



Original Article

Assessing the suitability of soft computing approaches for forest fires prediction

Samaher Al_Janabi^a, Ibrahim Al_Shourbaji^{b,*}, Mahdi A. Salman^a^a Department of Computer Science, Faculty of Science for Women (SCIW), University of Babylon, Iraq^b Department of Computer Network, Faculty of Computer Science and Information System, University of Jazan, Jazan, Saudi Arabia

ARTICLE INFO

Article history:

Received 5 May 2017

Revised 7 September 2017

Accepted 16 September 2017

Available online 20 September 2017

Keywords:

Forest fires

Soft computing

Prediction

Principle component analysis

Particle swarm optimization

Cascade correlation network

Multilayer perceptron neural network

Polynomial neural networks

Radial basis function

Support vector machine

ABSTRACT

Forest fires present one of the main causes of environmental hazards that have many negative results in different aspect of life. Therefore, early prediction, fast detection and rapid action are the key elements for controlling such phenomenon and saving lives. Through this work, 517 different entries were selected at different times for montesinho natural park (MNP) in Portugal to determine the best predictor that has the ability to detect forest fires. The principle component analysis (PCA) was applied to find the critical patterns and particle swarm optimization (PSO) technique was used to segment the fire regions (clusters). In the next stage, five soft computing (SC) Techniques based on neural network were used in parallel to identify the best technique that would potentially give more accurate and optimum results in predicting of forest fires, these techniques namely; cascade correlation network (CCN), multilayer perceptron neural network (MPNN), polynomial neural network (PNN), radial basis function (RBF) and support vector machine (SVM). In the final stage, the predictors and their performance were evaluated based on five quality measures including root mean squared error (RMSE), mean squared error (MSE), relative absolute error (RAE), mean absolute error (MAE) and information gain (IG). The results indicate that SVM technique was more effective and efficient than the RBF, MPNN, PNN and CCN predictors. The results also show that the SVM algorithm provides more precise predictions compared with other predictors with small estimation error. The obtained results confirm that the SVM improves the prediction accuracy and suitable for forest fires prediction compared to other methods.

© 2017 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Forest fires is one of the most important environmental risks that have several negative influences in different aspect of life such as economic, natural environment and health [38]. However, early detection, fast prediction and rapid actions can play an essential role in controlling and saving lives from forest fires danger and their negative consequences.

According to Zadeh [66], soft computing (SC) is a collection of advanced computing methodologies, which aim to accomplish

robustness, tractability, total low cost and better solution for a problem under investigation. This makes SC widely used techniques in solving complex problems in many different applications [14] and appropriate techniques selection is the first step to solve any problem in real scenarios [29]. These techniques can be typically divided into two groups: hard and soft computing. Fig. 1 shows the problem solving technologies and their classifications.

During the last years, several works have proved that SC techniques play a vital role in monitoring of forest fires [4,37,43]. In another works, SC was used to discover and predict valuable information about the relationships between variables for forest fires occurrence [1,17,44] and for estimating burned areas under future climate conditions [3,9,60]. Several recent papers have also focused on SC techniques applied to forest fires prediction [5,13,18,27,45,50,65].

Recently, many approaches have adopted artificial neural network (ANN) to improve forest fires detection in early stages. Bisquert et al. [10] investigated the potential of ANN and logistic regression to estimate forest fire danger. The results of this work indicated that good classification accuracy achieved by ANN

* Corresponding author.

E-mail addresses: samaher@uobabylon.edu.iq (S. Al_Janabi), alshourbaji@ibrahim@gmail.com (I. Al_Shourbaji), mahdi.salman@uobabylon.edu.iq (M.A. Salman).

Peer review under responsibility of King Saud University.



Production and hosting by Elsevier

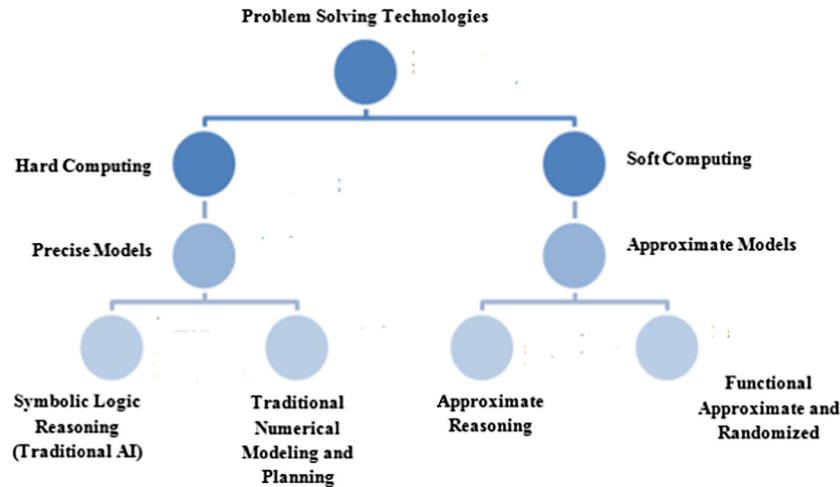


Fig. 1. Taxonomy of problem solving technologies.

technique. Similar results confirmed by the work of [24]. In another work, Maeda et al. [36] employed ANN to identify the areas of high risks of forest fires in Brazil. Their results showed that the ANN approach detects forest fires efficiently. A set of different learning parameters require to be identified in ANN including learning rate, hidden layers number and the nodes number in the hidden layer [48]. A huge number of training iterations may influence to over-train, and therefore, they could affect the accuracy of the prediction model [8].

A wide range of studies have reported for the adaptation of support vector machine (SVM) [34,49,61,67] and they found that the SVM achieved good results and it is able to effectively predict forest fire. This is because SVM does not require determining probabilities before analysis and therefore, makes the method more preferable.

This paper aims to investigate the performance of five SC techniques to determine the best and suitable predictor for forest fires. These methods include cascade correlation network (CCN), multi-layer perceptron neural network (MPNN), polynomial neural network (PNN), radial basis function (RBF) and support vector machine (SVM). To attain this goal, 517 different entries were selected at different times for montesinho natural park (MNP) dataset. The principle component analysis (PCA) was applied to find the critical patterns and particle swarm optimization (PSO) technique was used to segment the fire regions (clusters). The resulted data from PCA and PSO methods will be used as inputs in these predictors. The performance of these techniques will be evaluated based on five quality measures, which include root mean squared error (RMSE), mean squared error (MSE), relative absolute error (RAE), mean absolute error (MAE) and information gain (IG). The results from this work will assist and provide additional aids for systems operators to effectively monitor their system.

2. Material and methods

Forest fires detection require deployment a number of sensors scattered over a large area. The main function of these sensors is to collect various meteorological data such as moisture, pressure, temperature and humidity that can be used by the systems operators to take proper actions [26]. To infuse the gathered information from the different sensors and get meaningful information poses a significant challenge to the systems operators. These problems involve, finding the hidden patterns, reduce superfluous information and generate effective rules from the data [6,11]. This chal-

lenge would have been easier if the system has the ability to predict the state of the system based on the previously obtained values. For this, there is a prime importance to have an appropriate method, which can define the data easily according to the fire information. Fig. 2 illustrates the details of the study workflow.

From the above figure, the first phase is to preprocess the dataset. For this purpose, the principle component analysis (PCA) and particle swarm optimization (PSO) techniques were used to find the critical features and to segment the fire regions (clusters) respectively. Then, five predictors were applied to identify the best and suitable predictor that has the ability to predict forest fires. The final phase is to evaluate the predictors' performance based on five quality measures.

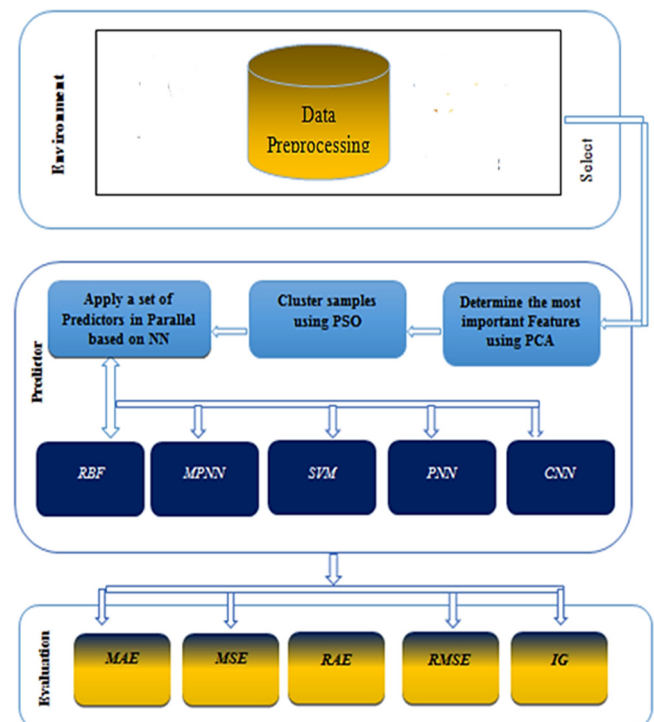


Fig. 2. Workflow process.

2.1. Case study and data collection

The montesinho natural park (MNP) was selected as a case study, which is located in the northeast of Portugal. Fig. 3 shows the geographical view of the park. The meteorological dataset was collected from the period of January 2000 to December which has an average temperature of 6–12 degree Celsius 2003. The data obtained from UCI machine learning repository, which have 517 different entries taken at different times. The input features consist of 12 attributes included under spatial and temporal components, fire weather index (FWI) component and weather conditions. The output feature represents the total burnt area. Data collection details can be founded in [16,54]. Several recent papers have used montesinho dataset in their works for forest fires prediction [15,23,25,30,46,57,62].

2.2. Principle component analysis (PCA)

PCA is one of the most widely used dimension reduction method, which is often applied to identify patterns in datasets of higher dimensions [28]. In PCA, a group of p -associated variables is transformed into a smaller set principal components (PCs) which can be used to find the dependences and associations that exist between variables in a compact manner. The PCA removes unimportant information from the variables and keeps the variation presents in the original dataset. PCA transforms the data variables for diagonalizing estimation of a covariance matrix $x_p, p = 1, 2, \dots, I, x_p \in \mathbb{R}^N$ which can be given by:

$$C = \frac{1}{I} \sum_{i=1}^I x_i x_i^T \quad (1)$$

2.3. Particle swarm optimization (PSO)

PSO [33] is a widely used optimization technique to find optimal solutions in various applications [7,31,51,64]. Each candidate solution termed as particle that holds the previous path coordinate when it has the best results. Every particle has its own position, velocity and best solution. The PSO method can be presented as follows:

The swarm position in a D space of the i th particle can be presented by:

$$X_i = \{X_{i1}, X_{i2}, \dots, X_{id}\} \quad (2)$$

Each particle maintains a memory of its previous best position. The best swarm position is given by:

$$P_{gbest} = \{P_{g1}, P_{g2}, \dots, P_{gd}\} \quad (3)$$

As well as velocity of each particle can be represented

$$V_i = \{V_{i1}, V_{i2}, \dots, V_{id}\} \quad (4)$$

A particle velocity should be updated and the following equation can be used

$$V_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (5)$$

where c_1, c_2 parameters are constants and r_1, r_2 are random numbers within 0–1. p_{id} is the best local solution of the i th particle for the iteration number up to the i th iteration. The best global solution of all particles is p_{gd} . The “inertia weight” w controls the effect of the previous velocity of the particle on its current one. If the value of w is greater than 1, then the particle favored searching over exploitation, else w is less than 1, the particle gave more importance to the current best positions.

2.4. Cascade correlation network (CCN)

CCN [21] is a supervised learning architecture that constructs a near multi-layer network topology with a high level of accuracy. CCN network structure holds no hidden nodes and each input connected to each output with an adjustable weight corresponding to each connection. When the network is incapable to build a correlation between the input and output, a hidden unit is added to the network from a pool of candidate units. The process continues until the error in the output units reaches to an acceptable level [52]. The output units are trained to reduce the familiar sum-squared error measure

$$E = \frac{1}{2} \sum_{o,p} (y_{op} - t_{op})^2 \quad (6)$$

where y_{op} is the observed value of output o for the training pattern p . The target output is t_{op} . E is minimized by gradient descent using



Fig. 3. Map of Portugal illustrating the location of montesinho natural park.

$$\frac{\partial E}{\partial \mathbf{w}_{oi}} = \sum_p (y_{op} - t_{op}) \mathbf{f}'_p \mathbf{I}_{ip} \quad (7)$$

where \mathbf{f}'_p represents derivative of the activation function of the output unit for pattern p . \mathbf{I}_{ip} is the value of hidden unit i . \mathbf{w}_{oi} presents the weight connecting input i to output unit o [47].

2.5. Multilayer perceptron neural network (MPNN)

MPNN is a widely used feed forward networks due to its clear architecture, fast operation, easy to implement and its capability to solve complex classification problems [32,55]. The MPNN network includes three main layers: input, hidden and output. These layers are used for data input, data transmitting and data output respectively. The hidden layer function is to pass the results to the output layer [22,40]. The output of each neuron can be written as

$$\mathbf{y}_i = f\left(\sum \mathbf{w}_{ij} \mathbf{x}_i\right) \quad (8)$$

where \mathbf{y}_i represents the input that a single node j receives. The function f can be a simple threshold, sigmoid or hyperbolic tangent function. The weights between the nodes i and j is denoted by \mathbf{w}_{ij} . \mathbf{x}_i represents the output from node i .

2.6. Polynomial neural networks (PNNs)

PNN is based on group method of data handling (GMDH) that uses a class of polynomials named partial descriptions (PDs). By picking the polynomial order between these numerous types of available forms and the most essential input variables, the best PDs can be obtained based on choosing the nodes of each layer and producing additional layers until the best performance is attained [41,68]. The association between input and output can be described as

$$\mathbf{y} = f(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \quad (9)$$

The estimated output $\hat{\mathbf{y}}$ can be read as

$$\begin{aligned} \hat{\mathbf{y}} &= \hat{f}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \\ &= c_0 + \sum_{k1} c_{k1} \mathbf{x}_{k1} + \sum_{k1k2} c_{k1k2} \mathbf{x}_{k1k2} + \sum_{k1k2k3} c_{k1k2k3} \mathbf{x}_{k1k2k3} + \dots, \end{aligned} \quad (10)$$

where c_k 's refer to the coefficients of the optimized model [42].

2.7. Radial basis function (RBF)

RBF neural network has a feed forward structure that contains an input layer, a single hidden and an output layer [12]. Whereas the hidden layers collect the data from the input layers and pass them to the Gaussian transfer function to transform and regulate the data nonlinearly. The function responses are then linearly merged to create the data of the output layer. RBF is widely applied in several applications such as system control, time series prediction, dynamic system problems and data classification because of its ability to forecast the behavior directly from input and output data [19,35,63]. RBF network tries to minimize the following error function (training error)

$$E = \frac{1}{2} \sum_{i=1}^k \sum_{j=1}^p e_j^2(t) \quad (11)$$

where $e_{j(t)}$ is the error of each output unit [39].

2.8. Support vector machine (SVM)

SVM [58] algorithm is commonly employed to solve classification and regression problems. It performs the classification by separating the data into two categories to determine the output and construct N -dimensional hyperplane. The main reason behind using SVM lies in dealing and addressing various nonlinear issues, flexible use of kernel functions to increase the ability to convert data from some high dimensional space. The results of the higher dimensional space are similar to the results in dimensional input space. The SVM algorithm is considered as an alternative technique for polynomial, radial basis function and multi-layer perceptron classifiers, the weights of the network are found and used to solve a quadratic programming problem with linear constraints. SVM aims to find the hyperplanes that have high generalization ability by separating the training data without errors and developing a robust and effective solution. The SVM classifier tries to minimize the following function

$$L_p = \frac{1}{2} \|\tilde{\mathbf{w}}\|^2 - \sum_{i=1}^n \alpha_i y_i (\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_i + \mathbf{b}) + \sum_{i=1}^n \alpha_i \quad (12)$$

where the number of training data is n and α_i , $i = 1 \dots n$ are the non-negative numbers such that the derivatives of L_p to α_i are zero, α_i are called the Lagrange multipliers, while the hyperplane is defined by Lagrangian function L_p , the vectors $\tilde{\mathbf{w}}$ and constant \mathbf{b} .

2.9. Quality measures

The following quality measures were selected in the performance evaluation of CCN, MPNN, PNNs, RBF and SVM models.

- Root mean squared error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\mathbf{y}_i - \mathbf{y}'_i)^2}{n}} \quad (13)$$

- Mean squared error (MSE):

$$MSE = \frac{\sum_{i=1}^n (\mathbf{y}_i - \mathbf{y}'_i)^2}{n} \quad (14)$$

- Relative absolute error (RAE):

$$RAE = \frac{\sum_{i=1}^n |\mathbf{y}_i - \mathbf{y}'_i|}{\sum_{i=1}^n |\mathbf{y}_i - \mathbf{y}''_i|} \quad (15)$$

- Mean absolute error (MAE):

$$MAE = \frac{\sum_{i=1}^n |\mathbf{y}_i - \mathbf{y}'_i|}{n} \quad (16)$$

where n is the number of data rows, \mathbf{y}_i presents the actual target value of record i and \mathbf{y}'_i is the predicted target value of record i .

- Information gain (IG)

$$\begin{aligned} Entropy(p_1, p_2, \dots, p_n) &= - \sum_{i=1}^n p_i \log_2 p_i \\ Information(Attribute) &= \sum_{i=1}^n \frac{FreSubAttribute}{Total\ Number\ of\ Records} * Entropy \\ InformationGain(IG) &= IBS - IAS \end{aligned} \quad (17)$$

where IBS refers to the information before splitting and IAS indicates to the information after splitting [2].

3. Results and discussion

To apply the SC techniques, a program was developed using python programming language. Python is an environment for

machine learning, text mining and business analysis. It is also widely used for research and education.

3.1. Used data

The collected meteorological dataset have 517 different entries were taken at different points of time and 13 input features are preprocessed thoroughly. The preprocessing phase was executed according to the following steps:

3.1.1. Preprocess of the dataset

In this stage, description features are converted into numerical using Min-Max normalization technique. This was done by extracting the input features from the dataset and preprocessing them. The normalized value, e_i for variable E in the i th row can be computed as:

$$\text{Normalized}(e_i) = \frac{e_i - E_{\min}}{E_{\max} - E_{\min}} \quad (18)$$

where E_{\min} presents the minimum value for variable E while E_{\max} represents the maximum value for variable E [53]. The procedure of normalization and coding are as follows:

3.1.1.1. Normalization. The data value can be scaled by using Min-Max method and the Linear transformation (L') of the original input ranged to a newly specified data range.

$$L' = [(L_{\min}) / (\max - \min)] * (\max' - \min') + \min' \quad (19)$$

where \min is the old minimum value, \min' is new minimum, \max is old maximum, \max' is new maximum.

For example, consider an old data that ranged from [0–100], we obtain an equation to migrate the data to range from [5–10].

$$\begin{aligned} L' &= [(L - 0) / (100 - 0)] * (10 - 5) + 5 \\ L' &= [L / 100] * 5 + 5 \\ L' &= (L / 20) + 5 \\ \text{Let } L &= 0 \text{ Then } L' = 5 \\ \text{If } L &= 10 \text{ Then } L' = (1 / 2) + 5 = (1 + 10) / 2 = 5.5 \end{aligned}$$

3.1.1.2. Coding (Convert linguistic terms to numeric form). To encode the attributes of linguistic variable (e.g. day and month), the following procedure was used

- **Create:** The repetition table by determining the repetition times for each linguistic term.
- **Rearrange:** The table by making the large value repeated in the middle and the lower one in the right and left. This process was repeated until the minimum repetition becomes at most left and most right.
- **Assign:** Code for each linguistic term depending on its new order in the repetition table.

3.1.2. Find the correlation measures

The main goal from this step is to find the correlation between the features which represent the most important variables in the dataset. Table 1 provides the correlation among these features using the following:

$$r = \frac{N \sum xy - \sum x \sum y}{\sqrt{N(\sum x^2) - (\sum x)^2} \sqrt{N(\sum y^2) - (\sum y)^2}} \quad (20)$$

where N present the vector size and $-1 \leq r \leq 1$, the positive sign (+) denotes to positive correlation and the negative sign (–) refers to negative correlation.

The value of the correlation that is greater than 0.8 is considered as a strong value while the ones less than 0.5 is contemplated as a weak value. The values depend on the data from one dataset to another and all other features have a strong correlation to each other except in RH column as shown in Table 1.

In this work, the correlation was only used as an initial process to find associations between the features. It was found that RH has a weak correlation with other variables. The PSO method was used to cluster the data without RH column and the resulted data were employed as inputs to the predictors. However, both the computational cost and the generated error are increased in the predictors remarkably. For this reason, we decide to apply PCA technique to optimize the selection of the most correlated variables in the dataset.

3.1.3. Find the eigenvalues and eigenvectors among features

Based on the eigenvalue criterion, each component should explain at least one variable's worth of the variability, and therefore only components with eigenvalues >1 should be selected. Every eigenvector has a corresponding eigenvalue and for the proportion of variance criterion, the analysis simply selects the components one by one until the desired proportion of variability is achieved. The minimum communality criterion states that enough components should be extracted, so that the communalities for each of these variables exceed a certain threshold, for example 50%. The maximum number of components that should be extracted is just prior to where the scree plot begins to straighten out into a horizontal line. The following illustrates the PCA pseudocode that was used to find eigenvalues and eigenvectors.

PCA procedure

Input: Process dataset

Output: Principal Components (PC)-database

Step1: Compute standardized data matrix $Z = [Z_1, Z_2, \dots, Z_m]$ based on $Z_i = (X_i - \mu_i) / \sigma_{ii}$ from the original dataset **Step 2:** Compute eigenvalues

let B be an $m \times m$ matrix and I be $m \times m$ identity matrix (diagonal matrix with 1's on the diagonal) **then** the scalars (numbers of dimension 1×1) $\lambda_1, \lambda_2, \dots, \lambda_m$ are said to be the eigenvalues of B if they satisfy $|B - \lambda I| = 0$

Step 3: Compute eigenvectors

let B be an $m \times m$ matrix and λ be an eigenvalues of B **then** nonzero $m \times 1$ vector e is said to be an eigenvector of B if $Be = \lambda e$.

Step 4: Compute i th PC, the i th PC of the standardized data matrix $Z = [Z_1, Z_2, \dots, Z_m]$ is given by $Y_i = e_i^T Z$

End procedure

The obtained eigenvalues, eigenvectors and cumulative variability of the dataset are presented. Table 2 provides the eigenvalues, while the eigenvectors among the features are summarized in Table 3. The variance of components is shown in Fig. 4.

Based on the results from the PCA method and for the purpose of improving the accuracy and speed up the models generation process, all the variables will be used in the next phase since there is a strong correlation between all variables in the MNP collected dataset.

3.1.4. Apply PSO technique

The PSO method is used to divide the correlated data after using the PCA method into clusters, so that items in the same group are as similar as possible and items in different groups are as dissimilar

Table 1
Correlation among the features.

Variables	X	Y	MONTH	DAY	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
X	1	0.540	−0.065	0.009	−0.021	−0.048	−0.086	0.006	−0.051	0.085	0.019	0.065	0.063
Y	0.540	1	−0.066	0.046	−0.046	0.008	−0.101	−0.024	−0.024	0.062	−0.020	0.033	0.045
MONTH	−0.065	−0.066	1	0.073	0.291	0.467	0.869	0.187	0.369	−0.095	−0.086	0.013	0.056
DAY	0.009	0.046	0.073	1	0.081	0.052	0.054	0.028	0.085	−0.145	−0.077	0.022	0.007
FFMC	−0.021	−0.046	0.291	0.081	1	0.383	0.331	0.532	0.432	−0.301	−0.028	0.057	0.040
DMC	−0.048	0.008	0.467	0.052	0.383	1	0.682	0.305	0.470	0.074	−0.105	0.075	0.073
DC	−0.086	−0.101	0.869	0.054	0.331	0.682	1	0.229	0.496	−0.039	−0.203	0.036	0.049
ISI	0.006	−0.024	0.187	0.028	0.532	0.305	0.229	1	0.394	−0.133	0.107	0.068	0.008
Temp	−0.051	−0.024	0.369	0.085	0.432	0.470	0.496	0.394	1	−0.527	−0.227	0.069	0.098
RH	0.085	0.062	−0.095	−0.145	−0.301	0.074	−0.039	−0.133	−0.527	1	0.069	0.100	−0.076
Wind	0.019	−0.020	−0.086	−0.077	−0.028	−0.105	−0.203	0.107	−0.227	0.069	1	0.061	0.012
Rain	0.065	0.033	0.013	0.022	0.057	0.075	0.036	0.068	0.069	0.100	0.061	1	−0.007
Area	0.063	0.045	0.056	0.007	0.040	0.073	0.049	0.008	0.098	−0.076	0.012	−0.007	1

Table 2
Eigenvalues among the different features.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
Eigenvalue	3.33	1.57	1.44	1.25	1.01	0.96	0.91	0.77	0.54	0.46	0.43	0.212	0.07
Variability	25.61	12.08	11.14	9.62	7.82	7.42	6.99	5.94	4.19	3.56	3.34	1.634	0.58
C. V.	25.61	37.70	48.84	58.46	66.29	73.72	80.72	86.67	90.87	94.44	97.7	99.42	100.0

C. V. = Cumulative Variability.

Table 3
Eigenvectors among the features.

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
X	−0.067	0.679	−0.015	−0.055	0.032	−0.126	0.017	0.116	−0.313	−0.214	−0.595	−0.047	−0.017
Y	−0.061	0.673	−0.017	−0.140	−0.011	−0.145	0.040	0.076	0.289	0.231	0.595	0.024	0.058
MONTH	0.408	−0.014	0.331	−0.096	0.069	0.024	0.193	0.378	−0.368	−0.058	0.247	0.006	−0.574
DAY	0.078	0.066	−0.180	−0.280	−0.497	0.463	0.611	−0.185	0.027	−0.067	−0.042	0.057	0.013
FFMC	0.359	0.060	−0.296	0.255	−0.035	−0.121	0.033	−0.237	−0.367	0.675	−0.077	0.214	0.023
DMC	0.404	0.095	0.283	0.085	0.010	−0.007	0.023	−0.235	0.556	0.194	−0.317	−0.430	−0.229
DC	0.462	−0.015	0.371	−0.101	0.030	−0.006	0.068	0.168	−0.095	−0.061	0.018	−0.005	0.770
ISI	0.292	0.098	−0.272	0.463	−0.021	−0.174	0.081	−0.334	−0.106	−0.567	0.293	−0.221	0.017
Temp	0.421	0.028	−0.275	−0.134	0.002	−0.016	−0.258	0.126	0.377	−0.252	−0.131	0.641	−0.108
RH	−0.183	0.107	0.620	0.259	−0.057	−0.007	0.080	−0.452	−0.018	−0.079	0.012	0.528	−0.065
Wind	−0.112	0.017	−0.069	0.611	0.218	0.142	0.462	0.477	0.253	0.061	−0.101	0.132	0.069
Rain	0.039	0.170	0.083	0.348	−0.416	0.572	−0.536	0.184	−0.078	0.033	0.069	−0.088	0.004
Area	0.060	0.133	−0.077	−0.118	0.721	0.596	−0.039	−0.268	−0.077	−0.016	0.063	0.001	0.014

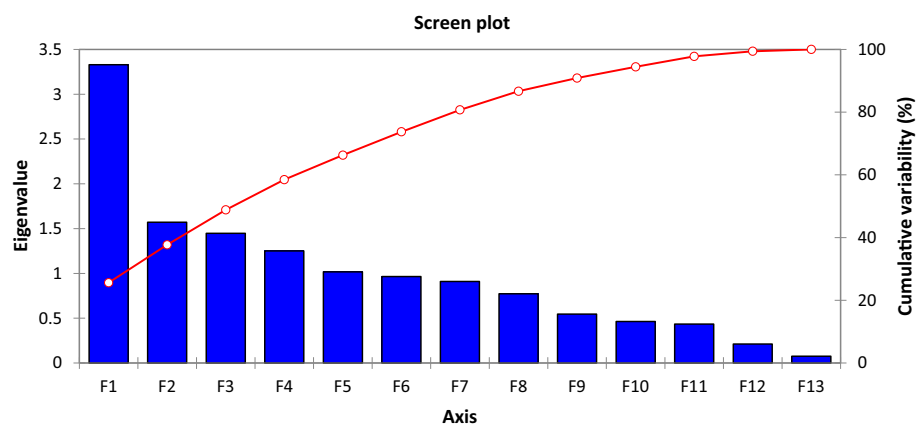


Fig. 4. The variance of components.

as possible. The PSO selects the best seeds for each cluster where each of them has 13 values and each value represents a feature.

In order to get the shortest and largest distances between clusters, weights were given to each cluster and the traditional

Euclidean distance between the seeds and clusters was computed. Tables 4 and 5 provide the initial and final values of centroid classes. Five optimal groups were generated by the PSO method and the final centroid classes are provided in Table 6.

Table 4

Initial centroid classes.

Class	X	Y	MONTH	DAY	FFMC	DMC	DC	ISI	Temp	RH	Wind	Rain	Area
1	4.491	4.296	7.389	3.324	90.402	106.962	537.491	9.244	18.483	45.981	4.033	0.011	7.504
2	4.929	4.347	7.571	3.837	90.224	118.626	553.101	9.241	18.905	44.643	3.946	0.002	14.139
3	4.179	4.221	7.632	3.979	91.023	117.019	569.926	9.005	19.122	43.284	4.053	0.008	20.472
4	5.055	4.418	7.418	3.509	90.597	109.219	548.775	8.348	18.740	45.582	3.741	0.071	15.794
5	4.651	4.208	7.396	3.830	90.991	103.895	533.243	9.306	19.234	41.792	4.324	0.011	7.206

Table 5

Final centroid classes.

Class	X	Y	MONTH	DAY	FFMC	DMC	DC	ISI	Temp	RH	Wind	Rain	Area	Sum of weights	Within class variance
1	4.92	4.42	3.06	3.32	86.72	26.69	68.01	6.47	12.28	46.20	4.58	0.00	5.38	89.00	2154.20
2	4.43	4.17	8.15	3.99	91.73	127.75	626.58	10.80	21.11	43.53	4.08	0.05	10.68	149.00	4438.87
3	4.48	4.20	8.79	3.60	91.73	140.85	734.60	8.97	20.18	44.21	3.61	0.01	8.93	217.00	6809.94
4	5.46	4.73	7.53	3.70	89.71	83.36	384.76	8.47	18.24	44.18	4.48	0.00	13.22	60.00	9643.06
5	7.00	5.50	8.50	4.50	93.65	171.75	686.50	11.25	26.30	27.00	4.45	0.00	918.56	2.00	64819.93

Table 6

Distance between the centroid classes.

	1	2	3	4	5
1	000.000	567.812	676.403	321.983	1112.710
2	567.812	000.000	108.840	245.932	911.079
3	676.403	108.840	000.000	354.582	911.610
4	321.983	245.932	354.582	000.000	958.583
5	1112.710	911.079	911.610	958.583	000.000

3.2. Apply the predictors

The parameters for each predictor used in this work were selected using trial and error method which can be a fundamental technique in problem solving and obtaining knowledge. To evaluate the predictors and their performance based on the 5 quality measures were used in this paper, the dataset was split into two parts: 50% of the data were used for training phase while the remaining data were utilized for the testing phase.

3.2.1. RBF network

The main parameters of this predictor include: the number of neurons = 100, the minimum radius = 0.01, the maximum radius = 519.669, the minimum Lambda = 0.01328, the maximum Lambda = 9.95337 and Regularization Lambda for final weights = 6.3458e–005 after 14 iterations.

3.2.2. Multilayer perceptron NN

The main parameters of this predictor include: the number of inputs = 12, the number of layers = 3 (1 hidden), hidden layer 1 neurons: search from 2 to 20, the hidden layer activation function is logistic, the output layer activation function is linear, the min. weight of hidden layer = –3.636e–001, the max. Weight of hidden layer = 3.409e–001; the min. weight of output layer = 9.638e–005, and the max. Weight of output layer = 1.358e–001.

3.2.3. Support vector machine (SVM)

The main parameters of this predictor include: the kernel function is radial basis function (RBF), the Epsilon = 0.001, C = 84.1830278, Gamma = 3800.28665 and =6.4728203.

3.2.4. Polynomial neural network

The main parameters of this predictor include: activation kernel function = Gaussian (one variable) $y = p1 + p2 * \exp(-(x1 - p3)^2/p4)$, $p1 = 31.30268$, $p2 = -27.3003$, $p3 = 1.137834$, $p4 = 6.307772$ and output area = $31.30268 - 27.3003 * \exp(-(temp + 1.137834)^2/6.307772)$

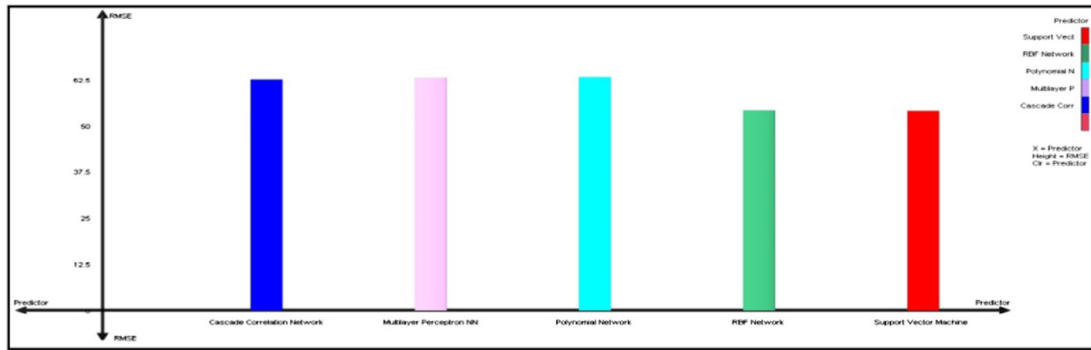
3.2.5. Cascade correlation network

The main parameters of this predictor include: the minimum neurons in hidden layer is 0, the maximum neurons in hidden layer is 50, the hidden neuron kernel are sigmoid and Gaussian functions, the output neuron kernel function: Sigmoid, the number of neurons in input layer = 12, the number of neurons in hidden layer = 2 (Sigmoid neurons = 1 and Gaussian neurons = 1), the minimum weight in hidden layer = –25.040615, the maximum weight in hidden layer = 13.892605, the number of neurons in output layer = 1, the minimum weight in output layer = –0.217332 and the maximum weight in output layer = 0.150124.

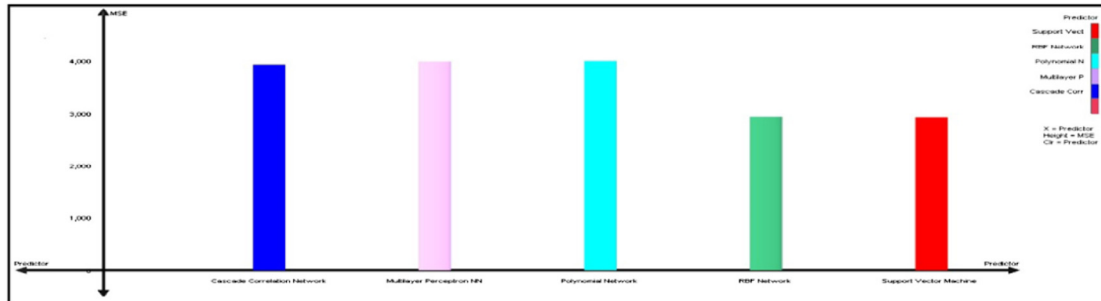
Table 7

Comparison among different predictors based on the quality measures.

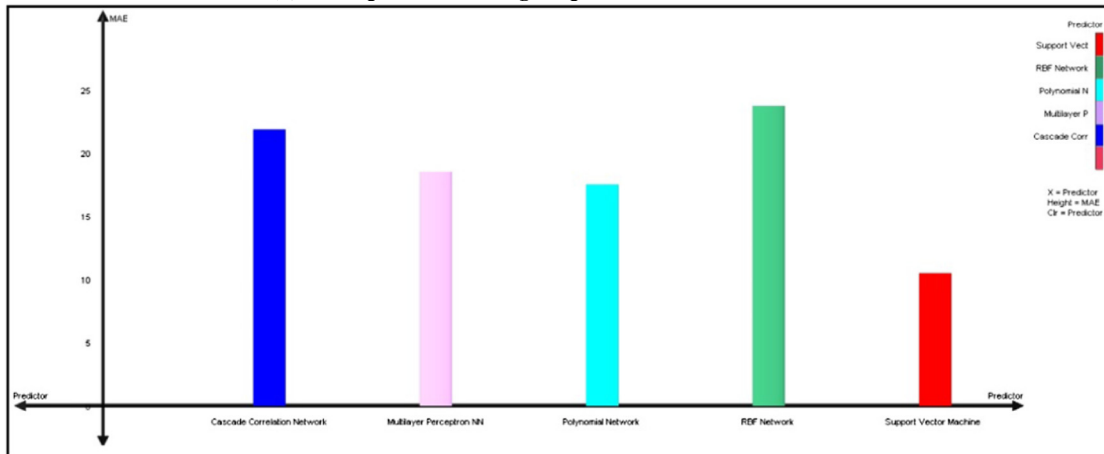
Predictor	Testing phase					Training phase				
	RMSE	MSE	RAE	MAE	IG	RMSE	MSE	RAE	MAE	IG
CNN	62.6	3926.7	21.9	551.4	1.803	68.3	4665.5	25.2	673.9	1.174
MPNN	63.1	3993.2	18.5	617.4	1.419	63.8	4076.4	19.6	652.2	1.146
PNN	63.2	3996.5	17.5	516.5	1.508	63.7	4069.3	17.2	466.0	0.982
RBF	54.2	2939.9	23.7	911.2	2.160	68.1	4638.2	26.2	887.9	1.077
SVM	54.0	2926.4	10.5	282.4	2.656	63.5	4042.0	18.3	502.0	0.817



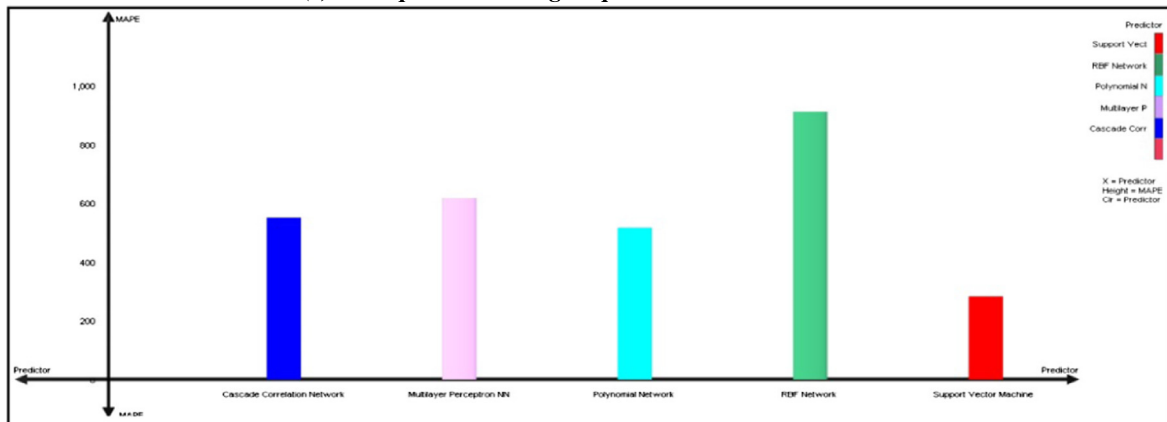
(a) Comparison among all predictors based on RMSE



(b) Comparison among all predictors based on MSE

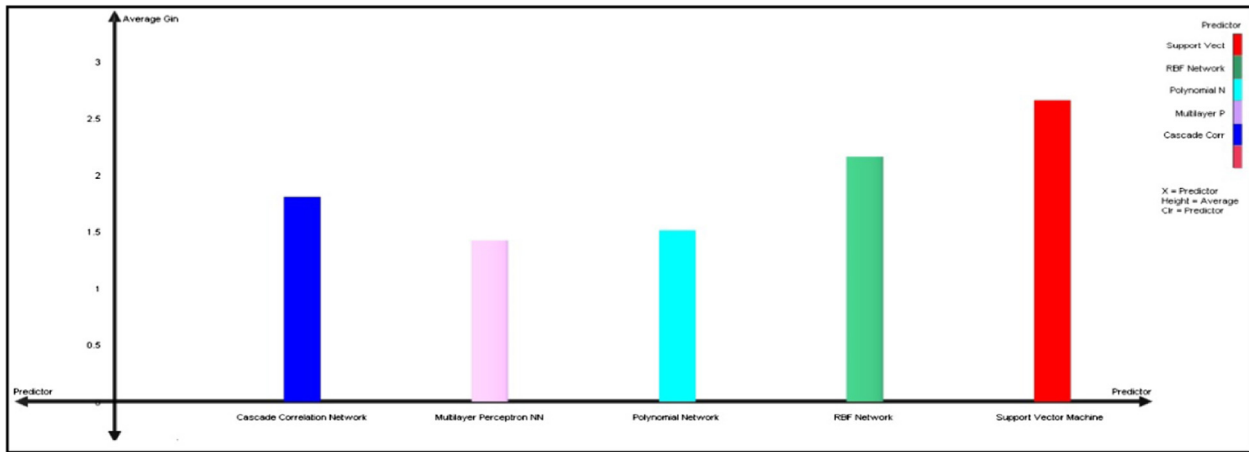


(c) Comparison among all predictors based on MAE



(d) Comparison among all predictors based on MAPE

Fig. 5. Evaluation the predictors based on the quality measures, (a)–(d) and (e). Comparison between predictors and the quality measures.



(e) Comparison among all predictors based on IG

Fig. 5 (continued)

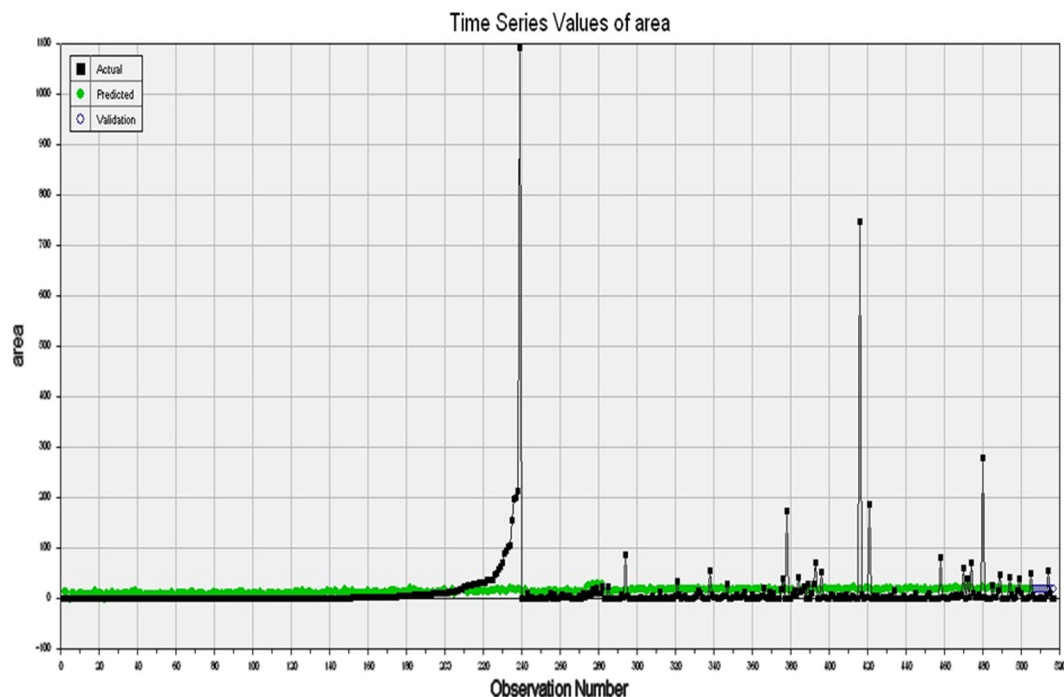


Fig. 6. Predicted values of burnt area based on the best predictor (SVM).

3.3. Performance analysis

The results in Table 7 show that SVM predicts fire probability well. A comparison of SVM results with other predictors reveals that the SVM outperforms the RBF, MPNN, PNN and CCN predictors. SVM has the smallest RMSE of 54.0, MSE of 2926.4, RAE of 10.5, and MAE of 2.656 and the highest IG of 2.656 in the testing stage. In general, the results indicate that SVM has the best prediction ability for forest fire compared to other selected SC methods.

Results of the five SC for forest fires prediction based on each statistical measure are shown in Fig. 5. According to the bar charts, the SVM has achieved the highest gain in (e) with minimal error in (a, b, c and d). Therefore, the SVM algorithm is the best predictor and it has the capability to effectively predict forest fires compared to other predictors. Similar results were obtained by the work of [16,61]. The difference between the expected and actual values

for the MNP dataset that generated by the best predictor SVM techniques provided in Fig. 6. Green color represents the predicted values while the black color presents the actual values.

4. Conclusion

In this article, several SC techniques were applied on MNP dataset to determine the effective predictor that would potentially give more accurate results for forest fires. The PCA was used to capture the correlation between features and PSO technique was applied to find the number of clusters existing in the resulted data after using the PCA technique. In the next stage, the obtained clustered data was used as inputs in the SC techniques. The quality measures used in this work make it obvious that SVM attained best results with a high level of accuracy compared to other methods. The findings in

this study would assist and provide additional aids for systems operators' to monitor their system effectively.

References

- [1] W. Aertsen, V. Kint, J. Van Orshoven, K. Ozkan, B. Muys, Performance of modelling techniques for the prediction of forest site index: a case study for pine and cedar in the Taurus mountains. Turkey XIII World Forestry Congress, 2009, pp. 18–23.
- [2] A.M. Al-Bakary, H.A. Samaher, Data construction using genetic programming method to handle data scarcity problem, *Int. J. Adv. Comput. Technol.* (2010).
- [3] G. Amatulli, A. Camia, J. San-Miguel-Ayanz, Estimating future burned areas under changing climate in the EU-Mediterranean countries, *Sci. Total Environ.* 450 (2013) 209–222.
- [4] V.D. Anezakis, K. Demertzis, L. Iliadis, S. Spartalis, A hybrid soft computing approach producing robust forest fire risk indices, in: IFIP International Conference on Artificial Intelligence Applications and Innovations, Springer International Publishing, 2016, pp. 191–203.
- [5] T. Artés, A. Cencerrado, A. Cortés, T. Margalef, Time aware genetic algorithm for forest fire propagation prediction: exploiting multi-core platforms, *Concurrency and Computation: Practice and Experience*, 2016.
- [6] Y.E. Aslan, I. Korpeoglu, Ö. Ulusoy, A framework for use of wireless sensor networks in forest fire detection and monitoring, *Comput. Environ. Urban Syst.* 36 (6) (2012) 614–625.
- [7] E. Assareh, M.A. Behrang, M.R. Assari, A. Ghanbarzadeh, Application of PSO (particle swarm optimization) and GA (genetic algorithm) techniques on demand estimation of oil in Iran, *Energy* 35 (12) (2010) 5223–5229.
- [8] I. Bashier, A. Hajmeer, M. Artificial neural networks: fundamentals computing design and application, *J. Microbiol. Methods* 43 (1) (2000) 3–31.
- [9] J. Bedia, S. Herrera, A. Camia, J.M. Moreno, J.M. Gutiérrez, Forest fire danger projections in the Mediterranean using ENSEMBLES regional climate change scenarios, *Climatic Change* 122 (1–2) (2014) 185–199.
- [10] M. Bisquert, E. Caselles, J.M. Sánchez, V. Caselles, Application of artificial neural networks and logistic regression to the prediction of forest fire danger in Galicia using MODIS data, *Int. J. Wildland Fire* 21 (8) (2012) 1025–1029.
- [11] K. Bouabdellah, H. Noureddine, S. Larbi, Using wireless sensor networks for reliable forest fires detection, *Proc. Comput. Sci.* 19 (2013) 794–801.
- [12] D.S. Broomhead, D. Lowe, Radial basis functions, multi-variable functional interpolation and adaptive networks (No. RSRE-MEMO-4148). Royal signals and radar establishment Malvern (United Kingdom), 1988.
- [13] D.T. Bui, Q.T. Bui, P.P. Nguyen, B. Pradhan, H. Nampak, P.T. Trinh, A hybrid artificial intelligence approach using GIS-based neural-fuzzy inference system and particle swarm optimization for forest fire susceptibility modeling at a tropical area, *Agric. For. Meteorol.* 233 (2017) 32–44.
- [14] D.K. Chaturvedi, *Soft Computing: Techniques and Its Applications in Electrical Engineering*, Springer, 2008.
- [15] T. Cheng, J. Wang, Integrated spatio-temporal data mining for forest fire prediction, *Trans. GIS* 12 (5) (2008) 591–611.
- [16] P. Cortez, A.D.J.R. Morais, A data mining approach to predict forest fires using meteorological data, in: Proceedings of the 13th Portuguese Conference on Artificial Intelligence (EPIA) Guimarães Portugal, 2007, pp. 512–523.
- [17] A. De Angelis, C. Ricotta, M. Conedera, G.B. Pezzatti, Modelling the meteorological forest fire niche in heterogeneous pyrologic conditions, *PLoS one* 10 (2) (2015) e0116875.
- [18] M. Denham, A. Cortés, T. Margalef, E. Luque, Applying a dynamic data driven genetic algorithm to improve forest fire spread prediction, in: International Conference on Computational Science, Springer Berlin Heidelberg, 2008, pp. 36–45.
- [19] H. Du, N. Zhang, Time series prediction using evolving radial basis function networks with new encoding scheme, *Neurocomputing* 71 (7) (2008) 1388–1400.
- [21] S.E. Fahlman, C. Lebiere, The cascade-correlation learning architecture, Gardner (1990).
- [22] M.W. Gardner, S.R. Dorling, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences, *Atmos. Environ.* 32 (14) (1998) 2627–2636.
- [23] A. Felber, P. Bartelt, The use of the Nearest Neighbour Method to predict forest fires, Proceedings of 4th International Workshop on Remote Sensing and GIS Applications to Forest Fire Management: Innovative Concepts and Methods in Fire Danger Estimation, Ghent, Belgium, 2003.
- [24] Y.J. Golderag, A. Mohammadzadeh, A.S. Ardakani, Fire risk assessment using neural network and logistic regression, *J. Indian Soc. Rem. Sens.* (2016) 1–10.
- [25] J. Han, K. Ryu, K. Chi, Y. Yeon, Statistics based predictive geo-spatial data mining: forest fire hazardous area mapping application, *Web Technol. Appl.* (2003), 602–602.
- [26] M. Hefeeda, M. Bagheri, Forest fire modeling and early detection using wireless sensor networks, *Ad Hoc & Sensor Wireless Networks* 7 (3–4) (2009) 169–224.
- [27] H. Hong, S.A. Naghibi, M.M. Dashtpazgerdi, H.R. Pourghasemi, W. Chen, A comparative assessment between linear and quadratic discriminant analyses (LDA-QDA) with frequency ratio and weights-of-evidence models for forest fire susceptibility mapping in China, *Arab. J. Geosci.* 10 (7) (2017) 167.
- [28] H. Hotelling, Analysis of a complex of statistical variables into principal components, *J. Educ. Psychol.* 24 (6) (1933) 417–441.
- [29] S. Hussein Ali, Designing a Software for Classifying Objects for Air Photos & Satellite Images using Soft Computing M.Sc. Thesis, Babylon University, 2005.
- [30] S. Jain, M.P.S. Bhatia, Performance investigation of support vector regression using meteorological data, *Int. J. Database Theory Appl.* 6 (4) (2013) 109–118.
- [31] M. Jiang, Y.P. Luo, S.Y. Yang, Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm, *Inform. Process. Lett.* 102 (1) (2007) 8–16.
- [32] T. Kavzoglu, P.M. Mather, The use of backpropagating artificial neural networks in land cover classification, *Int. J. Rem. Sens.* 24 (23) (2003) 4907–4938.
- [33] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks (1995) 1942–1948.
- [34] B.C. Ko, K.H. Cheong, J.Y. Nam, Fire detection based on vision sensor and support vector machines, *Fire Saf. J.* 44 (3) (2009) 322–329.
- [35] J. Liu, Radial Basis Function (RBF) Neural Network Control for Mechanical Systems: Design, Analysis and Matlab Simulation, Springer Science & Business Media, 2013.
- [36] E.E. Maeda, A.R. Formaggio, Y.E. Shimabukuro, G.F.B. Arcoverde, M.C. Hansen, Predicting forest fire in the Brazilian Amazon using MODIS imagery and artificial neural networks, *Int. J. Appl. Earth Obs. Geoinf.* 11 (4) (2009) 265–272.
- [37] E. Mahdipour, C. Dadkhah, Automatic fire detection based on soft computing techniques: review from 2000 to 2010, *Artif. Intell. Rev.* 42 (4) (2014) 895–934.
- [38] A.G. Motazeh, E.F. Ashtiani, R. Baniasadi, F.M. Chooobar, Rating and mapping fire hazard in the hardwood Hyrcanian forests using GIS and expert choice software, Acknowledgement to Reviewers of the Manuscripts Submitted to Forestry Ideas in 2013, p. 141.
- [39] R. Neruda, P. Kudová, Learning methods for radial basis function networks, *Future Gener. Comput. Syst.* 21 (7) (2005) 1131–1142.
- [40] S.N. Oğulata, C. Şahin, R. Erol, Neural network-based computer-aided diagnosis in classification of primary generalized epilepsy by EEG signals, *J. Med. Syst.* 33 (2) (2009) 107–112.
- [41] S.K. Oh, W. Pedrycz, The design of self-organizing polynomial neural networks, *Inf. Sci.* 141 (3) (2002) 237–258.
- [42] S.K. Oh, W. Pedrycz, B.J. Park, Polynomial neural networks architecture: analysis and design, *Comput. Electr. Eng.* 29 (6) (2003) 703–725.
- [43] J.A. Olivas, Forest fire prediction and management using soft computing, in: Proceedings of the International Conference on Industrial Informatics (INDIN), 2003, pp. 338–344.
- [44] S. Oliveira, F. Oehler, J. San-Miguel-Ayanz, A. Camia, J.M. Pereira, Modeling spatial patterns of fire occurrence in Mediterranean Europe using Multiple Regression and Random Forest, *Ecol. Manage.* 275 (2012) 117–129.
- [45] A.M. Özbayoglu, R. Bozer, Estimation of the burned area in forest fires using computational intelligence techniques, *Proc. Comput. Sci.* 12 (2012) 282–287.
- [46] K.S.N. Prasad, S. Ramakrishna, An autonomous forest fire detection system based on spatial data mining and fuzzy logic, *Int. J. Comput. Sci. Network Secur.* 8 (12) (2008) 49–55.
- [47] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation (No. ICS-8506), California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [48] Y. Safi, A. Bouroumi, Prediction of forest fires using artificial neural networks, *Appl. Math. Sci.* 7 (6) (2013) 271–286.
- [49] G.E. Sakr, I.H. Elhaji, G. Mitri, Efficient forest fire occurrence prediction for developing countries using two weather parameters, *Eng. Appl. Artif. Intell.* 24 (5) (2011) 888–894.
- [50] O. Satir, S. Berberoglu, C. Donmez, in: Mapping Regional Forest Fire Probability Using Artificial Neural Network Model in a Mediterranean Forest Ecosystem, Geomatics, Natural Hazards and Risk, 2015, pp. 1–14.
- [51] M. Shourian, S.J. Mousavi, A. Tahershamsi, Basin-wide water resources planning by integrating PSO algorithm and MODSIM, *Water Resour. Manage.* 22 (10) (2008) 1347–1366.
- [52] J.K. Spoerre, Application of the cascade correlation algorithm (CCA) to bearing fault classification problems, *Comput. Ind.* 32 (3) (1997) 295–304.
- [53] J.J. Storer, Computational Intelligence and Data Mining Techniques Using the Fire Data Set (MS Thesis), Bowling Green State University.
- [54] J. Storer, R. Green, PSO trained Neural Networks for predicting forest fire size: a comparison of implementation and performance, in: Neural Networks (IJCNN), 2016 International Joint Conference on, 2016, pp. 676–683.
- [55] A. Subasi, EEG signal classification using wavelet feature extraction and a mixture of expert model, *Expert Syst. Appl.* 32 (4) (2007) 1084–1093.
- [57] C.E. Van Wagner, P. Forest, Development and Structure of the Canadian Forest FireWeather Index System, For. Serv., Forestry Tech. Rep. in Can, 1987.
- [58] V.N. Vapnik, The Nature of Statistical Learning, Theory Springer-Verlag, New York, USA, 1995.
- [60] A.M. West, S. Kumar, C.S. Jarnevich, Regional modeling of large wildfires under current and potential future climates in Colorado and Wyoming, USA, *Climatic Change* 134 (4) (2016) 565–577.
- [61] D.W. Xie, S.L. Shi, Prediction for burned area of forest fires based on SVM model, in: Applied Mechanics and Materials, vol. 513, Trans Tech Publications, 2014, pp. 4084–4089.
- [62] L. Yang, C.W. Dawson, M.R. Brown, M. Gell, Neural network and GA approaches for dwelling fire occurrence prediction, *Knowl. - Based Syst.* 19 (4) (2006) 213–219.

- [63] H. Yu, T. Xie, S. Paszczynski, B.M. Wilamowski, Advantages of radial basis function networks for dynamic system design, *IEEE Trans. Industr. Electron.* 58 (12) (2011) 5438–5450.
- [64] T. Yu, L. Wang, X. Han, Y. Liu, L. Zhang, Swarm Intelligence Optimization Algorithms and Their Application. WHICEB 2015 Proceedings, 2015, p. 3.
- [65] C. Yuan, Y. Zhang, Z. Liu, A survey on technologies for automatic forest fire monitoring detection and fighting using unmanned aerial vehicles and remote sensing techniques, *Can. J. Forest Res.* 45 (7) (2015) 783–792.
- [66] L.A. Zadeh, Soft computing and fuzzy logic, *IEEE Software* 11 (6) (1994) 48.
- [67] J. Zhao, Z. Zhang, S. Han, C. Qu, Z. Yuan, D. Zhang, SVM based forest fire detection using static and dynamic features, *Computer Sci. Inform. Syst.* 8 (3) (2011) 821–841.
- [68] L. Zjavka, Wind speed forecast correction models using polynomial neural networks, *Renew. Energy* 83 (2015) 998–1006.