Neural Comput & Applic (2017) 28:505–519
DOI 10.1007/s00521-015-2075-9

ORIGINAL ARTICLE

Machine learning use in predicting interior spruce wood density utilizing progeny test information

Kostantinos Demertzis[1] · Lazaros Iliadis[1] · Stavros Avramidis[2] · Yousry A. El-Kassaby[3]

Received: 24 January 2015 / Accepted: 11 October 2015 / Published online: 16 November 2015

Abstract Several machine learning models were used to predict interior spruce wood density using data from open-pollinated progeny testing trial. The data set consists of growth (height and diameter which were used to estimate individual tree volume) and wood quality (wood density determined by X-ray densitometry, resistance to drilling, and acoustic velocity) attributes for a total of 1146 trees growing on comparable sites in interior British Columbia. Various machine learning models were developed for estimating wood density. The multilayer feed-forward artificial neural networks and gene expression programming provided the highest predictability as compared to the other methods tested, including those based on classical multiple regression which was considered as the comparisons benchmark. The utilization of machine learning models as a credible method for estimating wood density using available growth data as an indirect method for determining trees wood density is expected to become increasingly helpful to forest managers and tree breeders.

Keywords Machine learning · Artificial neutral networks (ANNs) · Interior spruce · Progeny test · Wood density

✉ Lazaros Iliadis
liliadis@fmenr.duth.gr

Yousry A. El-Kassaby
y.el-kassaby@ubc.ca

[1] Department of Forestry and Management of the Environment and Natural Resources, Democritus University of Thrace, 68200 Nea Orestias, Greece

[2] Department of Wood Science, Faculty of Forestry, University of British Columbia, Vancouver, BC V6T1Z4, Canada

[3] Department of Forest and Conservation Sciences, Faculty of Forestry, University of British Columbia, Vancouver, BC V6T 1Z4, Canada

1 Introduction

Most traditional tree improvement programs have advanced to either their second- or third-generation cycles with increased emphasis on including wood density as a major selection criterion [51]. Wood density is of tangible importance as it impacts several wood properties including bending strength and stiffness as expressed by the modulus of rupture (MoR) and the modulus of elasticity (MoE), respectively [4]. Tree breeders traditionally establish elaborate, long-term, and resource-dependent tree improvement programs following the recurrent selection scheme [1]. These programs consist of repeated cycles of breeding, testing, and selection, for estimating the genetic worth of the tested trees for identifying the elite genotypes for either their inclusion in new breeding cycles or production populations (i.e., seed orchards) [39]. An important feature of most tree breeding programs is determining the magnitude of genetic control (heritability) of the attributes under selection (e.g., growth, survival, resistant to pests, and wood density). While it is known that the heritability estimate is population-specific, nonetheless, the higher the heritability, the greater the genetic control over the attribute in question [12].

Wood density is commonly measured either by the volumetric approach [2] or by X-ray densitometry [9]. For this purpose, increment cores are extracted from the studied trees followed by sample preparation and wood density determination. This approach is intrusive and time-consuming and is a two-step method for sampling and wood density determination. Alternative in situ, one-step, non-destructive, and rapid methods for determining wood density of standing trees have been developed (resistance to drilling using Pilodyn [8] and Resistograph® [52] or acoustic velocity using Director® ST300 [6]) and successfully tested [5, 7, 11, 27, 43, 44].

The progeny testing phase of tree improvement programs involves the establishment of multiple sites planted throughout the target breeding zone, each with various numbers of replications for sampling sites' heterogeneity, and finally each replication accommodating several plots (single tree, rows, or blocks) of the tested offspring, with tree number often exceeding 100,000s [54, 55]. Estimating wood density using destructive and/or in situ methods is commonly done on a subsample of the progeny test due to the prohibitive large number of trees [11, 43]. The development of a heuristic rather than algorithmic approach for predicting wood density utilizing existing progeny testing information would be of great value as it will provide substantially faster results with even less reliance on estimating wood density as it could be derived from site, progeny pedigree, and other growth attributes. The feasibility of this approach was successfully demonstrated by Iliadis et al. [26] on Douglas fir [*Pseudotsuga menziesii* (Mirb.) Franco var. *menziesi*] progeny test using information from actual (X-ray densitometry) and in situ (acoustic velocity) wood density, multiple test sites, genetic pedigree, and growth (height, diameter, and volume) attributes in a nonlinear modeling approach to predicting wood density using artificial neural networks (ANNs). Iliadis et al. [28] using two ANNs [the multilayer feed-forward (MLFF) and modular (MOD) models] developed a significant relationship between actual and predicted wood density with coefficient of determination (R^2) values of 0.50 and 0.52 for MLFF and MOD, respectively, with substantially higher predictive power than that obtained using the classic multiple regression approach ($R^2 = 0.23$).

In the present study, we investigated and compared a multitude of machine learning methods to identify the best approach for predicting "interior spruce" wood density from progeny testing data. We utilized actual (X-ray densitometry) and in situ (acoustic velocity and drilling resistance) wood density, and growth (height, diameter, and volume) attributes from 1146 36-year-old trees representing 25 open-pollinated families growing on three test sites in interior British Columbia, Canada.

2 Materials and methods

2.1 Field experiment

A total of 25 open-pollinated "interior spruce" [common name for white (*Picea glauca* (Moench) Voss.] and Engelmann (*Picea engelmannii* Parry ex Engelm.) spruces and their hybrid [48] families provided the data for this study. Parent trees originated from low- to mid-elevations ($\approx$650–1500 m) east and southeast of Prince George breeding zone of British Columbia. Three test sites, namely Aleza Lake (Lat. 54°03′15.7″N, Long. 122°06′35.4″W, Elev. 700 mas), Prince George Tree Improvement Station (PGTIS) (Lat. 53°46′17.9″N, Long. 122°43′07.6″W, Elev. 610 mas), and Quesnel (Lat. 52°59′27.2″N, Long. 122°12′30.6″W, Elev. 915 mas) provided the data for this study. Test sites were established as a complete randomized block design with 5 or 10 replications (blocks) with either 10- or 15-tree row plots [32]. Four randomly selected trees from each family from only four replications were measured ($N = 1146$ trees). In the summer of 2009 (36 years old), each tree was measured for height (m) and diameter (cm) (a.k.a., diameter at breast height: dbh) and each tree's volume (m^3) was calculated following the method of Millman [38]. Additionally, each tree's wood quality was evaluated using two non-destructive methods (drilling resistance and acoustic velocity using Resistograph® IML F300 and Director® ST300, respectively) and one intrusive sampling methods requiring 5-mm cores extracted at dbh for X-ray densitometry determination (see El-Kassaby et al. [11], Radcliffe et al. [43], for details). Each tree's drilling resistance measurement was the average of two drills done at height of 1 m in two mutually perpendicular directions and provided an estimate of relative density. The acoustic velocity measurement also was the average of three readings and was used to obtain an indirect estimate of MoE. X-ray wood density data represent the benchmark for evaluating the two non-destructive wood quality assessment methods. The X-ray densitometry was performed by QTRS-01X Tree Ring Scanner (Quintek Measurement Systems Inc., USA) and was represented by the average density of each years' early and late wood across all 36 years (kg m^{-3}).

2.2 Data

Data related to the progeny tests' experimental design and trees' attributes [i.e., site, family, replication, height, diameter, volume, director (velocity), resistograph (drilling resistance)] represent eight independent parameters, namely site [3; PGTIS, Aleza Lake, and Quesnel], replication [4], family [25], height (m), diameter (cm), volume (m^3), director, and resistograph. The dependent parameter was core wood density obtained by X-ray densitometry (kg m^{-3}).

2.3 Data preprocessing

Initially, data preprocessing was conducted which aimed at phasing problems that might emerge during their selection (e.g., removal of extreme outliers and missing records). Following data "cleaning," data were normalized and scaled. This was done in order to fit the requirements of the learning algorithms and to phase the problem where high values prevail, by influencing more the cost function, without being more important at the same time. Both input and output data were scaled between the closed interval [−1, +1]. The following function (Eq. 1) was used by the *Normalize* filter of the Weka software for this task [22].

$$\Upsilon, * = \frac{\Upsilon_i - \mu}{\sigma} \tag{1}$$

where μ and σ are the sample mean and standard deviation for the specific feature, respectively. Individual trees with missing values (for one or more parameters) were removed, and the data set was reduced to 1143 data vectors. The estimation of the extreme values was done under the *interquartile range* filter (IQRF) of the Weka software [22], where extreme values and outliers are identified and removed based on interquartile range (IQR). IQR is the difference between the third (Q_3) and the first (Q_1) quartiles (IQR $= Q_3 - Q_1$). The quartiles divide the data set in four equal groups. The IQR includes the 50 % of the data around the median, whereas the rest 25 % is smaller than Q_1 and the 25 % is higher than Q_3 [56].

Also, feature reduction was performed in order to improve the regression results, by locating the cases with noise that reduces performance. In the feature selection process, we tried to exclude the parameters that might contain information irrelevant to the case under analysis. Principal components analysis (PCA) was used for the transformation of the vector feature space [22, 25]. PCA is a method to transform the correlated available parameters to new uncorrelated ones and at the same time to provide as much as possible information on the fluctuation of the initial features. It enables the concentration of the information included in the initial data.

Finally, a feature selection approach was performed. Feature selection is the process where a subset with the most relative and correlated variables is chosen for the development of the optimal model suitable for a specific case study [22, 25]. However, unfortunately after the application of PCA and feature selection, the efficiency and the obtained results were not improved and the performance remained very low.

3 Development of machine learning models

Various machine learning models capable of estimating wood density were developed, and the most successful are presented and discussed in the following section. In all cases, a tenfold cross-validation was employed in order to compare the machine learning models. Table 1 presents an overall evaluation and Table 2 parameters of all employed methodologies.

4 Artificial neural networks

ANNs are inspired from the human central nervous system and are widely used toward nonlinear modeling [16, 41] and generally are part of computational intelligence. They approximate the function of the human brain and the biological learning processes [42]. ANNs are networks of interconnected simple processing elements (neurons).

Table 1 Comparison of the performance of all machine learning methods employed

ID	Algorithm[a]	MSE	RMSE	Correlation coefficient (R)	Coefficient of determination (R^2)	Validation
1	MLFF ANN	0.0003175	0.0178215	0.8077000	0.6524	Tenfold CV
2	GEP	0.0003575	0.0189035	0.8026980	0.6443	Tenfold CV
3	CFF ANN	0.0004005	0.0200065	0.7919220	0.6271	Tenfold CV
4	GMDH	0.000426	0.0206280	0.7770410	0.6038	Tenfold CV
5	ELM RBF kernel	0.000558	0.0235000	0.7347205	0.5400	Tenfold CV
6	ELM linear kernel	0.002978	0.0545500	0.6122500	0.3750	Tenfold CV
7	SLR	0.000686	0.0262000	0.6113000	0.3737	Tenfold CV
8	DENFIS	0.003600	0.0600000	0.5742820	0.3299	Tenfold CV
9	RBF network	0.001024	0.0320000	0.2396000	0.5741	Tenfold CV
10	Multilayer perceptron	0.000441	0.0210000	0.7707000	0.5940	Tenfold CV
11	RFs	0.000462	0.0215000	0.7686000	0.5907	Tenfold CV
12	MLR	0.000449	0.0212000	0.7676000	0.5892	Tenfold CV
13	RBF regressor	0.000449	0.0212000	0.7671000	0.5884	Tenfold CV
14	MLP regressor	0.000458	0.0214000	0.7588000	0.5758	Tenfold CV
15	EFuNN	0.000900	0.0300000	0.7014270	0.4920	Tenfold CV
16	SVR	0.000445	0.0211000	0.7708000	0.5941	Tenfold CV

[a] See text for abbreviations

Table 2 Parameters of all machine learning methods employed

ID	Algorithm	Parameters
1	MLFF ANN	Architecture = 8-10-1 Training algorithm = back-propagation Transfer function = tansig Learning rate = 0.3 Momentum = 0.2 Training epochs = 100

2	GEP	Number of chromosomes = 30
		Head size = 10
		Genes = 5
		Constants per genes = 10
		Mutation = 0,00138
		Strategy = optimal evolution

3	CFF ANN	Architecture = 8-(10-5-10)-1 Training algorithm = back-propagation Adaption learning function = learngdm Transfer function = tansig Training epochs = 100
4	GMDH	Validation criterion = RMSE Variables ranking = by error Neuron function = polynomial Max. number of layers = 4 Max. power of a variable = 2 Min. power of a variable = 0 Max. total power in a term = 2 Max. number of variables in a term = 2
5	ELM RBF kernel	Number of hidden neurons = 25 Activation function = RBF Regularization coefficient = 1 Block = 1
6	ELM linear kernel	Number of hidden neurons = 25 Activation function = linear Regularization coefficient = 1 Block = 1
7	SLR	Ridge value = 1.0E-8
8	DENFIS	Distance threshold = 0.1 Number of nodes = 3 Epochs = 2
9	RBF network	Maximum number of iterations = 100 Minimum standard deviation for the clusters = 0.1 Number of clusters for *K*-means = 2 Ridge value = 1.0E-8
10	Multilayer perceptron	Hidden layers = 8 Learning rate = 0.3 Momentum weights updating = 0.2 Training time = 500 Validation threshold = 20
11	RFs	Maximum depth = unlimited Number of execution slots = 1 Number of trees = 100
12	MLR	Ridge value = 1.0E-8

Table 2 continued

ID	Algorithm	Parameters
13	RBF regressor	Size of thread pool = 1 Ridge value = 0.01 Delta value = 1.0E-6
14	MLP regressor	Hidden layers = 8 Size of thread pool = 1 Ridge value = 0.01 Delta value = 1.0E-6
15	EFuNN	Sensitivity threshold = 0.9 Error threshold = 0.1 Number of membership function = 3 Learning rate for $W1$ = 0.1 Learning rate for $W2$ = 0.1 Node age = 60 Max. field = 0.5 Number of activity rule nodes = 2
16	SVR	Cost parameter = 1 Degree of the kernel = 3 Epsilon criterion = 0.001 Gamma parameter = 1/max_index Epsilon for the loss function = 0.1

Following the analogue of the human brain, the interconnected neurons of the multilayer feed-forward (MLFF) ANNs are organized in three layers, namely input, hidden, and output (Fig. 1). Each neuron accepts a set of numerical inputs from various sources, and based on their information, an output is produced. The output is either addressed to the environment or forwarded as input to other neurons of the network. In an idealized artificial neuron, all received input information is weighted via appropriate elements w_i and summed up. The hidden neurons multiply each input signal with its corresponding synaptic weight (w_i), and they calculate the sum of the products. An

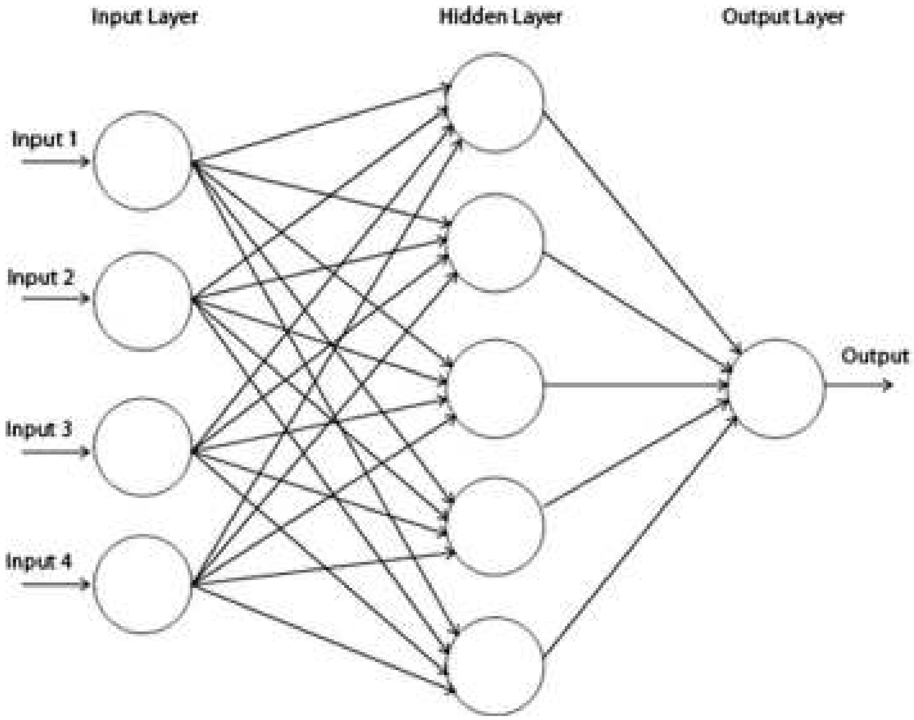

Fig. 1 Basic architecture of a typical multilayer feed-forward ANN

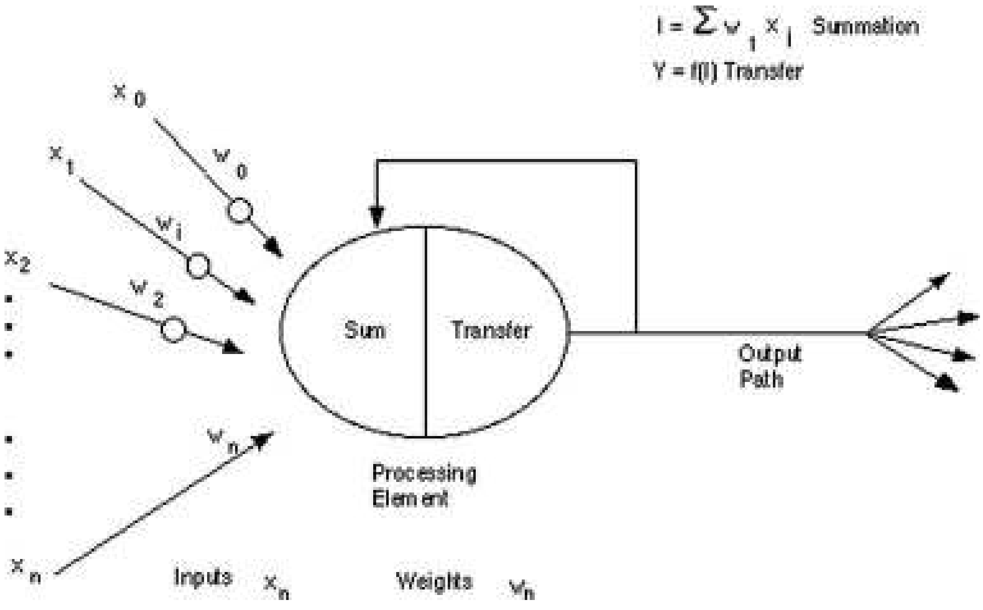

Fig. 2 An illustration of an artificial neuron and its output function

activation function transforms the sum of the weighted inputs and produces the output (Fig. 2).

ANNs pass through an iterative learning process in order to be able to capture complex signals from the data and to obtain better predictive performance. In each iteration, the weights of the ANNs are steadily adjusted, in order to minimize the differences between observed and predicted output [18]. The main target is to obtain a network capable of generalizing, meaning that it should have a reliable performance with data vectors that have not been used in the training phase. That is why overtraining and memorization have to be avoided by stopping the learning process when a predefined threshold of the error function in training samples is reached [19, 21, 25]. From this point of view, the tuning of the weights is actually an optimization approach. The level of good performance of the ANN is determined in the testing phase. The most common supervised learning optimization approach for ANNs is the back-propagation (BP) which can be considered as a gradient descent method to locate the optimal solution [46] and is a generalization of the delta rule [21, 34].

5 Multilayer feed-forward (MLFF) ANN

Several trial and error experiments should be performed in order to determine the best MLFF model. Various architectures and transfer functions were employed. The optimal wood density ANN was a MLFF ANN with a sigmoid transfer function in the hidden layer and a linear function in the output layer. This MLFF ANN had 8 input neurons, 16 hidden neurons, and 1 output. This was implemented in MATLAB [49] using the trainlm MATLAB's network training function that updates weights and bias values according to Levenberg–Marquardt optimization algorithm [20, 37]. The trainlm is often the fastest BP algorithm in the toolbox of MATLAB and is highly recommended as a first-choice supervised algorithm, although it does require more memory than other algorithms (http://www.mathworks.com/help/nnet/ref/trainlm.html).

The mean squared error (MSE, [35]), root-mean-square error (RMSE), the correlation coefficient R, and the coefficient of determination R^2 were used as performance metrics during training, validation, and testing processes. MSE (Eq. 2) stands for the average squared error between the network outputs a_i and the target outputs t_i.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \qquad (2)$$

The estimated average R^2 values for the optimal network were 0.6524, and the average RMSE = 0.0178215 (Table 1, ID #1). The convergence and performance of the MLFF ANN are shown in Fig. 3. The accuracy comparison of the machine learning algorithms with tenfold cross-validation is shown in Table 1.

Additionally, various other ANN architectures were developed, trained, and evaluated; however, they did not produce encouraging results. The degree of convergence for the best developed radial basis function (RBF) network was characterized by an average R^2 equal to 0.5740 and by an average RMSE equal to 0.0320 (Table 1, ID #9). For the multilayer perceptrons, the average R^2 was as high as 0.5939 and the average RMSE was equal to 0.0210 (Table 1, ID #10).

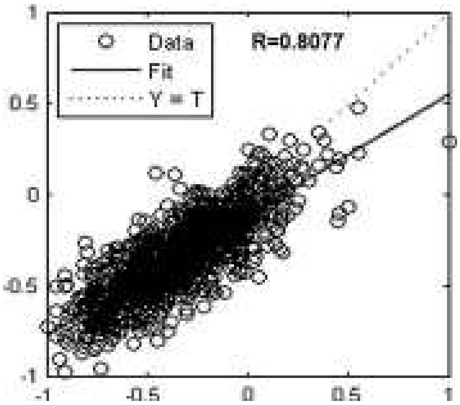

Fig. 3 Performance of MLFF ANN (MSE = 0.0003175, RMSE = 0.0178215, R = 0.8077, R^2 = 0.6524)

6 Cascade feed-forward (CFF) ANN

The cascade feed-forward (CFF) ANN shares the same basic architecture with the MLFF ANN. The main difference is that in the former, the neurons of each layer are connected with the neurons of all successive layers and not only with the nodes of the following one [17].

This architecture makes CFF networks more flexible and also more complex due to the fact that there are more weights involved. A back-propagation BP CFF ANN with 8 input nodes, 3 hidden sublayers with 10-5-10 neurons, and 1 output, using a sigmoid transfer function in the hidden layer, was the optimal one. Again, the tenfold cross-validation method was used. The average R^2 for the optimal network was as high as 0.6271, and average RMSE was 0.0200, respectively (Table 1, ID #3). This performance was slightly poorer than that observed for the MLFF ANN. The architecture of the developed CFF ANN and its performance are shown in Figs. 4 and 5, respectively. The convergence and performance of the CFF ANN are shown in Fig. 5.

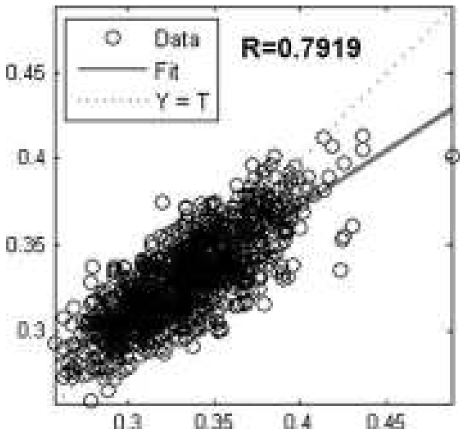

Fig. 5 Performance of CFF ANN (MSE = 0.0004005, RMSE = 0.0200065, R = 0.791922, R^2 = 0.62714)

7 Group method of data handling (GMDH) ANN

GMDH-type neural networks are also known as *polynomial neural networks* (*PNNs*). They constitute flexible ANN whose topology is not predefined, but it is developed through an iterative learning process [28]. GMDH algorithms are inductive, and they perform sorting of gradually complicated polynomial models where they finally chose the optimal solution by employing the "external criterion." A GMDH model with multiple inputs and one output is a subset of components of the Kolmogorov–Gabor *base function* ([33, 36], https://www.gmdhshell.com).

A GMDH ANN can use multilayer structures (Fig. 6), and it estimates the parameters of each obtained model and

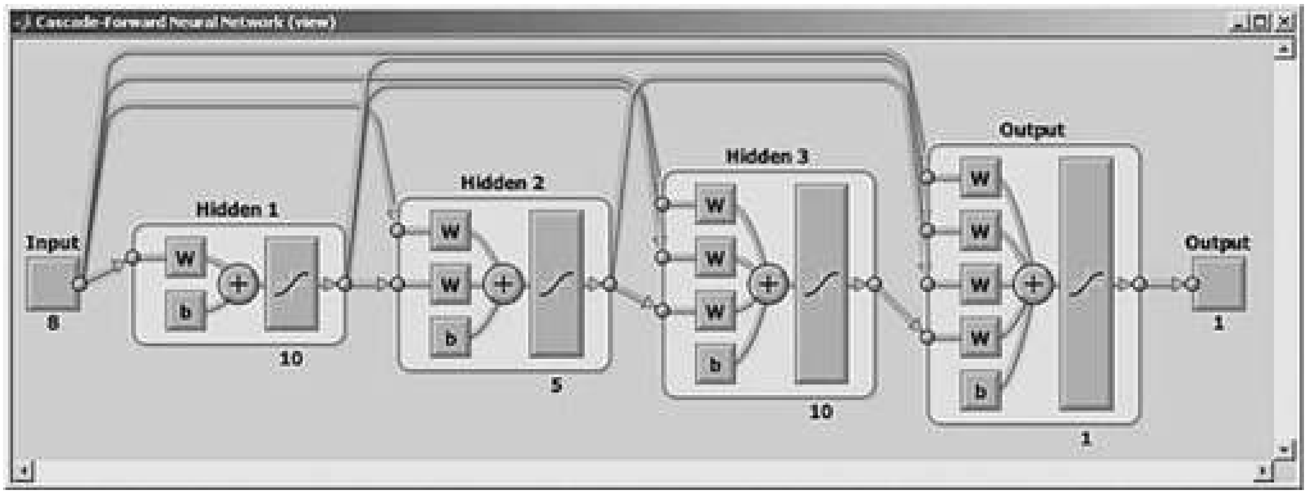

Fig. 4 Architecture of the CFF ANN

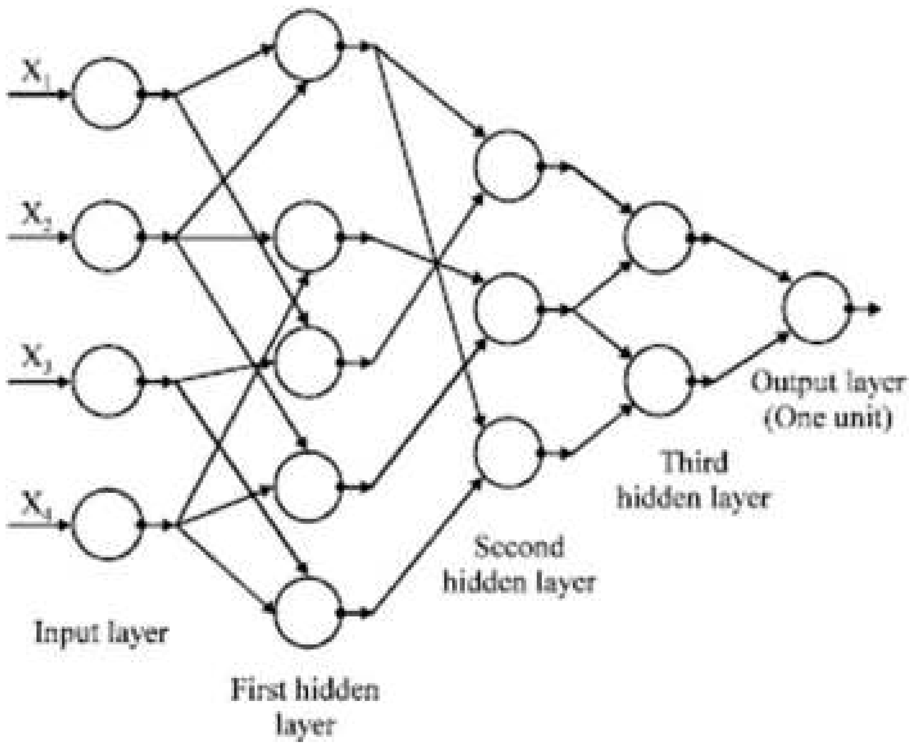

Fig. 6 Basic architecture of the group method of data handling neural networks (GMDH-type ANNs) or polynomial neural networks (PNNs)

validates it by using a separate part of the data that was not used in the previous process. Generally speaking, this method produces multilayer network models with linear, polynomial, logistic, gaussian, harmonic, and other nonlinear functions. It can identify nonlinear relations between the input and output parameters [40] and terminates when it fails to perform better [13]. Moreover, it might terminate if the reduction in the error level is <1 %, or if the number of layers has reached an upper predefined limit. Since the width of a network layer rarely improves the model, this algorithm reduces by 50 % the width of each successive layer. Thus, the number of neurons in layer k are $N_k = 0.5 \cdot N_{k-1}$ and in this way, the algorithm becomes faster and the possibility of performance reduction is low [3].

The average R^2 for the optimal network was equal to 0.6038, and RMSE was 0.0206, respectively (Table 1, ID #4). Along these lines, the GMDH approach produced similar performance to MLFF and CFF ANNs but with reduced accuracies.

8 Extreme learning machine (ELM)

Extreme learning machines (ELMs) are based on a fast learning algorithm for single-hidden-layer feed-forward neural networks (SLFFNs). The hidden layer (a.k.a., feature mapping) needs no tuning, and the hidden neurons are randomly created. The hidden neurons are independent from the target functions or from the training sets (Fig. 7). ELMs provide effective solutions in feed-forward and RBF ANNs and also in kernel learning methods (http://extreme-learning-machines.org). They can be applied in universal approximation and classification, ridge regression, optimization, ANN generalization performance, linear system stability, and matrix theory. Their contribution can be significant for large-scale computing and big data sets. ELMs can handle non-differentiable activation functions, and they do not have issues such as finding a suitable stopping criterion, learning rate, and learning epochs. According to the ELM theory [23], the RBF Gaussian kernel is employed as follows:

$$K(u, v) = \exp(-\gamma||u - v||^2) \tag{3}$$

where $\gamma = \frac{-1}{2\sigma^2}$. In fact, the parameter σ determines the width of the Gaussian kernel. In statistics, when we consider the Gaussian probability density function, it is called the standard deviation (σ). In the present study, we also used a linear kernel which is given by the inner product <x, y> plus an optional constant "c" as seen in the following Eq. 4:

$$K(x, y) = x^T y + c \tag{4}$$

According to the optimal Gaussian ELM model with RBF kernel, the RMSE value was equal to 0.0235 and the R^2 was as high as 0.54 (Table 1, ID #5). The application of the ELM with the linear kernel did not have reliable

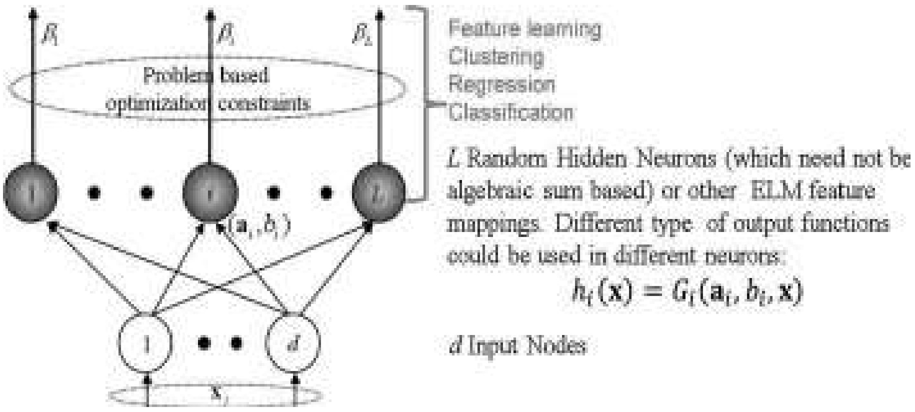
β_1
β_i
β_L
Problem based optimization constraints
Feature learning
Clustering
Regression
Classification
1
i
L
$(\mathbf{a}_i, b_i)$
L Random Hidden Neurons (which need not be algebraic sum based) or other ELM feature mappings. Different type of output functions could be used in different neurons:
$h_i(\mathbf{x}) = G_i(\mathbf{a}_i, b_i, \mathbf{x})$
1
d
d Input Nodes
$\mathbf{x}_j$

Fig. 7 General architecture of ELMs (after, Huang [25])

validation outcome with RMSE $= 0.0545$ and $R^2 = 0.3750$ (Table 1, ID #6).

9 Evolving connectionist systems

Evolving connectionist systems (ECOS) [30] are multimodular, connectionist architectures that facilitate modeling of evolving processes and knowledge discovery [47]. ECOS' advantage is that they learn fast by, for example, using one-pass training which is much faster than the iterative process of other ANN models.

ECOS keep on adapting their structure and functionality, through a continuous interaction with the environment according to: (i) a set of parameters that are subject to change, (ii) an incoming continuous flow of information with unknown distribution, and (iii) a goal (rational) criterion (subject to modification) applied to optimize the performance of the system. The ECOS function in an open space, using constructive processes, not necessarily of fixed dimensions. They learn in online incremental fast mode, possibly through one pass of data propagation. Lifelong learning is a main attribute of this procedure. They operate both as individual systems, and as part of an evolutionary population of such systems [24, 30]. ECOS are connectionist structures that evolve their nodes and connections through supervised incremental learning from input–output data. Their architecture comprises five layers: (1) input nodes, representing input variables; (2) input fuzzy membership nodes, representing the membership degrees of the input values to each of the defined membership functions; (3) rule nodes, representing cluster centers of samples in the problem space and their associated output function; (4) output fuzzy membership nodes, representing the membership degrees to which the output values belong to defined membership functions; and (5) output nodes, representing output variables (Fig. 8) [29].

ECOS produce local models through data clustering and by associating a local output function for each cluster. Rule nodes evolve from the input data stream to cluster the data, and the first-layer W_1 connection weights of these nodes represent the coordinates of the nodes in the input space. The second-layer W_2 represents the local models (functions) allocated to each of the clusters. Clusters of data are created based on the similarity between data samples either in the input space, or in both the input space and the output space. Samples that have a distance to an existing cluster center (rule node) N of less than a threshold $R_{\max}$ are allocated to the same cluster N_c. Samples that do not fit into existing clusters form new clusters as they arrive in time. Cluster centers are continuously adjusted according to new data samples, and new clusters are created incrementally. The similarity between a sample $S = (x, y)$ and an existing rule node $N = (W_1, W_2)$ can be measured in different ways, the most popular of them being the normalized Euclidean distance. The following ECOS architectures were used to model wood density.

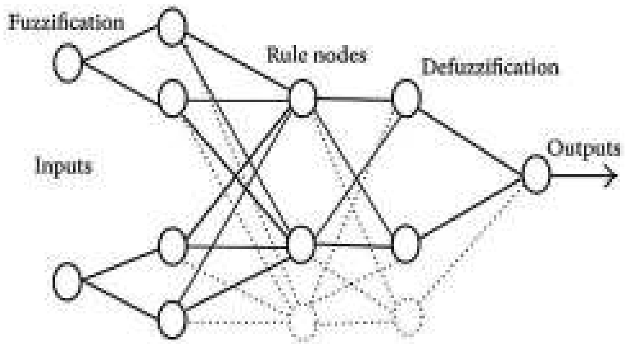

Fig. 8 Structure of EFuNN (after, Watts [50])

10 Dynamic evolving neuro-fuzzy inference system

The first ECO approach used was the dynamic evolving neuro-fuzzy inference system (DENFIS). This is an adaptive learning approach which evolves through hybrid (supervised/unsupervised) learning methods creating new fuzzy rules. These rules are updated through the system's operation. In this way, the output of a DENFIS system is estimated through a fuzzy inference system. DENFIS can be effectively used for learning complicated time series in an adaptive manner, bypassing the complexity problems which might emerge [31].

In the present study, we employed a tenfold cross-validation approach (www.theneucom.com) which resulted in an average RMSE of 0.0600 and an average R^2 equal to 0.3298, respectively (Table 1, ID #8), producing less satisfactory results when compared to the previous methods.

11 Evolving fuzzy neural networks

The second ECOS approach employed was the evolving fuzzy neural networks (EFuNN). EFuNN are feed-forward ANNs of five layers with each one performing a specific task. The input layer is the first. The second layer estimates the fuzzy membership degrees of the input values for proper linguistics like "low," "medium," and "high" by using specific fuzzy membership functions. The third layer represents the correlations between the input and the output parameters by using fuzzy *IF–THEN* rules. The fourth layer calculates the degrees to which output membership functions are matched by the input data, and the last one performs defuzzification in order to estimate the crisp values.

The EFuNN combines the characteristics of an ANN and a fuzzy inference system. It combines the following algorithms: modified back-propagation algorithm, genetic algorithm, structural learning with forgetting, training and zeroing, and combined modes. We employed a tenfold cross-validation which resulted in an average RMSE of 0.0300 and an average R^2 of 0.4920 (Table 1, ID #15). The use of this method showed a significant improvement compared to DENFIS, but still the MLFF (see above) and gene expression programming (GEP) (see below) outperformed this approach, whereas the cascade produced similar results.

12 Gene expression programming

Gene expression programming (GEP) [14] is an evolutionary algorithm which, like the genetic algorithms (GA) and genetic programming (GP), uses populations and chooses the cases according to their suitability (fitness). It imports new instances of potential solutions in the population by using one or more genetic operators. The fundamental steps of the GEP are schematically represented in Fig. 9 [14]. The process begins with the random generation of the chromosomes of a certain number of individuals (the initial population). The chromosomes of the individuals of the initial population are randomly generated using the symbols representing the functions and terminals chosen to

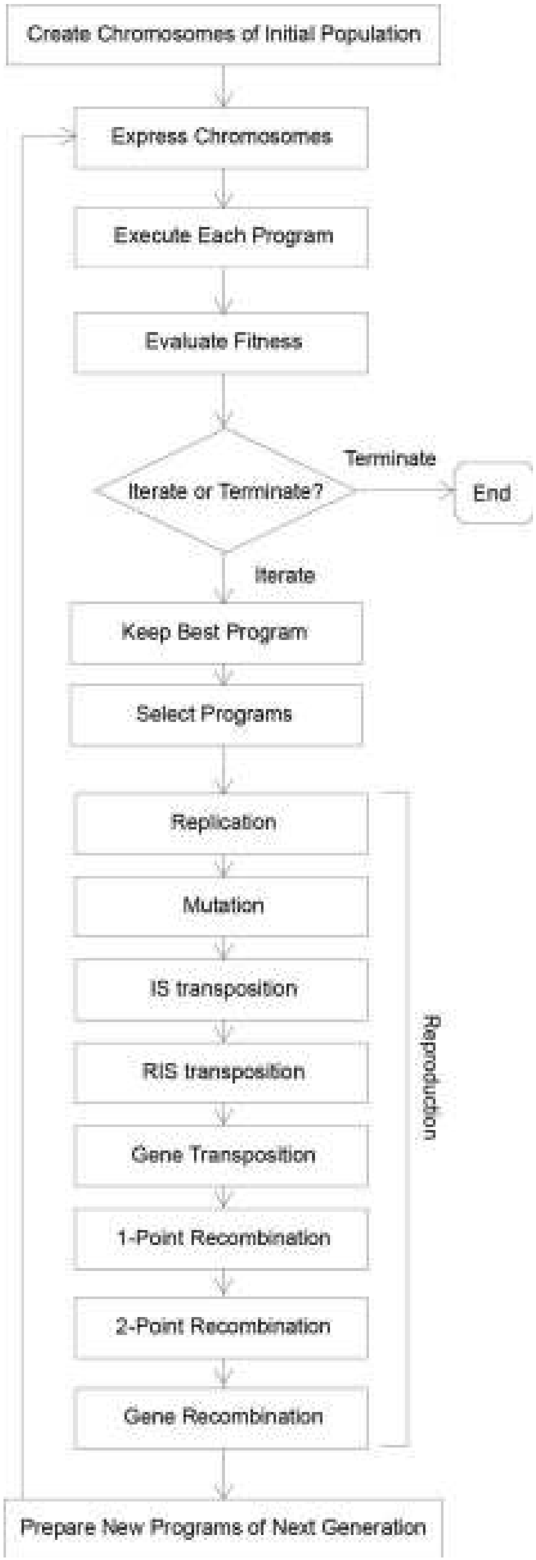

Fig. 9 Flowchart of a GEP algorithm

solve the particular problem. These initial individuals are the first set of candidate solutions to the problem at hand. Because they are totally random, these founder individuals are most probably not very good solutions. But they are, nonetheless, everything that is necessary because evolution takes care of the rest and, soon enough, very good solutions will start to appear. One important application of GEP is symbolic regression or function finding, where the goal is to find a symbolic expression that performs well for all fitness cases within a certain error of the correct value. For most symbolic regression applications, it is important to use small relative or absolute errors in order to discover a very good solution. But if we excessively narrow the range of selection and only allow the selection of individuals performing within a very small error, populations evolve very inefficiently and, most of the times, are incapable of finding a satisfactory solution. On the other hand, if we do the opposite and excessively enlarge the range of selection, numerous solutions with maximum fitness, fmax, will appear that are far from good solutions. So, a simple fitness function which only discriminated between cases solved within and without the chosen error will not suffice, and a more skilled approach is required. To solve this problem, we devised an evolutionary strategy that permits the discovery of very good solutions without halting evolution. With this scheme, the system is left to find for itself the best possible solution within a minimum error. For that purpose, a very broad limit for selection to operate is given, that allows not only the evolutionary process to get started but also the fine-tuning around the desired minimum error. And what is observed is that the individuals of early generations are usually very unfit, but their later descendants are continually reshaped by the genetic operators, and populations adapt wonderfully, finding better solutions that progressively approach a very good solution. The individuals are then selected according to their fitness (their performance in that particular environment) to reproduce with modification, leaving progeny with new traits. These new individuals are, in their turn, subjected to the same developmental process: expression of the genomes, confrontation of the selection environment, selection, and reproduction with modification. The whole process is repeated till the most effective solution is obtained [14]. In the present study with tenfold cross-validation configuration, the $R^2 = 0.6443$ and $\mathrm{RMSE} = 0.0189$, respectively (Table 1, ID #2). This method's performance was the second best after MLFF.

13 Multiple linear regression

Multiple linear regression (MLR) attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. Every value of the independent variable x is associated with a value of the dependent variable y. The population regression line (PRL) for p explanatory variables x_1, x_2, …, x_p is defined to be $\mu_y = \beta_0 + \beta_{1\times 1} + \beta_{2\times 2} + \cdots + \beta_{p\times p}$. The PRL describes how the mean response y changes with the explanatory variables. The observed values for y vary about their means $\bar{y}$ and are assumed to have the same standard deviation. The fitted values b_0, b_1, …, b_p estimate the parameters β_0, β_1, …, β_p of the population regression line [15]. In the case of the MLR, a tenfold cross-validation approach was performed. The performance results obtained by the MLR were average R^2 and average RMSE of 0.5892 and 0.0212, respectively (Table 1, ID #12). It should be stated that the results obtained from the MLR, which represent the benchmark for comparison with the other methods attempted in this study, are exceedingly inferior even when they were based on real data. Additionally, simple linear regression (SLR) was also attempted in the present study, but produced non-satisfactory results: average R^2 and RMSE of 0.3737 and 0.0262, respectively (Table 1, ID #7). RBF regressors (http://www.cs.waikato.ac.nz/~eibe/pubs/rbf_networks_in_weka_description.pdf) and MLP regressors [54] were also employed through tenfold cross-validation with average values of $R^2 = 0.5884$, $\mathrm{RMSE} = 0.0212$, and average values of $R^2 = 0.5757$, $\mathrm{RMSE} = 0.0214$ for RBF and MLP, respectively (Table 1, ID # 13 and 14). Both regressors algorithms were developed in University of Waikato, New Zealand. The RBF one is employed to train models where the loss function uses penalized squared error, with a quadratic penalty on the non-bias weights in the output layer. Other algorithms and approaches (e.g., support vector regression (SVR) [10] or rotation forests (RFs) [45]) have also been used to determine wood density; however, their performance was quite inferior to the approaches described above. SVM and SVR were proposed by Vladimir Vapnik and his co-workers in 1992. They define a certain loss function ignoring errors which are located in a certain distance from the actual value. This is achieved by employing the "epsilon intensive" function that defines a zero-error "tube" of width equal to 2e. Rotation forest is an algorithm based on feature extraction. According to this algorithm, the feature set is randomly split into K subsets (K is a parameter) and then principal component analysis is applied to each subset. All principal components are retained in order to preserve the variability information in the data.

More specifically, the optimal SVR had average $R^2 = 0.5941$ and average $\mathrm{RMSE} = 0.0211$, whereas the obtained performance results for the RFs were $R^2 = 0.5907$ and average $\mathrm{RMSE} = 0.0215$ (Table 1, ID # 11 and 16).

14 Conclusion

The importance of incorporating wood density as a selection criterion in advanced-generation breeding programs cannot be understated due to: (1) the often observed negative genetic correlation between growth rate and wood density as the former often represents the main selection goal of most improvement programs' first-generation cycle [11] and (2) the importance of wood density as an intrinsic characteristic of wood properties at large [4]. Irrespective of the method used to determine wood density, these methods are time-consuming and require the use of large number of samples. The development of precise predication methods that rely on other easily determined attributes such as growth traits would be of great values, specifically considering the large number of trees and multiple sites often used in progeny testing evaluation. The utilization of heuristic approaches for predicting wood density utilizing existing progeny testing information is expected to speed the assessment phase and assess breeders in making

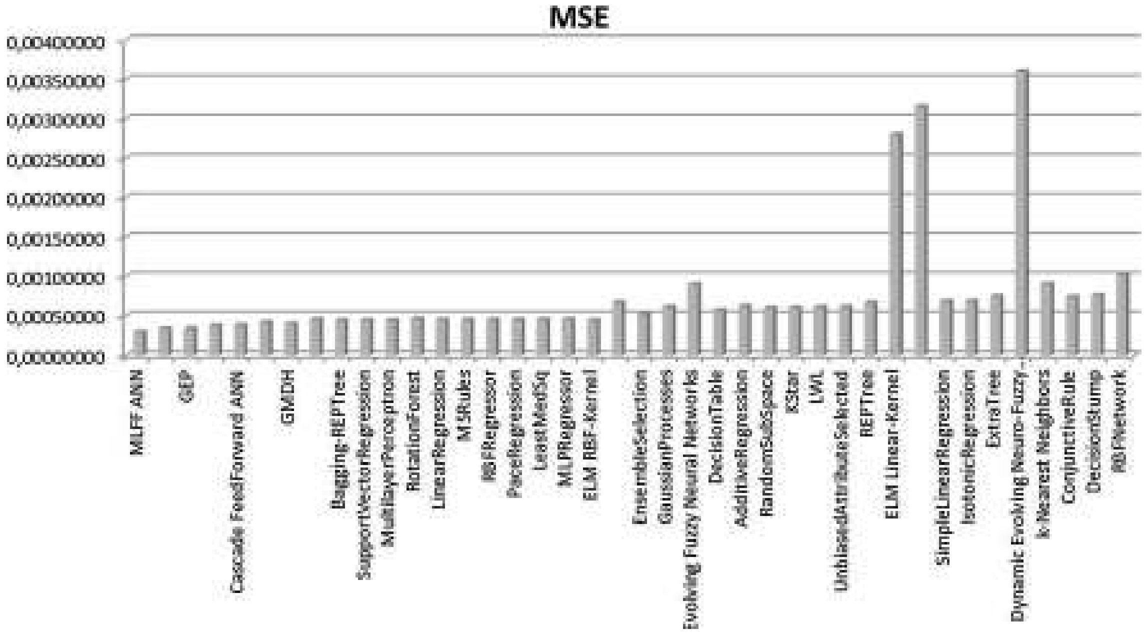

Fig. 10 Comparison of MSE (*note* MLFF ANN has the smallest error)

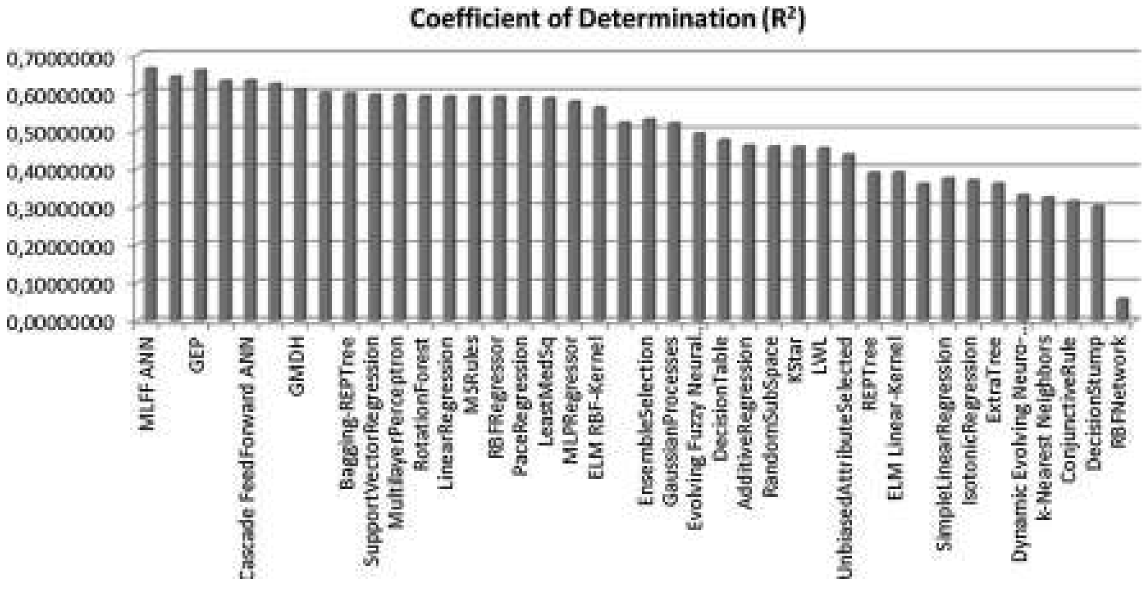

Fig. 11 Comparison of coefficient of determination (R^2) values (*note* MLFF ANN has the highest value)

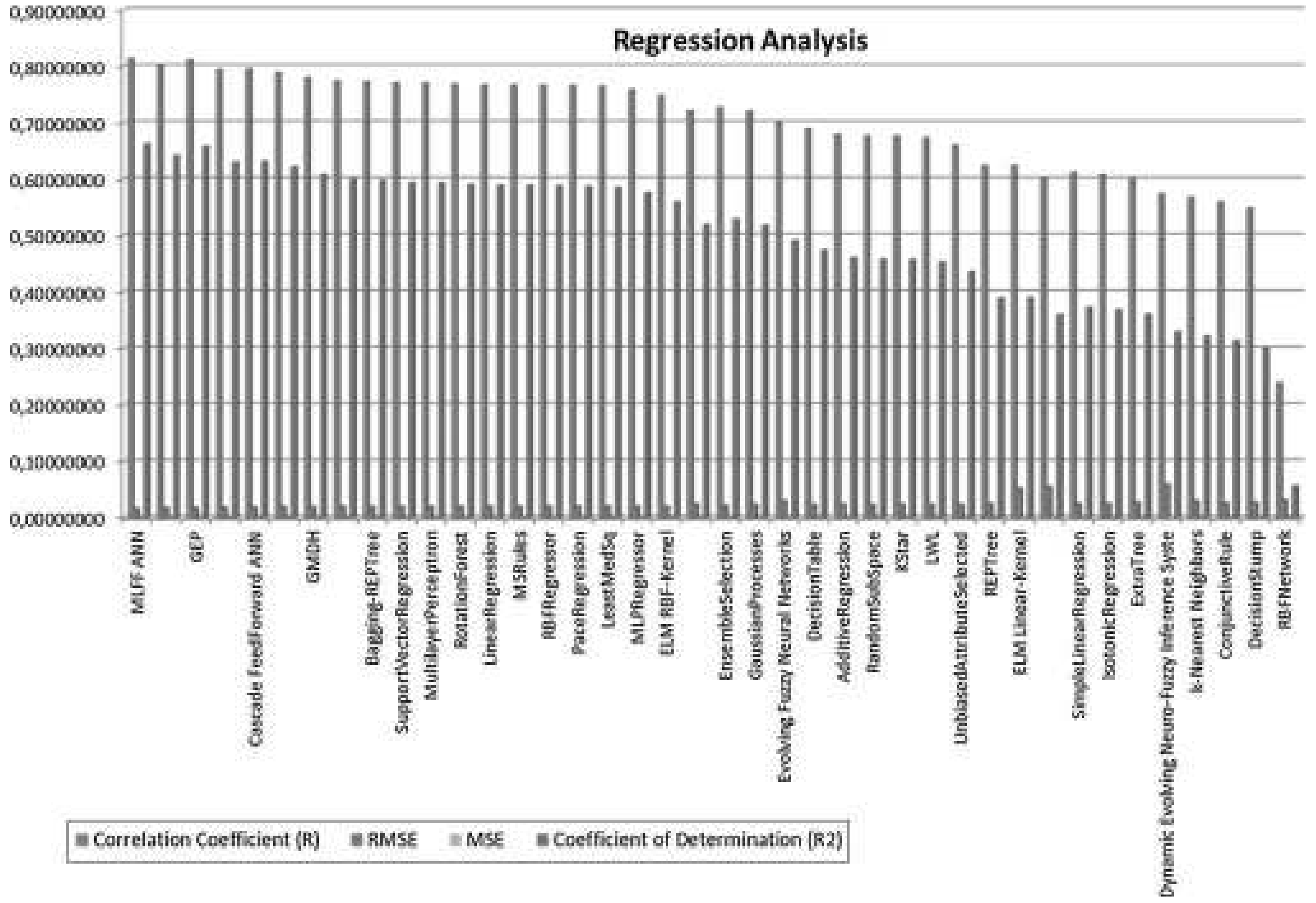

Fig. 12 Comparison of all evaluation indices (*note* MLFF ANN has the best performance)

informed selections with minimum efforts. ANNs have been developed for assessing wood density from Douglas fir progeny test data [26]. While successful, with results better than that obtained from the classical parametric multiple regression approach, the developed predictive models were of lower confidence to support selection decision in a breeding program framework. In the present study, we tested a multitude of machine learning approaches to identifying those with higher predicative ability. As shown in Figs. 10, 11, and 12, the multilayer feed-forward (MLFF) ANNs [19, 21, 25] and GEP [14] provided the highest predictability as compared to the other methods tested. We therefore recommend the use of these machine learning approaches to predict wood density for selection and/or other forest management decisions. These methods rely on existing data, and their application on other populations within the studied species requires further investigation to permit generalization.

Acknowledgments Thanks to Irena Fundova and Tomas Funda for data collection and Barry Jaquish for access to progeny test sites. Funds from the Natural Sciences and Engineering Research Council of Canada's Discovery and IRC grants, FPInnovations, and the Johnson's Family Forest Biotechnology Endowment to YAE are highly appreciated.

References

1. Allard RW (1960) Principles of plant breeding. Wiley, New York
2. American Society for Testing and Materials (ASTM) (1985) Standard test methods for specific gravity of wood and wood-based materials. American Society for Testing and Materials, Philadelphia. ASTM D 2395-02
3. Anastasakis L, Mort N (2009) Exchange rate forecasting using a combined parametric and nonparametric self-organising modelling approach. Expert Syst Appl 36:12001–12011
4. Andrews M (2002) Wood quality measurement-son et lumière. N Z J For Sci 47:19–21
5. Bouffier L, Raffin A, Rozenberg P, Meredieu C, Kremer A (2008) What are the consequences of growth selection on wood density in the French maritime pine breeding programme? Tree Genet Genomes 5:11–25
6. Carter P, Briggs D, Ross RJ, Wang X (2005) Acoustic testing to enhance western forest values and meet customer wood quality needs. In: Harrington CA, Schoenholtz SH (eds) Productivity of western forests: a forest products focus. Gen. Tech. Rep. PNW-GTR-642. U.S. Department of Agriculture, Forest Service, Pacific Northwest Research Station, Portland, pp 121–129

7. Chantre G, Rozenberg P (1997) Can drill resistance profiles (Resistograph) lead to within-profile and within-ring density parameters in Douglas fir wood? In: Zhang SY, Gosselin R, Chauret G (eds) Proceedings of CTIA-IUFRO international wood quality workshop: timber management toward wood quality and end-product values. Forintek Canada, Sainte-Foy, Quebec, Canada, pp 41–47
8. Cown DJ (1978) Comparison of the pilodyn and torsiometer methods for the rapid assessment of wood density in living trees. N Z J For Sci 8:384–391
9. Cown DJ, Clement BC (1983) A wood densitometer using direct scanning with X-rays. Wood Sci Technol 17:91–99
10. Deng N, Tian Y, Zhang C (2012) Support vector machines: optimization based theory, algorithms, and extensions. Chapman & Hall/CRC press data mining and knowledge discovery series. ISBN 9781439857922
11. El-Kassaby YA, Mansfield SD, Isik F, Stoehr M (2011) In situ wood quality assessment in Douglas-fir. Tree Genet Genomes 7:553–561
12. Falconer DS, Mackay TFC (1996) Introduction to quantitative genetics. Longman, New York
13. Farlow SJ (1984) Self-organizing methods in modelling: GMDH type algorithms. Marcel Decker Inc., New York
14. Ferreira C (2006) Gene expression programming: mathematical modeling by an artificial intelligence, 2nd edn. Springer, Berlin. ISBN 3540327967
15. Freedman DA (2005) Statistical models: theory and practice. Cambridge University Press, Cambridge
16. Gianola D, Okut H, Weigel K, Rosa G (2011) Predicting complex quantitative traits with Bayesian neural networks: a case study with Jersey cows and wheat. BMC Genet 12:87
17. Goyal S, Goyal GK (2011) Cascade and feed forward back propagation artificial neural network models for prediction of sensory quality of instant coffee flavoured sterilized drink. Can J Artif Intell Mach Learn Pattern Recognit 2:78–82
18. Gurney K (1997) An introduction to neural networks. Taylor and Francis Group Inc., London
19. Hagan MT, Demuth HB, Beale M (1996) Neural network design. PWS Publishing Company, Boston
20. Hanrahan G (2011) Artificial neural networks in biological and environmental analysis. CRC Press Inc., Boca Raton
21. Haykin S (1994) Neural networks: a comprehensive foundation. Macmillan Publishing Company, New York
22. Holmes G, Donkin A, Witten IH (1994) Weka: a machine learning workbench. In: Proceedings of 2nd Australia an New Zealand conference on intelligent information systems, Brisbane, Australia
23. Huang G-B (2014) An insight into extreme learning machines: random neurons, random features and kernels. Cognit Comput 6:376–390
24. Huang L, Song Q, Kasabov N (2008) Evolving connectionist system based role allocation for robotic soccer. Int J Adv Rob Syst 5:59–62
25. Iliadis L (2008) Intelligent information systems and applications in risk estimation. Stamoulis publication, Thessaloniki
26. Iliadis L, Mansfield SD, Avramidis S, El-Kassaby YA (2013) Predicting Douglas-fir wood density by artificial neural networks (ANN) based on progeny testing information. Holzforschung 67:771–777
27. Isik F, Li B (2003) Rapid assessment of wood density of live trees using the Resistograph for selection in tree improvement programs. Can J For Res 33:2426–2435
28. Ivakhnenko AG (1971) Polynomial theory of complex systems. IEEE Trans Syst Man Cybern 1:364–378
29. Kasabov N (2001) Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge-based learning. IEEE Trans Cybern 31:902–918
30. Kasabov N (2002) Evolving connectionist systems: methods and applications in bioinformatics, brain study and intelligent machines. Springer, New York
31. Kasabov N, Song Q (2002) DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Trans Fuzzy Syst 10:144–154
32. Kiss GK, Yanchuk AD (1991) Preliminary evaluation of genetic variation of weevil resistance in interior spruce in British Columbia. Can J For Res 21:230–234
33. Koshulko OA, Koshulko GA (2011) Validation strategy selection in combinatorial and multilayered iterative GMDH algorithms. In: Proceedings of 4th international workshop on inductive modelling, Kyiv, Ukraine, pp 51–54
34. Kriesel D (2007) A brief introduction to neural networks. http://www.dkriesel.com
35. Lehmann EL, Casella G (1998) Theory of point estimation, 2nd edn. Springer, New York
36. Madala HR, Ivakhnenko AG (1994) Inductive learning algorithms for complex systems modeling. CRC Press, Boca Raton
37. Marquardt D (1963) An algorithm for least-squares estimation of nonlinear parameters. SIAM J Appl Math 11:431–441
38. El-Dien OG, Ratcliffe B, Klápště J, Chen C, Porth I, El-Kassaby YA (2015) Prediction accuracies for growth and wood attributes of interior spruce in space using genotyping-by-sequencing. BMC Genomics 16:370. doi:10.1186/s12864-015-1597-y
39. Namkoong G, Kang HC, Brouard JS (1988) Tree breeding: principles and strategies. Monographs on theoretical and applied genetics 11. Springer, New York, p 11
40. Oh S-K, Pedrycz W (2002) The design of self-organizing polynomial neural networks. Inf Sci 141:237–258
41. Okut H, Wu X-L, Rosa GJM, Bauck S, Woodward BW, Schnabel RD, Taylor JF, Gianola D (2013) Predicting expected progeny difference for marbling score in 43 Angus cattle using artificial neural networks and Bayesian regression models. Genet Sel Evol 45:34
42. Pereira BDB, Rao CR (2009) Data mining using neural networks: a guide for statisticians. http://www.po.ufrj.br/basilio/publicacoes/livros/2009_datamining_Using_neural_networks.pdf
43. Ratcliffe B, Hart FJ, Klápšte J, Jaquish B, Mansfield SD, El-Kassaby YA (2014) Genetics of wood quality attributes in western larch. Ann For Sci 71:415–424
44. Rinn F, Scheweingruber FH, Schar E (1996) Resistograph and X-ray density charts of wood comparative evaluation of drill resistance profiles and X-ray density charts of different wood species. Holzforschung 50:303–311
45. Rodriguez JJ, Kuncheva L, Alonso CJ (2006) Rotation forest: a new classifier ensemble method. IEEE Trans Pattern Anal Mach Intell 28(10):1619–1630
46. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by backpropagating errors. Nature 323:533–536
47. Song Q, Kasabov N (2003) Weighted data normalization and feature selection. In: Proceedings of 8th intelligence information systems conference. Australia & N.Z, pp 87–92
48. Sutton BCS, Flanagan DJ, Gawley JR, Newton CH, Lester DT, El-Kassaby YA (1991) Inheritance of chloroplast and mitochondrial-DNA in Picea and composition of hybrids from introgression zones. Theor Appl Genet 82:242–248
49. The Mathworks Inc (2005) MATLAB: the language of technical computing, version 7.1.0.246 (R14) service pack 3. The MathWorks Inc., Natick
50. Watts MJ (2009) A decade of Kasabov's evolving connectionist systems: a review. IEEE Trans Syst Man Cybern Part C Appl Rev 39:253–269
51. White TL, Adams WT, Neale DB (2007) Forest genetics. CABI, Oxford

52. Winistorfer PM, Xli W, Wimmer R (1995) Application of drill resistance technique for density profile measurement in wood composite panels. For Prod J 45:50–53

53. Witten IH, Frank E (2011) Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann. ISBN 978-0-12-374856-0

54. Wu HX, Matheson AC (2004) General and specific combining ability from partial diallels of radiata pine: implications for utility of SCA in breeding and deployment populations. Theor Appl Genet 108:1503–1512

55. Yanchuk AD (1996) General and specific combining ability from disconnected partial diallels of coastal Douglas-fir. Silvae Genet 45:37–45

56. Zwillinger D, Kokoska S (2000) CRC standard probability and statistics tables and formulae. CRC Press, Boca Raton. ISBN 1-58488-059-7