionally, the study reveals the important role of MCDM in choosing the most adequate configuration when considering diverse criteria with different targets.*

Keywords: *Multilayer perceptron neural network, hyperparameters optimization, PROMETHEE II, technology adoption, economy*

Type of paper: Original scientific paper
Received: 19.05.2025
Accepted: 17.06.2025
Available online: 21.06.2025
DOI: 10.5937/jpmnt13-58965

1. Introduction

The role of artificial intelligence (AI) in contemporary approaches for data analysis, development of intelligent systems and automation is indispensable. AI facilitates the rapid execution of complicated tasks with minimal resource usage and great calculation precision (Yu & Zhu, 2020). AI supports human-driven decision-making by contributing valuable information derived from large data sets and forecasting future values or events (Chander & Das, 2024). One of the widely recognized AI models is the artificial neural network (ANN) model known as the multi-layer perceptron (MLP). As one of the basic types of machine learning, MLP NN belongs to the group of supervised learning models (Zhang et al., 2022). This group of models is

[*] Corresponding author

generated based on given input and output values that learn patterns of regularity, enabling them to predict future values (Zhang et al., 2022). Their significance is found in the ability to model the nonlinear and often complex relationships that accompany dynamic and uncertain systems (Cihan, 2025; Soares et al., 2025).

The success of MLP NN prediction models primarily requires quality and comprehensive data (Zhang et al., 2022). Afterwards, the choice of NN hyperparameters is crucial for the accuracy of the predicted values (Cros et al., 2025; Soares et al., 2025). The process of adjusting the hyperparameters within the NN is called optimization and it is applied to those parameters of the neural network that are not subject to changes in the learning process of the machine learning (ML) model (Yu & Zhu, 2020). As part of the optimization, numerous changes are made regarding the distribution of data sets used by the NN for training and testing, the selection of the activation function, the structure of the network or the speed of the learning process (Bekdaş et al., 2025; Shiomi et al., 2025). Inadequate selection of these hyperparameters potentially raises two of the most significant problems of neural networks. These include overlearning or overtraining and underlearning or undertraining of the neural network as pointed out by Cros et al. (2025). Overtraining of the model results in acceptance of all changes in the training data, whether significant or not, as important for the prediction of output values. In that case, the MLP NN model achieves ideal performance on the training data, but due to its inability to distribute important and unimportant information in the data, it achieves poor results on the test data. On the other hand, the lack of learning of MLP NN is the reverse problem because due to a smaller set of training data or insufficient capacity of the neural network, the model does not have the ability to learn even general regularities from the data. In both cases, the consequences are the same, poor generalization of the model and low predictive performance (Cros et al., 2025).

In the early research on the topic of optimization of neural network hyperparameters, experiments were conducted using the trial and error heuristic method (Bekdaş et al., 2025). The application of this method is recommended when working with a smaller data set and in the experimental stages of model development (Yu & Zhu, 2020). Optimization can also be done by applying the experience of researchers such as Rodriguez-Galiano et al. (2014) and (2015) recommend. However, in modern literature, various deterministic and stochastic approaches are known for solving optimization problems through developed algorithms (Ran et al., 2024). Among the most famous are Bayesian optimization, grid search, random search, particle swarm optimization (PSO), generic algorithm (GA), Gray wolf optimization (GWO) and others (Yu & Zhu, 2020; Al-Zaidawi & Çevik, 2025; Shiomi et al., 2025). The reason for switching to automatic optimization of hyperparameters is the availability of large data sets that need to be analyzed in a short time (Al-Zaidawi & Çevik, 2025; Cihan, 2025).

To avoid the use of overfitting or underfitting MLP NN models, the accuracy of the predictive values is checked. The verification is performed using standard measurements that calculate the error between the set and calculated values (Liyew et al., 2025). Ran et al. (2024) emphasizes that the use of individual measures can lead to confusion due to overtraining or undertraining of the model, so he advises the use of various measures, including mean square error (MSE), root mean square error (RMSE), mean absolute percentage error (MAPE), mean absolute error (MAE) and the coefficient of determination R2, which measures the predictive power of the model (Abdurrakhman et al. 2025; Liyew et al., 2025). The difference in the application of these metrics is in the procedure measurements because certain measurements rely on squared errors and absolute errors (MSE, RMSE and MAE), percentage errors (MAPE) and correlation (R), which further determines their effectiveness (Jierula et al., 2021).

In the automatic optimization process, what the algorithms lack is the influence of human knowledge in the domain of the research problem, which could condition the choice of hyperparameter values (Yu & Zhu, 2020). The results are based on the artificial network's ability to identify regularities among the data. So the solution to the problem comes down to the role of the human factor in conducting experiments under controlled conditions (Zhang et al., 2022). In addition to knowledge, the aim is to transparently build a model whose development dynamics could be implemented on similar problems (Ran et al., 2024). Therefore, this research aims to fill the research gap in the form of experimental development and optimization of MLP NN hyperparameters. An additional gap in the existing literature that is filled by this research concerns the improvement of the prioritization process of neural network models with the best results. In classic research, the selection of the best model is made by simply comparing the values of the error measures and the coefficient of determination (Abdurrakhman et al., 2025; Bekdaş et al., 2025; Liyew et al., 2025). In this research, we go a step further and use the PROMETHEE II multi-criteria decision-making method, which enables a comprehensive ranking of model performance based on criteria of a different nature. Although this method has been used in model optimization, its application in the context of MLP NN, with manual hyperparameter tuning, is rarely documented in the literature.

In accordance with emerging challenges, the main goal of this work is based on the optimization of the hyperparameters of the MLP neural network in order to improve the prediction performance. Despite the availability of tools for automatic optimization of hyperparameters, manual tuning was performed in all experiments as proposed by Yu and Zhu (2020) when working with small dataset. This strategy contributes to the understanding of model behavior and control over experimental settings. All changes are documented in a tabular format, ensuring transparency of the optimization and enabling further contributions from future scholars. A total of 17 experiments were conducted in order to indicate the effects that hyperparameters have on the output of neural network. Basically, all experiments deploy data on the use of technology in society, economy and public administration on the input data side and data on the achieved GDP per capita on the output data side. The aim of the specified experiments is to show the impact of changing hyperparameter configurations on the nonlinear model. To demonstrate the differences between the performance of the configurations, the calculation of errors and predictive power is performed, which are typically applied to the ML models. Given that these calculations are interconnected but also different in nature, it is suggested to use multi-criteria decision support. In this case the PROMETHEE II was used as a tool for the complete ranking of configurations. This approach yields transparent insights into the process of forming optimal configurations, as well as data-driven multi-criteria prioritization of these models. This methodological framework has a high application potential in solving problems in industry and the public sector because it adequately models and prioritizes complex nonlinear models. It is rarely found in literature where neural networks are applied.

2. Literature review

Optimization of hyperparameters within MLP neural networks has proven to be a key factor for improving the accuracy and stability of predictive models (Yu & Zhu, 2020). The analyzed papers indicate a broad array of optimization techniques - from classical algorithms to modern metaheuristic and probabilistic approaches (Al-Zaidawi & Çevik, 2025; Shiomi et al., 2025). Despite automated methods showing efficiency in application, the complexity of application in terms of demanding computing resources, hardware infrastructure and energy resources emerges (Çevik, 2025). In the following, a critical review of recent studies that applied different hyperparameter optimization strategies of MLP networks is given. Special attention is placed on the methods that compare the capability of predictive models.

Researchers have employed diverse solutions as hyperparameter optimization approaches. Heuristic methods employed in practice can be either deterministic or stochastic. The deterministic grid search method seeks for an optimal solution by calculating all possible combinations within the grid (Agrawal, 2024; Cihan, 2025). Conversely, the stochastic random search method identifies random combinations that are evaluated (Agrawal, 2024; Cihan, 2025). In addition to these, harmony search (HS), characterized by its stochastic nature and the Jaya algorithm (JA), classified as deterministic heuristic methods (Bekdaş et al., 2025), are also used in contemporary research. Then, probabilistic methods such as Bayesian optimization are in use, whose optimization is based on the probability of events (Tang et al., 2022; Agrawal, 2024; Atasever & Bozdağ, 2025; Cihan, 2025; Soares et al., 2025; Liyew et al., 2025). Unlike most automated methods, the Bayesian method limits the set of potential solutions within which the global optimum is classified (Soares et al., 2025). A large number of research studies emphasize the application of metaheuristic methods that are suitable for solving complex problems. In most cases, these algorithms are generated based on the functioning of biological and natural systems such as birds, ants, wolves, bats and flowers. Some of the most famous methods are Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Gray Wolf Optimizer (GWO), Simulated Annealing (SA), Harris Hawks Optimization (HHO), Bat Algorithm (BA) and Flower Pollination Algorithm (FPA) (Chander & Das, 2024; Ran et al., 2024; Abdurrakhman et al., 2025; Bekdaş et al., 2025; Shiomi et al., 2025). In addition to the use of standard methods, researchers also form hybrid models of metaheuristic methods such as World Cup Optimization with HHO and PSO-GWO (Al-Zaidawi & Çevik, 2025).

Modern research extensively introduces metaheuristic methods for hyperparameter optimization since the scholars point out their superiority in predictive accuracy. However, this approach entails high computational complexity, which limits its real-time application (Zghair & Issa, 2024). For example, although CNNs achieve high performance, their integration with Bayesian optimization remains a challenge due to high resource requirements (Tang et al., 2022). Stamoulis et al. (2018) propose a compromise solution, which is the application of Bayesian optimization and random search, with explicit restrictions on energy and memory consumption. This approach reduces the number of configurations that can be examined and narrows the search, but allows for more efficient execution in resource-limited environments. Although acknowledged as a success, it remains an open question whether the global optimal solution is really reached in this way. In contrast, classic methods such as manual searching, grid searching and random searching require fewer resources but often yield poor performances. Cihan (2025) and other authors (Liyew et al., 2025; Atasever & Bozdağ, 2025) highlight the advantages of Bayesian optimization even with less resource consumption but warn of the increasing number of hyperparameters and the need for significant expert knowledge during implementation. Due to these challenges, Al-Zaidawi & Çevik (2025) recommend the use of simpler and less computationally demanding algorithms, which provide a balance between efficiency and accuracy, especially in contexts where available resources are limited.

The application of hyperparameter optimization methods is performed over a wide range of machine learning models. In most cases optimization is carried out over neural models such as multilayer perceptron (MLP) (Jierula et al., 2021; Zghair & Issa, 2024; Abdurrakhman et al., 2025; Al-Zaidawi & Çevik, 2025; Shiomi et al., 2025), artificial neural network (ANN) (Agrawal, 2024; Chander & Das, 2024; Ran et al., 2024; Bekdaş et al., 2025; Soares et al., 2025) and more advanced convolutional neural network (CNN) models (Tang et al., 2022; Zghair & Issa, 2024; Al-Zaidawi & Çevik, 2025). In addition to these models, traditional machine learning models such as Random Forest (RF), Extreme Gradient Boosting (XGBoost), Light Gradient-Boosting

Machine (LightGBM), Elastic Net, Adaptive Boosting (AdaBoost), Gradient-Boosting Regressor (GBR), K-nearest Neighbors (KNN), and Decision Tree (DT) are used in practice (Cihan, 2025).

Depending on the machine learning model applied in the research, researchers approach the optimization of the appropriate hyperparameters. In the studies, optimization is carried out by applying different algorithms for training ML models, partitioning the data set, changing the number of neurons and hidden layers, activation functions in the output and hidden layers, as well as the learning rate. In the context of algorithms for ML model training, the quasi-Newton algorithm, conjugate gradient backpropagation with Powell-Beale updates, conjugate gradient backpropagation with Polak-Ribiere updates, the Levenberg-Marquardt algorithm, the resilient backpropagation algorithm, the scale conjugate gradient method, classical algorithms (ADAM, RMSprop) and evolutionary algorithms (simulated annealing, differential evolution) are most often applied (Jierula et al., 2021; Ran et al., 2024; Zghair & Issa, 2024; Abdurrakhman et al., 2025).

The ML model architecture is presented with different numbers of neurons and hidden layers within the network to improve the accuracy of the ML model (Jierula et al., 2021; Chander & Das, 2024; Abdurrakhman et al., 2025; Al-Zaidawi & Çevik, 2025; Shiomi et al., 2025). The commonly data partitioning allocates 70% of the data for training and 30% for testing (Chander & Das, 2024; Abdurrakhman et al., 2025). Some researchers also use a validation set to check the accuracy of the prediction model (Ran et al., 2024; Soares et al., 2025). The selection of an activation function depends on the nature of the research problem and the attributes of the data set. Tanh and sigmoid activation functions in the hidden layer and linear or softmax activation functions in the output layer are most often used (Ran et al., 2024; Abdurrakhman et al., 2025; Shiomi et al., 2025). A certain number of researchers optimize the learning rate as an important parameter for the stability and precision of the neural network (Tang et al., 2022; Al-Zaidawi & Çevik, 2025; Bekdaş et al., 2025).

Research innovations in the field of MCDM imply the integration of AI into the decision-making framework (Kumar, 2025). Within the domain of ML, MCDM methods have been employed to compare results of various AI algorithms (Onakpojeruo et al., 2025; Popović et al., 2025). The comparative analysis is not strictly connected with applying some of the MCDM techniques. The results can also be compared by non-conventional MCDM methods that support a multi-criteria approach (Cihan, 2025; Cros et al., 2025) or by directly comparing the error measure and the coefficient of determination of the AI algorithm (Abdurrakhman et al., 2025; Bekdaş et al., 2025; Liyew et al., 2025). Certain researchers use AI algorithms to forecast parameter values or optimal locations, which are subsequently used as alternatives in the ranking phase using MCDM methods (Talebi et al., 2025). In classification problems, the performance of different AI algorithms such as accuracy, precision, recall and f1 score are compared using MCDM (Al-Zaidawi & Çevik, 2025), but other prioritization methods out of the scope of MCDM remain in use (Jamasb et al., 2025; Tanveer et al., 2025). Despite the growing body of literature, there is no documented application of MCDM methods, with an emphasis on PROMETHEE II, on ranking AI algorithm performance metrics in the context of regression problems. The research gap points to the originality and scientific contribution of this research at the intersection of ML and decision support systems. PROMETHEE II enables the selection of a compromise solution in the experimental stages of testing neural network designs under complex decision environments.

3. Methodology and research data

Experimental analysis of neural network hyperparameters was performed using a multilayer perceptron. The theoretical framework of MLP NN is presented below to provide a more complete understanding of the influence of individual parameters on the development, learning and output of the neural network. In conjunction with it, the theoretical foundation of the multi-criteria decision-making approach PROMETHEE II was provided. This MCDM method will be used for the overall ranking of the experimental configurations.

3.1. Multilayer perceptron neural network (MLP NN)

A multilayer perceptron neural network (MLP NN) is a type of artificial neural network with a layered architecture consisting of an input layer, one or more hidden layers and an output layer (Sydenham & Thorn, 2005). The input layer consists of independent variables, i.e., input data utilized to estimate the target values (da Silva et al., 2017). The prediction is made for the dependent variables of the output layer of the neural network (da Silva et al., 2017).

The basic element in the hidden layer is an artificial neuron that is connected to the neurons of the next hidden layers or to the output layer (Abdurrakhman et al., 2025). The output of each artificial neuron is a result of two sequential operations (eq. 1-2) that perform linear transformation of the input data followed by a non-linear activation function. Mathematical formula has been adapted from Bishop (2006). The linear transformation of the weighted sum of inputs into the artificial neuron is presented as follows:

$$z_j^{(l)} = \left(\sum_{i=1}^{n} w_{ij}^{(l)} \times a_i^{(l-1)} \right) + b_j^{(l)} , \tag{1}$$

where $w_{ij}^{(l)}$ is the weight between neuron i in the layer (l-1) and neuron j in the layer l; $a_i^{(l-1)}$ is the activation of neuron i in the layer (l-1) and b_j^l is the bias of neuron j in the layer l.

The outcome of linear transformation is subjected to an activation function (σ), which introduces non-linearity in the model. The exact mathematical expression is presented in the equation as follows (Bishop, 2006):

$$a_j^{(l)} = \sigma(z_j^{(l)}) . \tag{2}$$

The outcome of the preceding artificial neuron is connected with neurons in the subsequent layer (Abdurrakhman et al., 2025). The connection is based on the chosen activation function that perceives the inner connections between the neurons. Based on these regularities, the neural network provides predicted values in the output layer. The algorithm is referred to as feedforward propagation. In the next step, the results of the prediction are being compared to the input (actual) values by the calculating function loss (L). The formula for calculating the function loss based on the mean square error is presented as follows (Bishop, 2006):

$$L = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{2} , \tag{3}$$

where y_i is the actual value of the input data in the neuron j and $\hat{y}_i$ is the predicted value of the neuron j.

When the function loss exhibits high values, the neural network employs the backpropagation algorithm. The algorithm performs a series of operations (eq. 4-9), first to calculate the value of gradient descent (GD) on the output layer (eq. 4) adapted from Nielsen (2015). Afterwards, the gradients are propagated on hidden layers (eq. 5), followed by weights (eq. 6) and biases (eq. 7). Subsequent mathematical formulations of eq. 4-7 are presented as follows:

$$\delta^{(L)} = \frac{\partial L}{\partial a^{(L)}} \circ \sigma'(z^{(L)}) \,, \tag{4}$$

where $\partial L/\partial a^{(L)}$ is the partial derivative of the error function related to activation in the output layer L, ∘ is the Hadamard's product and $\sigma' z^{(L)}$ is the derivative of the activation function in the output layer L.

$$\delta^{(l)} = (W^{(l+1)})^T \delta^{(l+1)} \circ \sigma'(z^{(l)}) \,, \tag{5}$$

where $W^{(l+1)T}$ is the transposed matrix of the weights in the layer (l+1), $\delta^{(l+1)}$ is the bias value in the layer (l+1) and $\sigma' z^{(l)}$ is the derivative of the activation function in the hidden layer l.

$$\frac{\partial L}{\partial W^{(l)}} = \delta^{(l)} (a^{(l-1)})^T \,, \tag{6}$$

where $\delta^{(l)}$ is the bias value in the layer l and $a^{(l-1)T}$ is the transposed matrix of the activations in the layer (l-1).

$$\frac{\partial L}{\partial b^{(l)}} = \delta^{(l)} \,, \tag{7}$$

where $\delta^{(l)}$ is the bias value in layer l.

The iterative procedure continues until the specified convergence threshold is achieved. The parameters are being updated using the following eq. 8 for weights update and eq. 9 for the bias update.

$$W^{(l)} = W^{(l)} - \eta \frac{\partial L}{\partial W^{(l)}} \,, \tag{8}$$

$$b^{(l)} = b^{(l)} - \eta \frac{\partial L}{\partial b^{(l)}} \,, \tag{9}$$

where $W^{(l)}$ are the updated weight values, η is the learning rate and $b^{(l)}$ are the updated bias values.

3.2. PROMETHEE II

The PROMETHEE II (Preference Ranking Organization METHOD for Enrichment Evaluation) method is part of the broader field of multi-criteria decision making (MCDM). It originally appeared in the literature by Jean-Pierre Brans in 1982, after which diverse modifications were propossed in the upcoming years (Brans & Mareschal, 1986; Parreiras & Vasconcelos, 2007; Behzadian et al., 2010; Arcidiacono et al., 2018; Doan & De Smet, 2018; Flachs & De Smet, 2025). Its benefit over previous PROMETHEE I method is the possibility of a complete ranking of alternatives in accordance with the set preferences of decision-makers (Behzadian et al., 2010; Pohl & Geldermann, 2024). Over time, PROMETHEE II has been reinforced by the development of software solutions (Pohl & Geldermann, 2024).

Let's observe a finite set of alternatives A={a_1, a_2, ..., a_n}, a set of criteria C={c_1, c_2, ..., c_m} and a set of criteria weights W={w_1, w_2, ..., w_m}, where $\sum_{j=1}^{m} w_j = 1$. The ratio of the alternative a_i to the criterion c_j is expressed as a function $f_j(a_i)$ given by the formula (Brans & Mareschal, 1986):

$$d_j(a_i, a_k) = f_j(a_i) - f_j(a_k) \,. \tag{10}$$

The symbol P_j represents the selected preference function between alternatives a_i and a_k, judged according to the criterion c_j (Flachs & De Smet, 2025). The value of H_j is a preference function that is determined based on the nature of the problem and can be level, Gaussian, usual, V-shaped, V-shaped with indifference criterion or U-shaped (Brans & De Smet, 2016).

$$P_j(a_i, a_k) = H_j(d_j(a_i, a_k)) . \tag{11}$$

The aggregation function of the preferences for alternative a_i in relation to the alternative a_k is shown by the following formula (Brans, Vincke & Mareschal, 1986):

$$\pi(a_i, a_k) = \sum_{j=1}^{m} w_j \cdot P_j(a_i, a_k) . \tag{12}$$

Positive preferential flow identifies how the evaluated alternative surpass remaining alternatives in the final ranking (Longsheng & Shah, 2025). It is calculated using the following formula (Brans, Vincke & Mareschal, 1986):

$$\phi^+(a_i) = \frac{1}{n-1} \sum_{k \neq i} \pi(a_i, a_k) . \tag{13}$$

In opposite, negative preferential flow reports how the observed alternative is outranked by other alternatives (Longsheng & Shah, 2025). This is calculated using the following formula (Brans, Vincke & Mareschal, 1986):

$$\phi^-(a_i) = \frac{1}{n-1} \sum_{k \neq i} \pi(a_k, a_i) . \tag{14}$$

The disparity between the positive and negative preferential flows is referred to as net preferential flow (Pohl & Geldermann, 2024). The formula for the net flow of preferences is given as follows (Brans, Vincke & Mareschal, 1986):

$$\phi(a_i) = \phi^+(a_i) - \phi^-(a_i) . \tag{15}$$

A higher difference value indicates an upper ranking of the alternatives.

3.3. Case study

The process of optimizing the hyperparameters of the MLP neural network was carried out on a dataset regarding the use of technology in society, business and e-government on economic development. The variables for measuring the impact of these three groups of indicators on the nation's gross domestic product (GDP) were collected from the latest report Network Readiness Index (Dutta & Lanvin, 2024) for the year 2024. This report measures the level of digital transformation in society through the NRI Index. The selected groups of data form a whole in terms of the participation of people, companies and state administration in the digital economy. The Table 1 presents more detailed information on individual variables, with definitions sourced from the NRI report (Dutta & Lanvin, 2024). In the NRI report, these three groups are classified as an integral part of the people pillar. In addition to this pillar, an integral part of NRI is technology, governance and impact, but their influence is omitted from further analysis because the research is focused on the influence of participants in the digital economy. GDP per capita was taken as a general indicator of economic development for the dependent variable. Data were collected for 133 economies of the world belonging to different levels of economic development.

Table 1. Variable definitions and units of measurement

Label	Variable name	Unit	Definition
Dependent variable:			
y	GDP per capita	$ per capita	Aggregate value of all goods and services produced within a country per capita.
Independent variable:			
x_1	Mobile broadband internet traffic within the country	NRI score 0-100	National broadband traffic volumes from at least 3G networks.
x_2	ICT skills in the education system		World Economic Forum survey on technology skills.
x_3	Use of virtual social networks		Percentage of the population using social media platforms.
x_4	Adult literacy rate		Percentage of the population able to read, write and understand simple statements.
x_5	AI talent concentration		Share of LinkedIn members with AI skills in respect to the entire population.
x_6	Firms with website		Percentage of businesses that have a website.
x_7	Number of venture capital deals invested in AI		Number of deals in respect to the AI venture capital investments.
x_8	Annual investment in telecommunication services		Annual investments by telecommunication provider.
x_9	Public cloud computing market scale		Market size of companies that provide cloud computing resources and services.
x_{10}	Government online services		E-government survey on user satisfaction with national websites and e-services.
x_{11}	Data Capabilities		The country's civil service skills to create, manage and use data.
x_{12}	Government promotion of emerging technologies		World Economic Forum survey on the government's promotion of AI among domestic businesses.
x_{13}	R&D expenditure by governments and higher education		Investments of government and higher institutions in R&D as a percentage of GDP.

Statistical analysis and data distribution analysis are given in the Table 2. High values of standard deviation and variance denote a significant difference in GDP per capita and data scatter. The standard deviation of variables x5 (AI talent concentration) and x7 (number of venture capital deals invested in AI) shows heterogeneity in relation to other variables while variable x8 (annual investment in telecommunication services) is compact. A variability is recorded in the values of variables x4 (adult literacy rate) and x6 (firms with website) while a more consistent range is observed in the values of variables x3 (use of virtual social networks) and x8 (annual investment in telecommunication services). Within the data distribution analysis variables x5 and x7 have the highest values of asymmetry and kurtosis, which shows a deviation from the normal distribution. Other variables tend to have a normal distribution.

Table 2. Descriptive statistics

	Mean	Std. Deviation	Variance	Skewness	Kurtosis
y	32782.86	30298.636	918007341.487	1.440	2.447
x_1	18.77	19.817	392.700	1.612	2.911
x_2	50.46	26.007	676.340	-.581	-.418
x_3	47.47	24.736	611.846	-.578	-.765
x_4	64.30	39.452	1556.466	-.807	-1.063
x_5	9.15	19.769	390.814	3.252	11.749
x_6	48.26	27.954	781.401	-.201	-1.046
x_7	12.62	23.446	549.717	2.744	7.360
x_8	49.44	20.119	404.792	-.783	1.415
x_9	22.06	19.849	393.998	1.126	1.046
x_{10}	61.60	23.225	539.378	-.478	-.616
x_{11}	28.11	25.583	654.515	.493	-.812
x_{12}	36.81	26.219	687.452	.415	-.541
x_{13}	15.34	19.262	371.035	1.860	3.678

The results of the normality test indicated that the data distribution does not meet the assumptions of normality. Thus, justifying the use of Spearman's correlation which is appropriate for non-parametric and non-linear relationships. The Table 3 shows the coefficients of the correlation ranks. A positive and statistically significant correlation value was recorded in the relationship between the dependent variable y and the independent variables x3 ($\varrho = 0.865$, $p = 0.01$), x10 ($\varrho = 0.748$, $p = 0.01$) and x13 ($\varrho = 0.739$, $p = 0.01$) listed in descending order. Variables such as the use of virtual social networks, government online services and R&D expenditure by governments and higher education have a dominant influence on GDP per capita and can be noted that their growth is simultaneous.

Table 3. Spearman`s correlation rang

Variable	x_{13}	x_{12}	x_{11}	x_{10}	x_9	x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	y
y	.739**	.465**	.417**	.748**	.630**	.459**	.615**	.549**	.636**	.013	.865**	.609**	.374**	1.000
x_1	.483**	.377**	.453**	.514**	.784**	.744**	.337**	.178*	.478**	.072	.465**	.331**	1.000	
x_2	.578**	.751**	.352**	.536**	.522**	.429**	.562**	.350**	.476**	-.172*	.593**	1.000		
x_3	.667**	.446**	.484**	.698**	.624**	.467**	.528**	.505**	.553**	.111	1.000			
x_4	-.134	-.078	.067	.111	-.134	-.088	-.165	-.048	-.248**	1.000				
x_5	.734**	.381**	.405**	.676**	.659**	.541**	.649**	.515**	1.000					
x_6	.536**	.210*	.331**	.574**	.440**	.260**	.518**	1.000						
x_7	.655**	.476**	.364**	.718**	.520**	.347**	1.000							
x_8	.550**	.387**	.415**	.525**	.834**	1.000								
x_9	.667**	.458**	.547**	.705**	1.000									
x_{10}	.688**	.500**	.557**	1.000										
x_{11}	.360**	.304**	1.000											
x_{12}	.414**	1.000												
x_{13}	1.000													

4. Research results and discussion

Hyperparameter optimization in the case study aims to find the best-performing prediction model that most accurately forecasts the impact of technology use on economic development. The hyperparameter tuning is carried out via an iterative search and evaluation of various parameters within the framework of the MLP neural network algorithm. During this process, different configurations of the MLP neural network are tested to tune the parameters for optimal algorithm results. The optimization process was supported by SPSS 17 software. For the purposes of hyperparameter optimization, 17 MLP neural network configurations were formed. The summarized configuration hyperparameter settings are given in the Table 4.

Table 4. Hyperparameter settings

Configuration	Data partition		Optimization algorithm	Learning rate	Activation function		Number of hidden layers	Units in hidden layers	
	Training	Testing			Hidden layer	Output layer		I	II
1	72.9%	27.1%	GD	0.4	Tanh	Identity	1	7	
2	80.5%	19.5%	GD	0.4	Tanh	Identity	1	6	
3	55.6%	44.4%	GD	0.4	Tanh	Identity	1	2	
4	74.4%	25.6%	GD	0.4	Tanh	Identity	1	10	
5	72.9%	27.1%	GD	0.4	Tanh	Identity	1	20	
6	60.9%	39.1%	GD	0.4	Tanh	Identity	2	5	4
7	64.7%	35.3%	GD	0.4	Tanh	Tanh	2	5	4
8	74.4%	25.6%	GD	0.4	Tanh	Sigmoid	2	6	5
9	73.7%	26.3%	GD	0.4	Sigmoid	Identity	2	6	5
10	72.2%	27.8%	GD	0.4	Sigmoid	Tanh	2	6	5
11	70.7%	29.3%	GD	0.4	Sigmoid	Sigmoid	2	6	5
12	66.9%	33.1%	GD	0.01	Sigmoid	Sigmoid	2	5	4
13	63.9%	36.1%	GD	0.1	Sigmoid	Sigmoid	2	5	4
14	69.9%	30.1%	GD	0.2	Sigmoid	Sigmoid	2	6	5
15	66.9%	33.1%	GD	0.01	Tanh	Sigmoid	2	5	4
16	72.9%	27.1%	GD	0.1	Tanh	Sigmoid	2	6	5
17	69.2%	30.8%	GD	0.2	Tanh	Sigmoid	2	6	5

The basic parameters that are fixed in these configurations are:

- Normalized data - the difference in the units of measurement of the dependent and independent variables requires data normalization to obtain stable results and values in a specific range (Cihan, 2025). Normalized data omit the prevailing effects of outlier values in the dataset (Singh & Singh, 2020),
- Optimization algorithm - gradient descent (GD),
- Stopping rule - maximum steps without decrease in error is 1,
- Maximum time for training - 15 minutes to prevent excessive duration of training,
- Input layer - 13 units of input variables,

– Error function - sum of squares.

Setting hyperparameters that are subject to variation in this case implies changing the following parameters:

– Data set partitions,
– Number of units in the hidden,
– Number of hidden layers,
– Activation function in the hidden layer and activation function in the output layer,
– Learning rate.

The Table 5 presents the performance of the generated configurations. Performance evaluation was executed by calculating MSE, RMSE, MAPE and R^2. These measures operate as baseline measures of the predictive accuracy of ML models. They have been selected based on their efficiency in evaluating ANNs or other ML methods as proposed by Jierula et al. (2021), Abdurrakhman et al. (2025) and (Liyew et al., 2025).

Table 5. Measuring configurations performance

Configuration	Error measures			
	MSE	RMSE	MAPE	R^2
conf_1	0.014	0.119	13.321	0.675
conf_2	0.015	0.121	13.495	0.662
conf_3	0.013	0.112	13.311	0.708
conf_4	0.014	0.117	13.605	0.683
conf_5	0.019	0.138	17.225	0.559
conf_6	0.017	0.130	15.980	0.609
conf_7	0.015	0.123	13.926	0.651
conf_8	0.012	0.111	12.975	0.714
conf_9	0.015	0.122	14.471	0.656
conf_10	0.017	0.129	15.489	0.617
conf_11	0.013	0.115	13.037	0.698
conf_12	0.013	0.112	12.897	0.709
conf_13	0.014	0.119	14.006	0.674
conf_14	0.017	0.130	15.549	0.613
conf_15	0.013	0.113	13.397	0.705
conf_16	0.012	0.110	12.186	0.719
conf_17	0.013	0.115	13.020	0.695

The first three experiments demonstrate the effect of altering the distribution of the data set on the training sample and the test sample. A standard data distribution of 70% training data and 30% testing data was taken as the reference point with automatic selection of activation function, number of hidden layers and number of units/neurons within layers. In conf_2 and conf_3 the data set distribution is set to 80% to 20% such in Cihan (2025) and 60% to 40% successively while maintaining other settings as in conf_1. All three configurations found one hidden layer with a varying number of neurons. The minimum number of neurons in the hidden layer is found in conf_3, which indicates that the 60%-40% distribution of the data leads to a limited training dataset. The maximum number of neurons in the hidden layer was found in conf_1, which indicates that this configuration formed a more complex MLP neural network model and explored the hidden connections among the data in more detail. The performance of conf_1 is minimally weaker compared to conf_3 ($MSE_1 > MSE_3$, $RMSE_1 > RMSE_3$, $MAPE_1 > MAPE_3$, $R^2_1 < R^2_3$) but due to the number of identified neurons in the hidden layer of the network, a distribution of 70% - 30% is chosen for subsequent experiments. The same data distribution is recorded in Abdurrakhman et al. (2025).

Within conf_4 and conf_5, changes were made in the number of neurons in the hidden layer. In the previous experiments, this number was calculated automatically, but in the next two it was set manually to 10 units (conf_4) and then to 20 units (conf_5). The objective is to examine the neural network's capacity to find hidden connections between data in more complex structures (Al-Zaidawi & Çevik, 2025). The performance of these configurations confirms the better results of conf_4 which offers optimal neural network complexity ($MSE_4<MSE_5$, $RMSE_4<RMSE_5$, $MAPE_4<MAPE_5$, $R^2_4>R^2_5$). The neural network in conf_5 detects even the smallest noises within the data, so the accuracy of the output prediction is questioned because the fluctuations in the data are learned more than the hidden connections between the data that are recognized as general regularities. The results are confirmed by Yu and Zhu (2020). Setting the neural network to predict the output based on the 20 units within the hidden layer led to poor configuration-specific predictive power because the model overfit the data and identified each noise as a pattern. The results are confirmed by Ran et al. (2024) and (Shiomi et al., 2025) who discussed that high increase in the number of neurons leads to overfitting.

Considering that configurations 1-5 automatically selected the number of hidden layers to be 1, in the next experiment (conf_6) the number of hidden layers is adjusted to 2. By adding a hidden layer in the neural network, the capacity of the network to recognize general regularities in the data is tested (Shiomi et al., 2025). The number of hidden layers refers to the network depth (Al-Zaidawi & Çevik, 2025). Conf_6 recognized 5 neurons in the first hidden layer and 4 layers in the second hidden layer, indicating the robust capacity of the neural network for data learning. In addition, conf_6 achieved a lower error level compared to conf_5 which was set to 20 units within a hidden layer, so it can be concluded that better performance was achieved with an additional hidden layer than with a large number of units within one hidden layer ($MSE_6<MSE_5$, $RMSE_6<RMSE_5$, $MAPE_6<MAPE_5$, $R^2_6>R^2_5$). This finding indicates the possibility of forming a neural network with two hidden layers and additional hyperparameter adjustments to achieve optimal results. The results are in line with Yu and Zhu (2020) suggestion to apply more hidden layers in order to obtain adequate solution with high accuracy. Accordingly, the hyperparameter number of hidden layers was set to a value of 2 for all further experiments.

The next hyperparameter that is subject to variation is the type of activation function in the hidden and output layers of the network. In the initial configurations 1-7, by automatically selecting the types of activation functions, the hyperbolic tangent is selected for the hidden layer of the neural network, while the identity type is selected for the output layer. In order to test the accuracy of the prediction of the configuration on the change of the form of the data coming out of the hidden and output layers, the types of activation functions were set. According to the optimal performance, conf_8 and conf_11 stand out. In conf_8, the activation function was modified in the output layer, while in conf_11, the activation function was changed in both layers. The hyperbolic tangent activation function processes data in the range -1 to 1, which improves the efficiency of handling non-linear and negative data, while the sigmoid activation function processes data on values in the range 0 to 1 and facilitates the classification of data or the probability of a scenario (Bekdaş et al., 2025; Shiomi et al., 2025). Therefore, in other experiments, the activation functions set in conf_8 and conf_11 are retained.

The final adjustments of the hyperparameters were made by changing the neural network's learning rate. The value of learning rate directly impacts the speed of identifying the optimal weights in the neural network (Yu & Zhu, 2020; Al-Zaidawi & Çevik, 2025). In the previous configurations, the standard value of 0.4 was taken. In order to attain a more gradual and stable learned neural network, configurations 12-17 were formed and learning rates were adjusted as proposed by Yu and Zhu (2020). In configurations 12-14, the sigmoid activation function was selected in both layers. The performance of conf_12 with 0.01 learning rate indicates the optimal values of neural network error measurement with the highest coefficient of determination compared to other increases in learning rate including the results of conf_11 ($MSE_{12}<MSE_{11}$, $RMSE_{12}<RMSE_{11}$, $MAPE_{12}<MAPE_{11}$, $R^2_{12}>R^2_{11}$). In configurations 15-17, a hyperbolic tangent function is set in the hidden layer and a sigmoid in the output layer. Performance measurements indicate that configurations with relatively lower prediction errors were created compared to configurations 12-14. The performance of conf_16 with a learning rate of 0.1 indicates the highest coefficient of determination (R^2_{16}=0.719) among all other configurations with the lowest error values (MSE_{16}=0.012, $RMSE_{16}$=0.110, $MAPE_{16}$=12.186). When compared, conf_16 achieved a 16% predictive power upgrade over conf_5 that is the lowest-ranked configuration.

The Figure 1 illustrates a comparison of the actual and predicted values of the dependent variable *y*. Actual values indicate the existence of outliers in the GDP per capita. By examining the data, extremely high values were revealed for the individual cases of Singapore, Ireland, Luxembourg and Qatar, which affects the reduction of the predictive power of the MLP model in all experiments.

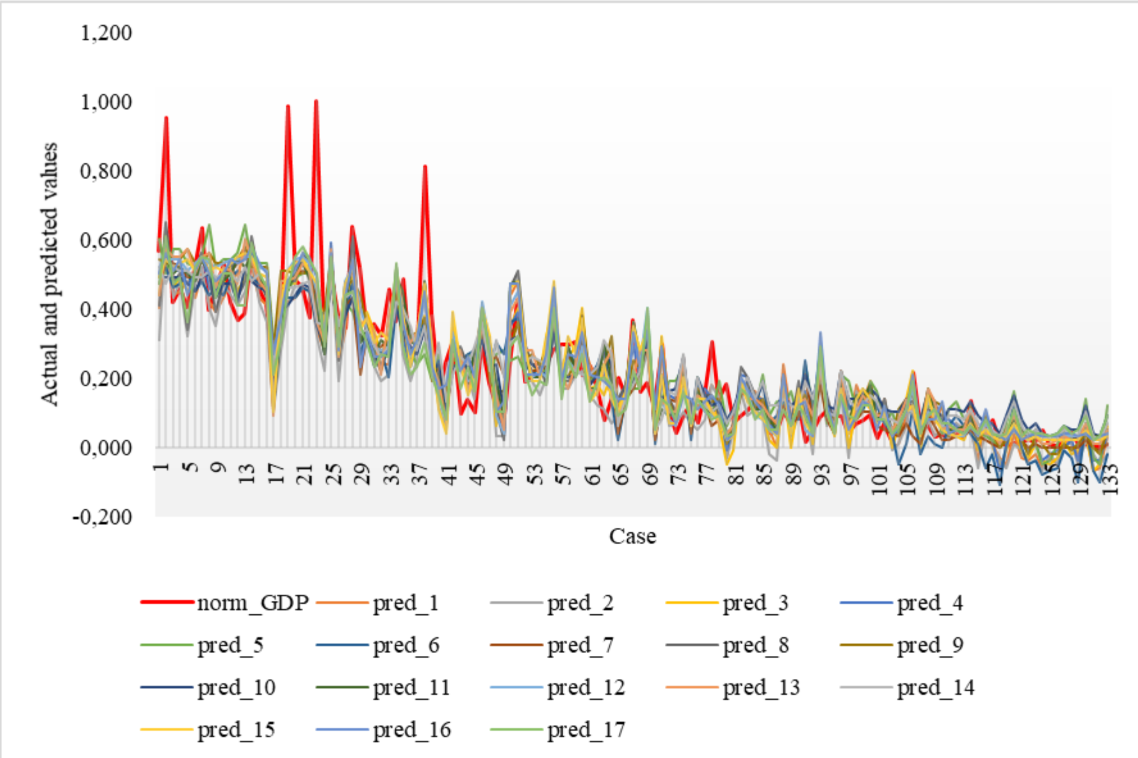

Figure 1. Accuracy of GDP per capita predicted values

The highest predictive power is shown by conf_16 with 71.9% correctly forecasted oscillations in the output data, which implies that this model successfully explains more than two-thirds of the variance. The high coefficient of determination of this configuration is supported by the high number of individual neurons in the hidden layers of the network. This result was achieved through hyperparameter optimization in which the learning rate was reduced to 0.1 compared to the standard value in the initial experiment of 0.4 and the activation function in the output layer was set to sigmoid. Marginally inferior performance is shown by conf_8, conf_3 and conf_15. The most inaccurate estimates are obtained through conf_5, in which 20 units are set within the hidden layer of the network with a percentage of determination coefficient of 55.9%. This shows that increasing the complexity of the neural network does not necessarily increase the prediction accuracy.

The graph in Figure 2 shows the patterns of normalized significance of independent variables obtained through several experiments. The independent variable x_3 (use of virtual social networks) plays a key role in forecasting the fluctuations of GDP according to the use of technology. A high signal level of the x_3 variable with the dependent variable determines the setting of a high value of the synaptic weight and dominance in the configurations. The variables x_1 (mobile broadband internet traffic within the country) and x_2 (ICT skills in the education system) have the greatest fluctuations in synaptic weights and are sensitive to hyperparameter optimization. The importance of their relationship with the dependent variable may depend on the modeling context. Variables x_6 (firms with website), x_8 (annual investment in telecommunication services) and x_{11} (data capabilities) show the least significance for the predictive power of the MLP model. In the configurations conf_6 and conf_12, a more even distribution of the significance of the variables is achieved. In these experiments, a second hidden layer was introduced and the learning rate was reduced, favoring the significance distribution. These two hyperparameters settings facilitate the neural network's stability to learn and extract latent relationships among the data, even the weak ones and reduce the dominance of individual variables. Significant unevenness is present in conf_8 and conf_11. In these configurations, the types of activation functions in the hidden and output layers are set. The sigmoid function, which was used in the output layer in both experiments, according to its properties, affects the reduction of the importance of neurons with negative values by assigning them values around zero and thus determines the signal flow to the output layer.

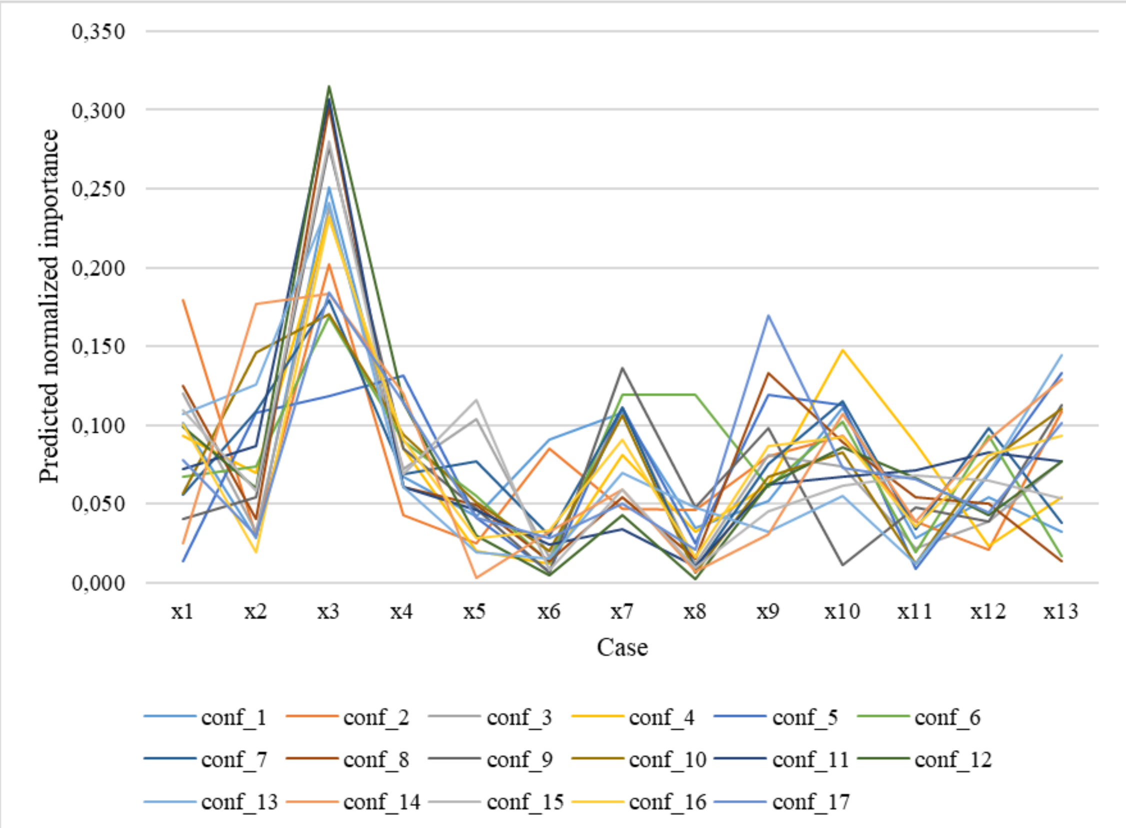

Figure 2. Predicted independent variable importance across configurations

Considering that the measures of configuration errors are limited to MSE, RMSE, MAPE and the coefficient of determination R^2 as generally accepted for evaluating the performance of ML models, a simultaneous application of multi-criteria decision support is employed. The initial matrix of multi-criteria decision-making is given in the Table 6. MSE, RMSE and MAPE are classified as cost criteria because the aim is to obtain the lowest possible values of these measures, while R^2 is defined as a benefit criterion because a higher value of this measure is more suitable for prioritization. In order to eliminate subjectivity in the preferences of these four criteria, the equivalent weight coefficients were adopted as the average value.

Table 6. Baseline settings for MCDM matrix

	MSE	RMSE	MAPE	R^2
Criterion type	Cost	Cost	Cost	Benefit
Preference function	Usual	Usual	Usual	Usual
Importance weights (w_j)	0.250	0.250	0.250	0.250

The authors selected the PROMETHEE II method as the most relevant for multi-criteria prioritization in the context of this research. The application decision is based on its ability to consider the nature of the criteria in relation to the prioritization objective and the possibility of complete prioritization of configurations based on the calculated net flow index where each configuration is compared against one another and contributes objectivity to the decision-making process. By comparing TOPSIS, VIKOR, COPRAS, MULTIMOORA and PROMETHEE ranking results Zlaugotne et al. (2020) recommend the use of a single MCDM for prioritization rather than multiple methods as the latter may lead to inconsistent results. The Table 7 gives the final PROMETHEE II calculations including positive flow, negative flow, their difference and the final rank of each configuration.

Table 7. PROMETHEE II-based preference rankings

Alternative	Positive flow (Phi+)	Negative flow (Phi-)	Net flow (Phi)	Rank
conf_1	0.003	0.002	0.0015	9
conf_2	0.002	0.002	0.0000	11
conf_3	0.005	0.000	0.0048	4
conf_4	0.003	0.001	0.0019	8
conf_5	0.000	0.014	-0.0144	17
conf_6	0.000	0.009	-0.0081	16
conf_7	0.002	0.003	-0.0016	12
conf_8	0.006	0.000	0.0058	2
conf_9	0.002	0.003	-0.0017	13
conf_10	0.001	0.007	-0.0067	14
conf_11	0.005	0.000	0.0041	6
conf_12	0.006	0.000	0.0054	3
conf_13	0.003	0.002	0.0006	10
conf_14	0.000	0.008	-0.0072	15
conf_15	0.005	0.000	0.0044	5
conf_16	0.007	0.000	0.0073	1
conf_17	0.004	0.001	0.0038	7

The results of prioritization using the PROMETHEE II method confirmed conf_16 as the highest-ranked alternative or configuration. This configuration achieved the lowest MSE, RMSE, and MAPE values and the highest R^2 value, thereby confirming its position as the most suitable hyperparameter setting of the MLP neural network. In the experiment, the hyperbolic tangent and sigmoid activation functions were set in the hidden and output layers respectively. Hyperbolic tangent in the hidden layer allows a stable learning rate since it distributes signals in the range -1 to 1, symmetrically around 0. Thus leaving more space for information flow even negative values unlike the sigmoid function that suppresses negative values importance. However, using sigmoid in the output layer is effective since its predicted values are approximately 0 or 1 and support the values of the dependent variable that are normalized. The generation of two hidden layers was chosen to contribute to the capacity of the neural network to learn regularities from the data. Additionally, the learning rate was lowered to 0.1, thus allowing for a greater number of optimal solutions to be considered. The weakest position was achieved by conf_5 with the highest values of MSE, RMSE and MAPE and the lowest value of R^2. This outcome is attributed to setting too high value for the number of units in the hidden layer, which led to the complexity of the network and hindered the ability to capture the hidden relationships in the data oscillations. The use of the identity function in the output layer failed to model non-linear relationships presented in this dataset. The results from using multi-criteria decision support match the error measurements from the neural network experiments. The strategy contributed to a comprehensive ranking of configurations taking into account multiple and diverse criteria such as accuracy-based criteria (MSE, RMSE, MAPE) and explanatory power (R^2). This approach facilitates the process of selecting the most convenient MLP neural network in a scenario where all performance measures are important and trade-offs must be taken into account.

5. Conclusion

Optimizing the hyperparameters of the neural network is a key step in improving the model's predictive power. This study develops an initial model that explores the impact of digital technology use in society, economy and public administration on economic power. A global non-linear model nature was formed based on a sample of 133 countries. In order to improve its efficiency 17 experimental configurations with different hyperparameters were conducted.

The original MLP model was autonomously generated based on SPSS generic settings. Modifications were made to the parameters of the training and testing data set partition, the number of hidden neurons within the hidden layer, the number of hidden layers, the activation functions in the hidden and output layers and the learning rate. In order to track the optimization effects, standard error measures were calculated to evaluate the accuracy of the ML model. The PROMETHEE II multi-criteria method was used as a decision support tool in choosing the optimal configuration. The chosen MCDM method enables the complete prioritization of configurations according to criteria that are different and often opposing criteria, such as minimization of model errors and maximization of the explanatory power of the model. The findings indicate that the most suitable results for the defined problem arise from more complex model architectures. These models explore variations among the data at a higher level and reveal data patterns. The concept of complex models includes those with a minimum of two hidden layers and an ideal number of artificial neurons integrated within. With a lower value of the learning rate, these models achieve excellent results due to their capacity and adequate speed to detect important changes in the data that would increase the predictive accuracy. The simple neural network models shown in the first experiments with one hidden layer and a high learning rate can skip significant information in the data because they learn quickly and have less capacity than models with an additional hidden layer. At the same time, models with a complex architecture such as an excessive number of artificial neurons in hidden layers can cause disorder within the neural network. This leads misidentifying of trivial information in the data as important. Furthermore it causes high values of prediction errors and a reduced coefficient of determination.

In the present scenario the influence of independent variables that have a strong correlation with the dependent variable has a dominant effect on the neural network. As a consequence the learning process assigns high values of synaptic weights which are propagated through the network. This problem can be managed by applying different activation functions that transmit the signals via the network in different forms. This allows a more uniform distribution of weighting factors among artificial neurons and enables data with lower or negative values to influence the network's prediction output.

A major limitation of this study is reflected in the sample size and the manual approach of hyperparameter optimization. This situation creates opportunities for future research to focus on applying more advanced strategies for automatic hyperparameter optimization on a larger dataset. Further research would aim to improve the performance and stability of the MLP model.

References

Abdurrakhman, A., Sutiarso, L., Ainuri, M., Ushada, M., & Islam, M. P. (2025). A Multilayer Perceptron Feedforward Neural Network and Particle Swarm Optimization Algorithm for Optimizing Biogas Production. Energies, 18(4), 1002.

Agrawal, P. (2024). A Survey on Hyperparameter Optimization of Machine Learning Models. In 2024 2nd International Conference on Disruptive Technologies (ICDT) (pp. 11-15). IEEE.

Arcidiacono, S. G., Corrente, S., & Greco, S. (2018). GAIA-SMAA-PROMETHEE for a hierarchy of interacting criteria. European Journal of Operational Research, 270(2), 606-624.

Atasever, Ü. H., & Bozdağ, A. (2025). Carbon footprint mapping of urban areas in Türkiye using hyperparameter-optimized machine learning techniques. International Journal of Environmental Science and Technology, 1-24.

Behzadian, M., Kazemzadeh, R. B., Albadvi, A., & Aghdasi, M. (2010). PROMETHEE: A comprehensive literature review on methodologies and applications. European journal of Operational research, 200(1), 198-215.

Bekdaş, G., Aydın, Y., Işıkdağ, U., Nigdeli, S. M., Hajebi, D., Kim, T. H., & Geem, Z. W. (2025). Shear Wave Velocity Prediction with Hyperparameter Optimization. Information, 16(1), 60.

Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning. Springer.

Brans, J. P., & De Smet, Y. (2016). PROMETHEE methods. Multiple criteria decision analysis: state of the art surveys, 187-219.

Brans, J. P., Vincke, P., & Mareschal, B. (1986). How to select and how to rank projects: The PROMETHEE method. European journal of operational research, 24(2), 228-238.

Chander, G. P., & Das, S. (2025). A hybrid decision support system in medical emergencies using artificial neural network and hyperbolic secant grey wolf optimization techniques. Cluster Computing, 28(1), 43.

Cihan, P. (2025). Bayesian Hyperparameter Optimization of Machine Learning Models for Predicting Biomass Gasification Gases. Applied Sciences, 15(3), 1018.

Cros, D., Rouan, L., Navratil, D., Tchounke, B., Leroy, N., Le Squin, S., Ulfah, N., Nodichao, L., & Beurier, G. (2025). Optimizing artificial neural network methodologies for enhanced genomic predictions: a case study with oil palm (Elaeis guineensis) data.

Da Silva, I. N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L. H. B., dos Reis Alves, S. F. (2017). Artificial neural network architectures and training processes (pp. 21-28). Springer International Publishing.

Doan, N. A. V., & De Smet, Y. (2018). An alternative weight sensitivity analysis for PROMETHEE II rankings. Omega, 80, 166-174.

Dutta. S,. & Lanvin. B. (2024). Network readiness index. Building a Digital Tomorrow: Public-Private Partnerships for Digital Readiness. Available at: https://networkreadinessindex.org/

Flachs, A., & De Smet, Y. (2025). Inverse optimization on the evaluations of alternatives in the PROMETHEE II ranking method. Omega, 103325.

Jamasb, B., Khayami, S. R., Akbari, R., & Taheri, R. (2025). An Automated Framework for Prioritizing Software Requirements. Electronics, 14(6), 1220.

Jierula, A., Wang, S., Oh, T. M., & Wang, P. (2021). Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data. Applied Sciences, 11(5), 2314.

Khaleq Zghair, N. A., & Issa, A. S. (2025). Development of an Intelligent System Based on Deep Neural Network Models with Advanced Algorithms for Hyper-parameter Tuning and Weight Updates Across Diverse Datasets. International Journal of Intelligent Engineering & Systems, 18(1), 847-859.

Kumar, R. (2025). A Comprehensive Review of MCDM Methods, Applications, and Emerging Trends. Decision Making Advances, 3(1), 185-199.

Liyew, C. M., Di Nardo, E., Ferraris, S., & Meo, R. (2025). Hyperparameter Optimization of Machine Learning Models for Predicting Actual Evapotranspiration, 5123630.

Longsheng, C., & Shah, S. A. A. (2025). A multi-method framework integrating ANP-ANN and PROMETHEE-GAIA for Circular Economy performance assessment: A case study of China. Journal of Cleaner Production, 145311.

Nielsen, M. A. (2015). Neural networks and deep learning (Vol. 25). Determination press.

Onakpojeruo, E. P., Uzun, B., & Al-Turjman, F. (2025). Reinforcement Learning Models in Stock Trading. NEU Journal for Artificial Intelligence and Internet of Things, 4(1), 72-85.

Parreiras, R. O., & Vasconcelos, J. A. (2007). A multiplicative version of Promethee II applied to multiobjective optimization problems. European Journal of Operational Research, 183(2), 729-740.

Pohl, E., & Geldermann, J. (2024). PROMETHEE-Cloud: A web app to support multi-criteria decisions. EURO Journal on Decision Processes, 12, 100053.

Popović, S., Viduka, D., Bašić, A., Dimić, V., Djukic, D., Nikolić, V., & Stokić, A. (2025). Optimization of Artificial Intelligence Algorithm Selection: PIPRECIA-S Model and Multi-Criteria Analysis. Electronics, 14(3), 562.

Qasim Jebur Al-Zaidawi, M., & Çevik, M. (2025). Advanced Deep Learning Models for Improved IoT Network Monitoring Using Hybrid Optimization and MCDM Techniques. Symmetry, 17(3), 388.

Ran, L., Yan, G., Goyal, V., Abdullaev, S., Alhomayani, F. M., Le, L. T., Ayadi, M., Alharbi, F. S., Alzubaidi, L. H., Albaijan, I., & Khan, B. (2024). Advancing solar thermal utilization by optimization of phase change material thermal storage systems: A hybrid approach of artificial neural network (ANN)/Genetic algorithm (GA). Case Studies in Thermal Engineering, 64, 105513.

Rodriguez-Galiano, V. F., Chica-Olmo, M., & Chica-Rivas, M. (2014). Predictive modelling of gold potential with the integration of multisource information based on random forest: a case study on the Rodalquilar area, Southern Spain. International Journal of Geographical Information Science, 28(7), 1336-1354.

Rodriguez-Galiano, V., Sanchez-Castillo, M., Chica-Olmo, M., & Chica-Rivas, M. J. O. G. R. (2015). Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. Ore geology reviews, 71, 804-818.

Shiomi, K., Sato, T., & Kita, E. (2025). Comparison of Particle Swarm Optimization Algorithms in Hyperparameter Optimization Problem of Multi Layered Perceptron. Computer Assisted Methods in Engineering and Science.

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. Applied Soft Computing, 97, 105524.

Soares, R. C., Silva, J. C., de Lucena Junior, J. A., Lima Filho, A. C., de Souza Ramos, J. G. G., & Brito, A. V. (2025). Integration of Bayesian optimization into hyperparameter tuning of the particle swarm optimization algorithm to enhance neural networks in bearing failure classification. Measurement, 242, 115829.

SPSS Inc. Released 2008. SPSS Statistics for Windows, Version 17.0. Chicago: SPSS Inc.

Sydenham, P. H., & Thorn, R. (Eds.). (2005). Handbook of measuring system design (Vol. 3). John Wiley & Sons.

Talebi, M., Rezvanjou, S., Ghafourian, H., & Shoushtari, F. (2025). Machine Learning-Driven Multi-Criteria Decision-Making Models for Optimizing Sustainable Infrastructure Deployment: Electric Vehicles and Renewable Energy Systems, 5121862.

Tang, S., Zhu, Y., & Yuan, S. (2022). An adaptive deep learning model towards fault diagnosis of hydraulic piston pump using pressure signal. Engineering Failure Analysis, 138, 106300.

Tanveer, H., Adam, M. A., Khan, M., & Ali, M. (2024). Analyzing the Performance and Efficiency of Machine Learning Algorithms, such as Deep Learning, Decision Trees, or Support Vector Machines, on Various Datasets and Applications. The Asian Bulletin of Big Data Management, 3, 126-136.

Yu, T., & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. arXiv preprint arXiv:2003.05689.

Zhang, W., Gu, X., Tang, L., Yin, Y., Liu, D., & Zhang, Y. (2022). Application of machine learning, deep learning and optimization algorithms in geoengineering and geoscience: Comprehensive review and future challenge. Gondwana Research, 109, 1-17.

Zlaugotne, B., Zihare, L., Balode, L., Kalnbalkite, A., Khabdullin, A., & Blumberga, D. (2020). Multi-criteria decision analysis methods comparison. Rigas Tehniskas Universitates Zinatniskie Raksti, 24(1), 454-471.