Applied Artificial Intelligence
An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Optimizing Models for Sustainable Drilling Operations Using Genetic Algorithm for the Optimum ANN

Nikolaos Efkolidis, Angelos Markopoulos, Nikolaos Karkalos, César García Hernández, José Luis Huertas Talón & Panagiotis Kyratsis

To cite this article: Nikolaos Efkolidis, Angelos Markopoulos, Nikolaos Karkalos, César García Hernández, José Luis Huertas Talón & Panagiotis Kyratsis (2019) Optimizing Models for Sustainable Drilling Operations Using Genetic Algorithm for the Optimum ANN, Applied Artificial Intelligence, 33:10, 881-901, DOI: 10.1080/08839514.2019.1646014

To link to this article: https://doi.org/10.1080/08839514.2019.1646014

APPLIED ARTIFICIAL INTELLIGENCE
2019, VOL. 33, NO. 10, 881–901
https://doi.org/10.1080/08839514.2019.1646014

Optimizing Models for Sustainable Drilling Operations Using Genetic Algorithm for the Optimum ANN

Nikolaos Efkolidis [a], Angelos Markopoulos[b], Nikolaos Karkalos[b], César García Hernández[a], José Luis Huertas Talón[a], and Panagiotis Kyratsis [c]

[a]Department of Design and Manufacturing Engineering, University of Zaragoza, Zaragoza, Spain; [b]Section of Manufacturing Technology, School of Mechanical Engineering, National Technical University of Athens, Athens, Greece; [c]Department of Mechanical Engineering & Industrial Design, Western Macedonia University of Applied Sciences, Kozani, Greece

ABSTRACT

In the present study, Artificial Neural Network (ANN) approaches were adopted for the prediction of thrust force (Fz) and torque (Mz) during drilling of St60 workpiece, according to important cutting parameters such as cutting velocity, feed rate, and cutting tool diameter. During the setup of an ANN, some essential difficulties like the determination of network architecture, the determination of weight coefficients and the selection of training algorithm should be addressed. A combination of genetic algorithm and neural networks (GA-ANN) formulates those difficulties as an optimization problem and resolve it by the help of a suitable optimization method. Finally, a comparison between ANN with network architecture determined by a simple trial and error approach and ANN with architecture determined by a GA-ANN approach is conducted. The comparison of the models showed clearly that adopting genetic algorithm (GA) equals to the improvement of the efficiency of the network performance.

Introduction

The development of today's world has increased the need for the production of necessary products for the consumers. From the manufacturing point of view, this equals to the increased demand for raw materials and energy. Manufacturing operations is an important part of energy consumption during the whole life cycle of a product. Sustainable manufacturing is dealing with the efficiency of production processes. Many efforts have been done in order for manufacturing sustainability to be improved. Among these is the creation of mathematical models focused on the maximization of productivity and cost reduction by identifying crucial parameters and processes influencing manufacturing effectiveness (Sujova, Marcinekova, and Hittmar 2017; Vijayaraghavan and Castagne 2016). Drilling is one of the most widely used

CONTACT Nikolaos Efkolidis efkolidi@unizar.es Department of Design and Manufacturing Engineering, University of Zaragoza, Zaragoza, Spain

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaai.

machining processes in the industry. The high pressure of the competition makes time and quality of the process to be considered as very crucial factors for the success. For this reason, many researchers have dealt with the modeling of machining in general as well as with drilling in particular. Any optimization in the process directly equals to a greener machining. This paper deals with the determining of ANN parameters and improving the efficiency of network performance by adopting genetic algorithm (GA).

Literature Review

Drilling is one of the most widely used machining operations in the industry. Therefore, many researchers have used the application of ANN for the purpose of prediction and modeling manufacturing effectiveness by identifying crucial parameters and processes. Nassef et al. (2018) used an ANN during a drilling process of a glass using abrasive jet machining. The ANN focused on the development of a model of kerf taper as a function of the process parameters. Genetic algorithm used for the optimization of the model by identifying the conditions to minimize the kerf taper. They proved that the kerf taper is reducing by applying an axial feed to the nozzle so that the standoff distance is kept constant during the machining process. Hynes, Kumar, and Sujana (2017) developed an ANN predictive model for the bushing length in thermal drilling of galvanized steel. Bushing length is directly linked with the tapping process and input parameters of the process play a basic role in fastening galvanized steel. The maximization of the bushing length was done by the use of a genetic algorithm under constraint limits. Kannan et al. (2015) tried to face out problems in drilling operations such as poor surface roughness and ovality. They developed Artificial Neural Network modeling technique and Genetic Algorithm (GA) optimization technique for the drilling process of 6 mm hole in brass plate. They found that suitable parameters selection plays a vital role in the improvement of drilled holes quality.

Goyal and Dubey (2014) provided artificial neural network and genetic algorithm for the modeling and optimization of geometrical quality characteristics such as hole taper and circularity during Laser trepan drilling (LTD) of 1.6 mm thick Inconel718 superalloy sheet. They verified that higher values of laser pulse frequency and trepanning speed in the used range had resulted in more circular holes with reduced taper. Kilickap and Huseyinoglu (2010), under the investigation of the influence of the cutting parameters on burr height produced when drilling AISI 304 stainless steel, developed an application of response surface methodology (RSM) and genetic algorithm (GA) for the selection of the optimum combination values of those parameters. The results showed that the minimum burr height was obtained at lower cutting speed and feed rates while at higher point angle. Even in Geological

Engineering, Khandelwal and Armaghani (2016) developed a multiple regression, artificial neural network (ANN) and hybrid genetic algorithm (GA)-ANN models for the estimation of a drilling rate index (DRI) prediction model based on rock material properties. The comparison of the three different models showed that the hybrid GA-ANN technique is the much better predictor of DRI compared to other developed models.

Artificial Intelligence Methods

Artificial Neural Networks

Artificial neural networks (ANN) constitute one of the most important Artificial Intelligence methods. They are inspired by the actual function of neural networks in nature, e.g. the neural system of living organisms. Biological neural networks are complex networks composed of a large number of neurons connected by synapses (Basheer and Hajmeer 2000). The neurons receive inputs via synapses and produce a suitable output after they are activated. The same principles regulate the function of ANNs; usually, ANNs contain a number of artificial neurons organized in a number of connected layers. In the case of Multi-layer Perceptron (MLP), the most common network architecture comprises an input layer, an output layer and a hidden layer between them.

In this type of network, data enters from the input layer and is processed towards the output layer (feed-forward). The basic element of ANN is the determination of weighting coefficients, or weights, which are related to the artificial synapses between neurons of different layers (Jain, Mao, and Mohiuddin 1996). Several methodologies have been proposed for the determination of network architecture, such as empirical rules or more organized schemes (Curteanu and Cartwright, 2011). For each neuron, a summation operation is first performed for the inputs which are multiplied by the appropriate weights, and then the output is produced with the use of an "activation function". The activation function should produce output in the range [0, 1] and exhibit a behavior comparable to that of the activation of a biological neuron. For that reason, functions with a sigmoid shape such as the hyperbolic tangent are preferred.

In order to establish a reliable correlation between input and output quantities, a training process consisting of three parts is realized. More specifically, during this process the appropriate weight values are determined based on data from input/output pairs. The data are usually divided into three sets: the training, validation and test sets. During training, weights are adjusted in order to learn the given input/output pairs. During validation, the weights are adjusted until the decrease of error stops, and finally, the test data are employed to determine the generalization ability of the networks, i.e. the capability to predict the response to unknown inputs. Training process is conducted by comparing the actual and predicted outputs and propagating the error from the output to the input layer (back-propagation of error) (Murata, Yoshizawa, and Amari 1994). Suitable termination criteria regulate the end of the training, after several iteration or *epochs*. The prediction ability of the error can be usually evaluated by the Mean Square Error (MSE):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{Y}_i - Y\right)^2 \tag{1}$$

where n is the number of input or output data, $\widehat{Y}$represents the predicted data and Y represents the actual output data. Furthermore, the correlation coefficient R or the coefficient of determination R^2 can also be used.

Genetic Algorithm

Genetic Algorithm (GA) constitutes one of the most commonly used methods for solving optimization problems. This method belongs to the Evolutionary Algorithms, which is the oldest and most important category of bio-inspired artificial intelligence methods. It was proposed originally in by Holland in the 1970s (Holland 1973) and then was further advanced by researchers such as Goldberg (Goldberg 1989). Since its invention, GA was successfully employed in many scientific disciplines and a considerable number of variants and combinations with other methods were proposed to increase its efficiency.

The basic idea behind GA is the use of the process of natural selection as a metaphor, in order to create an algorithm which can efficiently derive solutions close to the optimum ones. Natural selection is a biology term, which is related to the process of biological evolution; according to this theory, several biological mechanisms are responsible for the survival or modification of various characteristics of living beings during the timeline of evolution. In GA, various terms are related to biological ones, such as the population or the selection, crossover and mutation operators. The candidate solutions are conceived as chromosomes and the individual traits of the chromosome are the genes. The whole process is driven by the theory of survival of the best individuals, as evaluated by the fitness function, which reflects the goal of the optimization problem.

Thus, GA algorithm starts by determining an initial population of individuals- candidate solutions. Then, an iterative process begins, and the initial individuals are evolved towards more improved ones. At first, the fitness of the individuals is evaluated and using a selection process the individuals produce new offspring by combining their features; moreover, the features of the offspring can be determined at random by mutation. Apart from these basic operators, other strategies are also employed, such as the elitism, which aims for the preservation of the best solutions for a number of generations, or the use of multiple populations which evolve in parallel and eventually exchange features between them.

GA-ANN Method

As aforementioned, some of the fundamental problems appearing in the case of neural networks include: the determination of network architecture for each problem, the determination of weight coefficients as well as the selection of training algorithm. For that reason, it is possible to formulate this problem as an optimization problem and resolve it by the aid of a suitable optimization method. In the relevant literature, several researchers have employed a combination of genetic algorithm and neural networks, denoted in the present work as GA-ANN method, in various scientific disciplines.

Arifovic and Gençay (2001) investigated the use of genetic algorithm in order to determine the optimum neural network architecture and other network parameters and compared the effectiveness of this approach with models based on Schwarz and Akaike information criteria. Correa and Gonzalez (2011) employed two different algorithms, namely Genetic Algorithms and Binary Particle Swarm Optimization (BPSO) to optimize the architecture of a MLP neural network. The design variables for the optimization problem included not only the number of hidden layers and neurons, but also the type of activation function for hidden and output layers and the bias terms. They concluded that the optimization approach was superior to that of the default process of network architecture determination and that this approach led to a solution very close to the global optimum. Boithias, El Mankibi, and Michel (2012) also used an approach for neural network optimization using GA for prediction of indoor discomfort and energy consumption. In their model, they used parameters relevant to network architecture, training process and also included additional variables, which indicated if any of the input variables were unnecessary for the ANN model. This approach showed considerable accuracy and it was proposed that the derived model could be used for on-line controller setting purposes.

Idrissi et al. (2016) proposed a multi-objective optimization approach, with a view to determine both network architecture and suitable values for network weights. Their objective function had two goals: the minimization of the number of hidden layers and neurons and the minimization of MSE, while the constraints of the problem were the existence of at least one hidden layer and the removal of neurons of a hidden layer if it was not used. They concluded that their approach was capable of predicting results from three datasets with significant accuracy. Ul Islam et al. (2014) employed GA to optimize ANN architecture by determining which connections of neurons would be active or not. They employed an evaluation function based on MSE, minimization of number of connections and neurons. Their results indicated sufficiently low MAPE values, showing the efficiency of their approach. Benardos and Vosniakos (2007) developed a GA-ANN with novel criteria, in order to determine network architecture in a sufficiently reliable way. Their objective function included terms related to training error, generalization error, network architecture and solution space consistency and was helpful to obtain relatively low training and generalization errors.

Jeong, Min, and Kim (2012) employed a Generalized Additive Model (GAM) and GA algorithm to fine-tune the architecture of ANN model and the decay coefficient. After proper determination of the input factors for the model and suitable initial values were conducted, GA was employed to provide optimum architecture and decay coefficient value. The proposed approach was compared to other classification methods, as well as to a non-tuned NN and it was observed to outperform them. Khorani, Forouzideh, and Nasrabadi (2011) compared several optimization methods for the training of MLP. Using optimization algorithms, the network architecture and weights could be determined and the objective function was the minimization of MSE. They found that the combination of Imperialist Competitive Algorithm and GA was the optimum approach.

In their review paper, Ojha, Abraham, and Snášel (2017) thoroughly presented the advances conducted in the field of metaheuristic design of neural networks. They pinpointed that the main categories of metaheuristic-based training of neural networks are: connection weight optimization, architecture optimization, node optimization, and learning rule optimization, as well as their combinations. In their work, they did not only present details about the GA-ANN approaches, but provide useful information about future challenges in this field. Zhang and Wang (2008) presented a GA-ANN approach, with which they optimized the initial interconnecting weights and thresholds of an MLP network and observed its superiority against ordinary ANN networks.

Methodology

In this study, a St60 workpiece (150 mm×150 mm×15 mm) was placed in HAAS VF1 CNC machining center for the drilling operations. During the drilling experiments cutting forces were measured by a Kistler four components dynamometer type 9123 with all the appropriate accessories. The dynamometer signals were processed via charge amplifiers and an A/D converter to a personal computer. Thrust force and torque measurements were displayed and analyzed in order to implement an early error detection strategy.

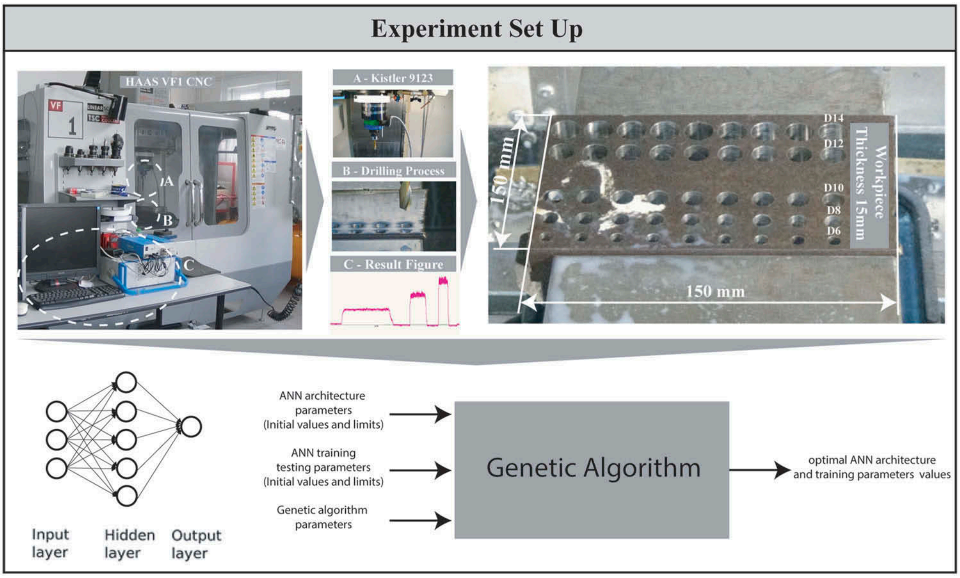


Figure 1. The workflow used for the research.

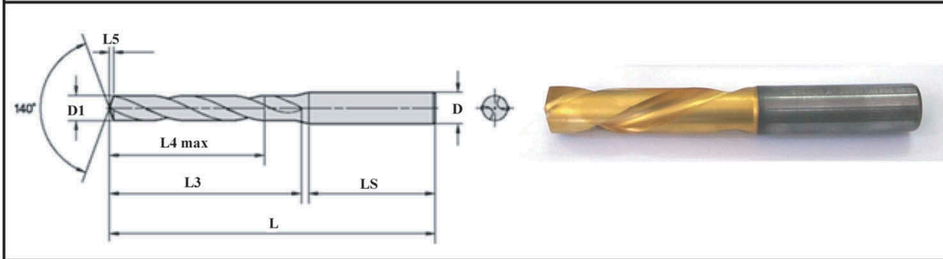
The cutting tools which used for the experiments were solid carbide drill tools (Kennametal – multilayer TiAlN-PVD-coated universal fine-grain grade) with diameters 6, 8, 10, 12, and 14 mm (Figure 2). The number of the total experiments was 45 as the full factorial combination of cutting speeds (10, 30, 50m/min), feed rates (0.05, 0.15 and 0.25mm/rev) and tool diameters were performed. The workflow of the research is depicted in Figure 1, and the cutting parameters, units, and notations are listed in Table 1.

The experimental results are illustrated in Figure 3. It presents the thrust force and the cutting torque measured, for all the cutting tools while all the combinations of the feed rates and cutting speeds were used. It is obvious that the thrust force and the cutting torque are directly related to different feed rates and cutting speeds. From the cutting tool point of view, according to the results, it is clear that when the tool diameter increases, both the thrust force and the cutting torque values are increased. The same is the case for the feed rate with respect to both the thrust force and the cutting torque. On the other side, cutting speed affects barely the experimental values. As a result, it seems that the importance of cutting speed is much lower than the importance of cutting tool diameter and feed rate, related to thrust force and cutting torque.

General Description

In the present paper, a GA-ANN approach is proposed to model the experimental data with increased accuracy. The investigation, concerning the

Cutting tools



Catalogue Number	B041A06000CPG	B041A08000CPG	B041A10000CPG	B041A12000CPG	B041A14000CPG
Grade	KC7325	KC7325	KC7325	KC7325	KC7325
(D1) drill diameter min (mm)	6	8	10	12	14
(L4) max drill depth (mm)	20	29	35	40	43
(L3) flute length (mm)	28	41	47	55	60
(LS) flute length (mm)	36	36	40	45	45
(D) shank diameter (mm)	6	8	10	12	14

Figure 2. Cutting tools analytical description.

Table 1. Cutting variables used for the experiments.

Parameters	Values
Cutting velocity (m/min) *V*	10, 30,50
Feed rate (mm/rev) *f*	0.05, 0.15, 0.25
Tool diameter (mm) *D*	6, 8, 10, 12, 14
Axial depth of cut (mm) ap	15
Workpiece dimension (mm)	150 × 150 × 15

efficiency of the proposed approach, will consist of several steps and it will contain a comparison with an ANN model, created using a simpler approach. Thus, at first, an investigation on the optimum architecture of ANN for the prediction of Fz force and Mz torque is conducted using a trial-and-error approach and an empirical rule, separately for Fz and Mz. Then, the proposed GA-ANN approach is employed. The developed model consists of several input variables, presented in Table 2 and the objective function consists of several terms, regarding the prediction accuracy of ANN. More specifically, two different GA-ANN models are examined, with one containing only the network architecture parameters as inputs and the second containing parameters concerning the training process as well. For each

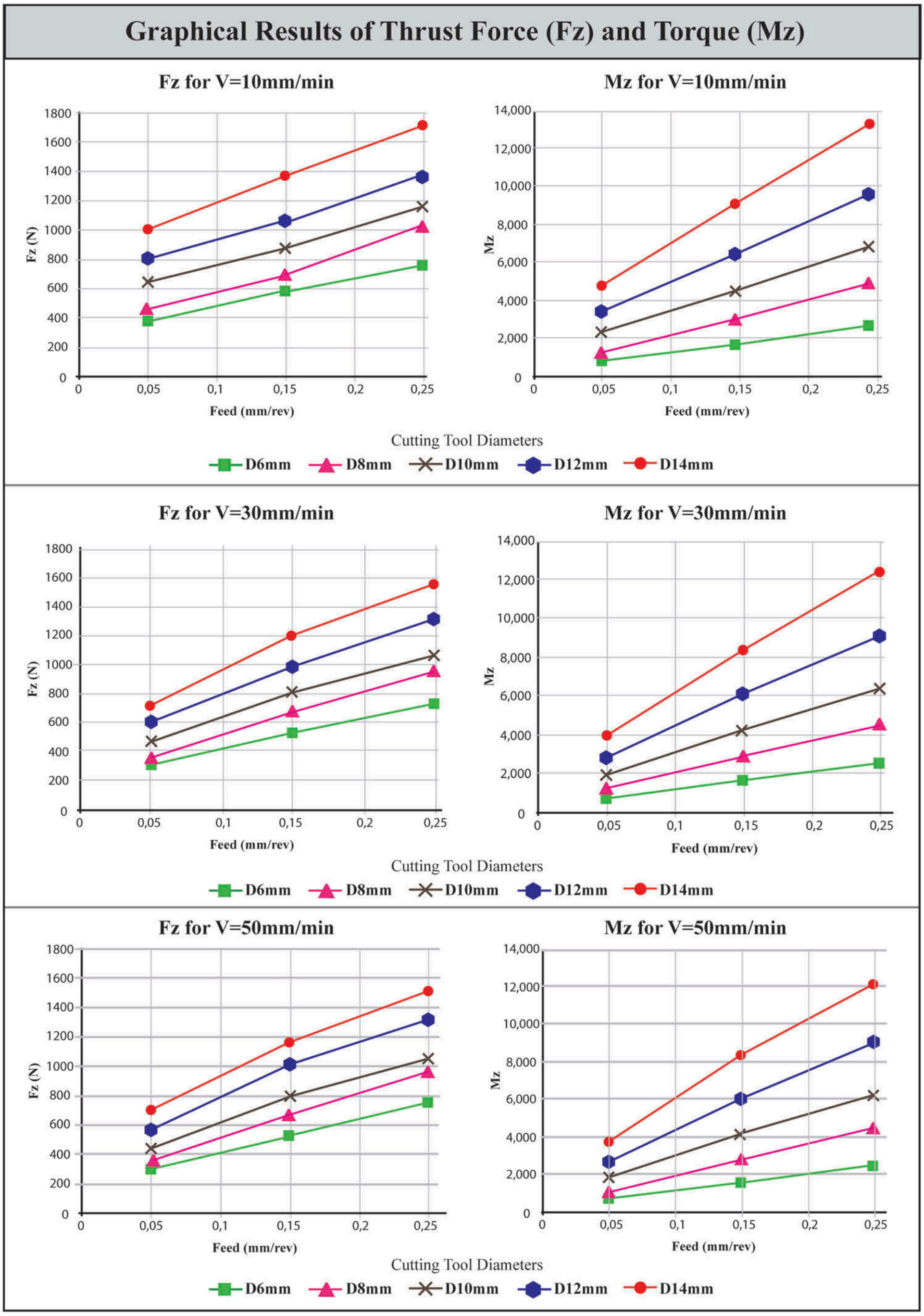


Figure 3. Results of the Fz and Mz for all the experiments conducted.

model, several cases (denoted hereafter as "scenarios") of optimization functions, presented in Table 3, are evaluated in order to determine also the overall best network. Every network has three inputs, namely cutting tool

Table 2. Design variables and their limits.

Variable	Lower bound	Upper bound
N_layers	1	2
Neurons1	2	15
Neurons2	2	15
TrainRatio	0.60	0.75
TestRatio	0.10	0.15
Train_algorithm	1	5
PerformFcn	1	2
Max_fail	6	15
Regularization	0.2	0.7
Gen_type	0	1

Table 3. Different optimization functions used in the present work.

Scenario No.	Objective function
1	MSE
2	0.5*MSE + 0.5 $(1/R^2)$
3	0.25*MSE + 0.25 *$(1/R^2)$ + 0.25 *t + 0.25 *MPE
4	0.20*MSE + 0.20 *$(1/R^2)$ + 0.10 *t + 0.50 *MPE
5	0.15*MSE + 0.15 *$(1/R^2)$ + 0.05 *t + 0.65 *MPE
6	0.10*MSE + 0.10 *$(1/R^2)$ + 0.025 *t + 0.775 *MPE

diameter, cutting speed and feed and one output, Fz or Mz. All ANN and GA-ANN models are developed in MATLAB.

Details for the Development of ANN Models

Simple ANN

As aforementioned, the trial-and-error approach for the optimum ANN architecture for the prediction of Fz and Mz will be based on an empirical rule for the determination of lower and upper limit for the number of hidden neurons. It is generally accepted (Zhang et al. 2012) that, for simpler problems with a small to medium dataset, a single hidden layer is sufficient and the range for the number of hidden neurons can be determined using the following formula:

$$N_h = \sqrt{N_{inp} + N_{out}} + a \qquad (2)$$

In the case examined in the present work, $N_{inp} = 3$, $N_{out} = 1$ and a is considered to vary between 0 and 10, so the range of hidden neurons number is 2 to 12. For each case, the networks are retrained 10 times in order to eliminate the influence of initial weight values to the result. Except for the number of hidden neurons, other ANN parameters are considered constant; training algorithm is Levenberg–Marquardt and early stopping technique is adopted to end training after MSE is not decreased after 6 consecutive epochs.

Results using the ordinary ANN approach are presented in Section 3.1. Although the performance of ANN for the determination of an optimum number of neurons was mainly assessed by MSE values, mean percentage error (MPE) is also calculated and used in the comparisons with GA-ANN approach. MPE can be calculated as follows:

$$\mathrm{MPE} = \frac{100\%}{n}\sum_{i=1}^{n}\frac{Y_i - \hat{Y}_i}{Y_i} \tag{3}$$

GA-ANN

The GA-ANN approach actually generalizes the process of determining the optimum network architecture for a given case. The main optimization problem is the minimization of prediction error and is achieved by altering the network architecture or other parameters. Thus, the objective function is related to error terms, whereas network architecture features, such as the number of hidden layers and hidden neurons (or also training parameters), are the design variables. The lower and upper limits for each design variable are presented in Table 2. The optimization process starts with random initial values, within the defined limits, for the design variables. At the end of the process, the best network determined has the optimum architecture or also optimum training parameters. For each scenario which was evaluated, five different runs of the GA-ANN were conducted and the best one was selected. The ANN models for the prediction of Fz and Mz were determined separately. For the genetic algorithm parameters, default values were assumed and all cases are run for a maximum of 500 generations.

As for the design parameters of the problem, the parameters that regulate ANN architecture are the *number of layers*, the *number of neurons in the first and the second hidden layer* (denoted as N_layers, Neurons1, and Neurons2 in Table 2). It was chosen that the number of layers should not exceed two, as in fact, it would be unnecessary to implement a three-layer ANN model for a medium-sized problem. Furthermore, the lower and upper limits for the number of neurons in each layer were chosen as 2 and 15, respectively, in order to be close to the limits predicted by the empirical Equation (2).

The two next parameters regulate the *ratio of data* (TrainRatio and TestRatio) reserved for training and testing procedure. The value of these parameters is important, as it will affect significantly the prediction accuracy. The default values were 0.7 and 0.15, respectively, but it was intended to investigate the optimum values of these parameters for the specific problem through an organized procedure.

The next parameter regulates the choice of *training algorithm* for the ANN. The choice of the training algorithm is also crucial for the neural network, as it has a direct impact on the efficiency of the network. Some algorithms perform better for smaller or larger network sizes and have also different memory requirements and scalability. In this work, five different training algorithms, namely BFGS quasi-Newton, Scaled Conjugate gradient, Conjugate gradient with Powell–Beale restarts (CGPB), Conjugate gradient with Polak-Ribière updates, and Levenberg–Marquardt, are tested. The choice of each training algorithm is encoded by integer numbers in the range 1–5.

As an additional parameter, it was chosen that the *performance function* (default:MSE) should be allowed to be changed between MSE and SSE. SSE is encoded as 1 and MSE as 2. The last parameters are related to the generalization capability of the network. The first of the two (max_fail) is related to the early stopping technique, as it regulates the allowed maximum number of epochs with no improvement in the performance function value and the second is related to the regularization procedure. For max_fail parameter, the lower bound is 6 (default value in MATLAB) and the upper is 15. As for the regularization parameter, it is allowed to vary in the range [0.2, 0.7]. If the value of Gen_type is 0, only early stopping technique is used, and if it is 1, regularization is also performed.

As for the scenarios, which are presented in Table 3, they reflect which is the objective function used in the present work. The first scenario is chosen to be the most common goal for neural network training, the minimization of MSE. The second scenario is a more complex one, and it takes into account the effect of correlation coefficient R as well, with equal weight. At last, the four other scenarios take into account all the possible performance and error indicators; the third one is considering equal weight for each parameter, namely MSE, R, MPE and time for training the neural network, but as the most effective goal is thought to be the minimization of minimum percentage error (Benardos and Vosniakos 2007), three different weight coefficients for MPE are tested, with subsequent alteration of the weights for the other parameters. It is to be noted that, when it is needed, scalarization was applied to ensure that the real weighting of each term will not be altered, due to large differences between the values of various constituents of the objective functions.

Results and Discussion

Results Using Simple ANN Approach

From the several runs conducted with one hidden layer and 2 to 12 neurons, results concerning prediction errors were obtained. The optimum network architecture is considered to be the one that minimizes MSE test, in order to obtain a model with sufficient generalization ability. The MSE test values for the best models of Fz and Mz are presented in Table 4. The minimum MSE value was attained with the ANN model with five neurons in the hidden layer for Fz and with the ANN model with seven hidden neurons for Mz. In Table 4, the values for R_{test} and MPE are presented, as well as the number of predicted values with the error between 5% and 10% (denoted as N_5) and the number of predicted values with error over 10% (denoted as N_{10}). It can be seen that

Table 4. Optimum network architecture and prediction accuracy indicators for the simple ANN approach.

Output variable	Number of hidden neurons	MSE_{test}	R_{test}	MPE	N_5	N_{10}
Fz	5	$8*10^{-4}$	98.5%	1.315%	11	4
Mz	7	$5*10^{-4}$	99%	−1.264%	3	4

R values are high, both for Fz and Mz models, MPE is low, but especially for Fz model, there exist several values with error more than 5%, something that could be ameliorated with the use of the GA-ANN approach.

Results Using GA/ANN

Model with Inputs Related Only to Network Architecture

At first, the results concerning the ANN model for Fz, as presented also in Table 5 and depicted also in Figure 4(a,b), will be discussed. For the cases performed by the use of the first objective function, the best network architecture was determined as 3–4-9–1. As it was expected, the goal of minimizing the MSE was proven effective and the total MSE was $7.05*10^{-5}$. Moreover, the total R was 99.97% and total MPE −0.098%, with all prediction errors lying below 5%. In the case where R was also included in the objective function, the results for the best network (3–6-1) indicate slightly worse values for MSE and R ($3.98*10^{-4}$ and 99.65%, respectively), whereas MPE value is significantly lower (0.008%). The same is observed in the case of the third scenario, where the MSE and R values are slightly worse, but the MPE value is very low, something that constitutes a strong indication that the change in error values occurs as a result of the minimization of all objective function terms, and not specifically one term. This is also evident at the last three scenarios, where the initial combination (scenario 4) exhibits slightly worse results, especially for MSE, but the results are ameliorated at the next two scenarios, as the weighting factor for MPE is larger. Another general conclusion is that, with the exception of scenario 2, in all other cases, the MPE error for all predicted values is below 5%, meaning that the GA-ANN approach manages not only to lower the average error, but keeps the individual values of error within acceptable limits. As for the training time, it varied between 0.04 and 0.11 seconds.

From the results, the best architecture can be also determined. As can be seen from Table 5, several networks have acceptable values of errors, but the

Table 5. Results using the GA-ANN approach for Fz model with inputs related to network architecture.

Scenario No.	Number of hidden layers	1^{st} layer neurons	2^{nd} layer neurons	MSE_{tot}	R_{tot}	MPE (%)	N_5	N_{10}
1	2	4	9	$7.05*10^{-5}$	99.98%	−0.0981	0	0
2	1	6	-	0.0004	99.65%	−0.0085	5	0
3	2	12	3	0.00379	99.09%	−0.0080	0	0
4	1	4	-	0.00124	99.74%	0.0136	0	0
5	1	8	-	0.00054	99.59%	−0.0040	0	0
6	1	5	-	0.00047	99.89%	−0.0036	0	0

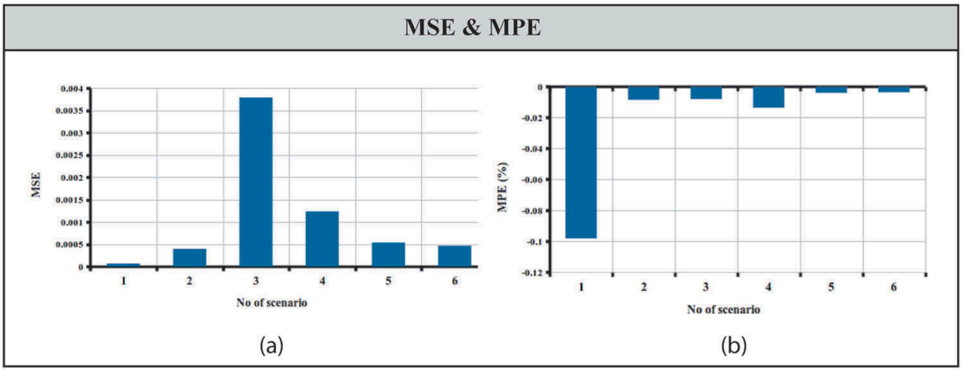


Figure 4. Results concerning: (a) MSE and (b) MPE for the best ANN models for Thrust Force according to each scenario.

network for the 6th scenario is superior, as it does not only have relatively low errors, but it also has the simplest architecture, something that leads to a quicker model and more capable to generalize (Ojha, Abraham, and Snášel 2017). It is also to be noted that, in the majority of the cases, the final network architecture consisted of only one hidden layer and in the most cases the number of hidden neurons was below eight, something that bears resemblance to the suggestions of the empirical rule and the results of simple ANN approach. However, as it will be later discussed, the GA-ANN approach provides a superior and more reliable way for the determination of an overall best performing architecture.

For the cases concerning Mz torque, for which results are presented in Table 6 and depicted in Figure 5(a,b), the predicted values obtained using the first objective function exhibited significantly low MSE value, high R value, and acceptable MPE value. The use of the second objective function led to the deterioration of these values, especially for MPE. However, the use of the most complex objective function led to an improvement for the MPE and from the 4th scenario to the 6th, the values of MSE and R were gradually ameliorated as well. All scenarios provided solutions with a single hidden layer as best and 4–6 hidden neurons. The solution of the 6th scenario can be considered as the best solution among them, as it exhibits the smaller MPE, large R_{tot} value and has the simplest architecture. Finally, the training time for these networks varied from 0.054 to 0.135 seconds.

Models with Additional Design Variables

In the cases presented in this subsection, the design variables of the optimization problem included not only the network architecture but several parameters related to the training procedure. Thus, conclusions for the optimum values of these parameters will be also drawn.

Table 6. Results using the GA-ANN approach for Mz model with inputs related to network architecture.

Scenario	Number of hidden layers	1^{st} layer neurons	2^{nd} layer neurons	MSE_{tot}	R_{tot}	MPE (%)	N_5	N_{10}
1	1	6	-	$4.2*10^{-6}$	99.99%	0.0521	0	0
2	1	6	-	$1.4*10^{-5}$	99.99%	−0.3488	0	0
3	1	5	-	$4.3*10^{-4}$	99.90%	0.0217	0	1
4	1	4	-	$4.8*10^{-5}$	99.97%	−0.0151	0	3
5	1	4	-	$4.9*10^{-5}$	99.76%	−0.0158	0	0
6	1	4	-	$5*10^{-4}$	99.98%	−0.0061	0	0

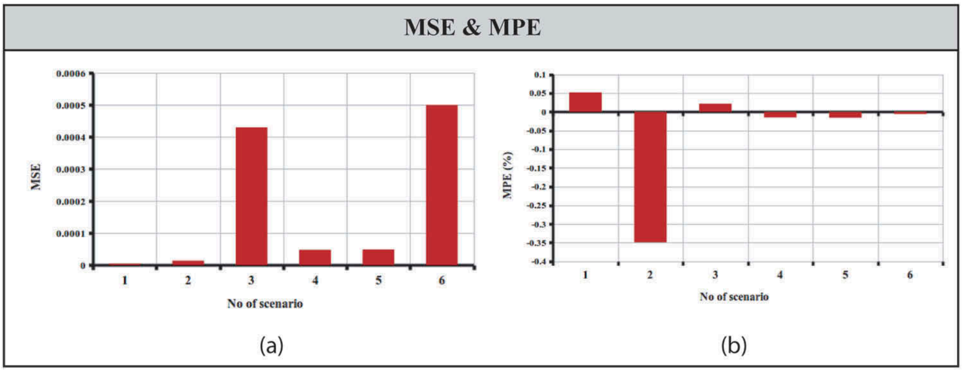


Figure 5. Results concerning: (a) MSE and (b) MPE for the best ANN models for Mz torque according to each scenario.

As for the GA-ANN models for Fz, the best results for each scenario are presented in Table 7 and depicted also in Figure 6(a,b); moreover, the optimum values for the additional input parameters are presented in Table 8. It can be observed that the use of the 3^{rd} scenario is deteriorating the MSE and R values, but the use of complex objective function with increasing weight for MPE provides significantly better results. For this case, the network obtained with the 5^{th} scenario is chosen as the best, as it has one of the lowest MSE, high R value, and no predicted values with error over 5% in comparison to the second best network for the 6^{th} scenario.

As for the additional design variables, training ratio varies between its whole range (0.6–0.75), as well as test ratio, whereas in most cases the best training algorithm is Levenberg–Marquardt and only once is the Conjugate Gradient with Powell–Beale restarts. The MSE performance function is the best in most cases and early stopping technique with about 10–12 values is sufficient for most cases. When regularization is also chosen, the max_fail epochs number is smaller than in cases with no regularization. Finally, the training time for these networks varied from 0.011 to 0.156 seconds.

As for the models concerning Mz, the results for the six scenarios are presented in Tables 9 and 10 and in specific, the MSE and MPE values for each scenario are depicted in Figure 7(a,b), respectively. The results are again

Table 7. Results using GA-ANN approach for Fz model with additional design variables.

Scenario No.	Number of hidden layers	1^{st} layer neurons	2^{nd} layer neurons	MSE_{tot}	R_{tot}	MPE (%)	N_5	N_{10}
1	1	5	-	0.00013	99.87%	−0.11042	6	0
2	1	11	-	$7.2*10^{-5}$	99.97%	0.10055	0	0
3	1	7	-	0.00530	99.75%	−0.04141	0	0
4	1	5	-	0.00364	99.75%	0.01876	6	0
5	1	6	-	0.00056	99.88%	−0.00169	0	0
6	1	3	-	0.00090	99.71%	−0.00346	3	0

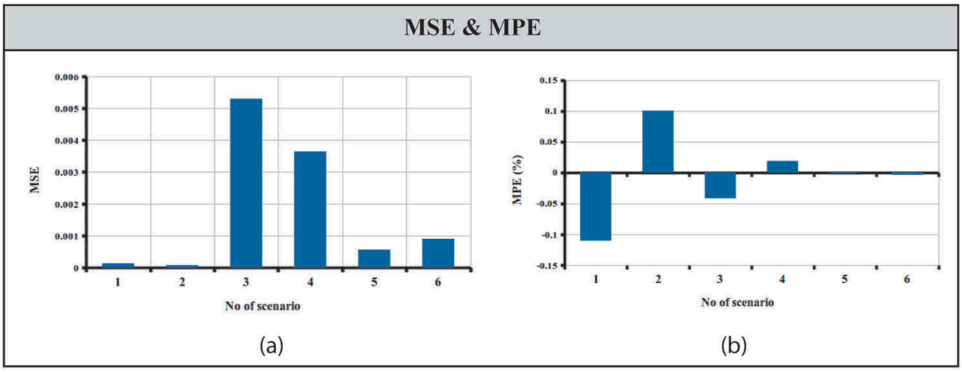


Figure 6. Results concerning: (a) MSE and (b) MPE for the best ANN models for Thrust Force according to each scenario.

Table 8. Values of additional design variables for each scenario.

Scenario	Tr. ratio	Test ratio	Training algorithm	Performance function	Max_fail	Regularization parameter	Gen_type
1	0.6012	0.142	CGPB	MSE	11	-	0
2	0.7387	0.1124	LM	MSE	14	-	0
3	0.6898	0.1317	LM	SSE	8	0.5901	1
4	0.6669	0.1481	LM	MSE	10	-	0
5	0.7437	0.1232	LM	SSE	12	0.4120	1
6	0.7486	0.1126	LM	MSE	15	-	0

overall better with the use of higher weighting coefficient for MPE, as for that case, the best MPE, low MSE, and high R total values are obtained. As with the previous cases, networks with a single hidden layer and 4–8 neurons have the best performance.

For the Mz models, training ratio value varied between 0.67 and 0.74, test ratio between 0.10 and 0.13, the best training algorithm was Levenberg-Marquardt, the best performance function was MSE for many cases and, in half of the cases, both early stopping and regularization were chosen. The best network architecture is shown to be that of the 6th scenario. Finally, the training time varied between 0.047 and 0.249 seconds.

Table 9. Results using GA-ANN approach for Mz model with additional design variables.

Scenario No.	Number of hidden layers	1^{st} layer neurons	2^{nd} layer neurons	MSE_{tot}	R_{tot}	MPE (%)	N_5	N_{10}
1	1	8	-	$9.4*10^{-6}$	99.99%	−0.18889	0	0
2	1	8	-	$4.4*10^{-5}$	99.99%	0.00298	0	0
3	1	4	-	0.0002	99.98%	−0.07751	1	0
4	1	5	-	$5.3*10^{-5}$	99.99%	−0.00920	0	0
5	1	5	-	0.0006	99.96%	0.00130	3	0
6	1	6	-	0.00015	99.98%	0.00128	1	0

Table 10. Values of additional design variables for each scenario.

Scenario	Tr. ratio	Test ratio	Training algorithm	Performance function	Max_fail	Regularization parameter	Gen_type
1	0.7212	0.1116	LM	MSE	14	-	0
2	0.7064	0.1233	LM	MSE	11	-	0
3	0.7310	0.1269	LM	MSE	7	-	0
4	0.6735	0.1055	LM	MSE	11	0.4077	1
5	0.7437	0.1292	LM	SSE	15	0.323	1
6	0.6857	0.1306	LM	SSE	11	0.6909	1

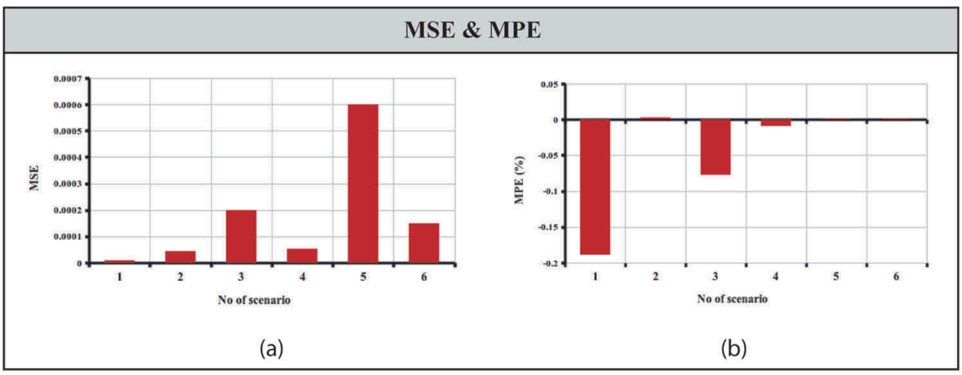


Figure 7. Results concerning: (a) MSE and (b) MPE for the best ANN models for Mz torque according to each scenario.

Comparison of Models

By observing the results obtained by GA-ANN models with 3 design variables and GA-ANN models with 10 design variables, several similarities and differences can be detected. At first, it becomes evident that, for cases with small to medium size datasets the theoretical rule of choosing a single hidden layer is proven correct, as the vast majority of resulting network architectures exhibited only a single layer. Moreover, the number of hidden neurons in most GA-ANN models was between 4 and 8. For all developed models R value was very high, over 99%, indicating the strong correlation of input and output data. MSE values varied in the range of 10^{-3} and 10^{-6} and MPE values between 10^{-1} to 10^{-3} (percent) in all cases. The number of predicted values with error 5%-10% and

Table 11. Comparison of best-performing networks for Fz GA-ANN models with 3 and 10 design variables (d.v.).

Scenario No.	Number of hidden layers	1^{st} layer neurons	2^{nd} layer neurons	MSE_{tot}	R_{tot}	MPE (%)	N_5	N_{10}
3 d.v.-6	1	5	-	0.00047	99.89%	−0.0036	0	0
10 d.v.-5	1	6	-	0.00056	99.88%	−0.00169	0	0

Table 12. Comparison of best-performing networks for Mz GA-ANN models with 3 and 10 design variables (d.v.).

Scenario No.	Number of hidden layers	1^{st} layer neurons	2^{nd} layer neurons	MSE_{tot}	R_{tot}	MPE (%)	N_5	N_{10}
3 d.v.-6	1	4	-	$5*10^{-4}$	99.98%	−0.0061	0	0
10 d.v.-6	1	6	-	$1.5*10^{-4}$	99.98%	0.00128	1	0

>10% was in all cases below 6, something that indicates that not only the average error was small but also the predicted values were not far from this average (towards unacceptable values, over 5 or 10%).

As for Fz models, the models with 3 design variables performed better in terms of MPE, than the models with 10 design variables for the first four objective functions, whereas the latter performed better in terms of the correlation coefficient. However, for the last two objective functions, their performance was more close and finally, the best network, as can be seen in Table 11, is the one obtained by the 5^{th} scenario of the GA-ANN models with 10 variables. This model, denoted as "10 d.v. −5" in Table 11, is slightly inferior in terms of MSE, but clearly superior in terms of MPE and so it is chosen as the best performing network.

As for Mz models, the models with 3 design variables exhibit generally higher MPE values, almost equal correlation coefficient values and slightly lower MSE values than the models with 10 design variables. As can be seen from Table 12, the best performing network is clearly the one with the 10 design variables, as it is superior in terms of MSE and MPE.

After the brief comparison of results between the two types of GA-ANN models is conducted, it is useful to compare these results in those of the simple ANN approach, presented in Section 3.1. By comparing the results from Tables 4, 11 and 12, it becomes obvious that the GA-ANN models with the definition of suitable objective functions can decrease the MPE very efficiently and also ensure that predicted values are kept below an acceptable limit of error, e.g. 5%. Moreover, in the case of GA-ANN models with 10 design variables, more important information is obtained, so that other parameters concerning the training procedure, e.g. training algorithm, performance function, etc. can also be properly selected.

Conclusions

During drilling process, cutting tools and cutting parameters are playing a crucial role for the manufacturing sustainability. The better use of cutting tools equals better product quality and longer tool life. In the present work, the aim was the generation of mathematical models for the prediction of the thrust force (Fz) and torque (Mz) related to the cutting tool diameters the feed rate and the cutting velocity during the drilling process. ANN approaches were adopted for the prediction of Fz and Mz during drilling of St60 specimen. More specifically, a comparison between ANN models with network architecture determined by a simple trial and error approach and ANN models with architecture determined by a GA-ANN approach is conducted. For the GA-ANN approach several different objective functions based on network performance and accuracy indicators are tested and several useful conclusions are drawn.

- ANN with architecture determined by simple trial and error approach can perform sufficiently, but lack in the minimization of MPE and restriction of predicted values error below acceptable limits.
- ANN developed using the GA-ANN approach can exhibit significantly lower MPE as well as no predicted values with error over 5%. The range of varied parameters for the GA-ANN approach and the number of candidate solutions checked can ensure that the developed ANN has architecture close to the globally optimum one. The overall best models had a network architecture of 3–6-1.
- Moreover, the results of GA-ANN approach are significantly useful, as they can verify some of the theoretical suggestions concerning ANN design. For example, it was shown that networks with a single hidden layer and 4–8 hidden neurons are sufficient for problems with small to medium datasets. Furthermore, it was verified that the Levenberg-Marquardt training algorithm is generally superior to the other for similar cases and that MSE is a reliable performance function for the MLP ANN.
- The conclusions of the present work can be employed for the creation of more advanced combined models of metaheuristics and ANN, with a view to develop ANN models suitable for larger-scale applications.

ORCID

Nikolaos Efkolidis http://orcid.org/0000-0001-9225-0311
Panagiotis Kyratsis http://orcid.org/0000-0001-6526-5622

References

Arifovic, J., and R. Gençay. 2001. Using genetic algorithms to select architecture of a feedforward artificial neural network. *Physica A: Statistical Mechanics and Its Applications* 289 (3–4):574–94. doi:10.1016/S0378-4371(00)00479-9.

Basheer, I. A., and M. Hajmeer. 2000. Artificial neural networks: Fundamentals, computing, design and application. *Journal of Microbiological Methods* 43:3–31. doi:10.1016/S0167-7012(00)00201-3.

Benardos, P. G., and G. C. Vosniakos. 2007. Optimizing feedforward artificial neural network architecture. *Engineering Applications of Artificial Intelligence* 20 (3):365–82. doi:10.1016/j.engappai.2006.06.005.

Boithias, F., M. El Mankibi, and P. Michel. 2012. Genetic algorithms based optimization of artificial neural network architecture for buildings' indoor discomfort and energy consumption prediction. *Building Simulation* 5 (2):95–106. doi:10.1007/s12273-012-0059-6.

Correa, A. B., and A. M. Gonzalez. 2011. Evolutionary algorithms for selecting the architecture of a MLP neural network: A credit scoring case. 11th IEEE international conference on Data Mining Workshops. doi: 10.1109/ICDMW.2011.80.

Curteanu, S., and H. Cartwright. 2011. Neural networks applied in chemistry. I. Determination of the optimal topology of multilayer perceptron neural networks. *Journal of Chemometrics* 25 (10):527–49. doi:10.1002/cem.1401.

Goldberg, D. 1989. *Genetic algorithms in search, optimization and machine learning. Reading.* MA: Addison-Wesley Professional.

Goyal, R., and A. K. Dubey. 2014. Quality improvement by parameter optimization in laser trepan drilling of superalloy sheet. *Materials and Manufacturing Processes* 29 (11–12):1410–16. doi:10.1080/10426914.2014.912313.

Holland, J. H. 1973. Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing* 2 (2):88–105. doi:10.1137/0202009.

Hynes, N. R. J., R. Kumar, and J. A. J. Sujana. 2017. Optimum bushing length in thermal drilling of galvanized steel using artificial neural network coupled with generic algorithm. *Matiriali in Tehnologije* 51 (5):813–22. doi:10.17222/mit.2016.290.

Idrissi, J., H. Ramchoun, Y. Ghanou, and M. Ettaouil. 2016. Genetic algorithm for neural network architecture optimization. 3rd international conference on Logistics Operations Management. doi:10.1109/GOL.2016.7731699.

Jain, A. K., J. Mao, and K. M. Mohiuddin. 1996. Artificial neural networks: A tutorial. *Computer* 29 (3):31–44. doi:10.1109/2.485891.

Jeong, C., J. H. Min, and M. S. Kim. 2012. A tuning method for the architecture of neural network models incorporateing GAM and GA as applied to bankruptcy prediction. *Expert Systems with Applications* 39 (3):3650–58. doi:10.1016/j.eswa.2011.09.056.

Kannan, T. D. B., Deepan Barathi, G. Rajesh Kannan, M. Umar, and S. Ashok Kumar. 2015. ANN approach for modelling parameters in drilling operation. *Indian Journal of Science and Technology* 8(22). doi:10.17485/ijst/2015/v8i22/79097.

Khandelwal, M., and D. J. Armaghani. 2016. Prediction of drillability of rocks with strength properties using a hybrid GA-ANN technique. *Geotechnical and Geological Engineering* 34 (2):605–20. doi:10.1007/s10706-015-9970-9.

Khorani, V., N. Forouzideh, and A. M. Nasrabadi. 2011. Artificial neural network weights optimization using ICA, GA, ICA-GA and R-ICA-GA: Comparing performances. IEEE workshop on hybrid intelligent models and applications (HIMA). doi:10.1109/HIMA.2011.5953956.

Kilickap, E., and M. Huseyinoglu. 2010. Selection of optimum drilling parameters on Burr height using response surface methodology and genetic algorithm in drilling of AISI 304 stainless steel. *Materials and Manufacturing Processes* 25 (10):1068–76. doi:10.1080/10426911003720854.

Murata, N., S. Yoshizawa, and S. Amari. 1994. Network information criterion-determining the number of hidden units for an artificial neural network model. *IEEE Trans Neural Networks* 5 (6):865–72. doi:10.1109/72.329683.

Nassef, A., A. Elkaseer, E. S. Abdelnasser, M. Negm, and J. A. Qudeiri. 2018. Abrasive jet drilling of glass sheets: Effect and optimisation of process parameters on kerf taper. *Advances in Mechanical Engineering* 10 (1):1–10. doi:10.1177/1687814017748435.

Ojha, V. K., A. Abraham, and V. Snášel. 2017. Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence* 60 (97–116). doi: 10.1016/j.engappai.2017.01.013.

Sujova, A., K. Marcinekova, and S. Hittmar. 2017. Sustainable optimization of manufacturing process effectiveness in furniture production. *Sustainability* 9 (6):923. doi:10.3390/su9060923.

Ul Islam, B., Z. Baharudin, M. Q. Raza, and P. Nallagownden. 2014. Optimization of neural network architecture using genetic algorithm for load forecasting. 5th international conference on Intelligent and Advances Systems (ICIAS). doi:10.1109/ICIAS.2014/6869528.

Vijayaraghavan, V., and S. Castagne. 2016. Sustainable manufacturing models for mass finishing process. *International Journal of Advanced Manufacturing Technology* 86 (1–4):49–57. doi:10.1007/s00170-015-8146-3.

Zhang, Q., and C. Wang. 2008. Using genetic algorithm to optimize artificial neural network: A case study on earthquake prediction. Second international conference on Genetic and Evolutionary Computing. doi.10.1109/WGEC.2008.96

Zhang, Z. H., D. N. Yan, J. T. Ju, and Y. Han. 2012. Prediction of the flow stress of a high alloyed austenitic stainless steel using artificial neural network. *Materials Science Forum* 724:351–54. doi:10.4028/www.scientific.net/MSF.724.351.