Original article

Weather forecasting using soft computing models: A comparative study

Nazim Osman **Bushara**

Faculty of Computer Science and Information Technology, Shendi University, Shendi, Sudan.

nazim_ob@yahoo.com

Abstract

One of the main fields of weather forecasting is rainfall prediction, which is significant for water resource management, food production plan and different activity plans in nature. The appearance of stretched dry period or intensive rain at the critical stages of the crop growth and development may lead to serious reduce crop yield. Certainly, the accurate forecasting in rainfall could present useful information for water resource administration, flood control and disaster relief. This study proposed several soft computing models for long term rainfall prediction based on monthly meteorological dataset for 13 years, the models are IBK, K-Star, M5P, adaptive neuro-fuzzy inference system (ANFIS), Meta vote, bagging, staking and ensemble by using different machine learning schemes such as hybrid intelligent system, data mining, meta learning and ensemble algorithms.. The results show the accuracies of both ANFIS and the ensemble model are satisfied and ANFIS showed relatively more accurate results.

Keywords: Weather forecasting, soft computing, neuro- fuzzy inference system, data mining.

Introduction

Weather forecasting is a complex task [1] because in the domain of meteorology all judgments are to be taken with a degree of uncertainty, because the chaotic nature of the atmosphere restricts the authenticity of deterministic forecasts. There are several types of weather forecasts made in relation to time such as short range, medium range and Long-range forecasts [2]. Weather forecasts still have their restrictions in spite of the use of modern technology. For example, short weather forecasts (for today or tomorrow) are likely to be more dependable than long term predictions. Some sources state that weather forecast accuracy declines significantly beyond 10 days [3].

Weather forecasting is complicated and not always accurate, especially for days further in the future, because the weather can be messy and unpredictable. Long-range weather forecasts are widely used in the energy industry, despite their limited proficiency they can still be a valuable tool for managing weather risk. Long term prediction of rainfall has several benefits for efficient resource planning and management including agriculture, famine and disease control, rainwater catchment and ground water management.

Many researchers employed ANFIS approach in wheather forecasting. Some explored Adaptive Neuro-Fuzzy Inference Systems (ANFIS) to forecast the average hourly wind speed, the results drove to 4 inputs models produced by grid partitioning and the 6 inputs models generated by subtractive clustering provided the best forecasting accuracy [4]. Network, mamdani and sugeno adaptive neuro fuzzy models were employed to predict rainfall of Ethiopian with different time lag [5]. The result showed the soft computing models perform the prediction with relatively small error and had better skill than techniques used by Ethiopian National Meteorological Service Agency (ENMSA) and other previous studies which used statistical techniques. Artificial Neural Networks (ANNs) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS) have been investigated for monthly rainfall estimation in Iran to determine the rainfall amount [6]. The results showed the accuracies of both models are satisfied, and ANNs showed relatively more accurate results. The performance of several Soft Computing (SC) models was compared: Evolving Fuzzy Neural Network (EFuNN), Artificial Neural Network using Scaled Conjugate Gradient Algorithm (ANNSCGA), Adaptive Basis Function Neural, Network (ABFNN) and General Regression Neural Network (GRNN) with Multivariate Adaptive Regression Splines (MARS) [7]. Simulation results revealed that MARS was a good forecasting tool and performed better than the considered SC models. Monthly rainfall was forecasted using ANFIS [8]. Eight models were developed using various membership functions and climatic parameters as inputs. The study showed that hybrid Model with seven membership functions and using three inputs gives best result to forecast rainfall for study area. ANFIS was applied for rainfall events evaluation [9]. Four parameters were the input variables for ANFIS model, each has 121 membership functions. Results showed a high agreement with the actual data.

Also, there are several studies which applied data mining and machine learning approaches in the field of weather forecasting and rainfall prediction; multiple linear regression (MLR) technique for the early rainfall prediction [10], the results proved that there is a close agreement between the predicted and actual rainfall amount prediction of rainfall. The development of a statistical forecasting method was described for using the multiple linear regression and local polynomial-based nonparametric approaches [11]. The experiments indicated that the correlation between observed and forecast rainfall was 0.6. The MPR technique was presented, to describe complex nonlinear relationship for prediction of rainfall and then compared the MPR and MLR technique based on the accuracy [12]. CART and C4.5 were proposed to predict rainfall [13]. The results showed

CART and C4.5 both have high accuracy and are efficient algorithms.

On the other side, the ensemble idea in supervised learning has been investigated since the late seventies. Combining two linear regression models was suggested [14]. An ensemble of similarly configured neural networks has suggested improving the predictive performance of a single one [15]. At the same time, Schapire [16] laid the foundations for the award-winning AdaBoost Freund and Schapire [17] algorithm by showing that a strong classifier in the probably approximately correct (PAC) sense can be generated by combining "weak" classifiers (that is, simple classifiers whose classification performance is only slightly better than random classification). After that, researchers from various disciplines such as statistics and AI considered the use of ensemble methodology. The P-AdaBoost algorithm has developed, which is a distributed version of AdaBoost [18]. Instead of updating the "weights" associated with instance in a sequential manner, P-AdaBoost works in two phases. In the first phase, the AdaBoost algorithm runs in its sequential, standard fashion for a limited number of steps. In the second, phase the classifiers are trained in parallel using weights that are estimated from the first phase. P-AdaBoost yields approximations to the standard AdaBoost models that can be easily and efficiently distributed over a network of computing nodes. A new boosting-by-resampling version of Adaboost has been proposed [19]. In the local Boosting algorithm, a local error is calculated for each training instance, which is then used to update the probability that this instance is chosen for the training set of the next iteration. After each iteration, in AdaBoost, a global error measure is calculated that refers to all instances. A new method has proposed to improve the performance of the Random Forests by increasing the diversity of each tree in the forests and thereby improving the overall accuracy [20]. A method was presented for improved ensemble learning, by treating the optimization of an ensemble of classifiers as a compressed sensing problem [21]. Ensemble learning methods improve the performance of a learned predictor by integrating a weighted combination of multiple predictive models. The idea of the ensemble is adapted for feature selection, an ensemble of filters have proposed for classification, aimed at achieving a good classification performance together with a reduction in the input dimensionality [22]. Studies have provided theoretical and empirical evidence that diversity is a key factor for yielding satisfactory accuracy-generalization performance with classifier ensembles. Some have tried to empirically assess the impact of using, in a sequential manner, three complementary approaches for enhancing diversity in classifier ensembles [23]. For this purpose, simulations were conducted on 15 well-known classification problems with ensemble models composed of up to 10 different types of classifiers. Overall, the results evidence the usefulness of the proposed integrative strategy in incrementing the levels of diversity progressively. A novel ensemble learning algorithm named Double Rotation Margin Forest (DRMF) have proposed, aims to improve the margin distribution of the combined system over the training set [24]. The classifier ensemble problem has formulated with sparsity and diversity learning in a general mathematical framework, which proves beneficial for grouping classifiers [25]. Also, some proposed a novel weighted rough set as a Meta classifier framework for 14 classifiers to find the smallest and optimal ensemble, which maximizes the overall ensemble accuracy [26]. They proposed a new entropy-based method to compute the weight of each classifier.

Ensemble technology does not always improve the performance of the base classifiers which composite of them, hybrid and ensemble model of forecasting method has been proposed for daily rainfall prediction based on ARIMA (Autoregressive Integrated Moving Average) and ANFIS at six certain areas in Indonesia [27]. To find an ensemble forecast from ARIMA and ANFIS models, the averaging and stacking method was implemented. The results showed that an individual ARIMA method yields a more accurate forecast in five rainfall data, whereas ensemble averaging multi model yields better forecast in one rainfall data. In general, these results indicated that a more complicated model does not always yield abetter forecast than a simpler ones.

The main purpose of this study is to compare between results of proposed nine models based on various soft computing technologies, to determine the best model capable to capture the dynamic behavior of the rainfall.

Materials and methods

Dataset

The meteorological dataset used in this study has been imparted from the Central Bureau of Statistics, Sudan for the interval 2000-2012 for 24 stations over the country with 3732 total number of examples. The dataset had eight attributes (station name, date, maximum temperature, minimum temperature, relative humidity, wind direction, wind speed and rainfall) containing monthly averages.

Feature selection

It is often an essential data processing step prior to applying a learning algorithm. Reduction of the attribute dimensionality leads to a better understandable model and simplifies the usage of different visualization technique and is the process of identifying and removing as much irrelevant and redundant information as possible. Reduces the dimensionality of the data, may allow learning algorithms to operate faster and more effectively and, accuracy can be improved later on future classification. It finds minimum set of attributes such that resulting probability distribution of data classes is as close as possible of original distribution. Methods used for attribute selection can be classified into two types. The filter approach and Wrapper approach. The filter approach actually precedes the actual classification process. The filter approach is independent of the learning algorithm, computationally simple fast and scalable. Using filter method, attribute selection is done once and then can be provided as input to different algorithms [28]. Wrapper approach uses the method of classification itself to measure the importance of attribute set, hence the attribute selection depends on the algorithm model used. Wrapper methods are too expensive for large dimensional database in terms of computational complexity and time since each attribute set considered must be evaluated with the classifier algorithm used. Filter methods are much faster than wrapper methods and therefore are better suited to high dimensional data sets. Some of these filter methods do not perform attribute selection but only attribute ranking hence they are combined with search method when one needs to find out the appropriate number of attributes. Such filters are often used with forward, backward elimination, bi-directional search, best-first search, and other methods [28,29]. Various AS techniques have been proposed in the literature such as:

a- Correlation-based Feature Selection (CFS)

CFS is a filter algorithm that ranks feature subsets according to a correlation based heuristic evaluation function. CFS assumes that useful feature subsets contain features that are predictive of the class but uncorrelated with one another. CFS computes a heuristic measure of the “merit” of a feature subset from pair-wise feature correlations and a formula adapted from test theory. Heuristic search is used to traverse the space of feature subsets in a reasonable time; the subset with the highest merit found during the search is reported [28].

b- Classifier subset evaluation

Is method of attribute subset evaluation techniques [30], which uses a classifier to evaluate the attribute set.

c- Relief Attribute Evaluation

The main idea of the Relief algorithm is to evaluate and estimate the quality of attributes according to distinguishing values between the instances that are near to each other [31]. Both Relief and its extension Relief are aware of the content information and can correctly estimate the quality of attributes in classification tasks with strong dependencies between attributes [32].

d- Wrapper Attributes Selection

It depends on an induction algorithm to estimate the merit of feature subsets [28].In this research to determine the most influencing and important variables that affect on the long term rainfall prediction out of the existing one, many attributes evaluator algorithms such as (correlation based feature selection subset evaluator, classifier subset evaluator, relief attribute evaluator and Wrapper subset evaluator) have been implemented with appropriate different search methods such as (best-first, evolutionary search, exhaustive search, genetic search, greedy stepwise, linear forward selection, PSO search, random search, scatter searchV1, subset size forward selection, Tabu Search and Ranker).

Data normalization

One of the steps of data pre-processing is data normalization. For example, normalization may improve the accuracy and efficiency of mining algorithms involving distance measurements [33]. The need to make harmony and balance between data, data must be normalized between 0 and 1. (Eq. 1) has been used to normalize the dataset in this study.

$$x_n = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \qquad (1)$$

Where x is actual data and $x_{\min}$ is minimum value of original attribute’s values and $x_{\max}$ is maximum value of original attribute’s values.

Supplied test set

With the base algorithms, different test options such as cross validation, percentage split and supplied test set have been tried [34]] and we found that supplied test set with ratio 70-30 for training and testing respectively has the best results. In case of Meta and ensemble models also the supplied test set with ratio 70-30 for training and testing respectively has been applied [35]. In ANFIS case different choices for dividing the dataset for training and testing have been tried, we used 60% for training and 40% for testing, 70% for training and 30% for testing, 80% for training and 20% for testing and 90% for training and 10% for testing respectively [36]. And the results showed that the best results have been obtained when we applied our ANFIS neuro-fuzzy model with ratio 70-30 of dataset for training and testing. In this study, we take the same ratio of 70% and the ratio of 30% for training and testing respectively for all proposed models.

Neuro-fuzzy inference systems

The architecture of the Neuro-Fuzzy Inference Systems that used in this study is ANFIS. Six layered ANFIS Model has been developed with the learning algorithm for training the network is hybridization of forward pass and backward pass, Input 1 is Date with 3 membership function (low, medium, high), input 2 is average minimum temperature with 3 membership function (low, medium, high), input 3 as relative humidity with 3 membership function (low, medium, high), input 4 as wind direction with 3 membership function (low, medium, high), and a single output as average monthly rainfall Precipitation whose degree of membership is Linear.

a- Membership function

In this study, the generalized bell shaped membership function has been used for the 4 input variables. The bell membership function for three different sets of parameters {a, b, c} is defined by (Eq. 2)

$$f(x;a,b,c)=\frac{1}{1+\left[\frac{x-c}{a}\right]^{2b}} \quad (2)$$

Where {a,b,c} is the parameter set. The parameters a and c represent the width and the center of the bell function, and b represents the slopes at the crossover points. As the values of these parameters change, the bell-shaped function varies accordingly, thus exhibiting various forms of membership functions on linguistic label Ai.

b- ANFIS Rules

For 4 rainfall predictors represented by the inputs date, minimum temperature, humidity and wind direction, having 3 categories namely low, medium, and high each, there would be 81 rules in the rule base; the output for each rule is written as a linear combination of input variables.

In the proposed ANFIS, the conjunction of the rule antecedents is evaluated by the operator product. Thus, the output of neuron i in Layer 3 obtained as:

$$y_i^{(3)}=\prod_{j=1}^{k} x_{ji}^{(3)} \quad (3)$$

where $x_{ji}^{(3)}$ are the inputs and $y_i^{(3)}$ is the output of rule neuron i in Layer 3.

c- Grid partitioning

For generating the initial fuzzy inference system (FIS), Grid Partitioning has been used. Once the grid partitioning technique is applied at the beginning of training, a uniformly partitioned grid which is defined by membership functions (MFs) with a random set of parameters is taken as the initial state of ANFIS. During training, this grid evolves as the parameters in the MFs change. With the grid partitioning technique, the number of MFs in the premise part of the rules must be determined.

d- Training of ANFIS model

Training of ANFIS model has been done using hybrid optimization method with error tolerance level 0.00001 for 100 epochs.

Hybrid learning algorithm combines the least-squares estimator with the gradient descent method.

- In the forward pass, a training set of input patterns is presented, neuron outputs are calculated on a layer-by layer basis, and rule consequent parameters are identified by the least-squares estimator.

- In the backward pass, the error signals are propagated back and the rule antecedent parameters are updated according to the chain rule.

Intelligent data analysis

The Base algorithms

1- IBK

It is a k-nearest-neighbour classifier that uses the same distance metric. The number of nearest neighbours can be specified explicitly in the object editor or determined automatically using leave-one-out cross-validation focus to an upper limit given by the specified value. A kind of different search algorithms can be used to speed up the task of finding the nearest neighbours. A linear search is the default but further options include KD-trees, ball trees, and so-called “cover trees” [37].

2- KStar

k-star algorithm can be defined as a method of cluster analysis which mainly aims at the partition of “n‟ observation into “k‟ clusters in which each observation belongs to the cluster with the nearest mean. We can describe K* algorithm as an instance based learner which uses entropy as a distance measure. The benefits are that it provides a consistent approach to handling of real valued attributes, symbolic attributes and missing values. K* is a simple, instance based classifier, similar to K-Nearest Neighbour (K-NN) [37]. New data instances, x, are assigned to the class that occurs most frequently amongst the k-nearest data points, y_j where j = 1, 2…k. Entropic distance is then used to retrieve the most similar instances from the data set. By means of entropic distance as a metric has a number of benefits including handling of real valued attributes and missing values. The K* function can be calculated as:

$$K^*(y_i, x) = -\ln P^*(y_i, x) \quad (4)$$

3- M5P

It is a model tree that generated in two stages, The first builds an ordinary decision tree, using as splitting criterion the maximization of the intra-subset variation of the target value. The second prunes this tree back by replacing subtrees with linear regression functions wherever this seems appropriate. M5rules algorithm produces propositional regression rules in IF-THEN rule format using routines for generating a decision list from M5′Model trees [38]. This model tree is used for numeric prediction and at each leaf it stores a linear regression model that predicts the class value of instances that reach the leaf. In determining which attribute is the best to split the portion T of the training data that reaches a particular node the splitting criterion is used. The standard deviation of the class in T is treated as a measure of the error at that node and calculating the expected reduction in error tests each attribute at that node. The attribute that is chosen for splitting maximizes the expected error reduction at that node. The standard deviation reduction (SDR), which is calculated by (5), is the expected error reduction.

$$SDR = sd(T) - \sum \frac{|T_i|}{|T|} \times sd(T_i) \qquad (5)$$

Base Meta Classifiers Used

1- Bagging

It is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid over fitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach [39].Bagging is a combination of bootstrapping and averaging used to decrease the variance part of prediction errors [40].

2- Vote

The vote is Meta learning scheme, which enables to create an ensemble of multiple base classifies. It provides a baseline method for combining classifiers. The default scheme is to average their probability estimates or numeric predictions, for classification and regression, respectively [41].

Ensemble methodology

1- Combination methods

There are two main methods for combining the base-classifiers outputs [42] weighting methods and Meta learning methods. Weighting methods are useful if the base-classifiers perform the same task and have comparable success. Meta-learning methods are best suited for cases in which certain classifiers consistently correctly classify, or consistently misclassify, certain instances.

Weighting methods:

When combining classifiers with weights, a classifier's classification has strength proportional to its assigned weight. The assigned weight can be fixed or dynamically determined for the specific instance to be classified.

Meta-combination methods:

Meta-learning means learning from the classifiers produced by the inducers and from the classifications of these classifiers on training data. The following sections describe the most well-known meta-combination methods. In this study, we used vote Meta-combination method to combine the base classifiers.

- Structure of ensemble classifiers

There are two styles to structure the classifiers of ensembles [43] parallel and cascading or hierarchical structure. In this study we use the Parallel Structure of ensemble classifiers. At this kind of structure all the individual classifiers are invoked independently, and their results are fused with a combination rule (e.g., average, weighted voting) or a meta-classifier (e.g., stacked generalization).

In this research, the meteorological dataset are used to train and test the system, each learner algorithm in the system is trained using the training data set, and then give an output. The outputs of all classifiers are combined using median probabilities as combination rule to give the final prediction.

- Classifiers Combination strategy

Combining rules are the simplest combination approach and it is probably the most commonly used in the multiple classifier system [44]. This combination approach is called non-trainable combiner, because combiners are ready to operate as soon as the classifiers are trained and they do not require any further training of the ensemble as a whole [45].

A theoretical framework for fixed rules combination was proposed by Josef Kittler et al. [46] they have discussed many possibilities of combining rule like the sum, product, max, min, average and median rules. In regression problems with vote meta scheme algorithm there are several methods for combination rules such as average of probabilities, minimum probability, maximum probability and median. In this research the median probabilities have been adopted as combination rule method because, it gives the best results for our dataset.

Median rule

Equation (6) can be used to compute the average a posteriori probability for each prediction over all the classifier outputs, i.e.

$$\text{assign } Z \rightarrow w_j \text{ if}$$

$$\frac{1}{R}\sum_{i=1}^{R} P(w_j \mid x_i) = \max_{k=1}^{m} \frac{1}{R}\sum_{i=1}^{R} P(w_k \mid x_i) \qquad (6)$$

Where:

Z is the example that has to predicted.

x_i Is given measurements, i=1,…, R.

R is the number of classifiers.

And w_k represent the possible predictions, k= 1,…, m.

Thus, the rule assigns an example to that prediction the average a posteriori probability of which is maximum. If any of the classifiers outputs an a posteriori probability for some prediction which is an outlier, it will affect the average and this in turn could lead to an incorrect decision. It is well known that a robust estimate of the mean is the median. It could therefore be more appropriate to base the combined decision on the median of the a posteriori probabilities. This then leads to the following rule:

$$\text{assign } Z \rightarrow w_j \text{ if}$$

$$\underset{i=1}{\overset{R}{med}}\, P(w_j \mid x_i) = \max_{k=1}^{m} \underset{i=1}{\overset{R}{med}}\, P(w_k \mid x_i) \qquad (7)$$

Performance criteria

Three different criteria are used in order to evaluate the effectiveness of the models and their ability to make precise predictions. The three criteria are Correlation Coefficient (CC) (Eq. 8), Root Mean Square Error (RMSE) (Eq. 9), and Mean Absolute Error (MAE) (Eq. 10). Higher CC, smaller MAE and RMAE show the better prediction effect [47]. That performance expressed below mathematically:

$$CC = \frac{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)\left(Y_i-\overline{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i-\overline{X}\right)^2\left(Y_i-\overline{Y}\right)^2}} \quad (8)$$

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(X_i-Y_i)^2}{n}} \quad (9)$$

$$\mathrm{MAE} = \sum_{i=1}^{n}\frac{\left|X_i-Y_i\right|}{n} \quad (10)$$

Where X_i is the observation data and Y_i computed data and n is the number of data. $\overline{X}$ is the mean of actual data and $\overline{Y}$ is the mean of the computed data.

Results and discussion

Table1 shows the different attributes evaluators with their appropriate search methods besides the number and order of the selected attributes.

As shown in figure 1 we can notice that as the result of applying different attributes evaluators with different search methods we have obtained 4 choices of dataset's attributes (1, 3, 4 and 7), the dominant choice was four attributes (Date, Minimum Temperature, Humidity and Wind Direction) with 75% as percentage ratio of accuracy. In this study, only the most influencing variables (Date, Minimum Temperature, Humidity and Wind Direction) that affect on the long term rainfall prediction out of the 7 variables have been used.

All models which have been developed by using the Base algorithms (IBK, KStar and M5P) or the Meta algorithms (Bagging and Vote) have been applied on the rain fall dataset with ratio of 70-30 for training and testing respectively and we obtained the following results which appear in Table 2.

As shown in Table 2, the biggest correlation coefficient 0.8901 has been achieved by the KStar base algorithm, the smallest mean absolute error was 0.0905 has been obtained by the IBK base algorithm and the lowest mean squared error 0.2285 has been accomplished by the KStar base algorithm. Figure 2 shows the comparison among both base and Meta algorithms according to the correlation coefficient, mean absolute error and root mean squared error.

According to time consuming to build and test models, we can see that the shortest time taken to build a model was 0.01 has been achieved by the both KStar base algorithm and Vote Meta algorithm. And also the shortest time taken to test model was 0.01 has been achieved by both M5P base algorithm and Bagging Meta algorithm.

As shown in Figure 2 according to correlation coefficient results the best one was 0.8901 has been obtained by KStar algorithm, M5P base algorithm came in the second order with correlation coefficient 0.8863. The results of Meta algorithms Bagging and Vote were 0.8529 and 0.8463 respectively too closed to each other. The worst correlation coefficient was obtained 0.8192 by IBK base algorithm.

On the other side, the lowest mean absolute error was achieved 0.0905 by IBK, both base algorithms M5P and KStar gave too closed values 0.1047 and 0.1091 respectively. Meta algorithms Bagging and Vote gave the highest mean absolute errors 0.1237 and 0.1287 respectively.

According to figure 3, the shortest time to build model 0.01 has been obtained by both KStar and Vote algorithms, IBK came in the second order with 0.02sec and M5P came third with 0.1sec for model building time. The longest building time was 0.16 by Bagging.

In the term of time taken to test model, the shortest one 0.01 was obtained by M5P and Bagging. The vote came second with 0.03 sec and IBK third with 0.34sec. The longest test time 4.59 sec has been achieved by KStar.

Table1. Results of the attribute selection

Attributes evaluator	Search method	No. of selected Attributes	Selected attributes
Correlation based Feature Selection subset evaluator	Best-first	4	Date, Min-T, Humidity, Wind D.
Correlation based Feature Selection subset evaluator	Evolutionary Search	4	Date, Min-T, Humidity, Wind D.
Correlation based Feature Selection subset evaluator	Exhaustive Search	4	Date, Min-T, Humidity, Wind D.
Correlation based Feature Selection subset evaluator	Genetic Search	4	Date, Min-T, Humidity, Wind D.
Correlation based Feature Selection subset evaluator	Greedy Stepwise	4	Date, Min-T, Humidity, Wind D.
Correlation based Feature Selection subset evaluator	Linear Forward Selection	4	Date, Min-T, Humidity, Wind D.
Correlation based Feature Selection subset evaluator	PSO Search	4	Date, Min-T, Humidity, Wind D.
Correlation based Feature Selection subset evaluator	Random Search	3	Min-T, Humidity, Wind D.
Correlation based Feature Selection subset evaluator	Scatter SearchV1	4	Date, Min-T, Humidity, Wind D.
Correlation based Feature Selection subset evaluator	Subset Size Forward Selection	4	Date, Min-T, Humidity, Wind D.
Correlation based Feature Selection subset evaluator	Tabu Search	4	Date, Min-T, Humidity, Wind D.
Classifier subset evaluator	Genetic Search	1	Wind D.
Classifier subset evaluator	Random Search	4	Date, Min-T, Humidity, Wind D.
Relief attribute evaluator	Ranker	7	station, Wind D, Date, Humidity, Min-T, Max-T, Wind S.
Wrapper subset evaluator	Genetic Search	1	Wind D.
Wrapper subset evaluator	Random Search	4	Date, Min-T, Humidity, Wind D.

Table 2. Comparison between the base algorithms and Meta algorithms

Base or Meta algorithm	Correlation coefficient	Mean absolute error	Root mean squared error	time taken to build model in sec	time taken to test model in sec
IBk	0.8192	0.0905	0.3005	0.02	0.34
KStar	0.8901	0.1091	0.2285	0.01	4.59
M5P	0.8863	0.1047	0.2322	0.1	0.01
Bagging	0.8529	0.1237	0.2614	0.16	0.01
Vote	0.8463	0.1287	0.267	0.01	0.03

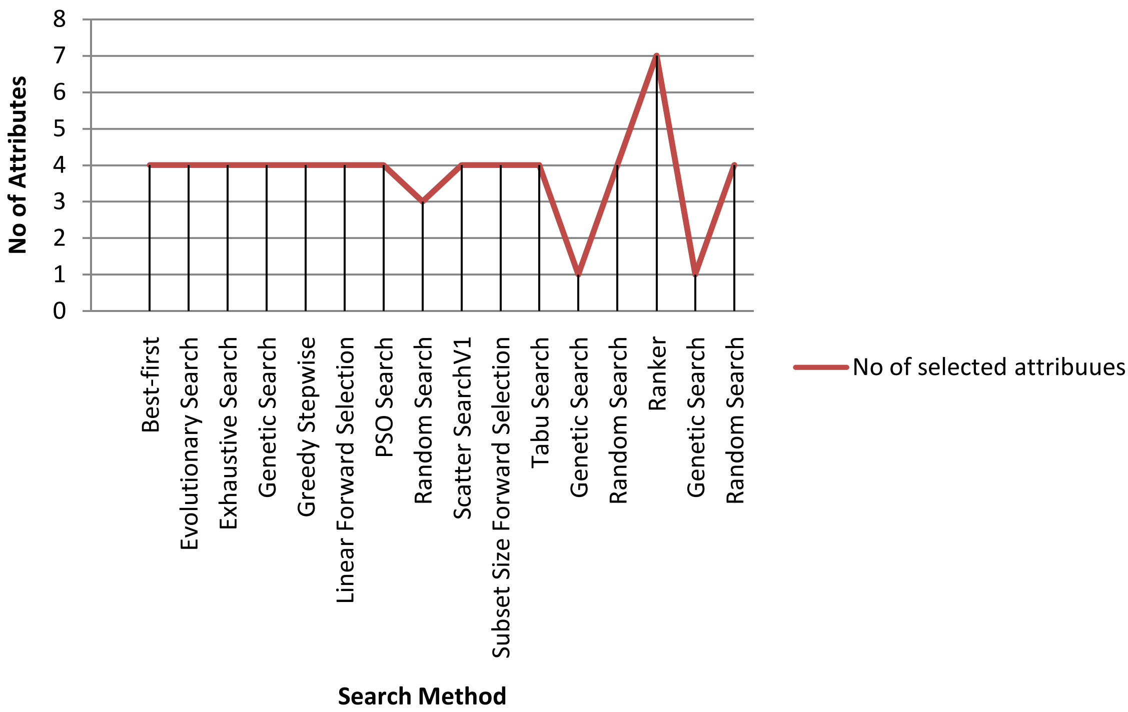

Figure 1. Search methods and the number of selected attributes.

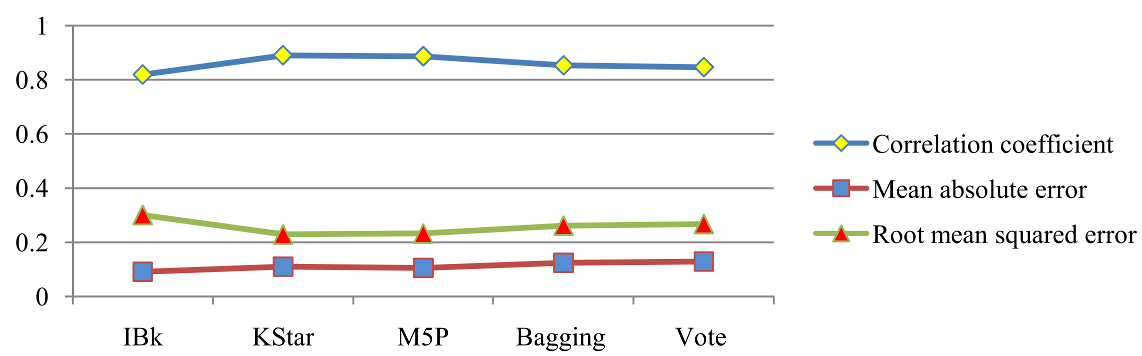

Figure 2. Comparison between base and Meta models according to the correlation coefficient, mean absolute error and root mean squared error

Table 3 shows the performance of the Ensemble models according to correlation coefficient, mean absolute error, root mean squared error, time taken to both build and test model. We construct an ensemble model of Meta Vote method combining with various base classifiers.Vote+2 algorithms (Kstar and IBK), Vote+3 algorithms (IBK, Kstar and M5P) and Vote+4 algorithms (IBk, Kstar, M5P, and Bagging).

As appears in Table 3 the best results (highest correlation coefficient, lowest of both mean absolute error and root mean squared error) achieved by Vote+3 ensemble method. Figure 4 shows the comparison among ensemble method algorithms according to correlation coefficient, mean absolute error and root mean squared error.

The shortest time for building and testing model has been achieved by Vote+2 ensemble method. Figure 5 displays a comparison between base and Meta algorithms according to the time taken to build model and time taken to test model in seconds.

As shown in Figure 4, ensemble Vote+3 outperformed the other models and produced the best results so far. The highest correlation coefficient 0.8986, the lowest of both mean absolute error 0.0888 and root mean squared error 0.1092 has been obtained ensemble Vote+3. Ensemble Vote+2 came in the second order with a 0.8861correlation coefficient, 0.1311 mean absolute errors and 0.2319 root mean squared error. The worst results have been obtained by ensemble Vote-4, the lowest correlation coefficient 0.8803 and the highest of both mean absolute error 0.1376 and root mean squared error 0.2728.

As shown in Figure 5, the shortest time to build model 0.08sec has been obtained by ensemble Vote+2 algorithms, ensemble Vote+3 came in the second order with 0.09sec. The longest building time was 0.78sec by ensemble Vote+4.

In the term of time taken to test model, the shortest one 4.44sec was obtained by ensemble Vote+2. Ensemble Vote+3 came second with 4.71 sec. The longest test time 4.77 sec has been achieved by ensemble Vote+4.

Table 4 displays a comparison between the best Ensemble model and its base algorithms according to correlation coefficient, mean absolute error, root mean squared error, time taken to build model and time taken to test model.

Table 3. Comparison between the ensemble models according to the correlation coefficient, mean absolute error, root mean squared error, time taken to build model and time taken to test model

Ensemble algorithm	Correlation coefficient	Mean absolute error	Root mean squared error	time taken to build model in sec	time taken to test model in sec
Vote+2	0.8861	0.1311	0.2319	0.08	4.44
Vote+3	0.8986	0.0888	0.1092	0.09	4.71
Vote+4	0.8803	0.1376	0.2728	0.78	4.77

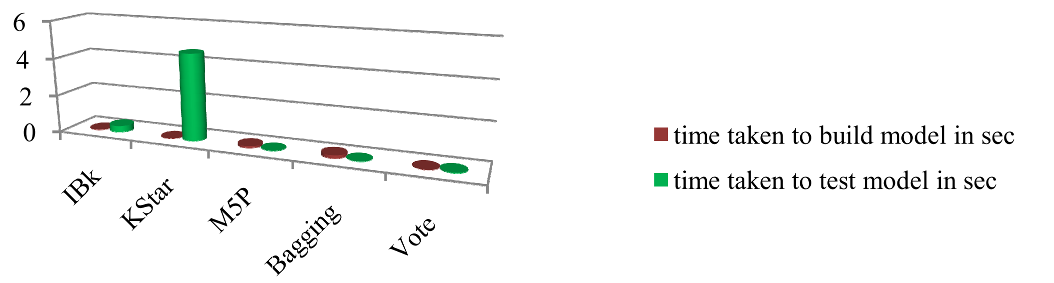

Figure 3. Comparison between base and Meta models according to time taken to build and test model

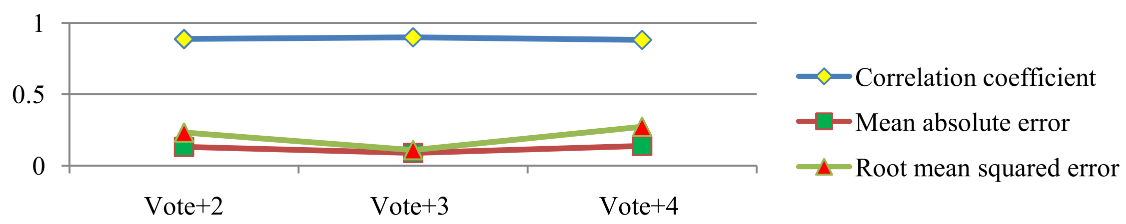

Figure 4. Comparison between ensemble models according to the correlation coefficient, mean absolute error and root mean squared error

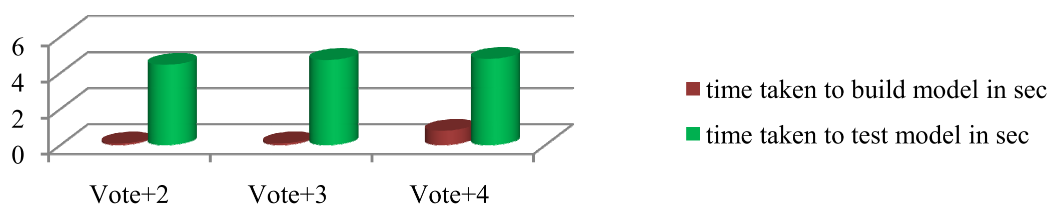

Figure 5. Comparison between the ensemble models according to time taken to build and test model.

As shown in table 4, ensemble Vote+3 outperformed its own basic algorithms and produced the best results. It obtained the highest correlation coefficient 0.8986, the lowest of both mean absolute error 0.0888 and root mean squared error 0.1092. Figure 6 shows the comparison between the best ensemble method Vote+3 and its basic algorithms according to correlation coefficient, mean absolute error and root mean squared error.

Also, Vote+3 achieved the longest time for both building model 0.09 sec and testing model 4.71 sec. Figure 11 displays a comparison between the best ensemble method Vote+3 and its basic algorithms according to time taken to build model and time taken to test model in seconds.

As shown in figure 6, we can notice that Vote+3 surpassed its base algorithms, because it achieved the highest correlation coefficient 0.8986, the lowest of both mean absolute error 0.0888 and root mean squared error 0.1092,. KStar came in the second order in term of both correlation coefficient 0.8901and root mean squared error 0.2285 but it had the highest mean absolute error0.1091. Thirdly in term of correlation coefficient came M5P 0.8863. IBK obtained the worst of both correlation coefficient 0.8192 and root mean squared error 0.3005, but at the same time it achieved the second best mean absolute error 0.0905.

From the previous results, we conclude that diversity is a key factor for yielding satisfactory accuracy-generalization performance with classifier ensembles.

As shown in Figure 7, ensemble Vote+3 obtained the second worse time to build model 0.09 sec and the longest testing time 4.71sec. The shortest building time 0.01sec was obtained by KStar algorithm. IBK came in the second order for both building time 0.02 sec and testing time 0.34sec. The longest building time 0.1sec has been acquired by M5P algorithm, but at the same time M5P achieved the shortest test time 0.01sec. KStar obtained the second worse test time 4.59 sec.

As Table 5 shows, ANFIS model outperformed the ensemble Vote+3 model in term of all performance criteria, it achieved the highest correlation coefficient 0.90, lowest of both mean absolute error 0.0074 and root mean squared error 0.0861.

As shown in figure 8, results of ANFIS and ensemble Vote+3 are too closed, but the results of ANFIS model are more accurate. ANFIS model provided higher correlation coefficient and lower of both mean absolute error and root mean squared error comparing with ensemble Vote+3.

Table 4. Comparison between the best Ensemble model and its base algorithms according to correlation coefficient, mean absolute error, root mean squared error, time taken to build model and time taken to test model

Algorithm	Correlation coefficient	Mean absolute error	Root mean squared error	time taken to build model in sec	time taken to test model in sec
IBk	0.8192	0.0905	0.3005	0.02	0.34
KStar	0.8901	0.1091	0.2285	0.01	4.59
M5P	0.8863	0.1047	0.2322	0.1	0.01
Vote+3	0.8986	0.0888	0.1092	0.09	4.71

Table 5. Comparison between the best ensemble Vote+3 and ANFIS model according to the correlation coefficient, mean absolute error and root mean squared error

Algorithm	Correlation coefficient	Mean absolute error	Root mean squared error
ANFIS	0.90	0.0074	0.0861
Vote+3	0.8986	0.0888	0.1092

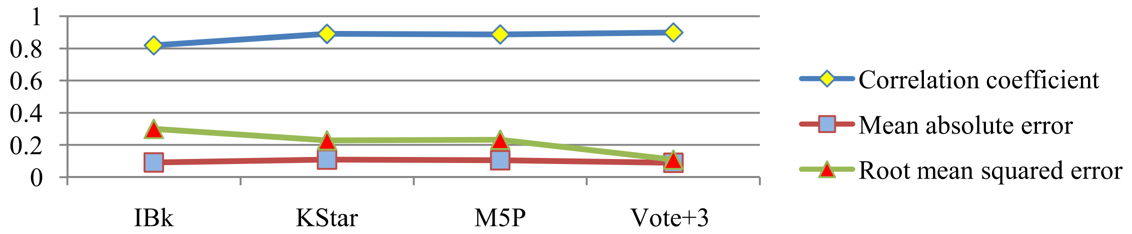

Figure 6. Comparison between the best ensemble Vote+3 and its basic algorithms according to correlation coefficient, mean absolute error and root mean squared error

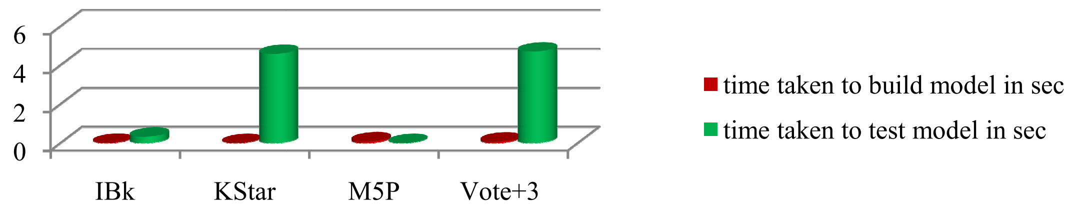

Figure 7. Comparison between the best ensemble method Vote+3 and its basic algorithms according to time taken to build model and time taken to test model in seconds.

Figure 8. Comparison between ensemble Vote+3 and ANFIS model according to correlation coefficient, mean absolute error and root mean squared error.

As shown in figure 9, ANFIS model outperformed all other proposed models and produced the highest correlation coefficient 0.90, lowest of both mean absolute error 0.0074 and root mean squared error 0.0861. Ensemble Vote+3 algorithm came in the second order and too closed to ANFIS with 0.8986 correlation coefficient, 0.0888 mean absolute error and 0.1092 root mean squared error.

Since the atmosphere is chaotic and weather data is nonlinear following a very irregular trend, soft computing techniques are considered to be better approach for developing effective and reliable nonlinear predictive models for weather analysis.

Conclusions

It is important to have reliable and accurate techniques to forecast rainfall in the long term. In this study, we proposed nine models based on different soft computing technologies namely IBK, KStar, M5P, Vote, Bagging, ensemble technology and ANFIS. Finally, we compared them until a model that produced satisfactory results was obtained.

Different evaluation measures such as correlation coefficient, mean absolute error, root mean squared error, time taken to build model and time is taken to test model in seconds have been used for comparing different models to determine which one is the highest performance.

Dividing dataset into 70-30 for training and testing respectively considered the best choice for dividing our dataset, since it provided the best results comparing with the other choices for both training and testing phases.

In this study, attribute selection has been made by using several search methods and it determined the most important attributes which influenced on forecasting process (minimum temperatures, humidity and wind direction) as predictors for long term rainfall prediction in Sudan. The empirical results also indicate that ANFIS neuro fuzzy and ensemble Vote+3 models perform better than others, however, ANFIS model outperformed the ensemble Vote+3model and produces better results. Also ANFIS model has the ability to interpret and explain its results by using rules, and this feature is not available for other models.

In general, these results indicate that more complicated models with high complexity such as ANFIS and ensemble sometimes yield better forecasts than simpler ones.

Table 6. Comparison between the Vote+3 and ANFIS with models in the literature

Technology used	Application	Performance	Reference
ANN architecture, with Genetic Optimizer (GO)	seasonal rainfall forecasting	CC = 0.8951	[48]
Evolving Fuzzy Neural Network (EFuNN)	forecast the monthly rainfall	RMSE = 0.0901	[49]
Multivariate Adaptive Regression Splines (MARS)	forecast monthly rainfall	RMSE = 0.0780	[7]
multilayered artificial neural network	seasonal rainfall prediction	MSE = 0.42	[50]
Bagging classifier using REPTree classifier as a base-learner.	Daily wind speed forecasting	CC = 0,8154 RMSE = 0,8774	[51]
ANFIS using grid partition and hybrid algorithm for learning	Monthly rainfall prediction	CC = 0.90 RMSE = 0.086139	proposed ANFIS
Ensemble IBK, Kstar, M5P	Monthly rainfall prediction	CC = 0.8986 RMSE = 0.1092	proposed Ensemble Vote+3

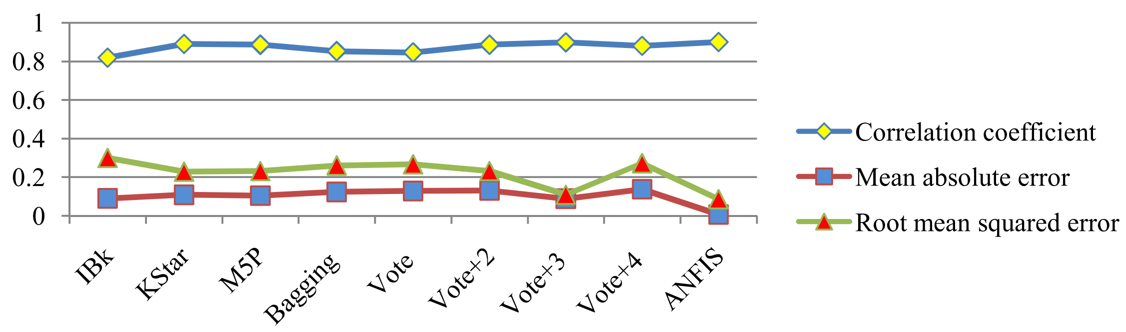

Figure 9. Comparison between all models according to correlation coefficient, mean absolute error and root mean squared error.

References

[1] Lynch P. 2008. The origins of computer weather prediction and climate modeling. *J Comput Phy* 227: 3431–3444.

[2] Bushara N, Abraham A. 2013. Computational intelligence in weather forecasting: A review. *J Net Innov Comput*1: 320–331.

[3] Jagannathan P. 1974. Long range weather forecasting, forecasting manual part IV-comprehensive articles on selected topics. India Meteorological Department.

[4] Castellanosand F, James N. 2009. Average hourly wind speed forecasting with ANFIS", 1[1th] American Conference on Wind engineering, San Juan, Puerto Rico.

[5] Bekuretsion B, Beshah T. 2013. Prediction of rainfall for Ethiopian Upper Blue Nile basin using soft computing models. *HiLCoE J Comput Sci Technol* 1: 26–34.

[6] Sharifi S, Delirhasannia R, Nourani V, Sadraddini A, Ghorbani A. 2013. Using artificial neural networks (ANNs) and adaptive neuro-fuzzy inference system (ANFIS) for modeling and sensitivity analysis of effective rainfall. Recent Advances in

Continuum Mechanics, *Hydrol Ecol* 4: 133–139.

[7] Abraham A, Steinberg D, Philip N. 2001. Rainfall forecasting using soft computing models and multivariate adaptive regression splines. *IEEE SMC Transactions*1: 1–6.

[8] Patel J, Parekh F. 2014. Forecasting rainfall using adaptive neuro-fuzzy inference system (ANFIS). *IJAIEM* 3: 262–269.

[9] Niksaz P, Latif A. 2014. Rainfall events evaluation using adaptive neural-fuzzy inference system. *Int J Inf Technol Comput Sci* 9: 46–51.

[10] Sethi N, Garg K. 2014. Exploiting data mining technique for rainfall prediction. *Int J Comput Sci Inf Technol* 5: 3982–3984.

[11] Singhratina N, Rajagop B, Clark M, Kumar K. 2005. Seasonal forecasting of Thailand summer monsoon rainfall. *Int J Climatol* 25: 649–664.

[12] Zaw W, Naing T. 2008. Empirical statistical modeling of rainfall prediction over Myanmar. *World Acad Sci Eng Technol* 2: 492–495.

[13] Ji S, Sharma S, Yu B, Jeong D. 2012. Designing a rule-based hourly rainfall prediction model. In Proceedings of the IEEE 13th International Conference on IRI: 303–308.

[14] Tukey J. 1981. Exploratory data analysis. *Biomet J* 23: 413–414.

[15] Hansen L, Salamon P. 1990. Neural network ensembles. *IEEE T Pattern Anal* 12: 993–1001.

[16] Schapire R. 1990. The strength of weak learn ability. *Mach Learn* 5: 197–227.

[17] Freund Y, Schapire R. 1996. Experiments with a new boosting algorithm. In: Machine learning: proceedings of the thirteenth international conference, 325–332.

[18] Merler S, Caprile B, Furlanello C. 2007. Parallelizing AdaBoost by weights dynamics. *Comput Stat Data Anal* 51: 2487–2498.

[19] Zhang C, Zhang J. 2008. A local boosting algorithm for solving classification problems. *Comput Stat Data Anal* 52: 1928–1941.

[20] Zhang L, Suganthan P. 2014. Random forests with ensemble of feature spaces. *Pattern Recogn* 47: 3429–3437.

[21] Li L, Stolkin R, Jiao L, Liu F, Wang S. 2014. A compressed sensing approach for efficient ensemble learning. *Pattern Recogn* 47: 3451–3465.

[22] Canedo V, Maroño N, Betanzos A. 2014. Data classification using an ensemble of filters. *Neurocomputing* 135: 13–20.

[23] Nascimento D, Coelho A, Canuto A. 2014. Integrating complementary techniques for promoting diversity in classifier ensembles: A systematic study. *Neurocomputing* 138: 347–357.

[24] Hu Q, Li L, Wu X, Schaefer G, Yu D. 2014. Exploiting diversity for optimizing margin distribution in ensemble learning. *Knowl-Based Syst* 67: 90–104.

[25] Yin X, Huang K, Yang C, Hao H. 2014. Convex ensemble learning with sparsity and diversity. *Inform Fusion* 20: 49–59.

[26] Own H, Abraham A. 2014. A Novel-weighted rough set-based meta learning for ozone day prediction. *Acta Polytec Hung* 11.

[27] Faulina R. 2013. Hybrid ARIMA-ANFIS for rainfall prediction in Indonesia. *Int J Sci Res* 2: 159–162.

[28] Freund Y, Schapire RE. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. *CoLT* 23–37.

[29] Freund Y, Schapire RE. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Sys Sci* 55: 119–139.

[30] Nachev A, Stoyanov B. 2012. Product quality analysis using support vector machines. *Int J Information Models Analys* l: 179–192.

[31] Kira K, Rendell L. 1992. A practical approach to feature selection. In proceedings of the ninth international workshop on Machine learning, 249–256.

[32] Kononenko I. 1994. Estimating attributes: analysis and extensions of RELIEF. in Machine Learning: *ECML* 94: 171–182.

[33] Shahi A, Atan R, Sulaiman N. 2009. An effective Fuzzy C-Mean and type-2 fuzzy

logic for weather forecasting. *J Theor Appl Inf Technol* 5: 556–567.

[34] Bushara N, Abraham A. 2014. Weather forecasting in Sudan using machine learning schemes. *JNIC* 2: 309–317.

[35] Bushara N, Abraham A. 2015. Novel ensemble method for long term rainfall prediction. *IJCISIMA* 7: 116–130.

[36] Bushara N, Abraham A. 2015. Using adaptive neuro-fuzzy inference system (ANFIS) to improve the long-term rainfall forecasting. *JNIC* 3: 146–158.

[37] Vijayarani S, Muthulakshmi M. 2013. Comparative analysis of bayes and lazy classification algorithms. *IJARCCE* 2: 3118–3124.

[38] Average W, Kotsiantis S, Kostoulas A, Lykoudis S, Argiriou A, Menagias K. 2007. Using data mining techniques for estimating minimum, maximum and average daily temperature values. *IJCISCE* 1: 382–386.

[39] Jeong D, Kim Y. 2005. Rainfall-runoff models using artificial neural networks for ensemble streamflow prediction. *Hydrol Process* 19: 3819–3835.

[40] Breiman L. 1996. Bagging predictors. *Mach Learn* 24:123–140.

[41] Witten I, Frank E, Hall M. 2011. Data mining practical machine learning tools and techniques, 3rd edt. Elsevier.

[42] Rokach L. 2010. Ensemble-based classifiers. *Artif Intell Rev* 33: 1–39.

[43] Seni G, Elder J. 2010. Ensemble methods in data mining: Improving accuracy through combining predictions. Morgan and Claypool Publishers.

[44] Felföldi L. 2008. Classifier combination systems and their application in human language technology. PhD. Thesis, University of Szeged, Hungary.

[45] Kuncheva L. 2003. Fuzzy vs non-fuzzy in combining classifiers designed by boosting. IEEE Trans Fuzzy Syst 11: 729–741.

[46] Kittler J, Hatef M, Duin R, Matas J. 1998. On combining classifiers. *IEEE T Pattern Anal Mach Intell* 20: 226–239.

[47] Ullah N, Choudhury P. 2013. Flood flow modeling in a river system using adaptive neuro-fuzzy inference system. *Environ Manag Sustain Develop* 2: 54–68.

[48] D-Nagesh K, Janga R, Rajib M. 2007. Regional rainfall forecasting using large scale climate teleconnections and artificial intelligence techniques. *J Intell Syst* 16: 307–322.

[49] Abraham A, S-Philip N, B-Joseph K. 2001. Will we have a wet summer? long-term rain forecasting using soft computing" models, modeling and simulation 2001, in: Proceedings of the 15th European Simulation Multi Conference on Society for Computer Simulation International, Prague, Czech Republic, 1044–1048.

[50] A0bhishek K, Kumar A, Ranjan R, Kumar S. 2012. A rainfall Prediction model using artificial neural network. *IEEE Control Syst Grad Res Coll* 82–87.

[51] Zontul M, Aydin F, Dogan G, Sener S, Kaynar O. 2013. Wind speed forecasting using reptree and bagging methods in Kirklareli-Turkey. *J Theor Appl Info Technol* 56: 17–29.