Supplementary Information

Machine learning techniques to estimate mechanised forest cutting productivity

Saulo Boldrini Gonçalves, Nilton Cesar Fiedler, Jeferson Pereira Martins Silva, Gilson Fernandes da Silva, Mayra Luiza Marques da Silva, Luciano José Minette, Daniel Pena Pereira, Domingos Manuel Mendes Lopes, Evandro Ferreira da Silva, Antônio Henrique Cordeiro Ramalho, Jeangelis Silva Santos, Marcelo Otone Aguiar, Jose de Oliveira Melo Neto, and Renisson Neponuceno de Araújo Filho

Southern Forests: A Journal of Forest Science 2021, 83(4): 01–08. <https://doi.org/10.2989/20702620.2020.1994342>

S1 Modelling with the boosted algorithm

The boosted algorithm is based on decision trees and it was developed to optimise the predictive performance. Such decision trees iteratively divide the database (parent node) into subsets (child nodes) (Zamo et al. 2014). The degree of division in the data was calculated according to certain fixed criterion (Shaikhina et al. 2017). Typically, it was done by using a cut-off threshold of the independent variables to decrease the variance of the dependent variable in the child nodes (Zamo et al. 2014).

To perform the division in the decision tree, the independent variable x_j and the cutoff point are selected. Thus, by dividing the space of the dependent variable, it leads to the greatest possible reduction in the variance of the child nodes, or R_1 (Eqn 1) and R_2 (Eqn 2).

$$R_1(j, s) = \{x / x_j < s\} \quad (1)$$

$$R_2(j, s) = \{x / x_j \geq s\} \quad (2)$$

The notation $\{x / x_j < s\}$ denotes the cut-off point of the independent variable space, where X_j assumes a value less than s , considering all independent variables $X_1, \dots, X_j$ and all possible values for s .

Thus, values of j and s that minimise (Eqn 3) are sought (James et al. 2013).

$$Erro = \sum_{i: x_i \in R_1(j, s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i: x_i \in R_2(j, s)} (y_i - \hat{y}_{R_2})^2 \quad (3)$$

where: y_i = i -th value of the dependent variable; $\hat{y}_{R_1}$ = estimated values in region 1 or child node; and $\hat{y}_{R_2}$ = estimated values in region 2 or child node.

This process occurs iteratively until the criterion for stopping is reached, which can be the final number of observations. The final group is called the leaf, and the predicted value is the average of the values pertaining to the leaves of the trees (Zamo et al. 2014).

The boosted technique is a model based on trees that have been adjusted iteratively through recursive binary divisions. The final adjusted model is a linear function of the sum of all trees multiplied by the learning rate based on all data. This procedure is demonstrated in the code below.

Input: Data $\{(x_i, y_i)\}_{i=1}^n$ and a differentiable Loss Function $L(y_i, F(x))$

Step 1: Initialise model with a constant value: $F_0(x) = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, \gamma)$

Step 2: For $m = 1$ to M

(A) Compute $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right] F(x) = F_{m-1}(x)$ for $i = 1, \dots, n$

(B) Fit regression tree to their r_{im} values and create terminal regions R_{jm} , for $j = 1, \dots, J_m$

(C) For $j = 1 \dots J_m$ compute $\gamma_{jm} = \arg \min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma)$

(D) **Update** $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} L(x \in R_{jm})$

Step 3: Output $F_M(x)$

where: x_i = model independent variables; y_i = dependent variables; n = number of observations; L = loss function; $F(x)$ = function that provides the predicted values; γ = estimated values; M = number of trees; ν = learning rate.

S2 Modelling with artificial neural networks (ANN)

Multilayer perceptron (MLP) ANNs have great potential for the study of function approximation because of their use in this study (Görgens et al. 2015; Nunes and Görgens 2016; Reis et al. 2018; Vieira et al. 2018; Silva et al. 2019). These are composed of an input layer, where the variables are presented to the network, intermediate, or hidden layers, where processing is performed, and the output layer, where the result is presented. The middle layer is responsible for identifying nonlinear patterns in the data using activation functions (Braga 2007).

In this study, the hyperbolic and logistic tangent functions were used in the intermediate layer and the linear function was used in the output layer. To use the MLP to estimate harvester productivity, it was necessary to adjust the synaptic weights of the connections between the processing units. These were adjusted in an iterative process commonly called learning or training (Braga 2007), and the training algorithms used were Levenberg–Marquardt and resilient backpropagation.

Entries were normalised at intervals of -1 to 1 and 0 to 1 , respectively, for the hyperbolic and logistic tangent activation functions. After normalisation, the inputs were inserted into the network and passed to the middle layer, where the weighted sum was calculated according to Eqn 4. The result of this calculation was transmitted to the hyperbolic Eqn 5 and logistic Eqn 6 tangent activation functions. Subsequently, calculations of the intermediate neurons served as input for the output neurons (Eqns 7 and 8).

$$a_j = \sum_{i=1} (x_i w_{ij}) + b_i \quad (4)$$

$$C_j = \frac{e^{a_j} + e^{-a_j}}{e^{a_j} - e^{-a_j}} \quad (5)$$

$$C_j = \frac{1}{1 + e^{a_j}} \quad (6)$$

$$d_k = \sum_{j=1} (c_j w_{jk}) + b_k \quad (7)$$

$$Y_K = d_k \quad (8)$$

Supplementary Figure S1: Structure of an artificial neural network (ANN)

Ten networks were trained by changing the number of neurons in the intermediate layer, which varied from 1 to 19, seeking to investigate from the application of a simpler structure to a more complex one. According to Braga (2007), the simpler structure is better on condition that it provides accurate estimates. Four ANN training stop criteria were established: maximum number of times equal to 1 000, maximum training time equal to 300 s, maximum error of 0.01, and early stopping. During training, the criterion that was reached first established the termination of processing.

3 Data-processing with adaptative network-based fuzzy inference systems (ANFIS)

ANFIS is a combination of ANN and fuzzy logic, in which the neural network generates rules or functions of relevance to fuzzy systems (Simões and Shaw 2007).

The output of the ANFIS system is modified according to the antecedent and consequent parameters. In the present work, a hybrid approach was considered, in which the antecedent parameters were adjusted by the descending gradient method and the consequent ones by the least-squares method (Akbarzadeh et al. 2014; Mathur et al. 2016).

Supplementary Figure S2: Structure of a neuro-fuzzy system (adapted from Haznedar and Kalinli (2018))

In layer 1, the fuzzification process takes place, in which the universe of real numbers is mapped to the fuzzy domain (Mathur et al. 2016). Each node in this layer creates a degree of association (μ_i) for input variables (x_1 and x_2), ranging from 0 to 1 using membership functions (A_i, B_k). In this study, a Gaussian-type function was used.

$$O_{1,j} = \mu_{A_j}(x_1) = \exp \left[- \left(\frac{x_1 - c_j}{a_j} \right)^2 \right] \quad (9)$$

$$O_{1,k} = \mu_{B_k}(x_2) = \exp \left[- \left(\frac{x_2 - c_k}{a_k} \right)^2 \right] \quad (10)$$

where: a_k, a_j, c_k and c_j are antecedent parameters to be estimated by the gradient-descending method (backpropagation).

In layer 2, the nodes are fixed and named as $O_{2,i}$. The output of each node is the product of all input signals (Akbarzadeh et al. 2014; Mathur et al. 2016).

$$O_{2,i} = \varpi_i = \mu_{A_k}(x_1) \mu_{B_k}(x_2) \quad (11)$$

In layer 3, the values of ω_i are normalised through Eqn 12:

$$O_{3,i} = \varpi_i = \frac{\omega_i}{\omega_1 + \omega_2} \quad (12)$$

In layer 4, defuzzification occurs, which is the transformation of fuzzy values to a real number (Simões and Shaw 2007).

$$O_{4,i} = \varpi_i f_i = \varpi_i (a_i x_1 + b_i x_2 + c_i) \quad (13)$$

where: the constants a_i , b_i and c_i are consequential parameters and can be adjusted by linear regression analysis using the least-squares method.

In layer 5, the sum $\varpi_i f_i$ provides the final output.

$$O_{5,i} = \sum_j \varpi_i f_i = \frac{\sum_i \omega_i f_i}{\sum_i \omega_i} \quad (14)$$

To implement the ANFIS, it is necessary to provide a data matrix containing the input (independent variables) and the output (dependent variable). To optimise the number of rules generated in the training, we used the Fuzzy C-means (FCM) (Bezdek 1981) clustering algorithm. Grouping consists of finding data points with greater similarity in the same cluster and less similarity between different cluster data (Su and Zhao 2017).

FCM finds the centres of the clusters, minimising the distance from x_k and v_i , with the objective function represented by the equations.

$$j = \sum_{k=1}^n \sum_{i=1}^c \mu_{ik}^m \|x_k - v_i\|^2 \quad (15)$$

$$v_i = \frac{\sum_{k=1}^n [\mu_i(x_k)]^2 x_k}{\sum_{k=1}^n [\mu_i(x_k)]^m} \quad (16)$$

$$\mu_{ik} = \frac{1}{\sum_{i=1}^c \left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|} \right)^{\frac{1}{m-1}}} \quad (17)$$

where: n = number of observations; c = number of clusters; x_k = k -th observation; v_i = i -th cluster centre; m = exponential constant with a value greater than 1; μ_{ik} = degree of the association function and initially occurs at random.

The FCM algorithm requires three parameters: number of clusters, exponent value, and number of iterations. In this study, the number of clusters ranged from 2 to 17 in intervals of 1, the exponent value ranged from 1.10 to 3.10 in intervals of 0.10, and the number of iterations was 100.

The associated functions for the clusters were of the Gaussian type and for the output were of linear type. The training algorithm was a hybrid type, which combined the backpropagation method with the least-squares method. The number of training periods varied between 1 and 20. The stopping criterion for training was an error equal to zero or early stopping. Early stopping is used to improve generalisation and avoid data overfitting. Therefore, the training is stopped at the instant the training database error starts to decrease and the validation database error starts to increase.

S4 Configurations analysed to estimate harvester productivity

S4.1 Configurations analysed for the boosted technique to estimate harvester productivity

Figure S3 shows the statistical indicators ($RMSE$ and r) for the configurations used, considering the training and validation sets. For the boosted technique it is necessary to specify and calibrate three parameters: learning rate, learning cycles, and learners.

Supplementary Figure S3: Correlation coefficient and RMSE (%) for training and validation data considering the settings analysed in the boosted technique

For the learning rate parameter, values from 0.1 to 0.9 were tested in intervals of 0.1. It was observed that the values of RMSE and r , respectively, ranged from 3 to 8 and 0.95 to 0.99 for the training set, and from 4 to 9 and 0.95 to 0.99 for the validation set. In addition, there was a decrease in the RMSE and an increase in r with an increase in the learning rate. However, there was a large gap between RMSE values of the training and validation sets. With a learning rate of 0.9, for example, the distance was 35%. This gap does not reach 10% when using a learning rate of up to 0.4.

When analysing the parameter learning cycles, in which a variation of 25 to 500 was tested with an interval of 25, it was observed that the values of RMSE and r , respectively, ranged from 2.5 to 10.39 and 0.93 to 0.99 for the training set, and from 3.33 to 10.56 and 0.93 to 0.99 for the validation set. In addition, as with the learning rate, there was a decrease in the RMSE and an increase in r with an increase in the number of learning cycles. Similarly, there was a large gap between the RMSE values in the training and validation sets with an increase in the number of learning cycles.

The validation error decreases monotonically with an increasing number of learning cycles; this occurs more quickly during the initial stages and then stabilises to an almost constant value as the iterations increase. Thus, the choice of a specific value for the number of learning cycles is not critical, on condition that the number is not too small.

Regarding the learning parameter, in which values from 3 to 17 were tested in intervals of 2, there were no major variations in the RMSE and r values for the training and validation sets.

S4.2 Configurations analysed for ANN technique to estimate harvester productivity

Figure S4 shows the statistical indicators (RMSE and r) for the settings used when the ANN weights were adjusted, with the training and validation sets.

Supplementary Figure S4: Correlation coefficient and RMSE (%) for training and validation data considering the settings analysed in the ANN technique

Based on the simulations to select the best ANN configuration to estimate the harvester productivity, there was no difference in the values of the statistical indicators used to assess the performance of the hyperbolic tangent (Tansig) and logistic (Logsig) functions. The values of RMSE and r , respectively, ranged from 11.59 to 13 and 0.88 to 0.91 for the hyperbolic tangent function in the training set. For the logistic function, these values ranged from 11.08 to 13.57 and 0.84 to 0.91. For the validation set, the statistical indicators followed the same pattern.

Variation in RMSE and r was observed with an increase in the number of neurons in the hidden layer of the ANN. The number of neurons in the hidden layer is linked to the complexity of the problem and, for more complex problems as studied in this work, more neurons are needed. It is worth mentioning that 37 neurons in the hidden layer were necessary to provide good statistical indicators in the training set ($RMSE = 11.76$; $r = 0.91$) and validation set ($RMSE = 11.66$; $r = 0.91$).

During the evaluation of the ANNs from the point of view of the training algorithm, it was noted that there were differences in the statistical indicators. The training algorithm modified the weights in the hidden layer iteratively until the error was reduced. Notably, the Levenberg–Marquardt algorithm presented superior statistical indicators, followed by the resilient and scaled conjugate algorithms.

S4.3 Configurations analysed for the ANFIS technique to estimate harvester productivity

Figure S5 shows the statistical indicators (RMSE and r) for the settings used when adjusting the ANFIS weights, considering the training and validation sets.

Supplementary Figure S5: Correlation coefficient and RMSE (%) for training and validation data considering the settings analysed in the ANFIS technique

The values of the RMSE and r , respectively, ranged from 12 to 14.5 and 0.85 to 0.89 for the training set, and from 12.83 to 14.90 and 0.84 to 0.99 for the validation set. In addition, a decrease in the RMSE and an increase in r were observed with an increase in the number of clusters. For the exponent parameter, it was found that the values of the RMSE and r , respectively, ranged from 12.72 to 14.85 and 0.86 to 0.89 for the training set, and from 12.85 to 15.00 and 0.84 to 0.88 for the validation set.