International Journal of Advanced Technology and Engineering Exploration, Vol 8(84)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2021.874586

Research Article

An application of logistic model tree (LMT) algorithm to ameliorate Prediction accuracy of meteorological data

Sheikh Amir Fayaz[1], Majid Zaman[2*] and Muheet Ahmed Butt[3]
Research Scholar, Department of Computer Sciences, University of Kashmir, J&K. India[1]
Scientist, Directorate of IT&SS, University of Kashmir, J&K. India[2]
Scientist, Department of Computer Sciences, University of Kashmir, J&K. India[3]

Received: 11-August-2021; Revised: 06-November-2021; Accepted: 10-November-2021
©2021 Sheikh Amir Fayaz et al. This is an open access article distributed under the Creative Commons Attribution (CC BY) License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract
Traditional and ensemble methods are linear models which are considered the most popular techniques for various learning tasks for the prediction of both nominal and numerical values. In this study, we demonstrate the novel concept and working of an algorithm, which customizes the idea of various classification problems with the use of logistic regression in place of linear regression, called a Logistic Model Tree (LMT) algorithm. This study briefly describes the analytical and mathematical implementation of LMT on geographical data for the prediction of rainfall. A step-wise approach is used for the construction of a LMT, which involves a decision tree inducer (C4.5) for the splitting criteria and logistic regression functions for the pruning in which standard regression errors using Cost-Complexity Pruning (CCP) are calculated at each node. This work assesses the abilities of the LMT for the prediction of rainfall across the Kashmir province of the Union Territory of Jammu & Kashmir, India. The implementation methodology was prepared based on six years of historical-geographical data of Kashmir province. It was collected from three different substations having four explanatory independent variables, namely: max temp, min temp and humidity measured at 12 A.M and 3 P.M, moreover a target variable indicating presence and absence of rain. The overall result shows that LMT performs better with the accuracy of 87.23%. At the later stage, we compared the performance of LMT to several algorithms on the same set of data, and show that LMT produces more accurate and compact results.

Keywords
C4.5, Logistic regression, Logistic model tree, Pruning, Information gain.

1.Introduction
Weather forecasting is considered as one of the approaches which are used to check the state of the atmosphere in the future at a specified location. A type of weather forecasting called as rainfall prediction has the highest influence in the farming & agricultural sector and other various sectors like natural disaster management etc. Accurate and timely rainfall prediction is one of the important factors in today's environment. Since the rainfall parameters keep on changing around the world in different places have different factors which are responsible for the rainfall. These parameters include the wind, vapour pressure, humidity at different intervals, temperature, season, evaporation etc. and are considered as the important meteorological parameters for predicting the rainfall in future time [1].

Since accurate and timely rainfall prediction still remains an open challenge for the researchers. Different machine learning techniques are used for the accurate prediction based on the historical geographic data. For this, we can use many regression and classification techniques to check the overall accuracy and the performance. Since, numerous researches have been conducted on weather forecasting and also keeping this fact in mind that different machine learning techniques produce different accuracies, so it is vital to choose the best algorithm that produces better accuracy and performance and model it as per the requirements [2].

Decision tree algorithms have been a choice of tool in the machine learning algorithm over the past decade. It was followed up by its modified version, including the more successful one i.e., random forest. However, there has been more precision and accuracy in modified versions of the various decision tree

*Author for correspondence

algorithms. Decision tree has not given good results for all threshold datasets [3].

One of the modified decision tree algorithms is the logistic Model tree (LMT), first implemented theoretically by Barros et al. [4] in 2011 on the continuous dataset. This concept was then followed by various researchers where the model tree concept was defined with various application measures like stream-flow [5], Top-down induction [6]. Generating rules [7] and so on. Authors in this study [5] concluded that model trees give better results over Artificial Neural Networks (ANN)-Multilayer Perceptron (MLP).

The LMT did not find its recognition across datasets because of the lack of implementation in major tools like python and MATLAB. In this paper, we have implemented LMT on geographical data for the prediction of rainfall, the novelty lies in the mathematical and analytical implementation of LMT and it has given many improvised results as compared to other traditional and ensemble algorithms.

The decision tree is a supervised classification learning technique algorithm. There are quite a lot of distinct ways to execute the same set of tasks and numerous ways to train the machine where we can choose anyone among them. It was developed in the late 1970' s by J. Ross Quinlan, is a graphical representation which consists of internal decision nodes and terminal leaf nodes. The decisions which are made can be explained very easily.

The response functions for the input $a = [1, a_1, a_2, \ldots\ldots, a_n]$ can be well-defined recursively as in Equation 1.

$$\begin{aligned} yx(\boldsymbol{a}) &= \{\, \rho x && if\ \ x\ is\ a\ leaf \\ yx(a) &= \{\, yml(a) && else\ if\ \ g(a) > 0 \\ yx(\boldsymbol{a}) &= \{\, ymr(a) && else\ if\ \ g(a) \leq 0 \end{aligned} \qquad (1)$$

Where x denotes the node tree, if x is the terminal node, then it will return the two values for the binary classification such that, $\rho x \in [0, 1]$. The symbol yml and ymr denotes the left and right nodes of the tree which internally depends on the value of g(a).

Moreover, if x is an internal node, then the decision depends on the left and right sub trees for the classification. A decision tree [8, 9] gives a convincing strategy for decision creation because they allow us to analyze completely the potential outcomes of a decision and provides an outline to enumerate the values of conclusions and the likelihoods. In decision tree learning, a machine learning algorithm called as Iterative Dicotomiser 3 (ID3) is used to generate decisions from the set of data. This construction of decisions uses recursive divide and conquer method (1) in which the training datasets are recursively partitioned into smaller subsets until no further partitions are required [10, 11].

If the dataset available contains numeric attributes, then the decision trees are geometrically inferred as a collection of hyper-planes where each plane acts as a normal to one of the axes. Decision trees with low complexity are naturally contemplated more comprehensibly and the complexity of the model can have the decisive effect on the accuracy and performance. The complexity of the tree can be measured by the following ways which includes the depth of the tree, the number of leaf nodes, total nodes and attributes used in constructing the tree. This tree complexity can be controlled by, some stopping criteria, which shortens the length of the tree without affecting the overall performance of the original decision tree, by employing the pruning method strategies [12].

Many decision tree inducers follow the top-down approach which includes ID3, C4.5 etc. Some of them consist of two abstract phases: growing and pruning (C4.5 and CART) phases and some of the other inducers implement growing phase only. *Table 1* describes some of the decision tree algorithms used in many classification and regression problems. *Table 1* includes the basic traditional algorithms which have been used over the years in various classification and regression processes. These algorithms include one of the simplest in the entire decision tree algorithms called as ID3 algorithm and followed that is the Successor of ID3 algorithm called as C4.5 and then other algorithms like Classification and Regression Trees (CART).

In this paper, we have implemented a theoretical LMT in mathematical and analytical format. This was followed up by implementation in python and its various libraries on "Google Colaboratory" (Google Colab – A well-suited approach that allows us to write and execute the python code through a browser). The snippet of the pseudocode is shown in section 3.5 below. It was observed that LMT performs better by achieving an accuracy measure of 87.23%. Furthermore, in this study, a brief comparative analysis of various traditional and

ensemble algorithms has been made with the LMT to check the overall performance of the model generated.

Table 1 Decision tree inducers

S. No	Algorithm	Description	Source
1	ID3 (Iterative Dicotomiser 3)	• Considered as one of the simplest in all decision tree algorithms. • Here in this algorithm the splitting criterion is decided by the highest information gain. • When instances belong to a single value of the resultant attribute it stops further growing. • This basic algorithm works with the labelled or categorical values only. • Missing data are ignored when the decision tree is built using this approach.	[12]
2	C 4.5	• Considered as the successor of ID3 Algorithm. • Divide and conquer, top-down and recursive approaches are used in the construction of DT. • Tree construction uses information theory concept, where splitting attribute is chosen which has the highest gain ratio. • It can handle numeric values also with the target parameter as discrete valued.	[12]
3	CART (Classification & Regression Trees)	• Constructs a binary tree where each of the internal nodes has exactly two edges. • A Twoing criterion is used as the splitting measure here. • Cost-complexity Pruning after the growing phase is used to prune the tree. • It provides prior probability distribution to the users. • One of the important features of CART is that it has the ability to construct regression trees where the leaves of the tree predict the number instead of class.	[12]

This study describes an efficient approach, LMT, used in the prediction of rainfall that constructs the step wise tree based symplectic logistic models. Like classification and regression trees, LMT builds a tree-based simulations, but, whereas regression trees have values at their leaves, the trees constructed by LMT can have multivariate linear and logistic models and these simulations are very much analogous to the symplectic linear functions. Furthermore, various traditional and ensemble algorithms are presented and compared with LMT to check which one performs better on the historical data of the Kashmir Province for the prediction of rainfall.

The objectives of this work are debated below:

1) To ameliorate a stepwise model that will predict the presence or absences of rainfall.
2) To show a stepwise mathematical implementation of the LMT model with a smaller number of rules as compared to other traditional and ensemble approaches.

The objective is fulfilled by introducing the step wise and mathematical implementation of any decision tree inducer (C4.5) with the logistic regression model on the leaf nodes. This research article is organized as: This section briefly defines Decision Trees and its various types of decision tree inducers Section 2 gives the detailed review analysis of model trees. Section 3 provides the information about and methods used in the implementation of LMT on the historical meteorological data of Kashmir province. A theoretical and mathematical concept of model trees was introduced in a machine learning environment. Furthermore, this section elaborates the step wise concept of building the logistic model trees using decision tree inducer. Section 4 discusses the experimental results based on the implementation of LMT. Section 5 briefly discusses the overall working of the paper & its limitations. Section 6 concludes the study with some future challenges which can be incorporated for the better enhancement of the implemented algorithm and its performance.

2.Literature review

As of now, massive amount of research work has been As of now, massive amount of research work has been performed by various researchers for the prediction of rainfall [13] using different data mining

as well as machine learning techniques. In this study, our main focus will be on model trees and its applications like ground water forecasting, hydrological time series forecasting, wavelet transforms for rainfall modelling and so on.

Onyari and Ilunga et al. (2013) [5] proposed a data mining technique, M5P-Model tree and ANN-MLP model for the prediction of stream flow. In this study the author uses Maximum Relevance Minimum Redundancy (mRMR) technique to choose the relevant inputs. The results show that the Root Mean Square Error (RMSE) value of 2.666 when M5P-model tree was used and this study concludes that the M5P-Momdel tree performs better as compared to ANN-MLP.

Samadi et al. (2014) [14] proposed an assessment of advanced machine learning algorithms M5 prime model and CART for the prediction of Scour depth downstream of free overfall spills.

These Decision tree algorithms were implemented on mahboobi [15] and Azar [16] data in which the training and test sets are used for the development purposes. The implementation of model trees was carried out in Weka – a visualization tool. Various parameters were evaluated for the performance and accuracies were calculated. The results indicate that the model tree performs better as compared to classification and regression tree algorithm. Furthermore, the number of rules in case of Model tree with logarithmic functions is less in number and with higher accuracy measure.

Raza (2015) [17] works on meteorological data for the prediction purposes in which two machine learning algorithms M5 Model Tree and Gene Expression Programming (GEP) are considered. The dataset used in this operation was collected from 1997-2013 of Delhi, India, and the operation was carried out on 7 parameters.

Kisi et al. (2017) [18] proposed a study on hydrological time series forecasting using three heuristic regression techniques which includes Least Square Support Vector Regressor (LSSVR), Multivariate Adaptive Regression Splines (MARS) and M5 model trees. The data used in this comparative study was taken from two sub stations of turkey. The results were compared by taking RMSE, Mean Absolute Error (MAE) and R^2 into consideration. It was observed that LSSVR model performs better results as compared to other two heuristic methods.

Rezaie-balf et al. (2017) [19] proposed a study on groundwater level forecasting using MARS & M5 model tree machine learning algorithms. The dataset comprises of around 10 years of data from Aug 1996 to July 2006. The parameters used in this study, are used to validate the models. Statistical performance evaluation measures were analyzed for validation which includes RMSE, NNSE and Coefficient of Determination.

Kaya et al. (2018) [20] uses two different machine learning approaches, including ANN and M5 Model tree for the ground water level prediction purposes. This study was carried out in Reyhanli region of Turkey, where around 196 data instances from the year 2000-2015 were analyzed on these heuristic approaches individually. It was observed that both models perform almost same and are thus considered to be very close to each other.

Nourani et al. (2019) [21] proposed a study on the hybrid wavelet transform and M5 model tree for rainfall runoff modelling. The experiment was carried out in three data divisions where the training and test splits were (75% – 25%), (60%– 40%) and (50% – 50%) respectively on daily and monthly basis. The data was first decomposed into time series using wavelet transform and then the obtained series was applied to M5 model as its input. The results were observed and it concludes the Wavelet M5 (WM5) model outperforms ANN and M5 model individually. Thus, according to this study hybrid model (R2 = 0.80) is preferred over ANN model (R2 = 0.23) and M5 model (R2 = 0.19) individually.

Bahmani et al. (2020) [22] proposed a wavelet transform for ground water level simulation using GEP and M5 model tree. These models are combined with the wavelet transform producing the two hybrid models Wavelet GEP (WGEP) and WM5. In this study, various parameters from meteorological data were observed, which includes groundwater level, temperature and precipitation. This research was carried out in Iran region. The experimental results conclude that temperature and groundwater level are more effective than precipitation values and also both models perform same, but GEP is considered as more complex than M5 model.

Adnan et al. (2021) [23] proposed a comparative study for rainfall modelling using 4 machine learning

algorithms. This study mainly focuses on the capability of Optimally Pruned Extreme Learning Machine (OPELM), MARS, and M5 model tree in hourly rainfall-runoff modelling. It was observed that the accuracy increases considerably and the increments in the RMSE, MAE are generally more than 90%. Furthermore, MARS-K means outperforms other alternatives used in this study.
Based on the literature review discussed above, we can conclude that the applications of LMT have been used in various tasks and also implemented on many datasets over the years, but it has not become a yardstick algorithm in the world of machine learning and data science. LMT in this case provided better results and faster implementation. However, its performance across datasets needs to be (re)evaluated. Further not much has been written about LMT as other algorithms like decision trees, random forests are discussed in many approaches. Also, it has neither become a part of any python library or MATLAB.

3.Methods and dataset

3.1Model trees

A branch of machine learning which is concerned with the construction or amending the working models where various exemplary cases are taken into consideration like handling the missing values and the noise. Many models are used for handling such type of typical problems involve classification and, for these, learning algorithms that generate decision trees are proven to be more efficient, robust and less complex [24]. In such cases the data used for the classification and prediction are discrete and categorized.

On the other hand, when the data available are numeric and continuous, it requires the learned model for the prediction associated with a case rather than the class to which the case belongs. In these tasks the researchers mainly divide data into smaller groups and implement decision tree algorithms for a valuable prediction or to classify the data by building the classification models. Such type of approaches often fails in its implementation and complexity because building the decision trees models can’t make implicit ordering of these classes [24].

A novel approach has been introduced by Quinlan (1992) [24] to deal with the continuous class learning problems called as model trees. The main function of the model tree algorithm is to combine any traditional decision tree inducer with the likelihood of linear or logistic regression at the leaf nodes of the constructed decision trees. Since the decision tree seems to be clear approach where as in case of regression function only few variables are involved.

Model trees can predict a numeric value like an ordinary regression tree works, that is defined over a permanent number of numeric or nominal parameters, but distinct model trees construct a piecewise or Hamiltonian linear estimations to the target function [25]. Thus, the resultant model tree constructs a tree with the logistic or linear regression functions at the leaf nodes [26]. The principal advantage of using this machine learning methodology is that it acts as a white box learning model where each and every step is defined by the mathematical expression that shows the dependencies between the attributes. Furthermore, model trees can perform feature selection implicitly, data preparation becomes an easy process i.e., less effort needs to be taken while performing data preparation [27], and it can handle missing values also.

Model Trees are broadly classified into LMT where classification trees and logistic regression at the leaves are combined. This can be seen as a piecewise logistic model which can deal with the binary and multiclass target variables. Linear Model Trees where linear regression models and any decision tree inducer are combined to produce better predictions and it can lead to better insights than either of the model alone.

3.2 Logistic model trees

This section briefly describes the stepwise construction of the LMT algorithm from a decision tree inducer with the logistic regression at the leaves and thus defines the equivalent of model tees for classification approaches.

3.2.1The model

The LMT is basically a stepwise approach which consists of a decision tree inducer for the splitting criteria and logistic regression for the pruning in which standard regression errors are calculated at each node. Probably there are two splitting principles for Logistic Model Trees viz: 1. Using information gain for splitting like in case of C4.5 algorithm and this approach is taken as the default approach. 2. The second is to enhance the purity measure when fitting logistic regression functions [28]. These different approaches can differ in the tree structures only, whereas; the accuracy measure almost remains same in both approaches. The stepwise construction of the LMT is shown below.

Step 1: In section 1, some of the decision tree inducers are defined and out of which we have chosen C4.5 as the decision tree induction algorithm for the stepwise implementation of LMT. The reason of choosing C4.5 over ID3 as a decision tree inducer is that it can handle unknown values in the training records, can deal with categorical and continuous attributes and C4.5 algorithm can deal with the over fitting problem reflected in ID3 algorithm.

The C4.5 builds a decision tree based on top-down, recursive, and "divide and conquer" approach. It constructs the decision tree based on the information gain theory concept in which the splitting attribute which has the highest information gain ratio will be chosen as the splitting node parameter. The information gain can be defined as the reduction in the entropy after the dataset is divided on an attribute.

To calculate the information gain of an attribute a comparison of the entropy (Equation 2) of the dataset after and before a transformation needs to be done. The attribute with the highest information gain ratio will lead to the construction of a decision tree by acting as a splitting node with the homogenous branches.

$$Entropy(D) = -\sum_{i=1}^{m} p_i \log_2 p_i \quad (2)$$

Here, P_i is the probability of class i.

Here, in C4.5 we need to calculate the gain ratio's instead of gains. To calculate the gains, we can simply apply the below Equation 3.

$$GainRatio\ (D,S) = \frac{Gain\ (D,S)}{SplitInfo(D)} \quad (3)$$

$$GainRatio\ (D,S) = \frac{Gain\ (D,S)}{H(\frac{|D1|}{|D|},\ldots\frac{|Dn|}{|D|})} \quad (4)$$

Where, D denotes the attributes with distinct values for each record (D1, D2, D3,….Dn) with S number of subsets. It is a ratio of information gain for a splitting attribute (Equation 4) and entropy of an attribute split (ignoring classes).

C4.5 presents its results in a decision tree. When it splits an attribute into two classes, that attribute is at the top of the decision tree. The attribute is split into two numerical values. For example, it could split the attribute class intensity into less than or equal to 69 and greater than 69. The information that is under less than or equal to 69 will split with another attribute with two numerical values.

Step 2: In this step the regression at each leaf node is applied which can result in the pruning of the original decision tree inducer (C4.5). Pruning of trees at interior nodes are then replaced by the regression plane instead of a constant value which can also result in the rules generated by the LMT. This is usually done when the branches of the tree are not useful in the later stages. The main advantage of this step is that it reduces the level of complexity of the classifier without affecting the overall performance of the original tree. There are two primary strategies for pruning: 1) with reduced error pruning in which the most popular class replaces the nodes and starting at the leaves. This approach is used to simplify the data and increasing speed. 2), with Cost-Complexity Pruning (CCP) where it is used to define the cost-complexity measure [26]. This can be computed as Equation 5:

$$CCP = \frac{err(prune(T,t),S) - err(T,S)}{|leaves(T)| - |leaves(prune(T,t))|} \quad (5)$$

Where, $err(T,S)$ is the error rate of tree T over dataset *S*, and $(prune(T,t),S)$ is the tree obtained by pruning the sub trees *t* after the regression is applied on the tree T.

A LMT is a tree structure with internal nodes N and terminal node T. Let *S* be whole dataset containing all the attributes in the original data. A disjoint subset of the original dataset *S* into S_t , where each region is denoted by a terminal node in the tree as (Equation 6) [27]:

$$S = \bigcup_{t \in T} S_t, \quad S_t \cap S_{t'} = \phi \quad for\ t \neq t' \quad (6)$$

In case of logistic regression trees, the leaves $t \in T$ are associated with the logistic function say f_t instead of a class label. This function f_t takes a subset $V_t \subseteq V$ of all attributes present in the dataset and its models the class membership probabilities as equated below [28] (Equation 7):

$$\Pr(G = j \mid X = x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_K(x)}} \quad (7)$$

Thus, below equation defines the model for the whole LMT (Equation 8):

$$f(x) = \sum_{t \in T} f_1(x) \,.\, I(x \in S_t) \quad (8)$$

Where, $I(x \in S_t) = 1$ if $(x \in S_t)$ else 0.

Thus, above cases define the two individual steps of LMT construction which include logistic regression and ordinary decision tree inducer (C4.5). Here the first is an LMT, which is pruned back to the root nodes and tree in which V_t= ∅ for all values of t∈T.

Since, a lot of data is present in every field like academic data [29–33], agricultural data [34], weather data [35–38], cloud data [39] and other type

of data [40–47] etc. Here, in this study the dataset used in this operation was collected from 2012-2017 of Kashmir province. The data were collected from National Data Centre (NDC), Pune, Indian Meteorological Department (IMD), India. It is the chief organization responsible for weather-related observations, weather forecasting, etc. Indian Meteorological Department (IMD) is considered as one of the six provincial specified meteorological data centres of the world meteorological society. This data comprises of various parameters which takes a vital role in the prediction of rainfall. These parameters are taken from the three different zones of the Kashmir province, which include North Kashmir region (Gulmarg), South Kashmir region (Qazigund) and Central Kashmir region (Srinagar). These three different zones are geographically located at 34.05°N 74.38°E with an average elevation of 8,690 ft., 33.59°N 75.16°E with an average elevation of 5,480 ft. and 34.5°N 74.47°E with an average elevation of 5,200 ft. respectively.

Below table (*Table 2*) shows the description of the data obtained and the various parameters for all the zones of Kashmir province which are used for the prediction of rainfall. Station ID defined in *Table 2* has 3 values: 42026, 42044, 42027 and station name respectively in accordance with station ID defines the name of the station of which the data belongs to. Further other parameters like instances, Location, Type, Attributes collected and Measurement define the total size of the dataset, their geographic location, type of the data which has been used in this paper, and the details of the attributes on which the experiment has been implemented respectively.

After the data has been cleaned and pre-processed the final instances of the dataset contains around 5580 entries out of which there are 3910 entries in which there is NO rainfall and 1670 entries in which there is rainfall. The part of the final dataset is shown in *Table 3*. This dataset has 5 parameters: Maximum temperature, Minimum Temperature, Humidity @ 3PM, Humidity @ 12 AM and rainfall, out of which 4 independent attributes with continuous values are present and the target attribute with a discrete value (Y|N) is used for the prediction purposes.

Table 2 Dataset description of various parameters

S. No	Station number	Station name	Station location	Instances	Attributes collected	Attribute measurement	Attribute type
1	42026	South Kashmir Region	33.59°N 75.16°E	1804	Rainfall	Mm (Discrete Y|N)	Continuous
					Humidity@ 3 PM	Percentage of relative Humidity, %	
					Max_Temperature	°C	
					Min_Temperature	°C	
					Humidity@ 12 AM	Percentage of relative Humidity, %	
2	42027	North Kashmir Region	34.05°N 74.38°E	1912	Rainfall	Mm (Discrete Y|N)	Continuous
					Humidity@ 3 PM	Percentage of relative Humidity, %	
					Max_Temperature	°C	
					Min_Temperature	°C	
					Humidity@ 12 AM	Percentage of relative Humidity, %	
3	42044	Central Kashmir Region	34.5°N 74.47°E	1864	Rainfall	Mm (Discrete Y|N)	Continuous
					Humidity@ 3 PM	Percentage of relative Humidity, %	
					Max_Temperature	°C	
					Min_Temperature	°C	
					Humidity@ 12 AM	Percentage of relative Humidity, %	

Table 3 Cleaned & processed data of Kashmir province

Max_Temp	Min_Temp	Humidity@12	Humidity@3	Rainfall
7.5	1	81	99	**Y**
16.8	12.5	98	79	**Y**

Max_Temp	Min_Temp	Humidity@12	Humidity@3	Rainfall
16.5	11.2	96	66	**Y**
19.5	14	96	98	**N**
25.7	12.9	96	72	**Y**
30.7	15.8	86	98	**N**
15.5	12.8	94	88	**Y**
20.5	16.4	73	98	**Y**
19.6	13	91	73	**Y**
20	14.5	61	98	**Y**
13	4.6	68	64	**N**
23.2	15.8	68	98	**N**
15	9.2	66	62	**N**
18	9	73	98	**N**
25.3	16.5	83	65	**Y**
15	5.4	77	76	**Y**
21.5	15.8	76	67	**N**
32.5	19.1	72	98	**Y**
19	9.5	70	79	**Y**
30.2	18.3	65	98	**Y**

3.3Methods: building logistic model tree on meteorological data

Here, in this section we demonstrated the implementation of LMT on the historical dataset of Kashmir province for the future prediction of rainfall. As discussed in section 4 the construction of LMT is a stepwise process in which we grow the initial tree based on the decision tree inducer (C4.5) and then applying the logistic regression at the leaf nodes. The implementation process has been carried out using Google Colab where all the python code was implemented. It has a RAM capable of 13 GB which is termed out to be good for traditional algorithms and small datasets. The hardware on which implementation was carried includes a Central Processing Unit (CPU) model: Intel(R) Xeon(R) CPU @ 2.30GHz. This configuration may vary for different platforms and systems.

Figure 1 shows the comprehensive stepwise approach adopted in this study. The flow process starts with the data collection process, after that the data pre-processing and data cleaning has been done using various analysis tools like PCA. The next step is the data distribution where the data are divided into training and test sets. In this approach we have used (30-70) % for test and training of the model respectively. After that the C4.5 algorithm has been implemented and the accuracy statistics of the implemented algorithm has been tested. In the next step LMT algorithm was implemented and this process goes on as shown in the below work flow process model.

3.4Building initial tree: C4.5 implementation

This is the first step of the construction of model trees in which a straight forward approach is used which involves the construction of classification tree inducer. Here, we have used C4.5 algorithm as our inducer and in this approach, we have used the information gain as the splitting criteria in which the attribute with the highest information Gain value will be chosen as the splitting node. Below snippet shows the pseudo code implemented in python, which is used to generate sets of rules using C4.5 algorithm on a given set of data.

Input: Training data samples with continuous values.
Output: Complete C4.5 tree.

```
def C4.5 (data, target, attribute)
    create root_node N for the tree
    if (T belongs to same category C)
            leaf node = N
            Mark N as Class C
            Return N
    For i = 1 to n
           Calculate info_gain (Ai)
    ta ← Testing attribute
    N.ta ← Attribute having the highest info_gain
    If (N.ta ← continuous)
         Find threshold
    For (each T in splitting of T)
          If (T ← empty)
             Child of N is a leaf node
          Else
             Child of N ← dtree T
```

Calculate classification error rate of node N
Return N [36]

Figure 2 shows the implementation of C4.5 algorithm for the meteorological dataset of Kashmir province. Here, we have shown the portion of the tree where the attribute humid12 was chosen as the parent node based on the highest information gain among all the attributes. This process is a repetitive in nature until all the nodes are calculated and the resultant will be the leaf nodes with values Y or N.

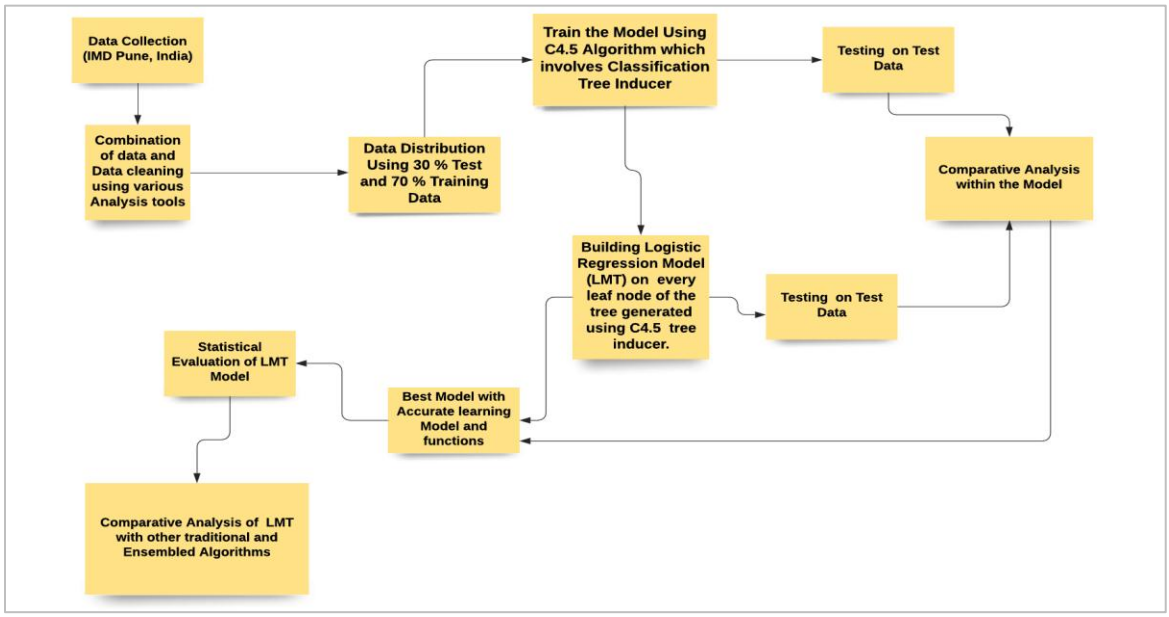

Figure 1 Flow of the proposed model

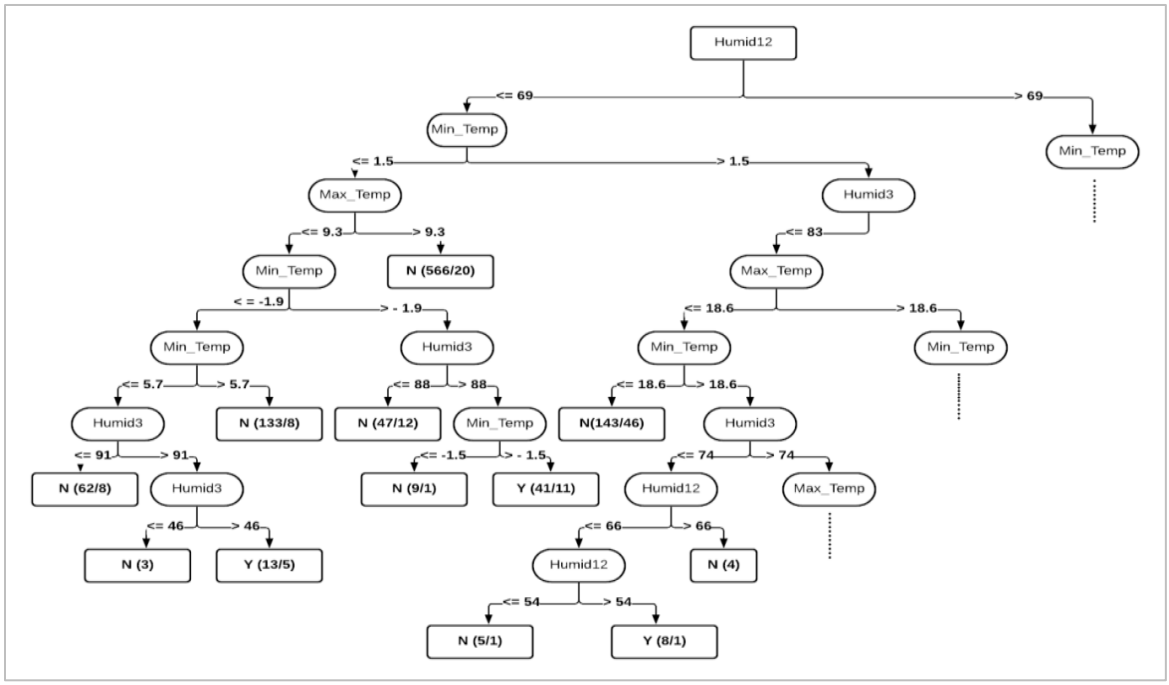

Figure 2 Implementation of C4.5 algorithm for the "Meteorological Data" of Kashmir Province

The snapshot of the portion of rules generated by the C4.5 algorithm is shown in *Figure 3*. These generated rules follow "IF-then ELSE" criteria.

In which all the attributes are processed and based on the highest information gain the node will be selected as the root node and the process continues and it results in the leaf node with values Y or N.

```
Humid12 <= 69
|   Min_Temp <= 1.5
|   |   Max-Temp <= 9.3
|   |   |   Min_Temp <= -1.9
|   |   |   |   Max-Temp <= 5.7
|   |   |   |   |   Humid3 <= 91: N (62.0/8.0)
|   |   |   |   |   Humid3 > 91
|   |   |   |   |   |   Humid12 <= 46: N (3.0)
|   |   |   |   |   |   Humid12 > 46: Y (13.0/5.0)
|   |   |   |   Max-Temp > 5.7: N (133.0/8.0)
|   |   |   Min_Temp > -1.9
|   |   |   |   Humid3 <= 88: N (47.0/12.0)
|   |   |   |   Humid3 > 88
|   |   |   |   |   Min_Temp <= -1.5: N (9.0/1.0)
|   |   |   |   |   Min_Temp > -1.5: Y (41.0/11.0)
|   |   Max-Temp > 9.3: N (566.0/20.0)
|   Min_Temp > 1.5
|   |   Humid3 <= 83
|   |   |   Max-Temp <= 18.6
|   |   |   |   Min_Temp <= 7.6: N (143.0/46.0)
|   |   |   |   Min_Temp > 7.6
|   |   |   |   |   Humid3 <= 74
|   |   |   |   |   |   Humid12 <= 66
|   |   |   |   |   |   |   Humid12 <= 54: N (5.0/1.0)
|   |   |   |   |   |   |   Humid12 > 54: Y (8.0/1.0)
|   |   |   |   |   |   Humid12 > 66: N (4.0)
|   |   |   |   |   Humid3 > 74
|   |   |   |   |   |   Max-Temp <= 18.5: Y (26.0/1.0)
|   |   |   |   |   |   Max-Temp > 18.5: N (3.0/1.0)
|   |   |   Max-Temp > 18.6
|   |   |   |   Min_Temp <= 7.3: N (135.0/2.0)
|   |   |   |   Min_Temp > 7.3
```

Figure 3 Rules generated by C4.5 Algorithm

Furthermore, after the construction of decision tree using C4.5 algorithm, the accuracy measure and the overall performance of the C4.5 was calculated shown in *Table 4*. The data were divided into 30-70 % as test and training set respectively, and it was observed that out of 1675 values only 1122 were correctly classified and on that basis other statistic measure were calculated.

Table 4 Accuracy statistics

Classifier (C4.5)	**Statistics**
Test Set	1675
Training Set	3905
Correctly Classified	1122
Wrongly Classified	553
Accuracy	66.99%
Error	33.01%
Cohen Kappa	-0.0225
Mean Absolute Error (MAE)	0.3391
RMSE	0.4421

3.5Building logistic regression model on C4.5 algorithm

After the construction of C4.5 decision tree algorithm on meteorological data of Kashmir province, a logistic regression model is implemented on each and every node of the decision tree because every node candidate leaf for pruning. In this approach, logistic regression models are built on each and every node of the tree and without taking the adjoining tree structure into consideration. We are assuming that we have split a node and at the child nodes we want to build a logistic regression function. But we have already applied a logistic regression function at the parent node so it has already become a base for fitting a logistic regression function of the child. Thus, after the splitting we are free to use LogitBoost iterations recursively on smaller sets for fitting the response variables at the child node only. So, using this approach will lead to the advantage of that it is computationally more efficient to build the logistic regression models at the child levels (lower levels) of the tree by extending the models which are already built at the parent level (higher level). This will lead to reduce the complexity of the model by building it from the mark.

The below snippet defines the pseudo code for the LMT algorithm. It contains a method called LOG_MODEL_TREE which is used to construct the tree based on the dataset DATA. This method calls a function get_CART which cross validates the CCP (Equation 5) for pruning which is implemented in C_prune.

```
LOG_MODEL_TREE (DATA) {
        Root = new Node ()
        x = get_CART (DATA)
        root.build_Tree (DATA, null)
        root.C_prune
}
Build_Tree (DATA, int_Linear_models) {
        numIterations = LB_Iterations (DATA,
            int_Linear_models)
        initLogitBoost (int_Linear_models)
        LinearModels = copyOf (int_LinearModels, DATA)
        for i = 1 to num_iterations
            LogitBoostIterations (linearModels, DATA)
        split = findSplit(DATA)
        localExamples = split.splitExamples (DATA)
        childs_nodes = new Nodes [split.numSubsets()]
        for s = i to childs_nodes.length
            childs_nodes.build_tree (localExamples [s],
            nodeModels)
}
LB_Iterations (DATA, int_Linear_models) {
        for fold = 1 to10
            initLogitBoost (int_Linear_models)
```

```
        train = trainCV (fold)
        linearModels = copyOf (int_Linear_models)
        for i = 1 to 100
            LogitBoostIterations (linearModels, train)
            logError [i] logErrors [i] + error (test)
    num_iterations = findBestIteration (logErrors)
    return num_iterations
}
```

The method build_Tree is used to build the LMT recursively and split the instance space. The parameter int_Linear_models contain the linear regression functions which are defined by the LogitBoost at the parent nodes (higher level) of the tree. Whereas, the method LB_Iterations defines the number of LogitBoost iterations to be performed. However, this snippet of the pseudo code doesn't define the stopping criteria, how to select the attribute while splitting and dealing with missing values.

Figure 4 shows the structure of the model tree found by LMT with only 10 rules (LM_1 to LM_10) as their leaves for the meteorological dataset of Kashmir province. Here, we have shown the full tree below where the attribute humid12 was chosen as the parent node based on the highest information gain among all the attributes. This process is a repetitive in nature until all the nodes are calculated and the resultant will be the leaf nodes with values Logistic functions at each leaf node of the generated tree.

Table 5 shows the both class models in LMT for geographical datasets of Kashmir province. These rules generated are the logistics rules which are divided into two classes based on the attribute values. Each rule (LM_1 to LM_10) has both class 0 and class 1 values separately defined in *Table 5*.

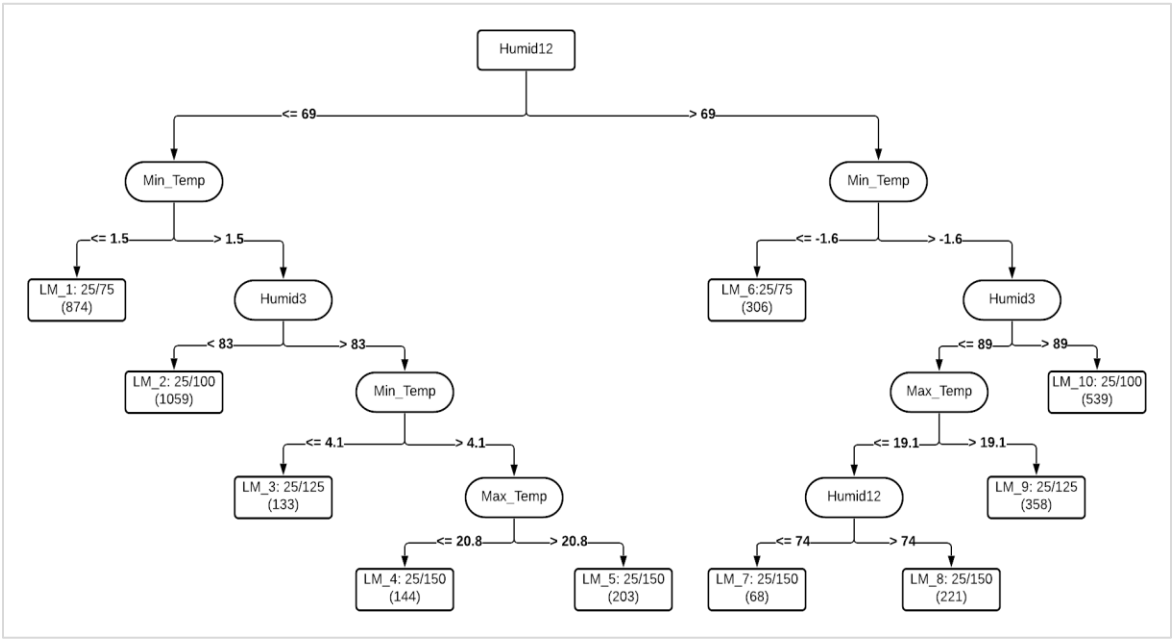

Figure 4 Logistic model tree for historical geographical data of Kashmir Province

Table 5 Models in logistic model tree for geographical data set of Kashmir Province

Model	Class 0	Class 1
LM_1:	2.25 + [Attribute_1] * 0.16 + [Attribute_2] * -0.2 + [Attribute_3] * 0 + [Attribute_4] * -0.03	-2.25 + [Attribute_1] * -0.16 + [Attribute_2] * 0.2 + [Attribute_3] * 0 +[Attribute_4] * 0.03
LM_2:	1.8 + [Attribute_1] * 0.11 + [Attribute_2] * -0.1 + [Attribute_3] * -0.01 + [Attribute_4] * -0.03	-1.8 + [Attribute_1] * -0.11 + [Attribute_2] * 0.1 + [Attribute_3] * 0.01 + [Attribute_4] * 0.03
LM_3:	2.46 + [Attribute_1] * 0.11 + [Attribute_2] * -0.12 + [Attribute_3] * 0.01 + [Attribute_4] * -0.04	-2.46 + [Attribute_1] * -0.11 + [Attribute_2] * 0.12 + [Attribute_3] * -0.01 + [Attribute_4] * 0.04
LM_4:	1.6 + [Attribute_1] * 0.06 + [Attribute_2] * -0.18 + [Attribute_3] * 0.02 + [Attribute_4] * -0.04	-1.6 + [Attribute_1] * -0.06 + [Attribute_2] * 0.18 + [Attribute_3] * -0.02 + [Attribute_4] * 0.04
LM_5:	6.42 + [Attribute_1] * 0.06 + [Attribute_2] * -0.05 + [Attribute_3] * 0 + [Attribute_4] * -0.08	-6.42 + [Attribute_1] * -0.06 + [Attribute_2] * 0.05 + [Attribute_3] * 0 + [Attribute_4] * 0.08

Model	Class 0	Class 1
LM_6:	2.93 + [Attribute_1] * 0.18 + [Attribute_2] * -0.18 + [Attribute_3] * -0.02 + [Attribute_4] * -0.03	-2.93 + [Attribute_1] * -0.18 + [Attribute_2] * 0.18 + [Attribute_3] * 0.02 + [Attribute_4] * 0.03
LM_7:	-0.58 + [Attribute_1] * 0.13 + [Attribute_2] * -0.18 + [Attribute_3] * 0.01 + [Attribute_4] * -0.01	0.58 + [Attribute_1] * -0.13 + [Attribute_2] * 0.18 + [Attribute_3] * -0.01 + [Attribute_4] * 0.01
LM_8:	0.04 + [Attribute_1] * 0.02 + [Attribute_2] * -0.02 + [Attribute_3] * 0 + [Attribute_4] * 0	-0.04 + [Attribute_1] * -0.02 + [Attribute_2] * 0.02 + [Attribute_3] * 0 + [Attribute_4] * 0
LM_9:	3.25 + [Attribute_1] * 0.03 + [Attribute_2] * -0.01 + [Attribute_3] * -0.01 + [Attribute_4] * -0.03	-3.25 + [Attribute_1] * -0.03 + [Attribute_2] * 0.01 + [Attribute_3] * 0.01 + [Attribute_4] * 0.03
LM_10:	6.21 + [Attribute_1] * 0.06 + [Attribute_2] * -0.08 + [Attribute_3] * -0.03 + [Attribute_4] * -0.05	-6.21 + [Attribute_1] * -0.06 + [Attribute_2] * 0.08 + [Attribute_3] * 0.03 + [Attribute_4] * 0.05
Where, Attribute_1 → Maximum Temperature Attrbiute_2 → Minimum Temperature Attribute_3 → Humidity at 12AM Attribute_4 → Humidity at 3PM		

The snapshot of the portion of rules generated by the LMT algorithm is shown in *Figure 5*. These generated rules follow "IF-then ELSE" criteria in which all the attributes are processed and based on the highest information gain the node will be selected as the root node and the process continues and it results in the leaf node with values Logistic model functions (LM-1 to LM_10).

```
Humid12 <= 69
|   Min_Temp <= 1.5: LM_1:25/75 (874)
|   Min_Temp > 1.5
|   |   Humid3 <= 83: LM_2:25/100 (1059)
|   |   Humid3 > 83
|   |   |   Min_Temp <= 4.1: LM_3:25/125 (133)
|   |   |   Min_Temp > 4.1
|   |   |   |   Max-Temp <= 20.8: LM_4:25/150 (144)
|   |   |   |   Max-Temp > 20.8: LM_5:25/150 (203)
Humid12 > 69
|   Min_Temp <= -1.6: LM_6:25/75 (306)
|   Min_Temp > -1.6
|   |   Humid3 <= 89
|   |   |   Max-Temp <= 19.1
|   |   |   |   Humid12 <= 74: LM_7:25/150 (68)
|   |   |   |   Humid12 > 74: LM_8:25/150 (221)
|   |   |   Max-Temp > 19.1: LM_9:25/125 (358)
|   |   Humid3 > 89: LM_10:25/100 (539)
```

Figure 5 Rules generated by LMT Algorithm

Furthermore, after the construction of LMT, the accuracy measure and the overall performance is calculated as shown in *Table 6*. The data here were divided into 30-70% as test and training set respectively, and it was observed that out of 1675 values only 1426 were correctly classified and on that basis other statistic measure were calculated.

Table 6 Accuracy statistics

Classifier (LMT)	Statistics
Test Set	1675
Training Set	3905
Correctly Classified	1426
Wrongly Classified	214
Accuracy	87.23%
Error	12.77%
Cohen Kappa	0.1024
Mean Absolute Error (MAE)	0.1728
RMSE	0.3092

4.Experimental results

In this paper, we applied step wise construction of novel LMT from the ordinary decision tree inducer to the logistic regression at each node of the tree. The experiment was performed on the meteorological data of Kashmir province, which contains 5 parameters including target parameter rainfall. This stepwise implementation was performed in python and its inbuilt libraries. The data observed of different variables has been taken for the period 2012-2017 (6 years). From the dataset 70% was used as a training data and 30% was used for testing purposes. Since each model takes four inputs (independent variables) and predicts one parameter at a time. The predicted results of LMT are shown in *Table 6*. Furthermore, *Table 5* shows the total number of rules generated by LMT on the geographical dataset of Kashmir province. The highest number of rules before fast regression was applied was around 15 and after the fast regression was applied the number of rules turn out to be 10 without affecting the overall performance of the algorithm.

4.1Result: implications and impact

Consequent upon implementation of LMT on historical geographical dataset, prediction is

remarkably increased from 81.07% obtained using Support Vector Machine (SVM) & Decision tree (ID3) on the said dataset, to 87.25% approximately. The execution performance of random forest on the CPU is somewhat higher than the execution performance using LMT. This highly puts an impact on the result generated when LMT was taken into consideration. However, this is subject to the execution of the specific dataset and it may vary on some other threshold datasets.

Furthermore, the mathematical complexity of the LMT is lesser as compared to ensembled random forest algorithm because in case of random forest algorithm the same dataset is horizontally divided into n number of decision trees, which adds to the complexity of the random forest algorithm.

4.2Comparative analysis

Since, based on the same set of data various traditional and ensemble algorithms were implemented before and the accurate measure of each algorithm was calculated [1]. *Table 7* gives the classification accuracy of various traditional & ensemble algorithms which are implemented on the same data which has been used in this paper, like: ID3, SVM, naïve Bayes, random forest, distributed decision tree (DDT), k-nearest neighbour (KNN), and fuzzy decision tree (FDT) algorithm.

4.3Calculation of error and other metrics

The various statistical measures on which the comparison has been made include: error, precision, recall, Cohen kappa, F-measure, etc.

Mathematically, these performance measures can be calculated using below formulas (Equation 9, 10 and 11):

$$\text{Error} = \frac{\text{Wrong Classified}}{\text{Test set}} \quad (9)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives+False Negatives}} \quad (10)$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives+False Positives}} \quad (11)$$

Where, wrongly classified denotes those values which do not match the values of the target attribute in the test set, and test set are those values on which the prediction accuracy is calculated. Where, true positive is an outcome where the model correctly predicts the positive class and in case of false positive the model incorrectly predicts positive class. False negative shows the miss rate.

Cohen Kappa

To calculate the Cohen Kappa, we need to follow the below steps:

Example:

Suppose there are 1786 values in the test set and out of which 1448 values are correctly classified and 338 values are wrongly classified.

Step 1: Calculate P_o

Again, let us say that 260 instances were rated YES by both & 1196 instances were rated NO by both. Accordingly, it is shown as Equation 12.

$$P_o = \frac{\text{Number in Agreement}}{\text{Total}} \quad (12)$$

P_o = (260+1196)/1786

P_o = 0.815

Step 2: The probability that both Data and Classifier would randomly say YES.

Data said YES to $\frac{518}{1786}$ instances = 0.2900

Classifier said YES to $\frac{332}{1786}$ instances=0.186

The total probability of both saying YES randomly = 0.2900×0.186=0.05394

Step 3: The probability that both Data and Classifier would randomly say NO.

Data said NO to $\frac{1268}{1786}$ instances = 0.71

Classifier said No to $\frac{332}{1786}$ instances = 0.814

The total probability of both saying NO randomly = 0.71×0.814=0.57794

Step 4: Calculate P_e by adding values from step 2 and 3 to get the overall probability that the data and classifier would randomly agree.

P_e =0.05394 + 0.57794

P_e =0.63188

Step 5: Calculate Cohen Kappa (Equation 13)

$$K = \frac{P_o - P_e}{1 - P_e} \quad (13)$$

$$K = \frac{0.8107 - 0.634274}{1 - 0.634274}$$

$$K = 0.482399392$$

We observed that LMT reaches the highest accuracy measure when other learners are taken into consideration. It significantly outperforms ID3, SVM Naïve Bayes, KNN, FDT (*Table 7*) etc.

On comparing these below implemented algorithms for the prediction of rainfall, LMT performs well because this model was able to predict the maximum portion of the data with higher accuracy, F-measure, and precision, recall and sensitivity values.

We can also confirm that the number of rules in LMT is only 10 whereas, in other learns the number of

rules ranges from 55 to 21. LMT produces significantly smaller sized trees as compared to other learners defined in *Figure 3*. It is because the leaves present in the LMTs have logistic regression functions.

Other methods are also being helpful but they need a large portion of training data to train in order to predict very small portion of test data.

Table 7 Comparative analysis of tree size and other performance measures for Geographical Data Set of Kashmir Province

Algorithm	(ID3)	SVM	Naïve Bayes	KNN	FDT	DDT	C4.5	LMT
Model	30-70	30-70	30-70	30-70	30-70	30-70	30-70	30-70
Accuracy	80.12	81.07	76.70	78.94	77.75	78.46	66.99	87.23
No. of Rules	51	---	---	---	---	21	55	10
Error	19.87	18.92	23.29	21.05	22.24	21.54	33.01	12.77
Precision	0.812	0.845	0.818	0.848	0.841	---	0.880	0.892
Recall	0.938	0.897	0.864	0.857	0.846	---	0.726	0.973
Cohen Kappa	0.456	0.519	0.411	0.485	0.458	---	-0.022	0.102
F-measure	0.87	0.871	0.84	0.853	0.844	---	0.796	0.931
Specify	0.938	0.897	0.864	0.857	0.846	---	0.238	0.098
Sensitivity	0.938	0.897	0.864	0.857	0.846	---	0.726	0.973

5.Discussion

Rainfall prediction has been done by numerous researchers across the globe providing varying results. Some recent contributions include various algorithms like the Non-Linear Autoregressive Models with Exogenous Inputs (NARX) model, and ANN approach-based models, and so on. It was concluded based on the implementation of the following algorithms which include decision trees, random forests, naïve Bayes, SVM, KNN, DDT. In order to further improvise the literature suggestions, Onyari and Ilunga [5], in his study concludes that LMT gives better results as compared to ANN-MLP. Accordingly, mathematical framework for its implementation in python on Google Colab was designed and implemented.

Consequent upon implementation, the performance of LMT increases to 87.23% from 66.9% as compared to the C4.5 algorithm and other above-mentioned approaches and thus implementation concluded that resolution stagnation is wrong and accordingly results were improvised in the case of LMT. The final accuracy statistics of the LMT on the geographical dataset produce astonishing results with the highest accuracy measure produced by any traditional and ensemble algorithm on the same set of data.

5.1Limitations

Since, we have mathematically implemented LMT on the geographical dataset for the prediction of rainfall and the results show the drastic improvisation in the performance on the same set of data using traditional and ensemble algorithms. However, this work did not carry out the effect of LMT on some other datasets. The results generated in this paper are necessarily limited and this does not include some other research tendencies. Therefore, it is highly recommended that the same implementation should be tested on different datasets. Furthermore, scientific results are not available where LMT has been used for prediction purposes so that we can go for the comparative evaluation of the LMT models. Also, one of the limitations of LMT in comparison to the neural network model is the LMT can perform axis-partitioning of the data, whereas, the neural network model can partition into any irregular regions of the data. This axis partitioning of the data may reduce the overall performance of the model tree. There are a number of decision functions in this model and parameters in LMT seem somewhat arbitrary. It is possible to use the Minimum Description Length (MDL) constructions for principal designs. A complete list of abbreviations is shown in *Appendix I*.

6.Conclusion and future strategies

In this study, we introduced the piecewise construction and implementation of novel LMT. The construction of LMT employs a step wise implementation in which a simple decision tree inducer (C4.5) was used to build the tree where information gain was taken as the splitting criteria. Afterwards logistic regression functions were applied to each node of the tree and well-known CCP was applied. Experimentally, LMT produces models that are more accurate than other learning models. The accurate measure of the traditional and ensemble models used in this study ranges between (66.99%–81.05%) on the historical geographical data of Kashmir province. However, after the implementation of LMT on the same set of data the accuracy measure surprisingly climbs up-to 87.23%, which is considered as the massive prediction accuracy gain and it leads to the increase in the performance results. Thus, both experimental and theoretical estimations of LMT in terms of the accuracy and interpretability would be helpful in practical applications and other bench marks datasets for prediction purposes.

Here, many areas are the hot spot for the future predictions:

1. Data splitting using model tree could expose the data sensitivity and there is always a trade-off between pruning factor and the size of the tree which needs to be further rectified.
2. The major drawback which needs to be investigated is that LMT has a high analytical complexity as compared to other decision tree inducers.
3. LMT uses a simple accusation global scheme for filling the missing values in the dataset; a more complex scheme for handling the missing values will be beneficial for improving the accuracy of the model.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Zaman M, Kaul S, Ahmed M. Analytical comparison between the information gain and gini index using historical geographical data. International Journal of Advanced Computer Science and Applications. 2020; 11(5):429-40.

[2] Zamani NW, Khairi SS. A comparative study on data mining techniques for rainfall prediction in Subang. In AIP conference proceedings 2018. AIP Publishing LLC.

[3] Fayaz SA, Zaman M, Butt MA. Knowledge discovery in geographical sciences—a systematic survey of various machine learning algorithms for rainfall prediction. In international conference on innovative computing and communications 2022 (pp. 593-608). Springer, Singapore.

[4] Barros RC, Ruiz DD, Basgalupp MP. Evolutionary model trees for handling continuous classes in machine learning. Information Sciences. 2011; 181(5):954-71.

[5] Onyari EK, Ilunga FM. Application of MLP neural network and M5P model tree in predicting streamflow: a case study of Luvuvhu catchment, South Africa. International Journal of Innovation, Management and Technology. 2013; 4(1):11-5.

[6] Malerba D, Esposito F, Ceci M, Appice A. Top-down induction of model trees with regression and splitting nodes. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004; 26(5):612-25.

[7] Holmes G, Hall M, Prank E. Generating rule sets from model trees. In Australasian joint conference on artificial intelligence 1999 (pp. 1-12). Springer, Berlin, Heidelberg.

[8] Rokach L, Maimon OZ. Data mining with decision trees: theory and applications. World Scientific; 2007.

[9] Ashraf M, Zaman M, Ahmed M. An intelligent prediction system for educational data mining based on ensemble and filtering approaches. Procedia Computer Science. 2020; 167:1471-83.

[10] Hassan M, Butt MA, Baba MZ. Logistic regression versus neural networks: the best accuracy in prediction of diabetes disease. Asian Journal of Computer Science and Technology. 2017; 6:33-42.

[11] Quinlan JR. Simplifying decision trees. International Journal of Man-Machine Studies. 1987; 27(3):221-34.

[12] Rokach L, Maimon O. Decision trees. In Data Mining and Knowledge Discovery Handbook 2005.

[13] Fayaz SA, Zaman M, Butt MA. Performance evaluation of GINI index and information gain criteria on geographical data: an empirical study based on JAVA and python. In international conference on innovative computing and communications 2022 (pp. 249-65). Springer, Singapore.

[14] Samadi M, Jabbari E, Azamathulla HM. Assessment of M5′ model tree and classification and regression trees for prediction of scour depth below free overfall spillways. Neural Computing and Applications. 2014; 24(2):357-66.

[15] Mahboobi E. The effect of sediment size on maximum scour depth in plunge pool (Unpublished Master's Thesis). University of Science and Technology, Tehran, Iran. 1997.

[16] Azar FA. Effect of sediment size distribution on scour downstream of free overfall Spillway(Unpublished Master's Thesis). Tarbiat Modares University, Tehran, Iran. 1998.

[17] Raza K. M5 model tree and gene expression programming for the prediction of metrological

parameters. In international conference on computers, communications, and systems 2015 (pp. 47-51). IEEE.
[18] Kisi O, Shiri J, Demir V. Hydrological time series forecasting using three different heuristic regression techniques. In Handbook of Neural Computation 2017. Academic Press.
[19] Rezaie-balf M, Naganna SR, Ghaemi A, Deka PC. Wavelet coupled MARS and M5 model tree approaches for groundwater level forecasting. Journal of Hydrology. 2017; 553:356-73.
[20] Kaya YZ, Üneş F, Demirci M, Taşar B, Varçin H. Groundwater level prediction using artificial neural network and M5 tree models. Air and Water. Environmental Components. 2018: 195-201.
[21] Nourani V, Davanlou TA, Molajou A, Gokcekus H. Hybrid wavelet-M5 model tree for rainfall-runoff modeling. Journal of Hydrologic Engineering. 2019; 24(5).
[22] Bahmani R, Solgi A, Ouarda TB. Groundwater level simulation using gene expression programming and M5 model tree combined with wavelet transform. Hydrological Sciences Journal. 2020; 65(8):1430-42.
[23] Adnan RM, Petroselli A, Heddam S, Santos CA, Kisi O. Comparison of different methodologies for rainfall–runoff modeling: machine learning vs conceptual approach. Natural Hazards. 2021; 105(3):2987-3011.
[24] Quinlan JR. Learning with continuous classes. In 5th Australian joint conference on artificial intelligence 1992 (pp. 343-8).
[25] Landwehr N, Hall M, Frank E. Logistic model trees. Machine Learning. 2005; 59(1-2):161-205.
[26] Frank E, Wang Y, Inglis S, Holmes G, Witten IH. Using model trees for classification. Machine Learning. 1998; 32(1):63-76.
[27] Mohd R, Butt MA, Baba MZ. Grey wolf-based linear regression model for rainfall prediction. International Journal of Information Technologies and Systems Approach. 2022; 15(1):1-8.
[28] Wang Y, Witten IH. Induction of model trees for predicting continuous classes. University of Waikato Research.1996.
[29] Altaf I, Butt MA, Zaman M. A pragmatic comparison of supervised machine learning classifiers for disease diagnosis. In third international conference on inventive research in computing applications 2021 (pp. 1515-20). IEEE.
[30] Zaman M, Butt MA. Information translation: a practitioners approach. In proceedings of the world congress on engineering and computer science 2012.
[31] Ashraf M, Zaman M, Ahmed M. To ameliorate classification accuracy using ensemble vote approach and base classifiers. In emerging technologies in data mining and information security 2019 (pp. 321-34). Springer, Singapore.
[32] Ashraf M, Zaman M, Ahmed M. Performance analysis and different subject combinations: an empirical and analytical discourse of educational data mining. In international conference on cloud computing, data science & engineering (confluence) 2018 (pp. 287-92). IEEE.
[33] Ashraf M, Zaman M, Ahmed M. Using ensemble stackingC method and base classifiers to ameliorate prediction accuracy of pedagogical data. Procedia Computer Science. 2018; 132:1021-40.
[34] Mohd R, Butt MA, Baba MZ. SALM-NARX: self adaptive LM-based NARX model for the prediction of rainfall. In international conference on I-SMAC (IoT in social mobile, analytics and cloud) 2018 (pp. 580-5). IEEE.
[35] Mohd R, Butt MA, Baba MZ. GWLM–NARX: grey wolf levenberg–marquardt-based neural network for rainfall prediction. Data Technologies and Applications. 2020; 54(1):85-102.
[36] Aljawarneh S, Yassein MB, Aljundi M. An enhanced J48 classification algorithm for the anomaly intrusion detection systems. Cluster Computing. 2019; 22(5):10549-65.
[37] Sidiq SJ, Zaman M, Ahmed M. How machine learning is redefining geographical science: a review of literature. Journal of Emerging Technologies and Innovative Research. 2019; 6(1):1731-46.
[38] Fayaz SA, Zaman M, Butt MA. To ameliorate classification accuracy using ensemble distributed decision tree (DDT) vote approach: an empirical discourse of geographical data mining. Procedia Computer Science. 2021; 184:935-40.
[39] Fayaz SA, Altaf I, Khan AN, Wani ZH. A possible solution to grid security issue using authentication: an overview. Journal of Web Engineering & Technology. 2019; 5(3):10-4.
[40] Zaman M, Quadri SM, Butt MA. Generic search optimization for heterogeneous data sources. International Journal of Computer Applications. 2012; 44(5):14-7.
[41] Zaman M, Butt MA. Enterprise data backup & recovery: a generic approach. International Organization of Scientific Research Journal of Engineering. 2013.
[42] Zaman M, Butt MA. Enterprise management information system: design & architecture. International Journal of Computational Engineering Research. 2013.
[43] Mohammad R, Ahmed MB, Zaman MB. Predictive analytics: an application perspective. International Journal of Computer Engineering and Applications. 2017; 11(8).
[44] Nayak D, Butt EM. Empowering cloud security through SLA. Journal of Global Research in Computer Science. 2013; 4(1):30-3.
[45] Hussain MW, Jamwal S, Zaman M. Congestion control techniques in a computer network: a survey. International Journal of Computer Applications. 2015; 111(2):7-10.
[46] Butt, EM, Quadri, SM, Zaman, EM. Star schema implementation for automation of examination records. In proceedings of the international conference on frontiers in education: computer science and computer engineering (FECS) 2012.

[47] Altaf I, Butt MA, Zaman M. Disease detection and prediction using the liver function test data: a review of machine learning algorithms. In international conference on innovative computing and communications 2022 (pp. 785-800). Springer, Singapore.

Sheikh Amir Fayaz is currently perusing his PhD from University of Kashmir, J&K. India. He has completed his PG (MCA) in the Department of Computer Science, University of Kashmir. His research area of interests are Data Mining and Machine Learning

Email: skh.amir88@gmail.com

Dr. Majid Zaman received the Ph.D. degree in 2009 from the University of Kashmir and is holding the position of Scientist in the department of Information Technology, University of Kashmir. J&K, India. His research area is Data Analytics, Data Mining, Data Science and Machine Learning.
Email: zamanmajid@gmail.com

Dr. Muheet Ahmed Butt received the Ph.D. degree in 2009 from the University of Kashmir and is holding the position of Scientist in the department of Computer Science, University of Kashmir. J&K, India. His research area is Data Analytics, Data Mining, Data Science and Machine Learning and Intrusion Detection.
Email: ermuheet@gmail.com

Appendix I

S. No.	Abbreviation	Description
1	ANN	Artificial Neural Network
2	ARIMA	Autoregressive Integrated Moving Average
3	BPNN	Back-propagation Neural Network
4	CART	Classification and Regression Trees
5	CCP	Cost-Complexity Pruning
6	CPU	Central Processing Unit
7	DDT	Distributed Decision Tree
8	FFNN	Feed Forward Neural Networks
9	GEP	Gene Expression Programming
10	Google Colab	Google Colaboratory
11	Humid3	Humidity Measure at 3 PM
12	Humid12	Humidity Measure at 12 AM
13	ID3	Iterative Dicotomizer 3
14	IMD	Indian Meteorological Department
15	KNN	K-Nearest Neighbour
16	LMT	Logistic Model Tree
17	LSSVR	Least Square Support Vector Regressor
18	MARS	Multivariate Adaptive Regression Splines
19	MATLAB	Matrix Laboratory
20	mRMR	Maximum Relevance Minimum Redundancy
21	MDL	Minimum Descriptive Length
22	MLP	Multilayer Perceptron
23	NARX	Non-Linear Autoregressive Models with Exogenous Inputs
24	NDC	National Data Centre
25	OPELM	Optimally Pruned Extreme Learning Machine
26	PMML	Predictive Model Markup Language
27	RF	Random Forest
28	RMSE	Root Mean Square Error
29	SVM	Support Vector Machine
30	WGEP	Wavelet GEP
31	WM5	Wavelet M5