

# Ensemble Learning for Rainfall Prediction

Nor Samsiah Sani<sup>1</sup>, Abdul Hadi Abd Rahman<sup>2</sup>  
Afzan Adam<sup>3</sup>

Center for Artificial Intelligence Technology  
Faculty of Information Science and Technology  
Universiti Kebangsaan Malaysia, Bangi, Malaysia

Israa Shlash<sup>4</sup>, Mohd Aliff<sup>5</sup>

Ministry of Agriculture  
Babil Province Branch, Baghdad, Iran<sup>4</sup>  
Instrumentation and Control Engineering  
Malaysian Institute of Industrial Technology  
Universiti Kuala Lumpur, Kuala Lumpur, Malaysia<sup>5</sup>

**Abstract**—Climate change research is a discipline that analyses the varying weather patterns for a particular period of time. Rainfall forecasting is the task of predicting particular future rainfall amount based on the measured information from the past, including wind, humidity, temperature, and so on. Rainfall forecasting has recently been the subject of several machine learning (ML) techniques with differing degrees of both short-term and also long-term prediction performance. Although several ML methods have been suggested to improve rainfall forecasting, the task of appropriate selection of technique for specific rainfall durations is still not clearly defined. Therefore, this study proposes an ensemble learning to uplift the effectiveness of rainfall prediction. Ensemble learning as an approach that combines multiple ML multiple rainfall prediction classifiers, which include Naïve Bayes, Decision Tree, Support Vector Machine, Random Forest and Neural Network based on Malaysian data. More specifically, this study explores three algebraic combiners: average probability, maximum probability, and majority voting. An analysis of our results shows that the fused ML classifiers based on majority voting are particularly effective in boosting the performance of rainfall prediction compared to individual classification.

**Keywords**—Ensemble learning; classification; rainfall prediction; machine learning

## I. INTRODUCTION

Time-series forecasting has recently gained the research interest and has been explored in multiple domains like stock market, finance and climate change studies. Time-series forecasting refers to the analysis process of a sequence of data points containing successive measurements that are made within a specific time interval. The domains mentioned above are currently heavily reliant on time-series data [1-3] Climate change research is one of the domains that utilizes time-series forecasting to analyse the varying weather patterns statistically for a particular period [4]. The nature of climate change data representation across time is the key characteristic of climate change [5].

Weather forecasting is a subset of climate change research that predicts the atmosphere's state at a future time and location [4]. An important application of weather forecasting is rainfall prediction, which is heavily used in various large-scale activities such as food production planning, water resource management and others that rely on water. It is therefore crucial to ensure that rainfall predictions can be further improved, especially with respect to their accuracy and predictive performance, so that the proper preparation and

planning of large-scale activities can be worked out beforehand.

Machine learning has made dramatic improvements and is a core sub-area of artificial intelligence. It also enables computers to discover themselves without being explicitly programmed. A set of machine learning algorithms can be used to obtain meaningful insights into the data that help make effective detection on phishing websites. However, it is still very far from reaching human performance. The machine still needs human assistance to predefine the algorithms on initialization. Several machine learning approaches for rainfall forecasting have been studied for various locations such as South Africa, China and other countries [6-9]. The classifiers that are used for rainfall prediction include the Naïve Bayes, decision tree, support vector machine, neural networks, random forest, genetic algorithm, support vector regression, M5 rules, radial basis neural networks, M5 model trees, and k-nearest neighbours [10-14].

This paper highlights the phishing webpage detection mechanism based on machine learning classification techniques. The rest of the paper is organized in the following manner: Section II presents the rainfall prediction methodology, Section III presents the utilization of machine learning classification techniques, Section IV presents the utilization of ensemble machine learning techniques, and Section V presents the experimental results gained after the implementation of the ensemble classification methods in the rainfall datasets.

## II. METHODOLOGY

Machine learning is one of the most exciting recent technologies. Machine learning had been positioned to address the shortages of human cognition as well as information processing, specifically in handling large data, their relations and the following analysis [15-16]. In general, machine learning studies the research and algorithms construction that can learn from, and derive predictions about data [17-18]. Therefore, the machine learning approach is selected to predict the rainfall.

The research methodology we used in our study can be segregated into four distinct phases. The first phase is the dataset phase, in which we manually identify the data for this study by analysing their sources, amount, and other details. Next, the pre-processing phase prepares the data for further processing by cleaning the data (i.e., addressing missing

values) and normalizing the data to limit the values to specific ranges. The pre-processed data are then used in the third phase to comparatively analyse the five ML techniques to identify the best technique from the five ML classifiers that are noted above. The fourth and final phase focuses on configuring the ensemble method to carry out assessment on the performance of the entire algorithm. Each of these four phases is described further in the subsections that follow.

### A. Dataset

The dataset was obtained from the Drainage and Irrigation Department, and the Malaysian Meteorological Department. The dataset consists of 1,581 instances and was organized into two classes. The first is the ‘active rainfall’ class, containing 428 instances, and the remaining instances are grouped as ‘no rainfall’.

The obtained data on description and location are illustrated in Table I. The features displayed in the dataset include the relative humidity, rainfall, temperature, flow, and water level. The feature details are as described in Table II. Table III provides the detailed measurements for each feature.

TABLE I. DETAILS OF THE DATASET

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 at Kajang Selangor
	Daily total rainfall	2917112	Kajang at Hulu Langat
	Daily means water level	2917401	Sungai Langat at Kajang Selangor

TABLE II. FEATURE DETAILS

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

TABLE III. MEASUREMENT FEATURE DETAILS

Attribute name	Attribute type	Attribute metre
Temperature	Continuous	°C
Humidity	Continuous	Percentage of relative humidity, %
Rainfall	Continuous	mm
River flow	Continuous	m <sup>3</sup> /s
Water level	Continuous	ms
Class	Nominal	Rainfall – yes Rain off - no

### B. Pre-processing

As noted above, the pre-processing phase ensures that the available data are prepared for further processing in subsequent phases. Here, raw data are usually negatively impacted by noise or incomplete information. The pre-processing phase is a crucial stage in enhancing the improvement of the prediction process by ensuring the data are regularized and filtered beforehand [15], [19]. Therefore, we applied two rather common pre-processing subtasks: cleaning and normalization. In this study, Waikato Environment for Knowledge Analysis (Weka) is used as a tool to perform the pre-processing task. Weka is java-based machine learning software that is developed by the University of Waikato, New Zealand, and it has various types of machine learning algorithms and operates on an open source license. It also provides various visualization tools for data analysis as well as predictive modelling.

### C. Cleaning

In the cleaning task, the data obtained are found to contain missing values, which are represented by characters such as ‘?’ and ‘\*’. In fact, such missing values can cause errors in the prediction process. Therefore, these missing values must be addressed. Table IV illustrates a sample of data containing missing values. A mean average mechanism is then used to populate the missing values. The mean average functions are obtained by summing all instances of an attribute that is selected and then dividing the sum by the number of records. In the second attribute (humidity), for example, the missing values are filled by firstly adding all instances (87.6, 88.9, 84.7, 85.2, 88.3, and 84.2), and then dividing the results by the total number of instances, which in this case is 6. Table V shows the mean average for each attribute.

TABLE IV. DATA WITH MISSING VALUES

Temperature	Humidity	Rainfall	Flow	Water level
27.9	85.3	?	3.94	22.37
27.3	86.2	?	3.82	22.36
27.8	83.6	*	3.67	22.34
27.7	*	*	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	*	11.1	22.57
24.7	90.3	*	10.62	22.54
26.0	86.2	*	10.23	22.53
27.7	-1.1	*	8.73	22.45
28.6	73.4	?	?	?
29.3	68.3	?	?	?
29.1	67.8	5.7	?	?
28.8	67.9	11.3	?	?
28.9	64.1	10.9	?	?

TABLE V. AVERAGE FEATURES

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

D. Normalization

In the normalization task, values are limited within a specific interval, in which the interval facilitates the prediction since the values are reduced into specific ranges. Normalization is crucial for particular algorithms like ANN and SVM. Table VI illustrates values prior to normalization. As shown in Table VI, the values are found to vary greatly, although these values are seen to decompose around the 20s and 80s for the first two features and around the 10s for the remaining three features. To unify these values, we chose an interval range of -1 to +1 and use the normalization mechanism that was introduced by [20] as defined in (1):

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

In Eq. (2.1),  $x$  is the data that requires to be normalized.  $X_{min}$  is the minimum value for all data, and  $X_{max}$  refers to the maximum value of all the input data. On the other hand,  $Y$  is the normalized data, while  $Y_{min}$  is the ideal minimum value.  $Y_{max}$  refers to the desired maximum value. Following the normalization task, all values for the five features are converted to be in the range of -1 to +1. As illustrated in Table VII, the data have been normalized to prepare for further processing.

TABLE VI. 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

TABLE VII. 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

E. Evaluation Metrics

For the purpose of evaluating the method proposed, the common information retrieval metrics are employed. The evaluation is carried out through the use of the common information retrieval metrics of recall, F-measure, and precision. Our model predicts 2 classes (rain or not), so sensitivity or recall can reflect the ratio of rain and no-rain correctly identified by the model. The R2, SSE, and MSE are better for continuous values, while our model does not predict such an output. Precision evaluates the true positives (TP) that are classified correctly and the false positives (FP) that are entities classified incorrectly, which could be computed using (2):

$$\text{Precision} = \frac{|TP|}{|TP|+|FP|} \tag{2}$$

The recall parameter is used in assessing the true positives (TP) with respect to the false negatives (FN), which are unclassified entities. This evaluation is calculated as shown in (3):

$$\text{Recall} = \frac{|TP|}{|TP|+|FN|} \tag{3}$$

Lastly, the average of the recall and precision, which is the F-Measure is computed as follows:

$$\text{F-measure} = 2 \cdot \frac{\text{precision} \times \text{Recall}}{\text{precision} + \text{Recall}} \tag{4}$$

III. MACHINE LEARNING MODELS

Numerous learning methods are selected in this study to benchmark the rainfall prediction performance. These are NB, C4.5, SVM, ANN, and RF, which are all supervised learning methods. A notable aspect of the supervised machine learning methods is that they select suitable methods together with parameters and features that are deemed suitable [21-25]. Two main experiments were carried out in order to evaluate the performance of the classifiers. The first identifies the best parameterization set of each classification model to be employed, since the model has a few alternatives as well as options, which would affect the method's success. Different tuning parameters are used to tune every classifier in order to yield highly accurate results. A series of experiments were carried out to obtain the optimal values of each classifier. The performances between the five classifiers are then evaluated and compared. The second experiment analyse the true performance of the classifiers for rainfall prediction.

A. C4.5 Algorithm

In this section, the J48 decision tree, which is included in Weka, is formed based on the C4.5 decision algorithm. C4.5 is one of the most effective classification methods [26]. Table VIII shows the pseudo code of the algorithm. A decision tree is generated by C4.5 in which every node splits the classes with reference to the information. Splitting criteria is selected based on the attribute having the highest normalized information gain. For example, our dataset contains temperature, humidity, rainfall, river flow, and water level. The C4.5 techniques first explores these features to determine which feature is the best for splitting data (a feature with high information). The feature is then used to split the data into the next feature until it reaches the last destination.

TABLE VIII. C4.5 PSEUDO CODE

```

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

The evaluation was performed using confidence factor, *MinNumObj*, and *Numfolds* parameters. The splitting mechanism splits the dataset into 60% for training and 40% for testing, and the evaluation was performed with the use of the common information retrieval metrics of the recall, precision, and F-Measure. Table IX illustrates the algorithm results. As shown in Table IX, several parameter values are used. The best results are achieved when the parameters are (*confidence factor* = 0.5, *MinNumObj* = 4, and *Numfolds* = 5), which results in a precision of 71.3%, a recall of 74.2% and 72.7% F-Measure.

B. Naïve Bayes

Naïve Bayes is classified as supervised machine learning method that belongs to the probabilistic classifiers family which applies Bayes theory to the independence assumption between features [27]. As a matter of fact, Naïve Bayes identifies the probability of every feature by calculating the assumptions. Table X depicts the pseudo code of the Naïve Bayes algorithm.

This section evaluates the Naïve Bayes technique being applied using Weka. For every known class value, NB computes every attribute conditional probability on the class value. Later, it obtains the joint conditional probability for the attributes using the product rule. This process is followed by the use of Bayes rule to obtain the class variable’s conditional probabilities. After completing this process for each class value, the class having the highest probability is reported.

The parameter tuning experiment was carried out in identifying the best parameters from a few different options available. There are two parameters that affect the performance of the NB classifier: debug and use Kernel Estimator. In this study, debug and use Kernel Estimator are tested on two different values (True and False) for choosing the optimal parameter of the NB classifier. When use Kernel Estimator=True, it means the NB model employs a kernel estimator for numeric attributes as opposed to a normal distribution. Moreover, if the debug parameter is set to False, it means that the classifier may not output any extra information to the console. As shown in Table XI, the best outcomes are attained when the parameter (*debug* = False, use *Kernel Estimator* = True) obtains 65.5% precision, 71.5% recall and 65.5% F-Measure.

TABLE IX. RESULTS FOR THE C4.5 PARAMETER TEST

C4.5 Parameter			Precision	Results Recall	F-Measure
Confidence factor	MinNumObj	Numfolds			
0.25	2	3	70.1%	73.4%	70.1%
0.5	4	5	71.3%	74.2%	72.7%
0.7	6	7	70%	73.4%	71.3%

TABLE X. NAÏVE BAYES PSEUDO CODE

```

Input: Dataset D
For each Feature f
Compute the assumptions of f values based on class label 1
End for
For each Feature f
Compute the assumption of f values based on class label 2
End for
Prediction class = Maximum (assumption label 1, assumption label 2)
Repeat for all features
    
```

TABLE XI. RESULTS FOR THE NAÏVE BAYES PARAMETER TEST

NB Parameter		Precision	Recall	F-measure
debug	Use Kernel Estimator			
False	True	65.5%	71.5%	65.5%
False	False	62.9%	68.4%	64.4%

C. Support Vector Machine

This section discusses the evaluation of the support vector machine method by using the libSVM package in Weka. Some parameters have to be fitted to the data to avoid errors due to the SVM being very sensitive to the presence of any inappropriate parameters. The support vector machine is a method that divides data into two sections with the use of a hyperplane<sup>12</sup>. This division process independently addresses every class label, and this could be carried out through classifying the data into class *x* and not class *x*, and then further categorizing the data into class *y* and not class *y*, where *x* and *y* are the two class labels. The classification is carried out by calculating the distance between every data point and the hyperplane’s margin. Table XII contains the description of the algorithm.

TABLE XII. SUPPORT VECTOR MACHINE PSEUDO CODE

```

Initialize yi = y1 for i ∈ I
Repeat
Compute SVM solution w, b for dataset with imputed labels
Compute outputs fi = (w, xi) + b for all xi in positive bags
Set yi = sgn(fi) for every i ∈ I and y1 = 1
For (every positive bag Bi)
If (Σi ∈ I Bi  $\frac{1+y_i}{2}$  = 0 )
Compute l = argmaxi ∈ I fi
Set yi = 1
End
End
While (imputed labels have changed)
Output (w, b)
    
```

The SVM algorithm uses a kernel that is a set of mathematical functions to allow for data classification in a higher dimensional space when such data could not be linearly separated in a lower dimensional space. Various kernel functions are available to govern the above, like linear, polynomial, nonlinear, radial basis function (RBF) and sigmoid. The SVM can be further classified into two categories, namely, the C SVM and the *nu* SVM. C and *nu* refer to the regularization of the parameters that aids in implementing a penalty on the misclassifications that happen when the classes are separated. C ranges from 0 to infinity and *nu* is always between 0-1.

The parameter evaluation was performed using the SVM type and kernel type parameters. The default parameter value for the SVM type is set as *nu*-SVC, which uses a range between 0 and 1 to represent the lower and upper bounds of the number of examples that are support vectors which lie at the wrong side of the hyperplane. Moreover, a number of different parameterization combinations are tested under the kernel type. Here, each kernel type parameter is changed one at a given time while SVM type is remained the same consistently so that any differences due to the kernel type parameter change can be recorded. Four techniques have been established in the literature already, namely, the linear, radial basis function (RBF), polynomial, and sigmoid, and these four are extensively tested in our experiments. Table XIII illustrates the results of the algorithm. The best results are reported when the parameters are (SVM type = *nu*-SVC, kernel type=RBF), where the precision is reported to be 71.1%, the recall is 72.8% and the F-Measure is 69.1%.

D. Neural Networks

Neural networks were originally motivated by modelling machines that replicate the brain's functionality. Every neural unit is linked to many others. Links could either be inhibitory or enforcing in nature, with regards to their activation state effect of the connected neural units. Every individual neural unit could have a summation function combining its input values [28]. This algorithm is used in regression, classification, prediction and clustering [28]. Table XIV depicts the pseudo code of the algorithm.

There are two parameters that significantly affect the neural network performance classifiers: the number of hidden layers and the value of learning rate. To get the optimal hidden layer value, a range of values are tested from 2 to 10 (at an increment of 2), and the learning rate is tested on five different values from 0.02 to 0.10 (at an increment of 0.02). As shown in Table XV, the best outcomes are attained when the parameters (learning rate = 0.02 and the hidden layer =4) obtain 72.7% precision, 74.5% recall and 73.2% F-Measure. Therefore, the optimal parameters of the ANN are set as the learning rate = 0.02 and the hidden layer=4. The parameter evaluation was performed with 100 iterations.

E. Random Forest

Random forest is a method employed for many purposes, including regression, classification, and prediction. This method is an ensemble of decision trees aiming to construct, within the training data, a multitude of decision trees and

generate the class as the output. Table XVI depicts the algorithm's pseudo code.

The random forest classifier is tuned using the maximum depth of the tree (Max Depth) and the number of features to randomly investigate (Num Features) and the number of tree (Num Tree) parameters. The Max Depth, Num Feature and Num Tree is tested on three different values which are (1, 5 and 10), (0, 3 and 5) and (10, 12 and 15), respectively. Experimental outcomes reveal the classification performance of the RF classifier is increased when the depth, the number of features and the number of tree increase. The obtained parameter tuning result is reported in Table XVII. The best results are reported when the parameters are (Max Depth=10, Num Feature=5, Num Tree=15), where the precision is reported to be 71.3%, the recall is 74.4% and the F-Measure is 70.7%. Thus, the optimal values are Max Depth=10 Num Feature=5 and Num Tree=15.

TABLE XIII. RESULTS FOR THE SUPPORT VECTOR MACHINE PARAMETER TEST

SVM parameter		Precision	Recall	F-Measure
SVM type	Kernel type			
<i>nu</i> -SVC	Linear	53%	71.8%	61.3%
<i>nu</i> -SVC	Polynomial	53%	68%	61.3%
<i>nu</i> -SVC	RBF	71.1%	72.8%	69.1%
<i>nu</i> -SVC	sigmoid	69.5%	70%	65.2%

TABLE XIV. NEURAL NETWORKS 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  $\Delta_i = \{ (y_i - a_i) \times g'(in_i) \sum_k w_{i,k} \Delta_k \}$
8. Repeat

TABLE XV. RESULTS FOR THE NEURAL NETWORKS PARAMETER TEST

ANN Parameter		Precision	Recall	F-measure
Learning rate	Hidden Layer			
0.02	2	72.7%	74.5%	73.2%
0.04	4	72.3%	73.6%	72.8%
0.06	6	72.3%	73.3%	72.7%
0.08	8	72.4%	73.3%	72.6%
0.10	10	72.4%	73.5%	72.9%

TABLE XVI. 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

TABLE XVII. RESULTS FOR THE RANDOM FOREST PARAMETER TEST

RF Parameter			Precision	Recall	F-Measure
Max Depth	Num Feature	Num Tree			
1	0	10	68%	69.9%	68.7%
5	3	12	53%	72.8%	61.3%
10	5	15	71.3%	74.4%	70.7%

#### F. The Performance Evaluation of Different Classifiers

Extensive parameterization tests were performed in quantifying each parameter's influence for the optimization of the classification models. Numerous parameters crucial to maximizing the model's performance are selected from the tests. Whereas, other parameters are classified as less sensitive. Based on the finalized parameters, the classification models are properly executed to quantify their performance on rainfall prediction. Five machine learning methods are identified in the study. They are the Naïve Bayes (NB), C4.5, neural network (ANN), support vector machine (SVM), and random forest (RF). This phase quantifies the performance of the mentioned machine learning methods and determines the best overall methods for rainfall prediction. From the observations during the predictive studies, the results revealed the most effective classification to be the ANN and that the NB yielded the weakest result. Hence, the ANN this time is benchmarked against the ensemble method. Table XVIII shows the comparison of the five classifiers.

TABLE XVIII. RESULTS FOR THE CLASSIFICATION MODELS

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

The performance levels of ML-based predictions vary between the studies, although a neural network classification technique has a slight performance advantage compared to other classifiers. With individual ML classifier techniques for rainfall predictions already extensively documented, the fusion of various ML classification techniques based on an ensemble methodology presents an opportunity to tap any possibility to improve prediction performance. Apart from that, the varying performance levels of such techniques create space for improvement through the combination of various methods or improving techniques.

#### IV. ENSEMBLE METHODS FOR MODEL PREDICTION

The combination of multiple classifiers that result in one subsequent model is known as the ensemble model. Recently, ensemble techniques have been increasingly utilized to improve the prediction performance of classification tasks [29]. In general, there are three common issues faced in most single classification techniques that can be improved when using multiple classifier instances.

1) *Statistical reasoning*: In the event the training data amount is not sufficient, a learning algorithm extracts a weak hypothesis. The combinations of many classifiers, however, have a tendency to find a stronger hypothesis.

2) *Computational reasoning*: An appropriate hypothesis for an individual classifier (such as neural networks) may be more difficult as well as time consuming. Combining multiple classifiers (experts) with an appropriate parameterization (considering speed, efficiency and accuracy) may provide a better hypothesis while reducing the computation time through the enforcement of each classifier's strength in this case.

3) *Representational reasoning*: An individual classifier at times could not represent true hypothesis in the hypothesis space. In the case of ensemble methods, the formation of weighted sum of the hypotheses from the hypothesis space expands the hypothesis space in providing a hypothesis that is more presentable [29].

Ensemble methodology basically works by weighting numerous individual classifiers and later combining them in order to obtain a new classifier that theoretically outperforms their individual instances. The use of different classifiers from various different learning algorithms is an effective method in addressing the diversity among classifiers since it has the potential in minimizing errors or increasing the prediction performance by basing on diverse approaches [29]. The purpose of ensemble methodology is therefore to build a predictive model through the integration of multiple models.

Various fields of study have reported successful outcomes from the use of the ensemble method, such as in healthcare, information retrieval and statistics. Research of the ensemble method has greatly increased from the 1990s onwards. In order to improve the single model's predictive performance. Based on the suggestion [30], an ensemble of neural networks configured similarly. Reference [31] laid out the foundations for the award-winning AdaBoost. Reference [31] and [32] algorithm revealed that through the combination of weak classifiers, a strong classifier in the probably approximately correct (PAC) sense can be generated. Reference [33] proposed a novel ensemble health care decision support method to assist an intelligent health monitoring system that utilizes a Meta classifier voting system made up of three base classifiers, which are the C4.5, random forest and random tree algorithms [32]. Reference [32] employed the ensemble neural network for breast cancer diagnosis, where the researchers combined several neural network outputs that are fused to construct an ensemble output using the simple averaging algorithm. From the study, they found that the ensemble neural network improved the generalization with less false positive malignant diagnoses while accelerating the learning process. Furthermore, [34] adapted multiple neural networks as a method to improve the robustness of predictions. They used several methods such as linear combinations and stacked generalizations to combine member networks. From the study, they found that two combination methods, i.e., selective combination and network combination with various structures, are the best performers that greatly improved model performance.

Rainfall forecasting gained the attention of many researchers' as it has interesting challenges represented by the complexity that lies beneath predicting specific factors that are linked to rainfall like wind and humidity [21-23], [35-36]. Current techniques for rainfall prediction utilize only individual classifiers such as neural networks [21], the k-nearest neighbours [22], support vector machine [22] and others. Based on encouraging results from the ensemble methods application in various fields, the ensemble classification technique is applied to rainfall prediction by leveraging three linear algebraic combiners: majority voting, average probability, and maximum probability. Since there have not been any applications of the ensemble method for rainfall forecasting in Malaysia, this research should serve in addressing numerous supervised learning methods for the above case.

To assess the performance of our ensemble classification for rainfall prediction, we first fused all five machine learning models (i.e., the Naïve Bayes, C4.5, neural network, support vector machine, and random forest). Their combinations are based on three linear algebraic combiners, which are majority voting, average probability, and maximum probability. Equation (7), (8) and (9) describe their mathematical derivations. Voting is essentially the general blueprint for combining classifiers into ensembles. Voting schemes are divided into unweighted as well as weighted voting schemes. Unweighted schemes are including maximum probability, minimum probability, the product of probabilities, majority voting, and the average of probabilities. Whereas, weighted schemes encompass simple weighted voting, best-worst weighted voting, rescaled weighted voting, and quadratic best-worst weighted voting [32-34].

This study only focuses on unweighted voting schemes. In principal, the  $k$  base classifier's binary outputs are being combined such that the output of the ensemble is chosen based on the highest number of votes. Any of the unweighted schemes are used to guide the classifiers. Equation (5) highlights the basic form of the classification ensemble calculation.

$$H(X) = arg i = 1 \dots n \max(LCi(X)) \quad (5)$$

Here,

$i = 1 \dots, n$  is the number of classes and

$J = 1 \dots, m$  is the number of classifiers that are contained in ensemble method.

$LCi(X)$  is thus represented as any combination scheme in determining the final output of the classifier ensemble.

#### A. Majority Voting

A classification of unlabelled instances is performed in this combining scheme based on the class having the highest (the most frequent) number of votes. This method is employed most of the time as a combination technique to compare newly proposed techniques [37]. The majority voting is defined by (6) as follows:

$$H(X) = arg i = 1 \dots n \max \{Si = \sum_{j=1}^m I(hi(X))\} = Y \quad (6)$$

#### B. Average of Probabilities

Every classifier output a probability distribution vector over all classes that are relevant in the probabilistic approach, as shown by (7). The individual probability values are averaged (or summed) by all classifiers for every class, and the class that yields the maximum value is chosen at last [32].

$$LCi(X) = \frac{1}{m} \sum_{j=1}^m Pj(wi \vee X) \quad (7)$$

#### C. Maximum Probabilities

The maximum probability approach is almost identical to the average probability approach described above [32]. Here, the only difference is in the selection of a probability with the maximum value, as highlighted in (8) below:

$$LCi(X) = maxj = 1 \dots m \{Pj(wi \vee X)\} \quad (8)$$

### V. RESULTS AND DISCUSSION

Table XVIII shows the comparison of the five classifiers using the three metrics based on the test dataset. From the table, the neural network outperforms the other techniques with a precision of 72.7%, a recall of 74.5%, and an F-Measure of 73.2%. The predictive results obtained from the neural network will be compared to the ensemble rainfall prediction approach.

We further discuss the experimental results that are obtained by applying the ensemble method with the three unweighted combiners (majority voting, average probability, and maximum probability) for rainfall prediction. This combination can be used to combine any of the five classifiers (Naïve Bayes, C4.5, support vector machine, neural networks, and random forest). The ensemble model works by combining classifiers from both groups of 'weak' and 'strong' classifiers, thereby forming an ensemble. Thus, in ensemble terms, the classifiers are weak learners, while the ensemble model is a strong learner. The evaluation of the ensemble methods is performed by the use of common information retrieval metrics as follow: recall, precision and F-Measure. The outcomes are based on a similar splitting mechanism for the dataset of 60% as training data and 40% as testing data.

Tables XIX, XX and XXI highlight the test data results from multiple combinations and the three ensemble methods based on the selected metrics. The single ANN classifier is also included in the comparison for benchmark purposes. For the precision metric, Table XIX demonstrates that the combination of the SVM, C4.5 and ANN methods via the majority voting scheme yielded the highest precision accuracy at 76%. This result is followed closely by the same combined ML tools but with the average probability scheme close to 75% accuracy. There is generally a 2% to 3% increase in precision accuracy if the best ensemble methods are to be compared to the single ANN classifier. On the other spectrum, the full or 4 ML tool combination that was tested resulted in disappointing precision accuracy for most of the weighting schemes, and was far below the 73% threshold that was posed by the single ANN classifier, except for the full combined ML tools with the maximum probability ensemble scheme, which scored an unusual 71% accuracy for such combinations.

TABLE XIX. COMBINATION OF TOOLS AND ENSEMBLE RESULTS BASED ON PRECISION METRIC

Combination	Precision		
	Majority Voting	Average Probability	Maximum Probability
Combination of (SVM, ANN, NB, C4.5, and RF)	53%	53%	71%
Combination of (ANN, NB, C4.5, and RF)	53%	53%	53%
Combination of (SVM, C4.5, and ANN)	76%	75%	71%
Combination of (SVM, C4.5, and NB)	74%	73%	71%
Combination of (NB and ANN)	70%	67%	67%
Single classifier (ANN)	73%		

For the recall accuracy, Table XX highlights a similar pattern to the precision metric, whereby the combination of the SVM, C4.5 and ANN tools and the use of the majority voting and average probability schemes scored the highest recall accuracy at 77%. This is a 2% increase from the single ANN classifier, which was 75%. Except for the two combination or ensemble schemes that were mentioned above, the remaining combination or ensemble schemes all performed slightly worse than the ANN classifier. Table XXI highlights the F-Measure accuracy for the employed classifiers and combiners. Again, the same combination of the SVM, C4.5 and ANN tools based on both the majority voting and average probability ensemble schemes yield the best accuracy as compared to the other classifiers. It is noted that the remaining fusion classifiers and ensemble schemes scored well below the 73% of the ANN classifier.

From the experiments, the fusion of the classifiers was shown to generally boost prediction diversity without compromising the individual prediction strengths of the individual classifiers. However, care needs to be considered, as not every fusion strategy works at improving performance over single classifiers. This is demonstrated by the fact that in all 3 performance metrics tests, the fusion of the 4 classifiers and all classifiers degrade the performance accuracy regardless of the ensemble scheme that is chosen. On the other hand, the selected fusion classifiers based on the majority voting scheme are superior to the single classifiers. Particularly, a combination of three tools with the minimum presence of the SVM and C4.5 algorithms ensure that superior performance can be achieved. Table XXII illustrates the confusion matrix for a two-class classifier (i.e., Rain and No Rain). The matrix is a summary of the prediction results that are obtained from the best ensemble method (i.e., majority voting) for the rainfall classification problem on the test dataset (i.e., 632 instances). In the context of our study, the entries in the confusion matrix carry the following meaning: true positive (TP) indicates the number of instances that correctly predict that it will rain, which is equivalent to 438 days; true negative (TN) shows the number of instances that correctly predict that it will not rain,

which is equivalent to 39 days; and false positive (FP) shows the number of instances that incorrectly predict that it will rain, which is equivalent to 133 days. FP is also known as false positive predictions. Finally, false negative (FN) indicates the number of instances that incorrectly predict that it will not rain, which equivalent to 22 days and is otherwise known as false negative predictions.

TABLE XX. COMBINATION OF TOOLS AND ENSEMBLE RESULTS BASED ON RECALL METRIC

Combination	Recall		
	Majority voting	Average probability	Maximum probability
Combination of (SVM, ANN, NB, C4.5, and RF)	73%	73%	68%
Combination of (ANN, NB, C4.5, and RF)	73%	73%	73%
Combination of (SVM, C4.5, and ANN)	77%	77%	68%
Combination of (SVM, C4.5, and NB)	76%	73%	68%
Combination of (NB and ANN)	73%	72%	72%
Single classifier (ANN)	75%		

TABLE XXI. COMBINATION OF TOOLS AND ENSEMBLE RESULTS BASED ON F-MEASURE METRIC

Combination	F-Measure		
	Majority voting	Average probability	Maximum probability
Combination of (SVM, ANN, NB, C4.5, and RF)	61%	61%	69%
Combination of (ANN, NB, C4.5, and RF)	61%	61%	61%
Combination of (SVM, C4.5, and ANN)	76%	75%	69%
Combination of (SVM, C4.5, and NB)	63%	73%	69%
Combination of (NB and ANN)	70%	68%	68%
Single classifier (ANN)	73%		

TABLE XXII. CONFUSION MATRIX ON MAJORITY VOTING WITH 40% TEST SET

Number of instances = 632		Predicted		
		Class: No rain	Class: Rain	Total
Actual	Class: No rain	39	133	172
	Class: Rain	22	438	460
Total		61	571	632

## VI. CONCLUSIONS

Rainfall forecasting is a process to predict potential rainy locations by considering numerous factors like humidity, wind speed, level of water, and temperature. The common methods employed in rainfall forecasting are supervised machine learning methods in which predefined example data are first trained and then followed by the prediction using the testing data. The key challenge of these methods is in identifying suitable mechanisms, the sensitivity of the objective functions, and the dependency on treating features. These differences lead to inconsistent performances, making the selection process of a suitable method for rainfall prediction a task that is challenging. This paper proposes an improved method to develop long-term (i.e., monthly) and short-term (i.e., daily) weather forecasting models for rainfall predictions using the ensemble technique. Therefore, this paper proposes an improved method to develop long-term (i.e., monthly) and short-term (i.e., daily) ensemble weather forecasting models for rainfall predictions by using three linear algebraic combiners (i.e., majority voting, average probability and maximum probability) for combining five rainfall prediction models (i.e., the Naïve Bayes, C4.5, neural network, support vector machine, and random forest). By leveraging daily meteorological data in Selangor, Malaysia, over a period of 6 years (from 2010 to 2015), 1581 instances were obtained and organized into two classes. The first is the 'active rainfall' class containing 428 instances, while the remaining instances are grouped as 'no rainfall'. We have experimented a group of base algorithm models including the NB, SVM, ANN, RF and C4.5. From the analysis, all five ML techniques that were mentioned are shown to perform very well, although the ANN technique in particular is generally found to perform the best, while the NB technique is relatively the weakest. The study further explored the ensemble's potential for further upper-bound improvements in the rainfall prediction model. The ensemble experiment analysis revealed that the rainfall prediction is indeed enhanced using the ensemble method. Particularly, the ensemble method based on majority voting is shown to provide better predictive performance with high precision, recall, and F-Measures compared to other experimented algebraic combiners. Overall, the combiners have been demonstrated to be superior to single classification methods. Such results complement previous findings on ML methods in rainfall prediction and hence, our recommendation is to use ensemble ML algorithms as an effective approach for the above. It is hoped that the outcomes of this study may help to find suitable machine learning techniques that improve the performance of rainfall forecasting predictions.

## ACKNOWLEDGMENT

The authors would like to extend their gratitude and thank Universiti Kebangsaan Malaysia (UKM) and Ministry of Education Malaysia (MOE) under the Research University Grant (project code: GUP-2019-060 and FRGS/1/2018/ICT02/UKM/02/6) for funding and supporting this research.

## REFERENCES

- [1] C. W. J. Granger, and P. Newbold, *Forecasting Economic Time Series*. Cambridge, MA: Academic Press, 2014.
- [2] A. S. Weigend, *Time Series Prediction: Forecasting the Future and Understanding the Past*, 1st ed., Milton Park, Abingdon: Routledge, 2018.
- [3] G. Nan, S. Zhou, J. Kou, and M. Li, "Heuristic bivariate forecasting model of multi-attribute fuzzy time series based on fuzzy clustering," *Int. J. Inf. Tech. Decis.*, vol. 11, pp. 167–195, January 2012.
- [4] M. E. Mann, and P. H. Gleick, "Climate change and California drought in the 21st century," *Proc. Natl. Acad. Sci.*, vol. 112, pp. 3858–3859, March 2015.
- [5] M. J. Lee, I. Park, J. S. Won, and S. Lee, "Landslide hazard mapping considering rainfall probability in Inje, Korea," *Geomatics, Nat. Hazards Risk*, vol. 7, pp. 424–446, January 2016.
- [6] C. C. Stephan, N. P. Klingaman, P. L. Vidale, A. G. Turner, M. E. Demory, and L. Guo, "A comprehensive analysis of coherent rainfall patterns in China and potential drivers. Part I: Interannual Variability," *Clim. Dyn.*, vol. 50, pp. 4405–4424, June 2017.
- [7] N. A. B. Klutse, B. J. Abiodun, B. C. Hewitson, W. J. Gutowski and M. A. Tadross, "Evaluation of two GCMs in simulating rainfall inter-annual variability over Southern Africa," *Theor. Appl. Climatol.*, vol. 123, pp. 415–436, February 2016.
- [8] K. Sittichok, A. G. Djibo, O. Seidou, H. M. Saley, H. Karambiri, and J. Paturel, "Statistical seasonal rainfall and streamflow forecasting for the Sirba watershed, West Africa, using sea-surface temperatures," *Hydrol. Sci. J.*, vol. 61, pp. 805–815, April 2016.
- [9] J. Wu, J. Long, and M. Liu, "Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm," *Neurocomputing*, vol. 148, pp. 136–142, January 2015.
- [10] J. Sulaiman, and S. H. Wahab, "Heavy rainfall forecasting model using artificial neural network for flood prone area," in *Proc. of IT Convergence and Security 2017*, H. C. Kim, and K. J. Kim, Eds. Singapore: Springer, 2018, pp. 68–76.
- [11] B. T. Pham, D. T. Bui, M. B. Dholakia, I. Prakash, and H. V. Pham, "A comparative study of least square support vector machines and multiclass alternating decision trees for spatial prediction of rainfall-induced landslides in a tropical cyclones area," *Geotech. Geol. Eng.*, vol. 34, pp. 1807–1824, December 2016.
- [12] S. Cramer, M. Kampouridis, A. A. Freitas, and A. K. Alexandridis, "An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives," *Expert Syst. Appl.*, vol. 85, pp. 169–181, November 2017.
- [13] M. Kim, Y. Kim, H. Kim, W. Piao, and C. Kim, "Evaluation of the k-nearest neighbor method for forecasting the influent characteristics of wastewater treatment plant," *Front. Env. Sci. Eng.*, vol. 10, pp. 299–310, April 2016.
- [14] S. Zainudin, D. S. Jasim, and A. Abu Bakar, "Comparative analysis of data mining techniques for Malaysian rainfall prediction," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 6, pp. 1148–1153, December 2016.
- [15] N. S., Sani, I. I. S. Shamsuddin, S. Sahran, A. H. A. Rahman, and E. N. Muzaffar, "Redefining selection of features and classification algorithms for room occupancy detection," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, pp. 1486–1493, October 2018.
- [16] P. S. Maya Gopal, "Performance evaluation of best feature subsets for crop yield prediction using machine learning algorithms," *Appl. Artif. Intell.*, vol. 33, pp. 621–642, June 2019.
- [17] J. D. Holliday, N. Sani, and P. Willett, "Calculation of substructural analysis weights using a genetic algorithm," *J. Chem. Inf. Model.*, vol. 55, pp. 214–221, February 2015.
- [18] J. D. Holliday, N. Sani, and P. Willett, "Ligand-based virtual screening using a genetic algorithm with data fusion," *MATCH Commun. Math. Comput. Chem.*, vol. 80, pp. 623–638, June 2018.
- [19] N. S. Sani, M. A. Rahman, A. A. Bakar, S. Sahran, and H. M. Sarim. 2018. Machine learning approach for bottom 40 percent households (B40) poverty classification. *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, pp. 1698–1705, September 2018.

- [20] K. K. Htike, and O. O. Khalifa, "Rainfall forecasting models using focused time-delay neural networks," in IEEE International Conference Computer and Communication Engineering (ICCCE), pp. 52–57, May 2010.
- [21] R. V. Ramana, B. Krishna, S. Kumar, and N. Pandey, "Monthly rainfall prediction using wavelet neural network analysis," *Water Resour. Manag.*, vol. 27, pp. 3697–3711, August 2013.
- [22] M. Devak, C. T. Dhanya, and A. K. Gosain, "Dynamic coupling of support vector machine and K-nearest neighbour for downscaling daily rainfall," *J. Hydrol.*, vol. 525, pp. 286–301, June 2015.
- [23] D. Gupta, and U. Ghose, "A comparative study of classification algorithms for forecasting rainfall, in 4<sup>th</sup> International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), pp. 1–6, September 2015.
- [24] S. Shabudin, N. A. Sani, K. A. Z. Ariffin, and M. Aliff, "Feature selection for phishing website classification," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 11, pp. 587–595, January 2020.
- [25] A. Abu, R. Hamdan, and N. S. Sani, "Ensemble learning for multidimensional poverty classification," *Sains Malays.*, vol. 49, pp. 447–459, 2020.
- [26] Chauhan, H., and A. Chauhan, "Implementation of decision tree algorithm c4.5," *Int. J. Sci. Res.*, vol. 3, pp. 1–3, October 2013.
- [27] M. Mayo, and E. Frank, "Improving naive bayes for regression with optimized artificial surrogate data," *Appl. Artif. Intell.*, vol. 34, pp. 484–514, May 2020.
- [28] Y. Wang, and L. Najjar, "Factor neural network theory and its applications," *Int. J. Inf. Tech. Decis.*, vol. 14, pp. 239–251, March 2015.
- [29] Y. Ren, L. Zhang, and P. N. Suganthan, "Ensemble classification and regression-recent developments, applications and future directions," *IEEE Comput. Intell. Mag.*, vol. 11, pp. 41–53, January 2016.
- [30] L. K. Hansen, and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 12, pp. 993–1001, October 1990.
- [31] Y. Freund, and R. E. Schapire, "Experiments with a new boosting algorithm," in Proceedings of the Thirteenth International Conference on International Conference on Machine Learning (ICML), vol. 96, pp. 148–156, July 1996.
- [32] A. J. Sharkey, N. E. Sharkey, and S. S. Cross. "Adapting an ensemble approach for the diagnosis of breast cancer, in International Conference on Artificial Neural Networks, pp. 281–286, September 1998.
- [33] A. S. M. Salih, and A. Abraham, "Novel ensemble decision support and health care monitoring system," *Int. J. Netw. Innov. Comput.*, vol. 2, pp. 041–051, 2014.
- [34] Z. Ahmad, and J. Zhang, "A comparison of different methods for combining multiple neural networks models," in IEEE Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN'02 (Cat. No. 02CH37290), vol. 1, pp. 828–833, May 2002.
- [35] J. Joseph, and T. K. Ratheesh, "Rainfall prediction using data mining techniques," *Int. J. Comput. Appl.*, vol. 83, pp. 11–15, January 2013.
- [36] S. S. Monira, Z. M. Faisal and H. Hirose, "Comparison of artificially intelligent methods in short term rainfall forecast," in 13th International Conference Computer and Information Technology (ICCIT), pp. 39–44, December 2010.
- [37] A. Onan, S. Korukoğlu, and H. Bulut, "A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification," *Expert Syst. Appl.*, vol. 62, pp. 1–16, November 2016.