



Application of artificial intelligence models and optimization algorithms in plant cell and tissue culture

Mohsen Hesami¹ · Andrew Maxwell Phineas Jones¹

Received: 29 June 2020 / Revised: 31 August 2020 / Accepted: 4 September 2020 / Published online: 28 September 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

Artificial intelligence (AI) models and optimization algorithms (OA) are broadly employed in different fields of technology and science and have recently been applied to improve different stages of plant tissue culture. The usefulness of the application of AI-OA has been demonstrated in the prediction and optimization of length and number of microshoots or roots, biomass in plant cell cultures or hairy root culture, and optimization of environmental conditions to achieve maximum productivity and efficiency, as well as classification of microshoots and somatic embryos. Despite its potential, the use of AI and OA in this field has been limited due to complex definition terms and computational algorithms. Therefore, a systematic review to unravel modeling and optimizing methods is important for plant researchers and has been acknowledged in this study. First, the main steps for AI-OA development (from data selection to evaluation of prediction and classification models), as well as several AI models such as artificial neural networks (ANNs), neurofuzzy logic, support vector machines (SVMs), decision trees, random forest (FR), and genetic algorithms (GA), have been represented. Then, the application of AI-OA models in different steps of plant tissue culture has been discussed and highlighted. This review also points out limitations in the application of AI-OA in different plant tissue culture processes and provides a new view for future study objectives.

Key points

- *Artificial intelligence models and optimization algorithms can be considered a novel and reliable computational method in plant tissue culture.*
- *This review provides the main steps and concepts for model development.*
- *The application of machine learning algorithms in different steps of plant tissue culture has been discussed and highlighted.*

Keywords In vitro culture · Data-driven model · Machine learning algorithm · Plant biotechnology · Computational approach · Shoot proliferation · Organogenesis · Embryogenesis · Androgenesis · Rhizogenesis

Introduction

Plant tissue culture can be considered the culture or cultivation of specific plant cells, organs, or tissues (explant) under axenic conditions, which is based on “totipotency” (Bhojwani and Dantu 2013). The term totipotency means that all plant cells (except male and female gametes) include the full range of genes, which makes it theoretically possible for individual cells under in vitro

condition to develop into healthy and true-to-type plants (Bhojwani and Dantu 2013). This process provides the foundation for “micropropagation” in which culture vessels are used for propagation from various explants. Nowadays, in vitro culture can be considered one of the most important methodologies for the breeding and propagation of many plant species. Without in vitro culture, different methods such as micropropagation, in vitro shoot regeneration, gynogenesis, androgenesis, the production of plant-derived metabolites, or somatic embryogenesis would not be achievable (Raj and Saudagar 2019; Hesami et al. 2018c). However, it is necessary to optimize the in vitro culture conditions (Fig. 1) for each species, and in some cases each genotype within a species, and for different stages of growth and development such as callogenesis, embryogenesis, shooting, and rooting (Gray and Trigiano 2018). For instance, the

✉ Andrew Maxwell Phineas Jones
amjones@uoguelph.ca

¹ Gosling Research Institute for Plant Preservation, Department of Plant Agriculture, University of Guelph, Guelph, ON N1G 2W1, Canada

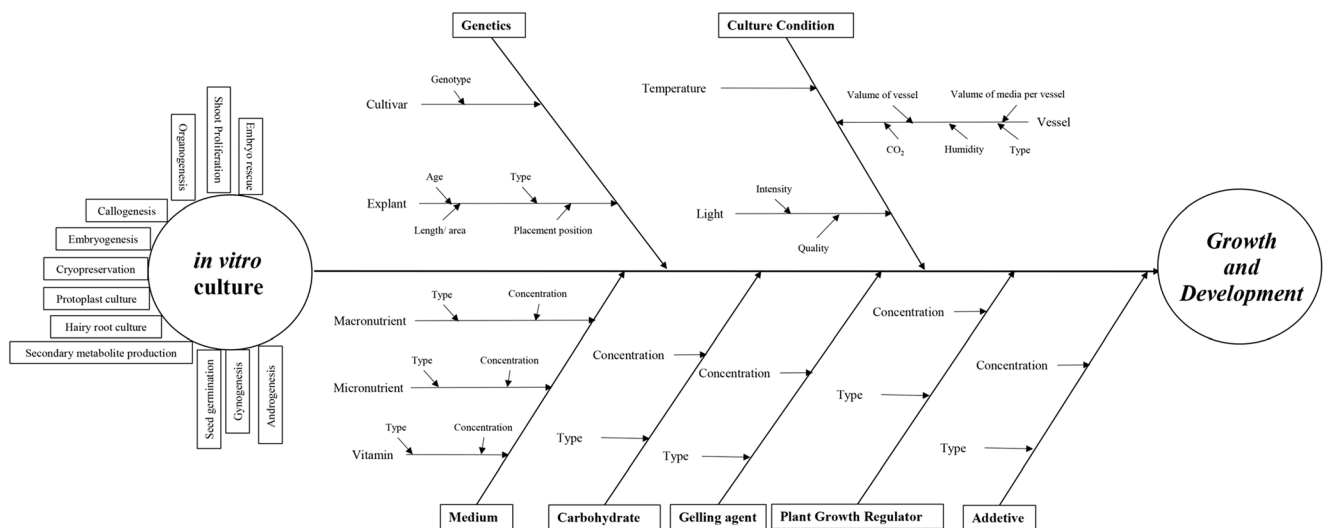


Fig. 1 The schematic diagram of the major factors affecting *in vitro* culture processes

composition and concentrations of macro and micro nutrients, vitamins, and amino acids have a profound effect on organogenesis as reported in several studies with different plants. Although a plethora of studies have used MS (Murashige and Skoog 1962) salts as a basal medium for organogenesis in different plants, the composition of MS medium is based on analysis of tissue ashes of tobacco (Arab et al. 2018; Jamshidi et al. 2019; Akin et al. 2020; Nezami-Alanagh et al. 2018). Since the nutrient requirements for different tissue culture systems and plant species vary, it is necessary to develop medium formulations optimized for specific species and stages of development for maximal efficiency. However, due to the large number of media components, design and modification of a medium for specific purpose needs high expertise and is time-consuming (Phillips and Garda 2019). Hildebrandt et al. (1946) indicated that more than 16000 different treatments were required for designing a new culture medium. Also, Murashige and Skoog (1962) spent about five years to establish and develop the culture medium by using eighty-one different combinations of macro- and micro-elements and vitamins. To ease this problem, computer technologies such as artificial intelligence (AI) would be helpful to reduce this long and cumbersome process.

Although there are numerous biological events that can readily be observed in different stages of *in vitro* culture, all of them are non-linear and non-deterministic and, furthermore, are impacted by multiple other factors as well (Osama et al. 2015; Zielinska and Kepczynska 2013). The complex interactions of many factors make optimization using traditional statistics problematic and would require unrealistic numbers of treatments. Therefore, the application of the appropriate AI models can be considered a useful and precise methodology to simulate and predict different growth and developmental processes under *in vitro* conditions to help optimize protocols with fewer treatments.

Recently, there has been an increase in plant tissue culture modeling using data-driven (also known as machine learning or AI) models (Prasad and Gupta 2008a; Osama et al. 2015; García-Pérez et al. 2020a; Kaur et al. 2020). AI models include various designs that might cover different views to *in vitro* processes (Tani et al. 1992). Modeling different steps of plant tissue culture is one of the most remarkable challenges in the field of *in vitro* culture (Frossyniotis et al. 2008). This rise is a consequence of the physical complexity of plant tissue culture and the time and cost needed for analyzing different elements of the *in vitro* culture process (Shiotani et al. 1994; Molto and Harrell 1993; Hesami et al. 2017c). AI models have been found to be very applicable and reliable methods to help cope with those challenges and problems by providing opportunities to construct AI models from experimental and observed data and, also, improving the response of decision-makers facing complex systems in plant tissue culture (Osama et al. 2013; Araghinejad et al. 2017; Hesami et al. 2020b). Since AI tools are able to model different *in vitro* systems and subsequent outcomes of biological processes without the requirement for a profound knowledge of the physical systems neighboring the process, these methods are becoming common among plant tissue culture researchers (Osama et al. 2015; Zielinska and Kepczynska 2013).

Data-driven models, as understood by the name, refer to various methods that predict and model a process based on real information obtained from the process (Maddalena et al. 2020; Moravej et al. 2020; Dezfooli et al. 2018). They consist of various models, generally classified into soft computing and statistical methods (Akbari and Deligani 2020; Ebrahimian et al. 2020). Data-driven models can be generally categorized as accurate, inexpensive, precise, and flexible methods, which make them appropriate approaches for predicting and studying different biological systems with

various complexity degrees according to our knowledge about a process (Araghinejad et al. 2017; Hosseini-Moghari et al. 2017). Although data-driven models were primarily used in different fields of science and technology, they could be considered novel methods regarding soft computing (Maddalena et al. 2020; Araghinejad et al. 2018). Modeling through AI models, which is dependent on the levels of our knowledge of mathematical and statistical equations, can be described as a solution determined by “engineering thinking and judgment” to the field of computational biology.

Recently, several reports have been published regarding various applications of artificial intelligence models in different procedures of plant tissue culture (Hesami et al. 2019b; Zielinska and Kepczynska 2013; Osama et al. 2015). All the reports confirmed the reliability and accuracy of AI in forecasting and optimizing growth and developmental processes under *in vitro* culture conditions. Thus, this review focuses on the recent developments in plant tissue culture using AI-OA. First, we have introduced the principle of modeling and optimizing as well as different well-known algorithms. Then, the application of AI-OA in different stages of plant tissue culture has been discussed.

Data-driven modeling basics

Preprocessing of data: first step before modeling

Although preprocessing data before using AI models is not required, it can sometimes improve the performance and accuracy of the models (Silva et al. 2019). Indeed, data preprocessing can guarantee that all data receive equivalent consideration during the training set. Two common preprocessing methods that can be applied are principle component analysis (PCA) and standardizing the data (Araghinejad et al. 2017). These approaches are described in this section.

Principal component analysis

The principal component analysis (PCA) is an approach that can be used for two purposes. The first is to eliminate the linear correlations among variables and the second is to decrease the data dimension. PCA replaces correlated variables by principal components (new uncorrelated variables). PCA uses a new orthogonal coordinate set to replace xy (Cartesian) coordinate set, where the first line crosses via the data scatter axis and the novel origin. The new coordinate process has merit over the previous version of the coordinate system that the first axis can be utilized to explain most of the variance, while the second axis provides only a little description. Therefore, the reduction of data dimension can be obtained by decreasing the second axis without missing much data

when a significant correlation is available (Ait-Sahalia and Xiu 2019).

Linear combinations of the n vector of correlated vectors are shown by PCs and X_i . The PC number is equal to n , so, the full variance of the dataset is obtained through all PCs together. The first few PCs explain most of the variance, therefore, some PCs which represent a little of the variance can be ignored. n PCs are determined based on the following equations:

$$\begin{aligned} PC_1 &= a_{11}X_1 + a_{12}X_2 + \dots + a_{1n}X_n \\ PC_2 &= a_{21}X_1 + a_{22}X_2 + \dots + a_{2n}X_n \\ &\vdots \\ PC_n &= a_{n1}X_1 + a_{n2}X_2 + \dots + a_{nn}X_n \end{aligned} \quad (1)$$

Figure 2 represents PCA for a two-dimensional (2D) dataset.

Standardizing

Data standardization as a simple method can improve the model efficiency. In this method, the specified range (usually between zero and one) is used for transmitting all variables. Data standardization prevents the negative effects of input variables with various ranges on the model efficiency. The following equation is one of the most common equations for the data standardization:

$$x_{normal} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2)$$

where x_{min} , x_{max} , and x_{normal} are the minimum, maximum, and normalized values of x , respectively. By using this equation for each of the output and input vectors, all variables are transmitted between 0 and 1 (Kumari and Swarnkar 2020).

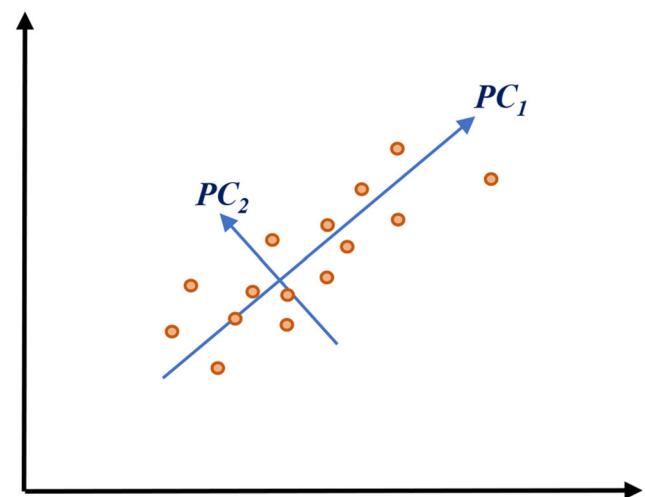


Fig. 2 An example of principal component analysis (PCA) for a two-dimensional data set

Network selection

Network selection involves choosing the network, initial weight matrix, size and number of hidden layers, etc. (Osama et al. 2015).

Training selection

Training selection should be started with initial weight and network topology and training the network on the training dataset. After reaching the satisfactory minimum error, the weights will be saved (Silva et al. 2019).

Testing and interpretation of results

The trained network is employed to test the dataset to obtain the error. If it is not satisfactory, the network architecture or training set requires to be modified (Silva et al. 2019).

Assessment of the developed model

The concluding assessment of the network quality is addressed after the training processes completed through testing datasets based on different performance criteria such as the sum of the squares of the error (SSE), mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), mean bias error (MBE), the linear Pearson's correlation coefficient (R), and coefficient of determination (R^2) (Silva et al. 2019; Osama et al. 2015).

$$SSE = 1/n \sum_{i=1}^n (y_i - \bar{y})^2 \quad (3)$$

$$MSE = 1/n \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4)$$

$$RMSE = \sqrt{\left(\sum_{i=1}^n (y_i - \hat{y}_i)^2 \right) / n} \quad (5)$$

$$MAE = 1/n \sum_{i=1}^n |y_i - \hat{y}_i| \quad (6)$$

$$MBE = 1/n \sum_{i=1}^n (y_i - \hat{y}_i) \quad (7)$$

$$R = \frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2} \sqrt{\sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}} \quad (8)$$

$$R^2 = \left[\frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2} \sqrt{\sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2}} \right]^2 \quad (9)$$

where y_i presents the i^{th} observed data, \bar{y} shows the mean of observed values, $\bar{\hat{y}}$ indicates the mean of predicted values, and n is the total number of predicted values.

Artificial intelligence models

Artificial neural networks

Different kinds of artificial neural networks (ANNs) including multilayer perceptron (MLP), generalized regression neural network (GRNN), radial basis function (RBF), and probabilistic neural network (PNN) are described in this section. Before introducing these ANNs, some terms should be defined:

Neuron: is the main unit of ANN, which based on a specific input variable and applying a transfer function, provides an appropriate response.

Architecture: is a network construction consists of input, hidden and output layers, number of neurons in each layer, the way of neuron connection, the flow of data (recurrent or straight), and specific transfer functions.

Train network: is the process of calibrating the ANN through input/output pairs.

Multilayer perceptron

The MLP as one of the most well-known ANNs includes one or more hidden layers, an input layer, and an output layer (Silva et al. 2019; Osama et al. 2015; Sheikhi et al. 2020) (Fig. 3a). A supervised training procedure is implemented by MLP that provides input and output variables to the network; the training set continues until the following equation would be minimized:

$$E = \frac{1}{K} \sum_{k=1}^K (y_k - \hat{y}_k)^2 \quad (10)$$

where K , y_k , and \hat{y}_k are the number of datapoints, the k th observed data, and the k th forecasted data. In a three-layer MLP with n inputs and m neurons in the hidden layer \hat{y} determined as:

$$\hat{y} = f \left[\sum_{j=1}^m w_j \cdot g \left(\sum_{i=1}^n w_{ji} x_i + w_{j0} \right) + w_o \right] \quad (11)$$

where x_i is the i th input variable, w_o represents bias related to the neuron of output, w_{j0} is bias of the j th neuron of the hidden layer, f represents transfer functions for the output layer, g is the transfer functions for hidden layer, w_{ji} is the weight connecting the j th neuron of hidden layer and the i th input variable, and w_j represents weight linking the neuron of output layer and the j th neuron of the hidden layer. Some of the well-known transfer functions are presented in Table 1.

Determining the construction of the MLP has the main function in its performance (Domingues et al. 2020; Niazian et al. 2018a). In the construction of this model, it is necessary to determine the number of neurons in each layer and the number of hidden layers (Niazian et al. 2018a). Hornik et al. (1989) revealed that the MLP with a sigmoid transfer function is general approximators; which shows that it could be trained to build any construction between the input and output variables. Therefore, the number of neurons in the hidden layer plays an important role in determining the construction of the MLP. Some studies (Sheikhi et al. 2020; Niazian et al. 2018a) have recommended the proper number of neurons (m) based on the number of data (K) or the number of input (n). For example, Tang and Fishwick (1993), Wong (1991), and Wanas et al. (1998) suggested “ n ,” “ $2n$,” and “ $\log(K)$ ” as the suitable neuron number. Eventually, the optimal number of neurons in the hidden layer should be calculated by using trial and error, however, the reported offers could be employed as an initiating point. A large number of neurons contributes to the complexity of the network while a low number of them makes for simplicity of the network, therefore should be noted that a too simple network results in underfitting, and, conversely, becoming too complex causes overfitting (Domingues et al. 2020; Hosseini-Moghari and Araghinejad 2015).

Radial basis function

RBF is a three-layer ANN consisting of an input layer, a hidden layer, and an output layer (Fig. 3b). This is the basis and principal for radial basis networks, which organizes statistical ANNs. Statistical ANNs refer to networks which in contrast to the traditional ANNs implement regression-based approaches and have not been emulated by the biological neural networks (Lin et al. 2020). In an RBF model, Euclidean distance between the center of each neuron and the input is considered an input of transfer function for that neuron. The most well-known transfer function in RBF is the Gaussian function, which is determined based on the following equation:

$$f(X_r, X_b) = e^{-[\|X_r - X_b\| * 0.8326/h]^2} \tag{12}$$

Table 1 The most well-known transfer functions

Name	Function
Linear	$f(x) = x$
Tangent sigmoid	$f(x) = \left(\frac{2}{1+e^{-ax}}\right) - 1 \quad \alpha > 0$
Log sigmoid	$f(x) = \frac{1}{1+e^{-ax}} \quad \alpha > 0$
Radial basis	$f(x) = e^{-\frac{x^2}{\sigma^2}}$

where X_r, X_b , and h are input with unknown output, observed inputs in time b , and spread, respectively. The output of the function close to 1 when $\|X_r - X_b\|$ approaches 0 and close to 0 when $\|X_r - X_b\|$ approaches a large value. Finally, the dependent variable (Y_r) by predictor X_r is determined as follows:

$$Y_r = \sum_{b=1}^m w_b * f(X_r, X_b) + w_0 \tag{13}$$

where w_0 and w_j are the bias and weight of linkage between the b th hidden layer and the output layer, respectively.

Generalized regression neural network

GRNN introduced by Specht (1991) is another kind of statistical ANNs with a very fast training process. In the GRNN model, the number of observed data and the number of neurons in the hidden layer are equal. This model consists of an input layer, pattern layer, summation layer, and output layer (Fig. 3c). The pattern layer is completely connected to the input layer. D -summation and S -summation neurons of the summation layer are connected to the output derived from each neuron of the pattern layer. D -summation and S -summation neurons calculate the sum of the unweighted and weighted of the pattern layer, respectively. The connection weight between S -summation neuron and a neuron of the pattern layer is equal to the target output, while the connection weight for D -summation is unity. The output layer obtains the unknown value of output corresponding to the input vector, only via dividing the output of each S -summation neuron through the output of each D -summation neuron (Lan et al. 2020). Consequently, the following equation is used to determine the output value:

$$Y_r = \frac{\sum_{b=1}^m T_b * f(X_r, X_b)}{\sum_{b=1}^m f(X_r, X_b)} \tag{14}$$

where Y_r represents the output value, and T_b is target associated with the b th observed data.

Probabilistic neural network

The probabilistic neural network (PNN) is a type of ANNs for classification aims. This model has a construction similar to that of the RBF model (Fig. 3d). When an input is provided, the first layer calculates distances from the calibration input vectors to the input vector and generates a probabilities vector as $f(X_r, X_b)$. In the last layer, the maximum of these probabilities is picked by a complete transfer function in the output. PNN is not a regression model and cannot be forecasted continuous data, therefore, it can be employed as a method to qualitative predicting (Ying et al. 2020).

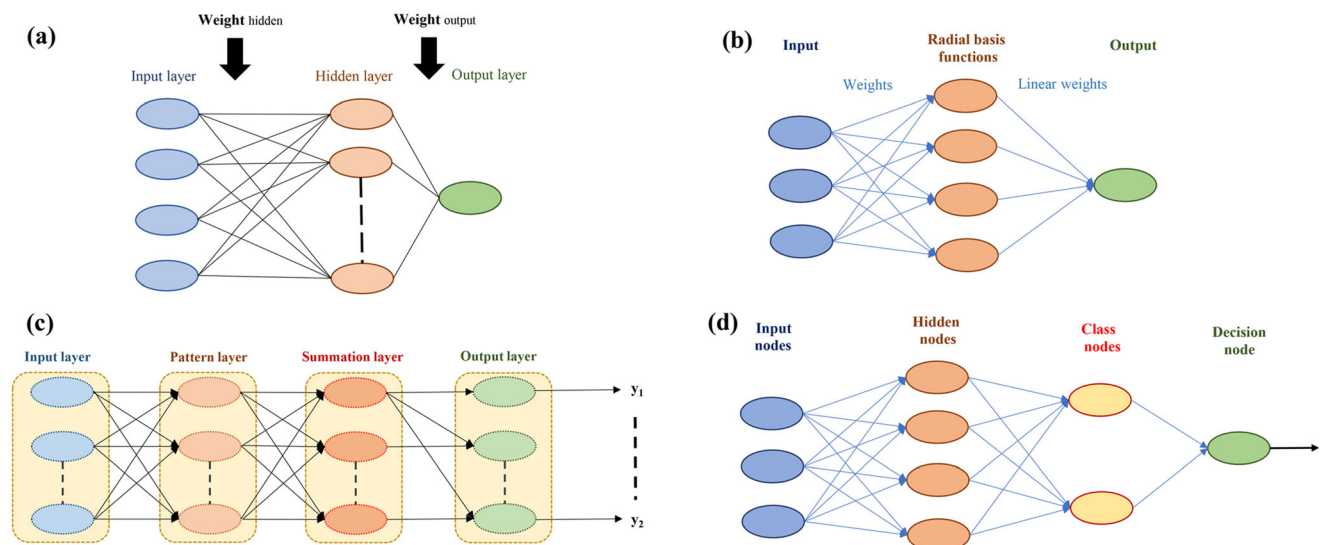


Fig. 3 The schematic diagram of different artificial neural networks (ANNs) including **a** multilayer perceptron, **b** radial basis function, **c** generalized regression neural network, and **d** probabilistic neural network

Neurofuzzy logic

Neurofuzzy logic is the result of the combination of the adaptive learning capabilities of ANNs and fuzzy logic (Dorzhigulov and James 2020). Designing neurofuzzy logic needs qualitative knowledge, but not quantitative. Neurofuzzy logic produces certain rules in an understandable and clear form: if there is a requirement, then there is a decision; therefore, demonstrating the relationships in observations. ANN models are applied to find the optimal levels of certain fuzzy logic parameters in neurofuzzy system and automatically determine fuzzy linguistic terms (rules) from numerical variables. The simplest and the most well-known method for constructing such models is to develop linguistic terms and membership functions by following these functions and then to review how this model operates. The network structure employed in neurofuzzy model consists of inference, fuzzyfication, and defuzzyfication facilities. The classic construction of the fuzzy logic indicates the description of operations conducted at each stage: (i) fuzzyfication, involving in defining the membership level of a special input variable to the size of each fuzzy sets comprising the possible ranges of inputs—this step is decreased to estimating functions or finding suitable variables in the tables, (ii) the application of fuzzy system indicators to define the degree to which a condition is reached in each of the functions, (iii) the use of the implication model, which causes the production of fuzzy sets representing each of the output variables happening in the conclusion, (iv) gathering all combinations of outputs for each of the representing outputs and all functions in one set of fuzzy, and (v) defuzzification, which involves in the specific value assigned to each of the outputs of the fuzzy set taken after gathering (Dorzhigulov and James 2020). Recently, several

neuro-fuzzy approaches such as ASMOD (adaptive spline modeling of data), ANFIS (adaptive neuro-fuzzy inference systems), and NEFPROX (neuro-fuzzy systems for function approximation) have been developed and implemented. Among them, the ANFIS model gives a directed and systematic strategy for modeling and generates the best design parameters in the minimum time (Prado et al. 2020).

ANFIS can explain the complex system behavior according to the fuzzy if–then rules which are based on Sugeno fuzzy inference system. Consider the fuzzy inference system with x and y as input variables and z as an output. A typical ruleset, for the first order Sugeno fuzzy model, with four fuzzy if–then rules can be represented as:

$$\begin{aligned}
 \text{Rule 1 : } & \text{if } x \text{ is } A_1 \text{ and } y \text{ is } B_1 \text{ then } z_1 = p_1x + q_1y + r_1 \\
 \text{Rule 2 : } & \text{if } x \text{ is } A_1 \text{ and } y \text{ is } B_2 \text{ then } z_2 = p_2x + q_2y + r_2 \\
 \text{Rule 3 : } & \text{if } x \text{ is } A_2 \text{ and } y \text{ is } B_1 \text{ then } z_3 = p_3x + q_3y + r_3 \\
 \text{Rule 4 : } & \text{if } x \text{ is } A_2 \text{ and } y \text{ is } B_2 \text{ then } z_4 = p_4x + q_4y + r_4
 \end{aligned}
 \tag{15}$$

where A_i and B_i ($i = 1, 2, 3, 4$) are the fuzzy sets, p_i, q_i , and r_i ($i = 1, 2, 3, 4$) are the design parameters that are calculated during the training set. The construction of ANFIS includes input node layer, rule node layer, average node layer, consequent node layer, and output node layer (Fig. 4a).

Layer 1 (input node layer): All the nodes in layer 1 are adaptive nodes. The outputs of the input node layer are the fuzzy membership grade of the input variables, which are determined by:

$$\begin{aligned}
 O_i^1 &= \mu_{A_i}(x) \quad i = 1, 2. \\
 O_i^1 &= \mu_{B_{i-2}}(x) \quad i = 3, 4.
 \end{aligned}
 \tag{16}$$

where μ and O_i^1 are membership functions and output from

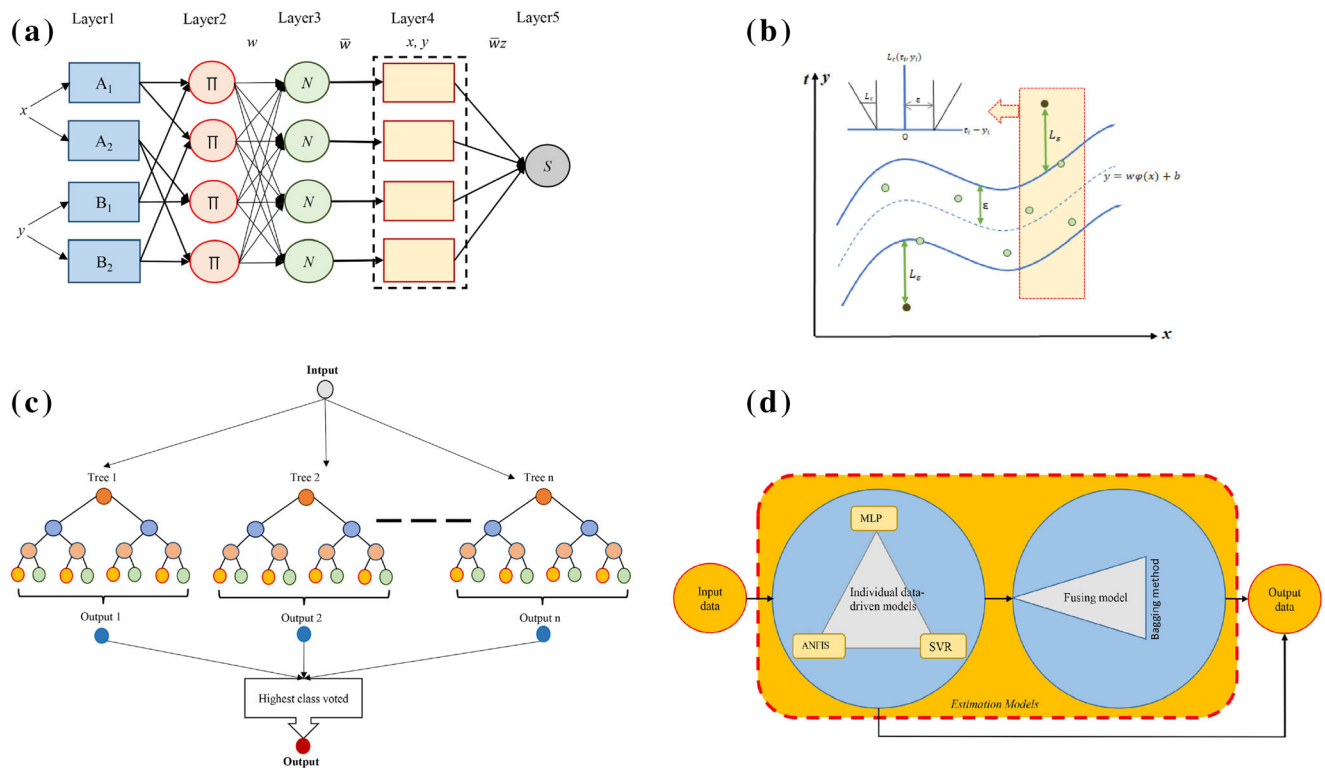


Fig. 4 The schematic view of **a** Adaptive Neuro-Fuzzy Inference Systems (ANFIS) model for a two-input Sugeno model with four rules, **b** support vector regression (SVR), **c** random forest (RF), and **d** data fusion approach with three individual models including MLP, ANFIS, and SVR

node i , respectively.

Layer 2 (rule node layer): In the second layer, the nodes are considered fixed nodes and labeled Π , the AND operator is implemented to achieve one output that displays the result of the prior for that rule. The k^{th} output of Layer 2 (w_k) is calculated as:

$$O_k^2 = w_k = \mu_{A_i}(x)\mu_{B_j}(x) \quad i = 1, 2. \quad j = 1, 2. \quad k = 1, 2, 3, 4. \quad (17)$$

which elucidates the firing strength of each rule. The firing strength indicates the level to which the prior section of the rule is satisfied, and it develops the function of the output for the rule.

Layer 3 (average node layer): In the third layer, the nodes are fixed nodes and labelled N . The normalization of firing strengths from the prior layer is the main function of layer 3. The normalized firing strengths are the outputs of the average node layer (\bar{w}_i), and can be determined as:

$$O_i^3 = \bar{w}_i = \frac{w_i}{\sum_{i=1}^4 w_k} \quad i = 1, 2, 3, 4. \quad (18)$$

Layer 4 (consequent node layer): This layer determines the contribution of each i th rule in the whole output. The product of the first-order Sugeno model and the normalized firing

strength is the output of each node in the fourth layer. Therefore, the outputs of the fourth layer can be shown as:

$$O_i^4 = \bar{w}_i z_i = \bar{w}_i (p_i x + q_i y + r_i) \quad i = 1, 2, 3, 4. \quad (19)$$

Layer 5 (output node layer): One single fixed node labeled S is the only node in the fifth layer. Summing all incoming signals is the task of this layer. As a result, the following equation represents the final output of the ANFIS:

$$O_i^5 = \sum_{i=1}^4 \bar{w}_i z_i \quad (20)$$

As can be seen, two sets of parameters should be determined and adjusted. The first set is premise parameters that are based on the input membership functions. The second set is consequent parameters $\{p, q, r\}$ which are based on the first-order Sugeno model. The least-squares approach is implemented to adjust the consequent parameters and backpropagation algorithm is used to optimize premise parameters. It has been documented that high performance in training the ANFIS can be obtained by a hybrid algorithm (Prado et al. 2020).

Support vector machine

Support vector machines (SVMs) developed by Vapnik (2013) are the types of AI models with supervised and

unsupervised learning that are utilized for classification, clustering, and regression analysis. Two types of SVM including support vector classification (SVC) and support vector regression (SVR) have been introduced in this section.

Support vector classification

SVC is a classification version of SVM; therefore, it can only be applied for qualitative predicting. SVC minimizes the risk of classification by dividing the decision space, which means that two groups have the highest distance from both lines. This means that among different separators, the separator is selected to have the highest distance of all groups. Figure 5 represents the differences between SVC classification and ANN classification methods. In Fig. 5, *a* and *b* are separators with error = 0; however, if a new input is added, these lines may lose their accuracy, while line *c* has the minimum risk for losing its precision. The following equation shows line *c*:

$$w^T x + b = 0 \tag{21}$$

Where *w* and *b* are the classifier parameters and *x* is the variables in the decision space. As can be seen in Fig. 5, for classification, a margin is considered by SVC which determines as follow:

$$w^T x + b = 1 \tag{22}$$

$$w^T x + b = -1 \tag{23}$$

$\frac{|b|}{\|w\|}$ represents the distance of a line from the origin. Therefore, the distance between the classifier line and the upper marginal line is computed as:

$$d = \left| \frac{|b-1|}{\|w\|} - \frac{|b|}{\|w\|} \right| = \frac{1}{\|w\|} \tag{24}$$

Thus, the width of the margin is calculated as:

$$D = \frac{2}{\|w\|} \tag{25}$$

The objective function of the SVC model can be considered the D value maximization or:

$$Min. L = \frac{1}{2} w^T w \tag{26}$$

Also, when variables in the decision are in the ownership of the first class (*y* = 1), $w^T x + b$ should be ≥ 1 , while with variables belonging to the second class (*y* = -1) in the decision space, $w^T x + b$ should be ≤ -1 . Thus, the optimization problem can be presented as:

$$Min. L = \frac{1}{2} w^T w$$

$$subject\ to : \tag{27}$$

$$y(w^T x + b) \geq 1$$

The mentioned optimization problem is applied for the “hard margin” method where a solid fringe is taken account of the SVs. However, more flexibility is needed for practical purposes. This SVC will be achieved by receiving an error of ξ , for each of the marginal lines. Therefore, the optimization can be represented as:

$$Min. L = \frac{1}{2} w^T w + C \sum_{i=1}^n \xi_i \quad i = 1, \dots, n$$

$$subject\ to : \tag{28}$$

$$y_i (W^T X + b) \geq 1 - \xi_i \quad i = 1, \dots, n$$

$$\xi_i \geq 0 \quad i = 1, \dots, n$$

where ξ_i and *C* are a slack variable providing a soft classification margin and a penalty parameter, respectively. The dual problem can be solved to simplifying optimization. The dual solution can be written as:

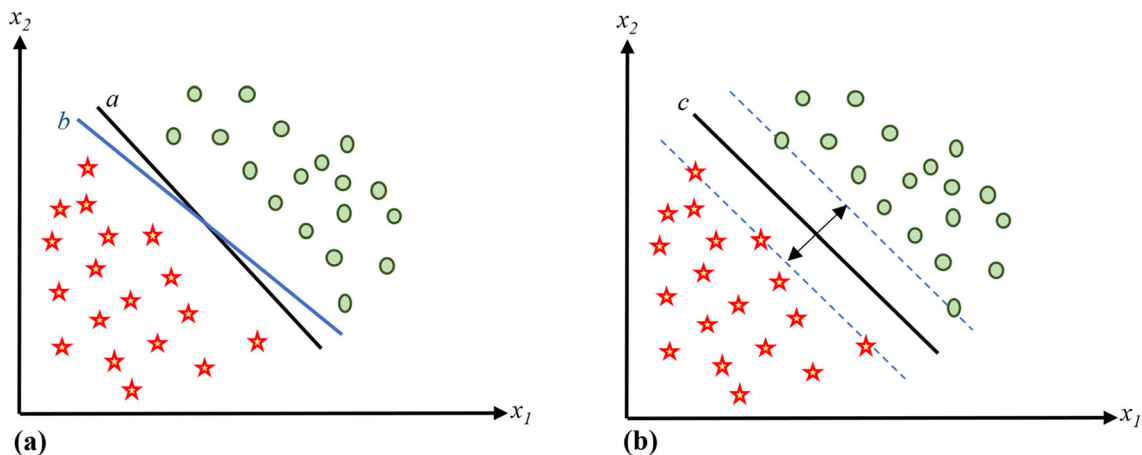


Fig. 5 The schematic view of classification with **a** artificial neural network (ANN) and **b** support vector classification (SVC) approach

$$\begin{aligned}
 \text{Max. } L_D &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i \cdot y_j \cdot \alpha_i \cdot \alpha_j \cdot x_i^T \cdot x_j \\
 \text{subject to :} & \\
 0 \leq \alpha_i &\leq C \quad i = 1, \dots, n \\
 \sum_{i=1}^n \alpha_i y_i &= 0 \quad i = 1, \dots, n
 \end{aligned} \tag{29}$$

where $\alpha_i \in \mathcal{R}^n$ presents Lagrange multipliers. In the training process, one α_i exists for each vector. Support vectors (SVs) correspond to the nonzero subset (α_i) and determine the decision surface, these SVs are the most informative. SVs present the location of the borderline (Fig. 5).

Mapping a higher-dimensional space through function $\varphi(x)$ can be obtained by the decision space variables (x). Therefore, the dual problem can be shown as:

$$\begin{aligned}
 \text{Max. } L_D &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i \cdot y_j \cdot \alpha_i \cdot \alpha_j \cdot k(x_i, x_j) \\
 \text{subject to :} & \\
 0 \leq \alpha_i &\leq C \quad i = 1, \dots, n \\
 \sum_{i=1}^n \alpha_i y_i &= 0 \quad i = 1, \dots, n
 \end{aligned} \tag{30}$$

where $k(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$ presents the kernel function. Thus, the SVC can be performed as follows:

$$f(x) = \text{sign} \left(\sum_{i=1}^n y_i \alpha_i k(x, x_i) + b \right) \tag{31}$$

This distinctive function is named as SVC. Table 2 has shown the kernel functions implemented in SVM’s formulations.

SVC presented in this section can be classified into two classes. When more than two categories are subjected, a suitable multiclass approach is required. There are two possible methods for this aim as following:

1. Modification of the SVC design to directly combine the quadratic solving algorithm and multiclass learning.
2. Incorporating several binary classifiers with two approaches:
 - “One against one” which employs pair comparisons among categories.
 - “One against the others” which compares a particular category with all the other categories.

Based on a comparative study (Kranjčić et al. 2019), the precision of these approaches has approximately been the same.

Support vector regression

The main difference between SVR and SVC is that in SVR, y instead of a binary number is considered a real number. When

$\{(x_i, t_i)\}_i^n$ is considered a dataset, n , t_i , and x_i are the total number of observed data, i th output vector, and i th input vector. The following equation can be used for SVR determination:

$$y = w\varphi(x) + b \tag{32}$$

Where b shows bias, w indicates weights, and the high dimensional feature space is shown as $\varphi(x)$, which is non-linearly constructed based on the input space x . SVR tries to minimize the risk by placing predicted variables between the lower and upper bounds of the dataset. Lower and upper borderlines are written as $y = w\varphi(x) + b - \varepsilon$ and $y = w\varphi(x) + b + \varepsilon$, respectively. Therefore, if a data is placed outside the marginal line, it must be adjusted (Fig. 4b). The following equations as an optimization set are applied for determining b and w coefficients:

$$\text{Min : } L = C \frac{1}{n} \sum_{i=1}^n L_\varepsilon(t_i, y_i) + \frac{1}{2} w \cdot w^T \tag{33}$$

$$L_\varepsilon(t_i, y_i) = \begin{cases} |t-y|-\varepsilon & |t-y| > \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

where C is the penalty parameter, L_ε shows insensitive loss function, and ε is acceptable error (tube size). ε and C are user-prescribed items. The following equation is used for the dual function with using Lagrange multipliers:

$$\begin{aligned}
 \text{Max } L_D &= \sum_{i=1}^n t_i (\alpha_i \alpha_i^*) - \varepsilon \sum_{i=1}^n (\alpha_i \alpha_i^*) \\
 &\quad - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i \alpha_i^*) k(x_i, x_j) \\
 \text{Subjected to :} & \\
 \sum_{i=1}^n (\alpha_i \alpha_i^*) &= 0 \\
 0 \leq \alpha_i &\leq C \quad i = 1, 2, \dots, n \\
 0 \leq \alpha_i^* &\leq C \quad i = 1, 2, \dots, n
 \end{aligned} \tag{35}$$

Then, w and b are calculated. The supporting vector is considered the Lagrange multipliers with non-zero values. Finally, the following equation is used for performing the SVR:

$$y = \sum_{i=1}^n (\alpha_i - \alpha_i^*) k(x, x_i) + b \tag{36}$$

RBF can be considered one of the common kernel functions (Tong et al. 2020). Thus, an SVR with RBF as the kernel function can be shown as $SVR(\gamma, C, \varepsilon)$.

Decision trees

Decision trees, as supervised machine learning models, are broadly employed in regression and classification tasks (Thomas et al. 2020). The decision tree structure consists of

Table 2 Some well-known kernel functions

Name	Function
Linear	$k(x_i, x_j) = x_i^T \cdot x_j$
Radial basis function (RBF)	$k(x_i, x_j) = \exp(-\gamma \ x_i - x_j\ ^2) \quad \gamma > 0$
Polynomial	$k(x_i, x_j) = (x_i^T \cdot x_j + 1)^d$
Sigmoid kernel	$k(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

a root node, internal and leaf nodes. The internal nodes represent the values of the attributes, and each leaf node of the tree includes the probability distribution and the label of the class. There are different decision tree building algorithms such as chi-squared automatic interaction detector (CHAID), exhaustive CHAID, and Classification and regression tree (CART). These models construct data in an easily interpretable way (Akin et al. 2018). The foremost differences among these models occur in the tree construction process. In CART, each node dissects into two nodes (binary splits), while in CHAID and exhaustive CHAID, each node dissects into more than two nodes (multiway splits) by default (Akin et al. 2018; Kusiak et al. 2010). All the methods restrict the tree size by avoiding over-fitting. CART first produces the whole tree and then makes a backward pruning (post-pruning) to optimize the size of the tree, while CHAID and exhaustive CHAID restrict any growth at the optimal size of the tree by employing a stopping criterion (Thomas et al. 2020). All descriptive variables are checked for the optimal split in the tree construction process, and trial splits are employed to find the best split point. The best split can be considered the one that maximizes the variance between groups and minimizes the variance within a group. Backward pruning (post-pruning) is carried out based on V -fold cross-validation. The whole tree is randomly allocated to the V number of groups. One of these groups is employed to validate the model and the residual is used to construct a model, and this step is repeated V times. V number of models are created and confirmed in this step (Thomas et al. 2020; Akin et al. 2018). The CHAID and exhaustive CHAID tree-producing methods include merging, splitting, and stopping processes. When the output variable is continuous (regression-type variables), the CHAID approach performs the best-next split based on the F test, while chi-square is employed when the output is categorical. Continuous variables are divided into groups with a similar sample size. The smallest Bonferroni adjusted p value is utilized for partitioning response variables. Descriptive variable pairs with the highest p -value are monitored for significance (which is generally considered 0.05). The pair is merged into a single node if the p -value is larger than 0.05. A group with three or more classes is analyzed to define the most significant split. When the node size is less than the predetermined minimum child and parent node size value, the tree growth process stops,

otherwise it continues (Rashidi et al. 2014). The exhaustive CHAID, as a modified CHAID method, needs more computing time because it uses more dependent variable testing and merging (Akin et al. 2018; Kusiak et al. 2010).

Random forest

RF as an ensemble learning method is introduced firstly by Breiman (2001). RF can be categorized as an ensemble of unpruned trees. This algorithm has been successfully employed in both classification and regression due to its simplicity in design and superior efficiency. Several merit points of the RF model such as prevent overfitting, the ability to handle noise, and the ability to manage a large number of features have been reported (Silva et al. 2019; Biau et al. 2019). The final output of the FR model is the combination of the output data of the individual trees (Fig. 4c). To maintain the minimum correlation among the individual trees while keeping the strength of the individual trees, each of the individual trees is thus built with two random injected sources for classification accuracy. First, each individual tree is trained based on randomly drawn with replacement (bootstrap replica) of the training data. Second, the algorithm defines the best split, at each node of each tree, based on a small variable subset that is randomly chosen from the whole variable set. Furthermore, each tree is completely grown to obtain a high variance of the tree outputs and low bias (Biau et al. 2019).

To solve regression problems, the mean squared error (MSE) is used to how data branches from each node. In fact, MSE determines the distance of each node from the forecasted real value, serving to determine which branch is better for the forest.

$$y = \sum_{i=1}^n (\alpha_i - \alpha_i^*) k(x, x_i) + b \quad (36)$$

where N is the number of data points, f_i is the value returned by the RF algorithm, and y_i is the value of data point i . To solve classification problems, the Gini index is applied to determine how nodes on the branch of a decision tree. Equation 31 determines which of the branches is more likely to happen based on the probability and class to calculates the Gini of each branch on a node.

$$Gini = 1 - \sum_{i=1}^c (p_i)^2 \tag{38}$$

where p_i corresponds to the relative frequency of the class in the dataset and c is the class number.

Entropy can be also used to calculate how nodes branch in a decision tree.

$$Entropy = \sum_{i=1}^c -p_i \times \log_2(p_i) \tag{39}$$

Entropy employs a certain outcome probability to decide how the node should branch. Unlike the Gini index, entropy is more mathematical intensive because of the logarithmic function applied in its formula.

Data fusion model

Nowadays, the necessity of increased precision and accuracy of AI models has encouraged researchers to establish applicable methods such as multi-model fusion-based (ensemble) approaches. The key idea of the ensemble model is fusing or combining data derived from fused information in order to obtain more precise predictions in comparing with implementing individual models (Alizadeh and Nikoo 2018). Many researchers in several fields of study have used data fusion (Alizadeh and Nikoo 2018; Hararuk et al. 2018; Meng et al. 2020). At more complex systems such as different stages of plant tissue culture, the ensemble model can be used to fuse the advantages and strengths of individual AI models. Several studies have demonstrated that ensemble models can be more reliable and accurate to model complex systems (Alizadeh and Nikoo 2018; Hararuk et al. 2018; Meng et al. 2020).

Data fusion is known as the process of combining and mixing data from various sources such as single outputs of several individual data-driven models (Fig. 4d) that the overall equation can be as follows:

$$\hat{y}_i = f(x_i) + \varepsilon_i \quad i = 1, 2, 3, \dots, n \tag{40}$$

where \hat{y}_i stands for target variable, x is a vector of independent indicators, ε stands for corresponding estimation error, and n is a number of observation data.

In order to develop data fusion models, Eq. (40) can be introduced to the following equation where several individual AI models are used as follows;

$$\begin{aligned} [\hat{y}_i] &= \begin{bmatrix} \hat{y}_{i1} \\ \hat{y}_{i2} \\ \cdot \\ \cdot \\ \hat{y}_{im} \end{bmatrix} = \begin{bmatrix} f_1(x_i) \\ f_2(x_i) \\ \cdot \\ \cdot \\ f_m(x_i) \end{bmatrix} + \begin{bmatrix} \varepsilon_{i1} \\ \varepsilon_{i2} \\ \cdot \\ \cdot \\ \varepsilon_{im} \end{bmatrix} \quad i \\ &= 1, 2, \dots, n \end{aligned} \tag{41}$$

where m stands for the number of individual model and $[\hat{y}_i]$ stands as matrix of estimations provided by each model.

Subsequently, the matrix of $[\hat{y}_i]$ will be considered input data in fusion models.

Many methods such as bagging, K -nearest neighbors (KNN) algorithm, ordered weighted averaged (OWA) method based on the ORLIKE weighting method (ORLIKE-OWA) and ORNESS weighting method (ORNESS-OWA) have been recommended for fusing individual models, which reported that the most powerful and uncomplicated among different approaches is the bagging method for data fusing.

Optimization algorithms

In plant tissue culture, the optimum selection between existing various alternatives can significantly reduce the costs and time. Thus, the use of optimization methods in the field of plant tissue culture is of particular importance (Osama et al. 2015; Zielinska and Kepczynska 2013; Guangrong et al. 2008). In general, optimization methods are categorized into two groups including classic and evolutionary algorithms. Among classic methods, linear programming (LP), dynamic programming (DP), stochastic dynamic programming (SDP), and non-linear programming (NLP) can be mentioned (Goudarzi et al. 2020; Moravej 2017). Each of these methods has limitations that restrict their use. The LP method has the limitation of being linear. In this method, the entire set of relations should be linear; therefore, this method is not applicable to the nonlinear problems, which are common in plant tissue culture. In the DP method, the linear limitation was removed, but this method is only applicable to the discrete case. The SDP method, which is a stochastic form of the DP, also has this limitation; moreover, the computation volume in the DP and SDP will exponentially increase if the dimensions of the problem increase that this case is the so-called curse of dimensionality. The NLP method has higher accuracy than other methods, although in this way, the process of solving the problem is time-consuming, and in complicated problems, it may stop in the local optimums and does not reach a global optimum solution (Goudarzi et al. 2020; Moravej 2018; Bozorg-Haddad et al. 2017). The second category of methods includes evolutionary or metaheuristic ones, which have been developed based on a natural process (Moravej and Hosseini-Moghari 2016; Bozorg-Haddad et al. 2016; Hosseini-Moghari et al. 2015; Haddad Omid et al. 2016). Since these methods are not dependent on the problem type in terms of being linear or nonlinear and converge in the global optimum solution (close to the optimum) with a high-speed, are enormously popular. In this regard, in this paper, the Genetic Algorithm (GA), which is the most well-known evolutionary method, was introduced.

Among the particular contexts that the GA can be potentially applied, optimization of different in vitro culture stages, optimization of the MLP neural network weights, optimization of the spread parameter in the RBF, GRNN, etc., neural networks can be pointed out. In general, at any situation in which there are unknown variables or variables that their optimum amounts should be obtained based on a given criterion (objective function), the GA can be employed (Jamshidi et al. 2019).

In this section, firstly, we will discuss the basic concepts of optimization; some expressions such as the objective function, decision variable, and state variable will be introduced, and the GA will be explained. Then, the concepts of chromosome, gene, genetic and evolutionary operators, and the performance quality of the GA will be stated.

Basic concepts of optimization

Since the GA is an optimization algorithm, before entering into the discussions explaining the GA, at first, basic concepts of optimization will be stated. These concepts are general and exist in all optimization methods.

Decision variable: the variable that an optimization process is conducted to find its optimum value.

State variable: the variable that will not be optimized directly, but after determining the optimum amount of the decision variable, the value of this variable is also calculated; in fact, its value depends on the decision variable value.

Objective function: the criterion that optimum values of the decision variables are obtained based on it. This criterion can be defined as minimizing or maximizing.

The presence of the state variable in the optimization problems is not permanent, though the decision variable and objective function are two principal parts of any optimization method. For example, consider a function that gives an input such as x , and concludes an output like y . In a simple situation, it can be shown in the form of $y = ax + b$; however, if we want to determine the appropriate amounts of a and b according to the available data, we will require to a criterion for comparing between calculated y s and observed ones for different values of a and b . This criterion can be *MSE* or *RMSE* or any other criterion. In this example, a and b are decision variables of the problem, and *MSE* or *RMSE* are the objective functions of the problem, which we intend to minimize them.

Genetic algorithm

The GA, for the first time, was designed by Holland (1992) and developed as a powerful tool of optimization. The GA is a searching algorithm, which is derived from the biological nature and the process of natural selection. This method is based on Darwin's notion, which says that in the environment, organisms that are much more stable than the others can always

survive (Yun et al. 2020). Before entering into the discussion of the GA algorithm behavior, it is essential to know some of the basic concepts about GA.

Basic definitions

In the genetic algorithm, seven concepts are widely used; these concepts include (1) gene, (2) chromosome, (3) population, (4) evolutionary operators, (5) genetic operators, (6) elitism, and finally (7) generation. Genetic algorithm is based on these concepts, which will be defined as follows:

1. **Gene:** in the GA, each decision variable of the problem is called a gene.
2. **Chromosome:** a set of genes that are actually a series of answers for the studied problem are called a chromosome. In a problem with one decision variable, the gene and chromosome are the same.
3. **Population:** a set of answers (chromosomes) is called population.
4. **Evolutionary operators:** in order to create new answers in the GA, it is needed to select parent chromosomes; this selection is performed by the evolutionary operators.
5. **Genetic operators:** after selecting the parent chromosomes, creating new generations is performed by genetic operations of crossover and mutation.
6. **Elitism:** In each iteration of the algorithm, the best answers are not undergone crossover and mutation, and are transferred to the next iteration intact, that they are called elite.
7. **Generation:** each iteration of the algorithm is called a generation.

Overall flowchart of genetic algorithm

Before implementing the GA, some parameters such as selection method, crossover fraction, mutation rate etc. should be determined. Then, a set of probable answers will be created. Indeed, a population is created through chromosomes. These chromosomes possess the genes equal to the number of problem dimensions. During the process of optimization, these genes are improved by means of genetic operators of the crossover and mutation. Chromosomes are chosen for moving to the next generation according to the sufficiency of their corresponding objective function. In this selection, the operators such as the roulette wheel and tournament selection, which are evolutionary operators, are used. Using a crossover operator, a number of genes from two selected chromosomes are replaced with each other, moreover, some genes randomly changed by mutation operator. In addition, using the elitism parameter, we are able to increase the chance of selecting the best chromosomes, and consequently, improve the algorithm

convergence. In the producing of each new generation, three operators including selection, crossover, and mutation, handle the process of optimization in such a way that created chromosomes make the objective function value better and better at each iteration to the extent that the optimization process will be ended by one of the stopping conditions (Yun et al. 2020). Figure 6 illustrates flowchart of the GA. The flowchart steps have been described in the following section.

Determining the parameters and stopping criteria

In order to implement the GA, the values of its parameters must first be determined, and then, for finishing algorithm process, the stopping conditions are necessary. In the determining part, some parameters such as the number of populations, crossover fraction, mutation rate, and the number of elites are determined. Optimum amounts of these values are given through the trial and error method. Crossover fraction is a percentage of the population on which crossover operation is performed. The mutation rate is a percentage of the genes of the population on which the mutation process is performed. Elites are a number of population members who are transferred unchanged to the next generation.

In order to end the algorithm procedure, we must consider the criterion; this criterion is arbitrary, but usually, the number of iterations (generation) is taken as the stopping criteria. In this way, the algorithm will stop after passing the maximum iterations. However, other criteria such as “to achieve a satisfactory answer” or “passing a certain number of iterations without observed improvement” or “time of algorithm implementation” can be selected as the stopping criteria.

Creating the initial population

The initial population, which its number is one of the algorithm parameters, is created randomly. This population is the number of chromosomes. The manner of creating an initial population is arbitrary but creating population between the upper and lower band of decision variables may be a good choice for the initial population.

Calculating the objective function value

Based on the created population and written cost function, the objective function value is calculated. In fact, equal to the number of population members, the objective function is calculated. Each objective function calculation is called an assessment; therefore, in a generation with a population equal to 50, the number of iterations is equal to 1, but the assessment number will be 50. The objective function value corresponding to each chromosome is used in the next stages of the algorithm.

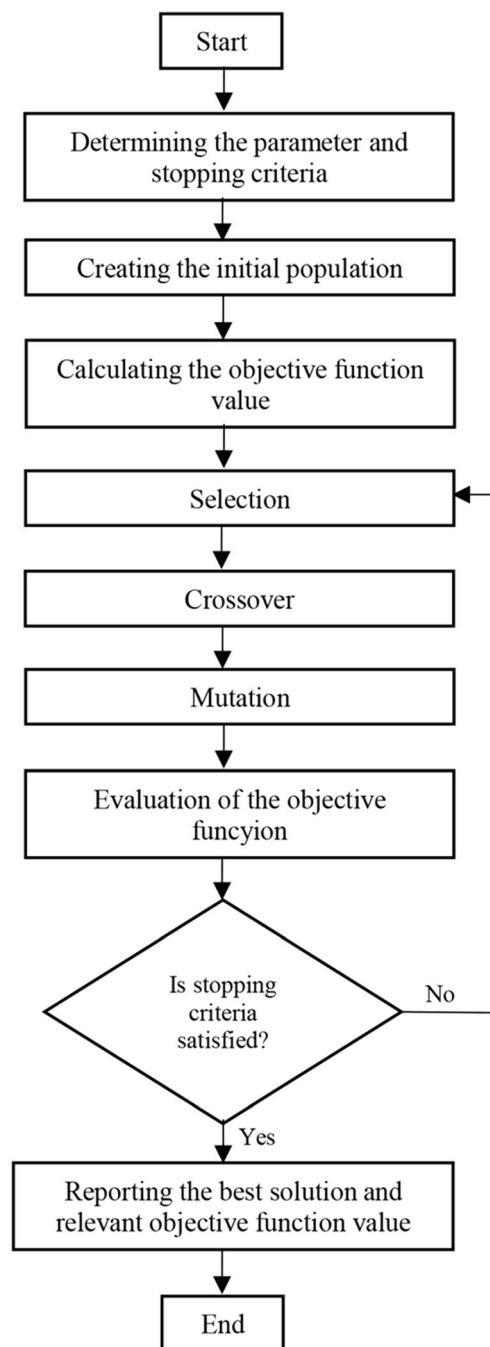


Fig. 6 The schematic diagram showing the step-by-step genetic algorithm (GA) optimization process

The selection

In order to perform the crossover, two chromosomes should be selected. These chromosomes are considered parent, and by combining their genes, two offspring are established that are replaced with parents in the next generation. The selection process is usually based on the fitness value of the relevant objective function of each chromosome. Each chromosome that has more optimal objective function values is more likely

to be selected and, as a result, to create the next generation. There are several ways to select that some of them are mentioned below.

Tournament Selection: In this method, some of the population members are randomly chosen. Among the selected set, a chromosome that shows the best fitness is considered parent. Selected members come back to the population, and once again, this procedure is performed, and the second parent is selected.

Roulette wheel: roulette wheel selection method is one of the most efficient and best-known methods of selecting. In this way, the probability of selecting chromosomes with higher fitness value of the objective function is greater; in other words, the probability of selection allocates to each chromosome corresponding to its fitness value.

To understand the mechanism of the roulette wheel, consider a circular plate (Fig. 7a), which has been divided into n parts (unequally). If we put an indicator in one side of the plate, and roll the plate, when the revolving plate stops, the parts that cover greater portions of the circle perimeter, have more probability to be the front of the indicator. The roulette wheel also employs a similar process for the selection of parent chromosomes. If the circle perimeter is equal to one, so the amount less than one will be allocated to the perimeter of each part (p_i). Thus, if we cut the circle perimeter from a point, we will consider it like a line (Fig. 7a).

As over the line, perimeters (probabilities) are placed in the form of cumulative; if we create a random number in the range of zero to one, it is more likely to be placed in front of the ranges with a greater perimeter when the upper limit of each band is considered an index of the location of the random number; in this way, if the random number q is placed in the range of zero to p_1 , the index 1 will be selected, and if is placed in the range of p_1 and $p_2 + p_1$, index 2, and so on.

The crossover

Survival of generations in nature is done by intercourse. In the GA, consequently, there is the nature of combination and intercourse. The combination is performed among chromosomes with gene replacement, and each parent chromosomes transfer their characteristics to their children. This process in the GA is arranged by means of the crossover operation. Crossover is a process in which the current generations of chromosomes mix, and create a new generation of chromosomes. Although the crossover operator may be applied on a chromosome or more than two chromosomes, usually, for each combination, two chromosomes are selected as the parent, and by joining them, two offspring are created. In the following section, some of the well-known crossover methods have been mentioned in the GA.

One-point crossover: this method was developed by Holland (1992). For mixing two chromosomes, parent

chromosomes are cut from one given point and their genes are replaced with each other. Figure 7 b displays the single point crossover operator.

Two-point crossover: in this method, the parent chromosomes are randomly cut from two points. In this way, each chromosome is divided into three parts, which in order to combine, the middle section is fixed and the surrounding parts of two chromosomes are replaced (Fig. 7c).

Uniform crossover: in the previous methods, replacements were merely possible at cut points, however, in the uniform method, potential of replacement is uniformly considered for all genes. The number of replaced genes in this method is not fixed, but usually, half of the chromosome length is taken into account. Figure 7 d shows a uniform crossover operator.

These abovementioned methods are well used for binary problems, but in the continuous problems, replacement without changing genes is not appropriate, because in the continuous problems infinite answers are possible. Moreover, in the case of raw replacement of genes, their values are only replaced between established amounts in the initial population. In continuous problems, special methods are used to replace genes. Some of these methods include the arithmetic method and the sequential method.

Arithmetic method: In this method, the genes of two chromosomes of A and the B , which are selected for the crossover process, are transferred to offspring chromosomes (a, b) by using the following equations:

$$\begin{aligned} a &= \alpha A + (1-\alpha)B \\ b &= \alpha B + (1-\alpha)A \end{aligned} \quad (42)$$

where α is selected in the interval of $[0,1]$. The amount of $\alpha = 0.5$ causes that the children are gained out of parental genes. We can consider the α as a random number in the interval of $[0,1]$ with different values for each gene.

Sequential method: in this method, parent chromosomes are cut from two points. Then, the middle section is kept constant for both chromosomes. The left and right parts, which are calculated for the first child in the second parent, have to start from the beginning of the chromosomes; genes that are absent in the middle of the first child are placed in the blank spaces randomly. In the second child, also, the middle part of the second parent is kept constant, and the previous process is performed based on the first parent.

The mutation

The aim of mutation in the GA is to create variety in the solutions. This operator beside the crossover operator leads to converge the GA toward the optimum solution. Mutation on chromosomes creates random variations. These changes contribute to enter the new genes into the population; as a

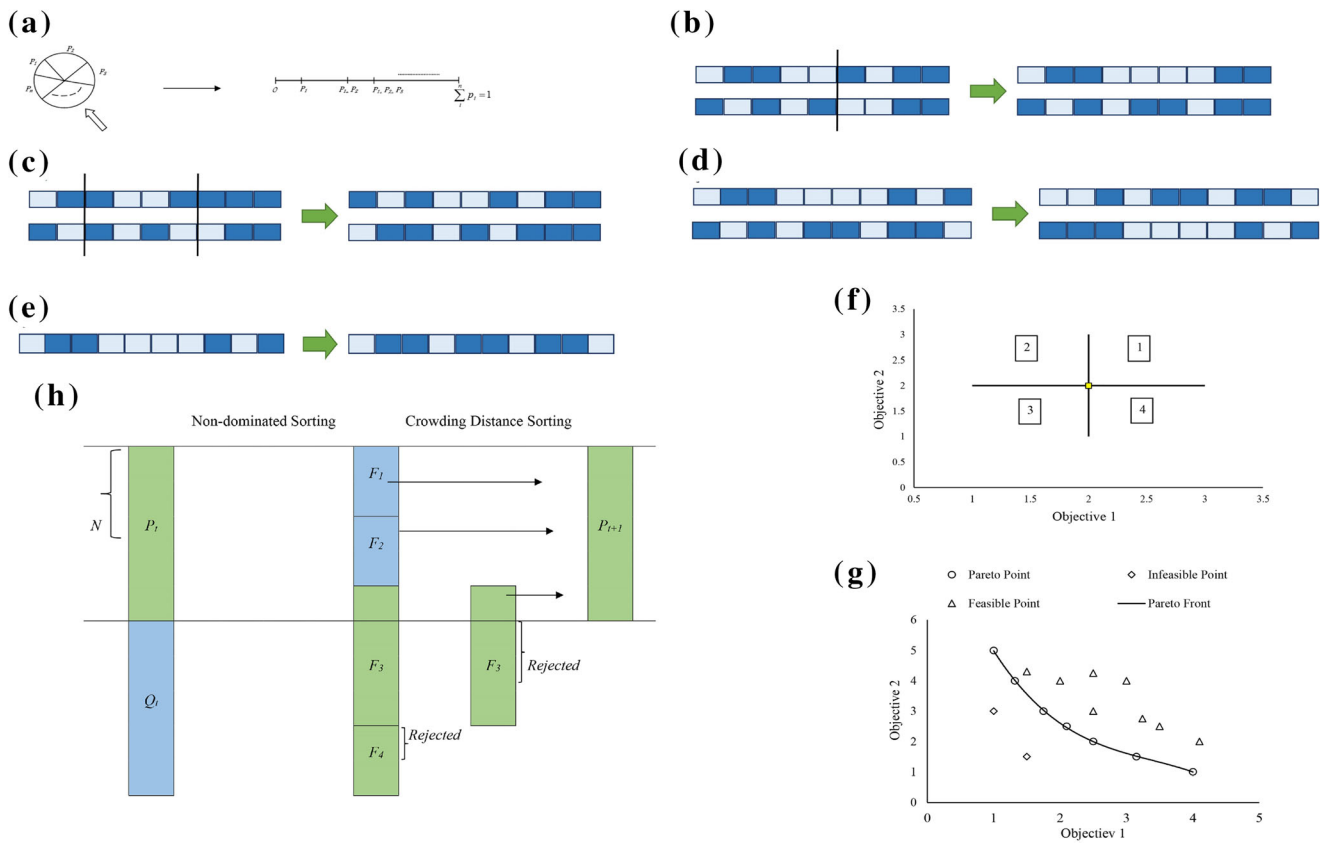


Fig. 7 The schematic view of **a** Roulette wheel, **b** one-point crossover, **c** two-point crossover, **d** uniform crossover, **e** the mutation on a chromosome, **f** the objective space of an optimization problem with two

objectives, where the yellow point in the center of areas is considered a solution, **g** the objective space of a two-objective optimization problem, and **h** the binary tournament selection

result, creating new genes causes the comprehensive search of the algorithm. The mutation operator prevents the algorithm from trapping in the local optimum solutions. Furthermore, a low mutation rate causes that changes in the low number of genes does not have clear effects on the problem solution; in the other words, the high mutation rate makes children have little resemblance to their parent; so, this fact results in the loss of the historical memory of the algorithm. Thus, the mutation rate should be optimally determined by the trial and error method. Figure 7 e shows the mutation on a chromosome.

Random mutation operator (uniform): in this method, a number of genes are randomly replaced with new genes. Having a mutation rate and generating random numbers in the interval of [0,1], we can replace the genes, which their relevant random number is less than the rate of mutation, with a new gene. Creating a new gene could be based on the relationship of the initial population production.

Gaussian mutation operator: in this method, also, selection of a gene, which is supposed to be replaced by another gene by mutation operator, is similar to the previous method. In this way, if the selected gene is equal to x_k , its value with the aid of a random number will be replaced with a normal distribution with mean x_k and standard deviation σ . In this case, σ

can be appropriated according to the permissible range of x_k (upper and lower limit of the variable).

The presented methods in the previous section concerning the selection, crossover, and mutation are only a part of the existing methods in this area, although, knowing the basic concepts, the use of other methods will be possible; in addition, the inventive methods can be applied in this context as well. With respect to what was stated, the population in the next generation will be composed of three parts: the first part, elites that will be moved to the next generation without any change; the second part, the population obtained from crossover process; and the third part, the population obtained from mutation process.

The cost function

In the previous sections, we observed that the genetic and evolutionary operators act based on being appropriate for the objective function value corresponding to each chromosome. Nevertheless, the question is how to calculate the objective function value? To this end, we need a cost function. The cost function is not related to the implementation of the GA algorithm. Conversely, it is related to the optimization problem

and should be written for each optimization problem separately. As it is clear from the name of the cost function, a function has to be defined. So, the desired function needs to input, and based on the input, it concludes output. A cost function in the GA takes chromosome into account as the input of the function, and the output of the function is the objective function value corresponding to that chromosome. In fact, the cost function is the main core for solving an optimization problem that the modeling system is carried out inside of it.

Multi-objective optimization

In many real problems of plant tissue culture, simultaneous optimization of multiple objective functions is considered. These functions are often in conflict with each other (George and Amudha 2020). For example, in the sterilization step, the sterilant has a negative effect on the explant viability and a positive impact on controlling contamination. Therefore, the ultimate aim of in vitro sterilization protocols would be the maximum explant viability and the minimum contamination.

In the multi-objective problems, in addition to the decision space of the problem, there is another space that is called objective space in which coordinate axes show the objective function value. Therefore, in order to define a multi-objective problem, if $K \subseteq R^n$ is considered the n -dimensional search space, $D \subseteq K$ as the justified space of multi-objective problem, and $O \subseteq R^m$ as the m -dimensional objective space, then we will have:

$$\begin{aligned} & \text{Min} \\ & \text{subjected to : } Z(x) = (z_1(x), z_2(x), \dots, z_m(x)) \\ & \quad g_p(x) \leq 0 \quad p = 1, \dots, P \\ & \quad h_q(x) = 0 \quad q = 1, \dots, Q \\ & \quad x_i \geq x_i^{\min} \quad i = 1, \dots, n \\ & \quad x_i \leq x_i^{\max} \quad i = 1, \dots, n \end{aligned} \quad (43)$$

where $g_p(x)$ is the p th inequality constraint, $h_q(x)$ is the q th equality constraint, P is the number of inequality constraints, Q is the number of equality constraints, x_i^{\min} is the lower limit of decision variables, and x_i^{\max} is the upper limit of decision variables.

In the single-objective optimization problems, judgment on the final optimal point was simple, because in this kind of problems, we only talk about one objective function. Hence, the optimal solution corresponding to the best value of the objective function can be regarded as the optimum solution of the problem. However, in judgment, in the multi-objective problems, it is not simple because the objective functions are usually in conflict with each other, so improving one will worsen the other. Therefore, we must establish a balance between the objectives. The aim of balancing is to find an equivalent solution between different objective functions of the problem. A balanced solution is concluded when it is not

already possible to improve each of the objective functions without deteriorating other function values. According to the offered definition, a balanced solution of a problem may be more than one, and here, an individual solution will not be presented for the problem like the single-objective optimization. Each balanced solution is so-called a non-dominated solution, and the set of these solutions is known as non-dominated set or Pareto Optimal Set. In addition, these non-dominated solutions in the objective space of the problem form a front namely Pareto Front.

Domination: The solution x_1 dominates solution x_2 if and only if:

1. The solution x_1 is worse than x_2 in none of the objective functions.
2. The solution x_1 is better than x_2 at least in one objective function.

The mathematical expression of domination is presented as follows:

$$x_1 \prec x_2 \Leftrightarrow \forall i \ z_i(x_1) \leq z_i(x_2) \quad \& \quad \exists i_0 \ z_{i_0}(x_1) < z_{i_0}(x_2) \quad (44)$$

$$\text{where } \prec \text{ is the Pareto dominance relation.} \quad (44)$$

The definition above for both objective vectors (Z_1, Z_2) in the objective space can be defined similarly. If Z_1 is not worse than Z_2 in any objective function and is better than it at least in one objective function, then we will have $Z_1 \prec Z_2$. According to what was stated, it can be concluded that the solution x_1 dominates the x_2 solution when $Z_1 \prec Z_2$.

Pareto Optimal: decision vector of $x^* \in D$ will be called Pareto optimal if there is no other decision vector similar to $x \neq x^*$ that dominates it. Moreover, decision vector of $Z(x)$ will be Pareto optimal when $x = x^*$.

Pareto Optimal Set: the set of all decision vectors related to Pareto Optimal is named Pareto Optimal Set. If POS is Pareto Optimal Set, then it will be defined as follows.

$$POS = \{x^* \in D \mid \nexists x \in D : x \prec x^*\} \quad (45)$$

Pareto Optimal front: The set of all objective vectors corresponding to the POS set is called the Pareto optimal front.

Figure 7 f is associated with the objective space of an optimization problem with two objectives. In this figure, if the yellow point in the center of four areas including 1, 2, 3, and 4 is placed as a solution, this point dominates all existing points in area 1, but all the existing points in area 3 dominate this point, whereas in areas 2 and 4, no point dominates this point as well as this point dominates no point in these areas. Considering areas 2 and 4, the necessity of providing a set

of optimal solutions rather than a single solution is completely clear, because in this case, it is not certain which solution is better than other solutions. In fact, solutions have no superiority over each other.

Figure 7 g shows the objective space of a two-objective optimization problem. As it is obvious, the points that have not been dominated by any other points (*Pareto point*) form Pareto front. A good Pareto front is a one that has a maximum possible length, which means that covers all the non-dominated feasible solutions of the problem (George and Amudha 2020). In this regard, finding the points of both Pareto front sides will be more valuable. *Infeasible* points are known as points that do not meet the problem constraints. In fact, they are a mapping form of the *infeasible* decision variables in the decision space over objective space. *Feasible* points are points that meet the problem constraints, but was defeated by other point or points. Thus, the Pareto points are *Feasible* points that are non-dominated. It is necessary to note that in solving the problem, Pareto points are usually an estimation of Pareto front and cannot be fully compliant. The more compatibility among them is gained, the more quality of the obtained solutions is observed. (George and Amudha 2020).

Multi-objective optimization methods such as single-objective methods are divided into two categories: classic and smart (evolutionary). Of the classic solving methods related to multi-objective optimization problems, weighted sum method, goal programming, and goal attainment method can be noted. In these methods, a multi-objective problem by considering different weights for each objective will repeatedly be solved again and again, and thus, the Pareto curve is obtained. In other words, in these methods, with regard to the weight for each objective, the problem is changed into a single-objective problem. Therefore, these methods are so-called decomposition methods. Evolutionary multi-objective methods are very diverse, which include the first and second versions of the micro genetic algorithm (μ GA, μ GA2), and the first and second versions of non-dominated sorting genetic algorithm (NSGA-II, NSGA). The most well-known evolutionary multi-objective method is without doubt the NSGA-II (George and Amudha 2020). This algorithm will be described in the following section.

The second version of the non-dominated sorting genetic algorithm

NSGA-II algorithm is one of the most well-known and powerful algorithms that is applied for solving multi-objective optimization problems and its effectiveness has been proven in solving various problems. The first version of the algorithm (NSGA) was presented by Srinivas and Deb (1994). In this method, the concept of dominance was used and the Pareto that had an appropriate scattering plot (diversity in solutions)

as well was considered more suitable Pareto. Important matters that exist in this optimization approach include:

- The solution over which there is no other solution certainly better than is rated further. The solutions are ranked and sorted based on how many solutions dominate them.
- Competence and quality of solutions is determined according to their rating.
- Uses the Fitness Sharing method to diversify the solutions. In this method, if the Pareto points are closer to each other from a certain distance such as σ that is called sharing parameters, the fitness amount will be decreased with sharing between points.

The sensitivity of performance and quality of solutions in the algorithm NSGA to the parameter σ was very high, so that it was so difficult to determine its appropriate amount; moreover, the lack of elitism and complex calculations in determining the non-dominated solutions was so inappropriate in such a way that it was not simply negligible. In this regard, the second version of the algorithm NSGA namely NSGA-II was introduced by Deb et al. (2000) and Deb et al. (2002). This algorithm and its unique manner of dealing with the multi-objective optimization problems has been repeatedly applied by different researchers to create newer various multi-objective optimization algorithms. In the NSGA-II version, instead of the concept of the fitness sharing method, another concept called crowding distance has been used. Furthermore, the algorithm used to find the non-dominated solutions has reflected a significant improvement in terms of computational issues.

As mentioned before, a good Pareto front has two characteristics: the first feature is the solution quality, which means that the non-dominated solutions are in the Pareto front, and the second feature is the appropriate distribution and diversity of Pareto points. In the algorithm NSGA-II, the quality of solutions is determined by the relevant rank of each solution. Determining the rank of each solution is done by an algorithm called non-dominated sorting (NS) algorithm. the diversity of solutions is another criterion that is quantized by crowding distance. Therefore, a good solution is a solution that, in the first place, has the best quality and, in the second place, includes the highest crowding distance. In the following parts, these subjects will be discussed.

Fast non-dominated sorting algorithm

Discussed Pareto front until now is called the first front or F_1 . If we do not take the F_1 into account, another Pareto front can be extracted namely F_2 . This process can be applied until determining all fronts. For non-dominated sorting of a population with the size of N , any solution will be compared with other solutions. All individuals of the population, which are

not dominated by any member, are placed in F_1 . For finding individuals of the next front, the previous process can be repeated by temporary ignoring of F_1 . This approach will nearly impose extra computational issues on us. In this section, the fast non-dominated sorting approach, which was introduced by Deb et al. (2000), will be discussed.

At first, for each solution, we consider two features; first feature n_i , which is the number of solutions that dominates i solution. The second feature S_i , which is a collection of the population individuals that are dominated by i solution. Whole solutions that have a n_i equal to zero, are known as members of F_1 (these solutions are rated as 1, the rate of anybody is the number front that is placed in). When F_1 is called the current front, for each solution in the current front if the j th individual is observed in the S_i group, one unit will be decreased from n_j . In this way, if n_j is equal to zero, that solution will be located in a separate list called H .

When all individuals of the current front are evaluated, individuals of the F_1 will be introduced as the first front, and the process by considering H as the current front will be repeated. From now, the NS is known the same as fast non-dominated sorting (FNS). Mathematical description of the abovementioned matters related to the non-dominated sorting of the population (P) are presented as follows:

Fast Nondominated Sort (P):

$$\begin{aligned}
 & \text{for each } p \in P \\
 & \text{for each } q \in P \\
 & \quad \text{if } p \prec q \text{ then} \\
 & \quad \quad S_p = S_p \cup \{q\} \\
 & \quad \text{elseif } q \prec p \text{ then} \\
 & \quad \quad n_p = n_p + 1 \\
 & \quad \text{if } n_p = 0 \text{ then} \\
 & \quad \quad F_1 = F_1 \cup \{p\} \\
 & \quad \quad k = 1 \\
 & \quad \quad \text{while } F_k \neq \emptyset \\
 & \quad \quad \quad H = \emptyset \\
 & \quad \quad \text{for each } p \in F_k \\
 & \quad \quad \text{for each } q \in S_p \\
 & \quad \quad \quad n_q = n_q - 1 \\
 & \quad \quad \quad \text{if } n_q = 0 \text{ then} \\
 & \quad \quad \quad \quad H = H \cup \{q\} \\
 & \quad \quad \quad k = k + 1 \\
 & \quad \quad \quad F_k = H
 \end{aligned} \tag{46}$$

Crowding distance

Among the optimal solutions presented with the same rank, the solution that could bring more diversity has superiority. Thus, the solutions that are in the vacant regions of the objective space of the problem are considered superior. This advantage is quantified using the crowding distance. This distance for individuals over each front is calculated separately. To this end, individuals existing in each front should be ranked in an

ascending format according to each objective function. As the smallest and largest values of the objective function are of great importance, the amount of crowding distance is considered equal to ∞ for them. For the rest of the solutions, crowding distance is computed based on the normalized difference amount between the objective function of two sides (before and after) of each solution. If we are going to calculate the crowding distance for L solutions existing in the τ front, we will have:

$$\begin{aligned}
 & \text{crowding-distance-assignment}(\tau) \\
 & \quad L = |\tau| \\
 & \quad \text{for each } i, \tau, CD = 0 \\
 & \quad \text{for each objective } m \quad \% \text{sort using each objective} \\
 & \quad \quad \tau = \text{sort}(\tau, m) \\
 & \quad \quad r_1.CD = \tau_L.CD = \infty \\
 & \quad \quad \text{for } i = 2 \text{ to } L-1 \\
 & \quad \quad \tau_i.CD = \tau_i.CD + \frac{|\tau_{i+1}^m - \tau_{i-1}^m|}{(z_{\max}^m - z_{\min}^m)}
 \end{aligned} \tag{47}$$

By calculating the crowding distance, we are able to define an operator, which includes in the first place domination and in the second place the crowding distance. This operator can be employed in different steps of the NSGA-II algorithm as a guide to move toward an optimum Pareto with an even distribution. This operator, which is called the crowded-comparison operator (\prec_n) is defined as follows; assume that each member of the population, such as i , has two characteristics including rank (i_{rank}) and crowding distance ($i_{distance}$)

$$i \prec_n j \Leftrightarrow \text{if } (i_{rank} < j_{rank}) \text{ or } (i_{rank} = j_{rank} \text{ and } i_{distance} > j_{distance}) \tag{48}$$

Therefore, between two solutions with unequal ranks, the superior solution is one with a lower rank, or if the rank of solutions is equal, the solution that has more crowding distance is a better solution. The criterion (\prec_n) is used to select the parent and create the new population.

Here, it is clearly characterized the difference between the single-objective GA algorithm and NSGA-II algorithm. Since in the NSGA-II the objective space is not sortable, we are not able to find the best solution by a sorting. So, in any part of the GA, that amount of the Cost was considered a criterion for selecting, in the NSGA-II criterion of (\prec_n) will be replaced with the Cost.

Binary tournament selection

The selection process in the algorithm NSGA-II is performed using the binary tournament selection method. In this method, the following steps are pursued:

- Two members of the population are chosen randomly.

- If two members are not of the same rank, the members with a lower ranking are selected.
- If two members are of the same rank, a member is selected that has further crowding distance.
- Selected members are returned into the population, and the previous process is repeated for the second parent selection.

After selecting the parents, the crossover operator is applied over them and offspring are created; then, mutation is applied on the population. So, the existing population will consist of three sections; the first section, the previous population; the second section, the population obtained from the crossover; the third section, the population achieved from mutation. After the merging of these populations, to the number of N , superior individuals (N is the allowable size of the population) should be selected. In this regard, at first, the population is sorted in ascending order based on non-dominated sorting (NS), and we begin to take individuals from F_1 to create a new population. If the number of F_1 individuals is less than N , all of its members will be transferred to the new population, and the remaining members will be transferred from subsequent F_s until the new population size would be equal to N . Continuing this process in a front like F_k , the size of the new population reaches N , in this front, all individuals are the same rank; so, in such condition, the front F_k based on the crowding distance is sorted in decreasing order, and decision-making criteria would be the crowding distance. Until the new population reaches size N , individuals are transferred to the next generation from the beginning of the front F_k . Figure 7 h illustrates this process well. If the number of F_j members is more than N , because the rank of all members in this front is equal to one, existing individuals in the front according to crowding distance are sorted in decreasing order, and the first N members are elected as the new population. Figure 7 h shows how to select a new population. In this figure, P is the main population and Q is the population resulting from the crossover and mutation.

The algorithm procedure such as the GA will continue until satisfying one of the stopping conditions. The obtained non-dominated solutions from solving the multi-objective optimization problem have no priority over other solutions, and depending on the circumstances, each of them can be considered an optimal decision.

AI-OA in plant tissue culture

In vitro culture consists of non-linear and non-deterministic developmental processes. In fact, in vitro culture stages are multi-variable procedures impacted by different factors such as plant genotype, culture medium, different types and concentrations of plant growth regulators (PGRs), etc. (Fig. 1)

(Zielinska and Kepczynska 2013). The data derived from the plant tissue culture process can be categorized as (1) binary inputs which have only two grades, e.g., non-embryogenic/embryogenic callus, (2) discrete variables which include more than two grades, such as the number of roots, shoots, and embryos, (3) continuous variables which can consist of any grade, e.g., length of shoots or roots, and callus weight, (4) time-series data, (5) temporal data, (6) fuzzy inputs that relate to the degree of vitrification, callus color, and the developmental stages of embryos, and (7) categorical variables, e.g., the type of reaction, or the type of phytohormones and carbohydrate sources (Prasad and Gupta 2008a; Osama et al. 2015). The complexity of this situation and the interactive nature of the variables makes optimization challenging using traditional approaches in which single variable are generally evaluated sequentially in isolation. To address this challenge, artificial intelligence models and optimization algorithms have recently been used for modeling, forecasting and optimizing different stages of plant tissue culture. AI and OA methods used in various steps of in vitro culture were presented in this section and the AI-OA application was summarized in Table 3. As summary of various studies using AI systems to optimize different stages of plant tissue is presented below.

In vitro sterilization

Surface sterilization is an initial step of micropropagation in which the final success of plant tissue culture is directly dependent on. The surface sterilization performance can be impacted by various factors, e.g., the type, age, and size of the explant, physiological phase (vegetative or reproductive) of the mother plant, physical in vitro conditions (temperature and light), type and concentration of sterilant, and immersion time. Several studies (Teixeira da Silva et al. 2016; Hesami et al. 2017a; Hesami et al. 2018b; Cuba-Díaz et al. 2020) revealed that treatments with longer immersion time and greater concentration of disinfectants led to better surface disinfection. However, there is a negative correlation between explant viability and high concentration of disinfectants with long immersion time such that the efficiency of sterilization must be balanced with explant health (Hesami et al. 2019b; Cuba-Díaz et al. 2020). Thus, the type/level of sterilant and immersion time must be optimized for each species and explant to obtain the best outcomes during the surface sterilization. Optimizing this step is costly and time-consuming and some disinfectants are not environmentally friendly and/or hazardous to human health. To ease this problem, A hybrid AI-OA could be a reliable and useful statistical methodology for forecasting and optimizing this step. For instance, Ivashchuk et al. (2018) used MLP and RBF methods for studying and predicting in vitro sterilization of *Bellevalia sarmatica*, *Nigella damascene*, and *Echinacea purpurea*. Different concentrations of lysoformin, biocide, liquid bleach,

Table 3 Application of artificial intelligence models and optimization algorithms in different stages of plant tissue culture

Plant Species	Purpose	Model	Inputs	Output(s)	Reference
<i>Bellevialia sarmatica</i> , <i>Nigella damascene</i> , <i>Echinacea purpurea</i> .	Predicting and optimizing the sterilization	MLP and RBF	Lysoformin, biocide, liquid bleach, chloramine B, silver nitrate, and immersion time	The percentage of contamination and explant viability	Ivashchuk et al. (2018)
<i>Chrysanthemum</i> × <i>grandiflorum</i>	Predicting and optimizing the sterilization	MLP-NSGA-II	NaOCl, AgCl ₂ , Nano-silver, AgNO ₃ , Ca(ClO) ₂ , H ₂ O ₂ , and immersion time	The percentage of contamination and explant viability	Hesami et al. (2019b)
-	Predicting temperature inside the culture containers	MLP	Four node temperatures	Nusselt numbers for the top surface and for the side surface	Murase et al. (1996)
<i>Cuminum cyminum</i>	Predicting and optimizing callogenesis	MLP	Area, minor axis length, feret diameter, weighted density, and perimeter	Fresh weight and volume of callus	Mansouri et al. (2016)
<i>Trachyspermum ammi</i>	Predicting and optimizing callogenesis	MLP	Explant age, the concentrations of kinetin, 2,4-D, and sucrose	Morphological features of callus (Feret diameter, perimeter, roundness, true density, and area of the callus)	Niazian et al. (2018b)
<i>Gyrinops walla</i>	Predicting and optimizing callogenesis	MLP	Different explants, NAA, BAP, coconut water, and different media (MS and WPM)	Days taken to initiate callus, callus weight, and callogenesis rate	Munasinghe et al. (2020)
<i>Daucus carota</i>	Predicting and optimizing cell growth	MLP	Time, the initial level of inoculum, the concentration of glucose, fructose, and sucrose	Final biomass level, the concentration of glucose, fructose, and sucrose	Albiol et al. (1995)
<i>Arabidopsis thaliana</i>	Classification of cells in protoplast culture	MLP	Shape and color of cells from digitized image	Type of cell (alive or dead)	Shiotani et al. (1994)
<i>Nicotiana tabacum</i>	Detection of plant viruses in cell culture	MLP	Bioelectric recognition assay (BERA) parameters	Electric response from plant cells	Frossyniotis et al. (2008)
<i>Swertia paniculata</i>	Predicting and optimizing the secondary metabolite production	MLP	Chitosan (CS) and salicylic acid (SA)	Mangiferin, swertiamarin, and amarogentin	Kaur et al. (2020)
<i>Corylus avellane</i>	Predicting and optimizing the secondary metabolite production	MLP-GA	Callus dry weight, total yield of paclitaxel, extracellular paclitaxel portion, extracellular paclitaxel, and intracellular paclitaxel	Elicitor adding day, intercept, cell suspension culture harvesting time, cell extract level, and culture filtrate level	Salehi et al. (2020a)
<i>Bryophyllum tubiflorum</i> , <i>B. daigremontianum</i> , <i>B. daigremontianum</i> × <i>tubiflorum</i>	Predicting and optimizing the secondary metabolite production	Neurofuzzy logic	total phenolic content, radical-scavenging activity, and flavonoid content	eight ions (NO ₃ ⁻ , NH ₄ ⁺ , K ⁺ , PO ₄ ²⁻ , Cl ⁻ , Ca ²⁺ , Mg ²⁺ , and SO ₄ ²⁻), different genotypes, various explants (aerial parts and roots), and solvents	García-Pérez et al. (2020a)
<i>Ipomoea batatas</i>	Classification of embryos	MLP	Embryo area, length, and symmetry, polar coordinates of an embryo's perimeter with respect to its centroid	Type of embryo	Molto and Harrel (1993)
<i>Apium graveolens</i>	Classification of non-embryogenic structures and somatic embryos as well as predicting the time needed for transferring the somatic embryos to the next phase	MLP	The ratio of length to width, area, distance dispersion, and circularity from images of embryos	Type of embryo and time needed for transferring the somatic embryos to the next phase	Uozumi et al. (1993)
<i>Daucus carota</i>	Classification of somatic embryos	MLP	Fourier traits and the embryo size	Type of embryo	Ruan et al. (1997)

Table 3 (continued)

Plant Species	Purpose	Model	Inputs	Output(s)	Reference
<i>Pseudotsuga menziesii</i>	Classification of somatic embryos	MLP	Fourier traits and the length, radius, circularity, width, perimeter, and area of callus	Type of embryo	Zhang et al. (1999)
<i>Trachyspermum ammi</i>	Predicting and optimizing the somatic embryogenesis	MLP	Explant age, the concentrations of kinetin, 2,4-D, and sucrose	The number of the somatic embryo	Niazian et al. (2018b)
<i>Chrysanthemum × grandiflorum</i>	Predicting and optimizing the somatic embryogenesis	ANFIS-NSGA-II	Fructose, 2,4-D, sucrose, BAP, glucose, and light quality	Callogenesis rate, somatic embryogenesis rate, and somatic embryo number	Hesami et al. (2019d)
<i>Chrysanthemum × grandiflorum</i>	Predicting and optimizing the somatic embryogenesis	MLP, SVR-NSGA--II	2,4-D, KIN, and SNP	Callogenesis rate, somatic embryogenesis rate, and somatic embryo number	Hesami et al. (2020b)
<i>Oryza sativa</i>	Predicting microshoot length	MLP and FNN	RGB brightness from the digitized image	Shoot length	Honda et al. (1997)
<i>Saccharum officinarum</i>	Classification of microshoots	MLP	The intensity of spectral brightness and a reflection of the leaf on digitized imaged	Quality of microshoots	Honda et al. (1999)
<i>Gladiolus hybridus</i>	Classification of microshoots	SOM and ANN	Grayscale level, mean brightness, the trichromatic elements (RGB), and the maximum pixel count in luminosity	The ability of the microshoots to form corms	Prasad and Dutta Gupta (2008)
<i>Actinidia deliciosa</i>	Predicting and optimizing shoot proliferation	MLP-GA	The concentrations of sucrose and light intensity	Proliferation rate, shoot number and shoot length	Gago et al. (2010c)
<i>Prunus armeniaca</i>	Predicting and optimizing shoot proliferation	Neurofuzzy logic	Apricot varieties, NO ₃ ⁻ , SO ₄ ²⁻ , NH ₄ ⁺ , K ⁺ , PO ₄ ²⁻ , Ca ²⁺ , Cl ⁻ , Mg ²⁺ , and BAP	Length and number of shoots and productivity	Gago et al. (2011)
<i>Chlorophytum borivilianum</i>	Predicting and optimizing shoot proliferation	GRNN, BPNN, and Elman-BPNN	The concentrations of sucrose, pH of culture medium, inoculum density, and volume of medium per vessel	Fresh weight of regenerated plants	Rizvi et al. (2012)
GF677 hybrid rootstock	Predicting and optimizing shoot proliferation	Neurofuzzy logic	NO ₃ ⁻ , NH ₄ ⁺ , K ⁺ , PO ₄ ²⁻ , Cl ⁻ , Ca ²⁺ , Mg ²⁺ , and SO ₄ ²⁻	Total number of shoots, number of healthy shoots, and number of buds	Alanagh et al. (2014)
<i>Solanum tuberosum</i>	Predicting and optimizing microshoot quality	MLP	Photometric traits derived from the digitized images	Chlorophyll content	Gupta and Pattanayak (2017)
<i>Actinidia deliciosa</i>	Predicting and optimizing shoot proliferation	Neurofuzzy logic	Culture media, number of subcultures, BAP, and GA ₃	Shoot number and shoot length	Arteta et al. (2018)
<i>Pistacia vera</i>	Predicting and optimizing <i>in vitro</i> physiological disorders	Neurofuzzy logic	Two genotypes, BAP, pyridoxine-HCl, thiamine-HCl, nicotinic-acid, glycine, IBA, Γ, NO ₃ ⁻ , PO ₄ ²⁻ , NH ₄ ⁺ , K ⁺ , SO ₄ ²⁻ , Cl ⁻ , Ca ²⁺ , Mg ²⁺ , Fe ²⁺ , Zn ²⁺ , BO ₃ ⁻ , Mn ²⁺ , Cu ²⁺ , Na ⁺ , MoO ₄ ²⁻ , Co ²⁺ , EDTA ⁻	Shoot-tip necrosis, shoot fasciation, basal callus formation, leaf necrosis, epinasty, leaf color, and hyperhydricity	Nezami-Alanagh et al. (2019)
<i>Pistacia vera</i>	Predicting and optimizing shoot proliferation	Neurofuzzy logic	Two genotypes, NO ₃ ⁻ , NH ₄ ⁺ , K ⁺ , PO ₄ ²⁻ , Cl ⁻ , Ca ²⁺ , Γ, Na ⁺ , Mg ²⁺ , Cu ²⁺ , SO ₄ ²⁻ , Fe ²⁺ , BO ₃ ⁻ , Mn ²⁺ , Zn ²⁺ , MoO ₄ ²⁻ , Co ²⁺ , EDTA ⁻	Shoot quality, proliferation rate, shoot length, shoot-tip necrosis, and basal callus	Nezami-Alanagh et al. (2018)
G×N15 rootstock	Predicting and optimizing shoot proliferation	MLP-GA	NO ₃ ⁻ , NH ₄ ⁺ , K ⁺ , PO ₄ ²⁻ , Cl ⁻ , Ca ²⁺ , Mg ²⁺ , and SO ₄ ²⁻	Length and number of shoots, proliferation rate, and quality index	Arab et al. (2016)
Pear rootstock		MLP-GA			

Table 3 (continued)

Plant Species	Purpose	Model	Inputs	Output(s)	Reference
	Predicting and optimizing shoot proliferation		NO_3^- , NH_4^+ , K^+ , PO_4^{2-} , Cl^- , Ca^{2+} , Mg^{2+} , and SO_4^{2-}	Length and number of shoots, proliferation rate, and quality index	Jamshidi et al., (2016)
G×N15 rootstock	Predicting and optimizing shoot proliferation	MLP-GA	BAP, KIN, TDZ, IBA, and NAA	Length and number of shoots, proliferation rate, quality index, and basal callus	Arab et al. (2017)
<i>Pistacia vera</i>	Predicting and optimizing shoot proliferation	Neurofuzzy logic and MLP-GA	Thiamine-HCl, nicotinic-acid, pyridoxine-HCl, glycine, IBA, BA, NO_3^- , NH_4^+ , K^+ , PO_4^{2-} , Cl^- , Ca^{2+} , Mg^{2+} , SO_4^{2-} , Fe^{2+} , BO_3^- , Mn^{2+} , Zn^{2+} , Cu^{2+} , MoO_4^{2-} , Na^+ , Co^{2+} , I^- , EDTA ⁻	Proliferation rate, shoot length, total and healthy fresh weight	Nezami-Alanagh et al. (2017)
<i>Corylus avellana</i>	Predicting and optimizing shoot proliferation	RSM and CHAID	NH_4NO_3 , K_2SO_4 , $\text{Ca}(\text{NO}_3)_2 \cdot 4\text{H}_2\text{O}$, $\text{CaCl}_2 \cdot 2\text{H}_2\text{O}$, KH_2PO_4 , and $\text{MgSO}_4 \cdot 7\text{H}_2\text{O}$	Shoot quality, multiplication rate, shoot length, and callogenesis rate	Akin et al. (2017)
<i>Corylus avellana</i>	Predicting and optimizing shoot proliferation	RSM, CHAID, and exhaustive CHAID	Genotypes, H_3BO_3 , $\text{MnSO}_4 \cdot \text{H}_2\text{O}$, $\text{Zn}(\text{NO}_3)_2 \cdot 6\text{H}_2\text{O}$, $\text{Na}_2\text{MoO}_4 \cdot 2\text{H}_2\text{O}$, and $\text{CuSO}_4 \cdot 5\text{H}_2\text{O}$	Shoot quality, multiplication rate, shoot length, and callogenesis rate	Akin et al. (2018)
<i>Wolffia arrhiza</i> and <i>Lemma minor</i>	Predicting and optimizing shoot proliferation	Solving multinomial task	Genotypes, NH_4NO_3 , $\text{Zn}(\text{NO}_3)_2 \cdot 6\text{H}_2\text{O}$, $\text{Ca}(\text{NO}_3)_2 \cdot 4\text{H}_2\text{O}$, KH_2PO_4 , $\text{CaCl}_2 \cdot 2\text{H}_2\text{O}$, $\text{MgSO}_4 \cdot 7\text{H}_2\text{O}$, $\text{MnSO}_4 \cdot \text{H}_2\text{O}$, K_2SO_4 , H_3BO_3 , $\text{CuSO}_4 \cdot 5\text{H}_2\text{O}$, $\text{Na}_2\text{MoO}_4 \cdot 2\text{H}_2\text{O}$, thiamine-HCl, nicotinic-acid, pyridoxine-HCl, and glycine	Biomass population, total soluble protein, total yield of TSP, dry weight, and protein in dry weight	Khvatkov et al. (2019)
<i>Fragaria bucharica</i> , <i>Fragaria chiloensis</i> , <i>Fragaria x ananassa</i>	Predicting and optimizing shoot proliferation	MARS	Genotypes, NH_4NO_3 , KH_2PO_4 , $\text{Ca}(\text{NO}_3)_2 \cdot 4\text{H}_2\text{O}$, $\text{MgSO}_4 \cdot 7\text{H}_2\text{O}$, $\text{CaCl}_2 \cdot 2\text{H}_2\text{O}$, and K_2SO_4	Shoot quality, multiplication, and leaf color	Akin et al. (2020)
<i>Centella asiatica</i>	Predicting and optimizing shoot proliferation	BPNN	MgSO_4 , CuSO_4 , ZnSO_4 , NO_3^- , and sucrose	Growth indices	Prasad et al. (2017)
Pear rootstock	Predicting and optimizing shoot proliferation	GP and RBF-GA	NO_3^- , NH_4^+ , K^+ , PO_4^{2-} , Cl^- , Ca^{2+} , Mg^{2+} , and SO_4^{2-}	Length and number of shoots, proliferation rate, and quality index	Jamshidi et al. (2019)
<i>Chrysanthemum × grandiflorum</i>	Predicting and optimizing shoot proliferation	RBF-NSGA-II	BAP, IBA, phloroglucinol, and sucrose	Proliferation rate, shoot length, shoot number, and basal callus weight	Hesami et al. (2019c)
<i>Bryophyllum daigremontianum</i> , <i>B. tubiflorum</i> , <i>B. daigremontianum × tubiflorum</i>	Predicting and optimizing shoot organogenesis	Neurofuzzy logic	Genotype, IAA, and BAP	Percentage of direct shoot regeneration, percentage of indirect shoot regeneration, number of direct shoots, number of indirect shoots, percentage of callus formation, percentage of rooting	García-Pérez et al., 2020b
<i>Triticum aestivum</i> L.	Predicting and optimizing shoot proliferation	GRNN-GA	Genotypes, explants, and different concentrations BAP, IAA, KIN, zeatin,	Shoot regeneration frequency	Hesami et al. (2020a)

Table 3 (continued)

Plant Species	Purpose	Model	Inputs	Output(s)	Reference
<i>Cucumis melo</i>	Predicting and optimizing shoot organogenesis	MLP-GA	IBA, 2,4-D, NAA, and CuSO ₄ Agar concentration, light duration, culture temperature, and relative humidity	Percentage of indirect shoot regeneration	Zhang et al., 2020
<i>Pinus taeda</i>	Predicting and optimizing shoot organogenesis	MLP	Total nitrogen concentration and the ratio nitrate: ammonium	Oxidation rate, regeneration rate, callus proliferation rate, buds-forming capacity index, and number of buds per explant,	Barone (2019)
<i>Solanum lycopersicum</i>	Predicting and optimizing androgenesis	MLP	Flower length, genotype, KIN, 2,4-D, cold pretreatment duration, and gum arabic	Callogenesis rate and number of regenerated callus	Niazian et al. (2019)
<i>Glycyrrhiza glabra</i>	Predicting and optimizing hairy root culture	MLP	Explant fresh weight, explant size, number of explants per flask, the month of inoculation, incubation temperature, and pH and volume of the medium	Total mean root fresh weight	Mehrotra et al. (2008)
<i>Glycyrrhiza glabra</i>	Predicting and optimizing hairy root culture	MLP and RNN	volume of medium, pH, inoculum density, and sucrose level	Total mean root fresh weight	Prakash et al. (2010)
<i>Rauwolfia serpentina</i>	Predicting and optimizing hairy root culture	ANN-HMM	The volume of medium per vessel, density of initial inoculum per vessel, sucrose, pH of the medium, and nitrate concentration	Total mean root fresh weight	Mehrotra et al. (2013)
<i>Artemisia annua</i>	Predicting and optimizing hairy root culture	MLP	Mist On/Off cycle time, media volume, initial sucrose concentration in media, initial packing density, and culture time	Total mean root fresh weight	Osama et al. (2013)
<i>Vinca minor</i>	Predicting and optimizing hairy root culture	GRNN and BPNN	Cyclooxygenase inhibitor and hydroxylase/-acetyltransferase elicitors and along with different precursors from secoiridoid and shikimate secoiridoid pools	Growth index, alkaloids, and vincamine	Verma et al. (2016)
<i>Medicago sativa</i>	Predicting and optimizing <i>in vitro</i> rooting	MLP	CO ₂ and sucrose	Leaf number, dry weight, and root initiation stage	Tani et al. (1992)
<i>Vitis vinifera</i> L., cv. Albariño and Mencia	Predicting and optimizing <i>in vitro</i> rooting and acclimatization	MLP-GA	Cultivar, IBA concentrations, and IBA exposure time	Mean number of roots, the mean number of plantlets leaves, the mean height of the plantlets, and the mean number of nodes of the plantlets	Gago et al., (2010a)
G×N15 rootstock	Predicting and optimizing <i>in vitro</i> rooting	MLP-GA	NH ₄ ⁺ , Ca ²⁺ , NO ₃ ⁻ , Cl ⁻ , and K ⁺	Number and length of roots, as well as fresh and dry weight of roots	Arab et al. (2018)
<i>Wrightia tinctoria</i>	Predicting and optimizing <i>in vitro</i> rooting	GP	NAA and charcoal	Basal callus diameter, the rhizogenesis rate, the longest root length, root number, and lateral root number	Mridula et al. (2018)
<i>Vitis vinifera</i> L., cv. Mencia	Predicting and optimizing <i>in vitro</i> rooting and acclimatization	Neurofuzzy logic	The type (IAA, IBA, NAA) and concentration of auxins and the sucrose level	Number and length of roots after 28 days of <i>in vitro</i> rooting, as well as plant height and survival	Gago et al. (2010b)

Table 3 (continued)

Plant Species	Purpose	Model	Inputs	Output(s)	Reference
<i>Actinidia deliciosa</i>	Predicting and optimizing acclimatization	Neurofuzzy logic	Light intensity and sucrose concentrations	percentage after 21 days of acclimatization survival rate, root length, shoot length, <i>in vitro</i> and <i>ex vitro</i> leaves per microshoots, <i>ex vitro/in vitro</i> leaves, microshoot dry weight, percentage of water content, stomatal density, percentage of open stomata, F_v/F_m , F_0 , Chl a + b content, and carotenoids content	Gago et al., (2014)

2,4-D: 2, 4-dichlorophenoxyacetic acid; BAP: 6-benzylaminopurine; IBA: indole-3-butyric acid; KIN: kinetin; TDZ: thidiazuron; NAA: 1-naphthaleneacetic acid; IAA: indole-3-acetic acid; SNP: sodium nitroprusside; GA₃: gibberellic acid; MS: Murashige and Skoog medium; WPM: woody plant medium; ANN: artificial neural network; MLP: multilayer perceptron; RBF: radial basis function; GRNN: generalized regression neural network; ANFIS: adaptive neuro-fuzzy inference systems; SVR: support vector regression; FNN: fuzzy neural network; SOM: self-organizing mapping; BPNN: back propagation neural network; RNN: regression neural network; HMM: hidden Markov model; GP: genetic programming; GA: genetic algorithm; NSGA-II: non-dominated sorting genetic algorithm-II; CHAID: chi-squared automatic interaction Detector; RSM: response surface methodology; MARS: multivariate adaptive regression splines

chloramine B, and silver nitrate and various immersion time were considered inputs and the percentage of contamination and explant viability were taken as outputs. Also, a different range from 9 to 19 of neurons in the hidden layer and function activation (linear and sigmoid) were used for constructing the MLP model. According to Ivashchuk et al. (2018), MLP models with different neurons in the hidden layer and both linear and sigmoid function activations can precisely predict sterilization efficacy. Furthermore, they reported that there were no significant differences between MLP and RBF for modeling and predicting sterilization. In another study, Hesami et al. (2019b) applied MLP-NSGA-II to model and optimize *in vitro* sterilization of chrysanthemum. They considered NaOCl, nano-silver, HgCl₂, AgNO₃, Ca(ClO)₂, H₂O₂, and immersion times as inputs, and explant viability and contamination rate as outputs. They used the 3-layer backpropagation network to run the MLP model. For output and hidden layers transfer functions of the linear (purelin) and the hyperbolic tangent sigmoid (tansig) were used, respectively. Furthermore, a Levenberg-Marquardt algorithm was used to determine the optimum bias and weights. They reported that the MLP model could precisely forecast contamination frequency ($R^2 > 0.97$) and explant viability ($R^2 > 0.94$). Moreover, they considered contamination frequency and explant viability as two objective functions in the NSGA-II process to determine the optimum values of sterilants and immersion time. One thousand generation, 200 initial population, 0.05 mutation rate, 0.7 crossover rate, two-point crossover function, a binary tournament selection function, and the

uniform of mutation function were considered. The ideal point of Pareto was selected such that explant viability and contamination frequency became the maximum and minimum, respectively. Based on sensitivity analysis, Hesami et al. (2019b) reported that NaOCl was the best sterilant for *in vitro* sterilization of chrysanthemum. Also, according to MLP-NSGAII, 1.62% Sodium hypochlorite at 13.96 minutes immersion time can cause the highest explant viability (99.98%) with no contamination. Moreover, according to their validation experiment, *in vitro* sterilization can be precisely predicted and optimized by MLP-NSGA-II.

Microenvironment inside the culture medium containers

Controlling microenvironments, such as level of ventilation, air temperature, CO₂ concentration, as well as light quality and intensity, inside the vessels is a vital requirement for growth and development under *in vitro* culture conditions (Prasad and Gupta 2008a; Tani et al. 1991). A finite element model (FEM) was built through the MLP model for forecasting the distribution of temperature inside the containers. For developing this approach, determined Nusselt numbers (Nu—heat transfer coefficient) were needed for analyzing the temperature distribution via forced convection (Murase et al. 1996; Murase and Okayama 2008). Up to four neurons as input variables, representing the temperatures of nodes determined in the FEM, were selected. Based on the temperature at various airspeeds, MLP determined the Nusselt equation and

then mentioned coefficients were applied to estimate convective heat transmission to containers. About a 5% error was observed between the predicted and observed temperatures, demonstrating relatively accurate modeling. Different heat transfer coefficient combinations were generated because of the randomness of the MLP network input variables. During the training process, the Nusselt equations were accurately and directly defined during the experiment by measuring the temperature (Murase and Okayama 2008). The environmental conditions during plant tissue culture could be studied and predicted with AI methodologies such as the mentioned models.

Callogenesis and cell culture

Callus can be considered an irregular mass of parenchymatous tissue with different types of cells and meristematic sites (Hesami and Daneshvar 2018a; Bhojwani and Dantu 2013). Although callus often presents cellular differentiation, it lacks any organized structure. The multicellular nature of the explants used for callogenesis leads to the cellular heterogeneity of the callus. The callus from the same cell or tissue may present significant differences based on the texture (compact or friable), color, and morphogenic and chemosynthetic potential (Hesami and Daneshvar 2018b). The calli could be friable or compact, light or dark colored, dry, or wet. Moreover, these traits can change with passing time in cultures with culture health, epigenetic or genetic changes, or changes of the medium composition (Hesami et al. 2018a). The callus can be reproduced as undifferentiated cells for an endless period by cyclic subcultures on fresh media or formed to differentiated organs (embryos, shoots, roots) by adjusting and optimizing the composition of the media. Callus cultures have many applications: (i) produce plant-derived metabolites, (ii) somaclonal variations, (iii) provide a system for different physiological and morphogenetic studies, and (iv) provide the material for initiating single cell and suspension cultures (Niazian 2019; Downey et al. 2019; Salehi et al. 2020b). The behavior of callogenesis is non-linear, complex, and time variant that cannot be clarified by simple stepwise algorithms. Also, there are a plethora of factors that affect callus formation (Hesami and Daneshvar 2018a; Munasinghe et al. 2020). Callus formation and classification to the suitable developmental phase has been implemented by using AI models. A combination of MLP and image processing has been used for modeling callogenesis in different plants. Mansouri et al. (2016) applied a supervised feedforward ANN trained with backpropagation methods to model *Cuminum cyminum* L. callogenesis. They selected area, minor axis length, feret diameter, weighted density, and perimeter parameters as input variables and fresh weight and volume of callus as outputs. They reported that the MLP model could precisely predict fresh weight ($R^2 > 0.89$), and volume ($R^2 >$

0.94) of calli. Also, they compared ANN models with multiple linear regression and showed that ANN models had better performance for modeling callogenesis. In another study (Niazian et al. 2018b), an image-processing method was used to investigate the morphological features of embryogenic calli of *Trachyspermum ammi* (L.) Sprague. Different concentrations of kinetin, 2, 4-Dichlorophenoxyacetic acid (2,4-D), and sucrose as well as the age of explants were applied as input variables, and MLP approach was used to forecast the physical features of embryogenic callus. Niazian et al. (2018b) reported that the lower values of MAE and RMSE, and the highest values of R^2 were obtained when all inputs were used to forecast the true density, perimeter, roundness, area, and Feret diameter of the callus in MLP models. Also, according to the sensitivity analysis, the 2,4-D had the highest importance in the callogenesis process that changed the physical characteristics of the embryogenic callus.

The benefit of AI models over traditional methods has been shown during the measurement of plant cell growth (Albiol et al. 1995). The MLP model was constructed with one hidden layer to determine the biomass growth of *Daucus carota* cells. The sigmoid function as the transfer function during training set was considered to build MLP. The input layer was constructed via 8 neurons for data on concentration of sucrose, fructose, and glucose, as well as the time of the initial biomass, while the neurons number in the output layer consisted of 4 neurons for data on the glucose, sucrose, fructose, and final biomass levels. Furthermore, the neurons number in the hidden layer was different and impacted the performance of the network for solving the problem. To validate the ANN results, cell growth was performed in the Celligen reactor with an inoculum of 0.58 g/L. With the established conditions, cells grew exponentially after a 29-day lag phase and biomass reached the maximum value of 4.8 g/L after 49 days. Albiol et al. (1995) reported that changes in biomass and sugar behavior were correctly identified by the network output, even though biomass data in the lag phase are overestimated slightly (with a mean absolute error of 0.94 g/L). Albiol et al. (1995) method is a reliable and useful alternative model to the deterministic mathematical method, even with minimum information and experimental data (Prasad and Gupta 2008a; Zielinska and Kepczynska 2013).

Recently, different AI models have been used for predicting and optimizing *in vitro* secondary metabolite production through callus culture. Kaur et al. (2020) applied MLP for predicting and optimizing mangiferin, swertiamarin, and amarogentin through *Swertia paniculata* Wall shoot culture based on various concentrations of chitosan (CS) and salicylic acid (SA) as input variables. They reported that MLP was able to accurately model and predict *in vitro* secondary metabolite production. In another study, Salehi et al. (2020a) employed MLP-GA for prediction and optimization of *in vitro* secondary metabolite production via

callus and cell culture of *Corylus avellane*. While callus dry weight, total yield of paclitaxel, extracellular paclitaxel portion, extracellular paclitaxel, and intracellular paclitaxel were considered input variables, elicitor adding day, intercept, cell suspension culture harvesting time, cell extract level, and culture filtrate level were chosen as target variables. Salehi et al. (2020a) reported that MLP-GA can be considered a reliable and accurate model for *in vitro* production of secondary metabolites. In another study (García-Pérez et al. 2020a), the neuro-fuzzy model was successfully employed for modeling and optimizing total phenolic content (TPC), radical-scavenging activity (RSA), and flavonoid content (FC) based on 11 input variables including eight ions (NO_3^- , NH_4^+ , K^+ , PO_4^{2-} , Cl^- , Ca^{2+} , Mg^{2+} , and SO_4^{2-}), different genotypes of *Bryophyllum* (*B. tubiflorum*, *B. daigremontianum*, and *B. daigremontianum* × *tubiflorum*), various explants (aerial parts and roots), and solvents. García-Pérez et al. (2020a) reported that neuro-fuzzy logic could serve as a promising approach for the prediction and optimization of secondary metabolite production.

Somatic embryogenesis

In plants, the fusion of male (sperm) and female (egg) gametes causes the production of a zygotic embryo. However, using plant tissue culture, individual somatic cells can be induced to go through similar developmental events to produce embryo-like structures, referred to as “somatic embryos.” The process of the formation of somatic embryos is considered somatic embryogenesis. As with zygotic embryogenesis, somatic embryogenesis consists of different developmental phases (Hesami et al. 2020b). For dicots, these include including globular, heart-shaped, torpedo, and cotyledonary stages, while monocots and gymnosperms go through different stages analogous to their respective zygotic embryogenic processes (Raza et al. 2020).

Somatic embryogenesis is a multi-variable *in vitro* regeneration system controlled by numerous different chemical and physical factors that change with the developmental stage of the explant. It is an economically important and useful propagation system for many plant species and has displayed broad applications in different fields of plant science (Hesami et al. 2019d). Although many plant species have been propagated by somatic embryogenesis, there are several problems such as low germination frequency which limits its widespread applications. *In vitro* somatic embryogenesis and its classification to the appropriate developmental phases have been implemented with the application of AI models. MLP model was used for the classification of non-embryogenic structures and celery somatic embryos. Furthermore, this model forecasted the time required for transferring the somatic embryos to the further developmental phase (Uozumi et al. 1993; Honda et al. 2001). The ratio of length to width, area,

distance dispersion, and circularity from digitalized images of somatic embryos was considered input variables. The developed model was able to classify non-embryogenic callus and somatic embryos, and it recognized different phases (globular, heart-shaped, and torpedo). After fourteen days in the second embryogenesis phase, the MLP model with more than 92% accuracy was successfully predicted the number of regenerated plants from the torpedo and heart structures with more than 92% accuracy.

Due to the fact that the identification of somatic embryos is costly, tedious, and time-consuming, pattern recognition and classification models developed by ANNs are already being broadly applied in the plant *in vitro* culture (Prasad and Gupta 2008a; Zielinska and Kepczynska 2013; Osama et al. 2015). Ruan et al. (1997) used AI technology, with 90% accuracy or higher, to identify the morphological features and patterns of carrot somatic embryos. A hierarchical decision tree including four nodes and three layers was applied to achieve an optimal classification. The developed model classified the somatic embryos into different classes according to the Fourier coefficients, which distinguished the morphological embryogenic structures. In analyzing the Fourier transform approach, these coefficients were achieved. The somatic embryos, in the first node, were divided into four classes: globular phase, torpedo stage, callus, and “other structures”. In the second node, the “other structures” class was grouped into three levels: “secondary”, “heart and oblong” and “cluster and twin”. The ANNs, in the third and fourth nodes, have divided the existed classes from the second node into individual classes. The MLP, which was constructed by a backpropagation method, engaged in each node. The input layer was constructed via 34 neurons representing the Fourier traits and the embryo size, while the neurons number in the output layer related to the number of classes on that node. Furthermore, the neuron number in the middle (hidden) layer was different and impacted the capacity of the model for solving the problem. Zhang et al. (1999) used a similar classification system for *Pseudotsuga menziesii* somatic embryos. In Zhang et al. (1999) method, a distinct and fast Fourier transform was used for transforming the geometric characteristics of embryo's images to numerical grades. Then, a hierarchical decision tree was constructed based on some of the morphological traits and the mentioned values were formed into the model included in two nodes. The MLP, which was constructed by a backpropagation method, engaged in each node. The input layer was constructed via nineteen neurons corresponding to the Fourier traits and the length, radius, circularity, width, perimeter, and area, while the neurons number in the hidden layer consisted of 30 or 25 neurons and influenced the capacity of the network to distinguish between normal and abnormal embryos. Zhang et al. (1999) method was assessed as an applicable model to optimize the conifers somatic embryogenesis and distinguish between normal and abnormal

somatic embryos through this method could help to maximize normal somatic embryogenesis.

Modeling and optimizing the medium composition and environmental conditions is one of the most important methods to maximize somatic embryogenesis. Niaziyan et al. (2018b) used MLP to predict and optimize the number of *Trachyspermum ammi* somatic embryo based on four inputs including the concentrations of kinetin, 2,4-D, and sucrose as well as explant age. They used different learning algorithms (Momentum, Levenberg-Marquart, and Conjugate gradient), activity function (LinearTanhAxon, SigmoidAxon, TainhAxon, and LinearSigmoidAxon), and topology and reported that the number of somatic embryo were predicted with more than 90% accuracy via Levenberg-Marquart algorithm, SigmoidAxon function, and 4-4-3-1 topology. In another study, Hesami et al. (2019d) applied ANFIS-NSGA-II to model and optimize somatic embryogenesis in chrysanthemum. To construct the model, the Gaussian membership function was used, and, also, fructose, 2,4-D, sucrose, 6-Benzylaminopurine (BAP), glucose, and light quality were considered input variables and callus formation rate, somatic embryogenesis rate, and somatic embryo number were considered output data. Furthermore, the number of epochs in the training process was set to ten. They reported that all of the R^2 of both sets (training and validation) of studied parameters were over 92%. Also, they linked the ANFIS to NSGA-II to optimize somatic embryogenesis. 1000 generation, 200 initial population, 0.05 mutation rate, 0.7 crossover rate, two-point crossover function, the uniform of mutation function, and a binary tournament selection function were chosen. Moreover, they considered somatic embryogenesis rate and embryo number as two-objective functions in the NSGA-II process to determine the optimum values of phytohormones, light quality, and carbohydrate sources. The ideal point of Pareto was selected such that somatic embryogenesis frequency and the number of somatic embryos became the maximum. According to their validation experiment, somatic embryogenesis can be precisely optimized by ANFIS-NSGA-II.

Shoot growth and multiplication

The high efficiency of shoot multiplication is required for the success of many micropropagation protocols. Shoot multiplication can be obtained through: (i) direct shoot regeneration from the explant, (ii) indirect regeneration from callus, and (iii) forced axillary branching (Arigundam et al. 2020; Zhang et al. 2020).

The shoot multiplication can be maximized and improved via manipulating the culture media composition. Optimizing the medium composition is too tedious, costly, laborious, and time-consuming. Hence, optimizing and forecasting the culture conditions and the media composition are very useful for selecting the most appropriate conditions and media

composition to obtain the maximum efficiency (Bhojwani and Dantu 2013; Niaziyan 2019; Hesami et al. 2017b; Arteta et al. 2018; García-Pérez et al. 2020b; Hesami et al. 2020a). Different AI models have been successfully applied for forecasting and optimizing shoot regeneration.

Honda et al. (1997) purposed an ANN model to estimate rice microshoot length derived from digitized images. Two various kinds of fuzzy neural network (FNN) were employed for modeling to discriminate between the different zones of the regenerated shoots. The FNN-A approach consisted of one model with three input variables and three outputs, while the FNN-B approach was used to develop three individual models for each output based on those three input variables. The sigmoid function and the backpropagation algorithm were used for the activation of every single neuron and training the model, respectively. The table rules of colors were applied as weights in the trained model and comparisons among them were performed to achieve the relations between the colors of the calli, the differentiated zones, and the media. The complexity range of the relations between the single elements of the color was calculated from the joint weights of the developed model. In this system, the developed model had a greater accuracy (95%) in the distinction of microshoots. The FNN-B was more efficient in recognizing the calli zones than the FNN-A model. A triplex image was rebuilt based on the output data of the FNN-B, which was finally introduced to a two-step process of thinning and the longest way extraction. The length of shoot was determined based on separating the shoot zone from the rest of the image. Elongated microshoots in the regenerating calli were calculated and compared with amounts predicted by the model. Honda et al. (1997) reported that the mean error between the predicted and observed micro shoot lengths was negligible.

One of the most important obstacles in the commercialization of *in vitro* culture protocols is the low quality of microshoots. Microenvironmental factors inside the culture medium such as temperature, light intensity, humidity, and CO₂ concentration have significant effects on the quality and growth of microshoots (Niaziyan 2019; Bhojwani and Dantu 2013). The establishment of an automatic decision-making model that corresponds to the microshoots quality was used to improve the microshoot quality. An ANN model was constructed to qualify and estimate the quality of *Saccharum officinarum* microshoots. The model was built based on photometric parameters which are true estimators assessing the quality of regenerated plants (Honda et al. 1999). The intensity of spectral brightness and a reflection of the leaf on digitized imaged were considered inputs. In a similar study, *Gladiolus hybridus* microshoots were sorted through photometric behavior from the leaf images such as grayscale level, mean brightness, the components of RGB (red, green and blue) coding, and the maximum pixel count in luminosity (Prasad and Gupta 2008b; Mahendra et al. 2004).

Photometric variables were used as inputs to construct different ANN models such as fuzzy ART and Kohonen self-organizing networks (Prasad and Gupta 2008b), as well as ART2-type resonance (Prasad and Gupta 2008b; Mahendra et al. 2004). The ART2 models (Prasad and Gupta 2008b; Mahendra et al. 2004) and fuzzy ART models (Prasad and Gupta 2008b) were trained through the image extraction of 25 microshoots leaves as inputs, while 55 leaves were considered for the testing set. The ART2-type model classified the testing set into two groups in a 19:36 ratio, while the fuzzy ART algorithm was grouped into 7 classes. However, this classification was incorrect because of the lack of a significant correlation with the micro-shoot ability to induce corms (Prasad and Gupta 2008b). The Kohonen self-organizing network is one of the most common types of the network which was used for clustering microshoots. The self-organizing mapping (SOM) with a linear function of distance network has a sextuple topology. A 25-element set was considered input. This type of model is a competitive algorithm that is dependent on the patterns of input. The model generates the output thusly to best rebuild dependence in the input vectors space. The application of the Kohonen self-organizing network in the SOM model organized a testing set of *Gladiolus hybridus* microshoots into 2 classes in a ratio of 28:27 (Prasad and Gupta 2008b). It was verified, via the biological validation of the microshoot classes, which microshoot classes are more competent for forming corms. Only in classes separated by the ART2 model was there a remarkable variation: 36.8 and 69.4% of formed corms, which shows the reliability and accuracy of this algorithm.

Gago et al. (2010b) used MLP-GA with one hidden layer applying a fast adaptive resilient backpropagation learning algorithm and a linear transfer function for modeling and optimizing shoot proliferation of *Actinidia deliciosa*. The concentrations of sucrose and light intensity were considered inputs and, also, proliferation rate, shoot number and shoot length were considered output data. They reported that all of the R^2 of validation set of studied parameters were over 93%. Therefore, they suggested that MLP-GA can be considered an alternative to traditional statistical methods. In another study, Gago et al. (2011) used hybrid neurofuzzy logic technology for modeling and predicting shoot proliferation of apricot through data mining strategy. The input variables were five apricot varieties, different essential mineral components in the culture media (NO_3^- , SO_4^{2-} , NH_4^+ , K^+ , Cl^- , PO_4^{2-} , Ca^{2+} , and Mg^{2+}), and different concentrations of BAP, while the output data were length and number of shoots and productivity (average shoot length \times shoot number). According to Gago et al. (2011), neurofuzzy method can be accurately predicted shoot proliferation and also can be used to understand relationships between different factors involved in shoot proliferation. Furthermore, this method could be expanded and developed by adding additional information on input variables and

output data, such as other growth regulators, additional mineral nutrient levels, additive compounds, physical conditions, etc. (Gago et al. 2011). In another study, Rizvi et al. (2012) applied three ANN models including GRNN, backpropagation neural network (BPNN), and Elman-BPNN for modeling and predicting shoot growth of *Chlorophytum borivillianum* by using bioreactors of large volumes. The concentrations of sucrose, pH of culture medium, inoculum density, and volume of medium per vessel were considered inputs and, also, fresh weight of regenerated plants was considered output. Their results showed that Elman-BPNN predicted fresh weight with better accuracy than GRNN and BPNN. Gupta and Pattanayak (2017) analyzed photometric traits derived from the digitized images of regenerated potato plantlet leaves in order to non-invasively estimate the chlorophyll content by using the ANN strategy. A feed-forward, backpropagation-type network was selected for an input layer (three nodes), with one hidden layer (one node), and an output layer corresponding to the forecasted chlorophyll content. According to Gupta and Pattanayak (2017), training function during the optimization of ANN construction had a significant impact and, also, the best training function on the basis of comparative analysis of root-mean-square error (RMSE) at zero epoch, among the eleven training functions tested, was achieved from “trainlm” function. In another study, Alanagh et al. (2014) employed neurofuzzy logic by considering NO_3^- , NH_4^+ , K^+ , PO_4^{2-} , Cl^- , Ca^{2+} , Mg^{2+} , and SO_4^{2-} as inputs and also the total number of shoots, the number of healthy shoots, and the number of buds as outputs for modeling shoot proliferation of GF677 hybrid rootstocks. To train and construct the model, structural risk minimization (SRM), a number of set densities: 2, set densities: 2, 3, adapt nodes: TRUE, Max. Inputs per SubModel: 4, and Max. Nodes per input: 15 were considered the selection criteria, and also, ridge regression factor: $1e^{-6}$ was considered the minimization parameters. Alanagh et al. (2014) reported that R^2 of testing sets of total shoots, healthy number, and bud number were 77.48, 91.78, and 90.78. Furthermore, they showed the neurofuzzy logic technology can be employed to establish a new medium and optimize in vitro culture protocols. A similar method (neurofuzzy logic) was used for studying in vitro physiological disorders of pistachio rootstocks (Nezami-Alanagh et al. 2019) and designing the culture medium for pistachio rootstocks (Nezami-Alanagh et al. 2018). In similar studies, Arab et al. (2016) and Jamshidi et al. (2016) tried to design a new culture medium for G \times N15 and pear rootstocks, respectively, using the MLP-GA method by considering NO_3^- , NH_4^+ , K^+ , PO_4^{2-} , Cl^- , Ca^{2+} , Mg^{2+} , and SO_4^{2-} as inputs and also growth parameters such as length and number of shoots and proliferation rate as outputs. Moreover, Arab et al. (2017) applied MLP-GA for modeling and optimizing hormonal combination for G \times N15 shoot proliferation. Indole-3-butyric acid (IBA), kinetin (KIN), BAP, thidiazuron (TDZ), and 1-

naphthaleneacetic acid (NAA) were considered inputs; and developed callus weight, number and length of micro-shoots, and the quality index were considered outputs. In these studies (Jamshidi et al. 2016; Arab et al. 2017), the feed-forward back-propagation learning algorithm, the transfer function included hyperbolic tangent sigmoid (tansig) and linear (purelin) functions for the hidden and output layers, respectively, a Levenberg-Marquardt algorithm for back-propagation with a gradient descent with momentum weight and bias learning function, 800–1000 epochs or iterations of the network for training set, and 0.01 level MS error as the performance function were considered to construct and develop MLP model. Furthermore, the roulette wheel as a selection method, 50 initial populations, 500 generations, 0.85 crossover rate, and 0.1 mutation rate were considered in the optimization process using GA. In a similar study, Nezami-Alanagh et al. (2017) employed neurofuzzy logic and MLP-GA to design and optimize culture medium for *Pistacia vera* by considering 26 factors (20 ions, 3 vitamins, and 2 PGRs) as inputs and proliferation rate, shoot length, total and healthy fresh weight as outputs. Nezami-Alanagh et al. (2017) reported that a new pistachio optimized medium caused to 3.73 ± 0.48 shoots per explants which was approximately two-fold of MS and DKW media. According to these studies, ANN-GA can be considered a useful approach for modeling and optimizing shoot proliferation.

Recently, different regression tree data mining techniques, such as chi-squared automatic interaction detector (CHAID), exhaustive CHAID, and classification and regression tree (CART), along with response surface methodology (RSM) have been used for modeling and optimizing macronutrients (Akin et al. 2017) and micronutrients (Akin et al. 2018) in hazelnut. Decision trees display better and superior analytical approaches when the purpose is to study nonlinear and interaction impacts between dependent and independent parameters (Kusiak et al. 2010). Moreover, there is no need for analytical assumptions (like normality assumption) between the response and estimator data. Also, regression tree models are able to control outliers and missing data. The algorithms can analyze ordinal, continuous, and nominal data sets (Thomas et al. 2020). Although RSM is capable of recognizing curvature within the response besides linear effects, it cannot combine nominal data such as cultivars; thus, it constructs separate individual models for each cultivar, which causes the data analysis more time-consuming and complex in plant tissue culture study. In another study, Khvatkov et al. (2019) applied the solving multinomial task from the series of quadratic equations for forecasting and optimizing the culture medium compositions in duckweeds. Also, Akin et al. (2020) applied multivariate adaptive regression splines (MARS) for modeling and predicting macronutrient of culture medium in strawberry. The MARS method is a nonparametric regression model that explains complex nonlinear interactions and

relationships through a sequence of spline functions of the independent values and handle both numerical and categorical data, without the requirement of the normality assumption of the linear models. In another modeling study, Prasad et al. (2017) applied BPNN for maximum biomass accumulation in multiple shoot cultures of *Centella asiatica* where $MgSO_4$, $CuSO_4$, $ZnSO_4$, NO_3 , and sucrose were taken as inputs and growth indices were considered outputs. The input layer was constructed via five input nodes, while the nodes number in the single hidden layer and output layer included three nodes and one node, respectively. Also, one thousand epochs as maximum, output layer learning rate (0.3), initial weight \pm range (0.5), momentum and learning rate (1), and data normalization between 0.1 and 0.9 were used for optimizing the model. They reported that the high correlation between the R^2 of training, testing and validation datasets showed the high efficacy of the ANN model used for better predictability and performance. In another study, Hesami et al. (2019c) used RBF-NSGA-II by considering BAP, sucrose, IBA, and phloroglucinol as inputs and, also, shoot number, proliferation frequency, basal callus weight, and shoot length as outputs, for forecasting and optimizing the composition of shoot proliferation medium for chrysanthemum. They used K -fold cross-validation ($K = 5$) for the validation set. They reported that all of the R^2 of training and validation sets of studied parameters were over 90%. Also, they linked the RBF to NSGA-II to optimize shoot proliferation. One thousand generation, 200 initial population, 0.05 mutation rate, 0.7 crossover rate, two-point crossover function, the uniform of mutation function, and a binary tournament selection function were considered. Moreover, they considered proliferation rate, shoot number, shoot length, and basal callus weight as four objective functions in the NSGA-II process to determine the optimum values of BAP, IBA, phloroglucinol, and sucrose. The ideal point of Pareto was selected such that shoot length, proliferation rate, and shoot number became the maximum, while basal callus weight became the minimum. According to their validation experiment, a medium composition can be precisely optimized by RBF-NSGA-II. Also, Jamshidi et al. (2019) compared the efficiency of multiple linear regression (MLR), RBF, and genetic programming (GP) models in combination with GA for predicting and optimizing the concentrations of medium components for shoot proliferation of pear rootstocks. According to their finding, GP and RBF methods led to more accurate results and can be considered reliable and accurate models for the shoot proliferation stage.

In vitro shoot organogenesis

Shoot organogenesis refers to the differentiation shoots from undifferentiated cells. Shoot organogenesis depends on the fact that the explant cells are highly differentiated because

they are chosen from differentiated parts of plants (Hesami et al. 2019a). When an explant is cultivated in the culture medium, its differentiated cells dedifferentiate to form a callus. Then, the callus cells redifferentiate and produce shoots in response to specific growth factors such as PGRs. Different factors and their interactions play conspicuous roles in plant tissue culture such as PGRs, environmental factors, medium compositions (Piunno et al. 2019; Niaziyan 2019). Barone (2019) employed multiple regression analysis and MLP methods for forecasting and studying the effect of nitrogen in shoot organogenesis of *Pinus taeda*. Total nitrogen contents and the ratio nitrate: ammonium were considered inputs, and also, the regeneration frequency, buds-forming capacity index, callogenesis frequency, number of buds per explant, and oxidation frequency were taken as outputs. MLP was constructed with a layer input with two neurons, one hidden layer with three neurons, and one neuron output. Barone (2019) reported that MLP, in comparison with multiple regression analysis, presented a better precision to model and predict shoot organogenesis, with higher R^2 and lower RMSE for all the studied parameters.

In vitro haploid plant production

Haploid plants can be produced in situ using various techniques such as pollination with irradiated pollen or wide hybridization with closely related, but sexually incompatible, species. In these events, the pollen does not fertilize the egg, but triggers embryo development, resulting in a haploid embryo (Kalinowska et al. 2019). In most cases, the haploid embryos abort and do not produce seed if left to mature in the plant. As such, embryo rescue techniques using plant tissue culture are usually required to recover haploid plants. Alternately, haploid plants can be produced by regenerating plants from maternal gametic tissues *in vitro* through gynogenesis (Kalinowska et al. 2019). Several cells within the egg sac can theoretically develop into haploid plants, including the egg cell, synergids, and antipodal cells, but it is the egg cell that typically develops into a plant when this approach is used. This can be done by culturing isolated ovules, placenta-attached ovules, or even whole flowers from unfertilized plants (Bhojwani and Dantu 2013). However, in most cases, producing haploids from maternal tissues is relatively inefficient and regeneration from paternal gametes through androgenesis is more common. To produce haploid plants from the male gametes, the developmental path of microspores is redirected from developing into mature pollen grains into plant regeneration, typically through somatic embryogenesis (Bhojwani and Dantu 2013). This can be accomplished by isolating the microspores at the appropriate stage of development and culturing them, or the whole anther can be established in culture. Whole anther culture provides a technically simple approach in which the anthers are surface

sterilized and cultured on semi-solid, liquid, or a 2-phase medium. While whole anther culture introduces the potential to regenerate diploid plants from the sporophytic tissues, making the identification and verification of haploid plants more challenging, isolated microspores from many species do not respond as well on their own and it can be the only viable approach in some species. In the case of isolated microspore culture, whole anthers are surface sterilized before being mascerated to liberate the microspores. The microspores are then isolated from the tissues and cultured using various techniques, reducing the probability of regenerating plants from parental tissues. Whether haploids are produced from isolated microspores or whole anther culture, the primary requirement is that the developmental process is re-directed from maturing into pollen toward plant regeneration (Wang et al. 2018). While all regeneration systems are species, and even genotype specific, this is particularly true in the case of haploid plant production and the health and growing conditions of the donor plant, optimal developmental stage of the specific explant, a suitable media composition, appropriate environmental conditions, and the right balance of plant growth regulators, are all essential for success. Another factor that is somewhat unique to microspore culture is the common need for thermal shock to redirect developmental processes. Most microspore regeneration systems require a cold shock, often applied to the flowers prior to culturing, and/or a heat shock applied to the cultured microspores (Kalinowska et al. 2019; Bhojwani and Dantu 2013).

While each species has their own unique requirements for microspore regeneration, the technology has been developed for a variety of taxonomically diverse species and several requirements and patterns can be gleaned (Wang et al. 2018). The most critical factors in developing androgenesis techniques are the identification/use of microspores at the ideal developmental stage and the use of a suitable stress treatment to re-direct the microspores' developmental pathway. Niaziyan et al. (2019) employed the hybrid system of image processing-ANN for a better understanding of callogenesis and androgenesis of tomato. The accurate flower bud length was estimated through the 4',6-diamidino-2-phenylindole (DAPI) analysis and the image processing method; and the results demonstrated that the maximum rate of the mid- to late-uninucleate microspore phase was obtained from the 5–6.9-mm length of flowers. The MLP model was constructed through three training algorithms (momentum, conjugate gradient, and Levenberg–Marquardt) and four activation functions (SigmoidAxon, LinearSigmoidAxon, TanhAxon, and LinearTanhAxon). Flower length, plant cultivar, and different concentrations of Kinetin, 2,4-D, and gum arabic, as well as cold pretreatment duration, were considered inputs, and also, callogenesis frequency and number of differentiated callus were selected as outputs for anther culture of tomato. Also, the developed MLP model was trained and validated with

various neurons in each layer and different numbers of hidden layers. Their results showed that the best MLP model for both outputs was a model with one hidden layer, Levenberg–Marquardt learning algorithm, 12–15 neurons in the first hidden layer, and Tan-Sigmoid transfer function in the hidden layer, based on the R^2 , RMSE, and mean absolute error (MAE).

Hairy root culture

Callus and suspension cultures can be considered a method to produce valuable secondary metabolites. However, secondary metabolite production in a specific organ occurs when callus is induced to organogenesis (Verma et al. 2016). For instance, in *Panax ginseng*, root cultures are essential to produce the secondary metabolite such as saponins. The roots are able to accumulate a wide range of secondary metabolites, indicating their biosynthetic potential. The main problem in *in vitro* secondary metabolite production is the slow growth rate of normal roots (Bhojwani and Dantu 2013). In contrast, *Agrobacterium rhizogenes* which developed hairy roots are characterized by a high growth rate under hormone-free conditions, extensive branching, low doubling time, genetic stability, and ease of maintenance (Baek et al. 2020). Several studies have tried to improve the efficiency of target products through hairy root cultures by manipulation of culture medium, O_2 starvation, physical factors, and metabolic engineering (Baek et al. 2020; Bhojwani and Dantu 2013; Solis-Castañeda et al. 2020; Goswami et al. 2018). These studies showed that optimized condition is a key factor in the success of hairy root culture. Therefore, AI-OA can be considered a powerful tool for this purpose.

Successful attempts have been made in modeling *Glycyrrhiza glabra* hairy root cultures (Mehrotra et al. 2008; Prakash et al. 2010). Mehrotra et al. (2008) employed an MLP to predict the inoculum properties (explant fresh weight, explant size, and the number of explants per flask) and *in vitro* culture conditions (the month of inoculation, incubation temperature, pH of the medium, and volume of the medium per vessel) for optimum root biomass production (the average of root fresh weight). An input layer, an output layer, and a hidden layer including seven neurons were used for constructing the MLP model. A hyperbolic tangent activation function (tansig) and a linear transfer function (purelin) were used for the hidden layer and the output layer, respectively. Mehrotra et al. (2008) reported that the model trained with the traincgb method caused the greatest output range, while the network trained and developed with the trainrp algorithm caused the best-predicted results. In another study, Prakash et al. (2010) employed two ANNs to assess culture parameters such as pH of the medium, inoculum density, sucrose concentration, and volume of medium for *Glycyrrhiza glabra* hairy root culture. The MLP as one of the used models was constructed by using

a linear transition function (purelin), and the sigmoid activation function (logsig), while a regression neural network (RNN) was the second model. Both models were determined to be reliable for forecasting the optimum culture conditions to induce hairy roots; however, the RNN model more accurately predicted. Afterward, the hybrid model, a hidden Markov model (HMM) in combination with ANN, was developed for hairy root culture. The volume of culture medium per vessel, pH, density of initial inoculum per culture vessel, and sucrose and nitrate concentration in the medium were taken as input, and also fresh weight biomass was considered output for modeling *Agrobacterium rhizogenes*—mediated hairy root cultures of *Rauwolfia serpentina*. Mehrotra et al. (2013) reported that pure ANN models and ANN-HMM could be precisely predicted the optimal conditions for the maximum fresh weight production.

In vitro rooting and acclimatization of microshoots

Successful *in vitro* rooting and acclimatization as ultimate stages are very important in plant tissue culture (Mridula et al. 2018; Shukla et al. 2020). Both steps strongly depend on different factors such as auxin concentrations (Gago et al. 2010a; Niazi 2019). Gago et al. (2010a) used MLP-GA to model and optimize relevant factors in *in vitro* rooting and acclimatization of grapevines (*Vitis vinifera* L., cv. Albariño and Mencia). MLP model with the backpropagation learning algorithm and one hidden layer with the asymmetric sigmoid transfer function was constructed by considering cultivar, IBA concentrations, and IBA exposure time as inputs and, also, the mean number of roots, the mean number of plantlets leaves, the mean height of the plantlets, and the average of node number as outputs. For modeling, the data were grouped into three sets including training, testing, and validation sets. Furthermore, GA was employed to find the optimum IBA level and the duration of IBA for *in vitro* rooting and acclimatization. Moreover, the ANN approach allowed the construction and development of a model corresponding both *in vitro* rooting and acclimatization which was able to model and forecast various *in vitro* conditions for both steps simultaneously and for different genotypes. It is possible to construct the model by developing its databases such as environmental factors, other PGRs, and different types of media as new inputs and, also, chlorophyll and carotene contents, stomata analysis, and the weight of plants as new outputs (Gago et al. 2010a). In another study, Arab et al. (2018) employed the MLP-GA approach to predict and optimize a new culture medium for *in vitro* rooting of G×N15 *Prunus* rootstock. NH_4^+ , Ca^{2+} , NO_3^- , Cl^- , and K^+ were taken as inputs and, also, the number and length of roots, as well as fresh and dry weight of roots were considered outputs. The feed-forward backpropagation learning algorithm, the transfer function included hyperbolic tangent sigmoid (tansig) and linear (purelin)

functions for the hidden and output layers, respectively, a Levenberg-Marquardt algorithm for back-propagation with a gradient descent with momentum weight and bias learning function, 800–1000 epochs or iterations of the network for training set, and 0.01 level MS error as the performance function were considered to construct and develop MLP model. Furthermore, the roulette wheel as a selection method, 50 initial populations, 500 generations, 0.85 crossover rate, and 0.1 mutation rate were considered in the optimization process using GA. Arab et al. (2018) reported that all of the R^2 of training and validation processes of studied parameters were over 90%; therefore, they suggested that ANN-GA can be used as a promising methodology for modeling and optimizing in vitro rooting step.

Comprehending the cause-effect relationships between PGRs and culture conditions play an important role in regenerating high-quality plantlets. Modeling and predicting the *in vitro* rooting and acclimatization of grapevine “Albariño” was continued through the neurofuzzy logic (Gago et al. 2010a). The type (Indole-3-acetic acid (IAA), IBA, NAA) and concentration of auxins and the sucrose level in the media were considered the input variables, while the number and length of roots after 28 days of in vitro rooting, as well as plant height and survival percentage after 21 days of acclimatization, were taken as outputs. First, a separate submodel was built for each output and then used for the training set, during which the structural risk minimization approach confirmed to be most precise. Gago et al. (2010a) neurofuzzy technology generated four conditional rule sets for in vitro rooting and acclimatization traits. Furthermore, the model optimization was done by choosing a combination so that the best traits were simultaneously achieved for both *in vitro* rooting and acclimatization. The developed neurofuzzy logic on the basis of these findings provided a general rule: if the concentration of IAA, NAA, or IBA and the level of sucrose is moderate, then length and number of roots, plant height and survival rate in both steps (*in vitro* rooting and acclimatization) obtain the highest value (Gago et al. 2010a). In another study, Gago et al. (2014) neurofuzzy logic employed for modeling the effects of light intensity and sucrose concentrations (inputs) on the survival rate, root length, shoot length, in vitro and ex vitro leaves per plantlet, ex vitro/in vitro leaves, plantlet dry weight, percentage of water content, stomatal density, percentage of open stomata, F_v/F_m , F_0 , Chl a+b content, and carotenoid content in the acclimatization of kiwifruit. To train and construct the model, structural risk minimization (SRM), a number of set densities: 2, set densities: 2, 3, adapt nodes: TRUE, Max. Inputs per SubModel: 4, and Max. Nodes per input: 15 were considered the selection criteria, and, also, ridge regression factor: $1e^{-6}$ was considered the minimization parameters. The mentioned studies showed that neurofuzzy logic method has made it possible to achieve the best and optimal levels and

combination of factors for the highest values of growth and development during in vitro rooting and acclimatization.

Conclusion and future perspective

Different plant tissue culture processes depend upon environmental and genetic factors and are considered nondeterministic, complex, and nonlinear processes. Historically, this has been addressed through sequentially optimizing various factors, a time-consuming and costly endeavor that fails to address the highly interactive nature of the variables. As computational power and sophistication improve, AI and OA are becoming the preferred and more promising approaches for modeling and optimizing complex systems to achieve better results in less time and using fewer resources. AI-OA methods provide a useful scope to analyze in vitro culture data, interpret the gathered data, and give deep insight into the in vitro biological systems. Moreover, the application of AI-OA brings conspicuous benefits due to the AI's ability to take nondeterministic and nonlinear relationships between the information, regardless of their type or origin, and even among incomplete datasets, without the need that the researcher has previous knowledge about these datasets. AI-OA methods can also be applied to develop and construct models that can describe the relationship between biological responses and different factors, which can further be employed to forecast future responses in particular circumstances. Furthermore, the application of AI-OA can now be carried out with a finite number of treatments, which subsequently cuts down the costs and time of developing plant tissue culture protocols on an industrial scale. Finally, adding new input variables and output parameters to the database of the developed model can easily improve the knowledge derived through the application of AI-OA. This may also provide a new perspective aimed at comprehending the regulatory, physiological, and developmental in vitro culture processes.

In the future, combinations of AI models (data fusion strategy) could be applied for developing more precise models that can forecast and optimize the outcome of tissue culture protocols and in vitro biological processes. Although there are no reports regarding the application of data fusion model, random forest, Naive Bayes classifier, singular value decomposition, convolutional neural networks, generative adversarial network, and gradient boosting in plant tissue culture, these models can be employed for solving classification and regression problems in in vitro culture study. Moreover, AI-OA methods could be employed for the automation and mechanization of in vitro plant breeding, genetic engineering, and genome editing technologies such as clustered regularly interspaced short palindromic repeats (CRISPR)-CRISPR-associated protein 9 (Cas9). Gene transformation is a multivariable procedure that many factors such as in vitro

regeneration parameters (PGRs, carbohydrate sources, medium composition, light, and temperature), bacterial optical cell density, antibiotic and chemical stimulants concentrations, and inoculation duration (immersion time), can affect its efficiency. Establishing and developing a suitable strategy for genetic *Agrobacterium*-mediated transformation can be considered a highly complex system because it is critical to comprehend the effect of different factors prompting the T-DNA delivery into various explants. Subsequently, further analyses are essential to check T-DNA integration and stability and to achieve the efficiency parameter of gene transformation. Furthermore, optimizing gene transformation protocol is necessary for being successful in genetic engineering. Therefore, AI-OA as a powerful and reliable strategy can pave the way for developing novel computational methodology in genetic engineering and genome editing.

Acknowledgments Not applicable.

Author contributions All authors have equal contribution in this review and have read and approved the manuscript.

Data availability All processed data are available without restriction upon inquiry.

Compliance with ethical standards This work does not involve any human participation nor live animals performed by any of the listed authors.

Conflict of interest The authors declare that they have no conflict of interest.

References

- Ait-Sahalia Y, Xiu D (2019) Principal component analysis of high-frequency data. *J Am Stat Assoc* 114(525):287–303. <https://doi.org/10.1080/01621459.2017.1401542>
- Akbari M, Deligani VJ (2020) Data driven models for compressive strength prediction of concrete at high temperatures. *Front Struct Civ Eng* 14:1–11. <https://doi.org/10.1007/s11709-019-0593-8>
- Akin M, Eyduran E, Reed BM (2017) Use of RSM and CHAID data mining algorithm for predicting mineral nutrition of hazelnut. *Plant Cell Tiss Org* 128(2):303–316. <https://doi.org/10.1007/s11240-016-1110-6>
- Akin M, Hand C, Eyduran E, Reed BM (2018) Predicting minor nutrient requirements of hazelnut shoot cultures using regression trees. *Plant Cell Tiss Org* 132(3):545–559. <https://doi.org/10.1007/s11240-017-1353-x>
- Akin M, Eyduran SP, Eyduran E, Reed BM (2020) Analysis of macro nutrient related growth responses using multivariate adaptive regression splines. *Plant Cell Tiss Org* 140:661–670. <https://doi.org/10.1007/s11240-019-01763-8>
- Alanagh EN, G-a G, Haddad R, Maleki S, Landín M, Gallego PP (2014) Design of tissue culture media for efficient *Prunus* rootstock micropropagation using artificial intelligence models. *Plant Cell Tiss Org* 117(3):349–359. <https://doi.org/10.1007/s11240-014-0444-1>
- Albiol J, Campmajó C, Casas C, Poch M (1995) Biomass estimation in plant cell cultures: a neural network approach. *Biotechnol Prog* 11(1):88–92. <https://doi.org/10.1021/bp00031a012>
- Alizadeh MR, Nikoo MR (2018) A fusion-based methodology for meteorological drought estimation using remote sensing data. *Remote Sens Environ* 211:229–247. <https://doi.org/10.1016/j.rse.2018.04.001>
- Arab MM, Yadollahi A, Shojaeiyan A, Ahmadi H (2016) Artificial neural network genetic algorithm as powerful tool to predict and optimize in vitro proliferation mineral medium for G× N15 rootstock. *Front Plant Sci* 7:1526. <https://doi.org/10.3389/fpls.2016.01526>
- Arab MM, Yadollahi A, Ahmadi H, Eftekhari M, Maleki M (2017) Mathematical modeling and optimizing of in vitro hormonal combination for G× N15 vegetative rootstock proliferation using Artificial Neural Network-Genetic Algorithm (ANN-GA). *Front Plant Sci* 8:1853. <https://doi.org/10.3389/fpls.2017.01853>
- Arab MM, Yadollahi A, Eftekhari M, Ahmadi H, Akbari M, Khorami SS (2018) Modeling and optimizing a new culture medium for in vitro rooting of G× N15 *Prunus* rootstock using artificial neural network-genetic algorithm. *Sci Rep* 8(1):1–18. <https://doi.org/10.1038/s41598-018-27858-4>
- Araghinejad S, Hosseini-Moghari S-M, Eslamian S (2017) Application of data-driven models in drought forecasting. In: *Handbook of Drought and Water Scarcity*. CRC Press, pp 423–440
- Araghinejad S, Fayaz N, Hosseini-Moghari S-M (2018) Development of a hybrid data driven model for hydrological estimation. *Water Resour Manag* 32(11):3737–3750. <https://doi.org/10.1007/s11269-018-2016-3>
- Arigundam U, Variyath AM, Siow YL, Marshall D, Debnath SC (2020) Liquid culture for efficient in vitro propagation of adventitious shoots in wild *Vaccinium vitis-idaea* ssp. minus (lingonberry) using temporary immersion and stationary bioreactors. *Sci Hortic* 264:109199. <https://doi.org/10.1016/j.scienta.2020.109199>
- Arteta T, Hameg R, Landín M, Gallego P, Barreal M (2018) Neural networks models as decision-making tool for in vitro proliferation of hardy kiwi. *Eur J Hortic Sci* 83(4):259–265. <https://doi.org/10.17660/eJHS.2018/83.4.6>
- Baek S, Ho T-T, Lee H, Jung G, Kim YE, Jeong C-S, Park S-Y (2020) Enhanced biosynthesis of triterpenoids in *Centella asiatica* hairy root culture by precursor feeding and elicitation. *Plant Biotechnol Rep* 14(1):45–53. <https://doi.org/10.1007/s11816-019-00573-w>
- Barone JO (2019) Use of multiple regression analysis and artificial neural networks to model the effect of nitrogen in the organogenesis of *Pinus taeda* L. *Plant Cell Tiss Org* 137(3):455–464. <https://doi.org/10.1007/s11240-019-01581-y>
- Bhojwani SS, Dantu PK (2013) *Plant tissue culture: an introductory text*. Springer
- Biau G, Scomet E, Welbl J (2019) Neural random forests. *Sankhya A* 81(2):347–386. <https://doi.org/10.1007/s13171-018-0133-y>
- Bozorg-Haddad O, Azarnivand A, Hosseini-Moghari S-M, Loáiciga HA (2016) Development of a comparative multiple criteria framework for ranking pareto optimal solutions of a multiobjective reservoir operation problem. *J Irrig Drain Eng* 142(7):04016019. [https://doi.org/10.1061/\(ASCE\)IR.1943-4774.0001028](https://doi.org/10.1061/(ASCE)IR.1943-4774.0001028)
- Bozorg-Haddad O, Azarnivand A, Hosseini-Moghari S-M, Loáiciga Hugo A (2017) WASPAS application and evolutionary algorithm benchmarking in optimal reservoir optimization problems. *J Water Resour Plan Manag* 143(1):04016070. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000716](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000716)
- Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
- Cuba-Díaz M, Rivera-Mora C, Navarrete E, Klagges M (2020) Advances of native and non-native Antarctic species to in vitro conservation: improvement of disinfection protocols. *Sci Rep* 10(1):1–10. <https://doi.org/10.1038/s41598-020-60533-1>
- Deb K, Agrawal S, Pratap A, Meyarivan TA (2000) Fast elitist non-dominated sorting genetic algorithm for multi-objective

- optimization: NSGA-II. In: International conference on parallel problem solving from nature. Springer, Berlin, pp 849–858. https://doi.org/10.1007/3-540-45356-3_83
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE T Evolut Comput* 6(2):182–197. <https://doi.org/10.1109/4235.996017>
- Dezfooli D, Hosseini-Moghari S-M, Ebrahimi K, Araghinejad S (2018) Classification of water quality status based on minimum quality parameters: application of machine learning techniques. *Model Earth Syst Environ* 4(1):311–324. <https://doi.org/10.1007/s40808-017-0406-9>
- Domingues GF, Soares VP, Leite HG, Ferraz AS, Ribeiro CAAS, Lorenzon AS, Marcatti GE, Teixeira TR, de Castro NLM, Mota PHS (2020) Artificial neural networks on integrated multispectral and SAR data for high-performance prediction of eucalyptus biomass. *Comput Electron Agric* 168:105089. <https://doi.org/10.1016/j.compag.2019.105089>
- Dorzhiulov A, James AP (2020) Deep neuro-fuzzy architectures. In: Deep learning classifiers with memristive networks. Springer, Berlin, pp 195–213
- Downey CD, Zoñ J, Jones AMP (2019) Improving callus regeneration of *Miscanthus × giganteus* JM Greef, Deuter ex Hodk., Renvoize ‘M161’ callus by inhibition of the phenylpropanoid biosynthetic pathway. *In Vitro Cell Dev-Pl* 55(1):109–120. <https://doi.org/10.1007/s11627-018-09957-z>
- Ebrahimi H, Dialameh B, Hosseini-Moghari S-M, Ebrahimi A (2020) Optimal conjunctive use of aqua-agriculture reservoir and irrigation canal for paddy fields (case study: Tajan irrigation network, Iran). *Paddy Water Environ* 18(3):499–514. <https://doi.org/10.1007/s10333-020-00797-5>
- Frossyniotis D, Moschopoulou G, Yialouris C (2008) Artificial Neural Network Selection for the Detection of Plant Viruses World. *J Agric Sci* 4(1):114–120
- Gago J, Landín M, Gallego PP (2010a) A neurofuzzy logic approach for modeling plant processes: A practical case of in vitro direct rooting and acclimatization of *Vitis vinifera* L. *Plant Sci* 179(3):241–249. <https://doi.org/10.1016/j.plantsci.2010.05.009>
- Gago J, Martínez-Núñez L, Landín M, Gallego P (2010b) Artificial neural networks as an alternative to the traditional statistical methodology in plant research. *J Plant Physiol* 167(1):23–27. <https://doi.org/10.1016/j.jplph.2009.07.007>
- Gago J, Pérez-Tornero O, Landín M, Burgos L, Gallego PP (2011) Improving knowledge of plant tissue culture and media formulation by neurofuzzy logic: a practical case of data mining using apricot databases. *J Plant Physiol* 168(15):1858–1865. <https://doi.org/10.1016/j.jplph.2011.04.008>
- Gago J, Martínez-Núñez L, Landín M, Flexas J, Gallego PP (2014) Modeling the effects of light and sucrose on in vitro propagated plants: a multiscale system analysis using artificial intelligence technology. *PLoS One* 9(1):e85989. <https://doi.org/10.1371/journal.pone.0085989>
- García-Pérez P, Lozano-Milo E, Landín M, Gallego PP (2020a) Combining medicinal plant in vitro culture with machine learning technologies for maximizing the production of phenolic compounds. *Antioxidants* 9(3):210. <https://doi.org/10.3390/antiox9030210>
- García-Pérez P, Lozano-Milo E, Landín M, Gallego PP (2020b) Machine learning technology reveals the concealed interactions of phytohormones on medicinal plant in vitro organogenesis. *Biomolecules* 10(5):746. <https://doi.org/10.3390/biom10050746>
- George T, Amudha T (2020) Genetic algorithm based multi-objective optimization framework to solve traveling salesman problem. In: *Adv Intel Syst Comput*. Springer, Berlin, pp 141–151
- Goswami M, Akhtar S, Osama K (2018) Strategies for monitoring and modeling the growth of hairy root cultures: An in silico perspective. In: *Hairy roots*. Springer, Berlin, pp 311–327
- Goudarzi A, Li Y, Xiang J (2020) A hybrid non-linear time-varying double-weighted particle swarm optimization for solving non-convex combined environmental economic dispatch problem. *Appl Soft Comput* 86:105894. <https://doi.org/10.1016/j.asoc.2019.105894>
- Gray DJ, Trigiano RN (2018) Introduction to plant tissue culture. In: *Plant tissue culture concepts and laboratory exercises*. Routledge, pp 3–7
- Guangrong H, Dehui D, Weilian H, Jiaxin J (2008) Optimization of medium composition for thermostable protease production by *Bacillus* sp. HS08 with a statistical method. *Afr J Biotechnol* 7(8)
- Gupta SD, Pattanayak A (2017) Intelligent image analysis (IIA) using artificial neural network (ANN) for non-invasive estimation of chlorophyll content in micropropagated plants of potato. *Vitro Cell Dev-Pl* 53(6):520–526. <https://doi.org/10.1007/s11627-017-9825-6>
- Haddad Omid B, Hosseini-Moghari S-M, Loáiciga Hugo A (2016) Biogeography-Based Optimization Algorithm for Optimal Operation of Reservoir Systems. *J Water Resour Plan Manag* 142(1):04015034. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000558](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000558)
- Hararuk O, Zwart JA, Jones SE, Prairie Y, Solomon CT (2018) Model-data fusion to test hypothesized drivers of lake carbon cycling reveals importance of physical controls. *J Geophys Res Biogeosci* 123(3):1130–1142. <https://doi.org/10.1002/2017JG004084>
- Hesami M, Daneshvar MH (2018a) In vitro adventitious shoot regeneration through direct and indirect organogenesis from seedling-derived hypocotyl segments of *Ficus religiosa* L.: an important medicinal plant. *HortScience* 53(1):55–61. <https://doi.org/10.21273/HORTSCI12637-17>
- Hesami M, Daneshvar MH (2018b) Indirect organogenesis through seedling-derived leaf segments of *Ficus religiosa*-a multipurpose woody medicinal plant. *J Crop Sci Biotechnol* 21(2):129–136. <https://doi.org/10.1007/s12892-018-0024-0>
- Hesami M, Daneshvar MH, Lotfi-Jalalabadi A (2017a) Effect of sodium hypochlorite on control of in vitro contamination and seed germination of *Ficus religiosa*. *Iranian J Plant Physiol* 7(4):2157–2162. <https://doi.org/10.22034/IJPP.2017.537980>
- Hesami M, Daneshvar MH, Lotfi A (2017b) In vitro shoot proliferation through cotyledonary node and shoot tip explants of *Ficus religiosa* L. *Plant Tissue Cult Biotechnol* 27(1):85–88. <https://doi.org/10.3329/ptcb.v27i1.35017>
- Hesami M, Naderi R, Yoosefzadeh-Najafabadi M, Rahmati M (2017c) Data-driven modeling in plant tissue culture. *J Appl Environ Biol Sci* 7(8):37–44
- Hesami M, Daneshvar MH, Yoosefzadeh-Najafabadi M (2018a) Establishment of a protocol for in vitro seed germination and callus formation of *Ficus religiosa* L., an important medicinal plant. *Jundishapur J Nat Pharm Prod* 13(4):e62682. <https://doi.org/10.5812/jjnpp.62682>
- Hesami M, Naderi R, Yoosefzadeh-Najafabadi M (2018b) Optimizing sterilization conditions and growth regulator effects on in vitro shoot regeneration through direct organogenesis in *Chenopodium quinoa*. *BioTechnologia* 99(1):49–57. <https://doi.org/10.5114/bta.2018.73561>
- Hesami M, Naderi R, Yoosefzadeh-Najafabadi M, Maleki M (2018c) In vitro culture as a powerful method for conserving Iranian ornamental geophytes. *BioTechnologia* 99(1):73–81. <https://doi.org/10.5114/bta.2018.73563>
- Hesami M, Daneshvar MH, Yoosefzadeh-Najafabadi M (2019a) An efficient in vitro shoot regeneration through direct organogenesis from seedling-derived petiole and leaf segments and acclimatization of *Ficus religiosa*. *J For Res* 30(3):807–815. <https://doi.org/10.1007/s11676-018-0647-0>
- Hesami M, Naderi R, Tohidfar M (2019b) Modeling and optimizing in vitro sterilization of chrysanthemum via multilayer perceptron-

- non-dominated sorting genetic algorithm-II (MLP-NSGAI). *Front Plant Sci* 10:282. <https://doi.org/10.3389/fpls.2019.00282>
- Hesami M, Naderi R, Tohidfar M (2019c) Modeling and optimizing medium composition for shoot regeneration of *Chrysanthemum* via radial basis function-non-dominated sorting genetic algorithm-II (RBF-NSGAI). *Sci Rep* 9(1):18237. <https://doi.org/10.1038/s41598-019-54257-0>
- Hesami M, Naderi R, Tohidfar M, Yoosefzadeh-Najafabadi M (2019d) Application of adaptive neuro-fuzzy inference system-non-dominated sorting genetic algorithm-II (ANFIS-NSGAI) for modeling and optimizing somatic embryogenesis of *Chrysanthemum*. *Front Plant Sci* 10:869. <https://doi.org/10.3389/fpls.2019.00869>
- Hesami M, Condori-Apfata JA, Valderrama Valencia M, Mohammadi M (2020a) Application of artificial neural network for modeling and studying in vitro genotype-independent shoot regeneration in Wheat. *Appl Sci* 10(15):5370. <https://doi.org/10.3390/app10155370>
- Hesami M, Naderi R, Tohidfar M, Yoosefzadeh-Najafabadi M (2020b) Development of support vector machine-based model and comparative analysis with artificial neural network for modeling the plant tissue culture procedures: effect of plant growth regulators on somatic embryogenesis of *chrysanthemum*, as a case study. *Plant Methods* 16(1):112. <https://doi.org/10.1186/s13007-020-00655-9>
- Hildebrandt AC, Riker A, Duggar B (1946) The influence of the composition of the medium on growth in vitro of excised tobacco and sunflower tissue cultures. *Am J Bot*:591–597. <https://doi.org/10.2307/2437399>
- Holland JH (1992) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press
- Honda H, Takikawa N, Noguchi H, Hanai T, Kobayashi T (1997) Image analysis associated with a fuzzy neural network and estimation of shoot length of regenerated rice callus. *J Ferment Bioeng* 84(4):342–347. [https://doi.org/10.1016/S0922-338X\(97\)89256-2](https://doi.org/10.1016/S0922-338X(97)89256-2)
- Honda H, Ito T, Yamada J, Hanai T, Matsuoka M, Kobayashi T (1999) Selection of embryogenic sugarcane callus by image analysis. *J Biosci Bioeng* 87(5):700–702. [https://doi.org/10.1016/S1389-1723\(99\)80138-8](https://doi.org/10.1016/S1389-1723(99)80138-8)
- Honda H, Liu C, Kobayashi T (2001) Large-scale plant micropropagation. In: *Plant Cells*. Springer, Berlin, pp 157–182
- Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. *Neural Netw* 2(5):359–366
- Hosseini-Moghari SM, Araghinejad S (2015) Monthly and seasonal drought forecasting using statistical neural networks. *Environ Earth Sci* 74(1):397–412. <https://doi.org/10.1007/s12665-015-4047-x>
- Hosseini-Moghari S-M, Morovati R, Moghadas M, Araghinejad S (2015) Optimum operation of reservoir using two evolutionary algorithms: imperialist competitive algorithm (ICA) and cuckoo optimization algorithm (COA). *Water Resour Manag* 29(10):3749–3769. <https://doi.org/10.1007/s11269-015-1027-6>
- Hosseini-Moghari S-M, Araghinejad S, Azarnivand A (2017) Drought forecasting using data-driven methods and an evolutionary algorithm. *Model Earth Syst Environ* 3(4):1675–1689. <https://doi.org/10.1007/s40808-017-0385-x>
- Ivashchuk OA, Fedorova V, Shcherbinina NV, Maslova EV, Shamraeva E (2018) Microclonal propagation of plant process modeling and optimization of its parameters based on neural network. *Drug Invent Today* 10(3):3170–3175
- Jamshidi S, Yadollahi A, Ahmadi H, Arab M, Eftekhari MJF (2016) Predicting in vitro culture medium macro-nutrients composition for pear rootstocks using regression analysis and neural network models. *Front Plant Sci* 7:274. <https://doi.org/10.3389/fpls.2016.00274>
- Jamshidi S, Yadollahi A, Arab MM, Soltani M, Eftekhari M, Sabzalipoor H, Sheikhi A, Shiri J (2019) Combining gene expression programming and genetic algorithm as a powerful hybrid modeling approach for pear rootstocks tissue culture media formulation. *Plant Methods* 15(1):136. <https://doi.org/10.1186/s13007-019-0520-y>
- Kalinowska K, Chamas S, Unkel K, Demidov D, Lermontova I, Dresselhaus T, Kumlehn J, Dunemann F, Houben A (2019) State-of-the-art and novel developments of in vivo haploid technologies. *Theor Appl Genet* 132(3):593–605. <https://doi.org/10.1007/s00122-018-3261-9>
- Kaur P, Gupta RC, Dey A, Malik T, Pandey DK (2020) Optimization of salicylic acid and chitosan treatment for bitter secoiridoid and xanthone glycosides production in shoot cultures of *Swertia paniculata* using response surface methodology and artificial neural network. *BMC Plant Biol* 20(1):225. <https://doi.org/10.1186/s12870-020-02410-7>
- Khvatkov P, Chernobrovkina M, Okuneva A, Dolgov S (2019) Creation of culture media for efficient duckweeds micropropagation (*Wolffia arrhiza* and *Lemna minor*) using artificial mathematical optimization models. *Plant Cell Tiss Org* 136(1):85–100. <https://doi.org/10.1007/s11240-018-1494->
- Kranjčić N, Medak D, Župan R, Rezo M (2019) Support Vector Machine Accuracy Assessment for Extracting Green Urban Areas in Towns. *Remote Sens* 11(6):655. <https://doi.org/10.3390/rs11060655>
- Kumari B, Swarnkar T (2020) Importance of data standardization methods on stock indices prediction accuracy. In: *Advanced computing and intelligent engineering*. Springer, Berlin, pp 309–318. https://doi.org/10.1007/978-981-15-1081-6_26
- Kusiak A, Li M, Zhang Z (2010) A data-driven approach for steam load prediction in buildings. *Appl Energy* 87(3):925–933. <https://doi.org/10.1016/j.apenergy.2009.09.004>
- Lan T, Tong C, Yu H, Shi X, Luo L (2020) Nonlinear process monitoring based on decentralized generalized regression neural networks. *Expert Syst Appl* 150:113273. <https://doi.org/10.1016/j.eswa.2020.113273>
- Lin J, Zhao Y, Watson D, Chen C (2020) The radial basis function differential quadrature method with ghost points. *Math Comput Simul* 173:105–114. <https://doi.org/10.1016/j.matcom.2020.01.006>
- Maddalena ET, Lian Y, Jones CN (2020) Data-driven methods for building control—a review and promising future directions. *Control Eng Pract* 95:104211. <https://doi.org/10.1016/j.conengprac.2019.104211>
- Mahendra, Prasad V, Gupta SD (2004) Trichromatic sorting of in vitro regenerated plants of gladiolus using adaptive resonance theory. *Curr Sci* 87(3):348–353
- Mansouri A, Fadavi A, Mortazavian SMM (2016) An artificial intelligence approach for modeling volume and fresh weight of callus—A case study of cumin (*Cuminum cyminum* L.). *J Theor Biol* 397:199–205. <https://doi.org/10.1016/j.jtbi.2016.03.009>
- Mehrotra S, Prakash O, Mishra B, Dwevedi B (2008) Efficiency of neural networks for prediction of in vitro culture conditions and inoculum properties for optimum productivity. *Plant Cell Tiss Org* 95(1):29–35. <https://doi.org/10.1007/s11240-008-9410-0>
- Mehrotra S, Prakash O, Khan F, Kukreja A (2013) Efficiency of neural network-based combinatorial model predicting optimal culture conditions for maximum biomass yields in hairy root cultures. *Plant Cell Rep* 32(2):309–317. <https://doi.org/10.1007/s00299-012-1364-3>
- Meng T, Jing X, Yan Z, Pedrycz W (2020) A survey on machine learning for data fusion. *Inf Fusion* 57:115–129. <https://doi.org/10.1016/j.inffus.2019.12.001>
- Molto E, Harrell RC (1993) Neural network classification of sweet potato embryos. In: *Optics in Agriculture and Forestry*. International Society for Optics and Photonics, pp 239–249
- Moravej M (2017) Discussion of “Modified Firefly Algorithm for Solving Multireservoir Operation in Continuous and Discrete

- Domains” by Irene Garousi-Nejad, Omid Bozorg-Haddad, and Hugo A. Loáiciga. *J Water Resour Plan Manag* 143(10): 07017004. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000644](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000644)
- Moravej M (2018) Discussion of Application of the Firefly Algorithm to Optimal Operation of Reservoirs with the Purpose of Irrigation Supply and Hydropower Production by Irene Garousi-Nejad, Omid Bozorg-Haddad, Hugo A. Lo, and Miguel A. Mari. *J Irrig Drain Eng* 144(1):07017019. [https://doi.org/10.1061/\(ASCE\)IR.1943-4774.0001259](https://doi.org/10.1061/(ASCE)IR.1943-4774.0001259)
- Moravej M, Hosseini-Moghari S-M (2016) Large scale reservoirs system operation optimization: the interior search algorithm (ISA) approach. *Water Resour Manag* 30(10):3389–3407. <https://doi.org/10.1007/s11269-016-1358-y>
- Moravej M, Amani P, Hosseini-Moghari S-M (2020) Groundwater level simulation and forecasting using interior search algorithm-least square support vector regression (ISA-LSSVR). *Groundw Sustain Dev* 11:1–18
- Mridula MR, Nair AS, Kumar KS (2018) Genetic programming based models in plant tissue culture: an addendum to traditional statistical approach. *PLoS Comput Biol* 14(2):e1005976. <https://doi.org/10.1371/journal.pcbi.1005976>
- Munasinghe SP, Somaratne S, Weerakoon SR, Ranasinghe C (2020) Prediction of chemical composition for callus production in *Gyrinops walla* Gaertner through machine learning. *Inf Process Agric* 7(2):1–12. <https://doi.org/10.1016/j.inpa.2019.12.001>
- Murase H, Okayama T (2008) Intelligent inverse analysis for temperature distribution in a plant culture vessel. In: *Plant Tissue Culture Engineering*. Springer, Berlin, pp 373–394
- Murase H, Tani A, Honami N, Takigawa H, Nishiura Y (1996) Inverse technique for analysis of convective heat transfer over the surface of plant culture vessel. *T ASABE* 39(6):2277–2282. <https://doi.org/10.13031/2013.27737>
- Murashige T, Skoog F (1962) A revised medium for rapid growth and bio assays with tobacco tissue cultures. *Physiol Plant* 15(3):473–497. <https://doi.org/10.1111/j.1399-3054.1962.tb08052.x>
- Nezami-Alanagh E, Garoosi G-A, Maleki S, Landin M, Gallego PP (2017) Predicting optimal in vitro culture medium for *Pistacia vera* micropropagation using neural networks models. *Plant Cell Tiss Org* 129(1):19–33. <https://doi.org/10.1007/s11240-016-1152-9>
- Nezami-Alanagh E, Garoosi G-A, Landin M, Gallego PP (2018) Combining DOE with neurofuzzy logic for healthy mineral nutrition of pistachio rootstocks in vitro culture. *Front Plant Sci* 9:1474. <https://doi.org/10.3389/fpls.2018.01474>
- Nezami-Alanagh E, Garoosi G-A, Landin M, Gallego PP (2019) Computer-based tools provide new insight into the key factors that cause physiological disorders of pistachio rootstocks cultured in vitro. *Sci Rep* 9(1):1–15. <https://doi.org/10.1038/s41598-019-46155-2>
- Niazian M (2019) Application of genetics and biotechnology for improving medicinal plants. *Planta* 249(4):953–973. <https://doi.org/10.1007/s00425-019-03099-1>
- Niazian M, Sadat-Noori SA, Abdipour M (2018a) Modeling the seed yield of Ajowan (*Trachyspermum ammi* L.) using artificial neural network and multiple linear regression models. *Ind Crop Prod* 117: 224–234. <https://doi.org/10.1016/j.indcrop.2018.03.013>
- Niazian M, Sadat-Noori SA, Abdipour M, Tohidfar M, Mortazavian SMM (2018b) Image processing and artificial neural network-based models to measure and predict physical properties of embryogenic callus and number of somatic embryos in ajowan (*Trachyspermum ammi* (L.) Sprague). In *Vitro Cell Dev-PI* 54(1): 54–68. <https://doi.org/10.1007/s11627-017-9877-7>
- Niazian M, Shariatpanahi ME, Abdipour M, Oroojloo M (2019) Modeling callus induction and regeneration in an anther culture of tomato (*Lycopersicon esculentum* L.) using image processing and artificial neural network method. *Protoplasma* 256(5):1317–1332. <https://doi.org/10.1007/s00709-019-01379-x>
- Osama K, Somvanshi P, Pandey AK, Mishra BN (2013) Modelling of nutrient mist reactor for hairy root growth using artificial neural network. *Eur J Sci Res* 97(4):516–526
- Osama K, Mishra BN, Somvanshi P (2015) Machine learning techniques in plant biology. In: *PlantOmics: The Omics of plant science*. Springer, Berlin, pp 731–754. https://doi.org/10.1007/978-81-322-2172-2_26
- Phillips GC, Garda M (2019) Plant tissue culture media and practices: an overview. *Vitro Cell Dev-PI* 55(3):242–257. <https://doi.org/10.1007/s11627-019-09983-5>
- Piunno KF, Golenia G, Boudko EA, Downey C, Jones AMP (2019) Regeneration of shoots from immature and mature inflorescences of *Cannabis sativa*. *Can J Plant Sci* 99(4):556–559. <https://doi.org/10.1139/cjps-2018-0308>
- Prado F, Minutolo MC, Kristjanpoller W (2020) Forecasting based on an ensemble autoregressive moving average-adaptive neuro-fuzzy inference system—neural network-genetic algorithm framework. *Energy* 197:117159. <https://doi.org/10.1016/j.energy.2020.117159>
- Prakash O, Mehrotra S, Krishna A, Mishra BN (2010) A neural network approach for the prediction of in vitro culture parameters for maximum biomass yields in hairy root cultures. *J Theor Biol* 265(4):579–585. <https://doi.org/10.1016/j.jtbi.2010.05.020>
- Prasad V, Gupta SD (2008a) Applications and potentials of artificial neural networks in plant tissue culture. In: *Plan Tissue Culture Engineering*. Springer, Berlin, pp 47–67
- Prasad V, Gupta SD (2008b) Photometric clustering of regenerated plants of gladiolus by neural networks and its biological validation. *Comput Electron Agric* 60(1):8–17. <https://doi.org/10.1016/j.compag.2007.05.006>
- Prasad A, Prakash O, Mehrotra S, Khan F, Mathur AK, Mathur A (2017) Artificial neural network-based model for the prediction of optimal growth and culture conditions for maximum biomass accumulation in multiple shoot cultures of *Centella asiatica*. *Protoplasma* 254(1): 335–341. <https://doi.org/10.1007/s00709-016-0953-3>
- Raj S, Saudagar P (2019) Plant cell culture as alternatives to produce secondary metabolites. In: *Natural bio-active compounds*. Springer, Berlin, pp 265–286
- Raza G, Singh MB, Bhalla PL (2020) Somatic embryogenesis and plant regeneration from commercial soybean cultivars. *Plants* 9(1):38. <https://doi.org/10.3390/plants9010038>
- Rizvi Z, Mishra P, Roy S, Kukreja A, Sharma A (2012) Application of artificial neural networks for predicting maximum in vitro shoot biomass production of safed musli (*Chlorophytum borivilianum*). *J Med Diag Meth* 1:1–6. <https://doi.org/10.4172/scientificreports.464>
- Ruan R, Xu J, Zhang C, Chi CM, Hu WS (1997) Classification of plant somatic embryos by using neural network classifiers. *Biotechnol Prog* 13(6):741–746. <https://doi.org/10.1021/bp9700972>
- Salehi M, Farhadi S, Moieni A, Safaie N, Ahmadi H (2020a) Mathematical modeling of growth and paclitaxel biosynthesis in *Corylus avellana* cell culture responding to fungal elicitors using multilayer perceptron-genetic algorithm. *Front Plant Sci* 11:1148. <https://doi.org/10.3389/fpls.2020.01148>
- Salehi M, Moieni A, Safaie N, Farhadi S (2020b) Whole fungal elicitors boost paclitaxel biosynthesis induction in *Corylus avellana* cell culture. *PLoS One* 15(7):e0236191. <https://doi.org/10.1371/journal.pone.0236191>
- Sheikhi A, Mirdehghan SH, Arab MM, Eftekhari M, Ahmadi H, Jamshidi S, Gheysarbigi S (2020) Novel organic-based postharvest sanitizer formulation using Box Behnken design and mathematical modeling approach: A case study of fresh pistachio storage under modified atmosphere packaging. *Postharvest Biol Technol* 160: 111047. <https://doi.org/10.1016/j.postharvbio.2019.111047>
- Shiotani S, Fukuda T, Arai F, Takeuchi N, Sasaki K, Kinoshita T (1994) Cell recognition by image processing: recognition of dead or living

- plant cells by neural network. JSME Int J 37(1):202–208. <https://doi.org/10.1299/jsmec1993.37.202>
- Shukla MR, Piumno K, Saxena PK, Jones AMP (2020) Improved in vitro rooting in liquid culture using a two piece scaffold system. Eng Life Sci 20:1–7. <https://doi.org/10.1002/elsc.201900133>
- Silva JCF, Teixeira RM, Silva FF, Brommonschenkel SH, Fontes EP (2019) Machine learning approaches and their current application in plant molecular biology: a systematic review. Plant Sci 284:37–47. <https://doi.org/10.1016/j.plantsci.2019.03.020>
- Solis-Castañeda GJ, Zamilpa A, Cabañas-García E, Bahena SM, Pérez-Molphe-Balch E, Gómez-Aguirre YA (2020) Identification and quantitative determination of feruloyl-glucoside from hairy root cultures of *Turbinicarpus lophophoroides* (Werderm.) Buxb. & Backeb.(Cactaceae). In Vitro Cell Dev-Pl 56(1):8–17. <https://doi.org/10.1007/s11627-019-10029-z>
- Specht DF (1991) A general regression neural network. IEEE T Neur Net 2(6):568–576. <https://doi.org/10.1109/72.97934>
- Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. Evol Comput 2(3): 221–248. <https://doi.org/10.1162/evco.1994.2.3.221>
- Tang Z, Fishwick PA (1993) Feedforward neural nets as models for time series forecasting. ORSA J Comput 5(4):374–385. <https://doi.org/10.1287/ijoc.5.4.374>
- Tani A, Kiyota M, Taira T, Alga I (1991) Effect of light intensity and aeration on temperature distribution inside plant culture vessel. Plant Cell Tiss Org 8(2):133–135
- Tani A, Murase H, Kiyota M, Honami N (1992) Growth simulation of alfalfa cuttings in vitro by Kalman filter neural network. Acta Hort 319:671–676. <https://doi.org/10.17660/ActaHortic.1992.319.108>
- Teixeira da Silva JA, Kulus D, Zhang X, Zeng S, Ma G, Piqueras A (2016) Disinfection of explants for saffron (*Crocus sativus* L.) tissue culture. Environ Exp Bot 14(4):183–198
- Thomas T, Vijayaraghavan AP, Emmanuel S (2020) Applications of decision trees. In: Machine learning approaches in cyber security analytics. Springer, Berlin, pp 157–184
- Tong S, Zhang X, Tong Z, Wu Y, Tang N, Zhong W (2020) Online ash fouling prediction for boiler heating surfaces based on wavelet analysis and support vector regression. Energies 13(1):59. <https://doi.org/10.3390/en13010059>
- Uozumi N, Yoshino T, Shiotani S, Suehara K-I, Arai F, Fukuda T, Kobayashi T (1993) Application of image analysis with neural network for plant somatic embryo culture. J Ferment Bioeng 76(6): 505–509. [https://doi.org/10.1016/0922-338X\(93\)90249-8](https://doi.org/10.1016/0922-338X(93)90249-8)
- Vapnik V (2013) The nature of statistical learning theory. Springer science & business media, Berlin
- Verma P, Anjum S, Khan SA, Roy S, Odstreilik J, Mathur AK (2016) Envisaging the regulation of alkaloid biosynthesis and associated growth kinetics in hairy roots of *Vinca minor* through the function of artificial neural network. Appl Biochem Biotechnol 178(6):1154–1166. <https://doi.org/10.1007/s12010-015-1935-1>
- Wanas N, Auda G, Kamel MS, Karray F 1998. On the optimal number of hidden nodes in a neural network. In: Conference Proceedings. IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No. 98TH8341), . IEEE, pp 918-921. doi: <https://doi.org/10.1109/CCECE.1998.685648>
- Wang G-F, Qin H-Y, Sun D, Fan S-T, Yang Y-M, Wang Z-X, Xu P-L, Zhao Y, Liu Y-X, Ai J (2018) Haploid plant regeneration from hardy kiwifruit (*Actinidia arguta* Planch.) anther culture. Plant Cell Tiss Org 134(1):15–28. <https://doi.org/10.1007/s11240-018-1396-7>
- Wong FS (1991) Time series forecasting using backpropagation neural networks. Neurocomputing 2(4):147–159. [https://doi.org/10.1016/0925-2312\(91\)90045-D](https://doi.org/10.1016/0925-2312(91)90045-D)
- Ying X, Tang B, Zhou C (2020) Nursing scheme based on back propagation neural network and probabilistic neural network in chronic kidney disease. J Med Imaging Health Inform 10(2):416–421. <https://doi.org/10.1166/jmhi.2020.2879>
- Yun Y, Chuluunsukh A, Gen M (2020) Sustainable closed-loop supply chain design problem: a hybrid genetic algorithm approach. Mathematics 8(1):84. <https://doi.org/10.3390/math8010084>
- Zhang C, Timmis R, Hu W-S (1999) A neural network based pattern recognition system for somatic embryos of Douglas fir. Plant Cell Tiss Org 56(1):25–35. <https://doi.org/10.1023/A:1006287917534>
- Zhang Q, Deng D, Dai W, Li J, Jin X (2020) Optimization of culture conditions for differentiation of melon based on artificial neural network and genetic algorithm. Sci Rep 10(1):3524. <https://doi.org/10.1038/s41598-020-60278-x>
- Zielinska S, Kepczynska E (2013) Neural modeling of plant tissue cultures: a review. BioTechnologia 94(3):253–268. <https://doi.org/10.5114/bta.2013.46419>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.