Journal of Theoretical and Applied Information Technology
15th December 2020. Vol.98. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

STATISTICAL COMPARISON BETWEEN EL-MLP AND EL-ANFIS, OPTIMIZED BY MEANS OF ANOVA, FOR THE PD CONTROL OF A MOBILE ROBOT

[1]DANTE GIOVANNI STERPIN, [1]JESÚS DAVID MARTÍNEZ VELANDIA, [2]FERNANDO MARTÍNEZ SANTA

[1]Electronic Engineering, Corporación Unificada Nacional de Educación Superior, Bogotá, Colombia

[2]Electric Engineering, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia

E-mail: [1]{dante_sterpin, jesus.martinez}@cun.edu.co, [2]fmartinezs@udistrital.edu.co

ABSTRACT

In this paper, two types of controller for a mobile robot with the wall-following task are statistically compared. One of them is a Multi-Layer Perceptron (MLP) while the other one is an Adaptive-Network-based Fuzzy Inference System (ANFIS). Here, such controllers are named: EL-MLP and EL-ANFIS, because they were trained by means of an analytical method known as Extreme Learning Machine (ELM). They were structurally optimized with a statistical method known as Analysis of Variance (ANOVA), and a t-Test between two populations with the best exemplars of each type of controller, demonstrates statistically that EL-ANFIS generalizes better than EL-MLP, due to its validation error mean, and variance, are significantly lower.

Keywords: *Multi-Layer Perceptron (MLP), Adaptive-Network-based Fuzzy Inference System (ANFIS), Extreme Learning Machine (ELM), Analysis of Variance (ANOVA), Hypothesis test between two populations (t-Test), PD Control.*

1. INTRODUCTION

The artificial neural networks and the Fuzzy inference systems have been widely used in engineering in several applications including mobile robots control. In specific applications of the Multi-Layer Perceptron (MLP) [1], the structural parameters of the neural network, as the number of hidden layers and the number of nodes in each of them are determined by the designer, applying previously known configurations. Due to the absence of a deterministic method to set up such parameters, different optimization techniques have been proposed [2, 3, 4, 5] in order to minimize such parameters using bioinspired algorithms such as ant colonies and genetic algorithms [6, 7, 8]. Among those techniques, the analysis of variance (ANOVA) a tool for the design and analysis of experiments [9, 10], was preferred in this work because it was recently considered as a statistically measurable method, in the optimization of neural networks [5]. Something similar happens to the Adaptive-Network-based Fuzzy Inference Systems (ANFIS), in order to set up the number of terms for fuzzification [11, 12, 13, 14, 15]. In this regard, optimization methods also have been proposed, some of which are based on genetic algorithms [16, 17, 18], particle swarm [19, 20, 21, 22], or c-means clustering [23], but the use of ANOVA to optimize the structure of ANFIS has not been realized yet.

Nowadays, there are performance comparisons between MLP and ANFIS, in control applications [24, 25], artificial vision [26, 27], and predictive modeling in diverse areas [28, 29, 30, 31, 32, 33, 22, 34, 35]. In this way, the absence of comparisons like these in control applications for mobile robots is very noticeable.

On the other hand, the Extreme Learning Machine (ELM) [36] has been laid down as a training method for Single hidden Layer Feedforward Networks (SLFN), including the MLP [37] and ANFIS [38, 39], since, in contrast to the backpropagation (BP) based training, the ELM method offers the advantages of having a very short learning time, and SLFN with great capability to generalize are obtained.

The main objective of the work reported in this paper is to do a statistical performance comparison between an MLP and an ANFIS, working both as controllers. Here, those systems were trained using ELM, and optimized by means of ANOVA. As a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

reference case, a differential mobile robot performing the wall-following navigation task was considered. The specific contribution of this paper is to show that ANFIS is a better generalizer than MLP for the control of a mobile robot, using ELM as a training technique which guarantees a great generalization for all SLFN, and additionally using ANOVA as a statistical optimization technique (previously used for the MLP but not for ANFIS). Thus, the novelty in this work is the application of ANOVA to optimize the architecture of an ANFIS, and the statistical comparison between MLP and ANFIS, as controllers in the context of mobile robotics. The justification for this is that ANOVA is a statistically measurable method that had not been applied before for ANFIS, and also there were not comparisons like this in mobile robotics.

In this paper, first, the used architectures of both MLP and ANFIS are detailed, in conjunction with their ELM based training. Then, the ANOVA based optimization of the MLP and a proposal for its application to the ANFIS are detailed. The method and the results of the statistic comparison between two populations, composed by certain number of optimized controllers of each type, are detailed in the last section.

2. METHODOLOGY

Both controllers, the ELM based MLP and the ELM based ANFIS, were implemented by means of computer simulations without using any real or virtual robot. Those simulations were implemented on Java programming language, for training and testing the controllers several times. After that, they were optimized by means of the analysis of variance (ANOVA). Finally, to obtain data of optimized controllers, for the statistical comparison, new simulations were done. It is important to clarify that the implemented simulations only give the outputs of the controllers, i.e. the speed level of each motor, but do not the position or speed of the whole robot.

Based on previous comparisons between MLP and ANFIS, in various contexts, ANFIS has been found to be more efficient and robust than MLP, therefore, the hypothesis of investigation was that ANFIS, as a robotic controller, could be a better generalizer than MLP.

3. EXTREME LEARNING MLP (EL-MLP)

The Multi-Layer Perceptron is the most representative connectionist system, and it is built by layers of artificial neurons. Figure 1 shows the MLP as an SLFN, so it can be trained by ELM. Its inputs receive the proportional error signal (P) and the derivative error signal (D), which are characteristic of PD controllers. Since its output gives the speed level of one motor, two MLP like that are used, and each one receives the same input signals. Therefore a "double" MLP controls the robot, and each MLP-controller has independent synaptic weights from each other.

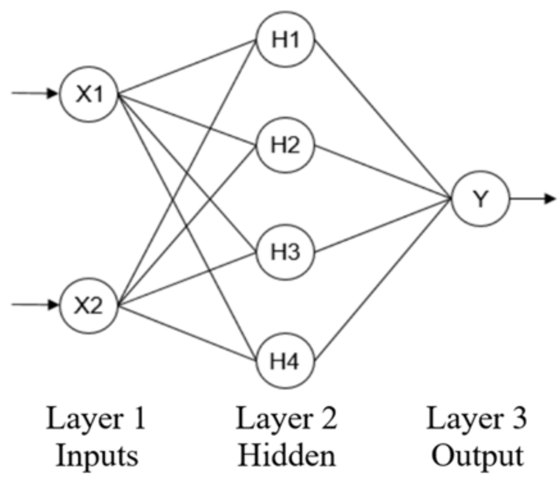


Figure 1: Structure used for each MLP-control

$$\mathbf{H} = \begin{bmatrix} F_1(\mathbf{w}_1 \cdot \mathbf{x}_1 - \theta_1) & F_2(\mathbf{w}_2 \cdot \mathbf{x}_1 - \theta_2) & F_3(\mathbf{w}_3 \cdot \mathbf{x}_1 - \theta_3) & \ddots & F_\phi(\mathbf{w}_\phi \cdot \mathbf{x}_1 - \theta_\phi) \\ F_1(\mathbf{w}_1 \cdot \mathbf{x}_2 - \theta_1) & F_2(\mathbf{w}_2 \cdot \mathbf{x}_2 - \theta_2) & F_2(\mathbf{w}_3 \cdot \mathbf{x}_2 - \theta_3) & \ddots & F_\phi(\mathbf{w}_\phi \cdot \mathbf{x}_2 - \theta_\phi) \\ F_1(\mathbf{w}_1 \cdot \mathbf{x}_3 - \theta_1) & F_2(\mathbf{w}_2 \cdot \mathbf{x}_3 - \theta_2) & F_3(\mathbf{w}_3 \cdot \mathbf{x}_3 - \theta_3) & \ddots & F_\phi(\mathbf{w}_\phi \cdot \mathbf{x}_3 - \theta_\phi) \\ \vdots & \vdots & \vdots & & \vdots \\ F_1(\mathbf{w}_1 \cdot \mathbf{x}_\pi - \theta_1) & F_2(\mathbf{w}_2 \cdot \mathbf{x}_\pi - \theta_2) & F_3(\mathbf{w}_3 \cdot \mathbf{x}_\pi - \theta_3) & \ddots & F_\phi(\mathbf{w}_\phi \cdot \mathbf{x}_\pi - \theta_\phi) \end{bmatrix}_{\Pi \times \Phi} \quad (1)$$

In order to apply the ELM technique, in each MLP-based controller, the synaptic weights between the Φ hidden neurons and the input neurons are considered as certain Alpha parameters and they are randomly initialized. After that, the response F_j of each hidden neuron is obtained, using the bipolar sigmoid as activation function, to

Journal of Theoretical and Applied Information Technology
15th December 2020. Vol.98. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

obtain the **H** matrix of the equation 1, given the Π examples of training set.

Vector **Y** contains the Π obtained outputs and it is calculated by the equation 2, using the **H** matrix and certain *Beta* parameters. In the MLP those *Betas* are the synaptic weights between the Φ hidden neurons and the output neuron. Being that the correspondence between **Y** and the desired outputs vector **T** is expected, according to the Π training examples, the equation 3 shows how such *Betas* are obtained. There, $\mathbf{H}^{\dagger}$ is the Moore-Penrose pseudoinverse of **H** and to calculate it the equation 4 is used, in which **I** is the identity matrix, and □ should tend to zero [40]. Thus, the resulting solution is more stable and has better generalization [41]. In this work □ is $1E-8$.

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y} = [y_1 \;\; y_2 \;\; y_3 \;\; \ldots \;\; y_\pi]^T \tag{2}$$

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{3}$$

$$\mathbf{H}^{\dagger} = \left(\lambda\mathbf{I} + \mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T \tag{4}$$

4. EXTREME LEARNING ANFIS (EL-ANFIS)

ANFIS is an Adaptive-Network-based Fuzzy Inference System, originally proposed in [42]. Figure 2 shows that its structure is composed by layers of nodes without connection values. ANFIS allows robots to obtain certain behavior, according to the inputs, using supervised learning like MLP. It is usually trained by a hybrid way, using the BP algorithm for certain *Alpha* parameters in the antecedent nodes, and using the Least Squares Method (LSM) for certain *Beta* parameters in the consequent nodes. In order to control the robot motors, ANFIS also process the PD control signals, and uses two structures like figure 2, giving the same input signals to each one. Thus a "double" ANFIS controls the robot, and each ANFIS-controller has independent parameters from each other.

In the 1st layer of each ANFIS the inputs are fuzzified by means of generalized bell function, as membership function, according to equation 5. Each input x_k is fuzzified using P membership functions, or fuzzy terms. In each of those terms, $\{a_\rho;\ b_\rho;\ c_\rho\}$ are the *Alfa* parameters. In order to apply the ELM technique, in each ANFIS-based controller, those *Alphas* are initialized with the proposed method in [38].

$$\mu_\rho(x_\kappa) = \frac{1}{1 + \left[\left(\frac{x_\kappa - c_\rho}{a_\rho}\right)^2\right]^{b_\rho}} \tag{5}$$

In the 2nd layer the stimulation level R_i of fuzzy rules is calculated, by means of the fuzzy conjunction among some fuzzifications μ_ρ, just like the equation 6 indicates. Figure 2 shows the [2×2] conventional conjunctions, assuming two fuzzy terms by each input x_k. In the 3rd layer those stimulation levels R_i are normalized, according to the equation 7, assuming that it could have as many as Ψ fuzzy rules.

$$R_i = AND\left[\mu_\rho^i(x_1) \;\ldots\; \mu_\rho^i(x_\kappa)\right] \tag{6}$$

$$N_i = \frac{R_i}{\sum_{j=1}^{\psi} R_j} \tag{7}$$

In the 4th layer the Sugeno-type response is calculated in each consequent node C_i, using the *Beta* parameters, which are their $\boldsymbol{m}_i$ coefficients, and their b_i constant, according to the equation 8. Figure 2 shows that the single hidden layer in ANFIS like a SLFN, is its 4th layer. The final output is obtained adding the Ψ partial outputs C_i, using the equation 9.

$$C_i = N_i\left(\mathbf{m}_i \cdot \mathbf{x} + b_i\right) \tag{8}$$

$$y = \sum_{i=1}^{\psi} C_i \tag{9}$$

Since the ELM must find the m_i coefficients, and the b_i constant, for each C_i node, the equation 10 details the H matrix of ANFIS. Unlike the H matrix of MLP, its number of columns is calculated multiplying the number Ψ of fuzzy rules by $(K+1)$, assuming K inputs.

$$\mathbf{H} = \begin{bmatrix} (x_1N_1)_1 & (x_2N_1)_1 \ldots (x_\kappa N_1)_1 & (N_1)_1 & \ddots & (x_1N_\psi)_1 & (x_2N_\psi)_1 \ldots (x_\kappa N_\psi)_1 & (N_\psi)_1 \\ (x_1N_1)_2 & (x_2N_1)_2 \ldots (x_\kappa N_1)_2 & (N_1)_2 & \ddots & (x_1N_\psi)_2 & (x_2N_\psi)_2 \ldots (x_\kappa N_\psi)_2 & (N_\psi)_2 \\ (x_1N_1)_3 & (x_2N_1)_3 \ldots (x_\kappa N_1)_3 & (N_1)_3 & \ddots & (x_1N_\psi)_3 & (x_2N_\psi)_3 \ldots (x_\kappa N_\psi)_3 & (N_\psi)_3 \\ \vdots & \vdots \quad \vdots & \vdots & & \vdots & \vdots \quad \vdots & \vdots \\ (x_1N_1)_\pi & (x_2N_1)_\pi \ldots (x_\kappa N_1)_\pi & (N_1)_\pi & \ddots & (x_1N_\psi)_\pi & (x_2N_\psi)_\pi \ldots (x_\kappa N_\psi)_\pi & (N_\psi)_\pi \end{bmatrix}_{\Pi \times (K+1)\Psi} \quad (10)$$

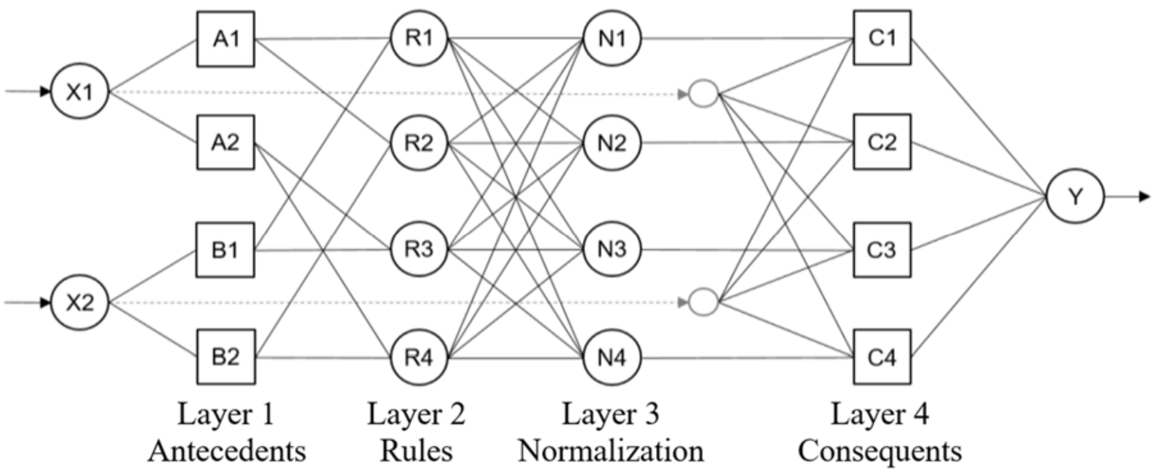


Figure 2: Structure used for each ANFIS-controller

Finally, the *Beta* parameters of ANFIS are obtained using the equation 3, and this calculation is repeated by 50 times [36], both for ANFIS and MLP. Each time the Π training examples were changed, choosing randomly the 80% of the training set. The remaining 20% was used to find the best configuration of *Betas*, choosing which show the least validation error.

5. EL-MLP AND EL-ANFIS OPTIMIZATION

Since we have already described how both MLP and ANFIS are trained with ELM, they are here after named: EL-MLP and EL-ANFIS. Now, with the necessity of optimizing both the number of MLP hidden neurons, and the number of ANFIS fuzzy terms, using a statistically measurable method, this section describes how ANOVA was applied for EL-MLP, according to the Sbarufatti's proposal [5], and how we propose to apply it for EL-ANFIS. For that, we made experiments with validation error as response variable, measuring it with the sum of square errors (SSE), given the 20% of the training set, in order to estimate the generalization capability of controllers.

The ANOVA method compares data populations, taking into account their sources of variability, thus it determines if those populations show significant differences, testing statistically if their population means are equivalent, or not. Generally, such differences are tested to know if any or a few experimental treatments improve the behavior of certain response variable, or dependent variable. Those treatments are either the values of certain independent variable, or the possible combinations of values of certain number of independent variables. The independent variables are often called: factors of the experiment. Assuming g applied treatments and n observations of the dependent variable, by each treatment, the ANOVA method partitions the total variability (SS_T) of the dependent variable, into the variability that is caused by the treatments (SS_G), and the variability that is caused by random (SS_R). Taking into account that (SS_G) has (g - 1) degrees of freedom, and (SS_R) has (N - g) degrees of freedom, where ($N = g * n$) is the total number of observations, their respective mean squares: (MS_G) and (MS_R) are computed, to calculate the test statistic F_o using the equation 11.

$$F_o = \frac{SS_G/(g-1)}{SS_R/(N-g)} = \frac{MS_G}{MS_R} \quad (11)$$

Since *Fo* follows a Fisher-Snedecor distribution with (g - 1) and (N - g) degrees of freedom, we can do an hypothesis testing, with $(100 * (1 - \alpha))$% of confidence, and if the respective P-value is less than the significance level α then the hypothesis of equivalent population means is rejected, and thus the test conclude that the application of any treatment has a significant effect on the dependent variable. However, this method requires that data must satisfy the randomness, normality and homogeneity assumptions [9].

5.1 Optimizing EL-MLP by means of ANOVA

Applying the ANOVA method to optimize the EL-MLP-based controllers, the respective treatments are defined by variations of the number Φ of hidden neurons. In this way, the aim is to find a minimum Φ value, such that its increase does not imply a significant improvement on the SEE, statistically speaking. Following the method proposed by Sbarufatti [5], we obtained data defining a range of possible values for the Φ independent variable: [1; 51]. Each of those 51 values sets up an experiment, as the minimum value ϕ of a sub-set with 10 levels: $[\phi; (\phi + 9)]$. Therefore, a factor with 10 treatments is applied to each of those 51 experiments, and each treatment is repeated 10 times. Aforementioned experiments design implies that each EL-MLP-based controller will be optimized after being trained $(10 * 51 * 10)$ times, but it is not a problem because ELM is used for training each time. Thus, at each experiment, ANOVA compares 10 populations, with 10 data each one, and with 95% of confidence ($\alpha = 0.05$) it determines if its SSE means are equivalent, or if any treatment has a significantly influence on the SSE mean. So, the minor ϕ for which the hypothesis of equivalent means has been accepted, and the normality and homogeneity assumptions too, it will be the minimum number of hidden neurons for the EL-MLP-based controller. In this work the normality was tested with Lilliefors [43], the homogeneity was tested with Levene [9], and the randomness is implicit in the computational manner of obtaining data.

Since original data satisfied neither normality nor homogeneity, they had to be transformed by means of the Box-Cox technique [9]. Figure 3 details the obtained P-values at each normality test, homogeneity test and ANOVA, during the optimization of the EL-MLP-controller for the left motor. There, the minor ϕ was 15 hidden neurons. Figure 4 details the same for the EL-MLP-controller for the right motor and there, the minor ϕ was 16 hidden neurons.

5.2 Optimizing EL-ANFIS by means of ANOVA

Previously we detailed that for EL-MLP the applied ANOVA is of one factor: Φ, to find the minimum number of hidden neurons, and the respective number of *Betas* is equal to the Φ value. For EL-ANFIS we propose to use K factors: $\{P_1;...; P_k\}$, in order to find the minimum number of fuzzy terms at each input x_k, however the number of *Betas* in EL-ANFIS increases $(K+1)$ times, the number of fuzzy rules added because of adding one to the value of any factor. We add by default as many fuzzy rules as the number of combinations of P of the other factors. In the factorial design of experiments, ANOVA additionally partitions the variability (SS_G) into the variabilities caused by each factor, and the variabilities caused by each interaction among factors. In the optimization of EL-ANFIS, ANOVA don't need such partition because in this application it needs to know just when there is not any influence of treatments on the SSE mean. Therefore, to optimize EL-ANFIS the test statistic F_o is calculated, and interpreted, just like in EL-MLP.

Applying that proposal to the EL-ANFIS-based PD controllers, then we have two factors: $\{P_1; P_2\}$, both varying among [2; 10]. Each possible combination of them sets up an experiment, as the minimum values: $\{\rho_1 ; \rho_2\}$, and using just three levels by factor: $[\rho_k ; (\rho_k + 2)]$, because of the increase of *Betas* when any factor increments its value. Thus, 81 experiments were performed with $[3 \times 3]$ treatments each one, and 10 repetitions by treatment. So, each EL-ANFIS controller had been trained $(81 * 9 * 10)$ times. The obtained data had to transform too, by means of the Box-Cox technique, in order to satisfy the normality and homogeneity assumptions, while the randomness is implicit in the computational manner to obtain data too. At each experiment ANOVA determines, with 95% of confidence ($\alpha = 0.05$), if any treatment has significant effect on the SSE mean, or not. The minimum values: $\{\rho_1 ; \rho_2\}$ which the hypothesis of equivalent means is accepted for, the minimum number of fuzzy terms at each input x_k. Figure 5 indicates with light gray the resulting eligible combinations, and a pair of X indicates the ones that our method finally chose because they also have the lowest SSE mean.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

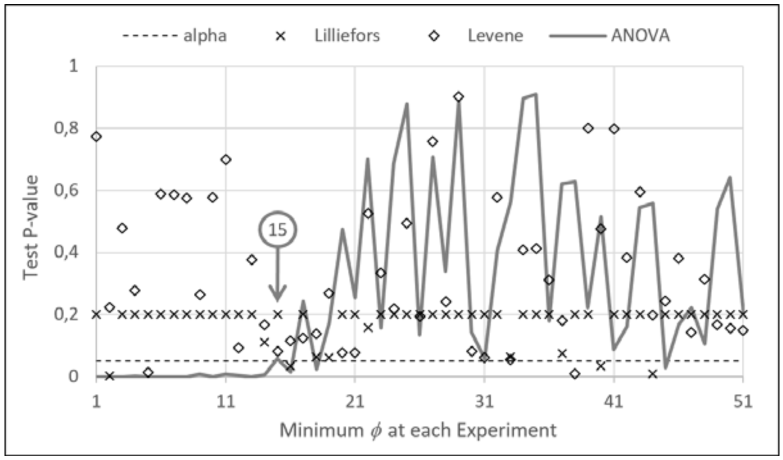


Figure 3: Normality, homogeneity and ANOVA P-values for the Left EL-MLP-controller

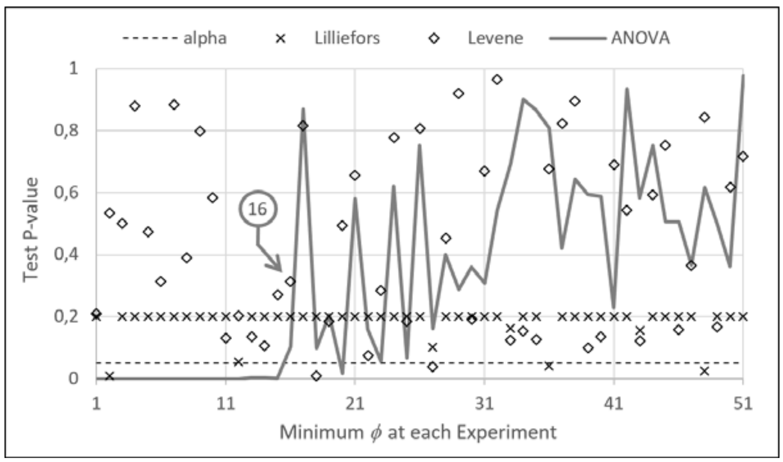
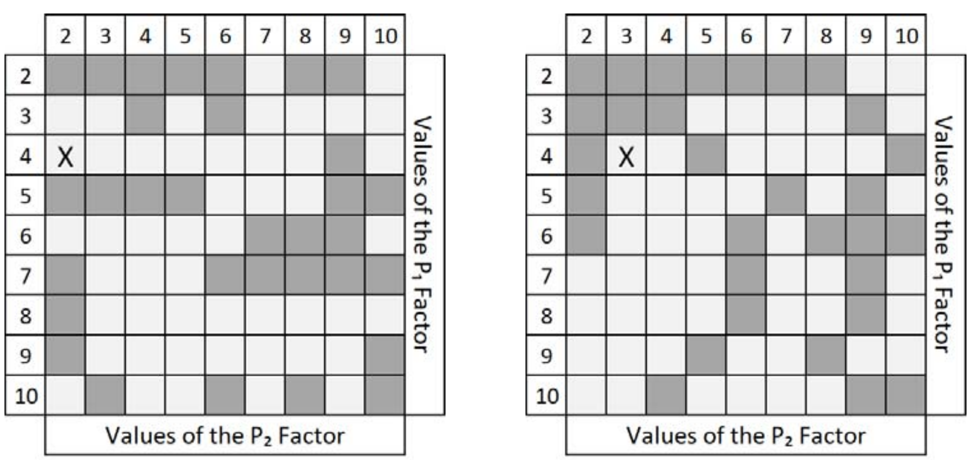


Figure 4: Normality, homogeneity and ANOVA P-values for the Right EL-MLP-controller



(a) Left EL-ANFIS-controller (b) Right EL-ANFIS-controller

Figure 5: Eligible and chosen combinations: {ρ1 ; ρ2} for each EL-ANFIS-controller

Journal of Theoretical and Applied Information Technology
15th December 2020. Vol.98. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

6. EL-MLP VERSUS EL-ANFIS COMPARISON

In the previous section, a "double" EL-MLP with 15 and 16 hidden neurons respectively, and a "double" EL-ANFIS with {4; 2} and {4; 3} fuzzy terms respectively, were presented, to control both the robot motors to perform the wall-following navigation task. Talking about the SSE mean shown by the left and right motor respectively, 0.04 and 0.02 was measured for the EL-MLP-based controllers, while 0.02 and 0.01 was measured for the EL-ANFIS-based controllers. So, at first sight, EL-ANFIS seems to show better generalization than EL-MLP, because they have less SSE mean in each robot motor. Since we need to know if that difference has statistical significance, we performed 30 optimizations for each controller type. Figures 6 and 7 detail histograms about the obtained controllers, and there we can see that the ANOVA based optimization technique gives diverse configurations for both controller types, thus the best SSE mean changes with each optimization run.

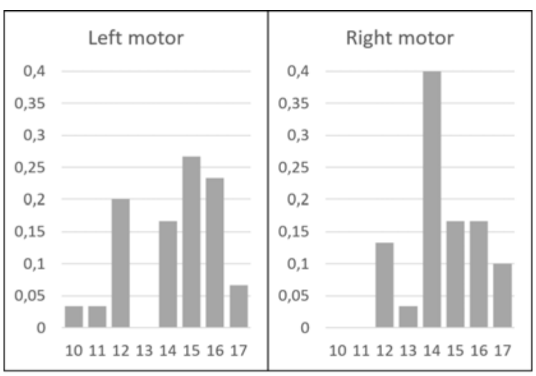


Figure 6: Distribution of Φ for optimized EL-MLP

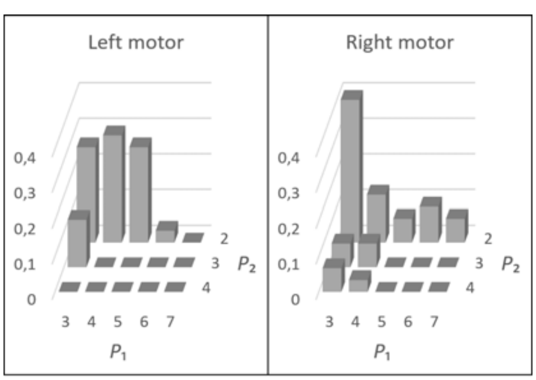


Figure 7: Distribution of {P1; P2} for optimized EL-ANFIS

In order to determine which controller type achieve the lower SSE mean, as generalization capability indicator, we performed an hypothesis testing for the means comparison between two population samples, often called: t-Test [9]. Each population has 10 samples by each of the 30 previously optimized controllers, i.e. 300 samples of each type, in order to have the typical power of 80% (β = 0.2) with a tolerance (δ = 0.005). So, it evaluated unilaterally, with 95% of confidence (α = 0.05), which controller type achieve a SSE mean significantly lower.

Table 1: Statistical comparison of SSE means

Motor	μ_1	μ_2	P-value (μ_1 - μ_2)
Left	0.04549	0.01809	5.175*E* - 28
Right	0.02440	0.00680	6.053*E* - 52

Table 2: Statistical comparison of SSE variances

Motor	σ_1	σ_2	P-value (σ_1^2 / σ_2^2)
Left	0.03652	0.01748	1.341*E* - 34
Right	0.01614	0.00584	2.836*E* - 60

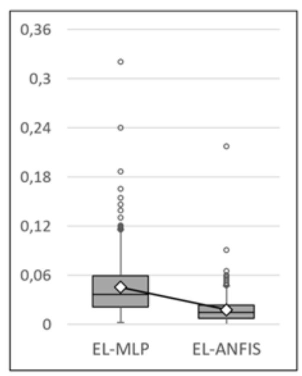


Figure 8: SSE box-plots and means of Left motor

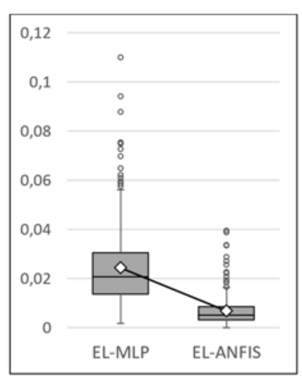
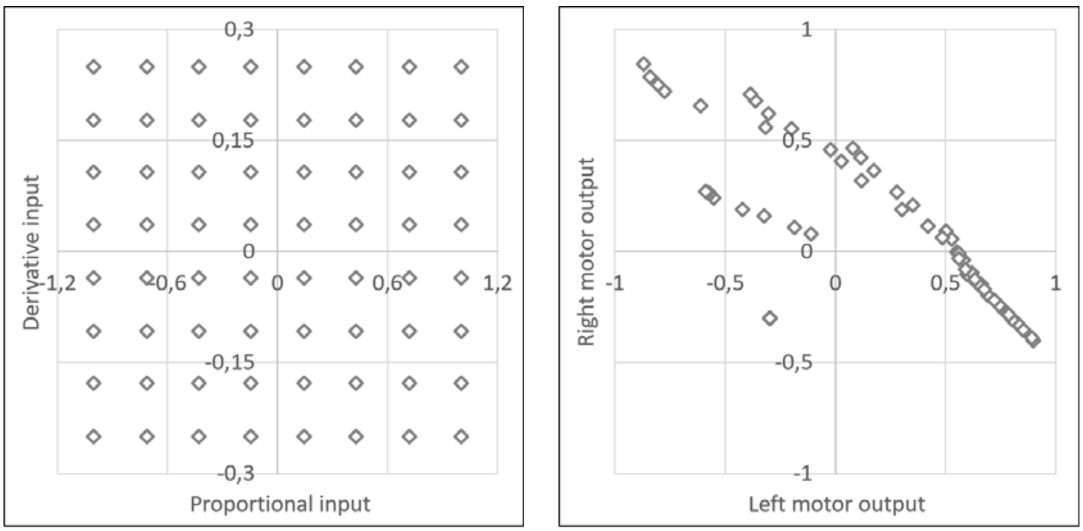


Figure 9: SSE box-plots and means of Right motor

Journal of Theoretical and Applied Information Technology
15th December 2020. Vol.98. No 23

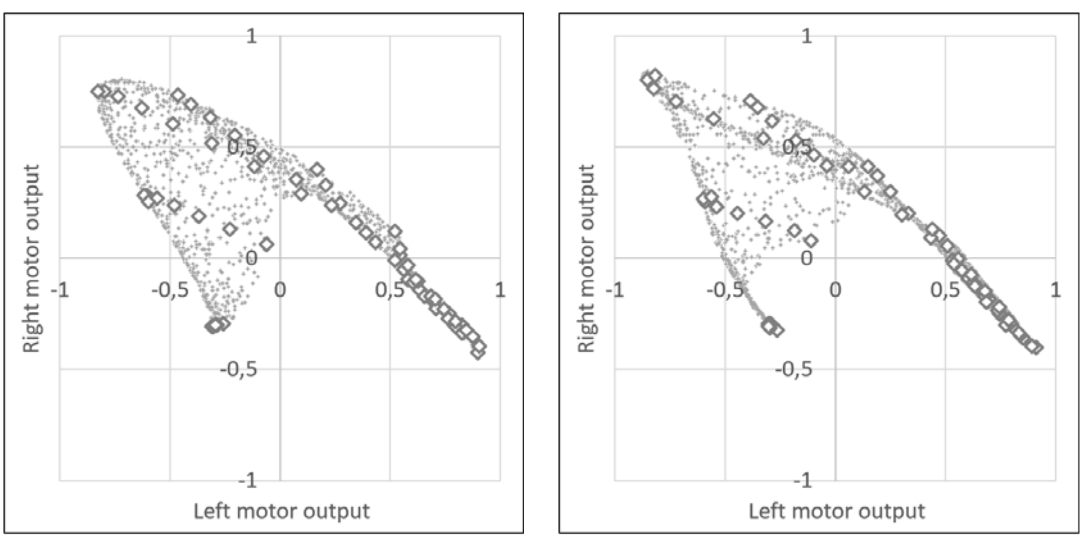
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Denoting each EL-MLP-based controller SSE mean as μ_1, and each EL-ANFIS-based controller SSE mean as μ_2, the table 1 displays its respective values, for both left and right robot motors. There, we can detail that the EL-ANFIS SSE means are less than the corresponding EL-MLP SSE means, in both motors. Furthermore, their respective standard deviations, in the table 2, and the box-plots in figures 8 and 9, show that EL-ANFIS is less dispersal than EL-MLP, in both motors. With the P-values in the table 1 we accept that SSE means are significantly different ($\mu_1 > \mu_2$) in both motors, after accept that the SSE variances are significantly different too ($\sigma_1^2 > \sigma_2^2$) in both motors, using the P-values in the table 2. So, we can assert that in this study, EL-ANFIS shows better generalization than EL-MLP, consuming about 7 ms. more in training time, maximum, because each EL-MLP was trained in a maximum of 21 ms., while each EL-ANFIS was trained in a maximum of 28 ms.



(a) Possible input situations (b) Desirable output actuations

Figure 10: Training set for both EL-MLP and EL-ANFIS "double" controllers



(a) Response of the "double" EL-MLP (b) Response of the "double" EL-ANFIS

Figure 11: Graphic example of the achieved generalization in each controller type

Finally, in order to detail a graphic example of the achieved generalization in each controller type, we considered the more frequent optimized values for Φ and {P_1; P_2}, according to figures 6 and 7. So, a "double" EL-MLP with 15 and 14 hidden neurons, in addition to a "double" EL-ANFIS with {4; 2} and {3; 2} fuzzy terms, were trained. Figure 10 details the 64 examples of training set used throughout this work. There, figure 10a shows its respective input situations and figure 10b shows its respective desirable responses.

Thus, figure 11 shows the 64 obtained responses to the situations in figure 10a, over the responses to 2000 random situations, uniformly distributed among those in figure 10a. An additional value that can be calculated here is the adjust coefficient R^2, given the desirable and obtained outputs for the validation examples. Respectively for each motor, the EL-MLP has R^2 of 0.9851 and 0.9698, and EL-ANFIS is definitely better: 0.9949 and 0.9959.

7. CONCLUSIONS

In this paper we report a statistical comparison between Multi-Layer Perceptron (MLP) and the Adaptive-Network-based Fuzzy Inference System (ANFIS), used as PD controllers for a differential mobile robot, performing the wall-following task. Here, it was verified that the Extreme Learning Machines (ELM) method trained such systems in the order of milliseconds, very much faster than with back-propagation (BP) [36], and that facilitates its structural optimization by means of the statistical technique called: Analysis of Variance (ANOVA). Aforementioned optimization method was recently proposed for neural networks trained by means of BP [5], and in this paper we show its application to Single Layer Feedforward Networks (SLFN) trained by means of ELM. Since ELM tends to give SLFN with great generalization capability [36], for this case study we have statistically demonstrated that the best EL-ANFIS-based controllers has better generalization capability than the best EL-MLP-based controllers. The quantitative evidence of it is an hypothesis testing that compared the validation error mean of 30 optimized controllers of each type. The great advantage of this work is that the optimization and comparison are performed with respect to the controllers generalization capability, and in a statistically measurable way, in contrast to the behavioral comparison shown in [25].

Considering that among the diverse comparisons between MLP and ANFIS, the last one tends to be more efficient and robust than the first one, it is important to notice that in this work EL-ANFIS clearly shows the same tendency. However, there is no reason why EL-MLP should be a priori rejected. Regarding the application of ANOVA as optimization method for EL-ANFIS, it is noticed that it is not recommended to explore values higher than seven fuzzy terms, because the fast increment of the number of *Betas* implies, as disadvantages, the increment of the training time and the increment of the on-line processing time, as a mobile robot controller. The use of such optimization method in other kind of applications could demand more data by treatment, but the use of ELM enables to obtain them quickly. In future works we could recommend, on the one hand, to consider the number of fuzzy rules as another factor in the ANOVA based optimization of EL-ANFIS. On the other hand, Radial Basis Function (RBF) neural networks could be included in the statistical comparison, training them with ELM and optimizing them with ANOVA too, just like we did with the EL-MLP here. Additionally, it is recommended to compare the performance of those controllers, using them in real robots, due to the comparison in this work was limited to estimate theoretically the generalization capability. Finally, it could investigate statistically if there is significant difference in the on-line processing time, controlling real robots with optimized examples of aforementioned systems.

REFERENCES:

[1] G. Panchal, A. Ganatra, Y.P. Kosta and D. Panchal, Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers, *International Journal of Computer Theory and Engineering* 3(2) (2011), 332–337.

[2] K.G. Sheela and S.N. Deepa, Review on methods to fix number of hidden neurons in neural networks, *Mathematical Problems in Engineering* 2013 (2013).

[3] S. Susan and M. Dwivedi, Dynamic growth of hidden-layer neurons using the non-extensive entropy, in: *2014 Fourth International Conference on Communication Systems and Network Technologies*, IEEE, 2014, pp. 491–495. ISBN ISBN 1479930709.

[4] R. Ata, Artificial neural networks applications in wind energy systems: A review, *Renewable*

and Sustainable Energy Reviews 49(534–562) (2015).

[5] C. Sbarufatti, Optimization of an artificial neural network for fatigue damage identification using analysis of variance, *Structural Control and Health Monitoring* 24(9) (2017), e1964.

[6] M.J. Er and F. Liu, Genetic algorithms for MLP neural network parameters optimization, in: *2009 Chinese Control and Decision Conference*, IEEE, 2009, pp. 3653–3658. ISBN ISBN 1424427223.

[7] C. Zanchettin, T.B. Ludermir and L.M. Almeida, Hybrid training method for MLP: Optimization of architecture and training, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41(4) (2011), 1097–1109.

[8] Y. Ghanou and G. Bencheikh, Architecture optimization and training for the multilayer perceptron using ant system, *International journal of computer science* 43(1) (2016), 10.

[9] D.C. Montgomery, *Design and analysis of experiments, 8th Edition*, Jhon Wiley & Sons, 2013. ISBN ISBN 978-1-118- 4692-7.

[10]S. Shankar, S. Thangarasu, T. Mohanraj and D. Pravien, Prediction of cutting force in turning process: An experimental and fuzzy approach, *Journal of Intelligent & Fuzzy Systems* 28(4) (2015), 1785–1793.

[11] F.C. Fernandes, L.M. Brasil, J.M. Lamas and R. Guadagnin, Breast cancer image assessment using an adaptative networkbased fuzzy inference system, *Pattern Recognition and Image Analysis* 20(2) (2010), 192–200.

[12] M. Alizadeh, M. Lewis, M.H.F. Zarandi and F. Jolai, Determining significant parameters in the design of ANFIS, in: *2011 Annual Meeting of the North American Fuzzy Information Processing Society*, IEEE, 2011, pp. 1–6. ISBN ISBN 1612849687.

[13] P. Raja and B. Pahat, A review of training methods of ANFIS for applications in business and economics, *International Journal of u-and e-Service, Science and Technology* 9(7) (2016), 165–172.

[14] D. Karaboga and E. Kaya, Adaptive network based fuzzy inference system (ANFIS) training approaches: A comprehensive survey, *Artificial Intelligence Review* (2018), 1–31.

[15] C.-U. Yeom and K.-C. Kwak, Performance comparison of ANFIS models by input space partitioning methods, *Symmetry* 10(12) (2018), 700.

[16] S. Ravi, M. Sudha and P.A. Balakrishnan, Design of intelligent self-tuning GA ANFIS temperature controller for plastic extrusion system, *Modelling and Simulation in Engineering* 2011 (2011), 12.

[17] E. Isen and S. Boran, A novel approach based on combining ANFIS, genetic algorithm and fuzzy c-means methods for multiple criteria inventory classification, *Arabian Journal for Scienceand Engineering* 43(6) (2018), 3229–3239.

[18] H. Moayedi, M. Raftari, A. Sharifi, W.A.W. Jusoh and A.S.A. Rashid, Optimization of ANFIS with GA and PSO estimating ratio in driven piles, *Engineering with Computers* (2019), 1–12.

[19] M.A. Shoorehdeli, M. Teshnehlab and A.K. Sedigh, Training ANFIS as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended Kalman filter, *Fuzzy Sets and Systems* 160(7) (2009), 922–948.

[20] D.P. Rini, S.M. Shamsuddin and S.S. Yuhaniz, Balanced the trade-offs problem of ANFIS using particle swarm optimisation, *TELKOMNIKA Telecommunication, Computing, Electronics and Control* 11(3) (2013), 611–616.

[21] D.P. Rini, S.M. Shamsuddin and S.S. Yuhaniz, Particle swarm optimization for ANFIS interpretability and accuracy, *Soft Computing* 20(1) (2016), 251–262.

[22]H. Kalani, M. Sardarabadi and M. Passandideh-Fard, Using artificial neural network models and particle swarm optimization for manner prediction of a photovoltaic thermal nanofluid based collector, *Applied Thermal Engineering* 113 (2017), 1170–1177.

[23] T. Çavdar, PSO tuned ANFIS equalizer based on fuzzy Cmeans clustering algorithm, *AEU-International Journal of Electronics and Communications* 70(6) (2016), 799–807.

[24] J.W. Moon, S.K. Jung, Y. Kim and S.-H. Han, Comparative study of artificial intelligence-based building thermal control methods: Application of fuzzy, adaptive neuro-fuzzy inference system, and artificial neural network, *Applied Thermal Engineering* 31(14–15) (2011), 2422–2429.

[25] C. Lakhmissi, Comparison between fuzzy, neural and neuro-fuzzy controllers for mobile robot path tracking, in: 2015 3rd International Conference on Control, Engineering & Information Technology (CEIT), IEEE, pp. 1-6.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[26] H. Fraihat, K. Madani and C. Sabourin, Learning-based distance evaluation in robot vision: A comparison of ANFIS, MLP, SVR and bilinear interpolation models, in: *2015 7th International Joint Conference on Computational Intelligence (IJCCI)*, Vol. 3, IEEE, 2015, pp. 168–173. ISBN ISBN 9897581650.

[27]K. Madani, H. Fraihat and C. Sabourin, Machine-Learningbased visual objects' distances evaluation: A comparison of ANFIS, MLP, SVR and bilinear interpolation Models, in: *International Joint Conference on Computational Intelligence*, Springer, 2015, pp. 462–479.

[28]Ö. Ki¸si, Daily pan evaporation modelling using a neuro-fuzzy computing technique, *Journal of hydrology* 329(3–4) (2006), 636–646.

[29] S. Karaman, I. Ozturk, H. Yalcin, A. Kayacier and O. Sagdic, Comparison of adaptive neuro-fuzzy inference system and artificial neural networks for estimation of oxidation parameters of sunflower oil added with some natural byproduct extracts, *Journal of the Science of Food and Agriculture* 92(1) (2012), 49–58.

[30]A. Guven and O. Kisi, Monthly pan evaporation modeling using linear genetic programming, *Journal of hydrology* 503 (2013), 178–185.

[31] F. Jafarkazemi, M. Moadel, M. Khademi and A. Razeghi, Performance prediction of flat-plate solar collectors using MLP and ANFIS, *Journal of Basic and Applied Scientific Research* 3(2) (2013), 196–200.

[32] W. Suparta and K.M. Alhasa, A comparison of ANFIS and MLP models for the prediction of precipitable water vapor, in: *2013 IEEE international conference on space science and communication (IconSpace)*, IEEE, 2013, pp. 243–248. ISBN ISBN 1467352330.

[33] A. Najah, A. El-Shafie, O.A. Karim and A.H. El-Shafie, Performance of ANFIS versus MLP-NN dissolved oxygen prediction models in water quality monitoring, *Environmental Science and Pollution Research* 21(3) (2014), 1658–1670.

[34] H. You, Z. Ma, Y. Tang, Y. Wang, J. Yan, M. Ni, K. Cen and Q. Huang, Comparison of ANN (MLP), ANFIS, SVM, and RF models for the online classification of heating value of burning municipal solid waste in circulating fluidized bed incinerators, *Waste Management* 68 (2017), 186–197.

[35] M. Moosavi, K.F. Rad and A. Daneshvar, A comparative study in the prediction of thermal conductivity enhancement of nanofluids using ANN-MLP, ANN-RBF, ANFIS, and GMDH methods, *Journal of the Iranian Chemical Society* 16(12) (2019), 2629–2637.

[36] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew, Extreme learning machine: A new learning scheme of feedforward neural networks, *Neural networks* 2 (2004), 985–990.

[37] J. Tang, C. Deng and G.-B. Huang, Extreme learning machine for multilayer perceptron, *IEEE transactions on neural networks and learning systems* 27(4) (2015), 809–821.

[38] G.N. Pillai, J. Pushpak and M.G. Nisha, Extreme learning ANFIS for control applications, in: *2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, IEEE, 2014, pp. 1–8. ISBN ISBN 1479945307.

[39] P. Jagtap and G.N. Pillai, Comparison of extreme-ANFIS and ANFIS networks for regression problems, in: *2014 IEEE International Advance Computing Conference (IACC)*, IEEE, 2014, pp. 1190–1194. ISBN ISBN 1479925721.

[40] A. Ben-Israel and T.N. Greville, *Generalized inverses: theory and applications*, Vol. 15, Springer Science & Business Media, 2003.

[41] G.-B. Huang, D.H. Wang and Y. Lan, Extreme learning machines: a survey, *International journal of machine learning and cybernetics* 2(2) (2011), 107–122.

[42] J.-S. Jang, ANFIS: Adaptive-network-based fuzzy inference system, *IEEE transactions on systems, man, and cybernetics* 23(3) (1993), 665–685.

[43] IBM, IBM SPSS Statistics Algorithms, 2017. https://www.ibm.com/support/pages/ibm-spss-statisticssubscription-documentation.