

Article

A Proposal of an Adaptive Neuro-Fuzzy Inference System for Modeling Experimental Data in Manufacturing Engineering

C. J. Luis Pérez

Materials and Manufacturing Engineering Research Group, Engineering Department, Public University of Navarre, Campus de Arrosadía s/n, 31006 Pamplona, Navarra, Spain; cluis.perez@unavarra.es

Received: 31 July 2020; Accepted: 17 August 2020; Published: 19 August 2020



Abstract: In Manufacturing Engineering there is a need to be able to model the behavior of technological variables versus input parameters in order to predict their behavior in advance, so that it is possible to determine the levels of variation that lead to optimal values of the response variables to be obtained. In recent years, it has been a common practice to rely on regression techniques to carry out the above-mentioned task. However, such models are sometimes not accurate enough to predict the behavior of these response variables, especially when they have significant non-linearities. In this present study a comparative analysis between the precision of different techniques based on conventional regression and soft computing is initially carried out. Specifically, regression techniques, based on the response surface model, as well as the use of artificial neural networks and fuzzy inference systems along with adaptive neuro-fuzzy inference systems will be employed to predict the behavior of the aforementioned technological variables. It will be shown that when there are difficulties in predicting the response parameters by using regression models, soft computing models are highly effective, being much more efficient than conventional regression models. In addition, a new method is proposed in this study that consists of using an iterative process to obtain a fuzzy inference system from a design of experiments and then using an adaptive neuro-fuzzy inference system for tuning the constants of the membership functions. As will be shown, with this method it is possible to obtain improved results in the validation metrics. The means of selecting the membership functions to develop this model from the design of experiments is discussed in this present study in order to obtain an initial solution, which will be then tuned by using an adaptive neuro-fuzzy inference system, to predict the behavior of the response variables. Moreover, the obtained results will also be compared.

Keywords: FIS; ANFIS; ANN; regression; modeling

1. Introduction

Over the last few years, an increasing number of research studies dealing with modeling of technological response variables as a function of process parameters in order to determine the most appropriate operating conditions has been observed in Manufacturing Engineering. Given the importance that these manufacturing processes have for industry, a large number of models have been developed in recent years, based on regression techniques, fuzzy inference systems, and artificial neural networks, generally through the application of supervised learning and feed forward networks. In this present research study, different techniques, based both on response surface methodology (RSM) and on soft computing, will be analyzed to determine their capacity to accurately predict the response variables. It should be noted that conventional regression techniques are effective when the models provide high values of the coefficient of determination. However, when the response

variables have high curvature or are irregular, conventional models are not effective in modeling the response variables. Therefore, when regression models are not adequate to predict the behavior of the response variables, it is necessary to use other alternative methodologies and, as shown further on, soft computing has significant advantages over conventional regression methods.

In this present study, a comparative study is first carried out between the accuracy in regression models, based on RSM, versus that of soft computing techniques such as Artificial Neural Networks (ANN) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS). First, from a Design Of Experiments (DOE), two sets of values generated from a function that can be made to exhibit both low curvature and high curvature within the range of variation of the DOE will be used, which is usually observed when modeling variables obtained in manufacturing processes where some part of them are well fitted by models obtained from regression, with high coefficients of determination, and some others, on the contrary, exhibit a behavior which is not possible to adequately model by using conventional regression techniques. Finally, a comparative analysis will be carried out between the results obtained with those that would be obtained in a real manufacturing study case and a comparison between those obtained will be carried out.

Furthermore, this comparison made between the regression and soft computing models does not use all the data from the DOE to fit the models, as is common practice. Instead, a part of the design is used to fit the model and another part for validation and testing, since the fact of using the entire DOE to fit the model does not allow its validation with independent data, and it may lead to results that are not completely reliable, as shown in this present study. Although it is true that there are different research studies that carry out comparative studies, the above-mentioned is not usually taken into consideration. In addition, this present study proposes a new methodology to obtain an ANFIS which is capable of efficiently modeling the response variables obtained from a DOE. An iteration with the membership functions is carried out to develop a Fuzzy Inference Systems (FIS) that will be tuned later by using an ANFIS. In this study, it will be shown that by using this method it is possible to efficiently model both response variables with low curvature and high curvature. The proposed method, which is shown in Section 4.4, is based on finding the shape of the initial membership functions in order to develop a FIS which will be then tuned with an ANFIS and which will later lead to a better fit in the validation data as well as in the train data.

The results obtained by the different techniques are also compared. Before approaching the present study, several research works that were found in the bibliography will be analyzed in the “State of the Art” section.

2. State of the Art

Over the last few years, regression techniques as well as ANN, FIS, and ANFIS have been widely employed for modeling several manufacturing processes. In regard to FIS, these have been widely used in distinct scientific fields, among others, dealing with control, pattern recognition, and modeling [1], where Takagi–Sugeno [2,3] and Mamdani [4,5] are the most commonly employed. Several studies can be found in the literature dealing with FIS, such as that of Muralova et al. [6] who proposed a Mamdani FIS for modeling the cutting speed in wire electrical discharge machining (WEDM). These authors employed a maximum of results for aggregation and the centroid to de-fuzzify the aggregated output. In another study, Aamir et al. [7] used a Mamdani FIS to predict the surface roughness and hole size as a function of feed rate and cutting speed in multi-hole drilling. These authors calculated the outputs based on the centroid method. Some other studies are those of Cuka and Kim [8] that developed a fuzzy inference system for tool condition monitoring in end-milling operations and Joshi et al. [9] who analyzed surface roughness and material removal rate (MRR) of Inconel 800HT, when machined with copper electrode on electrical discharge machining (EDM). On the other hand, Wang et al. [10] employed a fuzzy multicriteria decision-making model (MCDM) for raw material supplier selection in the plastic industry. Likewise, Lin et al. [11] applied fuzzy collaborative intelligence approach for fall detection in four existing smart technology applications and a methodology for obtaining technological

tables based on using a zero-order Sugeno FIS is shown in Ref. [12]. Some other studies, worth mentioning, are those of Cavallaro [13] that employed a Takagi–Sugeno FIS to assess the sustainability of biomass of production and that of Shabgard et al. [14] who employed a Mamdani inference system to predict MRR, electrode wear and surface roughness in the EDM and ultrasonic-assisted EDM (US/EDM) processes of tungsten carbide.

Comparing the performance of different models for the prediction of response variables has been examined by different authors and for different variables and technological processes. For example, it is worth mentioning the research studies of Bernardos et al. [15] which analyzed the arithmetic average roughness (Ra) in face milling using a feedforward ANN and a Levenberg–Marquardt algorithm. Furthermore, in the study of Devarasiddappa et al. [16] these authors employed an ANN to predict surface roughness in WEDM of Inconel 825, and Twardowski et al. [17] employed multilayer perceptron (MLP) for modeling tool wear during turning of hardened steel, among others. Another interesting technique for modeling technological variables in manufacturing processes are hybrid learning procedures that combine ANNs and FIS, which are known as ANFIS [18]. ANN and Fuzzy systems are commonly combined in soft computing to solve real-world problems. From the research study of Jang [18], several studies have been developed, as can be observed in the review of the state of the art of Shihabudheen and Pillai [19]. For example, in Maher et al. [20] an ANFIS was employed to predict cutting speed, surface roughness, and heat-affected zone in WEDM and in Çaydaş et al. [21] an ANFIS was used for modeling surface roughness and white layer thickness in WEDM of an AISI D5 tool steel. Likewise, Kang et al. [22] proposed a heating temperature estimation method for diagnosis and assessment of fire-damaged concrete structures by using an ANFIS. Al-Ghamdi [23] employed an ANFIS and polynomial modeling approaches to model the MRR in EDM of a Ti-6Al-4V alloy. A first order Sugeno along with a back-propagation neural network training algorithm was used by these authors [23]. Both ANNs and ANFIS have been widely used in Engineering and Technology fields, among many others. Comparison of ANFIS and MLP ANNs can be found in the bibliography dealing with different subjects. For example, in the study of Suparta et al. [24] the authors employed both ANFIS and MLP for prediction of precipitable water vapor. These authors found that the ANFIS provided a better performance than that obtained by using MLP. On the other hand, in the research study of Yilmaz and Kaynar [25] the authors employed multiple linear regression, ANN and ANFIS for prediction of swell percent of soil. These authors found that the ANN models had a better performance than that obtained with ANFIS and multiple regression.

Fuzzy modeling has been widely applied in several scientific and technological fields. There exists a large amount of industrial applications which are based on these techniques. Among the research studies found in the literature it is worth mentioning the application of soft computing techniques for airport classification, as shown in the research study of Postorino et al. [26], control of piezoelectric actuators, as shown in Li et al. [27] and in Cheng et al. [28], control of Brushless DC motors [29] and monitoring of fuel system of an industrial gas turbine, as shown in Bagua et al. [30]. Other applications deal with both detection and classification of defects, as shown in Versaci [31] and in Versaci et al. [32], fault diagnosis of rolling bearing in industrial robots [33], and fault detection in wind turbines [34], among many others.

In addition, fuzzy systems are able to handle uncertainties in an efficient way, as shown in Lin et al. [35], where a Takagi–Sugeno–Kang (TSK) type-2 fuzzy neural network was proposed for system modeling and noise cancellation, or in Biglarbegian et al. [36], where a design methodology based on interval type-2 TSK fuzzy logic controllers for modular and reconfigurable robots manipulators with uncertain dynamic parameters was shown, among many others [37,38]. Furthermore, Chen et al. [39] employed a type-2 fuzzy neural network to predict bearing health conditions and Tayyab et al. [40] applied fuzzy theory to consider uncertainty in demand information in a multi-stage lean manufacturing system. These authors employed the centroid to de-fuzzify the objective function. Other studies such as that of Faisal et al. [41] used particle swarm optimization (PSO) and biogeography-based optimization (BBO) algorithms for a multiple-objective optimization of the MRR and the arithmetical

mean roughness (Ra) for the EDM process, Alarifi et al. [42] employed genetic algorithms and particle swarm optimization to determine the parameters of an ANFIS model to predict the thermo-physical properties of Al₂O₃-MWCNT/thermal oil hybrid nanofluid and an analysis of the PSO implementation in designing parameters of manufacturing processes as well as a benchmark with other optimization techniques can be found in the review study of Sibalija [43]. On the other hand, Alajmi et al. [44] used an ANFIS-QPSO to predict the surface roughness of the dry and cryogenic turning process of AISI 304 stainless steel. Moreover, a comparison of prediction accuracy between ANFIS, ANFIS-GA, ANFIS-PSO, and ANFIS-QPSO was carried in this study in terms of the MAPE, root mean squared error (RMSE) and R² values for both dry and cryogenic turning processes [44].

Furthermore, neural networks have also been widely employed in several scientific and technological fields. Among the research studies found in the literature it is worth mentioning the application of neural networks for prediction of biogases concentration using spiking neural networks [45], feature recognition and process planning of casting dies [46], quality control in manufacturing processes [47], prediction of springback in sheet metal forming [48], aerodynamic data modeling [49], detection of skin diseases [50], automatic control of house elements [51,52] and energy forecasting in the manufacturing sector [53], among many other applications.

Finally, it is worth mentioning the employment of techniques based on the DOE and regression techniques that have also been widely used for the study of technological variables found in manufacturing processes as shown in research studies such as that of Airao et al. [54] which analyzed the effect of cutting speed, feed rate, and axial depth of cut on surface roughness obtained in end-milling of a stainless steel, that of Kasdekara et al. [55], which employed a 2⁴ full factorial (DOE) for determining the most important factors which influence MRR in Electro-chemical machining of AA6061 by using MLP and regression, and that of Ahmed et al. [56] in which the MRR, in laser milling of three alloys (Ti6Al4V, Inconel 718 and AA 2024), was evaluated using the response surface method and DOE. On the other hand, Aslantas et al. [57] obtained empirical relations between cutting speed, feed rate, depth of cut and surface roughness parameters using the RSM for the micro-turning process in a Ti6Al4V alloy and Su et al. [58] employed a multi-objective optimization method based on grey relational analysis and RSM along with Taguchi method for analyzing surface roughness and MRR in turning of an AISI 304 austenitic stainless steel. Regression analysis is also used by Zajac et al. [59] to make predictions of cutting tool durability in turning processes. Furthermore, in the research study from Torres et al. [60] the manufacturing of TiB₂ by using EDM was analyzed by using RSM. On the other hand, Torres et al. [61] employed a 4³ factorial DOE for modeling the behavior of the arithmetical mean roughness (Ra), the electrode wear (EW) and the MRR in the EDM machining of an Inconel®600 alloy using Cu-C electrodes (Inconel®600 alloy is a registered trademark of the Special Metals Corporation group of Companies). In Lin et al. [62] surface roughness in the end-milling process is analyzed as a function of spindle speed, cutting depth, and feed rate and machining vibration by using multiple regression and ANNs and in Karloopia et al. [63] cutting speed, depth of cut, and feed are used as independent variables to obtain cutting force and tool wear in Al–Si–TiB₂ composites by using linear regression.

As was observed in the “State of the Art” section, there are different methodologies that have been used for modeling technological variables in manufacturing processes, some of which will be analyzed in order to determine their performance in modeling response variables. The following section describes the methodology to be followed in this present study.

3. Methodology

As shown in the review of the state of the art, it is observed that regression models based on the use of experimental design have been widely employed for modeling the behavior of different technological variables, which in the manufacturing field, are usually related to surface finish, MRR, or tool wear, among many others. If there is enough data available, it would be considered interesting to validate the models with other independent experimental data to determine the validity of the fits. For this reason, the present study compares the results obtained when only one set of the DOE is

used to train the model and the rest of the DOE is used to validate the obtained models, following the methodology that is employed in the Deep Learning Toolbox™ of Matlab™2019b [64] to train, test, and validate ANNs and in order to compare the obtained results with both regression and soft computing tools (ANN and ANFIS).

Therefore, First, the data is randomly divided using the Matlab™2019b functions into three sets comprising 70%, 15%, and 15% of data (train, test, and validate, respectively). With the train data, a model will be fitted first, and then the thus obtained models will be validated using the validation data and later on they will be tested with independent test data [64]. RSM as well as ANN, FIS, and ANFIS will be employed to analyze the advantages and disadvantages of each. Moreover, a method for obtaining and ANFIS from the DOE is proposed.

It should be mentioned that if the RSM model can provide a highly adjusted R-squared value, it could be considered in advance that it is not necessary to use soft computing techniques since they require higher computational time than the previous ones and, in general, the models thus obtained are more complex. However, if more precision is required or the response surface of a certain variable has a high curvature in the design range or if it is very irregular, this will make conventional regression models ineffective and lead to low values of the coefficient of determination to be obtained. Therefore, the use of some other types of techniques should be considered, such as those based on soft computing.

In manufacturing processes, is it quite common for certain technological variables to have a more regular response behavior and, therefore, they can be modeled by methodologies such as RSM, and, on the other hand, there also exist some other technological variables that cannot be adequately modeled using conventional regression techniques. For example, in Reference [61] it is shown that the arithmetic average roughness (Ra) can be approximated by a polynomial regression using RSM, in an EDM process. However, due to the variability of the data, the wear that the electrode (EW) undergoes in this EDM process cannot be adequately modeled by using RSM, and for this reason, in order to adequately model this technological variable, it would be necessary to employ some other technique, for example based on using an ANFIS or an ANN. In this way, and using an ANFIS, Figure 1a,b have been generated in order to show the surface responses that would be obtained when modeling Ra (μm) and EW (%), in the case of positive polarity of the electrode, using data taken from Ref. [61]. As can be seen from the results shown in Ref. [61], in which only RSM-based polynomial regression is used, the R-squared value is greater than 90% in the case of average roughness (Ra), but this coefficient is much lower in the case of EW and, hence, although Ra could be modeled by using RSM, this is not the case for the EW.

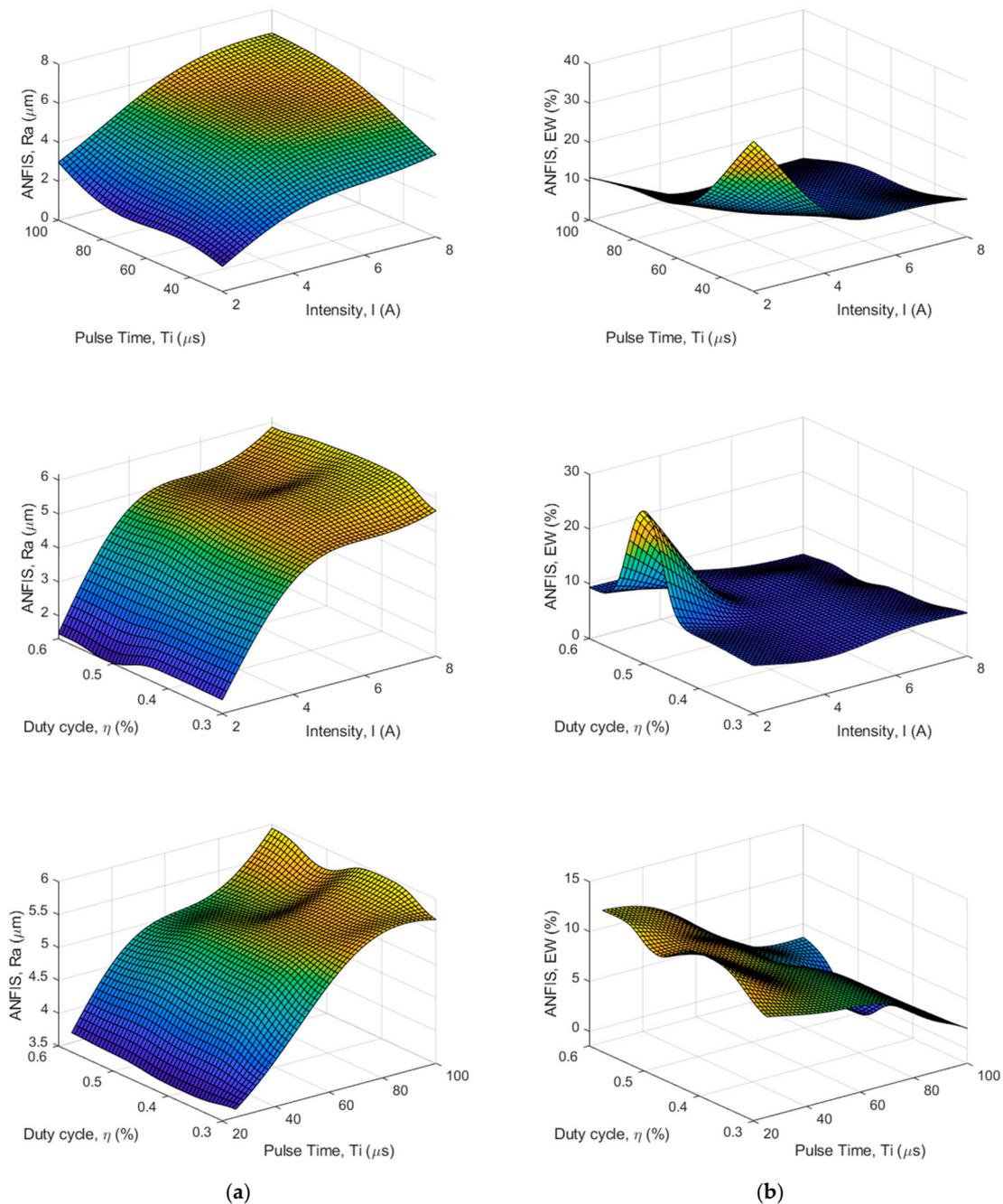


Figure 1. (a) Surface roughness average, Ra (μm), as a function of (Intensity, Pulse Time and duty cycle) (type 1 function) and (b) Electrode Wear, EW (%), as a function of (Intensity, Pulse Time and duty cycle) (type 2 function). Surface responses have been generated using an ANFIS from experimental data taken of Ref. [61].

In this present study the function defined by Equation (1) will be used where their constants (a_j) will be varied to have two types of response variables. This function has been chosen since it is possible to vary the values of the constant (a_j) to have a function that shows a behavior that could be modeled properly by conventional regression techniques and can also be made to present a behavior with greater curvature, which is difficult to model by using the RSM methodology. The first type of response variable would represent the behavior of a response variable that has low curvature within the study range, which is to be classified as type 1 function (f_1), i.e., functions in which if the RSM is used in a set of data obtained from these functions, the adjusted coefficient of determination of the

RSM regression models would provide values greater than 90%. Likewise, the second type of response variable would represent a behavior that either has a high curvature or where the experimental data are irregular in the design range considered, which may be classified as type 2 function (f_2).

As an example, from the determination coefficient of the regression shown in Ref. [61], Ra (μm) surface response could be classified, from the previously mentioned, as type 1 function and EW (%) surface response could be classified as type 2 function. A DOE with three factors and four levels for each factor (3^4) will be used. This arrangement of the design is similar to that used in Ref. [61]. However, any other type of DOE could be used. From this, the levels of the variables shown in Table 1 are considered, where the levels of variation of the variables have been normalized to fall into the range $[-1, 1]$ and the value obtained in the response variable will be left without normalizing it in order to directly obtain its value, without the need for a subsequent transformation. If necessary, a transformation of the dependent variable could also be performed so that it falls within a certain interval. Therefore, the -1 value corresponds to the minimum level of an input variable and the $+1$ value corresponds to the maximum value of this variable. It would be feasible to have any other range and to make a transformation to bring it to this range or to analyze it directly in the starting range.

$$f = a_0 + \frac{\sin(a_1x_1) - \sin(a_2x_2) + \sin(a_3x_3)}{1 + \exp((a_1x_1)^2 - (a_2x_2)^2 + (a_3x_3)^2)} \tag{1}$$

Table 1. Design factors and levels.

Design Factors		Levels and Values		
x_1	-1	$-1/3$	$1/3$	1
x_2	-1	$-1/3$	$1/3$	1
x_3	-1	$-1/3$	$1/3$	1

The responses surfaces of the function considered in this present study are shown in Figure 2. As can be observed, Figure 2a presents a behavior that, in advance, could be considered to be adequately modeled by the RSM since it does not exhibit a great degree of curvature within the considered interval. On the contrary, Figure 2b shows another function with a response surface in the considered interval which exhibits much greater curvature than the former.

It may be underlined that either an actual study case or any other function could have been used to carry out the first part of this study. As can be observed in Figures 1 and 2, the response variables have similar shapes, for both study cases, i.e., low curvature in one response variable and high curvature in the other. Since a DOE has been used in accordance with that shown in Table 1, therefore 4^3 experiments will be available. The values shown in Tables 2 and 3 that have been generated from the previously mentioned functions, which are shown in Figure 2, will be used in this study. To be consistent with the uncertainty of the measurements that the response variables may have, the values of the functions are considered to have been rounded to two decimal places.

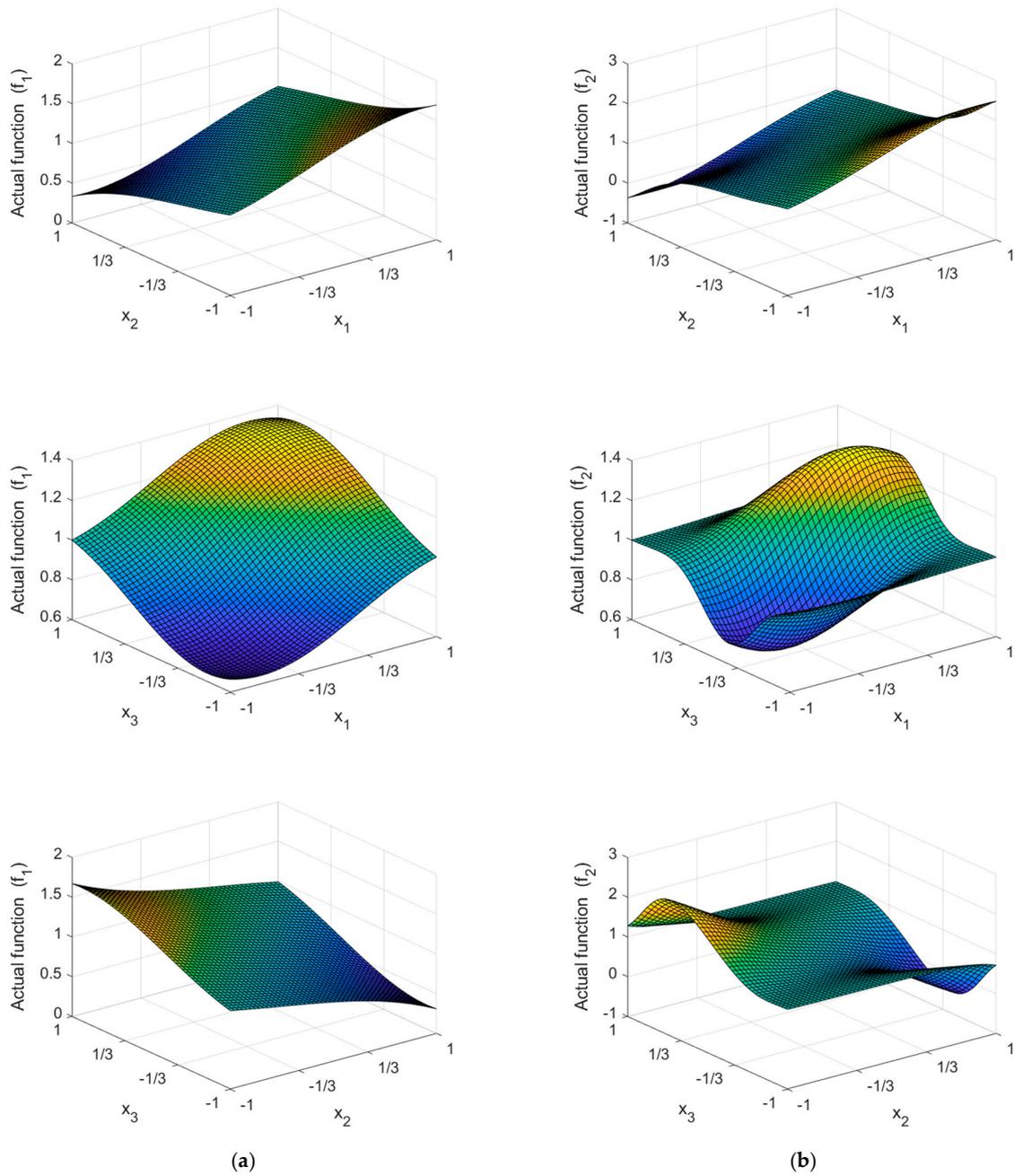


Figure 2. (a) Actual values of f_1 function (type 1), where $a_0 = 1, a_1 = a_2 = a_3 = 0.75$ and (b) Actual values of f_2 function (type 2), where $a_0 = 1, a_1 = 1, a_2 = 1.5, a_3 = 2$.

Table 2. Values obtained from type 1 function (f_1), where $a_0 = 1, a_1 = a_2 = a_3 = 0.75$.

(Tr = Train, Ts = Test, Vd = Validate)							
Exp.	f_1	Exp.	f_1	Exp.	f_1	Exp.	f_1
1, Tr	0.75	17, Ts	0.88	33, Tr	1.12	49, Vd	1.25
2, Vd	0.88	18, Tr	1.11	34, Tr	1.41	50, Vd	1.54
3, Tr	1.12	19, Tr	1.41	35, Vd	1.71	51, Ts	1.78
4, Tr	1.25	20, Vd	1.54	36, Tr	1.78	52, Vd	1.74
5, Tr	0.71	21, Ts	0.75	37, Tr	0.93	53, Tr	1.06
6, Vd	0.75	22, Ts	0.88	38, Tr	1.12	54, Ts	1.25
7, Tr	0.93	23, Tr	1.12	39, Tr	1.36	55, Tr	1.43
8, Tr	1.06	24, Tr	1.25	40, Tr	1.43	56, Vd	1.41
9, Tr	0.59	25, Vd	0.57	41, Tr	0.75	57, Tr	0.94
10, Tr	0.57	26, Tr	0.64	42, Tr	0.88	58, Tr	1.07
11, Tr	0.75	27, Tr	0.88	43, Tr	1.12	59, Tr	1.25
12, Tr	0.94	28, Tr	1.07	44, Tr	1.25	60, Tr	1.29
13, Tr	0.26	29, Ts	0.22	45, Tr	0.46	61, Tr	0.75
14, Tr	0.22	30, Tr	0.29	46, Tr	0.59	62, Ts	0.88
15, Tr	0.46	31, Tr	0.59	47, Ts	0.89	63, Tr	1.12
16, Ts	0.75	32, Ts	0.88	48, Vd	1.12	64, Tr	1.25

Table 3. Values obtained from type 2 function (f_2), where $a_0 = 1, a_1 = 1, a_2 = 1.5, a_3 = 2$.

(Tr = Train, Ts = Test, Vd = Validate)							
Exp.	f_2	Exp.	f_2	Exp.	f_2	Exp.	f_2
1, Tr	0.95	17, Ts	0.97	33, Tr	1.06	49, Vd	1.06
2, Vd	0.68	18, Tr	1.04	34, Tr	1.60	50, Vd	1.84
3, Tr	1.54	19, Tr	2.09	35, Vd	2.64	51, Ts	2.70
4, Tr	1.06	20, Vd	1.21	36, Tr	1.30	52, Vd	1.17
5, Tr	0.99	21, Ts	0.98	37, Tr	1.00	53, Tr	1.00
6, Vd	0.77	22, Ts	0.80	38, Tr	1.08	54, Ts	1.16
7, Tr	1.06	23, Tr	1.33	39, Tr	1.60	55, Tr	1.45
8, Tr	1.00	24, Tr	1.02	40, Tr	1.04	56, Vd	1.02
9, Tr	0.98	25, Vd	0.96	41, Tr	0.98	57, Tr	1.00
10, Tr	0.55	26, Tr	0.40	42, Tr	0.67	58, Tr	0.94
11, Tr	0.84	27, Tr	0.92	43, Tr	1.20	59, Tr	1.23
12, Tr	1.00	28, Tr	1.00	44, Tr	1.02	60, Tr	1.01
13, Tr	0.83	29, Ts	0.70	45, Tr	0.79	61, Tr	0.94
14, Tr	-0.70	30, Tr	-0.64	46, Tr	-0.09	62, Ts	0.46
15, Tr	0.16	31, Tr	0.40	47, Ts	0.96	63, Tr	1.32
16, Ts	0.94	32, Ts	0.94	48, Vd	1.03	64, Tr	1.05

As previously mentioned, in order to carry out the comparative study between the different models, data shown in Tables 2 and 3 are randomly divided into three sets (train, test, validate) according to (70%, 15% and 15%). Thus, the experiments used to test and to validate the models are those corresponding to the positions {16, 17, 21, 22, 29, 32, 47, 51, 54, 62} and {2, 6, 20, 25, 35, 48, 49, 50, 52, 56}, respectively, the rest of the remaining experiments will be used to train the models. It should be noted that it would be possible to use all the points to train the models. However, it is considered that either some additional experiments or some points of the DOE should be used to validate the models, and hence this procedure has been followed in this study.

Once the methodology to be followed in this study as well as the classification proposed for the response variables (Types 1 and 2) has been shown, the results obtained by using the different methodologies for data modeling will be analyzed. Next, the validity of the results obtained with the analytical function will be compared with those obtained when a real case is studied. In addition, a new method for obtaining an ANFIS based on the train data and the validation data of the DOE will be shown in Section 4.4 and the obtained results will be compared. As previously mentioned, this proposed method is based on finding the shape of the initial membership functions in order to develop a FIS which will be then tuned with an ANFIS and in this way a better fit in the validation data as well as in the train data will be obtained. Figure 3 shows a flowchart using a Nassi–Shneiderman diagram [65] in order to introduce the algorithm of the proposed method, which will be analyzed in Section 4.4 of this present study.

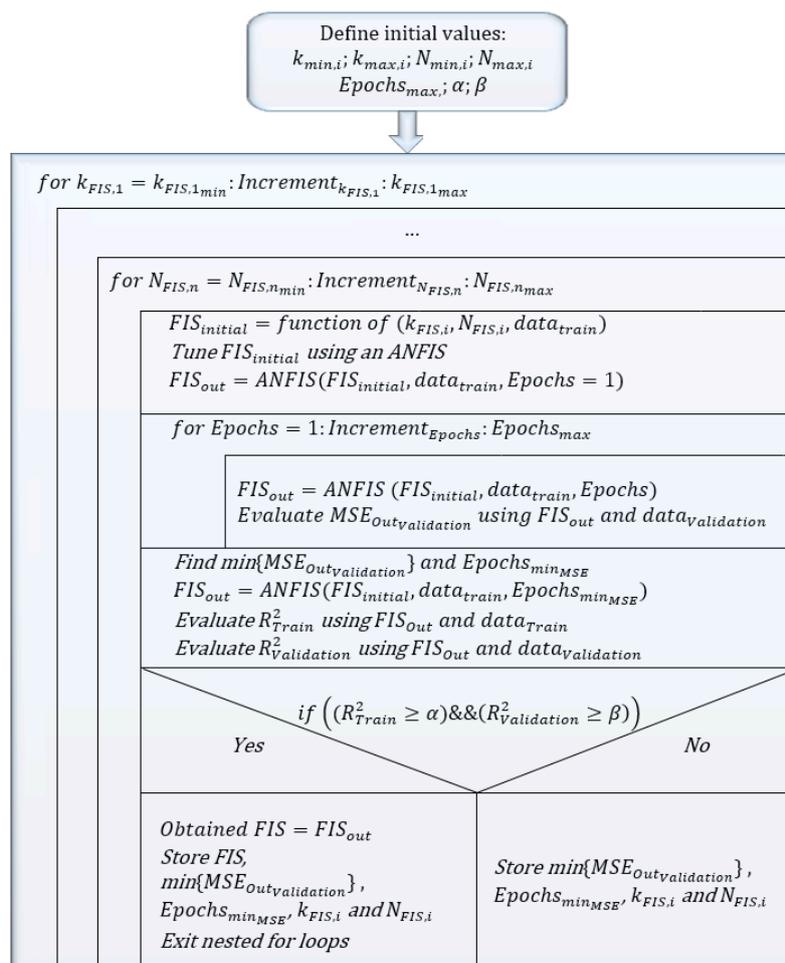


Figure 3. Flowchart of the proposed method.

4. Results and Discussion

In this section, the results obtained when modeling the function shown in Equation (1) are shown which, as indicated, can present a behavior as shown in Figure 2a that, in advance, could be expected that it will be adequately modeled by the RSM since it does not exhibit a great degree of curvature within the considered interval (type 1 function) and, on the contrary, it can be made to exhibit much greater curvature in the considered interval, as shown in Figure 2b, and then it is likely that conventional regression techniques would not be adequate (type 2 function). Hence, some other types of technologies for data analysis should be used.

4.1. Response Surface Model (RSM)

Equation (2) shows the RSM to be used to predict the behavior of type 1 (f_1) and type 2 (f_2) functions, which are shown in Figure 2a,b, by using data shown in Tables 2 and 3, respectively.

Likewise, as previously mentioned, Table 1 shows the levels of the independent variables analyzed in this study, where they are varied following a full factorial DOE.

RSM model :

$$y \sim (b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_1x_2 + b_5x_1x_3 + b_6x_2x_3 + b_7x_1^2 + b_8x_2^2 + b_9x_3^2) \tag{2}$$

Tables 4 and 5 show the results obtained by performing a regression fit using the train data included in Tables 2 and 3 and by using the model shown in Equation (2). As can be observed, the regression coefficient of determination has a value of 0.961 and an adjusted R-squared value of 0.951, for the case of f_1 function, which are high enough to be able to draw some conclusions from the regression models. However, in the case of f_2 function, these values are much lower (0.577 and 0.465), respectively. For this reason, the model shown in Equation (2) is not as good as the first one for making predictions about the response variable. Therefore, it is necessary to use some other types of models that allow a better approximation to the actual behavior of the said variable to be obtained.

Table 4. Estimated coefficients for f_1 using the train data and the RSM model.

b_0	b_1	b_2	b_3	b_4
0.9949	0.2684	−0.3436	0.2469	0.0002
b_5	b_6	b_7	b_8	b_9
0.0355	−0.0271	0.0245	−0.0316	0.0199

Number of observations: 44, Error degrees of freedom: 34. Root Mean Squared Error: 0.078. R-squared: 0.961, Adjusted R-squared: 0.951.

Table 5. Estimated coefficients for f_2 using the train data and the RSM model.

b_0	b_1	b_2	b_3	b_4
0.9831	0.2065	−0.4628	0.0979	0.1816
b_5	b_6	b_7	b_8	b_9
0.0723	−0.0753	0.1000	−0.1776	0.0226

Number of observations: 44, Error degrees of freedom: 34. Root Mean Squared Error: 0.374. R-squared: 0.577, Adjusted R-squared: 0.465.

In this present study, instead of using the regression model obtained from the adjusted R^2 , the complete regression model will be used, which both considers all the effects of the variables in the models and has a higher R^2 value, since the fact of using the model adjusted to the degrees of freedom could eliminate some of the independent variables.

Figure 4 shows the results obtained from the model fitted with the train data. In addition, the results of the validation metrics are shown in Figure 4. From these results, it is possible to affirm that this model could be applied to analyze the behavior of the response surface shown in Figure 2a.

However, although this regression model could be used to draw some conclusions about the response variable, it presents some limitations because the coefficient of determination when using test data is slightly lower than that obtained with the train data. This can also be observed if the main effects plots obtained with the fitted model and those obtained with the actual function are compared, which are shown in Figures 4 and 5, respectively. Figures 4 and 6 show the response surfaces obtained with the RSM model when fitting the data from Tables 2 and 3, for types 1 and 2 functions, respectively. The validation metrics employed for measuring the goodness of the fit are mean squared error (MSE), RMSE and mean absolute error (MAE). These validation metrics are defined by Equation (3), where y_j are the actual values and \hat{y}_j the estimated values.

$$MSE = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2, RMSE = \sqrt{MSE} \text{ and } MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \tag{3}$$

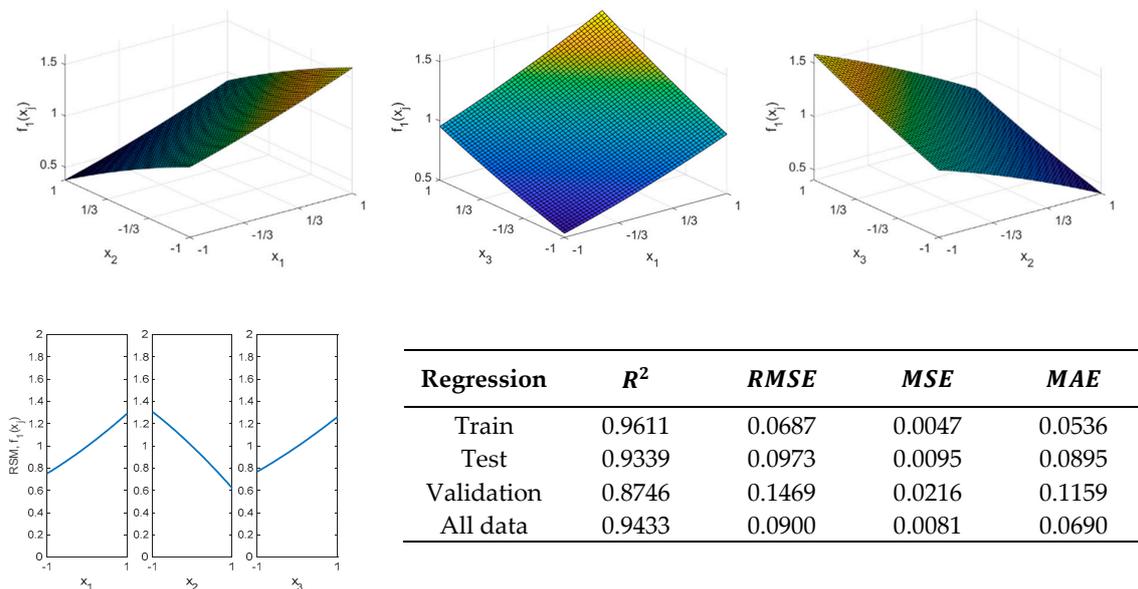


Figure 4. Response surfaces obtained, main effects plot and results of the validation metrics obtained with the RSM model for the case of type 1 function (f_1).

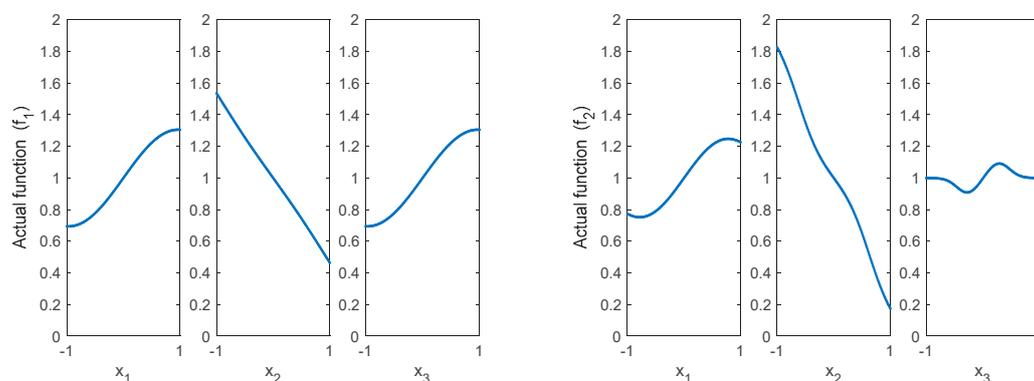


Figure 5. Actual main effects plot for f_1 and f_2 .

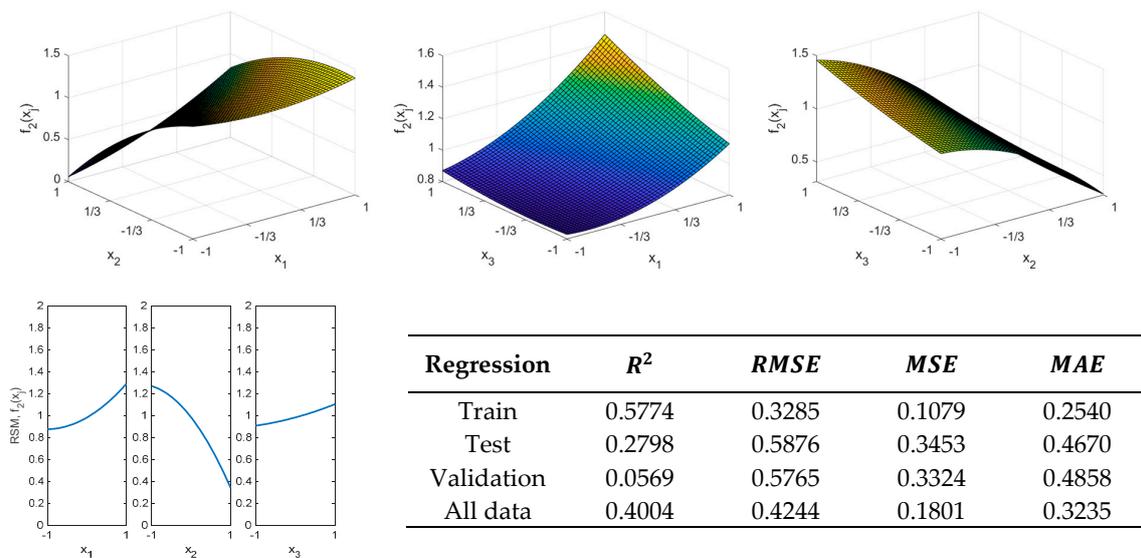


Figure 6. Response surfaces obtained, main effects plot and results of the validation metrics obtained with the RSM model for the case of type 2 function (f_2).

On the other hand, Figure 6 shows the regression results for f_2 function. As can be observed, the regression model cannot adequately predict the behavior of f_2 function. As mentioned earlier, if the evaluation metrics used to assess the goodness of fit (R^2 , RMSE, MBE, etc) provide appropriate values, the RSM model could be used to analyze the behavior of the response variables versus the inputs as well as analyzing the interaction factors. However, this is not the case for f_2 function and therefore, in this case, when using methodologies such as RSM, it would not be possible to trust the model obtained. Therefore, it is necessary to use some other types of methods that allow a better approximation to be obtained from the real behavior of the dependent variable.

As previously mentioned, from the validation metrics shown in Figures 4 and 6 it can be affirmed that the RSM model is suitable for modeling a function of type 1 (f_1) and, therefore, it can be used to analyze the influence of the input variables on the response variables despite the fact that the values obtained in the validation and in the test are lower than those of the training. For example, as can be seen in Figure 4, increasing the values of variables x_1 and x_3 increases f_1 function and increasing x_2 leads to a decrease in the function. If the values obtained with the regression are compared with those that would be obtained with the actual curve, which is shown in Figure 5, it is shown that in this case the regression provides adequate values and, therefore, it could be used to achieve an approximation to the actual behavior of the function. On the contrary, for f_2 function large discrepancies can be observed in the main effect plots, when comparing the actual results with those obtained from RSM, which are shown in Figures 5 and 6, respectively. As can be seen in Figure 6, f_2 function cannot be properly modeled by a regression using the RSM. Therefore, soft computing techniques will be used to avoid the above-mentioned drawbacks of the RSM models.

4.2. Artificial Neural Network (ANN) Modeling

As shown in the previous section, the RSM model could be used for modeling the behavior of f_1 function. However, in the case of f_2 function, RSM is not able to correctly predict its behavior. Therefore, other types of soft computing-based tools will be used for its modeling. First, a neural network will be trained, using the train data shown in Tables 2 and 3. The neural network to be used is shown in Figure 7, where the inputs correspond to the DOE factors [x_1, x_2, x_3]. In this present study the dimension of the input vector will be $[3 \times 44]$ since the neural network is to be trained with the 44 points indicated in Tables 2 and 3, for f_1 and f_2 , respectively. The number of neurons in the hidden layer will be varied, starting from $n = 1$, as shown in Figure 7 and the transfer function to be used

is a hyperbolic tangent sigmoid transfer function [64] which is shown in Equation (4). In addition, a Levenberg–Marquardt back-propagation algorithm will be used, using the functions of the Deep Learning Toolbox™ of Matlab™2019b [64]. From the above, a neural network will be used for f_1 and another for f_2 , since due to the variability that exists between the data generated from types 1 and 2 functions it is likely to have a different structure. To obtain the model fitted with the ANN, an iterative process will be carried out for each function that will consist of calculating $m = 50$ neural networks with each ANN structure and, among those ANNs, the one which obtain, first, the lowest MSE in the validation, and secondly the best results in the other validation metrics, will be chosen.

$$\tan \operatorname{sig}(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{4}$$

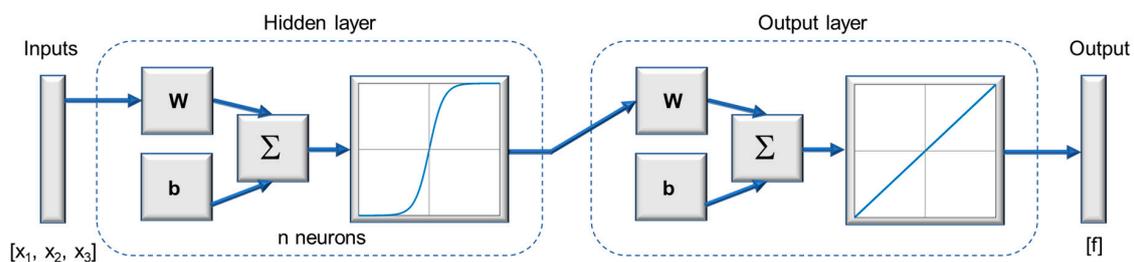


Figure 7. ANN with one hidden layer made of n neurons.

Figure 8 shows the results of the validation metrics shown in Equation (3), obtained using the ANN structure shown in Figure 7 with $n = 1$ neuron in the hidden layer. This ANN was obtained in the 28th iteration. As can be observed, these values are better than those obtained with the regression model, in the case of the f_1 function, which has been shown in Figure 4.

Iteration	$MSE_{validation}$	R^2_{Train}	R^2_{Test}	$R^2_{Validation}$	$R^2_{All\ data}$
2	0.0117	0.9527	0.9440	0.9233	0.9493
28	0.0115	0.9522	0.9469	0.9248	0.9496
35	0.0117	0.9528	0.9442	0.9232	0.9494

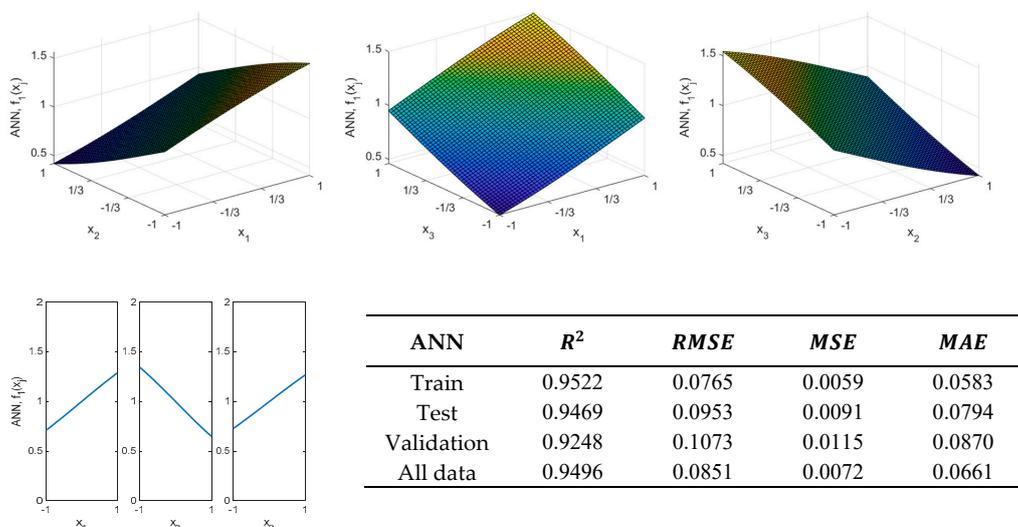


Figure 8. Response surfaces obtained, main effects plot and results of the validation metrics obtained with the ANN model for the case of type 1 function (f_1) ($n = 1$ neurons).

Equation (5) shows the results obtained by modeling the f_1 function using an ANN with one neuron in the hidden layer. In this case, the obtained expression is somewhat simpler than that obtained with the RSM and provides slightly better results. Therefore, it could be used to model the response of the f_1 function. If more precision is required, it would be necessary to increase the number of neurons in the hidden layer, but this would lead to more complex expressions to be obtained.

$$f_1 = 0.9789 - 0.9147 \left(\frac{2}{1 + e^{(-2(-0.3272x_1 + 0.3930x_2 - 0.3053x_3 - 0.0284))}} - 1 \right) \tag{5}$$

As previously mentioned, if more precision is required, an additional neuron could be added to the hidden layer. Thus, for example, if the hidden layer is made up of two neurons, the results shown in Figure 9 are obtained, where it is possible to observe that the ANN with two neurons, obtained in the 6th iteration, allows obtaining improved results in the validation metrics.

Iteration	$MSE_{validation}$	R^2_{Train}	R^2_{Test}	$R^2_{Validation}$	$R^2_{All\ data}$
4	0.0033	0.9793	0.9851	0.9832	0.9806
6	0.0033	0.9794	0.9851	0.9832	0.9806

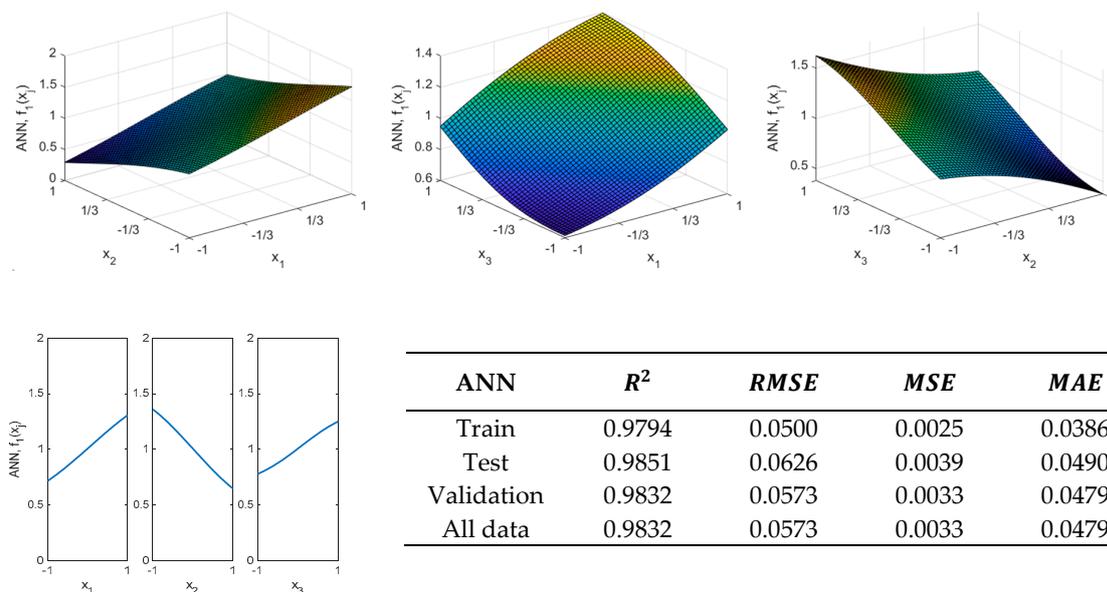


Figure 9. Response surfaces obtained, main effects plot and results of the validation metrics obtained with the ANN model for the case of type 1 function (f_1) ($n = 2$ neurons).

Moreover, it is possible to increase the number of neurons in the hidden layer. However, the obtained model will be more complex. Figure 10 shows results obtained when using the ANN, found in the 10th iteration, which has eight neurons in the hidden layer.

If the $RMSE$ results are compared between the RSM and ANN models, which are shown in Figures 4 and 9 ($n = 2$ neurons), i.e., $(RSM_{values} - ANN_{values}) / RSM_{values}$, it is observed that ANN_{values} are lower in all cases (-27.22% , -35.66% , -60.99% , -36.33%). A similar result is observed with the MAE (-27.99% , -45.25% , -58.67% , -39.57%). Moreover, Figure 11 shows that the coefficients of determination of the regression are higher in all cases (0.98, 0.99, 0.98, 0.98). Regarding the ANN with ($n = 1$ neurons).

Iteration	$MSE_{Validation}$	R^2_{Train}	R^2_{Test}	$R^2_{Validation}$	$R^2_{All\ data}$
10	8.75×10^{-6}	1.0000	0.9999	1.0000	1.0000
23	1.94×10^{-5}	1.0000	0.9999	0.9999	1.0000
27	3.63×10^{-5}	0.9998	0.9985	0.9998	0.9996

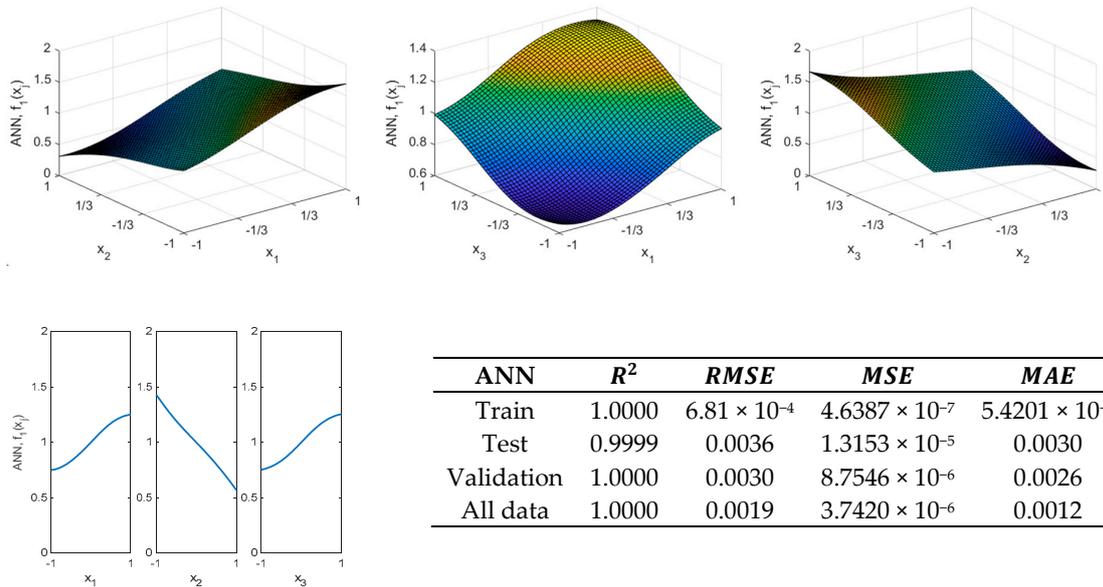


Figure 10. Response surfaces obtained, main effects plot and results of the validation metrics obtained with the ANN model for the case of type 1 function (f_1) ($n = 8$ neurons).

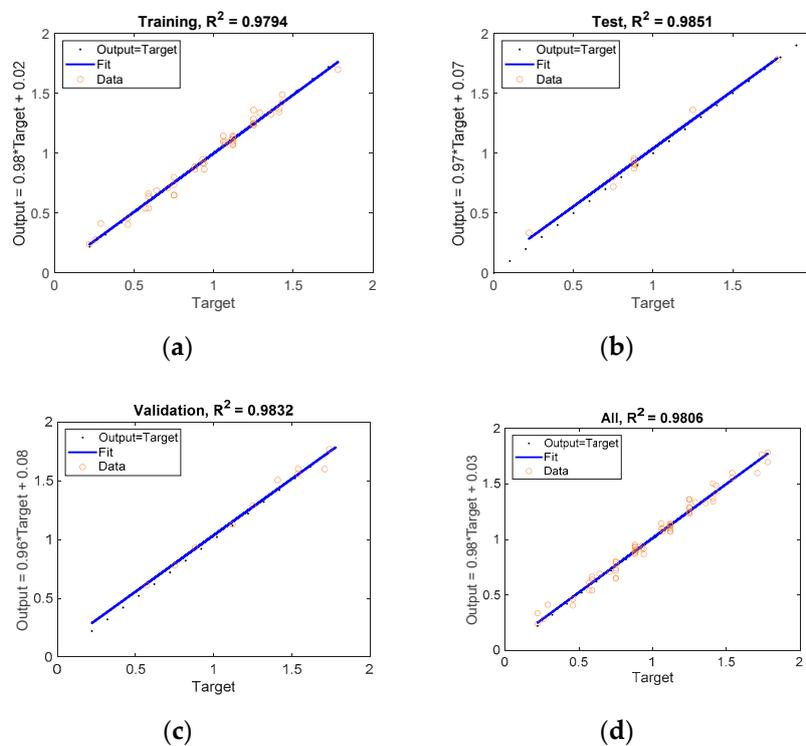


Figure 11. ANN ($n = 2$ neurons) results for f_1 (a) Train data; (b) Validation data; (c) Test data; (d) All data.

Figure 8 shows that the validation metrics obtained with this model are somewhat higher with the train data, but they are lower in the validation, test, and with all data (11.35%, −2.06%, −26.96%, −5.44%). A similar result is obtained with MAE (8.77%, −11.28%, −24.94%, −4.20%). In any case, since the RSM in this case provides values close to those given by the neural network, in this particular case the RSM model could be used since the values of the coefficient of determination of the regression for the f_1 function are (0.96, 0.93, 0.87, 0.94) in the analyzed cases (train, test, validate, and all data). However, the model with one neuron provides a somewhat improved model (0.95, 0.94, 0.92, 0.95) compared to that obtained with the RSM.

In the case of f_2 function a Levenberg–Marquardt back-propagation algorithm is also used in order to update weights and biases of the ANN by employing the Deep Learning Toolbox™ of Matlab™2019b [64]. Figure 12 shows results obtained with one neuron in the hidden layer.

Iteration	$MSE_{Validation}$	R^2_{Train}	R^2_{Test}	$R^2_{Validation}$	$R^2_{All\ data}$
9	0.2224	0.2752	0.8686	0.2889	0.3611
23	0.2431	0.5170	0.4585	0.2133	0.4574
46	0.2468	0.5184	0.4435	0.2091	0.4534

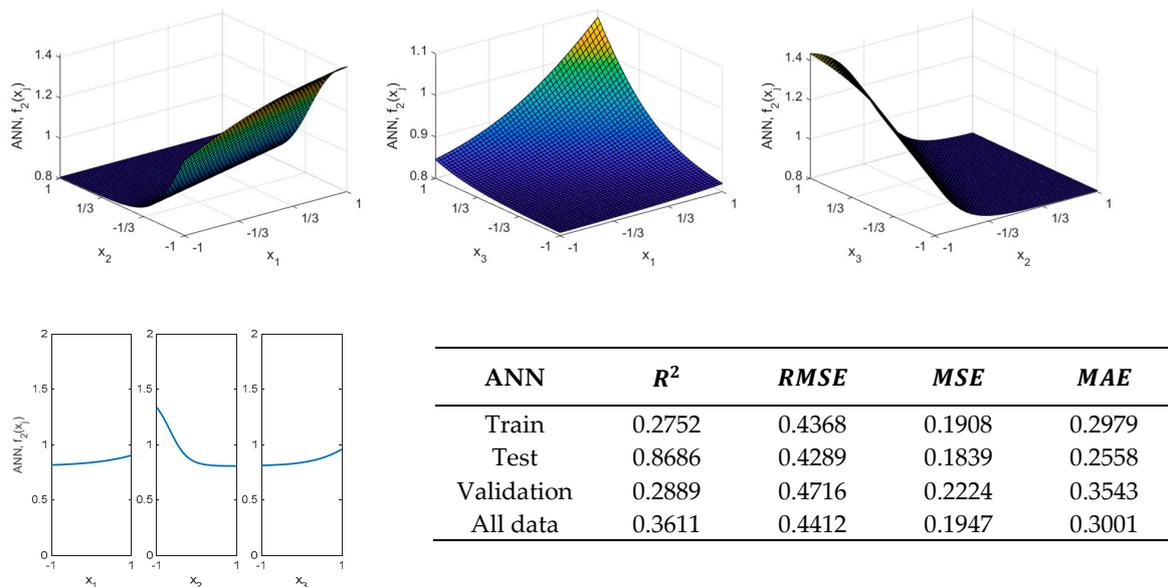


Figure 12. Response surfaces obtained, main effects plot and results of the validation metrics obtained with the ANN model for the case of type 2 function (f_2) ($n = 1$ neurons).

As Figure 12 shows, regression results do not improve with a neuron. Therefore, the number of neurons in the hidden layer will be increased until better results are obtained. For example, with two neurons in the hidden layer, an increase in the coefficient R^2 as well as a decrease in both RMSE and MAE are attained. Figure 13 shows that the validation metrics significantly improve, compared to those obtained with one neuron in the hidden layer when using the ANN obtained in the 6th iteration.

Iteration	$MSE_{Validation}$	R^2_{Train}	R^2_{Test}	$R^2_{Validation}$	$R^2_{All\ data}$
2	0.0312	0.9397	0.8665	0.9246	0.9228
6	0.0222	0.8963	0.8533	0.9662	0.8971
29	0.0323	0.9378	0.8676	0.9322	0.9222

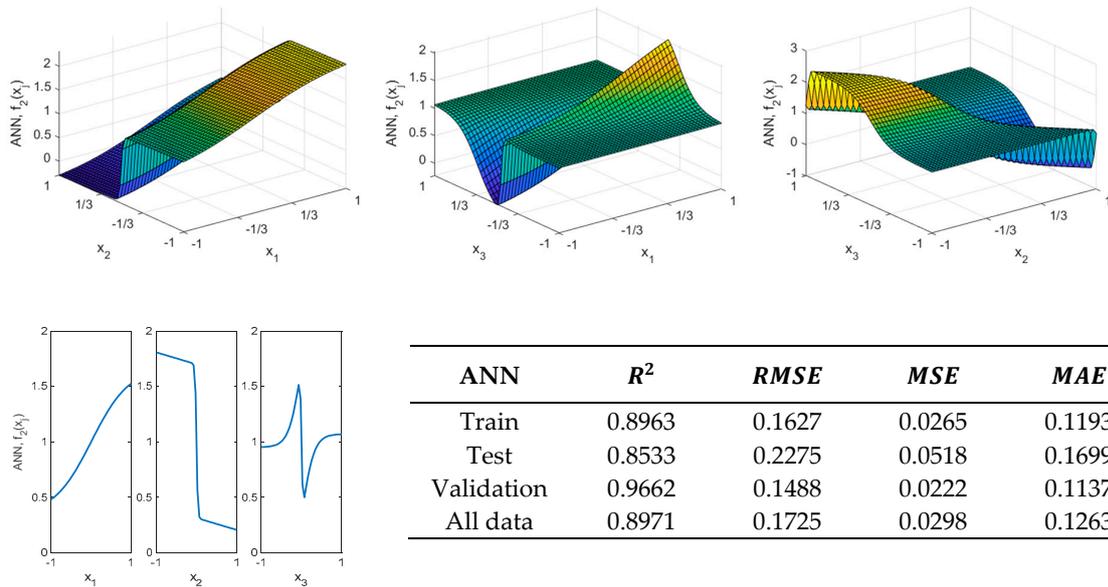


Figure 13. Response surfaces obtained, main effects plot and results of the validation metrics obtained with the ANN model for the regression in the case of type 2 function (f_2) ($n = 2$ neurons).

Figure 14 shows that the R-squared result is approximately 1 in the train data with the neural network, obtained in the 48th iteration, which has 8 neurons in the hidden layer. In addition, the results of the validation metrics are also improved for both test and validation data. However, as can be observed by comparing Figures 2 and 14, this neural network is neither able to reproduce the actual shape of the f_2 function. Furthermore, a two-layer ANN could also be used for modeling f_2 , as Figure 15 shows.

Although the transfer functions of this ANN are not required to be the same, those shown in Equation (4) will be used. In addition, a Levenberg–Marquardt back-propagation algorithm will be used by employing the Deep Learning Toolbox™ of Matlab™2019b [64].

As Figure 16 shows, by using two hidden layers a significant improvement is attained in the validation metrics shown by Equation (3). Figure 17 shows that the ANN₃₋₄₋₂₋₁, which is obtained after the 42nd iteration, can provide higher values of the coefficient of determination. Although the complexity of the model is higher than that of RSM, the RSM model does not provide adequate values for modeling the f_2 function.

On the other hand, the neural network, despite providing values closer to the behavior of f_2 , requires many parameters for this. In any case, the neural network can model the behavior of the f_2 function much better than the RSM and can provide much better values in the validation metrics than those obtained with the regression model. However, this ANN is neither able to fully reproduce the actual shape of the f_2 function, as can be observed when comparing the response surfaces obtained in Figures 2 and 16.

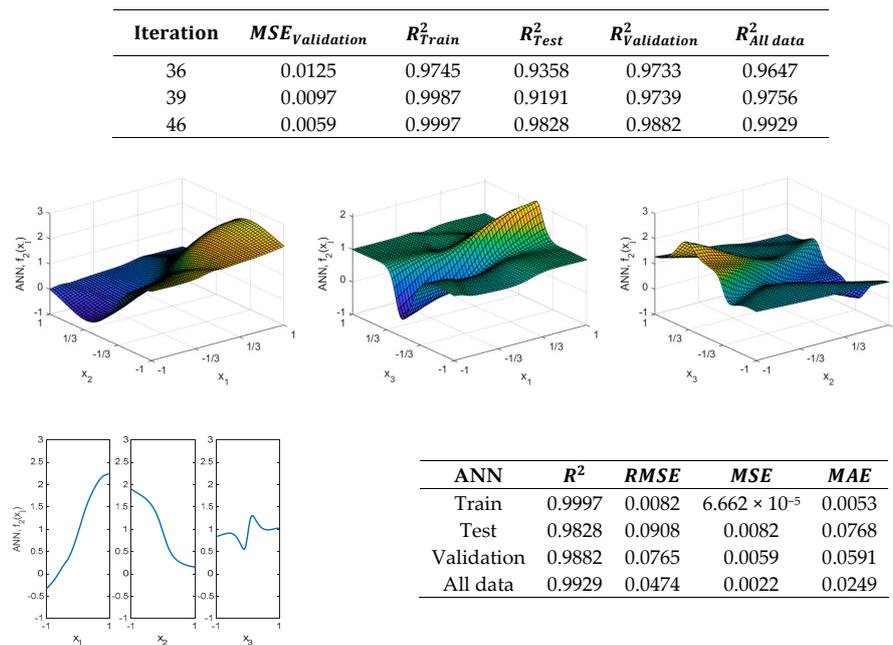


Figure 14. Response surfaces obtained, main effects plot and results of the validation metrics obtained with the ANN model for the case of type 2 function (f_2) ($n = 8$ neurons).

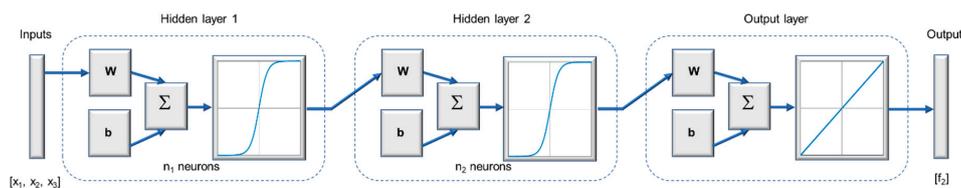


Figure 15. ANN with two hidden layers for modeling f_2 .

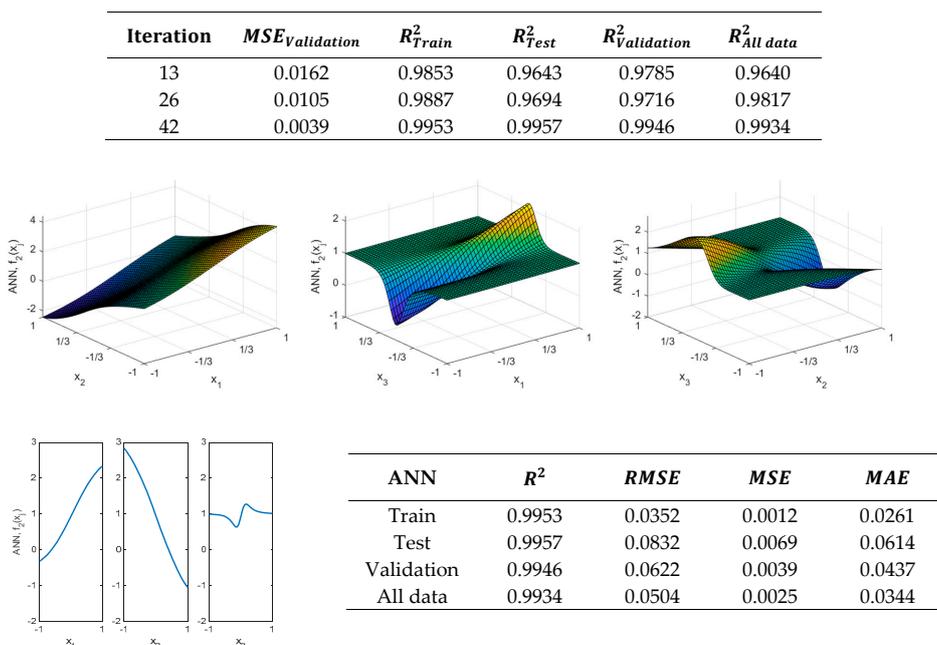


Figure 16. Response surfaces obtained, main effects plot and results of the validation metrics obtained with the ANN₃₋₄₋₂₋₁ model for the regression in the case of type 2 function (f_2).

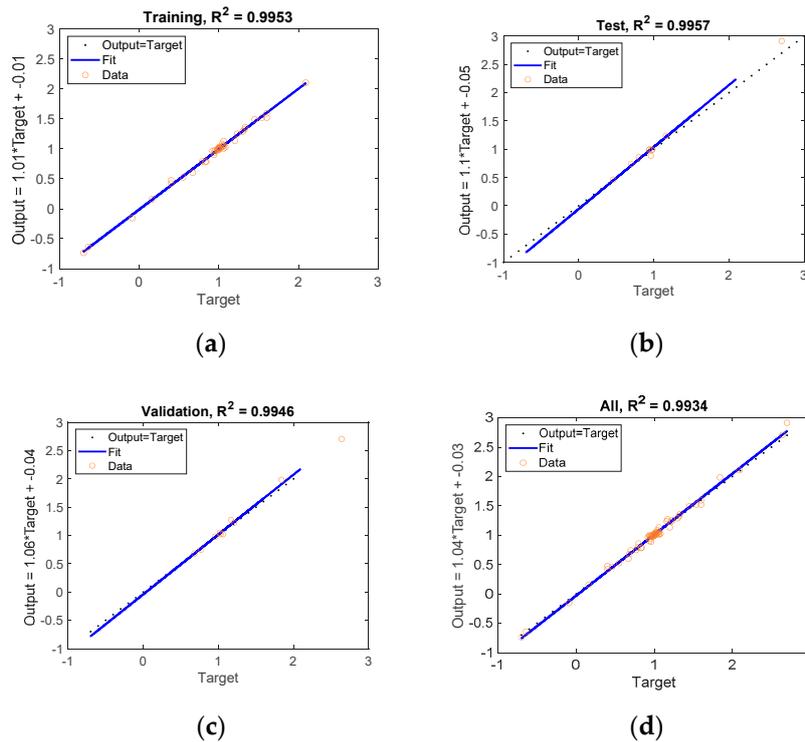


Figure 17. ANN₃₋₄₋₂₋₁ results for f_2 . (a) Train data; (b) Validation data; (c) Test data, (d) All data.

4.3. Adaptive Neuro-Fuzzy Inference System (ANFIS) Modeling

In this present study, two strategies are shown to model the behavior of types 1 and 2 functions using an ANFIS. First, a modeling based on the Matlab™2019b functions is used, which is shown in this current Section 4.3, and secondly, in Section 4.4, a new methodology is proposed. Figure 18 shows the two strategies that will be considered.

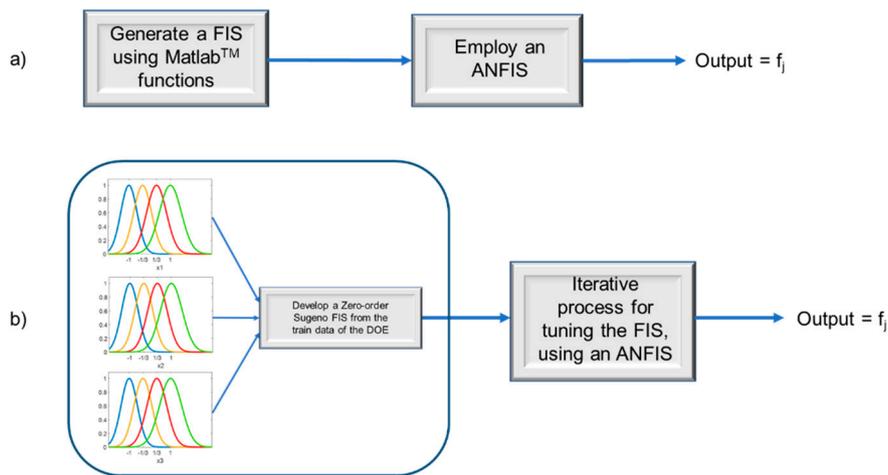


Figure 18. Strategies for obtaining an ANFIS for modeling f_1 and f_2 . (a) Using Matlab™ functions, (b) Proposed methodology.

With regard to the first strategy, the “genfis” function of Matlab™2019b [66] will be used to obtain a zero-order Sugeno FIS and then an ANFIS will be used to tune the parameters of the membership functions. Secondly, another strategy is proposed in this study which consists of using an iterative process to obtain the FIS from the DOE and then using an ANFIS for tuning the constants of the

membership functions. As will be shown in Section 4.4, with this proposed methodology, it is possible to obtain improved results in both the validation metrics and in the modeling of the actual shape of the function.

As was previously mentioned, a zero-order Sugeno FIS is employed by using the Fuzzy Logic Toolbox™ of Matlab™2019b [66], because the de-fuzzification process for a Sugeno system is computationally more efficient compared to that of a Mamdani system [66–68]. However, it should be mentioned that Mamdani systems are more intuitive and the rules are easier to understand, making them more suitable for expert systems, developed from human knowledge [66–68]. Both types of FIS could have been used.

Figure 19 shows two membership function that could be used for fuzzification of the inputs. The membership functions to be used in this study are of the Gaussian type. However, some other types of membership functions such as triangular or bell-shape could have been used.

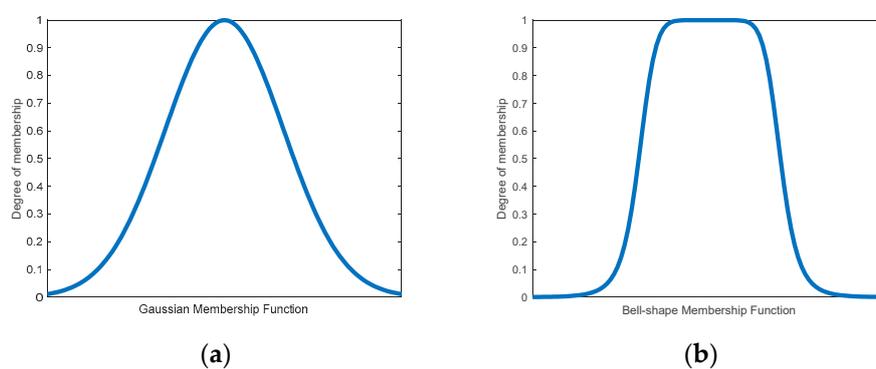


Figure 19. Degree of membership (a) Gaussian (b) Bell-shape.

As previously mentioned, the first strategy is to use the Matlab™2019b “genfis” function [66] to obtain a zero-order Sugeno FIS from the input data. It should be mentioned that it would have been possible to start from the inputs and the outputs and to use some algorithm of the type pattern search or genetic algorithm in order to find a set of rules that model the behavior of the response variables. However, this has not been done in this present study. Furthermore, the membership functions for fuzzification of the independent variables are Gaussian as shown in Equation (6).

$$\mu_x = e^{-\frac{(x-c)^2}{2\sigma^2}} \tag{6}$$

The aggregation method is the sum of fuzzy sets, and the aggregated output is obtained from the weighted average of all output rules. For the i_{th} rule, the implication method is obtained from Equation (7), where product implication method is used in Sugeno systems [66] and Equation (8) shows the final output of the Sugeno system.

$$\lambda_j(x) = \text{AndMethod}\{\mu_{j1}(x_1), \dots, \mu_{jn}(x_n)\} \tag{7}$$

$$\{f_1, f_2\} = \frac{\sum_{j=1}^{\text{Number of rules}} \lambda_j * z_j}{\sum_{j=1}^{\text{Number of rules}} \lambda_j} \tag{8}$$

The created FIS will have a set of “l” rules of the form:

$$\text{If } (x_1 \text{ is } x_{1,i}) \text{ and } (x_2 \text{ is } x_{2,j}) \text{ and } (x_3 \text{ is } x_{3,k}) \text{ then } ([f_1 \mid f_2] \text{ is output}_l)$$

From the above, a zero-order Sugeno FIS [2,3,66] is selected with Gaussian membership functions. Figure 20 shows the Gaussian membership functions for the zero-order Sugeno FIS employed for

modeling f_1 , before and after using the ANFIS. It should be mentioned that a greater number of membership functions could have been used for modeling f_1 , but two membership functions for each of the variables was enough. Therefore, 8 rules are generated for the zero-order Sugeno FIS. To train the fuzzy system an ANFIS employing the Fuzzy Logic Toolbox of Matlab™ along with a back-propagation algorithm has been used [66]. This training process adjusts the membership function parameters of a FIS such that the system models the input/output data.

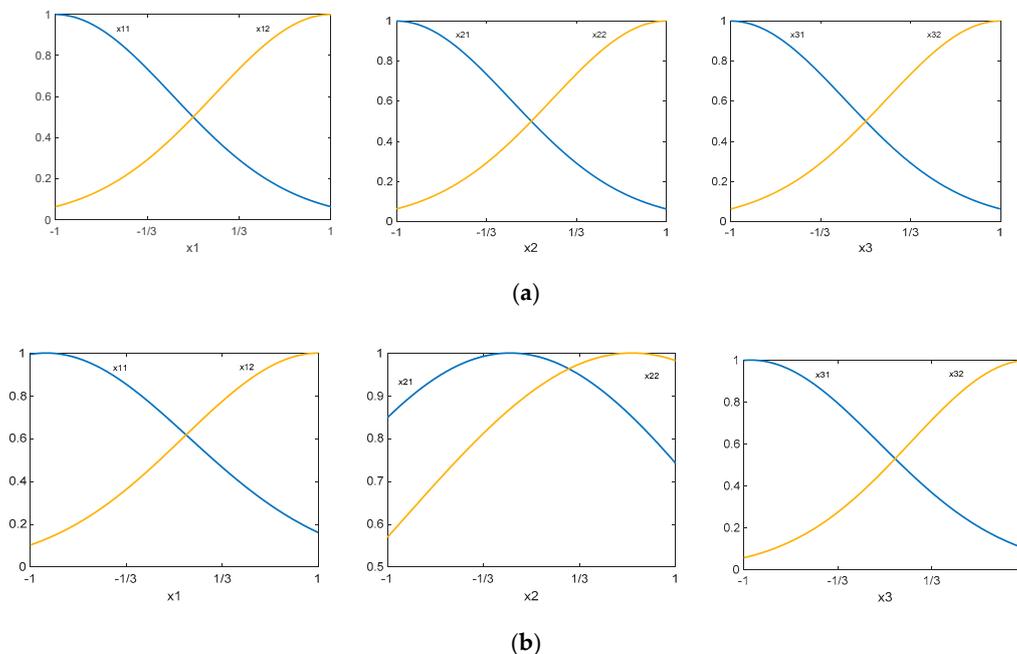


Figure 20. Membership functions for modeling type 1 function (f_1) (a) before the ANFIS and (b) after the ANFIS.

Table 6 shows the values of the Gaussian membership functions, where σ and c are the parameters of the Gaussian function shown by Equation (6) and Table 7 shows the rules obtained after the ANFIS. Figure 21 shows the results obtained by using a zero-order Sugeno FIS for modeling f_1 using two Gaussian membership functions for each of the inputs after 100 epochs.

Table 6. Values obtained for the membership of function type 1 (f_1).

	Before		After	
	σ	c	σ	c
x_{11}	0.84932	-1	0.98975	-0.88958
x_{12}	0.84932	1	0.94106	1.0059
x_{21}	0.84932	-1	1.4900	-0.1477
x_{22}	0.84932	1	1.6019	0.6995
x_{31}	0.84932	-1	0.90735	-0.94945
x_{32}	0.84932	1	0.84611	1.0282

Table 7. Values obtained for function type 1 (f_1).

1.	$(x_1 == x_{11}) \& (x_2 == x_{21}) \& (x_3 == x_{31})$	$\rightarrow (f_1 = 2.3719)$
2.	$(x_1 == x_{11}) \& (x_2 == x_{21}) \& (x_3 == x_{32})$	$\rightarrow (f_1 = 3.03)$
3.	$(x_1 == x_{11}) \& (x_2 == x_{22}) \& (x_3 == x_{31})$	$\rightarrow (f_1 = -1.5969)$
4.	$(x_1 == x_{11}) \& (x_2 == x_{22}) \& (x_3 == x_{32})$	$\rightarrow (f_1 = -1.3203)$
5.	$(x_1 == x_{12}) \& (x_2 == x_{21}) \& (x_3 == x_{31})$	$\rightarrow (f_1 = 2.797)$
6.	$(x_1 == x_{12}) \& (x_2 == x_{21}) \& (x_3 == x_{32})$	$\rightarrow (f_1 = 3.6319)$
7.	$(x_1 == x_{12}) \& (x_2 == x_{22}) \& (x_3 == x_{31})$	$\rightarrow (f_1 = -0.83758)$
8.	$(x_1 == x_{12}) \& (x_2 == x_{22}) \& (x_3 == x_{32})$	$\rightarrow (f_1 = -0.38687)$

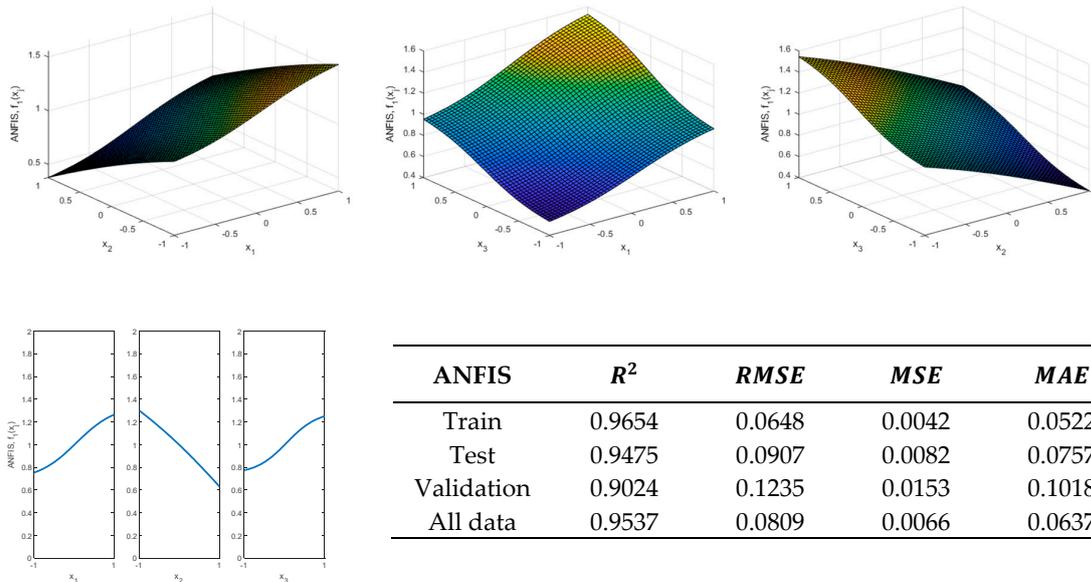


Figure 21. Response surface obtained with the zero-order Sugeno FIS employed for modeling f_1 using two Gaussian membership functions for the inputs along with an ANFIS (100 epochs, 8 fuzzy rules).

Regarding f_2 function, first a zero-order Sugeno was generated. From this, the membership functions before and after using an ANFIS are shown in Figure 22. Figure 23 shows the results obtained by modeling the function using the membership functions shown in Figure 22a and then using an ANFIS. In this case, it is shown in Figure 23 that after 500 epochs it has not been possible to adequately model the function f_2 .

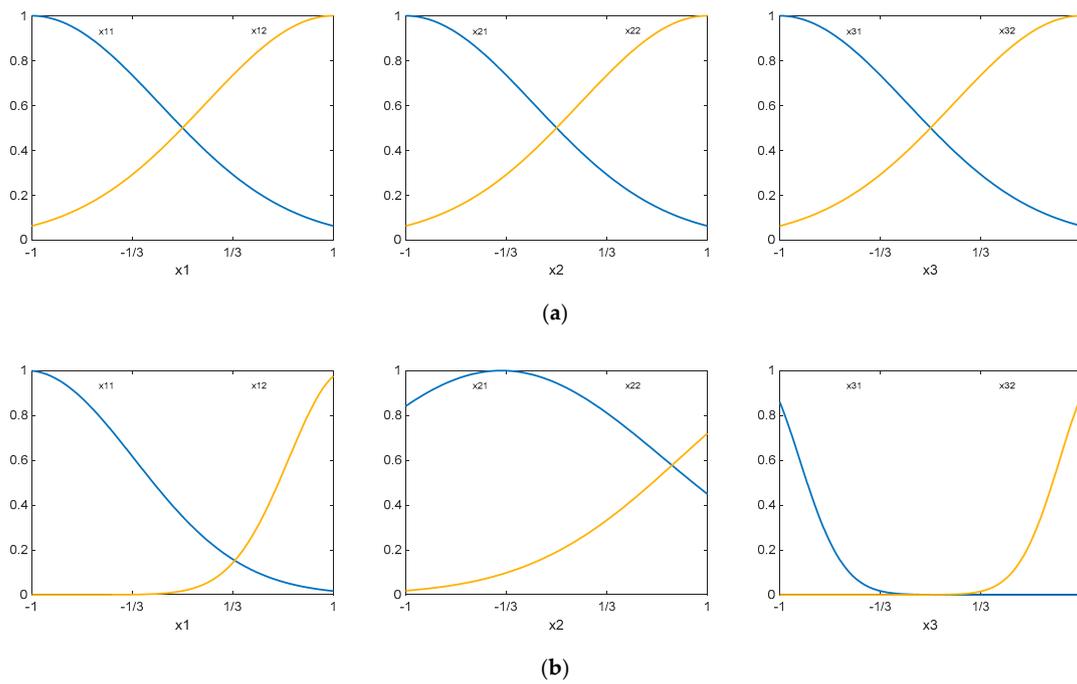


Figure 22. Membership functions for modeling type 2 function (f_2) (a) before the ANFIS and (b) after the ANFIS.

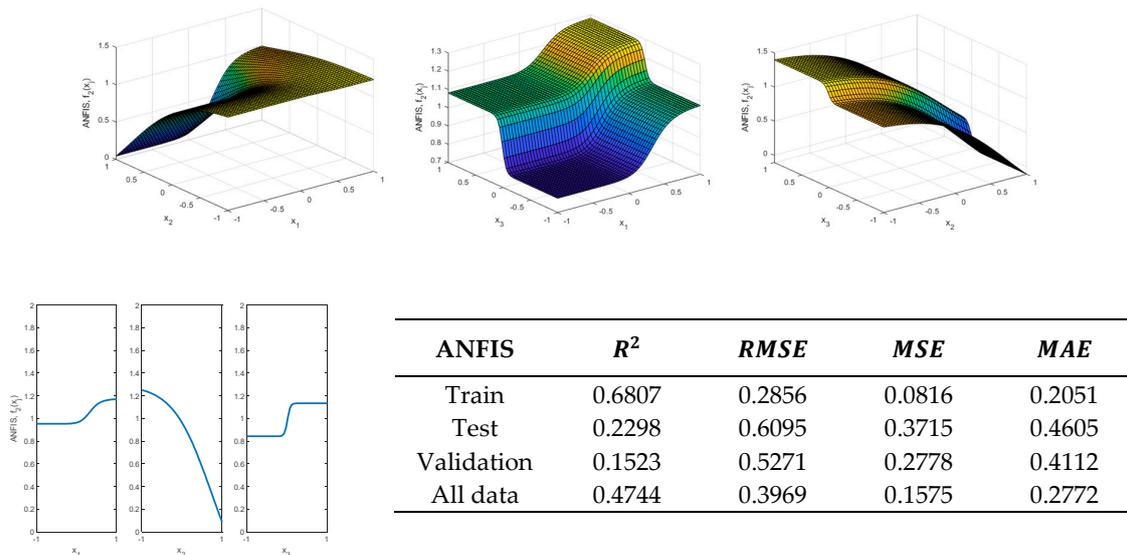


Figure 23. Response surface obtained with the zero-order Sugeno FIS employed for modeling f_2 using two Gaussian membership functions for the inputs along with an ANFIS (500 epochs, 8 fuzzy rules).

If the membership functions are increased to three, a satisfactory result is not obtained either, as Figure 24 shows.

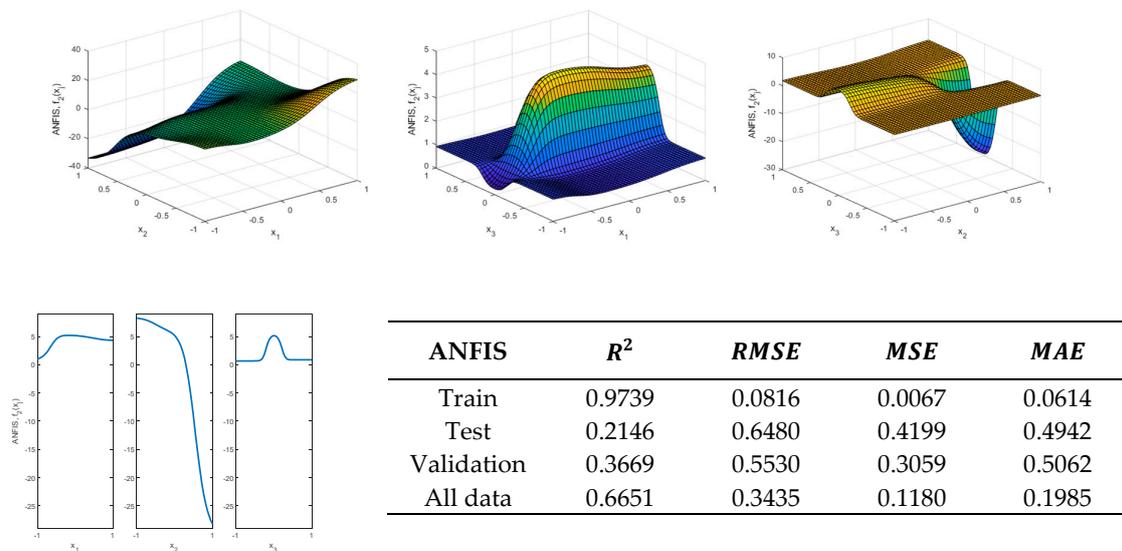


Figure 24. Response surface obtained with the zero-order Sugeno FIS employed for modeling f_2 using three Gaussian membership functions for the inputs along with an ANFIS (100 epochs, 27 fuzzy rules).

A similar behavior is obtained using four membership functions. Although the values of the validation metrics obtained with the train data are adequate, this is not the case with the validation and test data, as Figure 25 shows. Therefore, a new procedure for obtaining an ANFIS is to be proposed in the next section that avoids these drawbacks. As will be shown in the next section, the accuracy of the thus obtained model is higher than that obtained with the previous method.

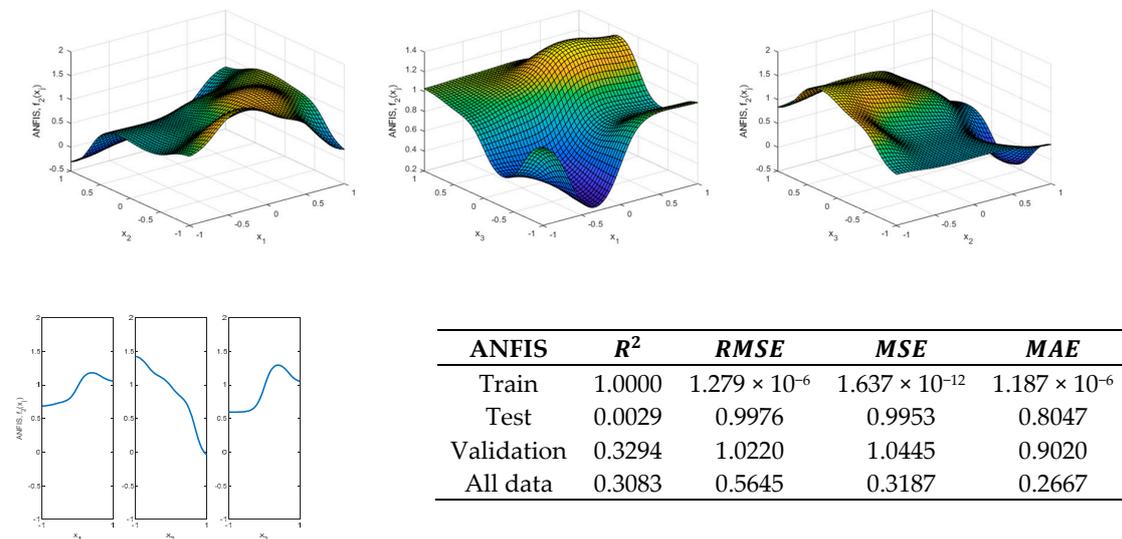


Figure 25. Response surface obtained with the zero-order Sugeno FIS employed for modeling f_2 using four Gaussian membership functions for the inputs along with an ANFIS (2 epochs, 64 fuzzy rules).

4.4. Proposal of a Method for Obtaining an Adaptive Neuro-Fuzzy Inference System

One of the advantages that FIS have over other computing tools is that it is possible to take advantage of the user experience to generate them. In this section, the FIS is generated directly from the train data of the DOE. Therefore, from data shown in Tables 2 and 3, it will be possible to directly obtain the set of rules that make up a zero-order Sugeno FIS. Then an iterative process will be followed, and the FIS parameters will be tuned using an ANFIS, according to the procedure shown below. Once

the adjusted models have been developed, the results obtained will be analyzed in a similar way to that used for the case of RSM and ANN.

The proposed methodology begins by selecting membership functions of the Gaussian type, such as those shown in Equation (9). The standard deviation values of the membership functions are obtained from Equation (10) and the corresponding levels of the DOE will be used for the mean values of the membership functions, as shown in Equation (11). Since the DOE has four levels for the input variables, as shown in Table 1, four Gaussian functions will be employed, as shown in Equations (9)–(11). From this, a zero-order Sugeno FIS will be developed from the DOE and then it will be tuned by using an iterative process and an ANFIS, i.e.:

$$\mu_{i,j} = e^{-\frac{(x-c_{i,j})^2}{2\sigma_{i,j}^2}} \tag{9}$$

$$\sigma_{i,j} = \frac{k_{FIS,i}^{j-1} * (x_{max,i} - x_{min,i})}{N_{FIS,i}} \tag{10}$$

$$c_{i,j} = \text{Level}(x_{i,j}) \tag{11}$$

where the levels of the variables $c_{i,j} = \text{Level}(x_{i,j})$ are obtained from the DOE shown in Table 1, where $j = 1..n =$ number of membership functions, $x_{max} = \max(x_i)$, $x_{min} = \min(x_i)$, $k_{i,j}$ and $N_{FIS,i}$ can be increased with respect to the previous one in order to introduce an imbalance in the form of the membership functions, if necessary, or they can be the same. Based on the above, different initial FIS can be obtained by varying the values of $k_{FIS,i}$ and $N_{FIS,i}$. As was previously mentioned, a zero-order Sugeno FIS will be used in this study. The proposed procedure is based on adjusting the parameters of the membership functions using an iterative process combined with an ANFIS.

To summarize the proposed methodology, the algorithm that has been programmed to obtain the results shown in this current section is as follows:

- (1) Select the range of variation of $k_{FIS,i}$ and $N_{FIS,i}$, i.e., $k_{min,i} \leq k_{FIS,i} \leq k_{max,i}$ and $N_{min,i} \leq N_{FIS,i} \leq N_{max,i}$
- (2) Select the increment for variation of $k_{FIS,i}$ and $N_{FIS,i}$
- (3) Select the maximum number of Epochs as well as the initial value of α and β , where α is the desired level for R^2_{Train} and β is the desired level for $R^2_{Validation}$.
- (4) Determine the strategy to calculate the values of the membership functions constants.

(4.1) A full iteration can be followed to look for $k_{FIS,i}$ and $N_{FIS,i}$.

To reduce the running time, two strategies may be considered (either 4.2. or 4.3).

(4.2) It is assumed that $k_{FIS,i} = k_{FIS}$ and $N_{FIS,i} = N_{FIS}$ (That is a double iteration is obtained).

(4.3) Using a constant k_{FIS} for all the membership functions and then iterate to obtain $N_{FIS,i}$ (That is: assume a value for k_{FIS} . For example, $k_{FIS} = 1$ and then iterate to obtain $N_{FIS,i}$) (That is a triple iteration is obtained).

Then, the algorithm continues as follows:

(4.1a) Iterate to obtain $k_{FIS,i}$ and $N_{FIS,i}$. (As shown in Algorithms 1, if exit n inputs, then n iterations would be obtained).

Algorithms 1. Iterate to obtain $k_{FIS,i}$ and $N_{FIS,i}$. (Case of n inputs)

```

for  $k_{FIS,1} = k_{FIS,1_{min}} : \text{Increment}_{k_{FIS,1}} : k_{FIS,1_{max}}$ 
  ...
  for  $k_{FIS,n} = k_{FIS,n_{min}} : \text{Increment}_{k_{FIS,n}} : k_{FIS,n_{max}}$ 
    for  $N_{FIS,1} = N_{FIS,1_{min}} : \text{Increment}_{N_{FIS,1}} : N_{FIS,1_{max}}$ 
      ...
      for  $N_{FIS,n} = N_{FIS,n_{min}} : \text{Increment}_{N_{FIS,n}} : N_{FIS,n_{max}}$ 
        Obtain a zero-order Sugeno FIS,  $FIS_{initial} = \text{function of } (k_{FIS,i}, N_{FIS,i}, \text{data}_{train})$ 
        Tune  $FIS_{initial}$  using an ANFIS, that is:
        For Epochs = 1:  $\text{Increment}_{Epochs} : Epochs_{max}$ 
           $FIS_{out} = \text{ANFIS}(FIS_{initial}, \text{data}_{train}, Epochs)$ 
          Evaluate  $MSE_{OutValidation}$  using  $FIS_{out}$  and  $\text{data}_{validation}$ 
        end
        Find  $\min\{MSE_{OutValidation}\}$  and  $Epochs_{minMSE}$  to obtain  $\min\{MSE_{OutValidation}\}$ 
         $FIS_{out} = \text{ANFIS}(FIS_{initial}, \text{data}_{train}, Epochs_{minMSE})$ 
        Evaluate  $R^2_{Train}$  using  $FIS_{out}$  and  $\text{data}_{Train}$ 
        Evaluate  $R^2_{Validation}$  using  $FIS_{out}$  and  $\text{data}_{Validation}$ 
        If  $((R^2_{Train} \geq \alpha) \&\& (R^2_{Validation} \geq \beta))$ 
          Obtained FIS =  $FIS_{out}$ 
          Store calculated FIS,  $\min\{MSE_{OutValidation}\}$ ,  $Epochs_{minMSE}$ ,  $k_{FIS,i}$  and  $N_{FIS,i}$ 
          Exit of the iteration loop
        Else
          Store  $\min\{MSE_{OutValidation}\}$ ,  $Epochs_{minMSE}$ ,  $k_{FIS,i}$  and  $N_{FIS,i}$ 
        end
      end
    end
  end
end
  ...
end
  ...
end
  ...
end

```

(4.1b) Iterate to obtain $k_{FIS,i}$ and $N_{FIS,i}$. (The present study considers three inputs variables. As shown in Algorithms 2, if a full iteration is carried out, then six iterations would be obtained).

Algorithms 2. Iterate to obtain $k_{FIS,i}$ and $N_{FIS,i}$. (Case of 3 inputs)

```

for  $k_{FIS,1} = k_{FIS,1_{min}} : \text{Increment}_{k_{FIS,1}} : k_{FIS,1_{max}}$ 
  for  $k_{FIS,2} = k_{FIS,2_{min}} : \text{Increment}_{k_{FIS,2}} : k_{FIS,2_{max}}$ 
    for  $k_{FIS,3} = k_{FIS,3_{min}} : \text{Increment}_{k_{FIS,3}} : k_{FIS,3_{max}}$ 
      for  $N_{FIS,1} = N_{FIS,1_{min}} : \text{Increment}_{N_{FIS,1}} : N_{FIS,1_{max}}$ 
        for  $N_{FIS,2} = N_{FIS,2_{min}} : \text{Increment}_{N_{FIS,2}} : N_{FIS,2_{max}}$ 
          for  $N_{FIS,3} = N_{FIS,3_{min}} : \text{Increment}_{N_{FIS,3}} : N_{FIS,3_{max}}$ 
            Obtain a zero-order Sugeno FIS,  $FIS_{initial} = \text{function of } (k_{FIS,i}, N_{FIS,i}, \text{data}_{train})$ 
            Tune  $FIS_{initial}$  using an ANFIS, that is:
            For Epochs = 1:  $\text{Increment}_{Epochs} : Epochs_{max}$ 
               $FIS_{out} = \text{ANFIS}(FIS_{initial}, \text{data}_{train}, Epochs)$ 
              Evaluate  $MSE_{OutValidation}$  using  $FIS_{out}$  and  $\text{data}_{validation}$ 
            end
            Find  $\min\{MSE_{OutValidation}\}$  and  $Epochs_{minMSE}$  to obtain  $\min\{MSE_{OutValidation}\}$ 
             $FIS_{out} = \text{ANFIS}(FIS_{initial}, \text{data}_{train}, Epochs_{minMSE})$ 
            Evaluate  $R^2_{Train}$  using  $FIS_{out}$  and  $\text{data}_{Train}$ 
            Evaluate  $R^2_{Validation}$  using  $FIS_{out}$  and  $\text{data}_{Validation}$ 
            If  $((R^2_{Train} \geq \alpha) \&\& (R^2_{Validation} \geq \beta))$ 
              Obtained FIS =  $FIS_{out}$ 
              Store calculated FIS,  $\min\{MSE_{OutValidation}\}$ ,  $Epochs_{minMSE}$ ,  $k_{FIS,i}$  and  $N_{FIS,i}$ 
              Exit of the iteration loop
            Else
              Store  $\min\{MSE_{OutValidation}\}$ ,  $Epochs_{minMSE}$ ,  $k_{FIS,i}$  and  $N_{FIS,i}$ 
            end
          end
        end
      end
    end
  end
end
  ...
end
  ...
end
  ...
end

```

(4.2) Algorithms 3 shows the procedure to obtain k_{FIS} and N_{FIS} .

Algorithms 3. Assumed $k_{FIS,i} = k_{FIS}$ and $N_{FIS,i} = N_{FIS}$.

```

for  $k_{FIS} = k_{FIS,min} : Increment_{k_{FIS}} : k_{FIS,max}$ 
  for  $N_{FIS} = N_{FIS,min} : Increment_{N_{FIS}} : N_{FIS,max}$ 
    Obtain a zero-order Sugeno FIS,  $FIS_{initial} = function\ of\ (k_{FIS}, N_{FIS}, data_{train})$ 
    Tune  $FIS_{initial}$  using an ANFIS, that is:
    For Epochs = 1:  $Increment_{Epochs} : Epochs_{max}$ 
       $FIS_{out} = ANFIS(FIS_{initial}, data_{train}, Epochs)$ 
      Evaluate  $MSE_{OutValidation}$  using  $FIS_{Out}$  and  $data_{Validation}$ 
    end
    Find  $min\{MSE_{OutValidation}\}$  and  $Epochs_{minMSE}$  to obtain  $min\{MSE_{OutValidation}\}$ 
     $FIS_{out} = ANFIS(FIS_{initial}, data_{train}, Epochs_{minMSE})$ 
    Evaluate  $R^2_{Train}$  using  $FIS_{Out}$  and  $data_{Train}$ 
    Evaluate  $R^2_{Validation}$  using  $FIS_{Out}$  and  $data_{Validation}$ 
    If  $((R^2_{Train} \geq \alpha) \&\& (R^2_{Validation} \geq \beta))$ 
      Obtained  $FIS = FIS_{out}$ 
      Store calculated  $FIS, min\{MSE_{OutValidation}\}, Epochs_{minMSE}, k_{FIS}$  and  $N_{FIS}$ 
      Exit of the iteration loop
    Else
      Store  $min\{MSE_{OutValidation}\}, Epochs_{minMSE}, k_{FIS}$  and  $N_{FIS}$ 
    end
  end
end
end
end

```

(4.3) Algorithms 4 shows the procedure to obtain $N_{FIS,i}$, when a value is assumed for k_{FIS} (For example, $k_{FIS} = 1$).

Algorithms 4. Iterate to obtain $N_{FIS,i}$ (assumed for k_{FIS})

```

for  $N_{FIS,1} = N_{FIS,1,min} : Increment_{N_{FIS,1}} : N_{FIS,1,max}$ 
  for  $N_{FIS,2} = N_{FIS,2,min} : Increment_{N_{FIS,2}} : N_{FIS,2,max}$ 
    for  $N_{FIS,3} = N_{FIS,3,min} : Increment_{N_{FIS,3}} : N_{FIS,3,max}$ 
      Obtain a zero-order Sugeno FIS,  $FIS_{initial} = function\ of\ (k_{FIS}, N_{FIS,i}, data_{train})$ 
      Tune  $FIS_{initial}$  using an ANFIS, that is:
      For Epochs = 1:  $Increment_{Epochs} : Epochs_{max}$ 
         $FIS_{out} = ANFIS(FIS_{initial}, data_{train}, Epochs)$ 
        Evaluate  $MSE_{OutValidation}$  using  $FIS_{Out}$  and  $data_{Validation}$ 
      end
      Find  $min\{MSE_{OutValidation}\}$  and  $Epochs_{minMSE}$  to obtain  $min\{MSE_{OutValidation}\}$ 
       $FIS_{out} = ANFIS(FIS_{initial}, data_{train}, Epochs_{minMSE})$ 
      Evaluate  $R^2_{Train}$  using  $FIS_{Out}$  and  $data_{Train}$ 
      Evaluate  $R^2_{Validation}$  using  $FIS_{Out}$  and  $data_{Validation}$ 
      If  $((R^2_{Train} \geq \alpha) \&\& (R^2_{Validation} \geq \beta))$ 
        Obtained  $FIS = FIS_{out}$ 
        Store calculated  $FIS, min\{MSE_{OutValidation}\}, Epochs_{minMSE}, k_{FIS}$  and  $N_{FIS,i}$ 
        Exit of the iteration loop
      Else
        Store calculated  $FIS$  as well as  $Epochs, MSE_{Out}, k_{FIS}$  and  $N_{FIS,i}$ 
      end
    end
  end
end
end
end
end

```

- (5) Among the results obtained, the one that optimizes the value of $MSE_{Validation}$ and at the same time meets the condition $R^2_{Train} \geq \alpha$ will be selected.
- (6) Once the iterations are finished, FIS_{out} will be obtained. Then both the validation metrics and the response surface could be obtained by using the tuned FIS.

In this present study, the procedure consisting of obtaining k_{FIS} and N_{FIS} will be followed. Once the optimal values (k_{FIS} and N_{FIS}) have been found, in the first case, or N_1, N_2 and N_3 , in the second case, the ANFIS, which optimizes the value of $MSE_{Validation}$, is obtained directly. As previously mentioned, other alternatives will be possible for obtaining $k_{FIS,i}$ and $N_{FIS,i}$, i.e., iterating with the full set of constants. However, they may require higher running time when α and β are increased. If α and β are set to 1, then a full iteration is carried out. Therefore, the results shown below have been obtained with the strategy of iterating to find (k_{FIS} and N_{FIS}), i.e., look for the values of k_{FIS} and N_{FIS} that lead to a lower $MSE_{Validation}$ (MSE value obtained with the validation data). If all simulation data are collected, then several solutions could be obtained and, in that case, the value that minimizes the $MSE_{Validation}$ will be selected. If there exist several solutions, then the values that minimize the R^2 -coefficients obtained with the train, validation, and test data in that order, will be chosen. According to the described procedure, an ANFIS is used together with a back-propagation algorithm to tune the values of the membership functions. Each of the thus obtained FIS are tuned using an ANFIS by employing the functions of the Fuzzy Logic Toolbox of Matlab™ [66]. Table 8 and Figure 26 show the programming results for f_1 function. As indicated, a double iteration is performed to find k_{FIS} and N_{FIS} and at the same time an ANFIS is employed to tune the FIS parameters obtained with each value of k_{FIS} and N_{FIS} .

Table 8. Iteration to find N_{FIS} and k_{FIS} for modeling f_1 function.

Epochs	$MSE_{Validation}$	R^2_{Train}	R^2_{Test}	$R^2_{Validation}$	$R^2_{All\ data}$	N_{FIS}	k_{FIS}
7	8.9098×10^{-5}	1.0000	0.9990	0.9997	0.9997	2.72	1.15
12	8.8023×10^{-5}	1.0000	0.9990	0.9997	0.9996	2.67	1.16
6	8.8487×10^{-5}	1.0000	0.9990	0.9997	0.9996	2.76	1.16

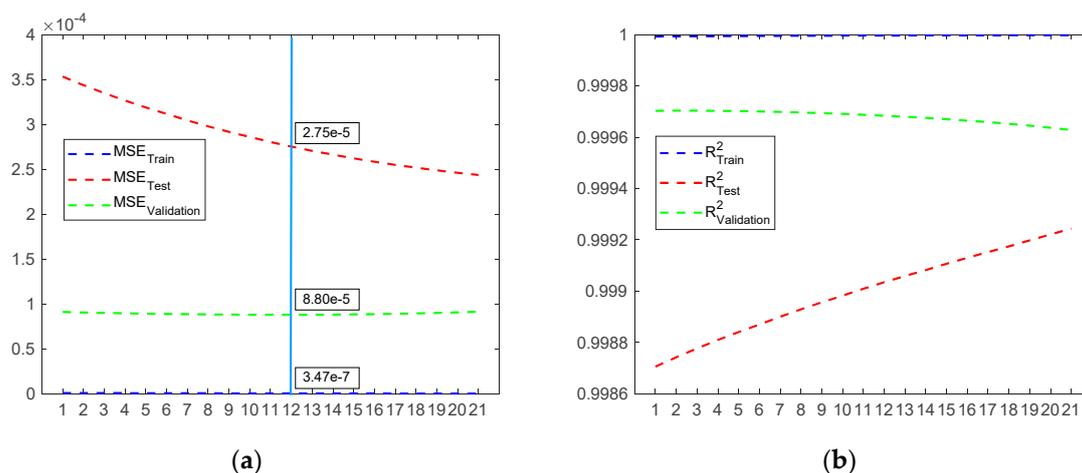


Figure 26. Best MSE and obtained squared-R (Number of Epochs = 10, $k_{FIS} = 1.16$, $N_{FIS} = 2.67$).

Thereafter, the number of epochs will be increased in the ANFIS for tuning the FIS, the number of epochs that optimizes $MSE_{Validation}$ will be selected and at the same time verifies that $R^2_{Train} \geq \alpha$ and that the maximum number of epochs is not reached. This will lead to the optimal of $MSE_{Validation}$ to be obtained and, subsequently, the obtained ANFIS will be evaluated both in the test and in the train data and finally in all the data.

Figures 27 and 28 show the results obtained with the ANFIS generated from the aforementioned methodology. As can be observed, the proposed method is able to find an approximation with high accuracy. The ANFIS thus obtained which led to better results was the one with $k_{FIS} = 1.16$ and $N_{FIS} = 2.67$ and after 10 epochs with the ANFIS.

As can be observed, the ANFIS obtained with the proposed methodology can provide better values of the validation metrics than those obtained with both the RSM model and with the ANN that has 1 and 2 neurons in the hidden layer and the obtained results are similar to those obtained with the ANN that has 8 neurons in the hidden layer, when modeling the behavior of f_1 . However, it should be noted that adding more neurons to the ANN can cause irregularities in the response surface to be obtained. As shown, the FIS obtained through the proposed methodology can provide accurate results that could be used to model the response of type 1 functions (f_1). As will be seen below with the methodology proposed in this present section, it is also possible to obtain satisfactory results, in the case of type 2 functions (f_2), opposite to that obtained in the previous section, as is shown in Figures 23–25.

The results shown in Table 9 and Figure 29 have been obtained with the strategy of iterating to find (k_{FIS} and N_{FIS}) in the case of function f_2 , i.e., look for the values of k_{FIS} and N_{FIS} that lead to a lower $MSE_{Validation}$. As with f_1 , a double iteration is carried out to find k_{FIS} and N_{FIS} and at the same time an ANFIS is employed to tune the FIS parameters obtained with each value of k_{FIS} and N_{FIS} . The number of epochs that optimizes $MSE_{Validation}$ will be selected and at the same time verifies that $R^2_{Train} \geq \alpha$ and that the maximum number of epochs is not reached. Result obtained are $k_{FIS} = 0.87$ and $N_{FIS} = 1.66$ for the initial FIS and the tuned FIS is achieved after 46 epochs using an ANFIS that tune the initial parameters of the FIS adjusted with the previously mentioned k_{FIS} and N_{FIS} . Figure 30 shows that it is possible to properly model the f_2 function using the methodology proposed in this present study. Furthermore, as can be seen in Figures 16 and 30, although the values provided by the validation metrics by using the ANN₃₋₄₋₂₋₁ are slightly better than those obtained by using the ANFIS, the ANFIS is able to approximate the actual shape of the function better than the neural network.

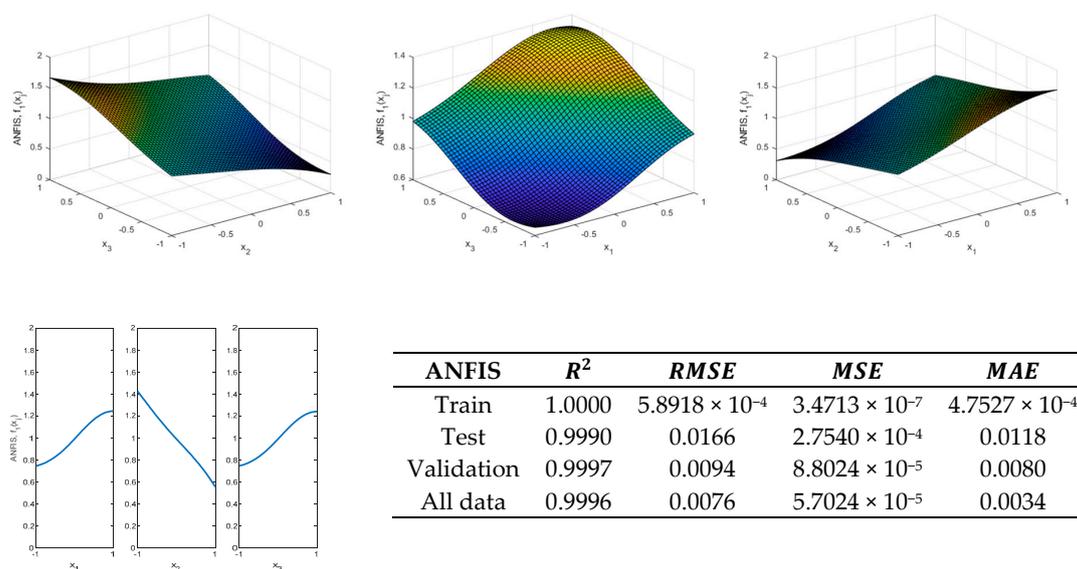


Figure 27. Response surface obtained with the zero-order Sugeno FIS employed for modeling f_1 using four Gaussian membership functions for the inputs and the proposed method for tuning the FIS using an ANFIS (Number of Epochs = 10, $k_{FIS} = 1.16$, $N_{FIS} = 2.67$).

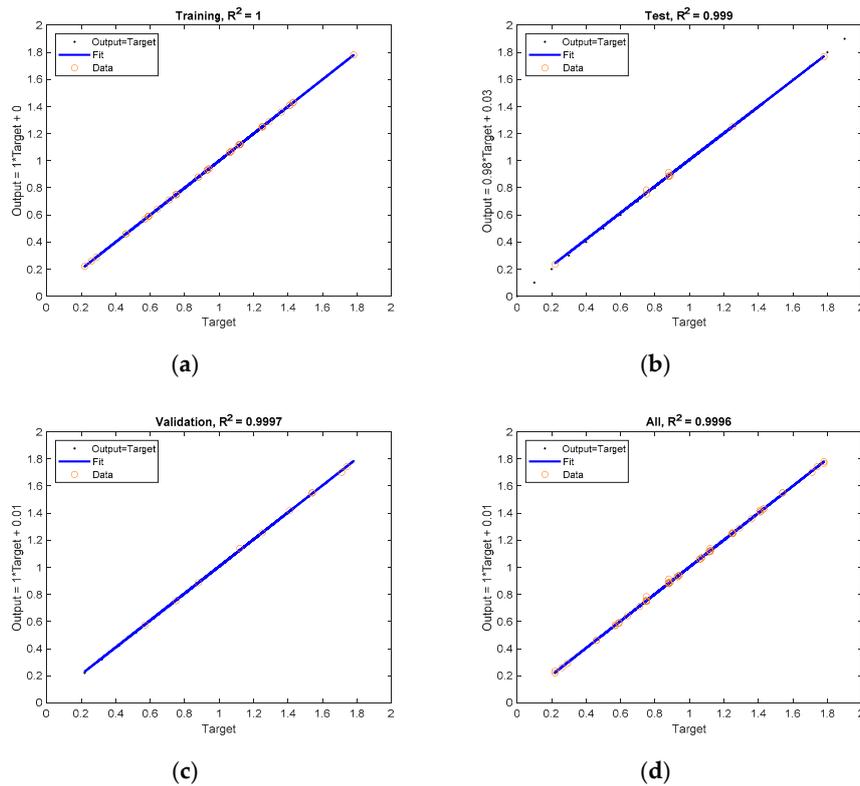


Figure 28. ANFIS results for f_1 . (Number of Epochs = 10, $k_{FIS} = 1.16$, $N_{FIS} = 2.67$) (a) Train data; (b) Validation data; (c) Test data; (d) All data.

Table 9. Iteration to find N_{FIS} and k_{FIS} for modeling f_2 function.

Epochs	$MSE_{Validation}$	R^2_{Train}	R^2_{Test}	$R^2_{Validation}$	$R^2_{All\ data}$	N_{FIS}	k_{FIS}
46	9.1368×10^{-3}	1.0000	0.9773	0.9874	0.9818	1.66	0.87
47	9.1924×10^{-3}	1.0000	0.9768	0.9873	0.9816	1.67	0.88
50	9.4521×10^{-3}	1.0000	0.9755	0.9875	0.9810	1.65	0.89

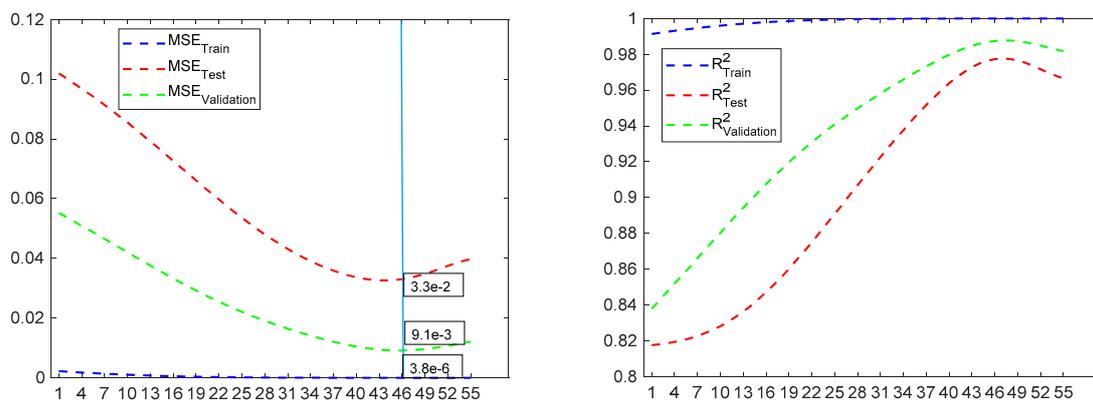


Figure 29. Best MSE and obtained squared-R (Number of Epochs = 46, $k_{FIS} = 0.87$, $N_{FIS} = 1.66$).

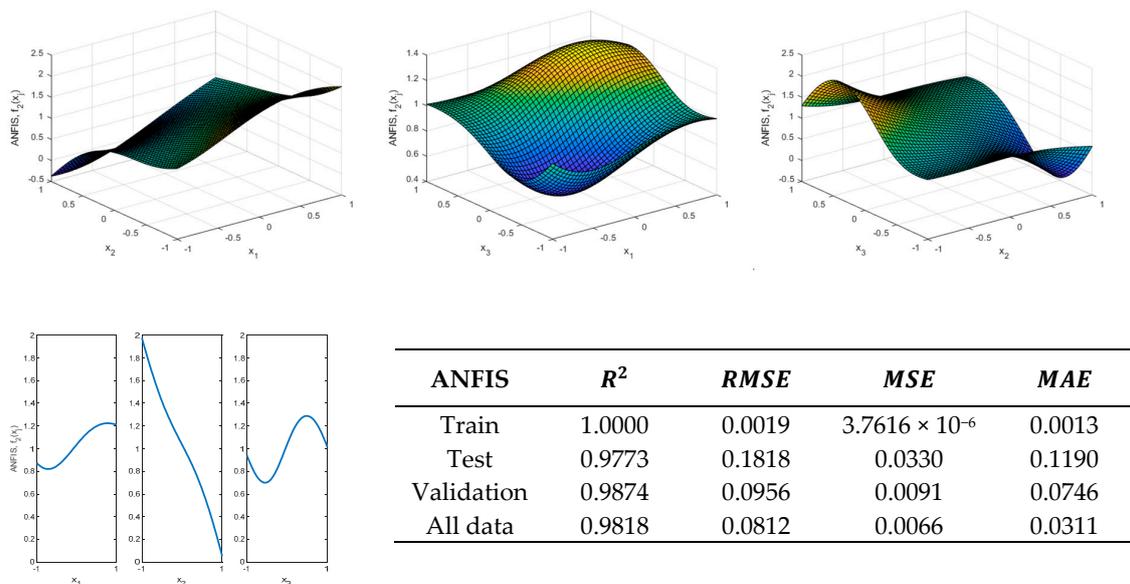


Figure 30. Response surface obtained with the zero-order Sugeno FIS employed for modeling f_2 using four Gaussian membership functions for the inputs and the proposed method for tuning the FIS using an ANFIS (Number of Epochs = 46, $k_{FIS} = 0.87$, $N_{FIS} = 1.66$).

4.5. Analysis of an Actual Study Case

Once the previous f_1 (type 1) and f_2 (type 2) functions have been analyzed, a real case will be outlined, to assess whether the results obtained with the analytical function can be extrapolated to those obtained in a real case. To do this, the same methodology as previously shown will be used, i.e., the design will be divided into one part to train the models and into others for testing and validation. Subsequently, a modeling using RSM and soft computing techniques will be applied. These models will be fitted with the train data and will be validated and tested in the other data sets. The experiment number to select the data for train, test, and validation is the same as those previously used; however, they could be anyone. The actual data obtained in the die-sinking EDM of a Nickel and Chromium alloy using Cu-C electrodes and positive polarity have been taken from Ref. [61]. Therefore, following the previously mentioned methodology, the experimental data are divided to 70%, 15%, and 15% for train, test, and validate, respectively. The train data will be used to obtain the models. A response surface model will be fitted first by using RSM and, secondly, soft computing tools (ANN and an ANFIS using the proposed method in Section 4.4 of this present study) will be used.

The use of a methodology based on RSM allows modeling of the arithmetic average roughness of the roughness profile (Ra), but this is not the case of the EW, as shown in Ref. [61]. From the determination coefficient of the regression shown in Ref. [61], and from the classification that is followed in this present study, Ra (μm) could be classified as a type 1 function and EW (%) could be classified as a type 2 function, which can also be observed from Figure 1. As is well-known (Ra), is one of the most commonly employed parameters in industry and it is defined from the UNE-EN-ISO 4287:1999 norm [69] as the arithmetic average roughness of the roughness profile, which is shown in Figure 31, where the EDM equipment is also shown; and the EW is defined as the relation between the volume of material that is removed from the electrode and that removed from the part, which is usually expressed in %.

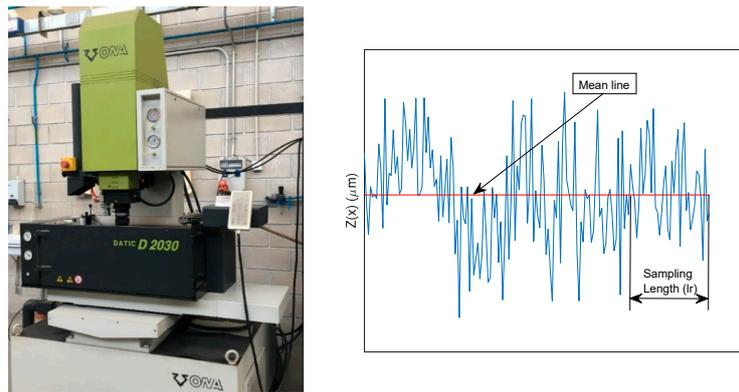


Figure 31. EDM machine ONA Datic D-2030-S, Roughness profile $Z(x)$ (μm) for determination of R_a parameter and Electrode Wear.

Figure 32 show a comparison between the results obtained, with the three methodologies analyzed in this present study (RSM, ANN, and ANFIS) and the actual results, for the case of R_a (μm), which as previously mentioned, could be considered to be a type 1 function, whose range of values is between $[1.17 \mu\text{m}, 7.41 \mu\text{m}]$. Likewise, Figure 33 shows the validation metrics obtained with an ANFIS ($k_{FIS} = 1.12, N_{FIS} = 5.21$), ANN_{3-8-1} and with the regression model for R_a values taken from Ref. [61].

On the other hand, Figure 34 show a comparison between the results obtained, with the three methodologies analyzed in this present study (RSM, ANN, and ANFIS) and the actual results, for the case of the EW (%), which could be considered to be a type 2 function, whose range of values is between $[0.3\%, 42.84\%]$.

Figure 35 shows the validation metrics obtained with the ANFIS ($k_{FIS} = 1.09, N_{FIS} = 10$), $\text{ANN}_{3-5-2-1}$ and with the regression model for EW values taken from Ref [61]. In the case of the EW, an $\text{ANN}_{3-5-2-1}$ was selected because the $\text{ANN}_{3-4-2-1}$ led to worse results.

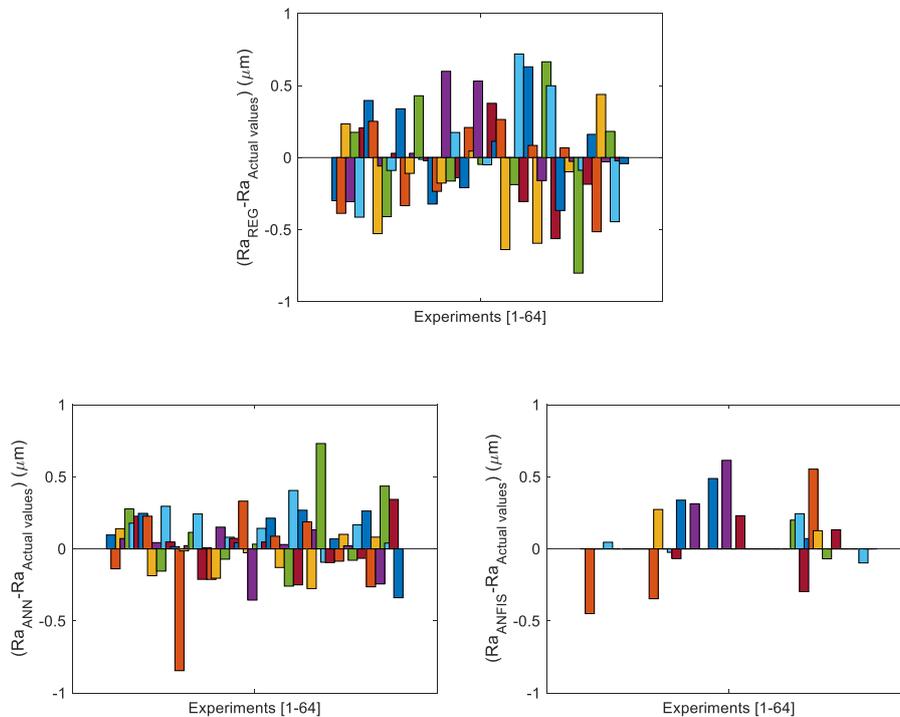


Figure 32. Difference between the actual values and those provided by the (Regression, ANN_{3-8-1} and ANFIS) models for modeling R_a (μm) (Type 1 function). The data for obtaining these graphs have been taken from Ref. [61].

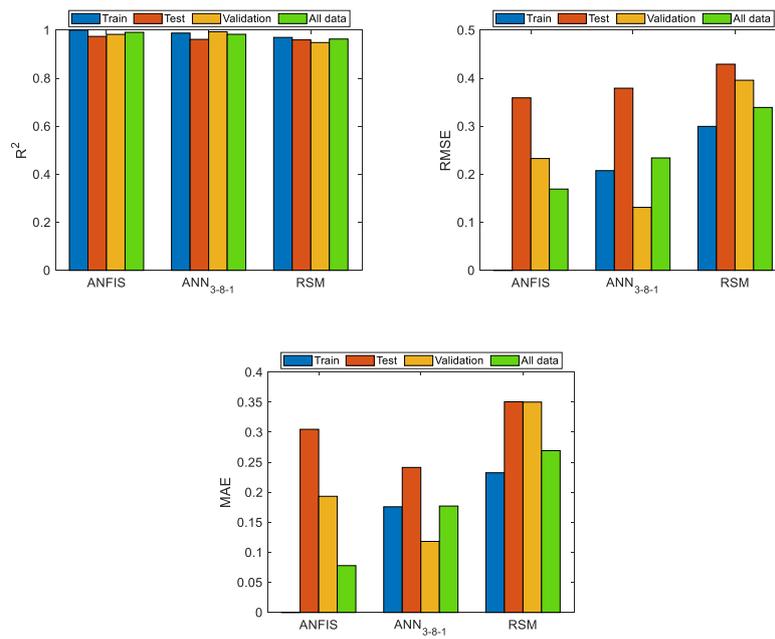


Figure 33. Validation metrics for the (ANFIS, ANN₃₋₈₋₁, Regression) models. Type 1 function (Ra). The data for obtaining these graphs have been taken from Ref. [61]).

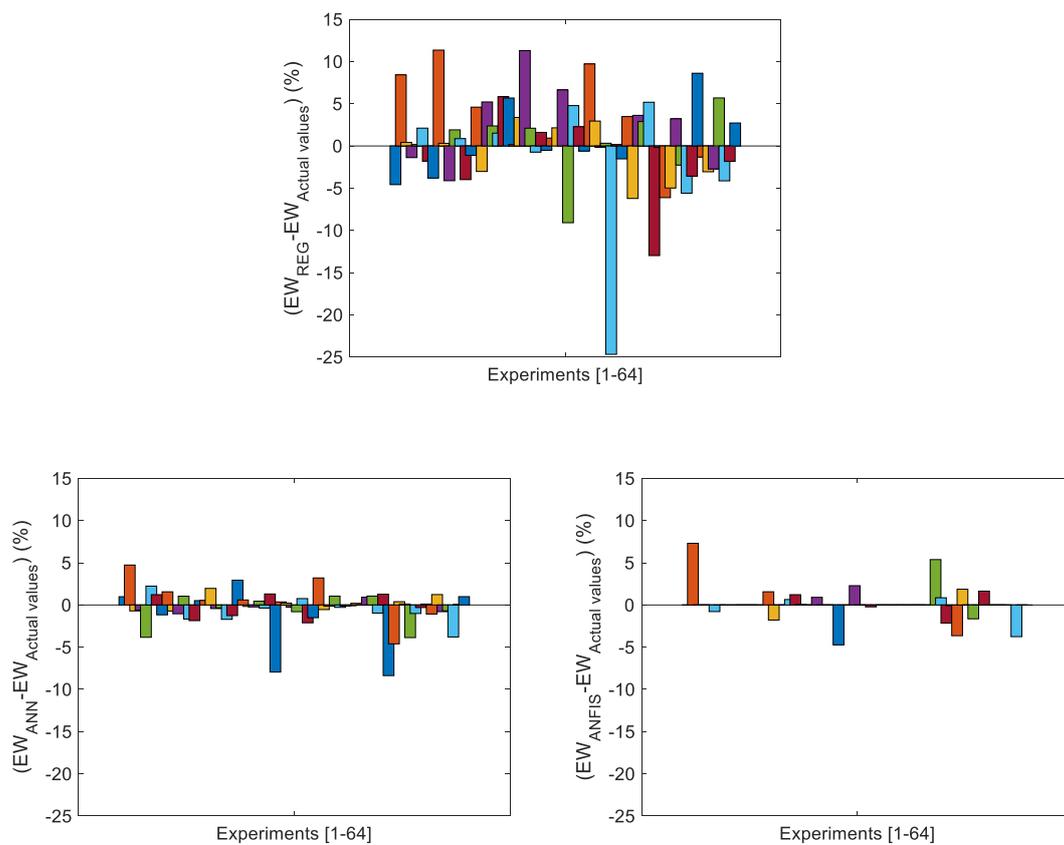


Figure 34. Difference between the actual values and those provided by the (Regression, ANN₃₋₅₋₂₋₁, and ANFIS) models for modeling EW (%) (Type 2 function). The data for obtaining these graphs have been taken from Ref. [61]).

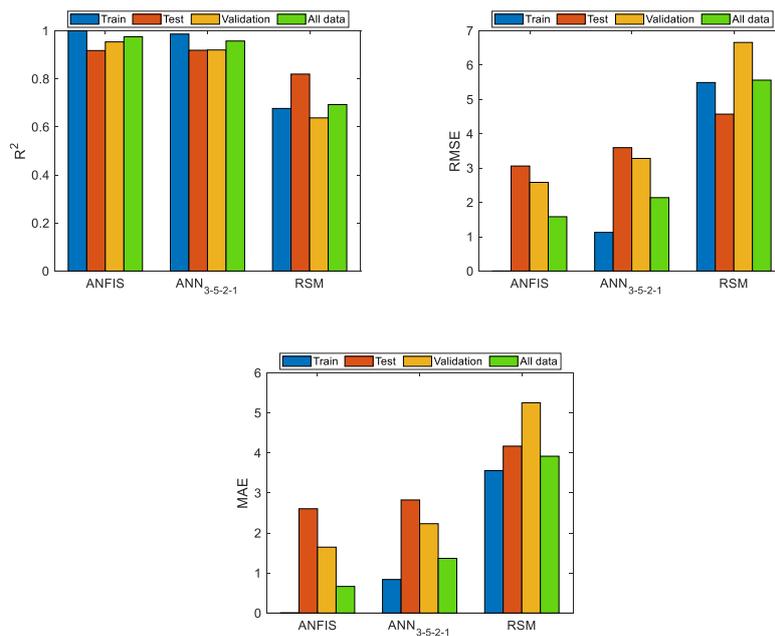


Figure 35. Validation metrics for the (ANFIS, ANN₃₋₅₋₂₋₁, Regression) models. Type 2 function (EW). The data for obtaining these graphs have been taken from Ref. [61]).

Figures 32–35 show the results obtained with the models fitted with the train data, but when these models are applied not only to the validation data but to all the experiments. As can be observed, both the ANFIS obtained with the proposed methodology and the ANNs present significant advantages over the regression models when there exists great variability in the data. Therefore, if there is a type 2 function, it is necessary to use models that allow greater precision to be obtained than that obtained with RSM. Furthermore, it can be observed that the ANFIS obtained from the proposed methodology have similar variability to the ANN and lower than the RSM when considering all experimental data.

5. Conclusions

In this present research study, two types of typologies that are usually found in the modeling of technological variables in manufacturing processes have been analyzed. These typologies have been denominated as type 1, with low variability in the response variable in relation to the independent variables, and type 2, with high variability in the response surface. A comparative study has been carried out between the precision of different models commonly used in Manufacturing Engineering, such as those based on RSM, ANN, and ANFIS, for modeling the behavior of technological variables.

It has been shown that response variables that correspond to typology 1 can be modeled by RSM. However, with modeling response variables that correspond to typology 2, it has been shown that the models obtained with conventional regression techniques based on RSM cannot adequately model the behavior of this kind of variables and therefore, in general, the results obtained are not accurate. Therefore, ANN or ANFIS-based techniques are highly effective in avoiding this problem. It has been shown that it is possible to use both neural networks and ANFIS to model the response variable with high accuracy. Both techniques allow the behavior of type 2 functions to be efficiently modeled.

In addition, in this present study a new methodology has been shown for obtaining a tuned FIS for modeling the behavior of both types of function topologies, which is based on the adjustment of the parameters of the membership functions that make up the FIS through an iterative process using an ANFIS for tuning the membership parameter of the thus generated FIS in order to minimize the MSE in the validation data. With this procedure, it has been shown that it is possible to obtain results with high precision for modeling both types 1 and 2 functions within the DOE range values.

The ANFIS obtained with the proposed methodology can provide better values of the validation metrics employed in this study in a better way than those obtained with both the RSM model and with the ANN that has 1 and 2 neurons in the hidden layer, and, the obtained results are similar to those obtained with the ANN that has 8 neurons in the hidden layer, when modeling the behavior of type 1 functions. Furthermore, when modeling type 2 functions, RSM was not able to accurately predict the behavior of the response variable. However, by using either an ANN or the methodology proposed in this present study, it was possible to accurately predict the behavior of this function.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Nguyen, A.T.; Taniguchi, T.; Eciolaza, L.; Campos, V.; Palhares, R.; Sugeno, R.M. Fuzzy Control Systems: Past, Present and Future. *IEEE Comput. Intell. Mag.* **2019**, *14*, 56–68. [[CrossRef](#)]
2. Takagi, T.; Sugeno, M. Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Trans. Syst. Man Cybern.* **1985**, *15*, 116–132. [[CrossRef](#)]
3. Nguyen, H.T.; Sugeno, M. *Fuzzy Systems, Modeling and Control*; Springer-Science+Business Media: New York, NY, USA, 1998.
4. Mamdani, E.H. Application of fuzzy algorithms for control of simple dynamic plant. *Proc. IEEE* **1974**, *121*, 1585–1588. [[CrossRef](#)]
5. Mamdani, E.H. Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis. *IEEE Trans. Comput.* **1977**, *26*, 1182–1191. [[CrossRef](#)]
6. Mouralova, K.; Hrabec, P.; Benes, L.; Otoupalik, J.; Bednar, J.; Prokes, T.; Matousek, R. Verification of Fuzzy Inference System for Cutting Speed while WEDM for the Abrasion-Resistant Steel Creusabro by Conventional Statistical Methods. *Metals* **2020**, *10*, 92. [[CrossRef](#)]
7. Aamir, M.; Tu, S.; Tolouei-Rad, M.; Giasin, K.; Vafadar, A. Optimization and Modeling of Process Parameters in Multi-Hole Simultaneous Drilling Using Taguchi Method and Fuzzy Logic Approach. *Materials* **2020**, *13*, 680. [[CrossRef](#)]
8. Cuka, B.; Kim, D.W. Fuzzy logic based tool condition monitoring for end-milling. *Robot. Comput. Integr. Manuf.* **2017**, *47*, 22–36. [[CrossRef](#)]
9. Joshi, K.K.; Behera, R.K.; Kumar, R.; Mohapatra, S.K.; Patro, S.S. Machinability Assessment of Inconel 800HT and its prediction using a hybrid fuzzy controller in EDM. *Mater. Today Proc.* **2019**, *18*, 5270–5275. [[CrossRef](#)]
10. Wang, C.-N.; Nguyen, V.T.; Chyou, J.-T.; Lin, T.-F.; Nguyen, T.N. Fuzzy Multicriteria Decision-Making Model (MCDM) for Raw Materials Supplier Selection in Plastics Industry. *Mathematics* **2019**, *7*, 981. [[CrossRef](#)]
11. Lin, Y.-C.; Wang, Y.-C.; Chen, T.-C.T.; Lin, H.-F. Evaluating the Suitability of a Smart Technology Application for Fall Detection Using a Fuzzy Collaborative Intelligence Approach. *Mathematics* **2019**, *7*, 1097. [[CrossRef](#)]
12. Luis Pérez, C.J. Using a Fuzzy Inference System to Obtain Technological Tables for Electrical Discharge Machining Processes. *Mathematics* **2020**, *8*, 922. [[CrossRef](#)]
13. Cavallaro, A. Takagi-Sugeno Fuzzy Inference System for Developing a Sustainability Index of Biomass. *Sustainability* **2015**, *7*, 12359–12371. [[CrossRef](#)]
14. Shabgard, M.R.; Badamchizadeh, M.A.; Ranjbar, G.; Amini, K. Fuzzy approach to select machining parameters in electrical discharge machining (EDM) and ultrasonic-assisted EDM processes. *J. Manuf. Syst.* **2013**, *32*, 32–39. [[CrossRef](#)]
15. Benardos, P.G.; Vosniakos, G.C. Prediction of surface roughness in CNC face milling using neural networks and Taguchi's design of experiments. *Robot. Comput. Integr. Manuf.* **2002**, *18*, 343–354. [[CrossRef](#)]
16. Devarasiddappa, D.; George, J.; Chandrasekaran, M.; Teyi, N. Application of Artificial Intelligence Approach in Modeling Surface Quality of Aerospace Alloys in WEDM Process. *Procedia Technol.* **2016**, *25*, 1199–1208. [[CrossRef](#)]
17. Twardowski, P.; Wiciak-Pikuła, M. Prediction of Tool Wear Using Artificial Neural Networks during Turning of Hardened Steel. *Materials* **2019**, *12*, 3091. [[CrossRef](#)] [[PubMed](#)]
18. Jang, J.S.R. ANFIS: Adaptive-Network-based Fuzzy Inference Systems. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685. [[CrossRef](#)]

19. Shihabudheen, K.V.; Pillai, G.N. Recent advances in neuro-fuzzy system: A survey. *Knowl. Based Syst.* **2018**, *152*, 136–162. [[CrossRef](#)]
20. Maher, I.; Ling, L.H.; Ahmed, A.D.S.; Hamdi, M. Improve wire EDM performance at different machining parameters-ANFIS modelling. *IFAC Pap.* **2015**, *48*, 105–110. [[CrossRef](#)]
21. Çaydaş, U.; Hasçalık, A.; Ekici, S. An adaptive neuro-fuzzy inference system (ANFIS) model for wire-EDM. *Expert Syst. Appl.* **2009**, *36*, 6135–6139. [[CrossRef](#)]
22. Kang, H.; Cho, H.-C.; Choi, S.-H.; Heo, I.; Kim, H.-Y.; Kim, K.S. Estimation of Heating Temperature for Fire-Damaged Concrete Structures Using Adaptive Neuro-Fuzzy Inference System. *Materials* **2019**, *12*, 3964. [[CrossRef](#)] [[PubMed](#)]
23. Al-Ghamdi, K.; Taylan, O. A comparative study on modelling material removal rate by ANFIS and polynomial methods in electrical discharge machining process. *Comput. Ind. Eng.* **2015**, *79*, 27–41. [[CrossRef](#)]
24. Suparta, W.; Alhasa, K.M. A Comparison of ANFIS and MLP Models for the Prediction of Precipitable Water Vapor. In Proceedings of the 2013 IEEE International Conference on Space Science and Communication (IconSpace), Melaka, Malaysia, 1–3 July 2013.
25. Yilmaz, I.; Kaynar, O. Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils. *Expert Syst. Appl.* **2011**, *38*, 5958–5966. [[CrossRef](#)]
26. Postorino, M.N.; Versaci, M. A Geometric Fuzzy-Based Approach for Airport Clustering. *Adv. Fuzzy Syst.* **2014**, 1–12. [[CrossRef](#)]
27. Li, P.-Z.; Zhang, D.-F.; Hu, J.-Y.; Lennox, B.; Arvin, F. Hysteresis Modelling and Feedforward Control of Piezoelectric Actuator Based on Simplified Interval Type-2 Fuzzy System. *Sensors* **2020**, *20*, 2587. [[CrossRef](#)] [[PubMed](#)]
28. Cheng, L.; Liu, W.; Hou, Z.G.; Huang, T.; Yu, J.; Tan, M. An Adaptive Takagi–Sugeno Fuzzy Model-Based Predictive Controller for Piezoelectric Actuators. *IEEE Trans. Ind. Electron.* **2017**, *64*, 3048–3058. [[CrossRef](#)]
29. Goswamia, R.; Joshib, D. Performance Review of Fuzzy Logic Based Controllers Employed in Brushless DC Motor. *Procedia Comput. Sci.* **2018**, *132*, 623–631. [[CrossRef](#)]
30. Bagua, H.; Guemana, M.; Hafaifa, A. Gas Turbine Monitoring using Fuzzy Control approaches: Comparison between Fuzzy Type 1 and 2. In Proceedings of the 2018 International Conference on Applied Smart Systems (ICASS'2018), IEEE, Médéa, Algeria, 24–25 November 2018.
31. Versaci, M. Fuzzy approach and Eddy currents NDT/NDE devices in industrial applications. *Electron. Lett.* **2016**, *52*, 943–945. [[CrossRef](#)]
32. Versaci, M.; Calcagno, S.; Cacciola, M.; Morabito, F.C.; Palamara, I.; Pellicanò, D. Chapter 7: Innovative Fuzzy Techniques for Characterizing Defects in Ultrasonic Nondestructive Evaluation. In *Ultrasonic Nondestructive Evaluation Systems, Industrial Application Issues*; Burrascano, P., Callegari, S., Montisci, A., Ricci, M., Versaci, M., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 200–232.
33. Sun, X.; Jia, X. A Fault Diagnosis Method of Industrial Robot Rolling Bearing Based on Data Driven and Random Intuitive Fuzzy Decision. *IEEE Access* **2019**, *7*, 148764–148770. [[CrossRef](#)]
34. Liu, X.; Gao, Z.; Chen, M.Z. Takagi–Sugeno Fuzzy Model Based Fault Estimation and Signal Compensation with Application to Wind Turbines. *IEEE Trans. Ind. Electron.* **2017**, *64*, 5678–5689. [[CrossRef](#)]
35. Lin, Y.Y.; Chang, J.Y.; Lin, C.T. A TSK-Type-Based Self-Evolving Compensatory Interval Type-2 Fuzzy Neural Network (TSCIT2FNN) and Its Applications. *IEEE Trans. Ind. Electron.* **2014**, *61*, 447–459. [[CrossRef](#)]
36. Biglarbegian, M.; Melek, W.W.; Mendel, J.M. Design of Novel Interval Type-2 Fuzzy Controllers for Modular and Reconfigurable Robots: Theory and Experiments. *IEEE Trans. Ind. Electron.* **2011**, *58*, 1371–1384. [[CrossRef](#)]
37. Dereli, T.; Baykasoglu, A.; Altun, K.; Durmusoglu, A.; Burhan, T. Industrial applications of type-2 fuzzy sets and systems—A concise review. *Comput. Ind.* **2011**, *62*, 125–137. [[CrossRef](#)]
38. Lei, Y.; Yang, B.; Jiang, X.; Jia, F.; Li, N.; Nandi, A.K. Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mech. Syst. Signal Process.* **2020**, *138*, 106587. [[CrossRef](#)]
39. Chen, C.; Vachtsevanos, G. Bearing condition prediction considering uncertainty: An interval type-2 fuzzy neural network approach. *Robot. Comput. Integr. Manuf.* **2012**, *28*, 509–516. [[CrossRef](#)]
40. Tayyab, M.; Sarkar, B.; Yahya, B.N. Imperfect Multi-Stage Lean Manufacturing System with Rework under Fuzzy Demand. *Mathematics* **2019**, *7*, 13. [[CrossRef](#)]
41. Faisal, N.; Kumar, K. Optimization of Machine Process Parameters in EDM for EN 31 Using Evolutionary Optimization Techniques. *Technologies* **2018**, *6*, 54. [[CrossRef](#)]

42. Alarifi, I.M.; Nguyen, H.M.; Naderi Bakhtiyari, A.; Asadi, A. Feasibility of ANFIS-PSO and ANFIS-GA Models in Predicting Thermophysical Properties of Al₂O₃-MWCNT/Oil Hybrid Nanofluid. *Materials* **2019**, *12*, 3628. [[CrossRef](#)]
43. Sibalija, T.V. Particle swarm optimisation in designing parameters of manufacturing processes: A review (2008–2018). *Appl. Soft Comput.* **2019**, *84*, 105743. [[CrossRef](#)]
44. Alajmi, M.S.; Almeshal, A.M. Prediction and Optimization of Surface Roughness in a Turning Process Using the ANFIS-QPSO Method. *Materials* **2020**, *13*, 2986. [[CrossRef](#)]
45. Capizzi, G.; Lo Sciuto, G.; Napoli, C.; Woźniak, M.; Susi, G. A spiking neural network-based long-term prediction system for biogas production. *Neural Netw.* **2020**, *129*, 271–279. [[CrossRef](#)] [[PubMed](#)]
46. Ding, L.; Matthews, J. A contemporary study into the application of neural network techniques employed to automate CAD/CAM integration for die manufacture. *Comput. Ind. Eng.* **2009**, *57*, 1457–1471. [[CrossRef](#)]
47. Ståhl, N.; Mathiason, G.; Falkman, G.; Karlsson, A. Using recurrent neural networks with attention for detecting problematic slab shapes in steel rolling. *Appl. Math. Model.* **2019**, *70*, 365–377. [[CrossRef](#)]
48. Jamli, M.R.; Farid, N.M. The sustainability of neural network applications within finite element analysis in sheet metal forming: A review. *Measurement* **2019**, *138*, 446–460. [[CrossRef](#)]
49. Hu, L.; Zhang, J.; Xiang, Y.; Wang, W. Neural Networks-Based Aerodynamic Data Modeling: A Comprehensive Review. *IEEE Access* **2020**, *8*, 90805–90823. [[CrossRef](#)]
50. Połap, D.; Winnicka, A.; Serwata, K.; Keşik, K.; Woźniak, M. An Intelligent System for Monitoring Skin Diseases. *Sensors* **2018**, *18*, 2552. [[CrossRef](#)]
51. Woźniak, M.; Połap, D. Intelligent Home Systems for Ubiquitous User Support by Using Neural Networks and Rule-Based Approach. *IEEE Trans. Ind. Inform.* **2020**, *16*, 2651–2658. [[CrossRef](#)]
52. Zielonka, A.; Sikora, A.; Woźniak, M.; Wei, W.; Ke, Q.; Bai, Z. Intelligent Internet-of-Things system for smart home optimal convection. *Trans. Ind. Inform.* **2015**, *14*, 1–11.
53. Mawson, V.J.; Hughes, B.R. Deep learning techniques for energy forecasting and condition monitoring in the manufacturing sector. *Energy Build.* **2020**, *217*, 109966. [[CrossRef](#)]
54. Airao, J.; Chaudhary, B.; Bajpai, V.; Khanna, N. An Experimental Study of Surface Roughness Variation in End Milling of Super Duplex 2507 Stainless Steel. *Mater. Today Proc.* **2018**, *5*, 3682–3689. [[CrossRef](#)]
55. Kasdekara, D.K.; Parashar, V.; Arya, C. Artificial neural network models for the prediction of MRR in Electro-chemical machining. *Mater. Today Proc.* **2018**, *5*, 772–779. [[CrossRef](#)]
56. Ahmed, N.; Rafaqat, M.; Ishfaq, K.; Ur Rehman, A.; Hassan, A.; Umer, U.; Ragab, A.E.; Al-Zabidi, A. Comparison of Laser Milling Performance against Difficult-To-Cut Alloys: Parametric Significance, Modeling and Optimization for Targeted Material Removal. *Materials* **2019**, *12*, 1674. [[CrossRef](#)] [[PubMed](#)]
57. Aslantas, K.; Danish, M.; Hasçelik, A.; Mia, M.; Gupta, M.; Ginta, T.; Ijaz, H. Investigations on Surface Roughness and Tool Wear Characteristics in Micro-Turning of Ti-6Al-4V Alloy. *Materials* **2020**, *13*, 2998. [[CrossRef](#)] [[PubMed](#)]
58. Su, Y.; Zhao, G.; Zhao, Y.; Meng, J.; Li, C. Multi-Objective Optimization of Cutting Parameters in Turning AISI 304 Austenitic Stainless Steel. *Metals* **2020**, *10*, 217. [[CrossRef](#)]
59. Zajac, J.; Duplak, J.; Duplakova, D.; Cizmar, P.; Olexa, I.; Bittner, A. Prediction of Cutting Material Durability by $T = f(v_c)$ Dependence for Turning Processes. *Processes* **2020**, *8*, 789. [[CrossRef](#)]
60. Torres, A.; Luis, C.J.; Puertas, I. EDM machinability and surface roughness analysis of TiB₂ using copper electrodes. *J. Alloy. Compd.* **2017**, *690*, 337–347. [[CrossRef](#)]
61. Torres Salcedo, A.; Puertas Arbizu, I.; Luis Pérez, C.J. Analytical Modelling of Energy Density and Optimization of the EDM Machining Parameters of Inconel 600. *Metals* **2017**, *7*, 166. [[CrossRef](#)]
62. Lin, Y.-C.; Wu, K.-D.; Shih, W.-C.; Hsu, P.-K.; Hung, J.-P. Prediction of Surface Roughness Based on Cutting Parameters and Machining Vibration in End Milling Using Regression Method and Artificial Neural Network. *Appl. Sci.* **2020**, *10*, 3941. [[CrossRef](#)]
63. Karloopia, J.; Mozammil, S.; Jha, P.K. Machinability, Modelling and Statistical Analysis of In-Situ Al-Si-TiB₂ Composites. *J. Compos. Sci.* **2019**, *3*, 28. [[CrossRef](#)]
64. Beale, M.H.; Hagan, M.T.; Demuth, H.B. *Deep Learning Toolbox™ User's Guide, R2109b, The MathWorks: Copyright 1992–2019; The MathWorks Inc.: Natick, MA, USA, 2019.*
65. Nassi, I.; Shneiderman, B. Flowchart techniques for structured programming. *ACM Sigplan Not.* **1973**, *8*, 12–26. [[CrossRef](#)]

66. The MathWorks Inc. *Fuzzy Logic Toolbox™ User's Guide* © Copyright 1995–2019; The MathWorks Inc.: Natick, MA, USA, 2019.
67. Versaci, M.; Calcagno, S.; Cacciola, M.; Morabito, F.C.; Palamara, I.; Pellicanò, D. Chapter 6: Standard Soft Computing Techniques for Characterization of Defects in Nondestructive Evaluation. In *Ultrasonic Nondestructive Evaluation Systems, Industrial Application Issues*; Burrascano, P., Callegari, S., Montisci, A., Ricci, M., Versaci, M., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 175–199.
68. Egaji, O.A.; Griffiths, A.; Hasan, M.S.; Yu, H.N. A comparison of Mamdani and Sugeno fuzzy based packet scheduler for MANET with a realistic wireless propagation model. *Int. J. Autom. Comput.* **2015**, *12*, 1–13. [[CrossRef](#)]
69. UNE-EN ISO 4287:1999. *Geometrical Product Specifications (GPS)-Surface Texture: Profile Method-Terms, Definitions and Surface Texture Parameters*; AENOR: Madrid, Spain, 1999.



© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).