J. Appl. Environ. Biol. Sci., 7(2S)20-29, 2017

ISSN: 2090-4274
Journal of Applied Environmental and Biological Sciences
www.textroad.com

Enhancing Malaysia Rainfall Prediction Using Classification Techniques

Nor SamsiahSani[1], Israa Shlash[1], Mohammed Hassan[1], Abdul Hadi[1], Mohd Aliff[2]

[1]Center For Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia
[2]Instrumentation and Control Engineering, Malaysian Institute of Industrial Technology, Universiti Kuala Lumpur, Pasir Gudang, Johor, Malaysia

Received: February 21, 2017
Accepted: April30,2017

ABSTRACT

Data mining is a process that aims to extract useful knowledge from cluttered and unorganized information. Climate change is a discipline involved with analyzing the varying distribution of weather for a specific period of time. Specifically, rainfall forecasting analyzes specific features such as humidity and wind are used to predict rainfall in specific locations. Rainfall prediction has of recent been subjected to several machine learning techniques with different degree of short-term (daily) and long-term (monthly) prediction performance. Selecting an appropriate technique for specific rainfall duration is a challenging task. Several approaches have been proposed for rainfall forecasting using various machine techniques. This study aims to provide a comparative analysis of the multiple machine learning classifiers for rainfall prediction based on Malaysian data. Several classifiers were explored which are f Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), Neural Network (NN) and Random Forest (RF). The analysis showed the most effective classifier to be the NN.

KEYWORDS: Rainfall Prediction, Machine Learning.

INTRODUCTION

Weather forecasting is a task which combines science and technology to predict the state of the atmosphere for a future time and a given location [23, 26]. Human has long 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, and rainfall prediction is a key tool [3] for human survivability.

Time-series data have been used in many domains including finance, brain-activity, speech pattern, stock markets and weather forecasting [6]. Rainfall forecasting is a form of time-series data which has caught many researchers' attentions due to its interesting challenges and complexities, especially in predicting specific factors associated with rainfall such as wind, humidity and temperature [28].

Several machine learning approaches were proposed in relation to rainfall forecasting, applied to locations in Korea, China and South Africa [2, 22, 29]. The machine learning techniques utilized for rainfall prediction include Neural Network [12], K Nearest Neighbour, Naive Bayes [11] and Support Vector Machine [15]. Thus, we aim to investigate the various techniques in order to identify the best performing learning technique for rainfall prediction. In addition, there is a need to incorporate machine learning methods to Malaysia setting which has rather extreme variability in rainfall occurrences.

There are several research efforts that have addressed rainfall prediction problem. Such efforts have used many prediction techniques and features of multiple pre-processing approaches. For instance, in [28] proposed a new pre-processing approach using moving average and singular spectrum analysis through machine learning. Such pre-processing task employed the classes of the training data to transform them into low, medium and high probability classes. Artificial Neural Network (ANN) was carried out in order to predict the classes on unseen portion of data (testing). The study used two daily mean rainfall series datasets from Zhenshui and Da'ninghe watersheds of China.

In [2] proposed a multi-layered artificial Neural Network with back-propagation algorithm configuration using data from www.Indiastat.com and the IMD website. The input parameters were the average humidity and the average wind speed for the 8 months of 50 years from 1960-2010 which produced the average rainfall in 8 months of every year from 1960-2010.

In [15] proposed a hybrid method of feature extraction and prediction technique for predicting daily rainfall data that has been collected from National Oceanic and Atmospheric Administration (NOAA) for more than 50 years. Basically, the features that have been utilized consist of humidity, pressure, temperature and wind speed.

Corresponding Author: Nor Samsiah Sani,Faculty of Information Science and Technology, National University of Malaysia, Bangi, Selangor, Malaysia, E-mail: norsamsiahsani@ukm.edu.my

Neural Network was used to classify the instances into low, medium and high classes based on a predefined training set.

Nikam and Meshram (2013) also proposed a Bayesian algorithm for rainfall prediction in India using historical data that are collected from Indian Metrological Department. The authors utilized 6 features including temperature, pressure level, mean sea level, relatively humidity, vapour pressure and wind speed. A Bayesian algorithm then trained the data based on the mentioned features. The model is observed to be more accurate with large training dataset.

Tukey (1989) suggested combining two linear regression models. In [21] further suggested an ensemble of similarly configured neural networks to improve the predictive performance of a single one. At the same time, in [25] laid the foundations for the award winning Ada Boost[8, 17] algorithm by showing that a strong classifier in the probably approximately correct (PAC) sense can be generated by combining weak classifiers (i.e., simple classifiers which classification performance is only slightly better than random classification.

In [30] proposed a new method to improve the performance of the random forests by increasing the diversity of each tree in the forests. During the training process of each individual tree, different rotation spaces are concatenated into a higher space at the root node. The best split is exhaustively searched within this higher space. The location of the best split decides the rotation. This method is 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.

In [24] 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 C4.5, random forest and random tree algorithms. The results obtained from the experiments showed that the proposed ensemble method achieved better compared with the outcomes of the other Base and Meta base classifiers.

MACHINE LEARNING TECHNIQUES

We selected several learning techniques to benchmark the rainfall prediction power. These are Support Vector Machine (SVM), Naïve Bayes (NB) and Neural Network (NN), all formed under the supervised learning techniques. A key characteristic behind supervised machine learning technique lies in selection of appropriate technique with appropriate features. Therefore, the performance levels of such techniques vary, opening the door for improvement by combining multiple techniques or improving techniques. Five machine learning methods were implemented to create the rainfall prediction models.

C4.5 Algorithm

C4.5 is one of the most effective classification methods. It works better for the prediction of post-graduation course than other decision tree induction classification algorithms [4]. Table 1 shows the pseudo code of algorithm.

Table 1: C4.5 pseudo code

Input: Dataset D
1. Tree = {a}6
2. If D is 'pure' OR other stopping criteria met then
3. Terminate
4. End if
5. For all attribute a ϵ do
6. Compute information-theoretic criteria if split on a
7. End for
8. a_{best}= Best attribute according to above computed criteria
9. Tree = Create a decision node that tests a_{best} in the root
10. D_u= Induced sub-datasets from D based on a_{best}
11. For all D_u do
12. $Tree_u$= J48 (D_u)
13. Attach $Tree_u$ to the corresponding branch of Tree
14. End for
15. Return Tree

Naïve Bayes

Naïve Bayes is a supervised machine learning technique belonging to the family of probabilistic classifiers which apply Bayes theory on the independence assumption between the features [31]. In fact, Naïve Bayes aims to identify the probability for each feature by computing the assumptions [10]. Table 2 contains the relevant pseudo code.

Support Vector Machine

A support vector machine is a technique which divides data into two portions using a hyperplane [14]. This division process addresses each class label independently. This can be performed by classifying the data into class x and not class x, then classifying the data into class y and not class y where x and y is the two class labels [32]. The classification performed by computing the distance between each data point and the margin of the hyperplane. Table 3 contains the description of the algorithm.

Table 2: Naïve Bayes pseudo code

Input: Dataset D

1. **For each Feature f**
2. **Compute the assumptions of f values based on class label 1**
3. **End for**
4. **For each Feature f**
5. **Compute the assumption of f values based on class label 2**
6. **End for**
7. **Prediction class = Maximum (assumption label 1, assumption label 2)**
8. **Repeat for all features**

Table 3: SVM pseudo code

1. **Initialize $y_i = y_1$ for i∈I**
2. **Repeat**
3. **Compute SVM solution w, b for dataset with imputed labels**
4. **Compute outputs f_i= (w, x_i) + b for all $x_i$in positive bags**
5. **Set y_i= sgn(fi) for every i∈I and $y_1 = 1$**
6. **For (every positive bag Bi)**
7. **If ($\sum_{i\in I} \frac{1+y_i}{2} = 0$)**
8. **Compute I = $\mathbf{argmax}_{i\in I} f_i$**
9. **Set $y_i = 1$**
10. **End**
11. **End**
12. **While (imputed labels have changed)**
13. **Output (w, b)**

Neural Network

A neural network is a computational approach based on a large collection of neural units loosely modelling the way the brain solves problems with large clusters of biological neurons connected by axons. Each neural unit is connected with many others. Links can be enforcing or inhibitory in their effect on the activation state of connected neural units. Each individual neural unit may have a summation function combining its inputs values as well as.

A threshold function or limiting function on each connection and on the unit itself such that it must surpass it before propagation to other neurons. These systems are self-learning and trained rather than explicitly programmed and excel in areas, in which the solution or feature detection is difficult to express in a traditional computer program [19]. This algorithm use in classification, regression, prediction and clustering [22]. Table 4 shows the pseudo code of the algorithm.

Table 4: Neural network pseudo code

1. **For iteration = 1 to T**
2. **For e = 1 to N (all examples)**
3. **X = input for example e**
4. **Y = output for example e**
5. **Run x forward through network, computing all {a_i }, {in_i }**
6. **For all weights (j, i)**
7. **Compute $\Delta_i = \begin{Bmatrix} (y_i - a_i) \times g'(in_i) \\ g'(in_i) \sum_k w_{i,k} \Delta_k \end{Bmatrix}$**
8. **Repeat**

Random Forest

Random forest is a technique used for many purposes including classification, regression and prediction [16]. Such technique is an ensemble of decision tree which aims at constructing a multitude of decision trees within the training and generating the class as an output [5]. Table 5 shows the pseudo code of such algorithm

Table 5: Random Forest pseudo code

```
1. For simple Tree T
2. For each node
3. Select m a random predictor variable
4. If the objective function achieved (m=1)
5. Split the node
6. End if
7. End for
8. Repeat for all nodes
```

METHODOLOGY

The research methodology has been set to accomplish the objective of this study, which represented by establishing a new ensemble model of multiple machine learning techniques for rainfall prediction. To do so, a research design which contains several phases. The first phase which is the dataset phase is used to identify the data examined in this study by illustrating its source, details and quantity. The second phase which is pre-processing prepares the data for processing. The phase includes two tasks; cleaning which will handle the missing values and normalization which aims to limit the value into specific range. Third is establish comparative analysis among the five techniques in order to identify the best machine learning techniques including Naïve Bayes (NB), C4.5, Support Vector Machine (SVM), Neural Network (NN) and Random Forest (RF).

Dataset

The data set was obtained from the Malaysian Meteorological Department and Drainage and Irrigation Department, Malaysia. Pre-processing tasks were carried out by cleaning and normalizing the data. The location and description of the data obtained can be shown in Table 6.

Table Details of dataset :6

Source	Daily Data	Station Number	Station Name
Malaysian meteorological department	24 hour mean temperature	48650	KLIA, Sepang
	24 hour mean relative humidity	2917401	Sungai Langat, Kajang, Selangor
	Daily total rainfall	2917112	
	Daily means water level	2917401	

In addition, the features located in the dataset include temperature, relative humidity, flow, rainfall and water level. Table 7 shows the details of such features.

Table 7: Features details

Feature	Valid Records	Missing Values
Temperature	1581	0
Relative humidity	1572	9
Flow	1464	117
Rainfall	1569	12
Water level	1464	117

The pre-processing phase prepares the data for processing. Essentially, all data includes irrelevant information and, noisy or uncompleted instances. Handling such data plays an essential role in terms of improving the performance of the prediction process [13]. Hence, two tasks were proposed for this purpose cleaning and normalization. These tasks are illustrated in detail in the following sub-sections. Table 8 provides detailed measurements for each feature.

Table Measurement features details :8

Attribute Name	Attribute Type	Attribute Meter
Temperature	Continuous	ºC
Humidity	Continuous	Percentage of relative humidity, %
Rainfall	Continuous	mm
River flow	Continuous	m^3/s
Water level	Continuous	ms
Class	Nominal	Rainfall-yes Rain off-no

The pre-processing phase prepares the data for processing. Essentially, all data includes irrelevant information and, noisy or uncompleted instances. Handling such data plays an essential role in terms of improving the performance of the prediction process [13]. Hence, two tasks were proposed for this purpose cleaning and normalization. These tasks are illustrated in detail in the following sub-sections.

The cleaning task aims to handle the missing values. In fact, such missing values have the ability to cause incorrect matches in the process of prediction [27]. Therefore, such missing values must be handled. Table 9 shows a sample of data with missing values.

Table 9: Data with missing values

Temperature	Humidity	Rainfall	Flow	Water Level
27.9	85.3	-76.9	3.94	22.37
27.3	86.2	-284	3.82	22.36
27.8	83.6	*	3.67	22.34
27.7	-1.1	0	10.68	22.54
27.3	84.2	11.4	11.93	22.61
27.4	82.8	40	14.6	22.69
27.3	82.3	8.9	20.24	22.89
26.8	85.8	7.7	14.04	22.68
27.3	81.4	0	11.1	22.57
24.7	90.3	0	10.62	22.54
26.0	86.2	0	10.23	22.53
27.7	-1.1	0	8.73	22.45
28.6	73.4	0	?	?
29.3	68.3	0	?	?
29.1	67.8	5.7	?	?
28.8	67.9	11.3	?	?
28.9	64.1	0	?	?
29.1	57.3	0	?	?
28.5	64.8	0	?	?
28.4	67.0	0	?	?
28.3	69.0	0	?	?

As shown in Table 9, the data contain missing values represented by the characters '?', '*' or minus values. In order to overcome such data, this study uses the mean average mechanism for determining such instances. Such mechanisms sum all the instances of a selected attribute, then dividing the sum on the number of records. For instance, in the second attribute (humidity), the missing values will be filled up by adding all instances (87.6, 88.9, 84.7, 85.2, 88.3 and 84.2) and then dividing the results by the number of all instances which is 6. Table 10 shows the same table after applying the mean average mechanism. Table 11 shows the results mean average for each attribute. Hence, the data is ready for the next processing task.

Table 0: Mean average mechanism

Temperature	Humidity	Rainfall	Flow	Water Level
22.3	87.6	2.31	2.78	2.79
26.4	88.9	5.74	4.29	5.74
22.9	84.7	1.68	6.78	1.25
27.8	85.2	5.03	5.46	4.56
24.1	88.3	5.03	4.29	4.56
26.5	86.4	5.03	4.29	4.56
26.9	86.4	2.69	1.64	6.47
29.3	84.2	10.4	2.14	8.46

Table 11: Results average of features

Attribute	Average
Humidity	27.528
Rainfall	81.265
River flow	5.477
Water Level	11.837

The normalization task aims to limit the values within a specific interval such interval will facilitate the prediction as the values will be reduced into a particular range. Normalization is essential for specific algorithms such as NN and SVM [18]. In this study, the interval will span from -1 to 1. Table 12 shows the values before normalization.

Table 12: Values before normalization

Temperature	Humidity	Rainfall	Flow	Water Level
22.3	87.6	2.31	2.78	2.79
26.4	88.9	5.74	4.29	5.74
22.9	84.7	1.68	6.78	1.25
27.8	85.2	5.03	5.46	4.56
24.1	88.3	5.03	4.29	4.56
26.5	86.4	5.03	4.29	4.56
26.9	86.4	2.69	1.64	6.47
29.3	84.2	10.4	2.14	8.46
21.2	86.4	5.03	4.65	4.56

As shown in Table 12, the values vary greatly. The first range within the 20s, the second range within the 80's, the third, fourth and fifth features within the 10's. Therefore, to unify these values, a normalization task will take place to limit such values between -1 to 1. Hence, the mechanism of normalization used by [12] will be used in this study. Such a mechanism can be formulated by using Equation (1):

$$\gamma = \frac{(y_{max} - y_{min}) \times (x - x_{min})}{(x_{max} - x_{min})} + y_{min} \quad (1)$$

From Equation (1), x is the data that has to be normalized. X_{min} is the minimum value of all data while X_{max} refers to the maximum value of all input data. Y is the normalized data, while Y_{min} is the desired minimum value. Y_{max} refers to the desired maximum value. As shown in Table 13, the data has been normalized preparing it for further processing.

Table 0: Normalization task

Temperature	Humidity	Rainfall	Flow	Water Level
-0.728	0.446	-0.855	-0.556	0.572
0.283	1	-0.068	0.031	0.245
-0.580	-0.756	-1	1	-1
0.629	-0.512	-0.392	1	-1
-0.283	1	-0.392	0.760	-1
0.308	1	-0.392	0.760	-1
0.407	1	-1	-1	-0.02

The experiments were divided into two sections; the first was to identify the best parameterization set of the machine learning techniques to be used as the machine learning techniques have a number of options and alternatives that may define the method's success. The best determined set of parameters were then used by the machine learning techniques to the dataset. Secondly, establish a comparative analysis among the five techniques which are Naïve Bayes, C4.5, Support Vector Machine, Neural Network and Random Forest in order to identify the best techniques.

C4.5 generates a decision tree, where each node splits the classes based on the information. The attribute with the highest normalized information gain is used as the splitting criteria. For example, our data set contains temperature, humidity, rainfall, river flow and water level. The C4.5 technique first explores these features to determine which feature is the best for splitting data (feature with high information).

The feature will then be used to split the data into the next feature until it reaches the last destination. The following is the tree generated by the C4.5 technique. The evaluation was performed using confidence factor (CF), MinNumObj (MNO) and Numfolds (NF) parameters. The splitting mechanism consists of 60% training and 40% testing, and evaluation was performed using the common information retrieval metrics precision, recall and F-measure. Table 6 shows the results of the algorithm.

Table 14: Results of C4.5

C4.5 Parameter					
CF	MNO	NF	Precision	Recall	F-Measure
0.25	2	3	70.1%	73.4%	70.1%
0.5	4	5	71.3%	74.2%	71.3%
0.7	6	7	70%	73.4%	72.7%

As shown in Table 14, several values of the parameters were used. The best results were achieved when the parameters were (confidence factor=0.5, Min Num Obj=4, Num folds=5), reporting a precision 71.3% and a recall of 74.2%, both of which will be used in this study.

Naïve Bayes technique being applied using Weka. For each known class value, NB calculates the probabilities for each attribute, conditional on the class value. Then, it uses the product rule to obtain a joint

conditional probability for the attributes. This is followed by using Bayes rule to obtain the conditional probabilities for the class variable. Once this was completed for all class values, the class with the highest probability will be reported [9].

Evaluation has been performed using debug (D), display Modelold Format (DMF) and use Kernel Estimator (KE). The splitting mechanism consists of 60% training and 40% testing. Moreover, the common information retrieval metrics precision, recall, and F-measure were used for evaluation purposes. Table 15 shows the results of the algorithm.

Table 15: Results of NB

NB Parameter					
D	**DMF**	**KE**	**Precision**	**Recall**	**F-Measure**
False	False	False	65.5%	False	False
True	True	True	62.9%	True	True
False	False	True	62.9%	false	False

The best values were realized when the parameters were (debug = False, display Modelold Format= False, use Kernel Estimator = False) where precision was 65.5%, recall was 71.5% and F-measure was 65.5%, all of which will be used in this study.

Support Vector Machine (SVM) was evaluated by applying the libSVM package in Weka. Some parameters have to be fitted to our data to avoid errors due to the SVM being very sensitive to the presence of any inappropriate parameters. The SVM finds the optimal hyperplane with larger margins that is far enough from the data with more support vectors. After these margins were found the SVM will split the data based on the class where the positive example (examples with class yes) above the hyperplane, and negative examples (examples with class no) under the hyperplane [1]. The evaluation was performed using the SVM type, cost and Gama parameters.

The splitting mechanism consists of 60% training and 40% testing. The evaluation was also done using the common information retrieval metrics precision, recall and F-measure. Table 15 shows the results of the algorithm.

Table 15: Results of SVM

SVM Parameter					
SVM Type	**Cost**	**Gamma**	**Precision**	**Recall**	**F-Measure**
C-svc	1	0	53%	72.8%	61.3%
C-svc	2	0.1	53%	72.8%	61.3%
Nu- svc	4	0.25	71.1%	68%	69.1%

The best results were reported when the parameters were (SVM type = Nu-svc, cost =4, gamma =0.25) where the precision was reported to be 71.1 and the F-measure was 69.1%, both of which will be used in this study.

Neural network technique was evaluated using the Multi-Layer Perceptron algorithm in Weka. Neural network has been performed with four hidden layers at a learning rate of 0.3. Back Propagation Neural Network takes all of the features as inputs to the input layer. There are initial weights for each neuron in the network. The network propagates the input pattern layer-to-layer until the output pattern is generated by the output layer. If this pattern differs from the desired output, an error is calculated, then it's propagated backwards through the network from the output layer to the input layer. The weights are modified as the error is propagated and this process of training is repeated until the error is very small [7].

The evaluation was performed using the learning rate (LR), momentum (M) and validation threshold (VL) parameters. The splitting mechanism consists of 60% training and 40% testing. Moreover, evaluation was conducted using the common information retrieval metrics of precision, recall and F-measure. Table 16 shows the results reported by the algorithm.

Table 16: Results of NN

NN Parameter					
LR	**M**	**VL**	**Precision**	**Recall**	**F-Measure**
0.3	0.2	20	72.7%	74.5%	73.2%
0.4	0.2	30	72.3%	73.6%	72.8%
0.5	0.2	10	72.3%	73.3%	72.7%

The best results were achieved when the parameters were (learning rate =0.3, momentum=0.2, validation threshold =20), resulting in a precision of 72.7, recall of 74.5% and an F-measure of 73.2%), all of which will be used in this study.

Random forest is similar to the decision tree with only a slight variation, where it builds an ensemble of trees (forest). The main principle behind the ensemble methods is that a group of "weak learners" can form a

"strong learner". The random forest works by combining trees with the notion of an ensemble. Thus, in the context of ensemble, the trees are weak learners while the random forest is a strong learner which is the random forest [30].

The evaluation was performed using parameters of max depth (MD), number of feature (NF) and number of tree (NT). The splitting mechanism consists of 60% training and 40% testing. Moreover, the evaluation was done using the common information retrieval metrics precision, recall and F-measure. Table 17 shows the results reported by the algorithm.

Table 17: Results of RF

RF Parameter					
MD	**NF**	**NT**	**Precision**	**Recall**	**F-Measure**
0	0	10	68%	69.9%	68.7%
1	1	12	53%	72.8%	61.3%
3	3	15	71.3%	74.4%	70.7%

Several parametric values were used. The greatest results have been achieved when the parameters were (Max depth = 3, Num feature=3, Num tree =15) where precision was 71.3%, recall was 74.4% and F-measure was 70.7%, all of which will be used in this study.

Analysis of Machine Learning Techniques

For evaluating the proposed method, the common information retrieval metrics recall, precision and f-measure merits are employed. The purpose of precision is to evaluate the True Positive (TP) and False Positive (FP), which are correctly and incorrectly classified entities respectively. It can be calculated by using Equation (2):

$$\text{Precision} = \frac{|TP|}{|TP|+|FP|} \quad (2)$$

The aim of recall is to evaluate the true positive in respect to the false negative, which is the entities that not classified at all. This may be calculated as shown in Equation (3):

$$\text{Recall} = \frac{|TP| + FN \vee}{|TP|} \quad (3)$$

With these two values, we often cannot determine if one algorithm is superior to another. For example, if one algorithm has higher precision, but lower Recall than another, how can the superior algorithm be determined? With these two values, we often cannot determine if one algorithm is superior to another. For example, if one algorithm has higher precision, but lower recall than another, how can the superior algorithm be determined? F-measure is the average of precision and recall, and is calculated as follows Equation (4):

$$F - \text{Measure} = 2 \times \frac{\text{precision} \times \text{Recall}}{\text{precision} + \text{Recall}} \quad (4)$$

RESULTS AND DISCUSSION

This section establishes a comparison between the five machine learning techniques that were applied in the experiments which comprised of NB, C4.5, SVM, NN and RF. This comparison was made using precision, recall and F-measure with the two approaches, first by evaluating the parameters of each technique and second establishes a comparative analysis between the five techniques to identify the best one. The evaluation was performed using a common information retrieval metrics of recall, precision and F-measure. The splitting mechanism consists of 60% training and 40% testing. The aim of precision is to evaluate the true positive (TP) entities which are the correctly classified entities, while the false positive (FP) are the incorrectly classified entities. Recall parameter is used for evaluation of the true positive with respect to the false negative, which are unclassified entities while F-measure is the average of precision and recall. Table 18 shows the comparison of the five classifiers. The results show that in the context of the second approach comparison (comparative analysis among the five techniques); neural network outperformed the other techniques due to its precision of 72.7%, recall of 74.5% and an F-measure of 73.2%.

Table 18: Results achieved by five classifiers

Name of Classifier	Precision	Recall	F-Measure
SVM	71.1%	68%	69.1%
C4.5	71.3%	74.2%	71.3%
NN	72.7%	74.5%	73.2%
NB	65.5%	71.5%	65.5%
RF	71.3%	74.4%	70.7%

CONCLUSION

This paper proposes a best method to develop long-terms (i.e. monthly) and short-terms (i.e. daily) weather forecasting model for rainfall prediction by using ensemble technique. Daily meteorological data from 2010 to 2015, for multiple stations in Selangor Malaysia has been used. The dataset contained five predictors for rainfall (temperature, relative humidity, flow, water level and rainfall). In our intensive experiments we have developed a group of base algorithm models (Naïve Bayes (NB), Support Vector Machine (SVM), Neural Network (NN), Random Forest (RF) and C4.5 algorithm). Furthermore, a comparative analysis was conducted to identify the best technique. From the observation during the predictive studies, the five machine learning techniques performed very well. However, between the techniques, NN generally produced a higher result in this predictive study, while NB yielded the weakest result. Based on the findings, it was decided that the NN technique should be used in the predictive approach.

The findings from this study offer several contributions to the current literature. First, it was shown that the use of machine learning techniques shows a good to significant improvement in rainfall prediction models study area. It is important to note that in general, the neural network method consistently outperforms the Naïve Bayes, Support Vector Machine, Random Forest (RF) and C4.5 algorithm. Hopefully, the outcomes from this study may help on addressing a suitable machine learning technique that has a significant impact on improving the performance of rainfall forecasting prediction.

REFERENCES

1. Abe, S., 2004. Fuzzy LP-SVMs for Multiclass Problems. In the Proceedings of the 2004 European Symposium on Artificial Neural Networks, pp: 429-434.
2. Abhishek, K., A. Kumar, R. Ranjan and S. Kumar, 2012. A Rainfall Prediction Model using Artificial Neural Network. In the Proceedings of the 2012 IEEE Control and System Graduate Research Colloquium, pp: 82-87.
3. Badr, H.S., B.F. Zaitchik and A.K. Dezfuli, 2015. A Tool for Hierarchical Climate Regionalization. Earth Science Informatics, 8(4): 949-958.
4. Barros, R.C., M.P. Basgalupp, A.C. De Carvalho and A. Freitas, 2012. A Survey of Evolutionary Algorithms for Decision-Tree Induction. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 42(3): 291-312.
5. Breiman, L., 2001. Random Forests. Machine Learning, 45(1): 5-32.
6. P. Brockwelland R. Davis, 2013. Time series: Theory and methods. Springer Science and Business Media.
7. Burr, G.W., R.M. Shelby, S. Sidler, C. Di Nolfo, J. Jang, I. Boybat, R.S. Shenoy, P. Narayanan, K. Virwani, E.U. Giacometti and B.N.Kurdi, 2015. Experimental Demonstration and Tolerancing of a Large-Scale Neural Network (165 000 Synapses) Using Phase-Change Memory as the Synaptic Weight Element. IEEE Transactions on Electron Devices, 62(11): 3498-507.
8. Freund, Y. and R. Schapire, 1996. Experiments with a New Boosting Algorithm. In the Proceedings of the 1996 13th International Conference on Machine Learning,pp: 148-156.
9. Ghazanfar, M.A. and A. Prugel-Bennett, 2010. A Scalable, Accurate Hybrid Recommender System. In the Proceedings of the 2010 3rd International Conference Knowledge Discovery and Data Mining, pp: 94-98.
10. Govindarajan, M., 2013. Sentiment Analysis of Movie Reviews using Hybrid Method of Naive Bayes and Genetic Algorithm. International Journal of Advanced Computer Research, 3(4): 139-145.
11. Gupta, D. and U. Ghose, 2015. A Comparative Study of Classification Algorithms for Forecasting Rainfall. In the Proceedings of the 2015 4th IEEE International Conference Reliability, Infocom Technologies and Optimization (Trends and Future Directions), pp: 1-6.

12. Htike, K.K. and O.O. Khalifa, 2010. Rainfall forecasting models using focused time-delay neural networks. In the Proceedings of the 2010 IEEEComputer and Communication Engineering, pp: 1-6.
13. Isa, D., L.H. Lee, V. Kallimani and R. Rajkumar, 2008. Text Document Preprocessing With the Bayes Formula for Classification Using the Support Vector Machine. IEEE Transactions on Knowledge and Data Engineering, 20(9): 1264-1272.
14. Isozaki, H. and H. Kazawa, 2002. Efficient Support Vector Classifiers for Named Entity Recognition. In the Proceedings of the 2002 19th International Conference on Computational Linguistics, pp: 1-7.
15. Joseph, J. and T. Ratheesh, 2013. Rainfall Prediction using Data Mining Techniques. International Journal of Computer Applications, 83(8): 11-15.
16. Liaw, A. andM. Wiener, 2002. Classification and Regression by Random Forest. R News, 2(3): 18-22.
17. Merler, S., B. Caprile and C. Furlanello, 2007. Parallelizing AdaBoost by Weights Dynamics. Computational Statistics and Data Analysis, 51(5): 2487-2498.
18. Monira, S.S., Z.M. Faisal and H. Hirose, 2010. Comparison of Artificially Intelligent Methods in Short Term Rainfall Forecast. In the Proceedings of the 2010 13th International Conference Computer and Information Technology, pp: 39-44.
19. O. Nelles,2013. Nonlinear system identification: From classical approaches to neural networks and fuzzy models. Springer Science and Business Media.
20. Nikam, V. and B. Meshram, 2013. Modeling Rainfall Prediction Using Data Mining Method: A Bayesian Approach. In the Proceedings of the 2013 5th IEEE International Conference onComputational Intelligence, Modelling and Simulation, pp: 132-136.
21. Peter M. Nørgård, O. Ravn, Niels K. Poulsen and Lars K. Hansen2000. Neural networks for modelling and control of dynamic systems-A practitioner's handbook, Springer.
22. Ramana, R., B. Krishna, S. Kumar and N. Pandey, 2013. Monthly Rainfall Prediction Using Wavelet Neural Network Analysis. Water Resources Management, 27(10): 3697-3711.
23. Rani, S. and G. Sikka, 2012. Recent Techniques of Clustering of Time Series Data: A Survey. International Journal of Computer Applications, 52(15): 1-9.
24. Salih, A.S.M. and A. Abraham, 2014. Novel Ensemble Decision Support and Health Care Monitoring System. Journal of Network and Innovative Computing, 2: 41-51.
25. Schapire, R.E. and Y. Singer, 2000. BoosTexter: A Boosting-Based System for Text Categorization. Machine Learning, 39(2-3): 135-168.
26. Thomas F. Stocker, D. Qin, Gian K. Plattner, M. Tignor, Simon K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and Pauline M. Midgley, 2013. Climate change 2013: The physical science basis. Cambridge University Press.
27. Teegavarapu, R.S. and V. Chandramouli, 2005. Improved Weighting Methods, Deterministic and Stochastic Data-Driven Models for Estimation of Missing Precipitation Records. Journal of Hydrology, 312(1): 191-206.
28. Wu, C.L. and K.W. Chau, 2013. Prediction of Rainfall Time Series Using Modular Soft Computing Methods. Engineering Applications of Artificial Intelligence, 26(3): 997-1007.
29. Wang, W., R. Arora, K. Livescu and J. Bilmes, 2015. On Deep Multi-View Representation Learning. In the Proceedings of the 2015 32nd International Conference on Machine Learning, pp: 1083-1092.
30. Zhang, L. and P.N. Suganthan, 2014. Random Forests with Ensemble of Feature Spaces. Pattern Recognition, 47 (10): 3429-3437.
31. Thabtah, F., O. Gharaibeh and R. Al-Zubaidy, 2012. Arabic Text Mining Using Rule Based Classification. Journal of Information and Knowledge Management, 11(1): 1-10.
32. Weston, J., 2014. Support vector machine (and statistical learning theory): Tutorial. Retrieved from http://www.cs.columbia.edu/~kathy/cs4701/documents/jason_svm_tutorial.pdf.