International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 7 (2015) pp. 116-130
© MIR Labs, www.mirlabs.net/ijcisim/index.html

Novel Ensemble Method for Long Term Rainfall Prediction

Nazim Osman Bushara[1] and Ajith Abraham[2]

[1] Faculty of computer science and information technology, Sudan University for Science and Technology (SUST),
P.O. Box 12094, SUDAPOST, Khartoum, Post Code 111 11, Sudan
nazim_ob@yahoo.com

[2] Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence,
P.O. Box 2259, Auburn, Washington 98071-2259, USA
ajith.abraham@ieee.org

***Abstract*: In the field of weather forecasting especially in rainfall prediction many researchers employed different data mining techniques to deal with that problem by using different predictors. This paper proposes a novel method to develop long-term weather forecasting model for rainfall prediction by using ensemble technique. Monthly meteorological data that obtained from Central Bureau of Statistics Sudan from 2000 to 2012, for 24 meteorological stations distributed among the country has been used. The dataset contained date, minimum temperature relative humidity, wind direction and rainfall as the predictors. In the experiments we built 10 base algorithm models (Gaussian Processes, Linear Regression, Multilayer Perceptron, IBk, KStar, Decision Table, M5Rules, M5P, REP Tree and User Classifier.), 7 Meta algorithms(Additive Regression, Bagging, Multi Scheme, Random Subset, Regressionby Discretization, Stacking, and Vote).The new novel ensemble method has been constructed based of Meta classifier Vote combining with three base classifiers IBK, K-star and M5P.The models have been evaluated by using correlation coefficient; mean absolute error and root mean-squared error as performance metrics. Also we use the both time taken to build the model and time taken to test model on supplied test set to compare and differentiate among the models results show that the new novel ensemble method has the best performance comparing to both basic and Meta algorithms.**

***Keywords*:** Long term weather forecasting, Rainfall prediction, Data Mining, Ensemble, Meta algorithm.

I. Introduction

Weather forecasting is the application of science and technology to predict the state of the atmosphere for a future time and a given location [1], Human kind has attempted to predict the weather since ancient times. One of the main fields of weather forecasting is rainfall prediction, which is important for food production plan, water resource management and all activity plans in the nature. The occurrence of prolonged dry period or heavy rain at the critical stages of the crop growth and development may lead to significant reduce crop yield.

There are several types of weather forecasts made in relation to time:

- A short-range forecast is a weather forecast made for a time period up to 48 hours.
- Extended forecasts are for a period extending beyond three or more days (e.g. a three to five-day period) from the day of issuance.
- Medium range forecasts are for a period extending from about three days to seven days in advance.
- Long-range forecasts are for a period greater than seven days in advance but there are no absolute limits to the period.

The success of the seasonal forecasts depends on a detailed knowledge of how the atmosphere and ocean interact. Short-range forecast predictions, where the forecast is made for a time period for today or tomorrow (up to 48 hours), are generally more accurate than the other types of forecasts. Weather forecasts still have their limitations despite the use of modern technology and improved techniques to predict the weather. For example, weather forecasts for today or tomorrow are likely to be more dependable than predictions about the weather about two weeks from now. Some sources state that weather forecast accuracy falls significantly beyond 10 days [2]. Weather forecasting is complex and not always accurate, especially for days further in the future, because the weather can be chaotic and unpredictable. For example, rain or snow cannot always be predicted with a simple yes or no. Moreover, the Earth’s atmosphere is a complicated system that is affected by many factors and can react in different ways.

Long-range weather forecasts are widely used in the energy industry, despite their limited skill; long-range forecasts can still be a valuable tool for managing weather risk.

Knowledge Discovery in Databases (KDD) is an automatic, exploratory analysis and modeling of large data repositories.

KDD is the organized process of identifying valid, novel, useful, and understandable patterns from large and complex data sets. Data Mining (DM) is the core of the KDD process [3], involving the inferring of algorithms that explore the data, develop the model and discover previously unknown patterns. The model is used for understanding phenomena from the data, analysis and prediction.

Data Mining is the search for the relationships and global patterns that exist in large databases but are hidden among vast amount of data, such as the relationship between patient data and their medical diagnosis [4]. This relationship represents valuable knowledge about the database, and the objects in the database, if the hidden database is a faithful mirror of the real world registered by the database. Data Mining refers to using a variety of techniques to identify nuggets of information or decision making knowledge in the database and extracting these in such a way that they can be put to use in areas such as decision support, prediction, forecasting and estimation. The data is often voluminous, but it has low value and no direct use can be made of it. It is the hidden information in the data that is useful [4].

Meteorological data mining is a form of Data mining [5] concerned with finding hidden patterns inside largely available meteorological data, so that the information retrieved can be transformed into usable knowledge. Useful knowledge can play important role in understanding the climate variability and climate prediction. In turn, this understanding can be used to support many important sectors that are affected by climate like agriculture, vegetation, water resources and tourism.

Meta-learning is a technique that seeks to compute higher-level classifiers (or classification models), called meta-classifiers, that integrate in some principled fashion multiple classifiers computed separately over different databases. Meta-learning means learning from the classifiers produced by the inducers and from the classifications of these classifiers on training data.

Meta-learning improves efficiency by executing in parallel the base-learning processes (each implemented as a distinct serial program) on (possibly disjoint) subsets of the training data set (a data reduction technique). This approach has the advantage, first, of using the same serial code without the time-consuming process of parallelizing it, and second, of learning from small subsets of data that fit in main memory. Meta-learning improves predictive performance by combining different learning systems each having different inductive bias(e.g. representation, search heuristics, search space) [6]. By combining separately learned concepts, meta-learning is expected to derive a higher level learned model that explains a large database more accurately than any of the individual learners. Furthermore, meta-learning constitutes a scalable machine learning method since it can be generalized to hierarchical multi-level meta-learning.

The idea of ensemble methodology is to build a predictive model by integrating multiple models. It is well-known that ensemble methods can be used for improving prediction performance [7].

Building an ensemble consists of two steps: (1) constructing varied models and (2) combining their estimates. One may generate component models by, for instance, varying case weights, data values, guidance parameters, variable subsets, or partitions of the input space. Combination can be accomplished by voting, but is primarily done through model estimate weights [8].

Diversity is a crucial condition for obtaining accurate ensembles [9, 10, 11, and 12]. According to [13], diversified classifiers lead to uncorrelated classifications, which in turn improve classification accuracy. However, in the classification context, there is no complete and agreed upon theory to explain why and how diversity between individual models contributes toward overall ensemble accuracy [14].

An important aspect of ensemble methods is to determine how many base classifiers and which classifiers should be included in the final ensemble. Several algorithms, such as bagging, predetermine the ensemble size, by using a controlling parameter such as number of iterations that can be set by the user. Other ensemble algorithms try to determine the best ensemble size while training. When new members are added to the ensemble, we check if the performance of the ensemble has improved. If it is not, the procedure stops and no new base classifier are trained. Usually these algorithms also have a controlling parameter, which bounds the number of base classifiers in the ensemble. An algorithm that decides when a sufficient number of classification trees have been created was proposed by Robert et al. [15].

Ensemble methodology imitates our second nature to seek several opinions before making a crucial decision. The core principle is to weigh several individual pattern classifiers, and combine them in order to reach a classification that is better than the one obtained by each of them separately. Researchers from various disciplines such as pattern recognition, statistics, and machine learning have explored the use of ensemble methods since the late seventies. Given the growing interest in the field, it is not surprising that researchers and practitioners have a wide variety of methods at their disposal. An ensemble is largely characterized by the diversity generation mechanism and the choice of its combination procedure.

While ensemble approaches to classification usually make use of non-linear combination methods like majority voting; regression problems are naturally tackled by linearly weighted ensembles. These types of ensembles have a much clearer framework for explaining the role of diversity than voting methods. In particular the Ambiguity decomposition [10].

II. Related Research Works

The ensemble idea in supervised learning has been investigated since the late seventies. Tukey [16] suggested combining two linear regression models. The main progress in the field was achieved during the Nineties. Hansen and Salamon [17] have suggested an ensemble of similarly configured neural networks to improve the predictive performance of a single one. At the same time Schapire [18] laid the foundations for the award winning AdaBoost Freund and Schapire [19] 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; Merler et al. [20] have developed the P-AdaBoost algorithm, which is a distributed version of AdaBoost. 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.

Zhang and Zhang [21] have proposed a new boosting-by-resampling version of Adaboost. 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.

Alhamdoosh and Wang [22] have employed the random vector functional link (RVFL) networks as base components, and incorporated with the NCL strategy for building neural network ensembles. The basis functions of the base models are generated randomly and the parameters of the RVFL networks can be determined by solving a linear equation system. An analytical solution is derived for these parameters, where a cost function defined for NCL and the well known least squares method are used. To examine the merits of their proposed algorithm, a comparative study was carried out with nine benchmark datasets. Results indicate that their approach outperforms other ensembling techniques on the testing datasets in terms of both effectiveness and efficiency.

In [23] DeWeberand Wagner have compared four models with different groups of predictors to determine how well water temperature could be predicted by climatic, landform, and land cover attributes, and used the median prediction from an ensemble of 100 ANNs as their final prediction for each model. The final model included air temperature, landform attributes and forested land cover and predicted mean daily water temperatures with moderate accuracy as determined by root mean squared error (RMSE) at 886 training sites with data from 1980 to 2009 (RMSE = 1.91 _C). Based on validation at 96 sites (RMSE = 1.82) and separately for data from 2010 (RMSE = 1.93), a year with relatively warmer conditions, the model was able to generalize to new stream reaches and years. The most important predictors were mean daily air temperature, prior 7 day mean air temperature, and network catchment area according to sensitivity analyses. Forest land cover at both riparian and catchment extents had relatively weak but clear negative effects. Predicted daily water temperature averaged for the month of July matched expected spatial trends with cooler temperatures in headwaters and at higher elevations and latitudes. Their ANN ensemble is unique in predicting daily temperatures throughout a large region, while other regional efforts have predicted at relatively coarse time steps. The model may prove a useful tool for predicting water temperatures in sampled and un sampled rivers under current conditions and future projections of climate and land use changes, thereby providing information that is valuable to management of river ecosystems and biota such as brook trout.

Li et al. [24] have explored the influence of the classification confidence of the base classifiers in ensemble learning and obtain some interesting conclusions. First, they extended the definition of ensemble margin based on the classification confidence of the base classifiers. Then, an optimization objective is designed to compute the weights of the base classifiers by minimizing the margin induced classification loss. Several strategies were tried to utilize the classification confidences and the weights. It is observed that weighted voting based on classification confidence is better than simple voting if all the base classifiers are used. In addition, ensemble pruning can further improve the performance of a weighted voting ensemble. They also have compared the proposed fusion technique with some classical algorithms. The experimental results also show the effectiveness of weighted voting with classification confidence.

Zhang and Suganthan[25] proposed a new method to improve the performance of the Random Forests by increasing the diversity of each tree in the forests and there by improve the overall accuracy. During the training process of each individual tree in the forest, different rotation spaces are concatenated into a higher space at the root node. Then the best split is exhaustively searched within this higher space. The location where the best split lies decides which rotation method to be used for all subsequent nodes. The performance of the proposed method here is evaluated on 42 benchmark data sets from various research fields and compared with the standard Random Forests. The results showed that the proposed method improves the performance of the Random Forests in most cases.

Salih and Abraham [26] proposed a novel ensemble health care decision support for assisting an intelligent health monitoring system, their ensemble method was constructed based of Meta classifier voting combining with three base classifiers J48, Random Forest and Random Tree algorithms. The results obtained from the experiments showed that the proposed Ensemble method achieved better outcomes that are significantly better compared with the outcomes of the other Base and Meta base classifiers.

Li et al. [27] have presented a method for improved ensemble learning, by treating the optimization of an ensemble of classifiers as a compressed sensing problem. Ensemble learning methods improve the performance of a learned predictor by integrating a weighted combination of multiple predictive models. Ideally, the number of models needed in the ensemble should be minimized, while optimizing the weights associated with each included model. They solved this problem by treating it as an example of the compressed sensing problem, in which a sparse solution must be reconstructed from an under- determined linear system. Compressed sensing techniques are then employed to find an ensemble, which is both small and effective. The experiments showed that their method gave better accuracy, while being significantly faster than the compared methods

Chen, et al. [28] have proposed a unified evolutionary training scheme (UETS) which can either train a generalized feed forward neural network or construct an ANN ensemble. The performance of the UETS was evaluated by applying it to solve the n-bit parity problem and the classification problems on five datasets from the UCI machine-learning repository. By comparing with the previous studies, the experimental results reveal that the neural networks and the ensembles trained by the UETS have very good classification ability for unseen cases.

In the RAndom k-labELsets (RAKEL) algorithm, each member of the ensemble is associated with a small randomly selected subset of k labels. Then, a single label classifier is trained according to each combination of elements in the subset. Rokach et al. [29] have adopted a similar approach, however, instead of randomly choosing subsets, they selected the minimum required subsets of k labels that cover all labels and meet additional constraints such as coverage of inter-label correlations. Construction of the cover is achieved by formulating the subset selection as a minimum set covering problem (SCP) and solving it by using approximation algorithms. Every cover needs only to be prepared once by offline algorithms. Once prepared, a cover may be applied to the classification of any given multi-label dataset whose properties conform with those of the cover. The contribution of their work was two-fold. First, they introduced SCP as a general framework for constructing label covers while allowing the user to incorporate cover construction constraints. They demonstrated the effectiveness of this framework by proposing two construction constraints whose enforcement produces covers that improve the prediction performance of random selection. Second, they provided theoretical bounds that quantify the probabilities of random selection to produce covers that meet the proposed construction criteria. The experimental results indicated that the proposed methods improve multi-label classification accuracy and stability compared with the RAKEL algorithm and to other state-of-the-art algorithms.

One of the most important steps in the design of a multi-classifier system (MCS), also known as ensemble, is the choice of the components (classifiers). This step is very important to the overall performance of a MCS since the combination of a set of identical classifiers will not outperform the individual members. The ideal situation would be a set of classifiers with uncorrelated errors – they would be combined in such a way as to minimize the effect of these failures, Canuto et al. [30] have presented an extensive evaluation of how the choice of the components (classifiers) can affect the performance of several combination methods (selection-based and fusion-based methods). An analysis of the diversity of the MCSs when varying their components is also performed. As a result of this analysis, it is aimed to help designers in the choice of the individual classifiers and combination methods of an ensemble.

The idea of ensemble is adapted for feature selection. Canedo et al. [31] have proposed an ensemble of filters for classification, aimed at achieving a good classification performance together with a reduction in the input dimensionality. With this approach, they tried to overcome the problem of selecting an appropriate method for each problem at hand, as it is overly dependent on the characteristics of the datasets. The adequacy of using an ensemble of filters rather than a single filter was demonstrated on synthetic and real data, paving the way for its final application over a challenging scenario such as DNA microarray classification.

Jin et al. [32] have proposed a fuzzy ARTMAP (FAM) ensemble approach based on the improved Bayesian belief method is presented and applied to the fault diagnosis of rolling element bearings. First, by the statistical method, continuous Morlet wavelet analysis method and time series analysis method many features are extracted from the vibration signals to depict the information about the bearings. Second, with the modified distance discriminant technique some salient and sensitive features are selected. Finally, the optimal features are input into a committee of FAMs in different sequence, the output from these FAMs is combined and the combined decision is derived by the improved Bayesian belief method. The experiment results show that the proposed FAMs ensemble can reliably diagnose different fault conditions including different categories and severities, and has a better diagnosis performance compared with single FAM.

Studies have provided theoretical and empirical evidence that diversity is a key factor for yielding satisfactory accuracy-generalization performance with classifier ensembles. Nascimento et al [33] have tried to empirically assess the impact of using, in a sequential manner, three complementary approaches for enhancing diversity in classifier ensembles. 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.

Hu et al. [34] have proposed a novel ensemble learning algorithm named Double Rotation Margin Forest (DRMF) that aims to improve the margin distribution of the combined system over the training set. They utilized random rotation to produce diverse base classifiers, and optimize the margin distribution to exploit the diversity for producing an optimal ensemble. They demonstrated that diverse base classifiers are beneficial in deriving large-margin ensembles, and that therefore their proposed technique will lead to good generalization performance. They examined their method on an extensive set of benchmark classification tasks. The experimental results confirm that DRMF outperforms other classical ensemble algorithms such as Bagging, AdaBoostM1 and Rotation Forest. The success of DRMF is explained from the viewpoints of margin distribution and diversity.

D'Este et al. [35] have developed three novel voting methods are presented for combining classifiers trained on regions with available examples for predicting rare events in new regions ;specifically the closure of shellfish farms. The ensemble methods introduced are consistently more accurate at predicting closures. Approximately 63% of locations were successfully learned with Class Balance aggregation compared with 37% for the Expert guidelines, and 0% for One Class Classification.

Kourentzes et al. [36] have proposed a mode ensemble operator based on kernel density estimation, which unlike the mean operator is insensitive to outliers and deviations from normality, and unlike the median operator does not require symmetric distributions. The three operators are compared empirically and the proposed mode ensemble operator is found to produce the most accurate forecasts, followed by the median, while the mean has relatively poor performance. The findings suggested that the mode operator should be considered as an alternative to the mean and median operators in forecasting applications. Experiments indicated that mode ensembles are useful in automating neural network models across a large number of time series, overcoming issues of uncertainty associated with data sampling, the stochasticity of neural network training, and the distribution of the forecasts.

Yin et al. [37] formulated the classifier ensemble problem with the sparsity and diversity learning in a general mathematical framework, which proves beneficial for grouping classifiers. In particular, derived from the error-ambiguity decomposition, they designed a convex ensemble diversity measure. Consequently, accuracy loss, sparseness regularization, and diversity measure can be balanced and combined in a convex quadratic programming problem. They proved that the final convex optimization leads to a closed-form solution, making it very appealing for real ensemble learning problems. They compared their proposed novel method with other conventional ensemble methods such as Bagging, least squares combination, sparsity learning, and AdaBoost, extensively on a variety of UCI benchmark data sets and the Pascal Large Scale Learning Challenge 2008 web spam data. Experimental results confirmed that their approach has very promising performance.

Díez-Pastor et al. [38] have presented two new methods for tree ensemble constructions are: G-Forest and GAR-Forest. In a similar way to Random Forest, the tree construction process entails a degree of randomness. The same strategy used in the GRASP metaheuristic for generating random and adaptive solutions is used at each node of the trees. The source of diversity of the ensemble is the randomness of the solution generation method of GRASP. A further key feature of the tree construction method for GAR-Forest is a decreasing level of randomness during the process of constructing the tree: maximum randomness at the root and minimum randomness at the leaves. The method is therefore named ‘‘GAR’’, GRASP with annealed randomness. The results conclusively demonstrate that G-Forest and GAR-Forest outperform Bagging, AdaBoost, MultiBoost, Random Forest and Random Subspaces. The results are even more convincing in the presence of noise, demonstrating the robustness of the method.

OwnandAbraham [39] proposed a novel weighted rough set as a Meta classifier framework for 14 classifiers to find the smallest and optimal ensemble, which maximize the overall ensemble accuracy. they proposed a new entropy-based method to compute the weight of each classifier. Each classifier assigns a weight based on its contribution in classification accuracy. Thanks to the powerful reduct technique in rough set, which guarantee high diversity of the produced reduct ensembles. The higher diversity between the core classifiers has a positive impact on the performance of minority class as well as in the overall system performance. Experimental results with ozone dataset demonstrated the advantages of weighted rough set Meta classifier framework over the well-known Meta classifiers like bagging, boosting and random forest as well as any individual classifiers.

The use of ensemble technique

Many researchers have worked on the ensemble of multiple algorithms to improve the performance of classification or prediction in data mining or machine learning.. In our study we seek to develop a novel ensemble model for long term rainfall prediction by using Meta classifier Vote combining with three base classifiers IBK, K-star and M5P, for increasing not only the accuracy of the prediction, but also to lead to greater confidence in the results.

III. Intelligent Data Analysis: Methodologies Used

We use the following 10 methods as individual base algorithms to create the rainfall prediction models:

A. *The Base algorithms*

1) *Gaussian Process**es***

GP is based on the assumption that observations follow a normally distributed stochastic process. This leads to the conclusion, that new observations do not change the probability distribution of earlier ones. Based on this simple property Gaussian process regression allows predictions for unknown values [40]. A Gaussian process is stochastic process, any linear functional applied to the sample function Xt will give a normally distributed result. We can write:

$$f \sim GP(m,K) \quad (1)$$

That mean the random function f is distributed as a GP with mean function m and covariance function K.

2) *Linear Regression*

Is an approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables denoted X. In linear regression, data are modeled using linear predictor functions, and unknown model parameters are estimated from the data. Such models are called linear models [41]. In the case of prediction or forecasting, linear regression can be used to fit a predictive model to an observed data set of y and X values. After developing such a model, if an additional value of X is given without its accompanying value of y, the fitted model can be used to make a prediction of the value of y [42]. If we have a data set $\{y_i, x_{i1}, \cdots, x_{ip}\} \overset{n}{i} = 1$ of n statistical units, a linear regression model assumes that the relationship between the dependent variable y_i and the p-vector of regressors x_i is linear. This relationship is modeled through a disturbance term or error variable ε_i — an unobserved random variable that adds noise to the linear relationship between the dependent variable and regressors. Thus the model takes the form:

$$y = \beta_1 x_{i1} + \cdots + \beta_p x_{ip} = x_i^T \beta + \varepsilon_i \quad (2)$$

Where: $i = 1 \cdots n$,T denotes the transpose, so that $x_i^T \beta$ is the inner product between vectors x_i and β. often these n equations are stacked together and written in vector form as:

$$Y = X\beta + \varepsilon \quad (3)$$

Where:

$$Y=\begin{pmatrix}y_1\\y_2\\\vdots\\y_n\end{pmatrix},X=\begin{pmatrix}x_1^T\\x_2^T\\\vdots\\x_n^T\end{pmatrix}=\begin{pmatrix}x_{11}\cdots x_{1p}\\x_{21}\cdots x_{2p}\\\vdots\ \ddots\ \vdots\\x_{n1}\cdots\ x_{np}\end{pmatrix},\beta=\begin{pmatrix}\beta_1\\\beta_2\\\vdots\\\beta_n\end{pmatrix},\varepsilon=\begin{pmatrix}\varepsilon_1\\\varepsilon_2\\\vdots\\\varepsilon_n\end{pmatrix}$$

3) Multilayer Perceptron

Is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. A MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique called back propagation for training the network. MLP is a modification of the standard linear perceptron and can distinguish data that are not linearly separable. It has the ability to cope with the nonlinearities; the speed of computation, the learning capacity and the accuracy made them valuable tools for Time series prediction [43].

4) IBK

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” [44].

5) 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 KNearest Neighbour (K-NN) [44]. 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:

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

Where P* is the probability of all transformational paths from instance x to y.

6) Decision Table

Is precise yet compact way to model complicated logic. Decision tables, like flowcharts and if-then-else and switch-case statements, associate conditions with actions to perform. Each decision corresponds to a variable, relation or predicate whose possible values are listed among the condition alternatives. Each action is a procedure or operation to perform, and the entries specify whether (or in what order) the action is to be performed for the set of condition alternatives the entry corresponds to. Many decision tables include in their condition alternatives the don't care symbol, a hyphen. Using don't cares can simplify decision tables, especially when a given condition has little influence on the actions to be performed. In some cases, entire conditions thought to be important initially are found to be irrelevant when none of the conditions influence which actions are performed.

7) M5 Rules

M5Rules generates a decision list for regression problems using separate-and-conquer. In each iteration, it builds a model tree using M5 and makes the "best" leaf into a rule. The algorithm divides the parameter space into areas (subspaces) and builds in each of them a linear regression model. It is based on M5 algorithm. In each iteration, a M5 Tree is generated and its best rule is extracted according to a given heuristic. The algorithm terminates when all the examples are covered.

8) M5P

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 [45]. 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.

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

Where Ti corresponds to T1, T2, T3 ... sets that result from splitting the node according to the chosen attribute. The linear regression models at the leaves predict continuous numeric attributes. They are similar to piecewise linear functions and when finally they are combined a non-linear function is formed [46]. The aim is to construct a model that relates a target value of the training cases to the values of their input attributes. The quality of the model will generally be measured by the accuracy with which it predicts the target values of the unseen cases. The splitting process terminates when the standard deviation is only a small fraction less than the standard deviation of the original instance set or when a few instances remain.

In another word we can say that, the algorithm of M5P is based on decision trees, however, instead of having values at tree's nodes, it contains a multivariate linear regression model at each node. The input space is divided into cells using training data and their outcomes, and then a regression model is built in each cell as a leaf of the tree.

9) REPTree

Builds a decision/regression tree using entropy as impurity measure and prunes it. Only sorts values for numeric attributes once [47]. With the help of this method, complexity of decision tree model is decreased by “reduced error pruning method” and the error arising from variance is reduced [48]. Let Y and X be the discrete variables that have the values {y1, …,yn} and {x1, …, xn}. In this case, entropy and conditional entropy of Y are calculated as shown in equation (6) and (7). After that, information gain of X is calculated as shown in equation (8).

$$H(Y) = -\sum_{i=1}^{k} P(Y = y_i)\log P(Y = y_i) \quad (6)$$

$$H(Y \mid \mathrm{X}) = -\sum_{i=1}^{l} P(X = x_i) H(Y \mid \mathrm{X} = \mathrm{x_i}) \quad (7)$$

$$IG(Y; X) = H(Y) - H(Y \mid \mathrm{X}) \quad (8).$$

In decision trees, pruning is done in two ways. These are pre-pruning and post-pruning. If the number of instances that reach a node is lowers than the percentage of the training set, that node is not divided. It is considered that variance of the model which is generated by the training with a small number of instances and accordingly the generalization error will increase. For this reason, if the expansion of the tree is stopped when building the tree, then this is called pre-pruning. Another way of building simple tress is post-pruning. Generally, post-pruning gives better results than pre-pruning in practice [49]. Since the tree does not take steps backward and continues to expand steadily while it is being built, the variance increases. Post-pruning is a way to avoid this situation. In order to do this, firstly, unnecessary sub-trees should be found and pruned.

In post-pruning, the tree is expanded until all the leaves are pure and there is no error in training set. After that, we find the sub-trees that lead to memorizing and prune them. In order to this, we firstly use a major part of training set as growing set and the remaining part as pruning set. Later, we replace each sub-tree with a leaf that is trained by the instances which are covered by the training set of that sub-tree and then we compare these two options on pruning set. If the leaf does not lead to more errors on pruning set, we prune the sub-tree and use the leaf; otherwise we keep the sub-tree [50, 51]. When we compare and contrast pre-pruning and post-pruning, we see that pre-pruning produces faster trees; on the other hand, post-pruning produces more successful trees [49].

10) UserClassifier

Is special in that it is interactive and lets the user to construct his own decision tree classifier. For the UserClassifier it is best to have numeric attributes because they can be well represented in pixel plots. In the UserClassifier the nodes in the decision tree are not simple tests on attribute values, but are regions the user interactively selects in these plots. So if an instance lies inside the region it follows one branch of the tree, if it lays outside the region it follows the other branch. Therefore each node has only two branches going down from it [52].

B. Base Meta Classifiers Used

1) Additive Regression

Is a kind of algorithm for numerical prediction that can build standard regression model (e.g. tree) and gather residuals, learn model predicting residuals (e.g. tree), and repeat. To predict, it simply sum up individual predictions from all models and also it minimizes squared error of ensemble if base learner minimizes squared error.

Additive regression is another effective ensemble learning method, which uses a set of base learners to achieve greater predictive accuracy. Additive regression implements forward stage wise additive modeling. It starts with an empty ensemble and incorporates new members sequentially. At each stage the model that maximizes the predictive performance of the ensemble as a whole is added, without altering those already in the ensemble. The first regression model – for example, a MLP could be used – maps the input data to the outputs as usual. Then the residuals between the predicted and observed values are corrected by training a second model – e.g., another MLP. Adding the predictions made by the second model to those of the first one yields fewer errors on the training data. The methodology continues with the next model, which learns to predict the residuals of the residuals, and so on [53].

For the additive model Y has been modeled as an additive combination of arbitrary functions of the Xs, which appears in formula (9)

$$Y = A + \sum_{j=1}^{k} f_j(X_j) + \varepsilon \quad (9)$$

Where f_j represent arbitrary functions that can be estimated by lowess or smoothing splines.

Therefore, Additive regression is a form of regression gradient boosting: it enhances performance of basic regression methods [54]

2) Bagging

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 [55].Bagging is a combination of bootstrapping and averaging used to decrease the variance part of prediction errors [56].

3) MultiScheme

Is the simplest technique, as there is no second-level (or meta-) classifier and there is no explicit combination of the individual classifier predictions. MultiScheme, which is also known as Select Best, simply selects a single base-classifier as the predictor for an instance, according to which performed the best on the training data for the instance’s class [57].

It selects the best classifier from a set of candidates using cross validation of percentage accuracy (classification) or mean-squared error (regression). The number of folds is a parameter. Performance on training data can be used instead.

4) Random SubSpace

Is subspace Method–combination of random subsets of descriptors and averaging of predictions [58]. Random subspace is an ensemble classifier that consists of several classifiers and outputs the class based on the outputs of these individual classifiers. Random subspace method is a generalization of the random forest algorithm. Whereas random forests are composed of decision trees, a random subspace classifier can be composed from any underlying classifiers. Random subspace method has been used for linear classifiers, support vector machines, nearest neighbours and other types of classifiers. This method is also applicable to one-class classifiers [59].

Each classifier in the ensemble is trained on a random subset of features. The subsets can be intersecting or disjoint. The outputs are aggregated by majority vote. Like Bagging and AdaBoost, the random subspace method is only a ‘shell’ and can be used with any base classifier. Classifiers that are stable with respect to small changes in the training data may become diverse if trained on different subsets of features [60].

5) Regression by Discretization

Is a meta-learning scheme that applies to regression problems, it based on Random Forest (RD-RF). This is a regression scheme that employs a classifier (random forest, in this case) on a copy of the data which have the property/activity value discretized with equal width. The predicted value is the expected value of the mean class value for each discretized interval (based on the predicted probabilities for each interval) [61].

6) Stacking

Is a meta-classification ensemble introduced by Wolpert [62].The concept of Stacking is to use the predictions of the base-classifiers as attributes in a new training dataset that keeps the original class labels. This new training dataset is learned by a meta-classifier to get the final prediction of the ensemble. Stacking can be viewed as a generalization of Voting [63]

Stacking or stacked generalization is a general method of using the combination of the output from several models in order to achieve a greater predictive accuracy. The final output of the ensemble can be calculated using:

$$y_t = \sum_{K=1}^{N} C_K \hat{Z}_{k,t} + e_t \quad (10)$$

Where $\hat{Z}_{k,t}$ is output from model K for observation t and the coefficients C_K are estimated in order to construct the final output of the ensemble by minimizing the function G . The function G expressed as:

$$G = \sum_{t=1}^{n} \left[z_t - \sum_{K=1}^{N} C_K \hat{Z}_{k,t} \right]^2 \quad (11)$$

With using constrain $\sum_{K=1}^{N} C_K = 1$ and $0 \le C_K \le 1$.In [64] Breiman suggested minimizing the function G that can give better generalization for the model.

In stacking, the result of a set of different base learners at the level-0 is combined by a Meta learner at the level-1. The role of the Meta learner is to discover how best to combine the output of the base learners [65].

7) Vote

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 [57].

C. Ensemble methodology

1) Combination methods

There are two main methods for combining the base-classifiers’ outputs [7]: 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 paper we used vote Meta-combination method to combine the base classifiers.

2) Structure of ensemble classifiers

There are two types for structuring the classifiers of ensembles [8] parallel and Cascading or Hierarchical structure. In this paper 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). Figure.1 shows the structure of the proposed ensemble classifiers.

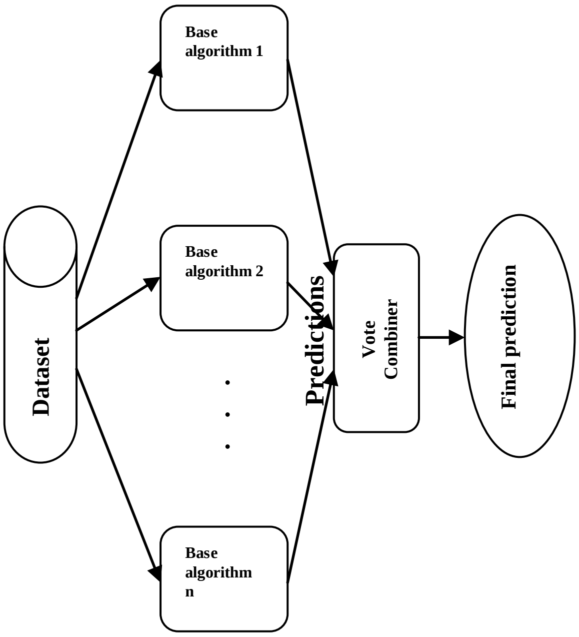

Figure 1. The overview of ensemble classifiers framework.

As shown in Figure 1, 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.

3) Classifiers Combination strategy

Combining rules are the simplest combination approach and it is probably the most commonly used in the multiple classifier system [66]. 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 [67].

A theoretical framework for fixed rules combination was proposed by Josef Kittler et al. [68] 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 paper we have adopted the median probabilities as combination rule method because, it gives the best results for our dataset.

Median Rule

Equation (12) can be used to compute the average a posteriori probability for each prediction over all the classifier outputs, i.e.

assign $Z \rightarrow w_j$ if

$$\frac{1}{R}\sum_{i=1}^{R} P(w_j \mid x_i) = \max_{k=1}^{m} \frac{1}{R}\sum_{i=1}^{R} P(w_k \mid x_i) \quad (12)$$

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:

assign $Z \rightarrow w_j$ if

$$\underset{i=1}{\overset{R}{med}}\, P(w_j \mid x_i) = \max_{k=1}^{m} \underset{i=1}{\overset{R}{med}}\, P(w_k \mid x_i) \quad (13)$$

D. Rainfall dataset

The meteorological data that used in this paper produced by meteorological authority, Sudan and has been brought from Central Bureau of Statistics, Sudan for 13 years from 2000 to 2012 for 24 meteorological stations over the country. These stations are: (Khartoum, Dongola, Atbara, AbuHamad, Karima, WadiHalfa, Wad Medani, El Deweim, Kassala, Port Sudan, El Gadarif, Elobied, El Nihood, Kadugli, Nyala, Elgeneina, El Fashir, Kosti, El damazen, New Halfa, Babanusa, Rashad, Abu Naam, Sinnar). The dataset had eight (8) attributes containing monthly averages data. In this paper we used only the more important 4 attributes (Date, Minimum Temperature, Humidity and Wind Direction) [5] that affect the prediction of rainfall. Figures (2), (3), and (4) show the numerical attributes in dataset minimum temperature, relative humidity and rainfall respectively.

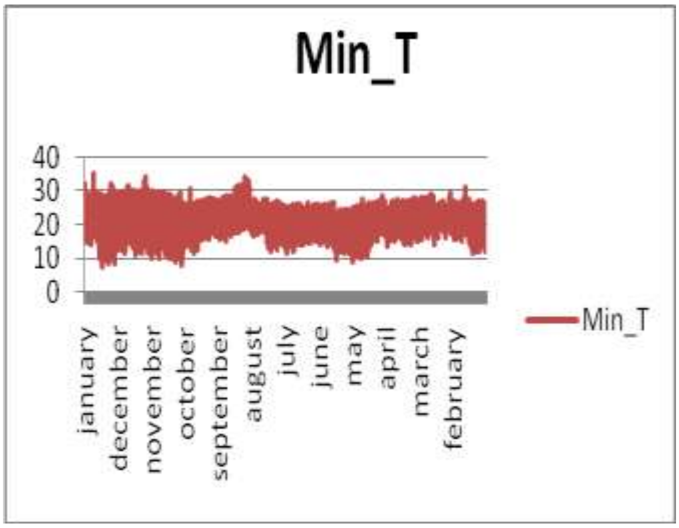

Figure 2. Minimum temperature.

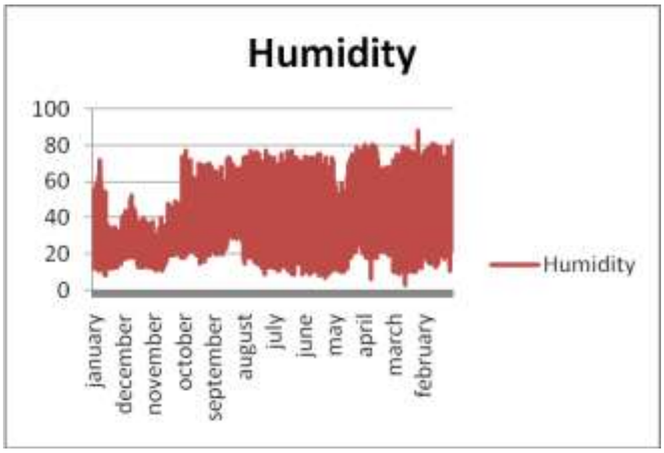

Figure 3. Relative humidity.

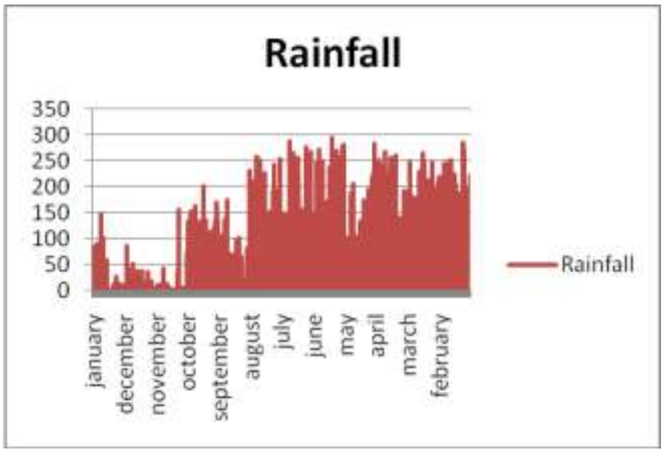

Figure 4. Rainfall.

E. Test option

The result of applying the chosen classifier will be tested according to a certain test option [69], there are several test modes such as Use training set, Supplied test set, Cross-validation and Percentage split. In this paper we use supplied test set as the test option. In this method the classifier is evaluated on how well it predicts the class of a set of instances loaded from a file.Accordance with that our rainfall dataset, which contains 3732 instanceshas been divided into two parts with ratio of 70 to 30 for training and testing respectively. The first part contained 2612 examples for training models and the other one contained 1120 examples for testing models.

F. Evaluation

For evaluating the models performance and comparing between them the following performance metrics have been used:

1) Correlation Coefficient (CC):

This measures the statistical correlation between the predicted and actual values. This method is unique in that it does not change with a scale in values for the test cases [70]. Karl Pearson's correlation coefficient formula is used and it is shown in equation (14).

$$R_{x,y} = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\ \left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}\ \sqrt{\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2}} \quad (14)$$

A higher number means a better model, with a 1 meaning a perfect statistical correlation and a 0 meaning there is no correlation at all.

2) Mean Absolute Error (MAE):

Mean-absolute error is one of the most commonly used measures of success for numeric prediction. This value is computed by taking the average of the differences between each computed value (predicted) and its corresponding correct value (actual) [71]. MAE calculations are shown below in equation (15).

$$MAE = \frac{|a_1 - c_1| + \cdots + |a_n - c_n|}{n} \quad (15)$$

Assuming that the actual output is a, expected output is c.
To make regression more robust Minimize absolute error, not squared error.

3) Root mean-squared Error (RMSE):

The Root mean-squared Error is simply the square root of the mean-squared-error. The mean-squared error gives the error value the same dimensionality as the actual and predicted values. Error rate of an estimator arises just because of an arbitrary estimation or lack of information that may provide an accurate estimation [72].RMSE formula is shown in equation (16).

$$RMSE = \sqrt{\frac{(a_1 - c_1)^2 + \cdots + (a_n - c_n)^2}{n}} \quad (16)$$

If the values of MAE and RMSE rates are closer to zero, the error rates will be lower. In addition, acceptable error values for MSE and RMSE are different for each learning problem.

4) Time

in our study we use the both time taken to build the model and time taken to test model on supplied test set to compare and differentiate among the models.

IV. Experimental Result

Table 1. shows the performance of the base classifier models according to correlation coefficient, mean absolute error, root mean squared error, while Table 2. displays time taken to build model and time taken to test model on supplied test set for the base algorithm..

Base algorithm	**CC**	**MAE**	**RMSE**
Gaussian Processes (GP)	0.8656	0.1638	0.2512
Linear Regression (LR)	0.8642	0.1643	0.2527
Multilayer Perceptron (MP)	0.8594	0.1327	0.2654
IBK	0.8192	0.0905	0.3005
KStar	0.8901	0.1091	0.2285
Decision Table (DT)	0.8351	0.1219	0.2775
M5Rules(M5R)	0.8642	0.1113	0.2529
M5P	0.8863	0.1047	0.2322
REPTree(RT)	0.8262	0.1286	0.2841
User Classifier (UC)	0.8801	0.2352	0.32

Table 1. Performance of the base algorithms.

Base algorithm	**Training Time (sec)**	**Testing Time (sec)**
Gaussian Processes (GP)	87.2	1.97
Linear Regression (LR)	0.1	0.01
Multilayer Perceptron (MP)	27.04	0.02
IBK	0.02	0.34
KStar	0.01	4.59
Decision Table (DT)	0.1	0.02
M5Rules(M5R)	0.3	0.02
M5P	0.1	0.01
REPTree(RT)	0.5	0.01
User Classifier (UC)	0.6	0.01

Table 2. Base algorithms training and testing time.

According to the Experimental results that appear in the Table 1, we find that KStar algorithm has the maximum correlation coefficient 0.8901, the minimum root mean squared error 0.2285 and the third lower mean absolute error 0.1091. M5P algorithm comes in second place after KStar as the second highest correlation coefficient 0.8863; the second less mean absolute error 0.1047 and second less root mean squared error 0.2322. User Classifier algorithm comes in third place in terms of the standard correlation coefficient 0.8801, but it's the worst on both levels of mean absolute error 0.2352 and root mean squared error 0.32.IBK algorithm has the minimum mean absolute error 0.0905, but at the same time it has the second biggest root mean squared error 0.3005 and unsatisfactory correlation coefficient 0.8192 compared with the other base algorithms.

Figure 5 compared between the base algorithms in terms of correlation coefficient, mean absolute error and root mean squared error. We can observe that the most accurate base algorithm in the term of correlation coefficient and root mean squared error is Kstar, while IBK algorithm has the lowest mean absolute error.

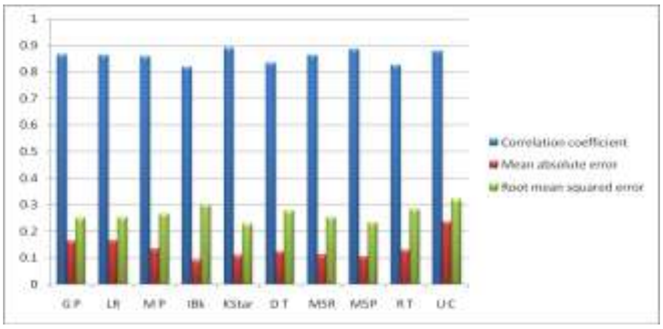

Figure 5. Performance comparison between the base algorithms.

Figure 6 compared between the base algorithms in terms of time taken to build model and time taken to test model on supplied test set. We can easily find that the longest time to build a model is 87.2seconds belong to Gaussian Processes (GP), while the shortest time are 0.01, 0.02 second belong to Kstar and IBK respectively. If we look to the time taken to test model on supplied test set we conclude that the longest test time 4.59 belong to Kstar and the shortest test time 0.01 belong to Linear Regression, M5P, REPTree and User Classifier.

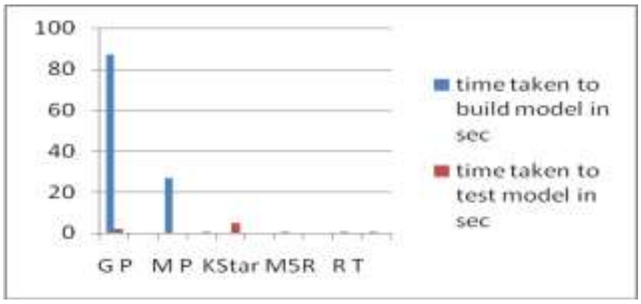

Figure 6. Comparison between the base algorithms for training and testing.

Table 3. Shows the performance of the individual Meta classifiers according to correlation coefficient, mean absolute error, root mean squared error. The best correlation coefficient 0.8651 belongs to Random SubSpace Meta classifier, bagging Comes second has 0.8529 correlation coefficient and vote Meta classifier in third place has 0.8463 correlation coefficient. The worst correlation coefficient 0 resulted from both multi scheme and staking Meta methods. In terms of the Mean absolute error Bagging and vote have the lowest 0.1237 and 0.1287 respectively, while multi scheme and staking have the highest 0.4783 and 0.504 respectively. Most of the tested Meta methods their Root mean squared error in the range of 0.2563-0.2965 except multi scheme and staking these have 0.4892 and 0.5273 respectively.

Meta Classifier	CC	MAE	RMSE
Additive Regression	0.8052	0.196	0.2965
Bagging	0.8529	0.1237	0.2614
multi scheme	0	0.4783	0.4892
Random SubSpace	0.8651	0.1636	0.2563
Regression by Discretization	0.8154	0.1397	0.2914
staking	0	0.504	0.5273
vote	0.8463	0.1287	0.267

Table 3. Performance of the individual Meta algorithms.

Table 4. Shows time taken to build model and time taken to test model on supplied test set for the individual Meta method.

Meta Classifier	Training Time (sec)	Testing Time (sec)
Additive Regression	0.08	0.01
Bagging	0.16	0.01
multi scheme	0.03	0.01
Random SubSpace	0.11	0.02
Regression by Discretization	0.15	0.03
staking	0.02	0.01
vote	0.01	0.03

Table 4. Individual Meta algorithms training and testing time.

Figure 7 compared between the Meta algorithms in terms of correlation coefficient, mean absolute error and root mean squared error. We can infer that Random Sub-Space is the best Meta method in both Correlation coefficient 0.8651and Root mean squared error 0.2563, while Bagging has the lowest Mean absolute error 0.1237. On the other hand multi scheme and staking Meta models deal very bad with our data and give the worst results 0 Correlation coefficient, and the highest Mean absolute error and Root mean squared error comparing with other proposed Meta methods.

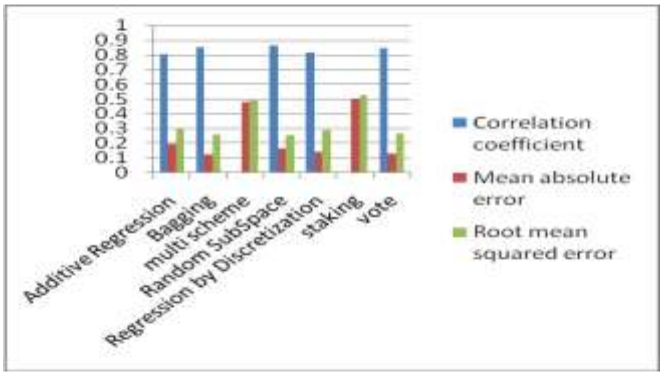

Figure 7. Performance Comparison between the Meta algorithms.

Figure8. Compared between the Meta algorithms in terms of time taken to build model and time taken to test model on supplied test set. We can note that the longest time to build individual meta model is 0.16seconds belong to Bagging, while the shortest time are 0.01second belong to Vote. If we look to the time taken to test model on supplied test set we conclude that the longest test time 0.03belong to both Vote and Regression by Discretization, while the shortest test time 0.01 belong to Additive Regression, Bagging and multi scheme.

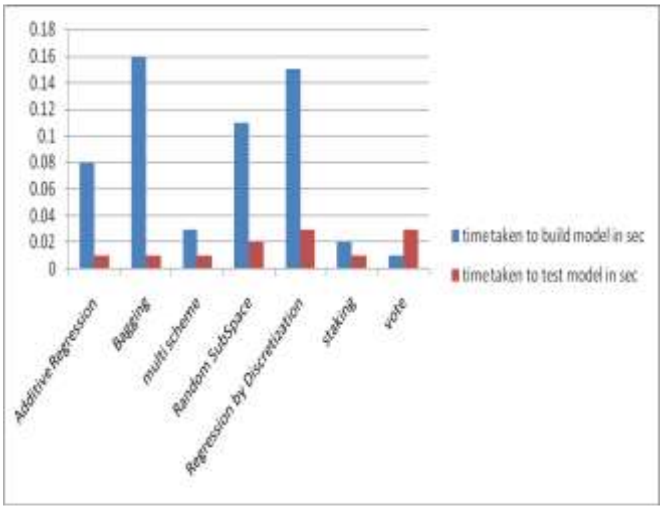

Figure 8. Comparison between the Meta algorithms for training and testing times.

Table 5. shows the performance of the Ensemble models s according to correlation coefficient, mean absolute error, root mean squared error. We construct ensemble model of Meta Vote method combining with various base classifiers.Vote+2 algorithms (Kstar and linear regression), Vote+3 algorithms (IBK, Kstar and M5P), Vote+4 algorithms (IBk, Kstar, M5P, and linear regression), Vote+5 algorithms (IBk, Kstar, M5P, REPTree and User Classifier), Vote+6 algorithms (Linear

Regression, IBk, Kstar, M5P, REPTree and User Classifier), Vote+7 algorithms (Linear Regression, IBk, Kstar, M5Rules, M5P, REPTree and User Classifier), Vote+8 algorithms (Linear Regression, IBk, Kstar, Decision Table, M5Rules, M5P, REPTree and User Classifier),Vote+9 algorithms (Linear Regression, Multilayer Perceptron, IBk, Kstar, Decision Table, M5Rules, M5P, REPTree and User Classifier).

Meta method	CC	MAE	RMSE
Vote+2 algorithms	0.8861	0.1311	0.2319
Vote+3 algorithms	0.8986	0.0888	0.1092
Vote+4 algorithms	0.8803	0.1376	0.2728
Vote+5 algorithms	0.884	0.1328	0.2379
Vote+6 algorithms	0.8753	0.1235	0.2418
Vote+7 algorithms	0.8852	0.1378	0.2375
Vote+8 algorithms	0.8835	0.1327	0.2383
Vote+9 algorithms	0.8832	0.1369	0.2386

Table 5. Performance of the Ensemble models

Table 6. Shows time taken to build model and time taken to test model on supplied test set for ensemble methods

Meta method	Training Time (sec)	Testing Time (sec)
Vote+2 algorithms	0.08	4.44
Vote+3 algorithms	0.09	4.71
Vote+4 algorithms	0.78	4.77
Vote+5 algorithms	3.95	4.83
Vote+6 algorithms	8.41	4.87
Vote+7 algorithms	34.81	4.54
Vote+8 algorithms	35.17	4.81
Vote+9 algorithms	67.42	4.34

Table 6. Ensemble methods training and testing time.

According to experimental results, which has showed in Table 3, we can conclude that the proposed ensemble method achieved the best performance overall other ensemble methods. Ensemble Vote+3 algorithm has the highest correlation coefficient0.8986, the lowest of both mean absolute error and root mean squared error0.0888 and 0.1092 respectively. Ensemble Vote+2 algorithms come second in terms of both correlation coefficient0.8861and root mean squared error0.2319but it has 0.1311mean absolute error. From other point of view Ensemble Vote+6 algorithms comes second in terms of mean absolute error 0.1235 however it has correlation coefficient0.8753 and root mean squared error0.2418.

Figure 9 compared between the Ensemble models in terms of correlation coefficient, mean absolute error and root mean squared error. As shown in Figure 9, we find that the results of the proposed ensemble methods performance are close, but also we notice ensemble Vote+3outperformed the rest of the other ensemble methods, Because it had the highest correlation coefficient and the lowest of both mean absolute error and root mean squared error. The minimum correlation coefficient (0.8803) resulting from the ensemble Vote+4 algorithms, also it has the second highest mean absolute error (0.1376) after the worst one, ensemble Vote+7 algorithms, which has (0.1378). Ensemble Vote+4 algorithms has the biggest root mean squared error (0.2728).

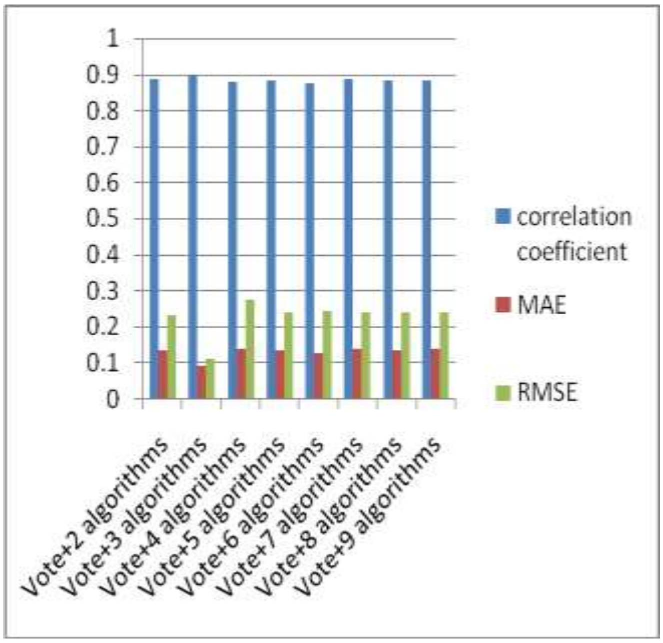

Figure 9. Performance comparison between the Ensemble models.

Figure 10 compared between the Ensemble models in term of time taken to build model and time taken to test model on supplied test set. We can deduce the longest time to build ensemble model is 67.42seconds belong to ensemble Vote+9 algorithms, while the shortest time is0.08 second belong to ensemble Vote+2 algorithms. If we look to the time taken to test model on supplied test set we conclude that the longest test time 4.87belong to ensemble Vote+6 algorithms, while the shortest test time 4.34belong to ensemble Vote+9 algorithms.

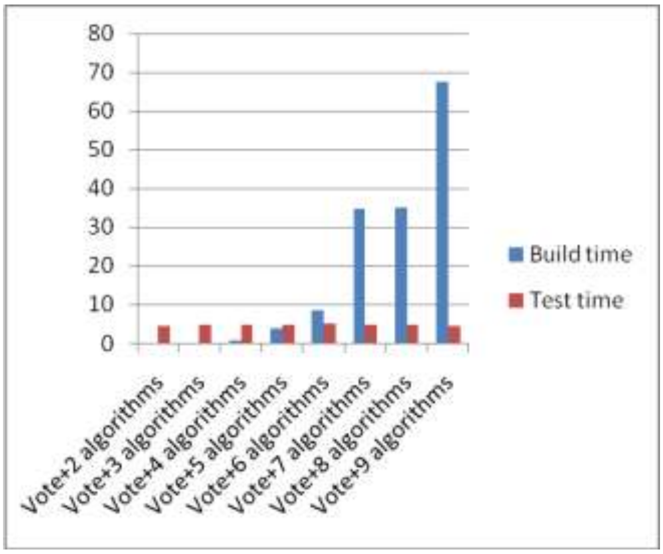

Figure 10. Comparison between the Ensemble models for training and testing time.

V. Discussion

According to the results obtained from the experiments we can argue that the proposed ensemble method of the base algorithms IBK, Kstar and M5P combining by Meta vote method has achieved the best performance comparing with the 10 bases algorithms, seven Meta methods and the other ensemble combination options. The proposed ensemble (Vote+3 algorithms) has the highest Accuracy in terms of correlation coefficient; mean absolute error and root mean squared error in comparison with the rest. Also it required0.09 second as a time for building the model and 4.71senconds as time taken to test model on supplied test set. The minimum time for building model has been achieved by both Kstar base algorithm and individual Meta vote algorithm. On the other side the maximum time for building a model has been achieved by Gaussian Processes base algorithm. While The minimum time for testing model on supplied test set has been achieved by both the base algorithms: (Linear Regression, M5P, REPTree and user classifier) and individual meta methods: (Additive Regression, Bagging, multi scheme and staking). However the ensemble Vote+6 algorithms have achieved the maximum time for testing model on supplied test set.

Both individual Meta methods (multi scheme and staking) have the lowest correlation coefficient (0) comparing with all other models, and that means there is no correlation at all between the predicted and actual values. At the same context the both individual Meta methods (multi scheme and staking) have the biggest two mean absolute errors. The worst is stacking then multi scheme Meta method. Also if we take the root mean squared error as a measure performance we find the both individual Meta methods (multi scheme and staking) have the highest root mean squared errors. Based on the previous results we can decide that the worst algorithm in our study is staking Meta method followed directly by multi scheme Meta method.

VI. Conclusions

This paper proposes a novel method to develop long-term weather forecasting model for rainfall prediction by using ensemble technique. Monthly meteorological data from 2000 to 2012, for 24 meteorological stations in Sudan has been used. The dataset contained six predictors for rainfall (date, minimum temperature relative humidity and wind direction). In our intensive experiments we developed group of base algorithm models (Gaussian Processes, Linear Regression, Multilayer Perceptron, IBk, KStar, Decision Table, M5Rules, M5P, REP Tree and User Classifier.), individual Meta algorithm models (Additive Regression, Bagging, Multi Scheme, Random Subset, Regression by Discretization, Stacking, and Vote) and ensemble models. The new novel ensemble method has been constructed based of vote Meta classifier combining with three base classifiers IBK, K-star and M5P. The models have been evaluated by using correlation coefficient; mean absolute error and root mean-squared error as performance metrics**.** Also we use the both time taken to build the model and time taken to test model on supplied test set to compare and differentiate among the models. Our empirical results showed that the new novel ensemble method has the best performance comparing to both basic and Meta algorithms. Our novel ensemble model increases not only the accuracy of the prediction, but also leads to greater confidence in the results.

References

[1] N. Bushara and A. Abraham, “Computational Intelligence in Weather Forecasting: A Review”, *Journal of Network and Innovative Computing*, 1(1), pp. 320-331, 2013.

[2] P. Jagannathan, “Long Range Weather Forecasting”, *Forecasting Manual Part IV - Comprehensive Article s on Selected Topics*, India Meteorological Department, 14 December 1974.

[3] O. Maimonand L. Rokach, *Data Mining and Knowledge Discovery Handbook*, Second Edition, Springer New York Dordrecht Heidelberg London, 2010.

[4] A. Pujari, *Data Mining Techniques*, third edition, Universities Press, 2013.

[5] N. Bushara and A. Abraham, “Weather Forecasting in Sudan Using Machine Learning Schemes”, *Journal of Network and Innovative Computing*, 2(1), pp. 309-317, 2014.

[6] T. Mitchell, “Generalization as search”, *Artificial Intelligence*, 18(2), pp. 203–226, 1982.

[7] L. Rokach, “Ensemble-based classifiers”, *Artificial Intelligence Review*, 33, pp. 1–39, 2010.

[8] G. Seni and J. Elder, *Ensemble Methods in Data Mining: Improving Accuracy through Combining Predictions*, Morgan & Claypool Publishers, www.morganclaypool.com, 2010.

[9] K.Tumer and J. Ghosh, “Error correlation and error reduction in ensemble classifiers”, Connection science, special issue on combining *artificial neural networks: ensemble approaches*, 8(3,4), pp.385–404, 1996.

[10] A. Krogh and J. Vedelsby, “Neural network ensembles, cross validation and active learning”, *Advances in Neural Information Processing System*, 7, pp. 231–238, 1995.

[11] L. Kuncheva, *Combining pattern classifiers*, Wiley Press, New York, 2005.

[12] L. Kuncheva and C. Whitaker, “Measures of diversity in classifier ensembles and their relationship with ensemble accuracy”, *Machine Learning*, 51(2), pp. 181–207, 2003.

[13] X. Hu, “Using rough sets theory and database operations to construct a good ensemble of classifiers for data mining applications”, *In Proceeding IEEE International Conference on Data Mining*, pp 233–240, 2001.

[14] G. Brown,J. Wyatt, R. Harris and X. Yao, “Diversity creation methods: a survey and categorization”, *Information Fusion*, 6(1), pp. 5–20, 2005.

[15] R. Banfield, L. Hall, K. Bowyer and W. Kegelmeyer, “A comparison of decision tree ensemble creation techniques”, IEEE Transactions on Pattern Analysis and Machine Intelligence 29(1), pp. 173–180, 2007.

[16] John. W. Tukey, “Exploratory Dta Analysis”, *Biometrical Journal*, 23(4), pp. 413–414, 1981.

[17] L. Hansen and P. Salamon, “Neural Network Ensembles”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), pp. 993-1001, 1990.

[18] R. Schapire, “The strength of weak learnability”. *Machine Learning*, 5(2), pp.197–227, 1990.

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

[20] S. Merler, B. Caprile and C. Furlanello, “Parallelizing AdaBoost by weights dynamics”, *Computational statistics and data analysis*, 51, pp. 2487–2498, 2007.

[21] C. Zhang and J. Zhang, “A local boosting algorithm for solving classification problems”, *Computational Statistics and Data Analysis*, 52(4), pp. 1928–1941, 2008.

[22] M. Alhamdoosh and D. Wang, “Fast decorrelated neural network ensembles with random weights”, *Information Sciences*, 264, pp. 104–117, 2014.

[23] J. DeWeber and T. Wagner, “A regional neural network ensemble for predicting mean daily river water temperature”, *Journal of Hydrology*, 517, pp.187–200, 2014.

[24] L. Li , Q. Hu , X. Wu and D.Y u, “Exploration of classification confidence in ensemble learning”, *Pattern Recognition*, 47, pp. 3120–3131, 2014.

[25] L. Zhang and P. Suganthan, “Random Forests with ensemble of feature spaces”, *Pattern Recognition*, 47, pp. 3429–3437, 2014.

[26] A. Salih and A. Abraham, “Novel Ensemble Decision Support and Health Care Monitoring System*”, Journal of Network and Innovative Computing*, 2, pp. 41-51, 2014.

[27] L. Li, R. Stolkin, L. Jiao, F. Liu and S. Wang, “A compressed sensing approach for efficient ensemble learning”, *Pattern Recognition*, 47, pp. 3451–3465, 2014.

[28] W. Chen, L. Tseng and C. Wu, “A unified evolutionary training scheme for single and ensemble of feed forward neural network”, *Neurocomputing*, http://dx.doi.org/10.1016/j.neucom.2014.05.057i, 2014.

[29] L. Rokach, A. Schclar and E. Itach, “Ensemble methods for multi-label classification”, *Expert Systems with Applications*, 41(16), pp. 7507–7523, 2014.

[30] A. Canuto, M. Abreu, M. Oliveira, J. Xavier and A. Santos,” Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles”, *Pattern recognition letters*, 28(4), pp. 472-486, 2007.

[31] V. Canedo, N. Maroño and A. Betanzos, “Data classification using an ensemble of filters”, *Neurocomputing*, 135, pp. 13–20, 2014.

[32] M. Jin, R. Li, Z. Xu and X. Zhao, “Reliable fault diagnosis method using ensemble fuzzy ARTMAP based on improved Bayesian belief method”, *Neurocomputing*, 133, pp. 309–316, 2014.

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

[34] Q. Hu, L. Li, X. Wu, G. Schaefer and D. Yu, “Exploiting diversity for optimizing margin distribution in ensemble learning”, *Knowledge-Based Systems*, 67, pp. 90-104, 2014.

[35] C. D'Este, G. Timms, A. Turnbull and A. Rahman, “Ensemble aggregation methods for relocating models of rare events”, *Engineering Applications of Artificial Intelligence*, 34, pp. 58–65, 2014.

[36] N. Kourentzes, D. Barrow and S. Crone, “Neural network ensemble operators for time series forecasting”, *Expert Systems with Applications*, 41, pp. 4235–4244, 2014.

[37] X. Yin, K. Huang, C. Yang and H. Hao, “Convex ensemble learning with sparsity and diversity”, *Information Fusion*, 20, pp. 49–59, 2014.

[38] J. Díez-Pastor, C. García-Osorio and J. Rodríguez, “Tree ensemble construction using a GRASP-based heuristic and annealed randomness”, *Information Fusion*, 20, pp. 189–202, 2014.

[39] H. Own and A. Abraham, “A Novel-weighted Rough Set-based Meta Learning for Ozone Day Prediction”, *ActapolytechnicaHungarica*, 11(4), 2014.

[40] *C.* Rodriguez, J. Pucheta, H. Patino, J. Baumgartner, S. Laboret and V. Sauchelli, “Analysis of a Gaussian process and feed-forward neural networks based filter for forecasting short rainfall time series”, *The International Joint Conference on Neural Networks (IJCNN)*, pp. 1 – 6, 2013.

[41] M.Rahman, M.Rafiuddin and M.Alam, “Seasonal forecasting of Bangladesh summer monsoon rainfall using simple multiple regression model”, *Journal of Earth System Science* 122(2), pp. 551–558, 2013.

[42] C. Udomboso and G. Amahia, “Comparative Analysis of Rainfall Prediction Using Statistical Neural Network and Classical Linear Regression Model”, *Journal of Modern Mathematics and Statistics*, 5 (3), pp. 66-70, 2011.

[43] R. Deshpande, “On The Rainfall Time Series Prediction Using Multilayer Perceptron Artificial Neural Network”, *International Journal of Emerging Technology and Advanced Engineering*, 2(1), pp. 148-153, 2012.

[44] S. Vijayarani and M. Muthulakshmi, “Comparative Analysis of Bayes and Lazy Classification Algorithms”, *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2 (8), pp. 3118-3124, 2013.

[45] W. Average, S. Kotsiantis, A. Kostoulas, S. Lykoudis, A. Argiriou and K. Menagias, “Using Data Mining Techniques for Estimating Minimum, Maximum and Average Daily Temperature Values”, International *Journal of Computer, Information, Systems and Control Engineering*, 1(2), pp. 382-386, 2007.

[46] B. Bhattacharya and D. P. Solomatine, “Neural networks and M5P model trees in modeling water level- discharge relationship”, *Neurocomputing* , 63, pp. 381-396, 2005.

[47] K. Wisaeng, “A Comparison of Decision Tree Algorithms For UCI Repository Classification”, *International Journal of Engineering Trends and Technology*, 4(8), pp. 3393-3397, 2013.

[48] I. H. Witten and E. Frank. *Data Mining Practical Machine Learning Tools and Techniques*, 2nd Edition, Elsevier Inc., 2005.

[49] E. Alpaydın, *Introduction to Machine Learning*, The MIT Press, 2004.

[50] J. R. Quinlan, "Simplifying decision trees", *International Journal of Man-Machine Studies*, 27 (3), pp. 221 – 234, 1987.

[51] M. Zontul, F. Aydin, G. Dogan, S. Sener and O. Kaynar, "Wind Speed Forecasting Using RepTree and Bagging Methods in Kirklareli-Turkey", *Journal of Theoretical and Applied Information Technology*, 56 (1), pp. 17-29, 2013.

[52] S. Peyvandi, H. M.Shirazi and A. Faraahi, "Proposing a Classification Algorithm for User Identification According To User Web Log Analysis", *Australian Journal of Basic and Applied Sciences*, 5(9), pp. 645-652, 2011.

[53] G. Wang, J. Hao, J. Ma, H. Jiang,"A comparative assessment of ensemble learning for credit scoring", *Expert Systems with Applications*, 38(1), pp. 223-230, 2011.

[54] J. Friedman, "Stochastic Gradient Boosting", *Computational Statistics and Data Analysis*, 38, pp. 367-378, 1999.

[55] D. Jeong and Y. Kim, "Rainfall-runoff models using artificial neural networks for ensemble streamflow prediction", *Hydrological Processes,* 19(19), pp. 3819–3835, 2005.

[56] L. Breiman, "Bagging predictors*", Machine Learning*, 24(2):123-140, 1996.

[57] I. Witten, E. Frank and M. Hall, Data Mining Practical *Machine Learning Tools and Techniques*, third edition, Elsevier, 2011.

[58] T. Ho,"The Random Subspace Method for Constructing Decision Forests", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), pp. 832-844, 1998.

[59] M. Skurichina and R. Duin, "Bagging, Boosting and the Random Subspace Method for Linear Classifiers", *Pattern Analysis & Application* , 5, pp. 121-135, 2002.

[60] C. Lai, M. Reinders and L. Wessels, "Random subspace method for multivariate feature selection", *Pattern recognition letters*, 27(10), pp. 1067–1076, 2006.

[61] S. M. Mwagha, M. Muthoni and P. Ochieg, "Comparison of Nearest Neighbor (ibk), Regression by Discretization and Isotonic Regression Classification Algorithms for Precipitation Classes Prediction", *International Journal of Computer Applications*, 96 (21), pp. 44-48,2014.

[62] D. H. Wolpert, "Stacked generalization", *Neural Networks*, 5 (2), pp. 241-260, 1992.

[63] K. Seewald, & J. Furnkranz, "An evaluation of grading classifiers", *4th International Conference on Advances in Intelligence Data Analysis*, pp. 115-124, 2001.

[64] L. Breiman, "Stacked regression," *Machine Learning*, (24), pp. 49-64, 1996.

[65] M. Graczyk, T. Lasota, B. Trawiński, and K. Trawiński, "Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal.", *Intelligent Information and Database Systems*, pp. 340-350, Springer Berlin Heidelberg, 2010.

[66] L. Felföldi, "Classifier Combination Systems and their Application in Human Language Technology," *A dissertation submitted for the degree of doctor of philosophy of the University of Szeged, Hungary*, 2008.

[67] L. Kuncheva, "Fuzzy vs Non-fuzzy in combining classifiers designed by boosting " *IEEE Transaction on Fuzzy Systems*, 11(6), pp. 729- 741 , 2003.

[68] J. Kittler, M. Hatef, R. Duin and J. Matas, "On combining classifiers," *IEEE Trans Pattern Analysis and Machine Intelligence*, 20 (3), pp. 226–239, 1998.

[69] R. Kirkby, E. Frankand P. Reutemann, *WEKA Explorer User Guide for Version 3-5-8*, University of Waikato, pp. 1-22, 2008.

[70] M. Zontul, F. Aydin, G. Dogan, S. Sener and O. Kaynar, "Wind Speed Forecasting Using RepTree and Bagging Methods in Kirklareli-Turkey*", Journal of Theoretical and Applied Information Technology*, 56 (1), pp. 17-29, 2013.

[71] O. Folorunsho, "Comparative Study of Different Data Mining Techniques Performance in knowledge Discovery from Medical Database", *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(3), pp. 11-15, 2013.

[72] E.L. Lehmann and G. Casella, *Theory of Point Estimation*, Second Edition, Springer: New York, 1998.