Deep learning method Rainfall prediction for Kerala

A Project Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of

Bachelor of Technology

by

Vipin Kumar Seth
(111701030)

COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

CERTIFICATE

This is to certify that the work contained in the project entitled **Deep learning method Rainfall prediction for Kerala** is a bonafide work of **Vipin Kumar Seth** (**Roll No. 111701030**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Palakkad under my guidance and that it has not been submitted elsewhere for a degree.

Dr. Sahely Bhadra

Assistant Professor

Department of Computer Science & Engineering

Indian Institute of Technology Palakkad

Acknowledgements

I couldn't imagine anything better than to unequivocal my most profound thankfulness to all individuals who outfitted me the likelihood to complete this report. Exceptional appreciation to my undertaking mentor **Dr. Sahely Bhadra**, whose commitment in invigorating rules and consolation, helped me to facilitate my venture and also in scripting this report. At last, my profound and true appreciation to my family for their continuous and unmatched love, support and giving me an agreeable environment to complete my tasks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Problem definition and Motivation: Rainfall prediction stays a genuine concern and has pulled in the consideration of governments, businesses, hazard the board substances, just as established researchers. Rainfall is a climatic factor that influences numerous human exercises like rural creation, development, power age, ranger service and the travel industry, among others. To this degree, rainfall forecast is basic since this variable is the one with the most elevated connection with antagonistic characteristic occasions, for example, avalanches, flooding, mass developments and torrential slides. These episodes have influenced society for quite a long time. Consequently, having a proper methodology for rainfall prediction makes it conceivable to take preventive and moderation measures for these natural phenomena. In 2018-2019, Palakkad region of Kerala has experienced exceptionally high rainfall which has caused loss of human existence, harm to property, pulverization of harvests, loss of domesticated animals, and decay of medical issue attributable to waterborne infections. Thus, it is important to have a accurate prediction system for rainfall which can also meet the prediction of extreme events. This project deals with the prediction of amount of rainfall going to happens for the next seven days in Palakkad region of Kerala.

State of art: Normally, the weather and rainfall are presumed to possess non-linear and is elaborate phenomena. For predicting correct rainfall, we necessitate superior computer modeling and simulation. There has a great deal of exploration in this field and the current strategies utilize techniques dependent on numerical statistics or satellite image processing where they study the cloud images or using AI. The AI strategy is getting a ton of consideration and it is conservative and has incredible capacity. Heaps of precipitation forecast papers are there and they talk about the diverse prediction techniques. However, comes up short at whatever point the intricacy of the dataset which contains past precipitation increments and unfit to give a forecast of outrageous and uncommon occasions. Data-driven model forecasts utilizing deep learning algorithms are promising for these purposes.
This project fundamentally centers on the forecast of amount of precipitation in the Palakkad region of Kerala. The concentrate likewise lies in distinguishing extraordinary occasions as a piece of it. This report discusses the work done after mid-sem and the tentative arrangements.

1.1 Organization of The Report

This report has seven chapters: Chapter 1 is Introduction, Chapter 2 is a review of the current method begin used in rainfall prediction, Chapter 3 talks about the Data analysis, Chapter 4 talks about the implementation of a fuzzy paper, Chapter 5 talks about prediction using HMM model, Chapter 6 talks about the assessing the model on different stations data and Chapter 7 concludes the project and tells about the future path to be taken.

Chapter 2

Review of Current Method

The India Meteorological Department (IMD) is an organization of the Ministry of Earth Sciences of the Government of India. It is a significant organization responsible for meteorological perceptions, atmosphere assessing, and seismology. Right now, IMD is liable for the precipitation prediction in India. IMD chiefly utilizes a statistical model to anticipate the precipitation and they don't give a precise forecast of precipitation. Generally, they give a scope of the measure of precipitation or it utilizes the terms like large excess, excess, normal, deficient, large deficient to give the prediction of the precipitation. Further the statistical model also fails to catch the outrageous precipitation occasions.

2.1 Conclusion

Due to the vulnerability of disasters due to extreme rainfall events, it is a significant and moving errand to think of cutting edge anticipating which will give sufficient prediction of outrageous rainfall occasions notwithstanding the ordinary rainfall occasions. We have seen how classical models like ARIMA failed to handle the extremity of data hence demands a better model to unfold the underlying complexity involved in time series prediction.

Chapter 3

Data Analysis

In this chapter, the overall architecture includes four major components: Data Exploration and Pre-processing, Data Evaluation, Model Implementation, and Model Evaluation.

3.1 Data Exploration and Pre-processing

Exploratory Data Analysis is valuable to machine learning problems since it allows us to get closer to the certainty that the future results will be valid, correctly interpreted, and applicable to the desired contexts. Such a level of certainty can be achieved only after raw data is validated and checked for anomalies, ensuring that the data set was collected without errors. We have data to stations corresponding to 4 regions:

1. Malappuram which has 7 stations namely: Angadipuram, Manjeri, Nilambur, Perinthalamanna, Ponnani, Tiruvangadi, Palakkad.

2. Palakkad which has 9 stations namely: Alathur, Alattur, Mannarkad, Ottapalam, OBSY, Parli, Pattembi, Cherapalaseri, Chittur.

3. Thrissur has 8 stations: Chalakudi, Enamakkal, Kodungallur, Mukundarpuram, Ollukara, Peechi, Thalipilly, Thrissur.

4. Coimbatore has 10 stations namely: Anaimalai, Attakatti, Nirardam, Parambikulam, Pollachi, Sholiyarnagar, Solayar, Topslip, Nirar, Valparai.

Here we can think that why we require data from other regions when we are focusing on the Palakkad region because the climate of a region gets affected by the climate of its neighbors region. For the rest of this chapter, we have used the data of Palakkad-Alathur station. It has data of rainfall recorded each day from the year 1974 to the year 2014. We have cleaned the data and included only the years having all the 12 months. Fig. 3.1 is the instance of data where DF means daily rainfall.

1	YEAR	MN	DRF01	DRF02	DRF03	DRF04	DRF05	DRF06	DRF07	DRF08	...	DRF22	DRF23	DRF24	DRF25	DRF26	DRF27
0	ALATHUR	HYDRO	Distt.	:	PALAKKAD	CATCHMENT	No.	:	101	LATITUDE	...	NaN	NaN	NaN	NaN	NaN	NaN
1	YEAR	MN	DRF01	DRF02	DRF03	DRF04	DRF05	DRF06	DRF07	DRF08	...	DRF22	DRF23	DRF24	DRF25	DRF26	DRF27
2	1974	3	0	6	0	0	0	0	0	0	...	0	0	0	0	0	0
3	1974	4	0	0	0	0	0	0	0	0	...	1	0	0	0	0	0
4	1974	5	27	0	0	0	0	0	0	0	...	0	0	0	0	13	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
462	2015	8	0	0	0	14	1	9	0.3	22.7	...	3.7	0	1.5	0	0	0
463	2015	9	0	0	0	0.5	3.7	4.5	0.5	4.5	...	0	0	0	0	0	2.5
464	2015	10	7	1	0	3.5	5.5	3.3	8.4	0.5	...	0	0	0	0	0	5.5
465	2015	11	0.2	4	0	3.6	72.2	0	0.1	11.5	...	0	0	0.7	0	0	0
466	2015	12	0	0	0	0	0	0	0.1	0	...	0	0	0	0	0	0

Fig. 3.1 An instance of Data

We converted the data in the column fashion and removed the non relevent information for the time begin so that it become easy for data evaluation and for model implementation. See Fig 3.2.

3.2 Data Evaluation

Our data is Time Series data. Time Series(TS) is a collection of data points collected at constant time intervals. These are analyzed to determine the long term trend to forecast the future or perform some other form of analysis. But what makes a TS different from say a regular regression problem? There are 2 things:

1. It is time-dependent. So the basic assumption of a linear regression model that the observations are independent doesnt hold in this case.

	Date	RF
0	1975-01-01	0.0
1	1975-01-02	0.0
2	1975-01-03	0.0
3	1975-01-04	0.0
4	1975-01-05	0.0
...	...	...
13663	2012-05-29	0.0
13664	2012-05-30	0.0
13665	2012-05-31	0.0
13666	2012-06-01	0.0
13667	2012-06-02	0.0

Fig. 3.2 An instance of Data in column fashion

2. Along with an increasing or decreasing trend, most TS have some form of seasonality trends, i.e. variations specific to a particular time frame. For example, if you see the sales of a woolen jacket over time, you will invariably find higher sales in winter seasons.

Data stationarity is another important aspect of TS data. A TS is said to be stationary if its statistical properties such as mean, variance remain constant over time. But why is it important? Most of the TS models work on the assumption that the TS is stationary. Intuitively, we can say that if a TS has a particular behavior over time, there is a very high probability that it will follow the same in the future. Also, the theories related to stationary series are more mature and easier to implement as compared to non-stationary series. There are three basic criteria for a series to be classified as stationary series:

1. The mean of the series should not be a function of time rather should be a constant.

2. The variance of the series should not be a function of time.

3. The covariance of the ith term and the (i + m)th term should not be a function of

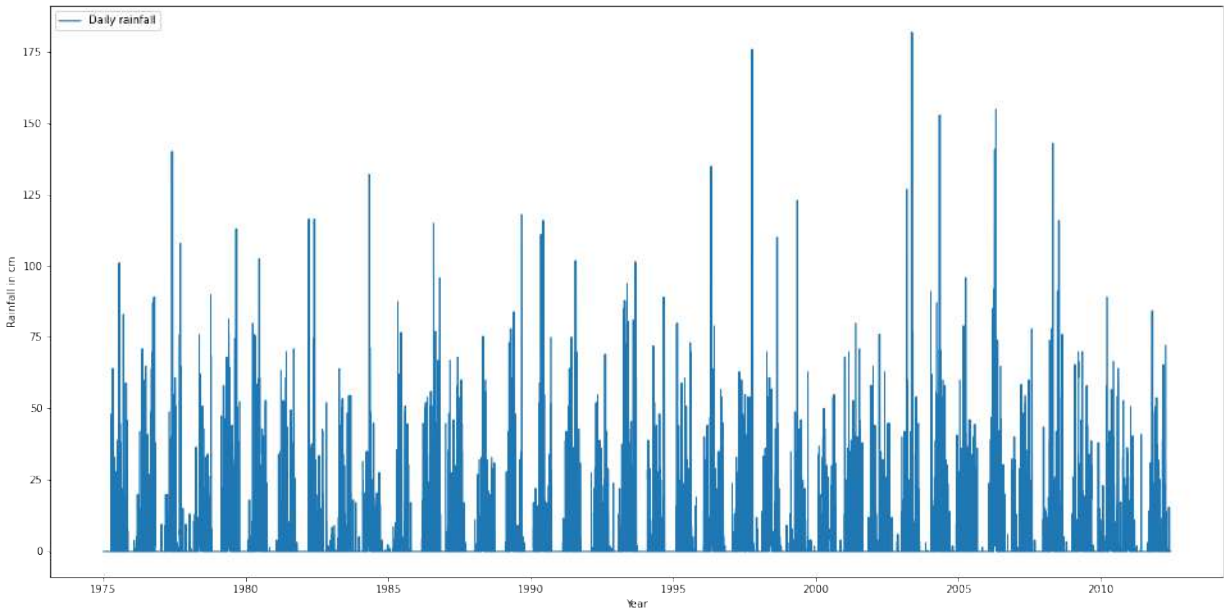


Fig. 3.3 Visualization of data

time.

Fig 3.3 give us the visual of data. Seeing the plot we can't make any decision on it's stationarity. So here, I will use two statistical tests would be used to check the stationarity of a data:

- Augmented Dickey-Fuller (ADF) Test: This is one of the most popular statistical tests. It can be used to determine the presence of unit root in the series, and hence help us understand if the series is stationary or not. The null and alternate hypothesis of this test is:

  – Null Hypothesis: The series has a unit root.

  – Alternate Hypothesis: The series has no unit root.

  If we fail to reject the null hypothesis, we can say that the series is non-stationary. This means that the series can be linear or difference stationary(I have talked about it in the next paragraph).

- Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test: KPSS is another test for checking the stationarity of a time series. The null and alternate hypotheses for the KPSS test are opposite that of the ADF test. The authors of the KPSS test have defined the null hypothesis as the process is trend stationary, to an alternate hypothesis of a unit root series.
  - Null Hypothesis: The process is trend stationary.
  - Alternate Hypothesis: The series has a unit root (series is not stationary).

Now,let's talk about types of stationarity. There are usually 3 types of stationarity:

- Strict Stationary: A strict stationary series satisfies the mathematical definition of a stationary process. For a strict stationary series, the mean, variance and covariance are not the function of time. The aim is to convert a non-stationary series into a strict stationary series for making predictions.
- Trend Stationary: A series that exhibits a trend is referred to as a trend stationary series. Once the trend is removed, the resulting series will be strict stationary. The KPSS test classifies a series as stationary on the absence of unit root. This means that the series can be strict stationary or trend stationary.
- Difference Stationary: A time series that can be made strict stationary by differencing falls under difference stationary. ADF test is also known as a difference stationarity test.

Results of the ADF test is the Table 3.1. The ADF tests give the following results test statistic, p-value, and the critical value at 1%, 5%, and 10% confidence intervals. The results of ADF on dataset is:

We interpret this result using the p-value from the test.

- p-value > 0.05: Fail to reject the null hypothesis (H0), the data has a unit root and is non-stationary.

Test Statistic:	-11.83828
p-value:	7.697380e-22
Lags Used:	37.00
Critical Value(1%):	-3.430830
Critical Value(5%):	-2.861752
Critical Value(10%):	-2.566883

Table 3.1 Result of ADF test

- p-value ≤ 0.05: Reject the null hypothesis (H0), the data does not have a unit root and is stationary.

So, here our data is stationary, and further, the critical values give us the cutoff for the confidence intervals of data. Here, Test statistic > critical value(1%), So, we can say that data has stationary in 99% confidence interval.

Test Statistic:	0.021673
p-value:	0.100000
Lags Used:	42.000000
Critical Value: (10%):	0.347000
Critical Value: (5%):	0.463000
Critical Value (2.5%):	0.574000
Critical Value: (1%):	0.739000

Table 3.2 Result of KPSS test

Result of KPSS test is the Table 3.2. We interpret the same way as ADF except the p-value. Here, for p-value > 0.05 is said to be stationary data. So, our data is stationary and also it lies in 99% confidence interval. So, combining both result, we can say the data is trend stationary and difference stationary.

3.3 Model Implementation

A popular and widely used statistical method for time series forecasting is the ARIMA model. An ARIMA model is a class of statistical models for analyzing and forecasting time series data. It explicitly caters to a suite of standard structures in time-series data, and as such provides a simple yet powerful method for making skillful time-series forecasts.

ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. It is a generalization of the simpler AutoRegressive Moving Average and adds the notion of integration.

This acronym is descriptive, capturing the key aspects of the model itself. Briefly, they are:

- AR: Autoregression. A model that uses the dependent relationship between an observation and some number of lagged observations.
- I: Integrated. The use of differencing of raw observations (e.g. subtracting an observation from observation at the previous time step) to make the time series stationary.
- MA: Moving Average. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

Each of these components are explicitly specified in the model as a parameter. A standard notation is used of ARIMA(p,d,q) where the parameters are substituted with integer values to quickly indicate the specific ARIMA model being used. The parameters of the ARIMA model are defined as follows:

- p: The number of lag observations included in the model, also called the lag order.
- d: The number of times that the raw observations are differenced, also called the degree of differencing.
- q: The size of the moving average window, also called the order of moving average.

Our ADF and KPSS test suggest us that we only require p and q for our data. So, we will keep d= 0. An importance concern here is how to determine the value of p and q. We use two plots to determine these numbers:

- Autocorrelation Function (ACF): It is a measure of the correlation between the the TS with a lagged version of itself. For instance, at lag 5, ACF would compare series

at time instant 't1'...'t2' with series at instant 't1-5' ... 't2-5' (t1-5 and t2 being endpoints).

- Partial Autocorrelation Function (PACF): This measures the correlation between the TS with a lagged version of itself but after eliminating the variations already explained by the intervening comparisons. Eg at lag 5, it will check the correlation but remove the effects already explained by lags 1 to 4.

See Fig. 3.4 for ACF plot. The 4 horizontal lines in the plot are 95% and 99% confidence band. Seeing the plot we can say that p = 66(first time the correlation comes in confidence of 99%).

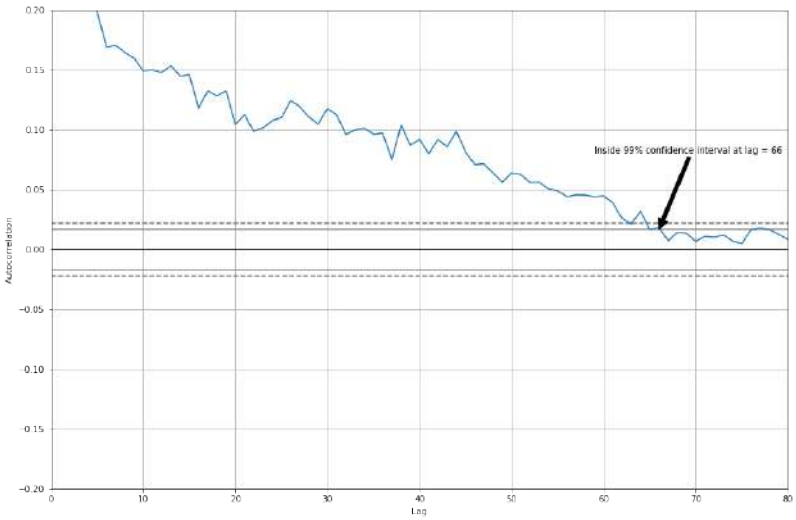


Fig. 3.4 Autocorrelation Plot

The Fig. 3.5 shows the PACF plot. The horizontal lines in the plot is 99% confidence band. Seeing the plot we can say that q = 16(first time the partial correlation comes in confidence of 99%).

Now that we have values of p & q in our hand we can implement the ARIMA model. Since the number of data samples is large and the expected lag(p = 68) is also quite high, it will require high computing power. So currently trained the model for p = 40 and q = 0. Fig 3.6 shows the output of the ARIMA model.

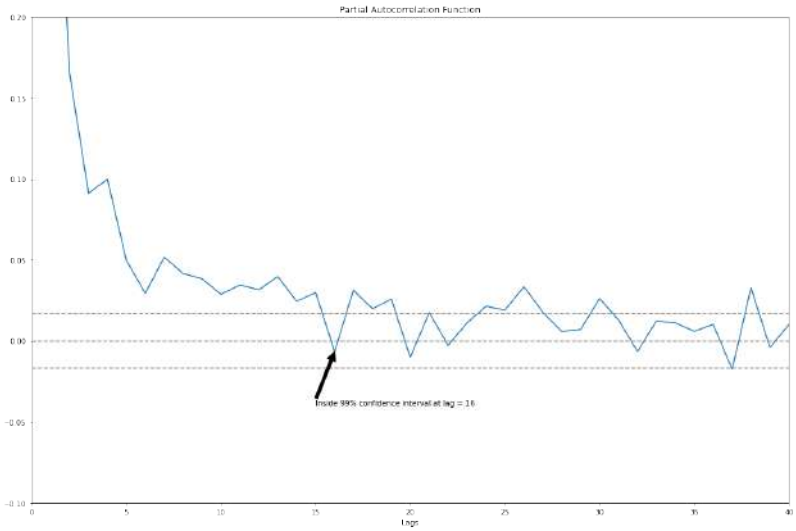


Fig. 3.5 Autocorrelation Plot

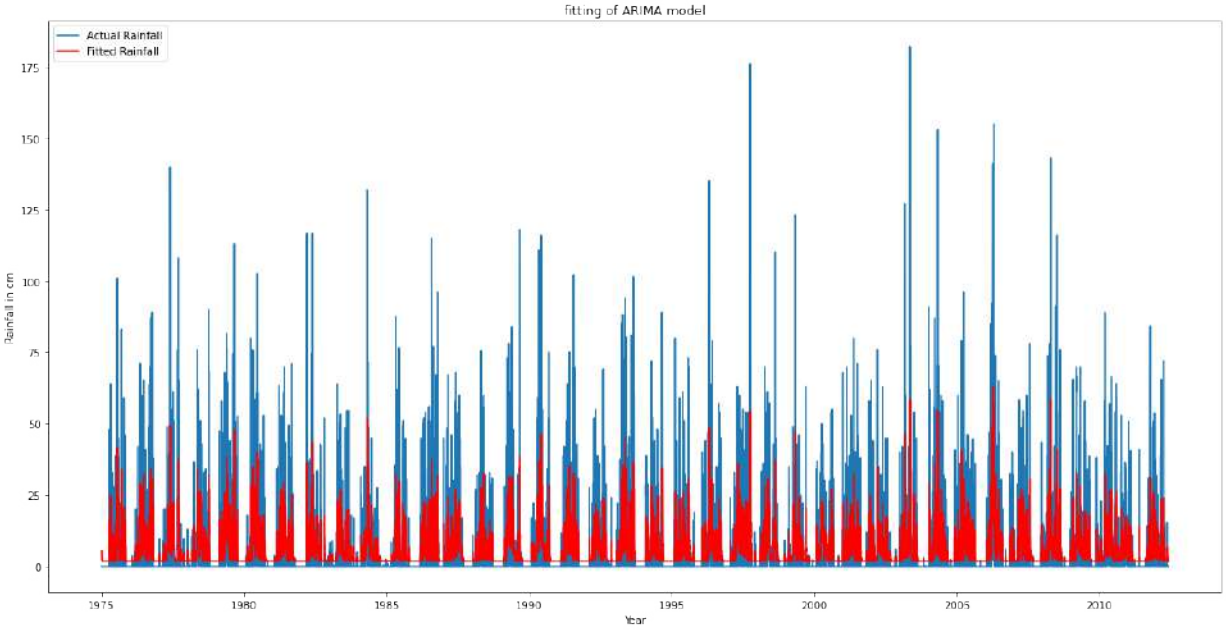


Fig. 3.6 Output of ARIMA model(p = 68, q = 0)

3.4 Model Evaluation

The ARIMA model has performed well so far. It was able to get the structure of data and was able to learn about the ups and downs in data. we also studied the error. Fig 3.6 shows the distribution of residual errors and Fig. 3.7 shows the density plot of residual errors.

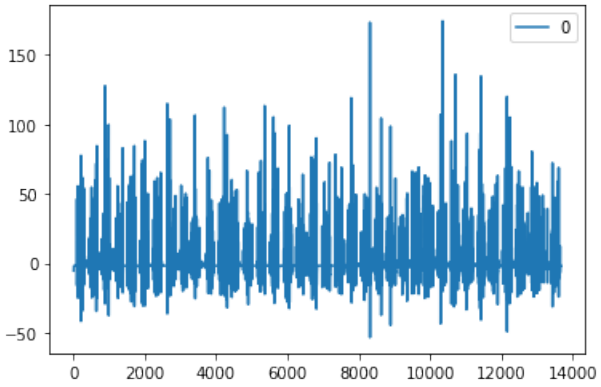


Fig. 3.7 Distribution of residual error

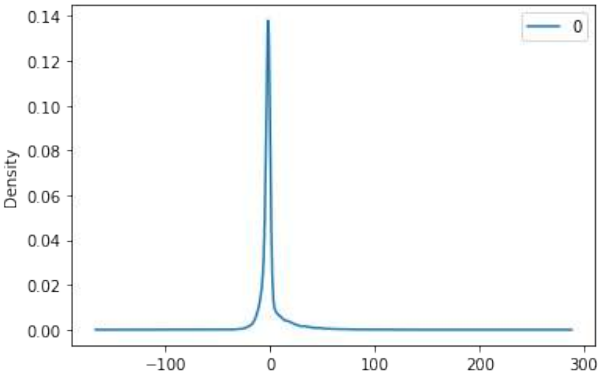


Fig. 3.8 Density plot of residual error

The Table 3.3 tells statistics of resduial errors and the root means square error for p = 40 is 1862184.1853

Seeing the plot and statistics we can say the following points:

- A good amount of the residual error are around 0. This is visible from density plot.This means the model has learned the value close to the actual value.
- The distribution of error and statistics tells that some of the residual error are very high in both direction which is not good.
- Number of positive residual error are more than negative residual errors.

Count:	13668.000
Mean:	0.0010500
Std:	11.739895
Min:	-53.262201
25th percentile:	-3.2410520
50th percentile:	-1.7146270
75th percentile:	-1.7146270
Max:	174.208874

Table 3.3 Statistics of residual error

3.5 Conclusion

In this chapter, we showed the insights of data, explained about Time Series, Stationarity and ARIMA. we checked the data for different types of stationarity and applied ARIMA model on it after doing correlation plot. I also talked about the residuals error of the model.

Chapter 4

Implementation of Paper

In the chapter, I will talk about the implementation of the paper titled: **Indian summer monsoon rainfall (ISMR) forecasting using time series data: A fuzzy-entropy-neuro based expert system**[1]. I selected this paper because of its result on test data which shows the capability of the model capturing the extreme events in the rainfall prediction very well.

4.1 Data Cleansing

Data Cleansing is an important task for identifying, correcting, or removing inaccurate and corrupt data or to fill the data with the missing points. I already mentioned the data in the previous report. I examined that data and found that some of the data points are missing. We have to fill up the missing data as we can't drop the data because it's a time series data and has to continuous. A good method is to fill the data seeing the neighboring values, but for the time being, I have currently filled all the missing values with 0. Since the paper talks about the monthly rainfall I summed the daily rainfall to the monthly rainfall. Fig 4.1 show the instance of the data. Fig 4.2 shows the monthly mean and standard deviation of the data which clearly shows the high mean in Jun to Oct month and also the high deviation.

	Year	Month	RF
0	1976	1	0.0
1	1976	2	0.0
2	1976	3	0.0
3	1976	4	200.0
4	1976	5	52.0
...	...	...	...
427	2011	8	405.1
428	2011	9	244.7
429	2011	10	177.1
430	2011	11	222.9
431	2011	12	17.7

432 rows × 3 columns

Fig. 4.1 Instance of the monthly rainfall data

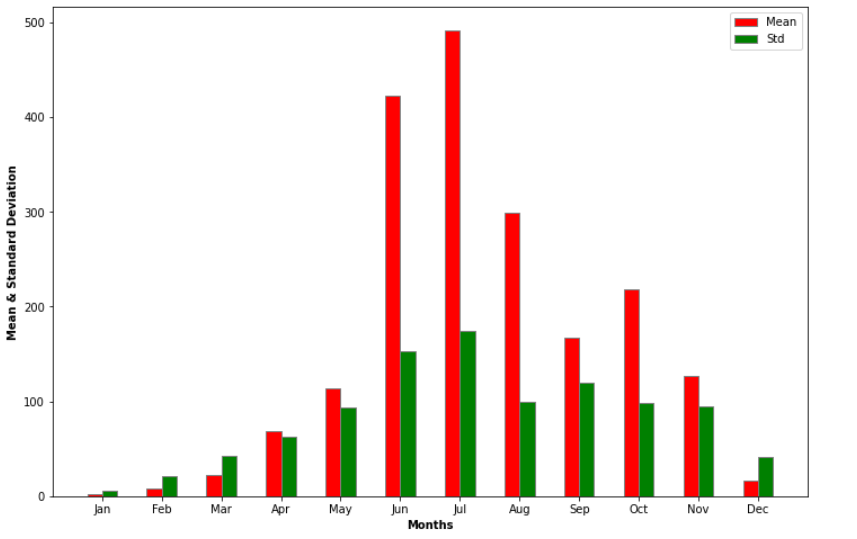


Fig. 4.2 Mean and standard deviation analysis of Monthly Rainfall time series data set for the period 1976 - 2011.

4.2 Backgrounds for study and the proposed model

The ideas of fuzzy set, entropy are given in the accompanying ensuing subsections to set up foundations for the study. In the resulting subsections, the architecture of the proposed model is additionally given followed by the model portrayal.

4.2.1 *Fuzzy set: an overview*

The classical set hypothesis is based on the major idea of "set" of which an individual is either a part or not a part. A sharp, fresh, and unambiguous differentiation exist between a member and a nonmember for any well-defined set of entities in this theory, and there is an exact and clear limit to demonstrate if an element has a place with the set. At the end of the day, when one poses the inquiry "Is this element an individual from that set?" The appropriate response is either "yes" or "no." This is valid for both the deterministic and the stochastic cases. In particular, in the classical set hypothesis, it isn't permitted that an element is in a set and not in the set simultaneously. Along these lines, some true application issues can't be portrayed and taken care of by the classical set theory, including every one of those including components with partial participation of a set. Unexpectedly, fuzzy set theory acknowledges partial enrollments, and, therefore, in a sense generalizes the classical set hypothesis. Hence, fuzzy sets are the sets with vague/imprecise boundaries. The capacity of fuzzy sets to communicate gradual advances from membership to non-membership and the other way around has a wide utility. It furnishes us not just with a significant and amazing portrayal of estimation uncertainties, yet additionally with an important portrayal of vague ideas communicated in natural language.

Definition 1. Universe of Discourse: Let L_{bd} and U_{bd} be the lower and upper bound of the time series data, respectively. Based on L_{bd} and U_{bd}, universe of discourse (U) can

be defined, as:

$$U = [L_{bd}, U_{bd}] \tag{4.1}$$

The fuzzy set can be defined for both continuous and discrete universe of discourse. Their definition are given below.[1]

Definition 2. Fuzzy set for the discrete case: A fuzzy set, $\tilde{A}_i$, in the discrete universe of discourse. U, can be represented as follows:

$$\tilde{A}_i = \mu(a_1)/a_1 + \mu(a_2)/a_2 + ... + \mu(a_n)/a_n \tag{4.2}$$

where μ is the membership function of the fuzzy set $\tilde{A}_i, \mu(a_k)$ represents the degree of membership of a_k associated to the fuzzy set $\tilde{A}_i, \mu(a_k) \in [0, 1]$ and $1 < k < n$. Here, the symbol "+" represents the operations of union instead of the operation of summation, and the symbol "/" represents the separator rather than the commonly used algebraic symbol of division.[1]

Definition 3. Fuzzy set for continuous case : A fuzzy set, $\tilde{A}_i$, in the continuous universe of discourse, U, can be represented as follows:

$$\tilde{A}_i = \left\{ \int \mu(a)/a \right\} \tag{4.3}$$

In this notation, integral sign represents the theoretic aggregation operator for continuous variables.[1]

Definition 4. Fuzzy representation of time series: Let $U = \{u_1, u_2, u_3, ...u_n\}$ be a universe of discourse of a particular time series data set. A time series value can be represented based on the categorize of information, as: "good", "moderate", and "low". Now, we can

define the fuzzy set. $\tilde{M}$, based on element of the universe of discourse, U, as:

$$\tilde{M} = \mu(m_1)/u_1 + \mu(m_2)/u_2 + ... + \mu(m_n)/u_n \tag{4.4}$$

here, $\mu(m_k)$ represents the degree of membership of any time series value, m_k, associated to the fuzzy set, $\tilde{M}$. Here, each $u_i(1, 2, ..., n)$ represents the interval, which can be obtained by the discretization of U.[1]

Definition 5. Fuzzy M-factors time series model: Let $\tilde{A}(t), \tilde{B}(t), \tilde{C}(t), ..., \tilde{M}(t)$ be the fuzzy representation of any time series factors: $A, B, C, ..., M$, respectively. If the forecasting problem of A is solved by using only $\tilde{A}(t)$, then it is called fuzzy $one-factors\ time\ series\ model$. If remaining fuzzy representations, such as $\tilde{B}(t), \tilde{C}(t), ..., \tilde{M}(t)$, are used with $\tilde{A}(t)$ to solve the forecasting problem, then it is called $fuzzy\ M-factors\ time\ series\ model$.[1]

Definition 6. Fuzzy M-factors Relationship, FMFR: Let $\tilde{A}(t), \tilde{B}(t), \tilde{C}(t), ..., \tilde{M}(t)$ be the fuzzy representation of any time series factors: $A, B, C, ..., M$, respectively. Assume that $\tilde{M}(t)$ is caused by $\tilde{A}(t-1), \tilde{B}(t-1), \tilde{(}C-1), ..., \tilde{L}(t-1)$, then FMFR can be expressed, as:

$$\tilde{A}(t-1), \tilde{B}(t-1), \tilde{(}C-1), ..., \tilde{L}(t-1) \rightarrow \tilde{M}(t) \tag{4.5}$$

here, left-hand size and right-hand side of FMFR can be referred as previous state and current state, respectively.[1]

Definition 7. Fuzzy Information - Gain (FIG) : It is the measure of uncertainty associated with the fuzzy set, $\tilde{A}_i$, for an event, that has occurred or will occur in terms of degree of membership. It is denoted as G_x. Mathematically, it can expressed, as:

$$G_x = -\sum_{i=1}^{n} \mu(\tilde{A}_i) log_2 \mu(\tilde{A}_i) \tag{4.6}$$

here, μ represents the degree of membership of the fuzzy set, $\tilde{A}_i$. The function, G_x, is known as the "FIG"[1].

4.2.2 *Architecture of the proposed model*

Each phase of the model is explained as below:

Phase 1. Prepare the rainfall time series data set.

Phase 2. Define the universe of discourse of the data set. Define $U = [\theta_{min} - \alpha_1, \theta_{max} + \alpha_2]$ where θ_{min} and θ_{max} are the minimum and the maximum values of monthly rainfall and α_1 and α_2 are two positive constants. From our data we have $\theta_{min} = 0.0$ and $\theta_{min} = 889.5$. Therefore, I let $\alpha_1 = 0.1$ and $\alpha_2 = 1.0$. Thus the universe of discourse turns out to be $U = [-0.1, 890.5]$.

Phase 3. Partition the U into various equal lengths of intervals. This is called discretization of universe of discourse. Let the intervals be $u_1, u_2, ..., u_n$ where each can defined as:

$$u_i = \left[Lower_U + (i-1)\frac{Upper_U - Lower_U}{n}, Lower_U + i\frac{Upper_U - Lower_U}{n} \right] \tag{4.7}$$

for $i = 1, 2, ..., n$. We know that $Lower_U = -0.1$ and $Upper_U = 890.5$ and k represents the number of intervals and I took n = 30. The table 4.1 shows the values of $u_1, u_2, ..., u_n$.

Phase 4. Filter out the intervals in which no data points(rainfall) falls. After doing this we are left with 28 intervals. Table 4.2 shows the filtered intervals.

Phase 5. Define fuzzy set for each of the intervals. We define fuzzy sets $\tilde{A}_1, \tilde{A}_2, ..., \tilde{A}_{28}$ on the universe of discourse, U.

Phase 6. Calculate the degree of membership value for each data point(rainfall). Here I have used triangular membership function. It depends on two scalar parameters, $lower(a_i)$ and $upper(b_i)$ value of the each intervals. It can be defined as follow:

$$\mu(x, a_i, b_i) = \frac{x - a_i}{b_i - a_i}; a_i \leq x \leq b_i \tag{4.8}$$

For example: A data point 19.6 lying in the fuzzy set $\tilde{A}_1$ (-0.1 , 29.59) will have the degree of membership as:

$$\mu(19.6, -0.1, 29.59) = \frac{19.6 - (-0.1)}{29.59 - (-0.1)}; -0.1 \leq 19.6 \leq 29.59 = 0.66$$

Phase 7. Determine the FIG values of each data points based on Eq 4.6. In Eq 4.6 the function G_x takes the input as degree of membership and gives FIG value as output. For example, FIG value for membership value 0.66 is

$$G_x = -0.66 \times log_2(0.66) = 0.40 \tag{4.9}$$

The summation in the Eq 4.6 can be taken out as the sets are disjoints, so a data point will lie just in one interval and will have membership value as 0 for interval it doesn't lie.

Phase 8. The FIG value will be input to the Artificial neural network.

Phase 9. Defuzzify the FMFRs. As stated in paper, below are stated steps for defuzzification process:

1. Establish the relationship of the FIG value as follow:

$$G_x(\tilde{A}_{i1}(m)), G_x(\tilde{A}_{i2}(m+1)), ..., G_x(\tilde{A}_{in}(m+k)) \rightarrow Desired_{outcome}\langle M, Y\rangle \tag{4.10}$$

2. In the paper k = 4 is being used and the FIG values are input to the Artificial neural network(ANN). The output of the ANN is the desired FIG value.

3. Let this desired FIG value be either completely or closely belongs to the fuzzy set, $\tilde{A}_j(m)$.

4. Find the interval, where the fuzzy set, $A_j(m)$, belongs to. Let this interval be: $u_j = [a_j, b_j]$. Here, a_j and b_j represents the lower and the upper bounds of the

intervals, respectively.

5. Find the degree of membership value for the fuzzy set, $A_j(m)$ $w.r.t$ the interval, u_j, as: $\mu(\tilde{A}_j(m))$.

6. Now, obtain the desired forecasted value, as:

$$F(t) = \mu(\tilde{A}_j(m))x(b_j - a_j) + a_j \tag{4.11}$$

4.2.3 *Implementation of model architecture on our data set*

I applied the preprocessing of the data needed before feeding to the model. The table 4.1 shows the intervals and the corresponding intervals. I filtered it and disposed of the intervals in which no data points lie. Table 4.2 shows the filtered intervals and the corresponding fuzzy set.

Intervals	Fuzzy sets
u_1 = (-0.1, 29.59)	$\tilde{A}_1$
u_2 = (29.59, 59.27)	$\tilde{A}_2$
u_3 = (59.27, 88.96)	$\tilde{A}_3$
u_4 = (88.69, 118.65)	$\tilde{A}_4$
u_5 = (118.65, 148.33)	$\tilde{A}_5$
...	...
u_{28} = (801.44, 831.13)	$\tilde{A}_{28}$
u_{29} = (831.13, 860.81)	$\tilde{A}_{29}$
u_{30} = (860.81, 890.5)	$\tilde{A}_{30}$

Table 4.1 Intervals and the corresponding fuzzy set

The subsequent stage is to find the degree of membership of the data point in the interval they belong to. Since the intervals are disjoint, one data point will lie only in one interval. Fig 4.3 shows the membership value of data point for the interval they belong to. Now we find the FIG value using Eq 4.6. I calculated the FIG values of each data point which can be seen in Fig. 4.4. Next, I trained an ARIMA model on the FIG values as the input. The paper utilizes the ANN to train on FIG value. Here I trained the FIG values on

Intervals	Fuzzy sets
u_1 = (-0.1, 29.59)	$\tilde{A}_1$
u_2 = (29.59, 59.28)	$\tilde{A}_2$
u_3 = (59.28, 88.96)	$\tilde{A}_3$
u_4 = (88.96, 118.65)	$\tilde{A}_4$
u_5 =(118.65, 148.33)	$\tilde{A}_5$
u_6 = (148.33 178.02)	$\tilde{A}_6$
u_7 = (178.02, 207.71)	$\tilde{A}_7$
u_8 = (207.71, 237.40)	$\tilde{A}_8$
u_9 = (237.40, 267.08)	$\tilde{A}_9$
u_{10} = (267.08, 296.77)	$\tilde{A}_{10}$
u_{11} = (296.77, 326.45)	$\tilde{A}_{11}$
u_{12} = (326.45, 356.14)	$\tilde{A}_{12}$
u_{13} = (356.14, 385.83)	$\tilde{A}_{13}$
u_{14} = (385.83, 415.51)	$\tilde{A}_{14}$
u_{15} = (415.51, 445.20)	$\tilde{A}_{15}$
u_{16} = (445.20, 474.89)	$\tilde{A}_{16}$
u_{17} = (474.89, 504.57)	$\tilde{A}_{17}$
u_{18} = (504.57, 534.26)	$\tilde{A}_{18}$
u_{19} = (534.26, 563.95)	$\tilde{A}_{19}$
u_{20} = (563.95, 593.63)	$\tilde{A}_{20}$
u_{21} = (593.63, 623.32)	$\tilde{A}_{21}$
u_{22} = (623.32, 653.01)	$\tilde{A}_{22}$
u_{23} = (653.01, 682.70)	$\tilde{A}_{23}$
u_{24} = (682.70, 712.38)	$\tilde{A}_{24}$
u_{25} = (712.38, 742.07)	$\tilde{A}_{25}$
u_{26} = (742.07, 771.75)	$\tilde{A}_{26}$
u_{27} = (801.44, 831.13)	$\tilde{A}_{27}$
u_{28} = (860.8, 890.5)	$\tilde{A}_{28}$

Table 4.2 Filtered Intervals and corresponding fuzzy set

	Year	Month	RF	DOM
0	1976	1	0.0	0.003369
1	1976	2	0.0	0.003369
2	1976	3	0.0	0.003369
3	1976	4	200.0	0.740400
4	1976	5	52.0	0.754997
...	...	...	...	...
427	2011	8	405.1	0.649225
428	2011	9	244.7	0.246126
429	2011	10	177.1	0.969010
430	2011	11	222.9	0.511790
431	2011	12	17.7	0.599596

Fig. 4.3 Data point and their Degree of membership

	Year	Month	RF	DOM	FIG
0	1976	1	0.0	0.003369	0.027668
1	1976	2	0.0	0.003369	0.027668
2	1976	3	0.0	0.003369	0.027668
3	1976	4	200.0	0.740400	0.321055
4	1976	5	52.0	0.754997	0.306119
...	...	...	...	...	...
427	2011	8	405.1	0.649225	0.404603
428	2011	9	244.7	0.246126	0.497798
429	2011	10	177.1	0.969010	0.044010
430	2011	11	222.9	0.511790	0.494582
431	2011	12	17.7	0.599596	0.442464

Fig. 4.4 Data point and corresponding FIG value

ANN and ARIMA model. ARIMA model has given preferable outcomes over ANN. The following is its implementation.

Implementation of ARIMA model on FIG value: I investigated the FIG and plotted it. Fig 4.5 shows it's plot. Decomposition gives a valuable dynamic model to considering time series generally and for better understanding issues during time series analysis and foreseeing. Time series decomposition includes considering the series as a blend of level, trend, seasonality, and clamor segments. I decomposed the FIG information and Fig 4.6 shows the decomposition. In that, we can see that the data has a trend part for a short span at regular intervals. I cross confirmed it by plotting the moving mean and standard deviation of data which can be found in Fig 4.7. It can appear to be that the moving mean (additionally called as moving mean) isn't consistent which asserts the presence of a trend in the data. To eliminate the trend I used the differencing strategy. Differencing is performed by deducting the previous perception from the current perception. Fig 4.8 shows the data subsequent to differencing. It clearly shows that the mean has gotten close to steady.

Next, I plotted the partial autocorrelation plot of the differenced data to find the ideal number of lag to be used. Fig 4.9 demonstrates the ideal number lag to be 8 (First time the value lying inside the concealed region). I trained the ARIMA model on differenced information with lag = 8. Fig 4.10 shows the fitted FIG value. The fit is decent as it can catch the limit in data and can anticipate abrupt sharp turns in the data. Our subsequent stage is to go from FIG value to rainfall data, however, I confronted an issue which I have discussed in the accompanying section.[2]

4.2.4 *Problems with the defuzzification process*

As mentioned in the paper about the defuzzification process, the value we get from the model will the desired FIG value. Now the real task lies in going from FIG value to the rainfall prediction.

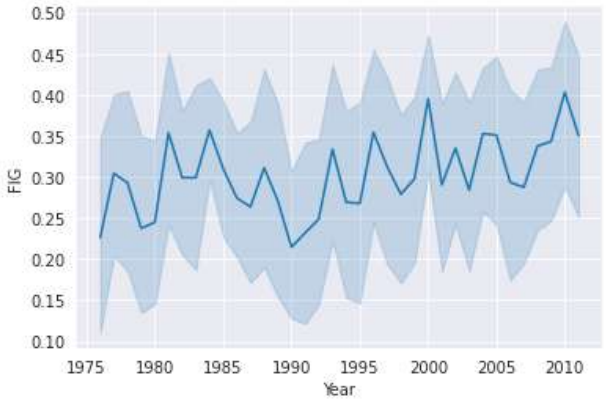


Fig. 4.5 Plot of the FIG value

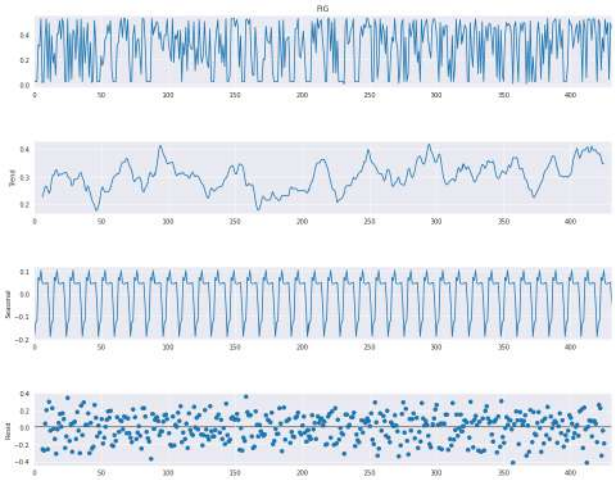


Fig. 4.6 Decomposition of FIG value

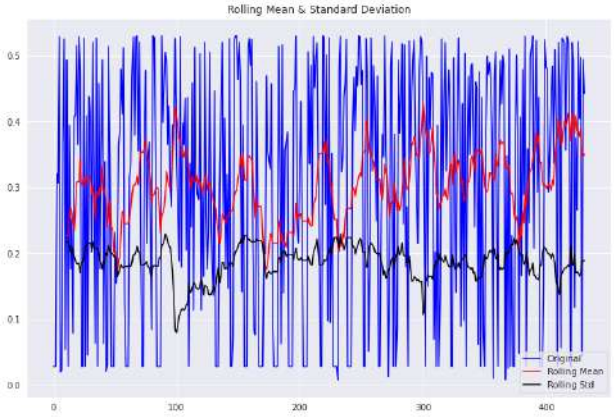


Fig. 4.7 Rolling mean and standard deviation

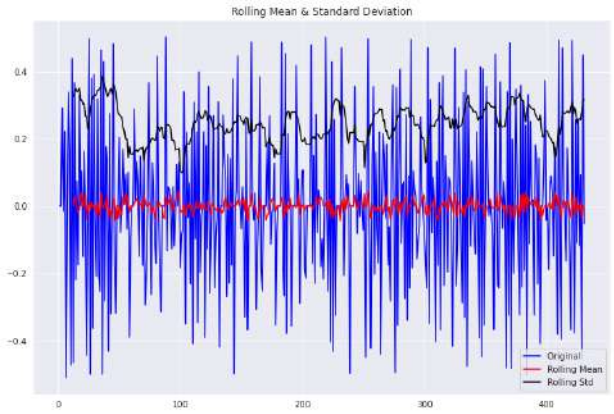


Fig. 4.8 Rolling mean and standard deviation after differencing

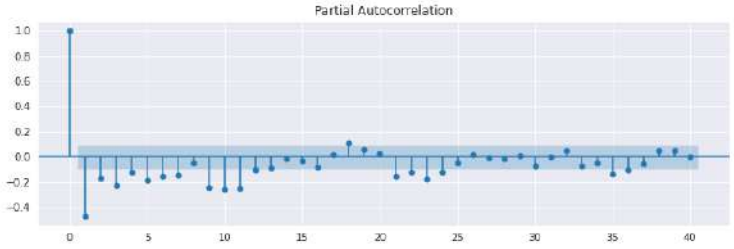


Fig. 4.9 Partial autocorrelation plot

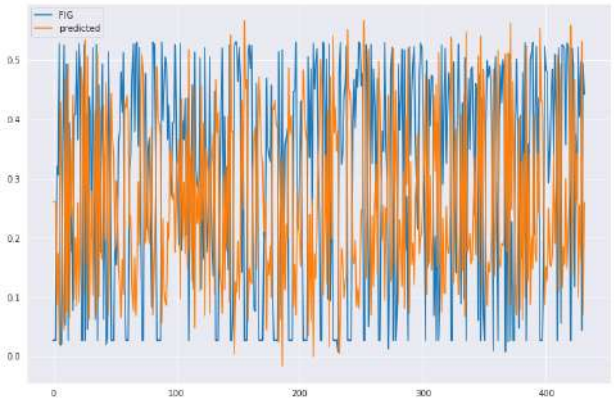


Fig. 4.10 Fitted FIG on the ARIMA model

It is the reverse process of as we went from rainfall to FIG. The paper has not explained the process very thoroughly rather has explained it briefly. Now, going from FIG to the degree of membership is not an easy task. Fig 4.11 show the plot of the Eq 4.6. First and foremost thing the function is not one to one function that for a given FIG value there will be two membership value. I tried to find a closed-form solution of the Eq 4.6(solving $\mu(\tilde{A}_i)$ in terms of G_x). After struggling a lot and failing to get a solution, I tried the approximation method by approximating the $log_2(x)$ using the Taylor series. The resulting equation is:

$$y = \frac{-x}{ln2}\left(ln(0.5) + \frac{x - 0.5}{0.5}\right) \tag{4.12}$$

Solving the above equation for x we will get

$$x = (0.423287 - (3.27608 \times 10^{-13} \sqrt{(16694000676541669163663336 - 32291368543925277409935 36 \cdot y)})) \tag{4.13}$$

Plot of the Eq 4.13 is shown in the Fig 4.12. The dotted line show the Eq 4.6 while the solid line shows the Eq 4.13 which is the approximation. We can see that the Eq 4.13 has did a good approximation of the first half of the curve. We only want to approximate the first half of the curve because the second half of the curve is a reflection of the first half about the line $x = 0.423287$.

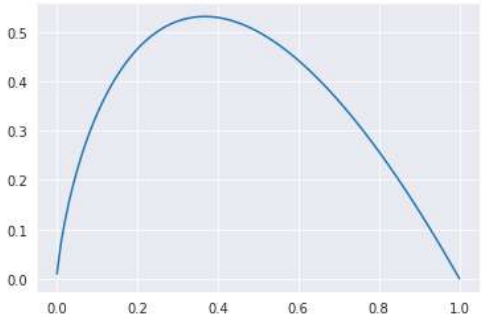


Fig. 4.11 Plot of Eq 4.6

The approximation has diminished our one obstacle of going from the FIG value to

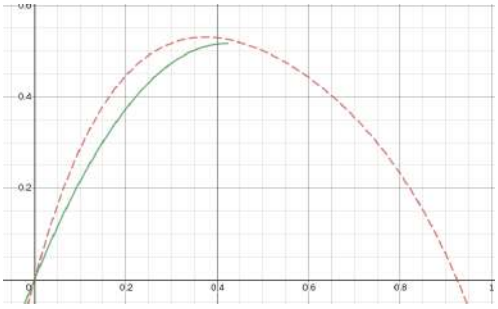


Fig. 4.12 Plot of Eq 4.13

membership value yet we don't have a clue about this membership value is for which fuzzy set. It will be that fuzzy where our anticipated rainfall will lie and we don't have this information in our grasp. Even I tried to capture the FIG to rainfall conversion through a neural network but failing due to not able to define a differentiable loss function.

4.3 Conclusion

In this part, I talked altogether about the execution of a paper on our dataset. We effectively pre-processed the data, characterized the intervals and relating fuzzy sets, and has converted rainfall over to the corresponding FIG value. We prepared an ARIMA model in the wake of analyzing the FIG value and decomposing it. The ARIMA model has demonstrated a decent execution and has caught the extremity in FIG value. We stalled out in changing over from FIG to rainfall, anyway has decreased one stage by approximating the FIG to membership value work by utilizing the approximation technique.

Chapter 5

Forecasting Using Hidden Markov Model

A Hidden Markov Model (HMM) is a finite state machine which has some fixed number of states. It provides a probabilistic framework for modelling a time series of multivariate observations. Hidden Markov models were introduced in the beginning of the 1970s as a tool in speech recognition [3]. This model based on statistical methods has become increasingly popular in the last several years due to its strong mathematical structure and theoretical basis for use in a wide range of applications. In recent years researchers proposed HMM as a classifier or predictor for speech signal recognition DNA sequence analysis, hand-written characters recognition, natural language domains etc. It is clear that HMM is a very powerful tool for various applications.

5.1 The Hidden Markov Model

Hidden Markov Model is characterized by the following:

1. Number of states in the model

2. Number of observation symbols

3. State transition probabilities

4. Observation emission probability distribution that characterizes each state

5. Initial state distribution

Below are some notation for HMM:

N = number of states in the model.

M = number of distinct observation symbols per state (observation symbols correspond to the physical output of the system being modelled).

T = length of observation sequence.

O = observation sequence, i.e., $O_1, O_2, O_3, ...O_T$.

Q = state sequence $q_1, q_2, ..., q_T$ in the Markov model.

A = a_{ij} transition matrix, where a_{ij} represents the transition probability from state i to state j.

B = $b_j(O_t)$ observation emission matrix, where $b_j(O_t)$ represent the probability of observing O_t at state j.

$\pi = \pi_i$ the prior probability, where π_i represent the probability of being in state i at the beginning of the experiment, i.e., at time t = 1.

λ = (A,B,π) the overall HMM model.

To work with HMM, the following three fundamental questions should be resolved:

1. Given the model λ = (A,B,π) how do we compute $P(O|\lambda)$, the probability of occurrence of the observation sequence $O = O_1, O_2, ..., O_T$.

2. Given the observation sequence O and a model λ ,how do we choose a state sequence $q_1, q_2, ..., q_T$ that best explains the observations.

3. Given the observation sequence O and a space of models found by varying the model parameters A,B and π, how do we find the model that best explains the observed data.

There are established algorithms to solve the above questions. In our task we have used the forward-backward algorithm to compute the $P(O|\lambda)$, Viterbi algorithm to resolve problem 2, and Baum-Welch algorithm to train the HMM. The details of these algorithms are given in the tutorial by Rabiner [3]

5.2 HMM Log Likelihood Similarity Forecasting

In this section we develop an HMM based tool for time series forecasting, for instance for the rainfall forecasting. While implementing the HMM, the choice of the model, choice of the number of states and observation symbol (continuous or discrete or multi-mixture) become a tedious task. For instance we have used left-right HMM with 6 states. In our problem, for simplicity, we consider 1 input features for a data that is the amount of total rainfall received in the current week. The next week's rainfall amount is taken as the target associated with the input feature. Our observations here being continuous rather than discrete, we choose empirically as many as 4 mixtures for each state for the model density.

In the experiment, our objective was to predict the next weeks rainfall amount using aforementioned HMM model. The idea behind our new approach in using HMM is that of using the training dataset for estimating the parameter set (A, B,π) of the HMM. Using the trained HMM, likelihood value for current weeks dataset is calculated. For example, say the likelihood value for the day is θ, then from the past dataset using the HMM we locate the instance that would produce the same θ or nearest to the θ likelihood value. That is we locate the past week where the rainfall behaviour is similar to that of the current week. **Assuming that the next weeks rainfall should follow about the same past data pattern**, from the located past week we simply calculate the difference of that weeks rainfall amount and next to that weeks rainfall amount. Thus the next weeks rainfall amount forecast is established by adding the above difference to the current weeks rainfall amount[4].

5.3 Experimentation: Training and Testing

In the earlier chapter we used rainfall data from 1975 to 2010 and further many data points were missing. In this chapter we used the data with no missing values and have the daily rainfall data from 1st Jan 1970 to 30th Apr 2018 (2521 weeks). We aggregated the rainfall on the weekly basis. I rounded the data for weekly basis and divided the dataset into two sets(80% and 20%), one training set (1st Jan 1970 to 20 Aug 2008 : 2016 weeks) and one test set (21 Aug 2008 to 25 Apr 2018 : 505 weeks). Next, I trained the HMM model with 6 states and 4 Gaussian mixture components. Fig. 5.1 shows the training of the HMM model. Here we can see that the log likelihood of the model increases (as expected) and start oscillating. Hence I trained for 14 iterations i.e up to the increase of the log likelihood. After the model got trained, I used the test data for prediction. Fig 5.2 shows the weekly rainfall prediction done by HMM model. Here we can see that the model has given a good prediction and it is able to capture the nuances in the data. We can see the good fitting on test data but the model has failed to capture some spikes and also has givens false spikes. The performance of the HMM model is assessed by means of correlation coefficient (CC) of the actual and forecasted values, root mean square error (RMSE) and mean absolute error (MAE) which is shown in table 5.1. The parameters CC, RMSE and MAE can be defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(A_i - P_i)^2}$$

$$MAE = \frac{1}{N} \cdot |(A_i - P_i)|$$

$$CC = \frac{n\sum(A_i P_i) - (\sum A_i)(\sum P_i)}{\sqrt{n(\sum A_i^2) - (\sum A_i)^2} \cdot \sqrt{n(\sum P_i^2) - (\sum P_i)^2}}$$

where P_i and A_i are the predicted and actual rainfall data for week i respectively, and N is the total number of weeks. The value of CC is such that $-1 < CC < +1$. For a good forecasting, correlation coefficient (CC) value greater than 0.8 is generally considered as good forecasting accuracy and a small RMSE and MAE value indicates good forecasting.

RMSE	61.78
MAE	32.99
CC	0.55

Table 5.1 Performance Metrics of HMM model on test data [2008 - 2018]

Further assessing the model, I divided the rainfall into 6 different continuous intervals namely $[0, 80), [80, 160), [160, 240), [240, 320), [320, 400), [400 - \infty)$ and then calculated the confusion matrix. Fig. 5.2 shows the same. Here we can see that we have good amount of prediction in interval 0-80 and for other intervals since the instances are less hence we see lesser number corresponding to it.

5.4 Testing on 2018, 2019, 2020 Year Rainfall Data

We all know about the flood in Kerala in Year 2018. It has occurred due to heavy rainfall. The real testing of model would be able to predict the high rainfall in year 2018 and 2019 and moderate rainfall in 2020. We have daily rainfall data from 01-Jun to 30-Sep of each year. We calculated the weekly data and did the prediction using HMM model. Fig [5.3, 5.4, 5.5] shows the corresponding prediction. Here we can say that the model has shown not so great performance. It has failed to captures the spikes in the data and further it has shown false spikes. This can be seen in all year i.e 2018, 2019 and in 2020.

Year	RMSE	MAE	CC
2018	85.72	65.03	0.81
2019	120.70	93.00	0.53
2020	84.74	62.21	0.51

Table 5.2 Performance Metrics of HMM model on 2018, 2019, 2020 rainfall

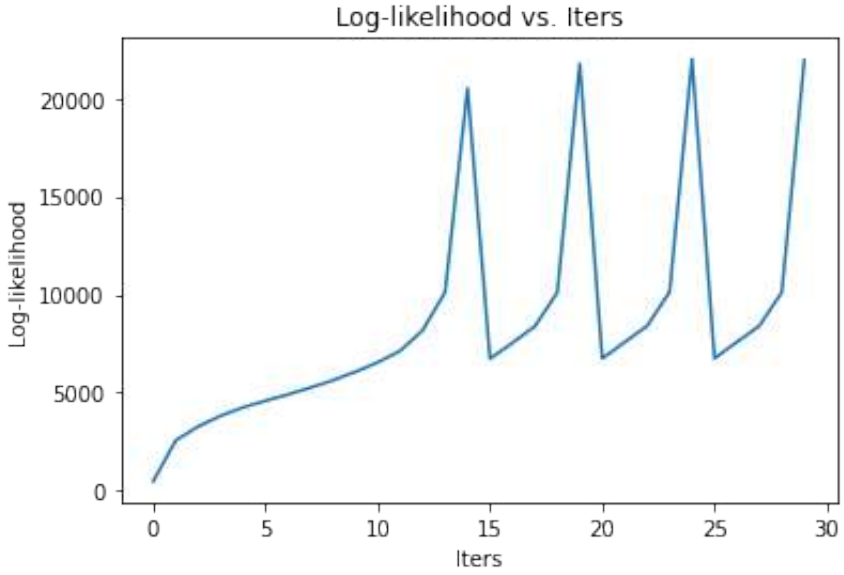


Fig. 5.1 Convergence of model

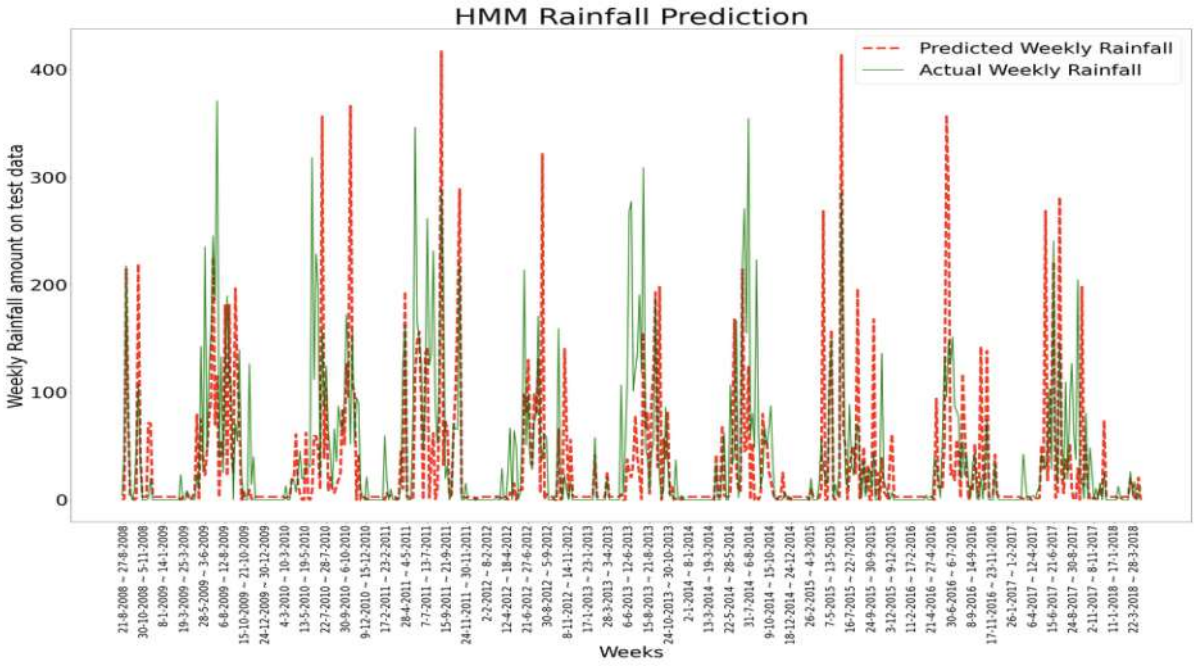


Fig. 5.2 Rainfall Prediction by HMM Model

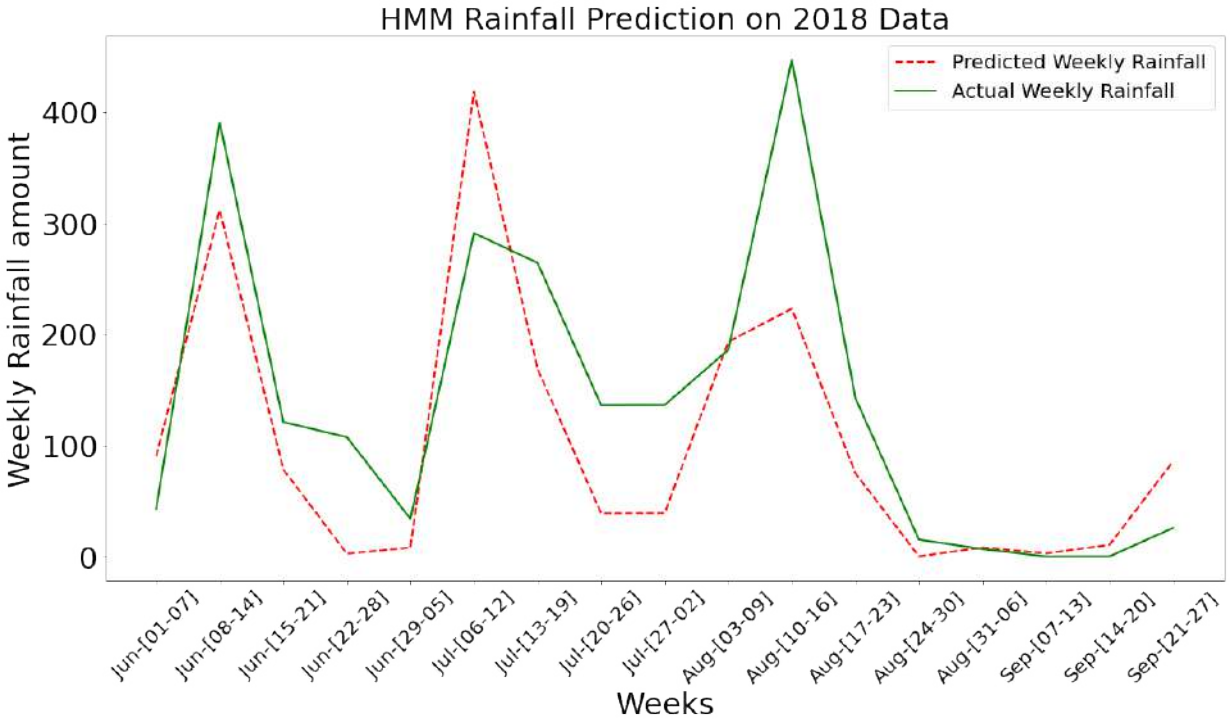


Fig. 5.3 HMM prediction on 2018 weekly rainfall data

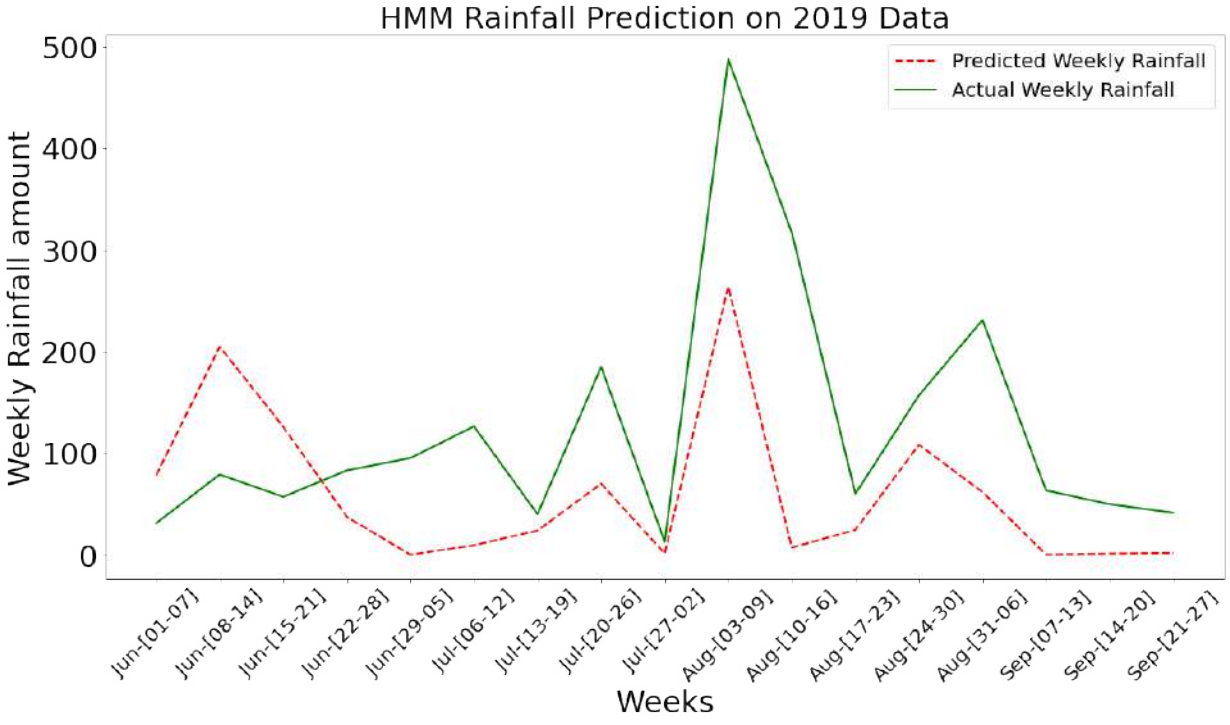


Fig. 5.4 HMM prediction on 2019 weekly rainfall data

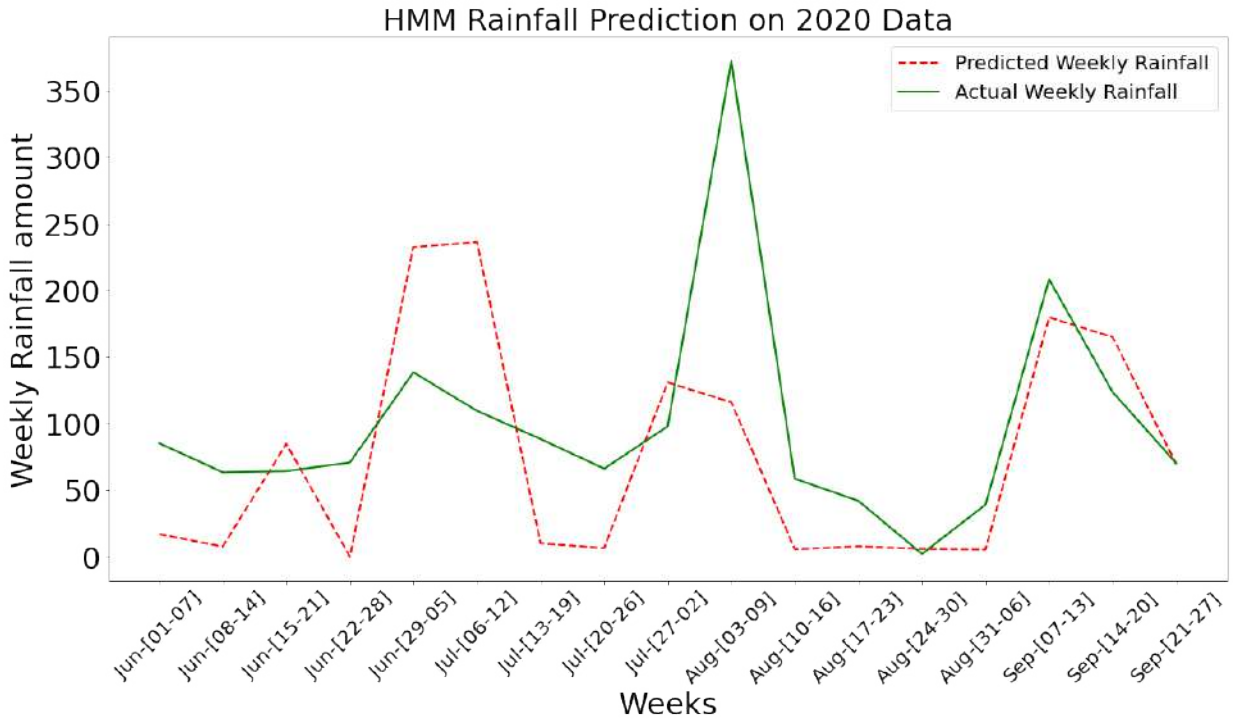


Fig. 5.5 HMM prediction on 2020 weekly rainfall data

5.5 Modification in the current HMM model

We have seen the above HMM model and it's performance. The model has shown a good capacity for rainfall prediction. However, the current model take only one experience from the past. A quick generalization will be to give the model more experience from the past. The likelihood similarity between two instance actually tells us the resembles of features between them. Hence, instead of going for only one past likelihood we will go use more than (let's say K). And we will do the difference of past those day's weekly rainfall (having likelihood similar to current) and next to those day's. Next we will take weightage average of these differences and weightage of a difference will be inverse of difference of likelihood values hence more the difference lesser will be weightage. The detailed algorithm is explained below:

1. Let's say the current likelihood is θ.

2. Choose top K likelihood values $\theta_1, \theta_2, ..., \theta_K$ from past which are similar to θ i.e

$\theta - \theta_1 < \theta - \theta_2 < ... < \theta - \theta_K < \theta - \theta_x$ where θ_x is general likelihood value $\notin \theta_1, ..., \theta_K$.

3. Now take difference of these weeks rainfall amount and next to these weeks rainfall amount. Let's say these differences are $d_1, d_2, ..., d_K$. Also calculate the inverse of difference of likelihood i.e $\beta_i = 1/(|\theta - \theta_i|)$.

4. Calculate the weightage of the i_{th} difference as $w_i = (\beta_i)/\sum \beta_j$.

5. Hence the final difference is $D = w_1 * d_1 + ... + w_K * d_K$.

6. Add this difference to the current rainfall to get the next week's rainfall.

Here K is known as Experience factor.

5.5.1 Finding a good Experience factor K

In order to find a good K, I took the help of performance metrics. I plotted RMSE, MAE and CC against K. Fig 5.6 shows the corresponding plots. The plots very clearly shows that as we increase the K, RMSE and MAE decreases while CC increases. Earlier the change is rapid but slows down as K increases. Finding a good experience factor using RMSE or MAE is not a easy task. But we can have a look of CC and can have cutoff of good CC. Here I am keeping cutoff 0.80 (i.e 80%) which occurs at around K = 300. Hence a good choice of experience factor will be 300. Fig 5.7 shows the prediction of the model.

5.6 Testing on 2018, 2019, 2020 Rainfall Data

Now we have seen a good performance on the test data. It time check it's performance on 2018, 2019 and 2020 data. Fig [5.8, 5.9, 5.10]. Here we can see that the model has shown a really good performance and is able to capture the spikes (extremity) in the data. The performance metrices shown in table also depicts the same scenario.

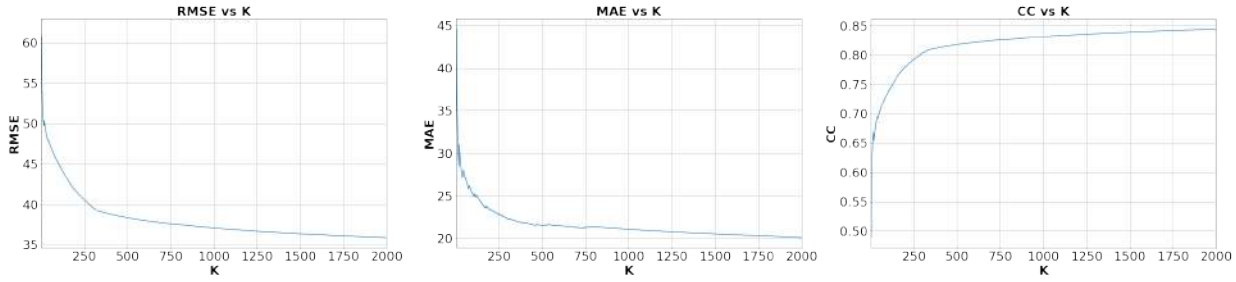


Fig. 5.6 Performance metrices vs K

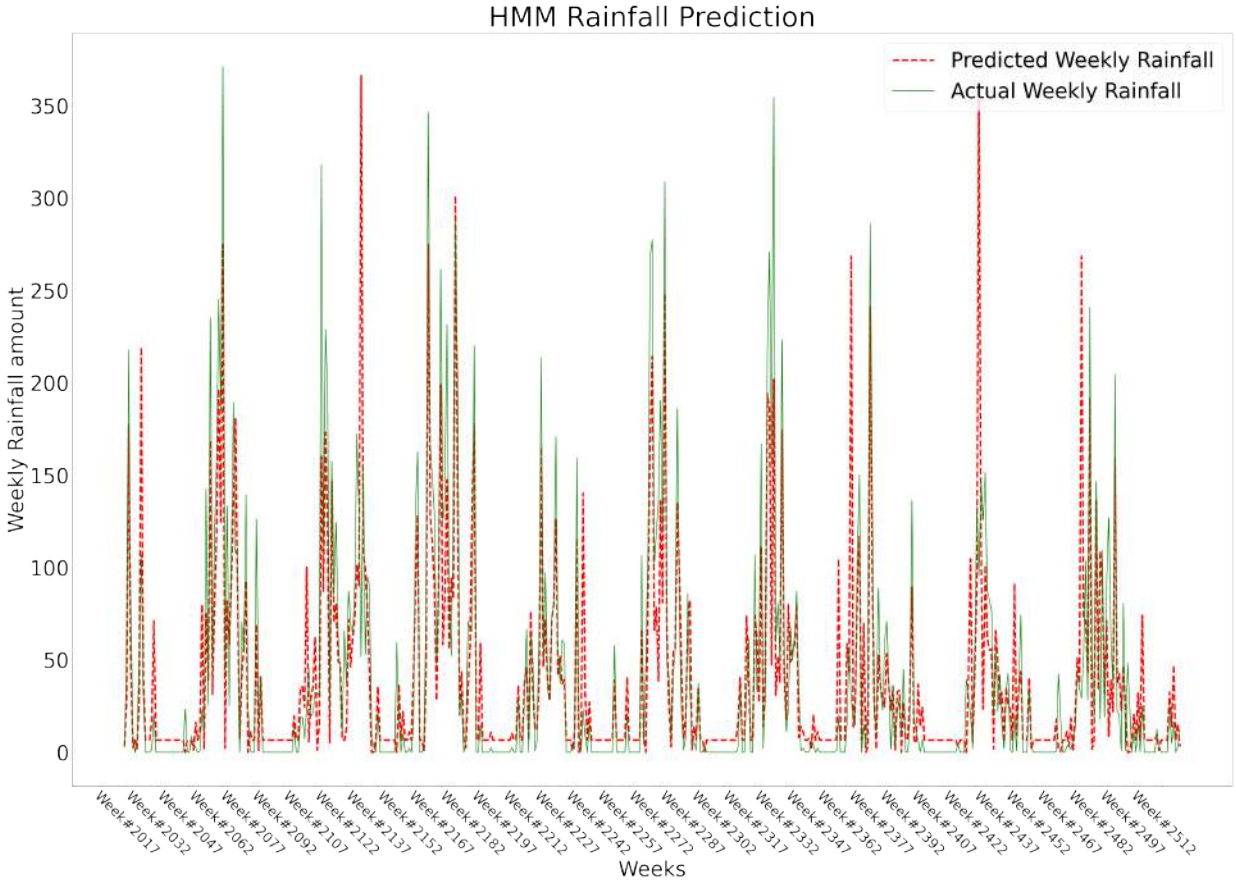


Fig. 5.7 HMM prediction on test data with experience factor = 300

Year	RMSE	MAE	CC
2018	29.92	24.67	0.98
2019	50.53	33.90	0.94
2020	28.90	24.00	0.95

Table 5.3 Performance Metrics of HMM model with experience factor on 2018, 2019, 2020 rainfall

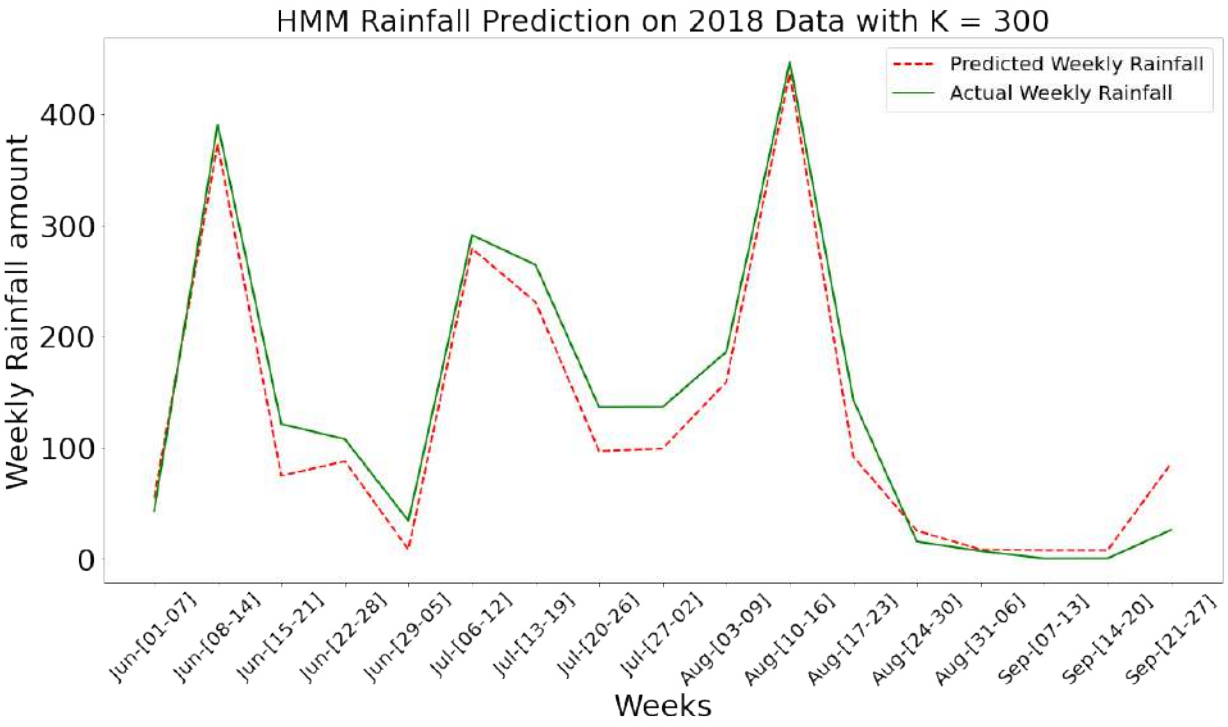


Fig. 5.8 HMM prediction on 2018 data with experience factor

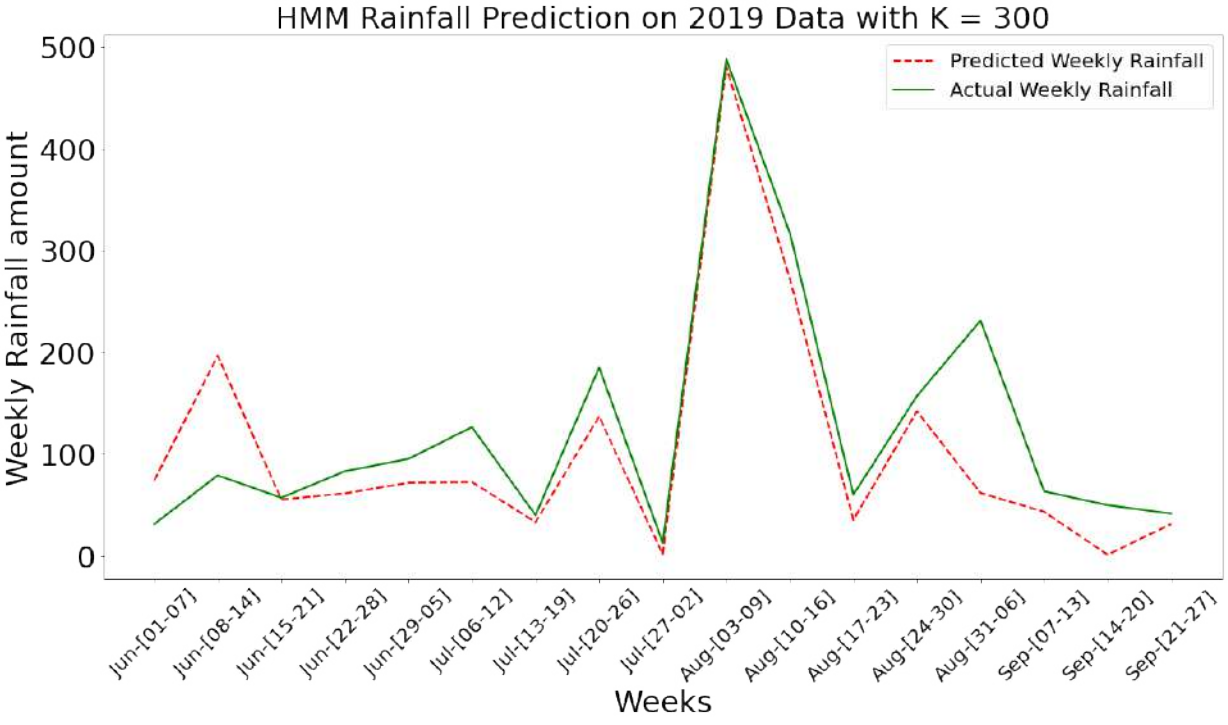


Fig. 5.9 HMM prediction on 2019 data with experience factor

5.7 Conclusion

In this part, we talked about the prediction using HMM model. We have seen a very new method of doing prediction using HMM model. We found that HMM has shown great results and its performance has improved a lot when we introduced the hyperparameter Experience factor. We have found that the model can handle the sudden sharp changes in the rainfall data is also able to predict the extremity in the data.

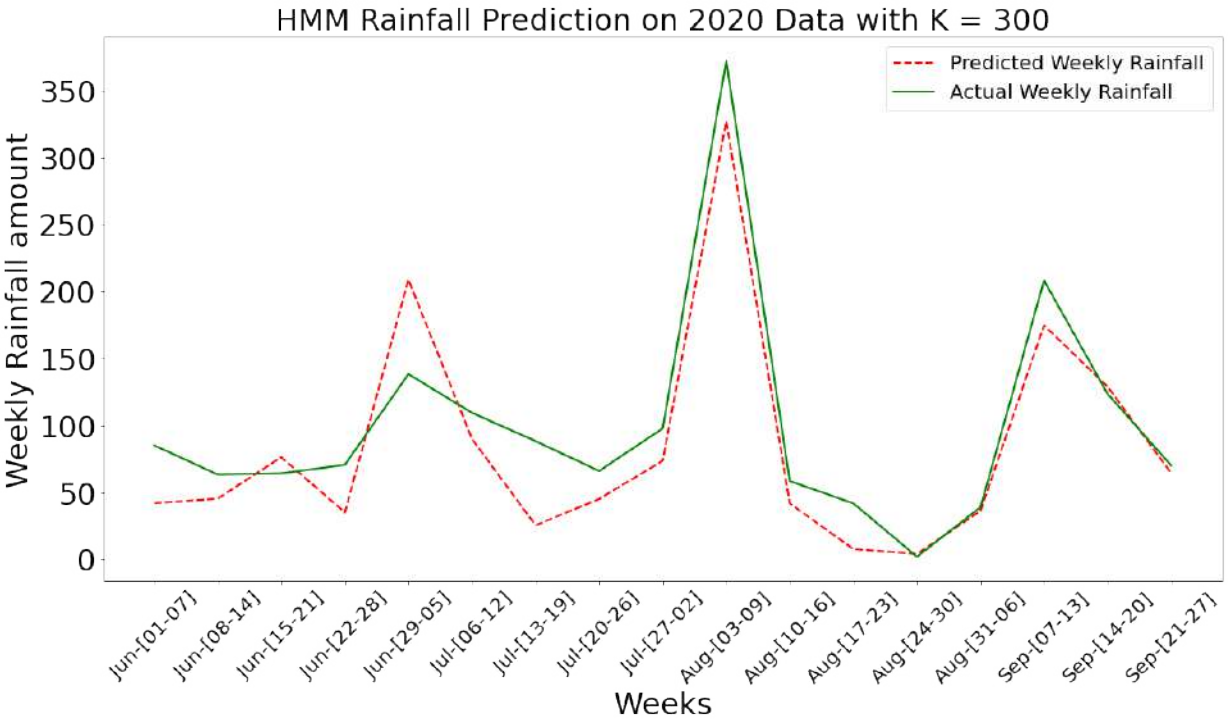


Fig. 5.10 HMM prediction on 2019 data with experience factor

Chapter 6

Finding Experience Factor and Testing The Model On Stations Rainfall Data

In the previous chapter, we found that the graph of RMSE vs K is a rectangular hyperbola. So RMSE decreases as we increase the K. In this chapter to locate a good K, we are using the property of rectangular hyperbola. We will try to locate the vertex of the rectangular hyperbola. Vertex is the point of the sharpest turn in the curve. That means after vertex the changes in RMSE value will low. For a standard rectangular hyperbola $xy = c^2$ it's asymptotes are the coordinate axis i.e $x = 0$ and $y = 0$ and hence we can locate it's vertex at (c, c) . But in our case, we don't know the asymptotes because the graph we obtained is a general rectangular hyperbola and its equation is unknown to us. The other way we are following is by drawing a diagonal line and then the meet point of the line and the curve is the vertex point. See Fig.6.1 for the same.

Hence locating the vertex of the graph RMSE vs K will give us a good K. We have seen in the previous chapter that the HMM model is showing a great result on 2018, 2019, and 2020 rainfall data. In this chapter, we will assess the model based on the rainfall data of

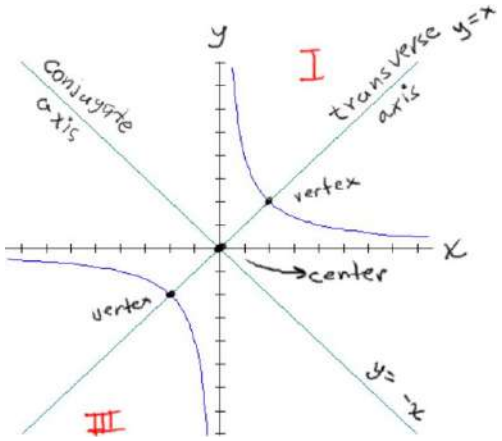


Fig. 6.1 Finding vertex of a rectangular hyperbola

different stations as stated in chapter 3. This will give us the information that the model is not data-driven instead perform generally well on different kinds of data and guarantees that it will do equally well in real-life also.

6.1 Testing

We have described the station's wise rainfall data in chapter 3. Since the data is of variable length for different stations. So on average we have 80% of data for traininig the model and kept 20% for testing purpose. Out of this 20%, we have 10% of data as validation set to find the hyperparameter K. Below is the list of figures of prediction on different sations where (a) of figure shows the data and it's split into training, validation and test, (b) of figure shows the training of the HMM model, (c) of figure shows finding of a good K for the data and (d) shows the prediction of the HMM model on the test data.

Fig [6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.13] shows the prediction of HMM model on 7 different region of **Malappuram**

Fig [6.8, 6.9, 6.10, 6.11, 6.12, 6.14, 6.15, 6.16, 6.35] shows the prediction of HMM model on 6 different region of **Palakkad**

Fig [6.17, 6.18, 6.19, 6.20, 6.21, 6.22, 6.23, 6.24] shows the prediction of HMM model on 6 different region of **Thrissur**

Fig [6.25, 6.26, 6.27, 6.28, 6.29, 6.30, 6.31, 6.32, 6.33, 6.34] shows the prediction of HMM model on 6 different region of **Coimbatore**

Table 6.1 shows the performance metrices of HMM on station data.

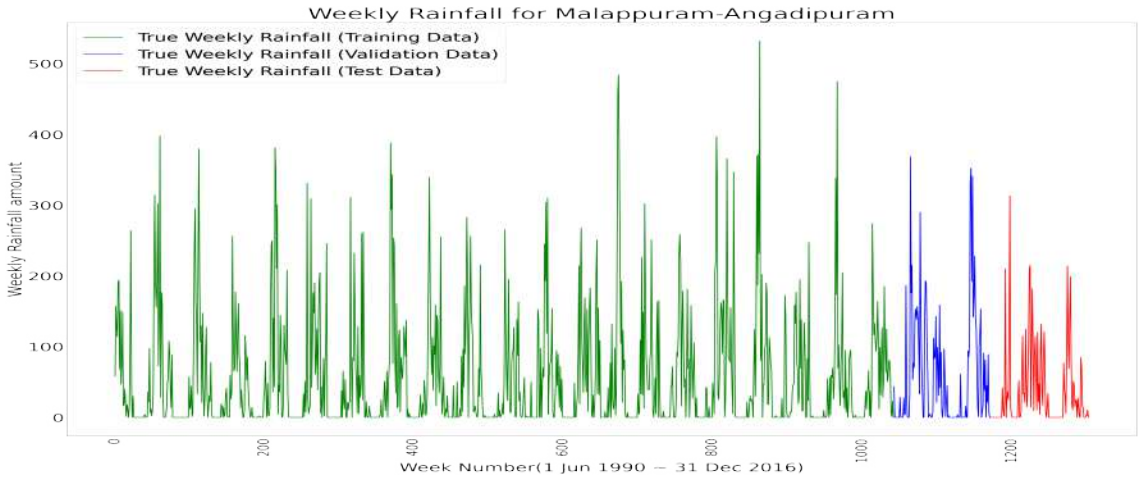
Stations Name	**RMSE**	**MAE**	**CC**
Malappuram-Tiruvangadi	34.89	16.06	0.99
Palakkad-Alathur	33.01	16.68	0.97
Palakkad-Alattur	27.73	14.94	0.96
Palakkad-Cherapalaseri	45.55	26.62	0.97
Palakkad-Mannarkad	32.90	18.68	0.97
Palakkad-Ottapalam	47.78	25.91	0.94
Malappuram-Palakkad	31.38	14.13	0.92
Palakkad-OBSY	24.31	12.51	0.97
Palakkad-Parli	34.92	17.36	0.97
Palakkad-Pattembi	40.10	20.45	0.95
Thrissur-Chalakudi	63.49	29.54	0.89
Thrissur-Enamakkal	49.28	22.10	0.89
Thrissur-Kodungallur	39.71	19.56	0.95
Thrissur-Mukundarpuram	69.81	31.47	0.94
Thrissur-Ollukara	38.43	19.07	0.97
Thrissur-Peechi	33.24	19.15	0.98
Thrissur-Thalipilly	37.98	17.33	0.93
Thrissur-Thrissur	46.57	27.05	0.97
Coimbatore-Anaimalai	53.30	27.89	0.0.97
Coimbatore-Attakatti	25.76	12.10	0.90
Coimbatore-Nirardam	58.60	29.26	0.93

Malappuram-Angadipuram	38.62	19.91	0.95
Malappuram-Manjeri	47.39	24.90	0.95
Malappuram-Nilambur	36.45	18.15	0.96
Malappuram-Perinthalamanna	46.61	25.28	0.96
Malappuram-Ponnani	43.36	18.38	0.91
Coimbatore-Parambikulam	41.68	18.86	0.91
Coimbatore-Pollachi	17.32	8.12	0.97
Coimbatore-Sholiyarnagar	76.43	28.06	0.87
Coimbatore-Solayar	39.55	16.26	0.97
Coimbatore-Topslip	17.09	7.81	0.94
Coimbatore-Nirar	72.71	35.90	0.95
Coimbatore-Valparai	40.93	20.78	0.94
Palakkad-Chittur	36.01	14.28	0.90

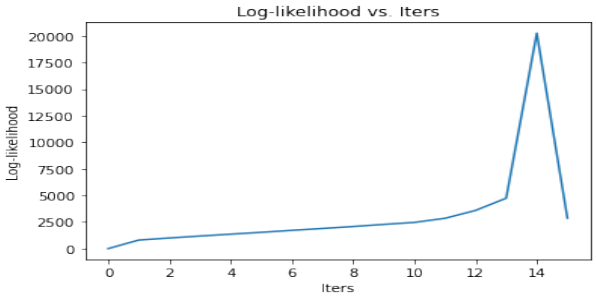
Table 6.1: Performance metrices of HMM model on station rainfall data

6.2 Conclusion

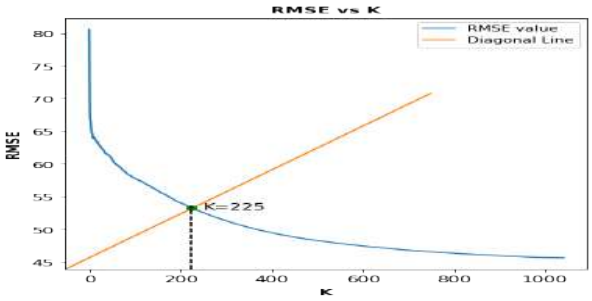
In this chapter, we have further evaluated our HMM model on a variety of different types of data. We have found that the model is not data-driven as it has shown a great result on all the station's rainfall data.



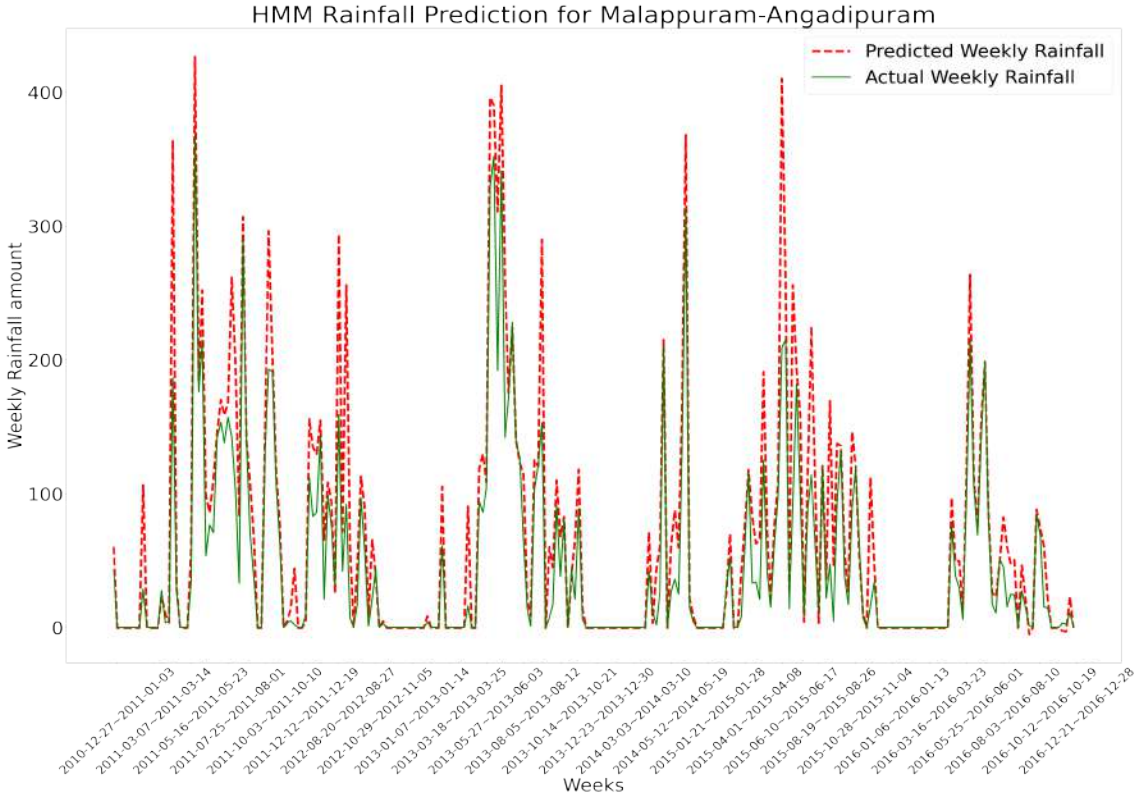
(a)



(b)

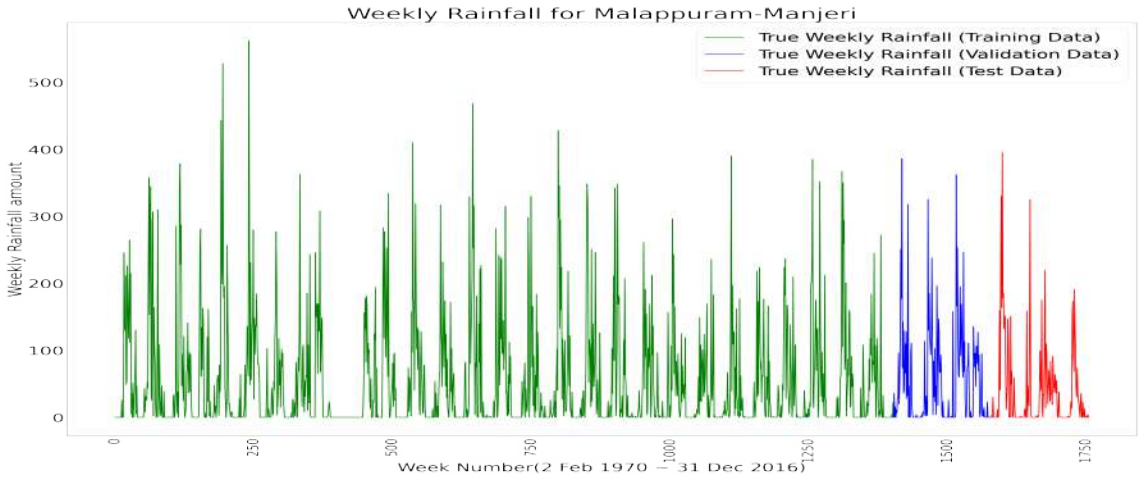


(c)

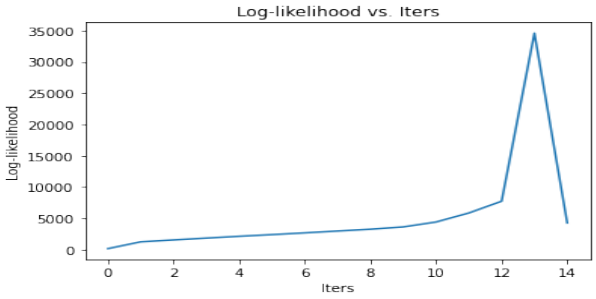


(d)

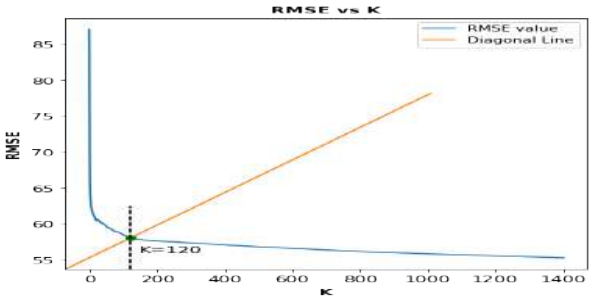
Fig. 6.2 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Angadipuram



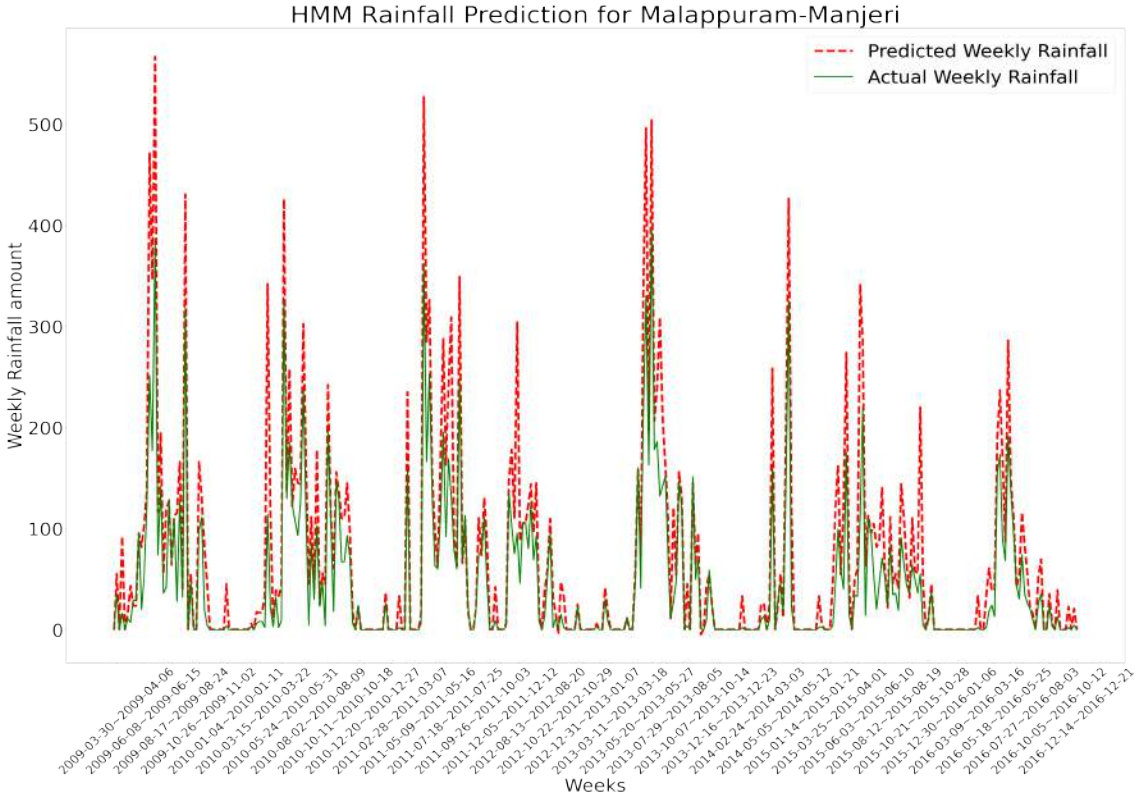
(a)



(b)

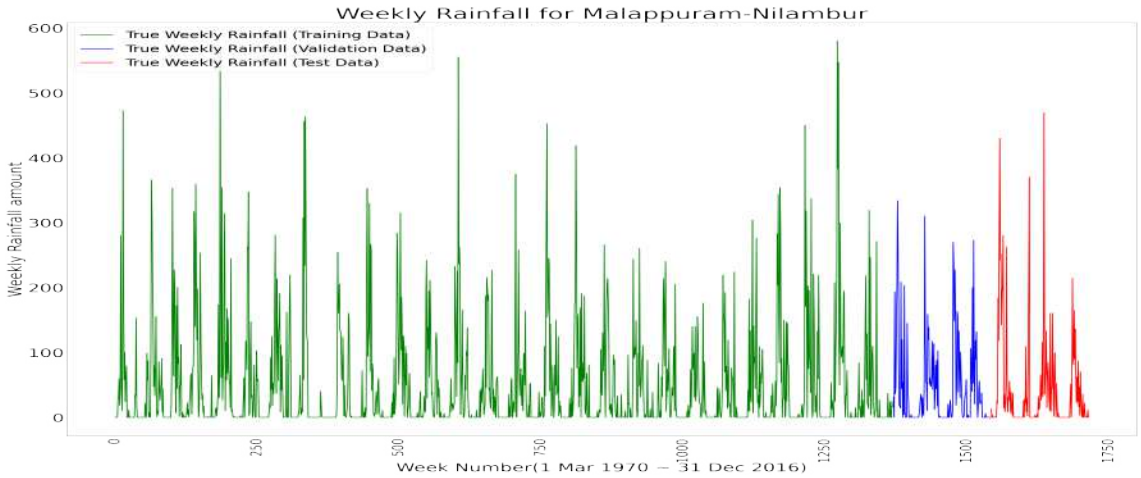


(c)

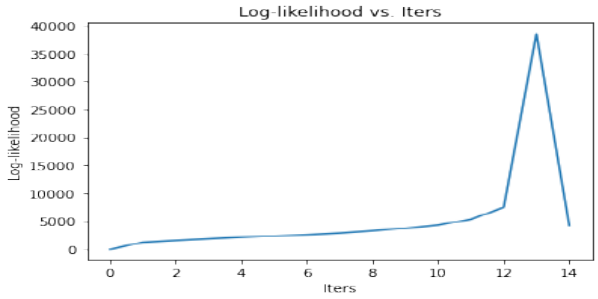


(d)

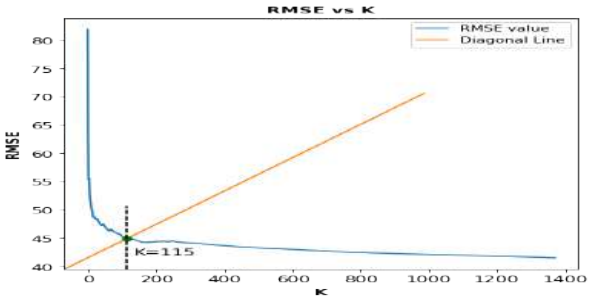
Fig. 6.3 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Manjeri



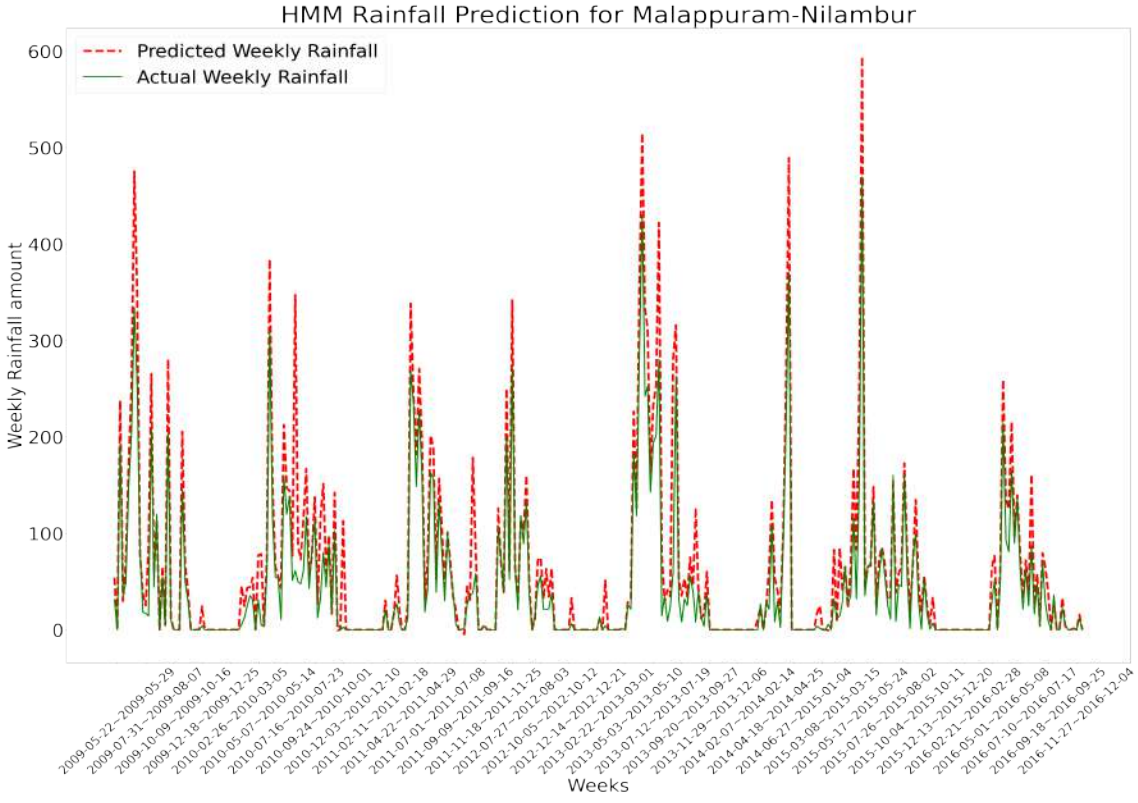
(a)



(b)

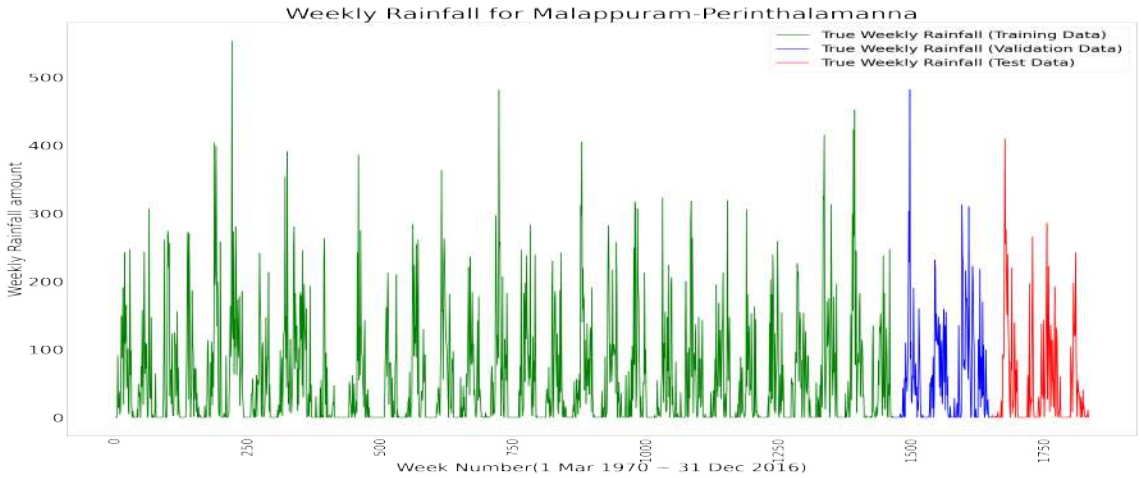


(c)

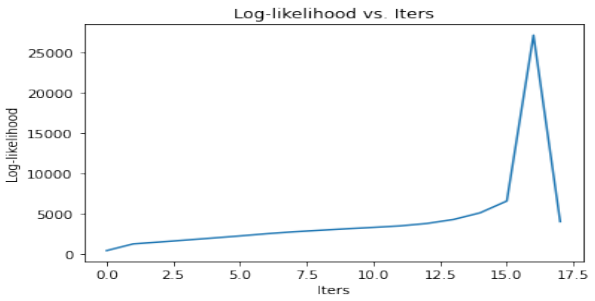


(d)

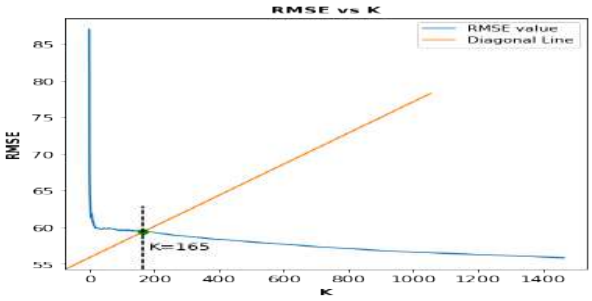
Fig. 6.4 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Nilambur



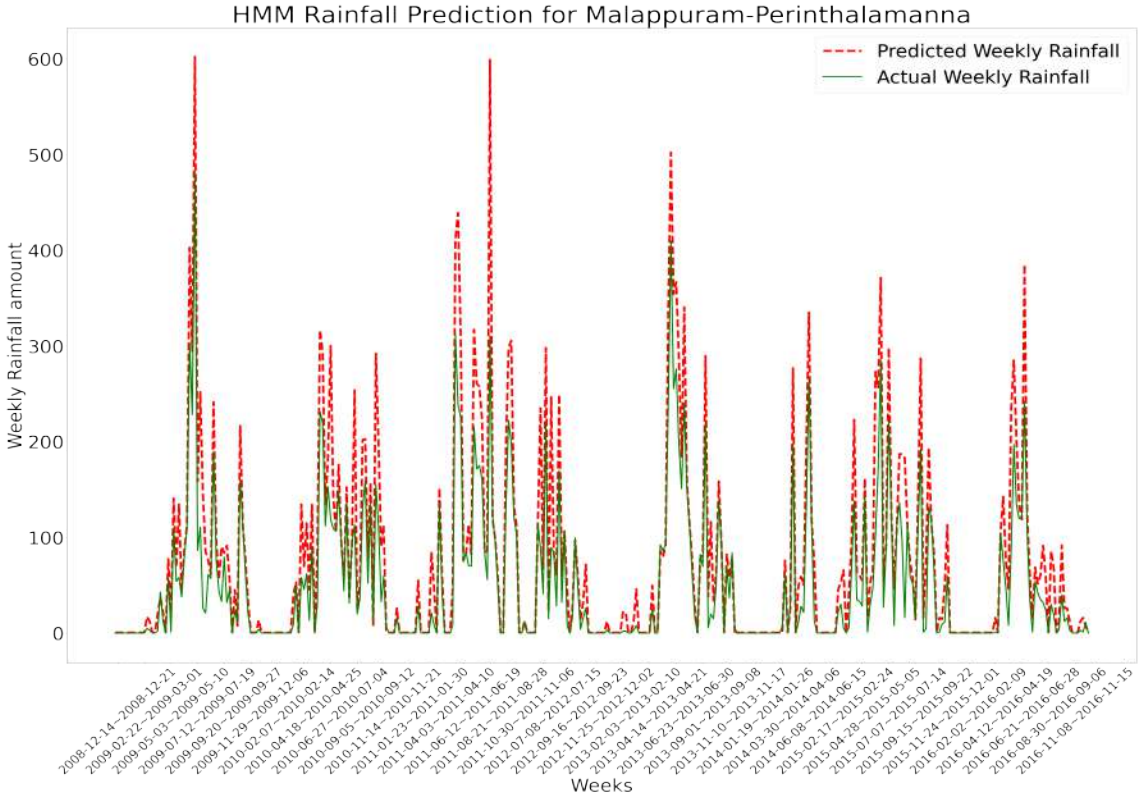
(a)



(b)

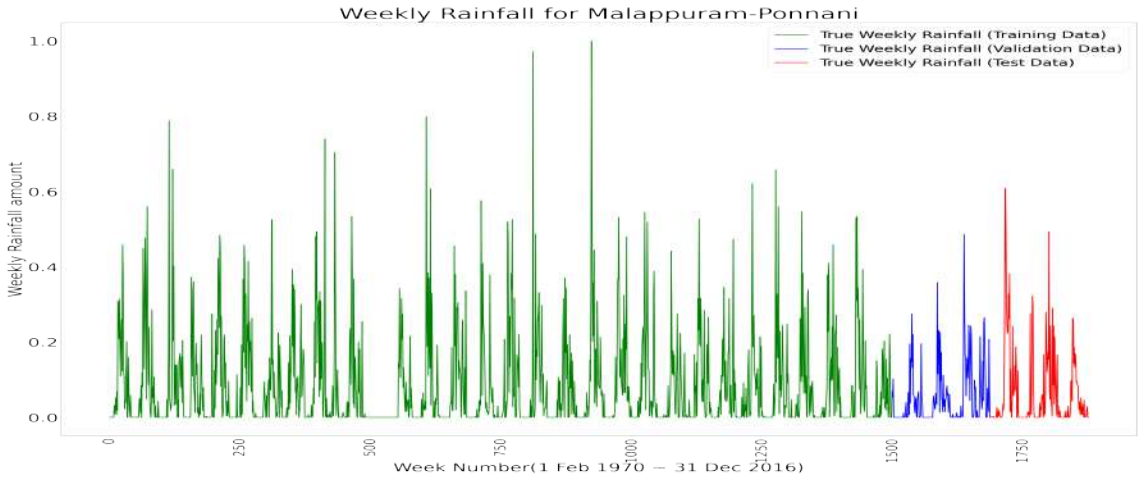


(c)

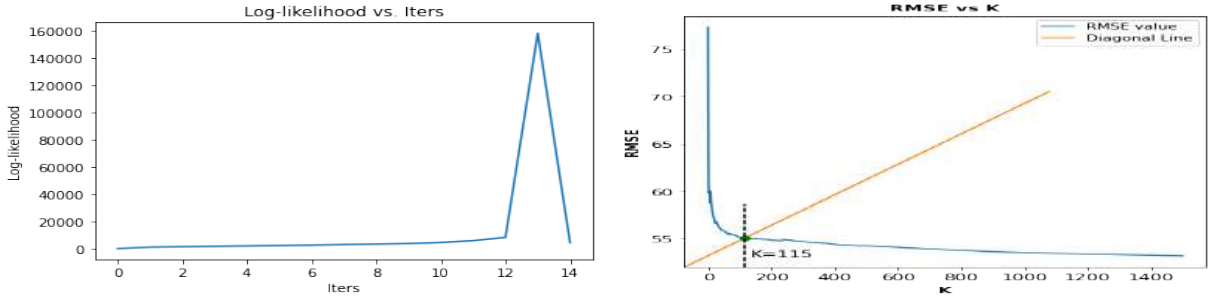


(d)

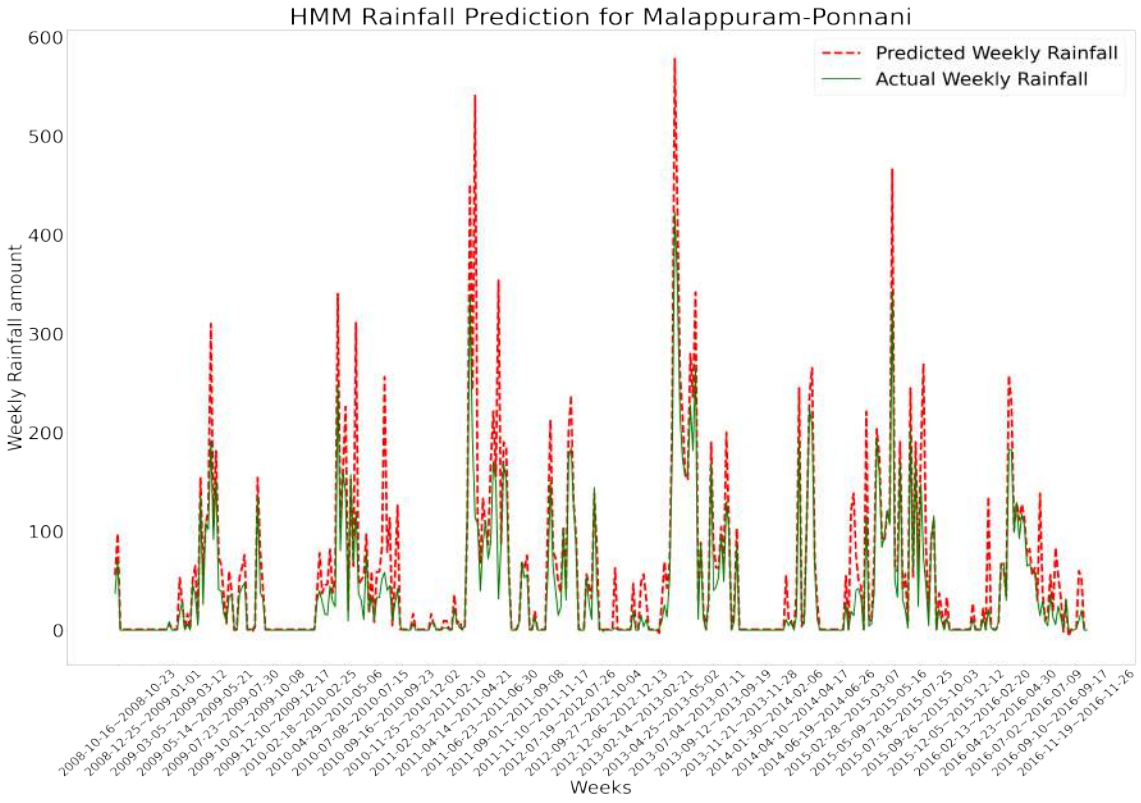
Fig. 6.5 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Perinthalamanna



(a)

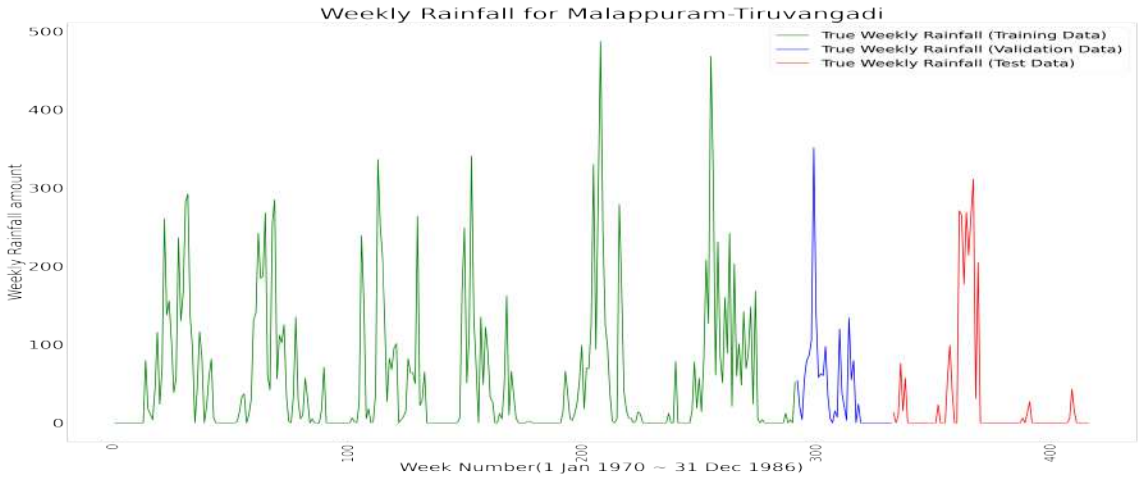


(b) (c)

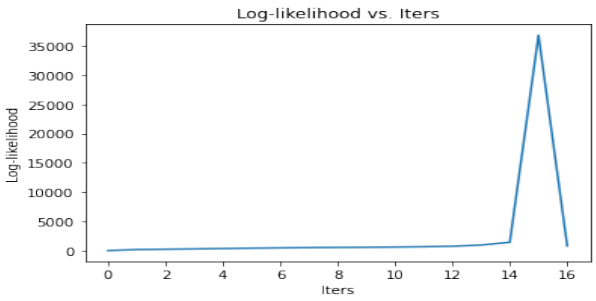


(d)

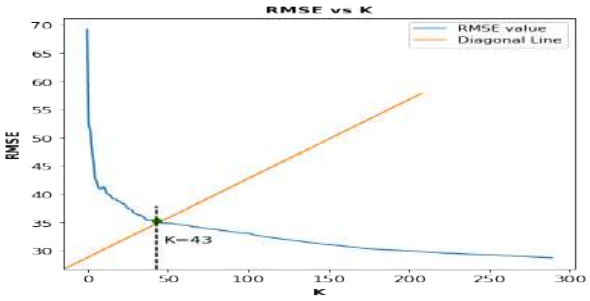
Fig. 6.6 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Ponnani



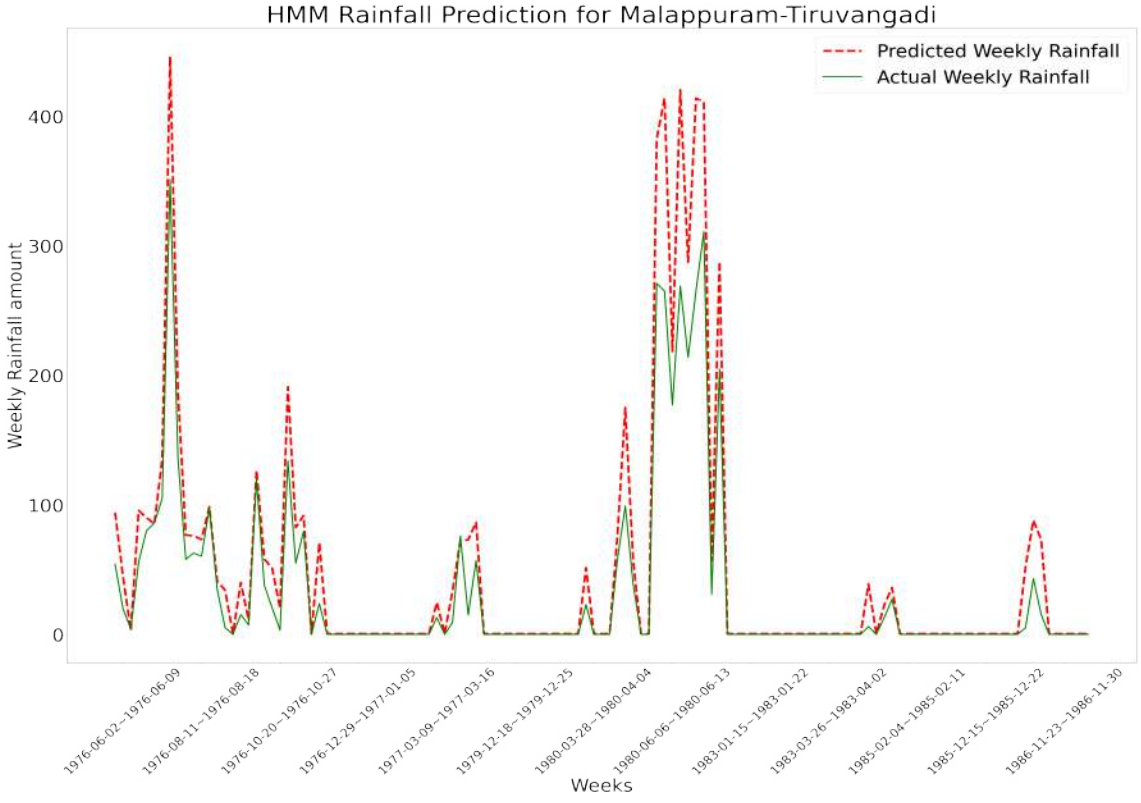
(a)



(b)

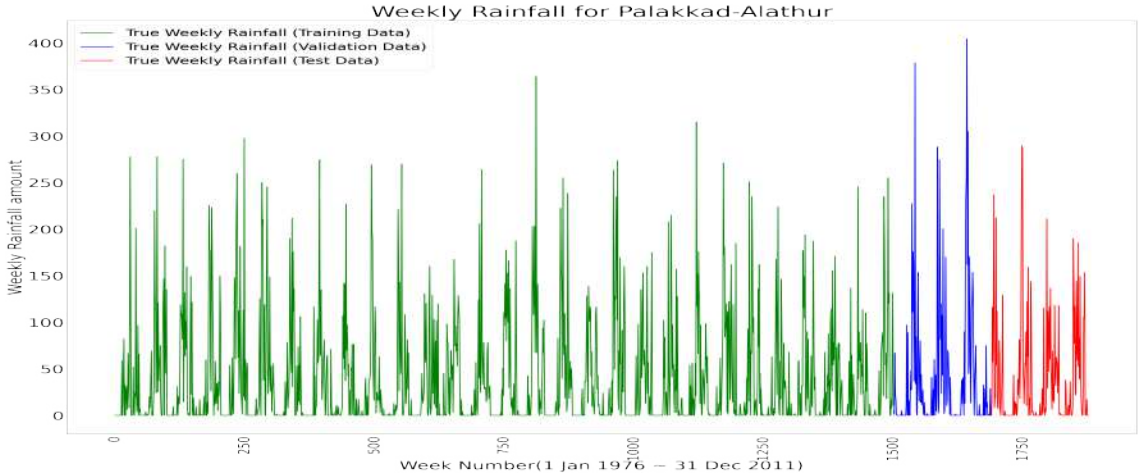


(c)

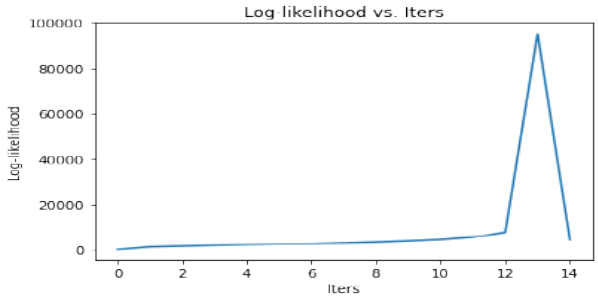


(d)

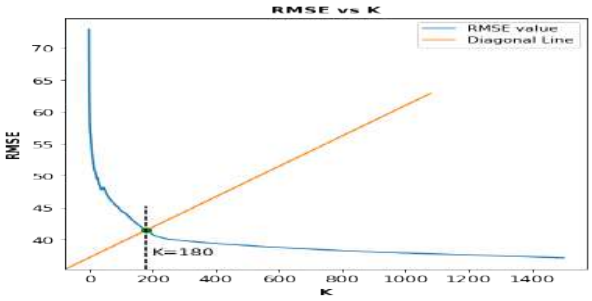
Fig. 6.7 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Tiruvangadi



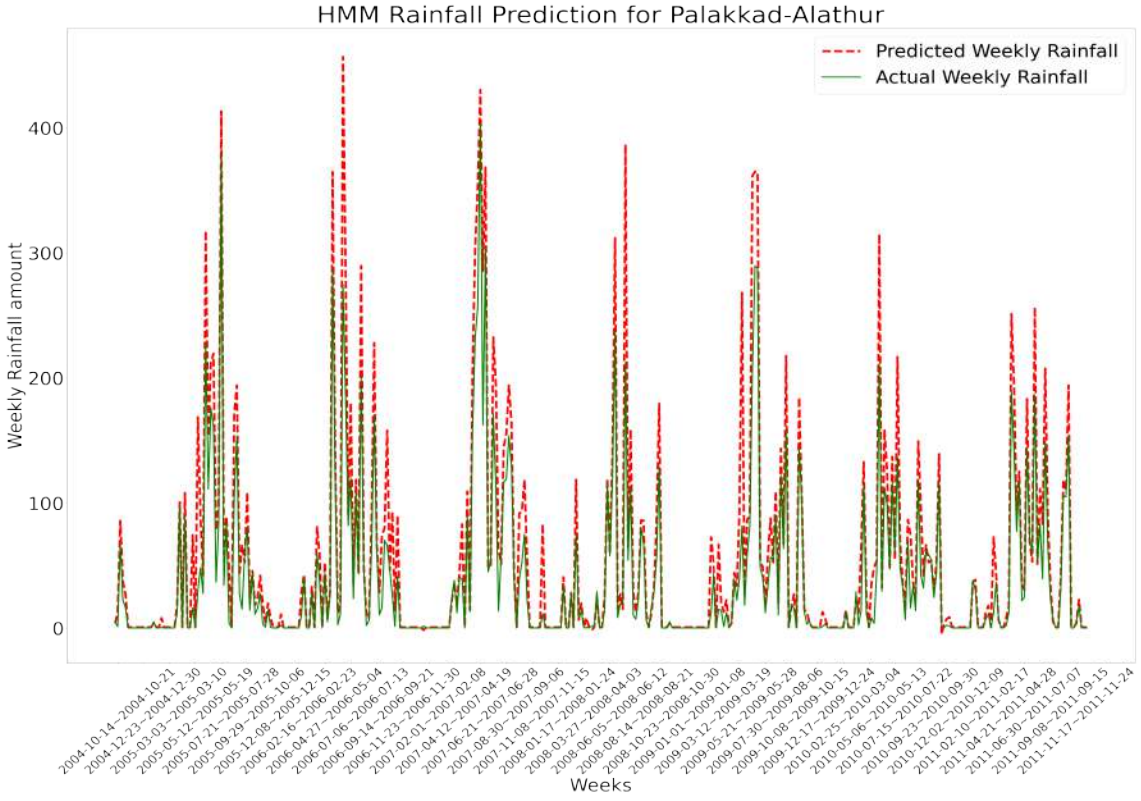
(a)



(b)

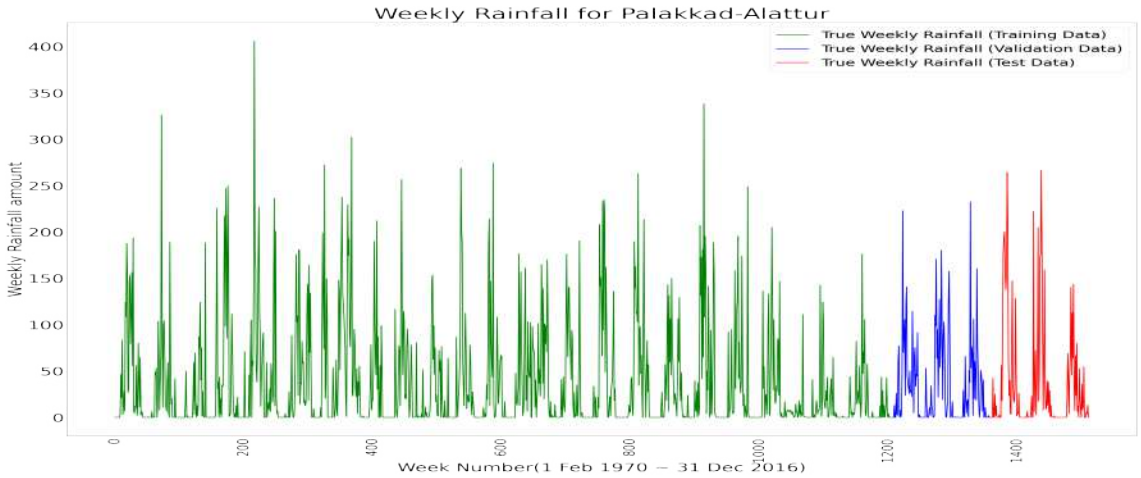


(c)

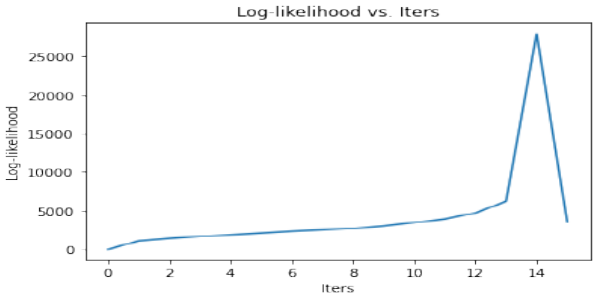


(d)

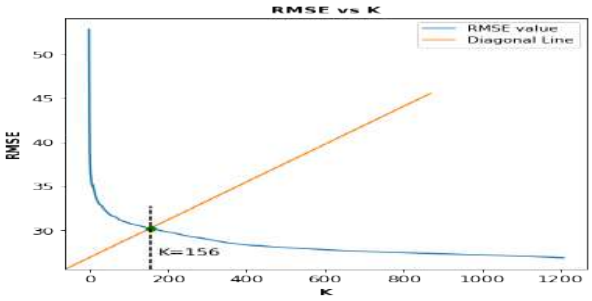
Fig. 6.8 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Alathur



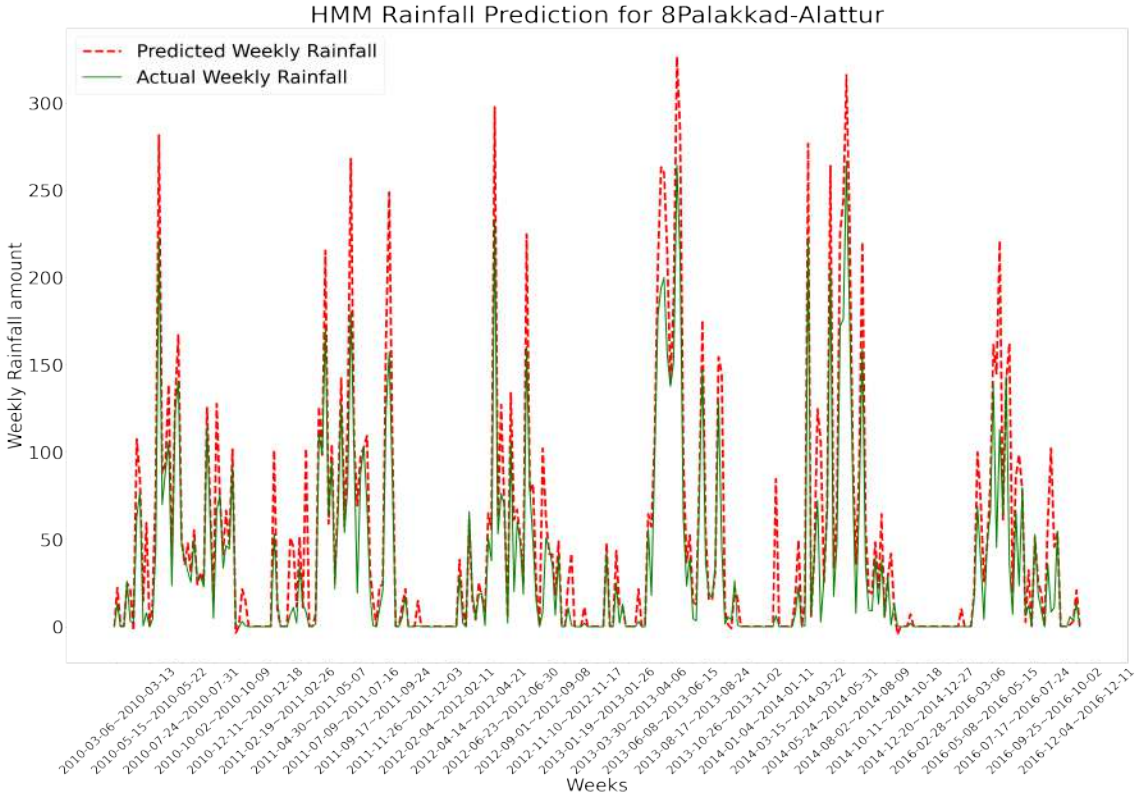
(a)



(b)

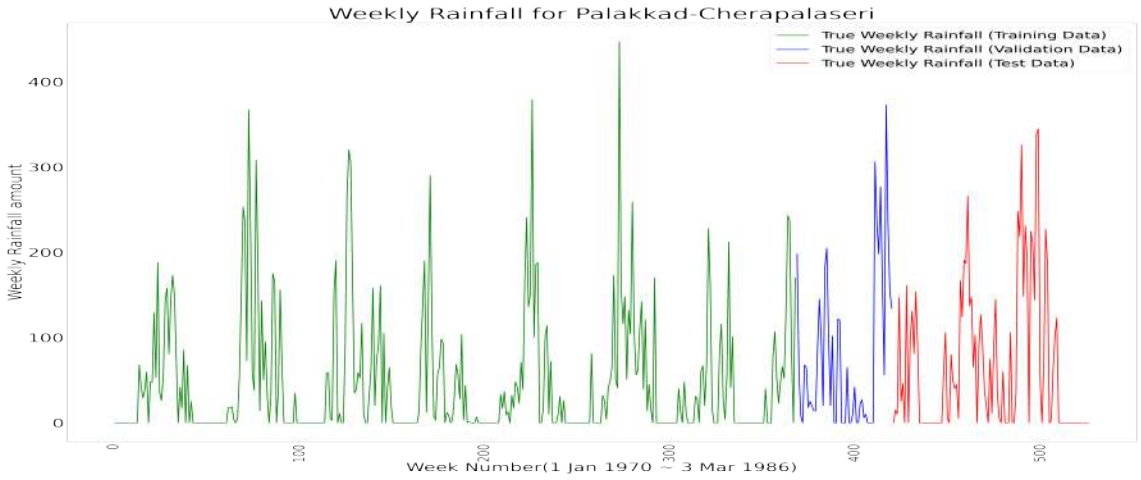


(c)

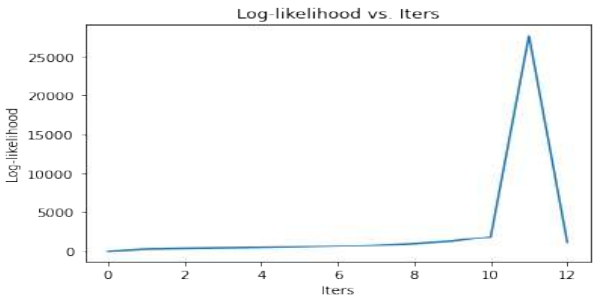


(d)

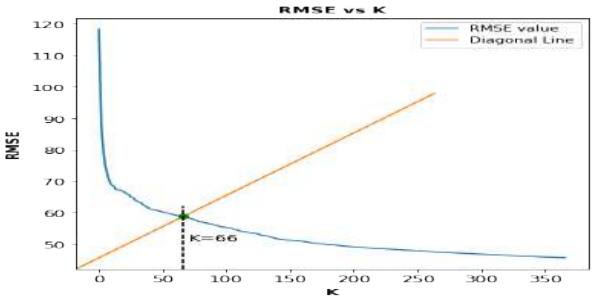
Fig. 6.9 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Alattur



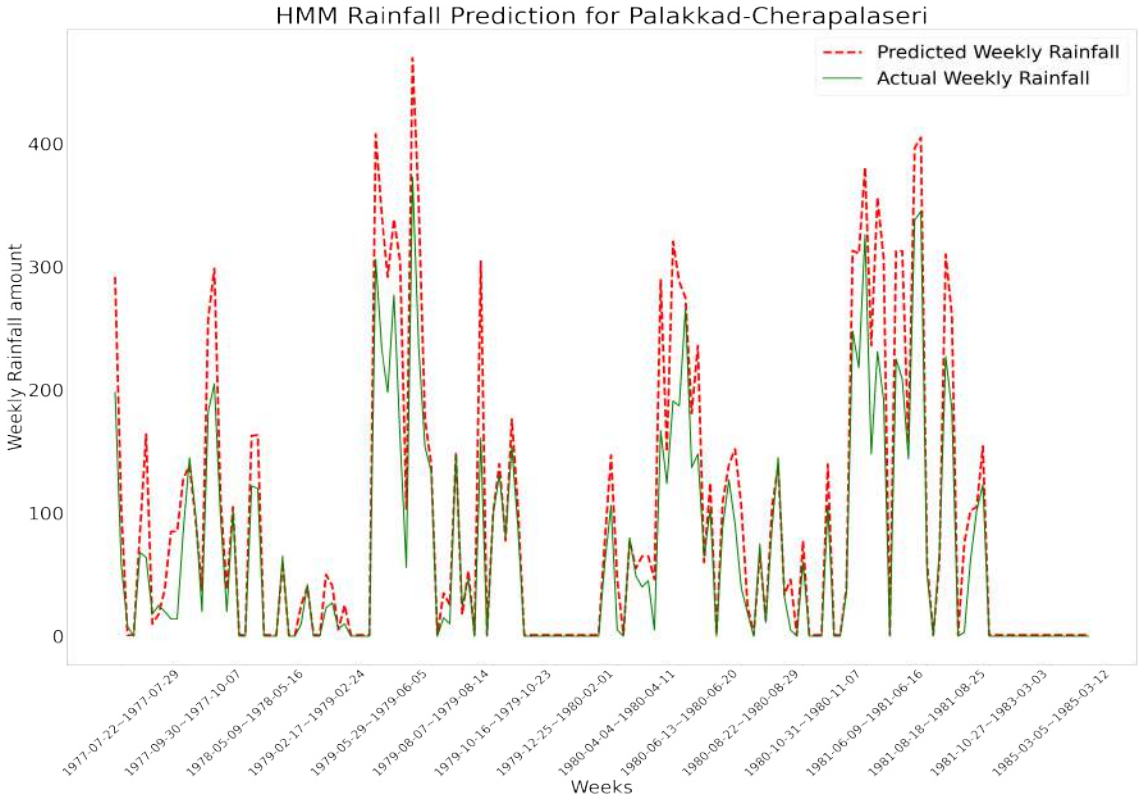
(a)



(b)

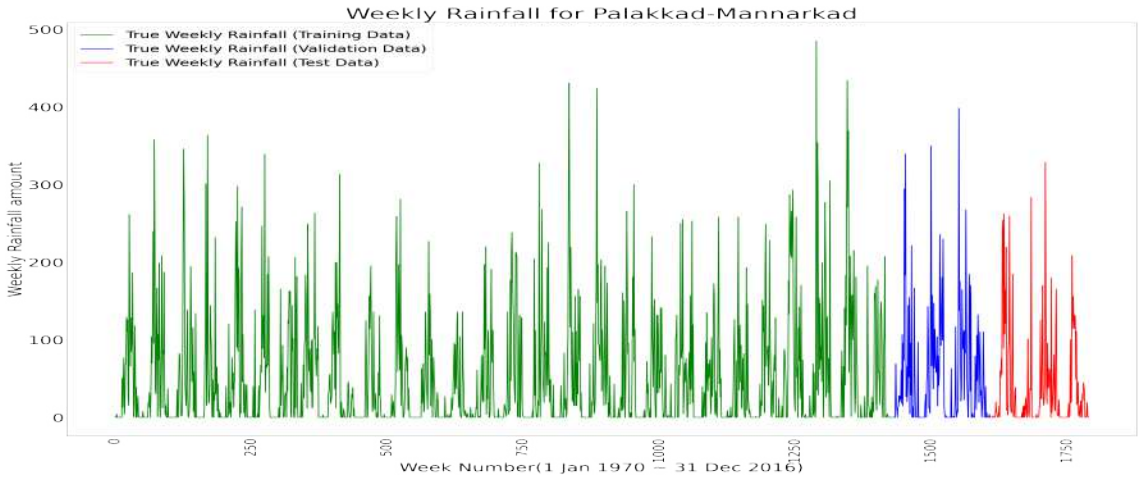


(c)

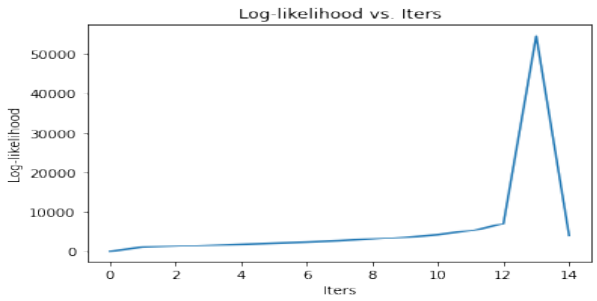


(d)

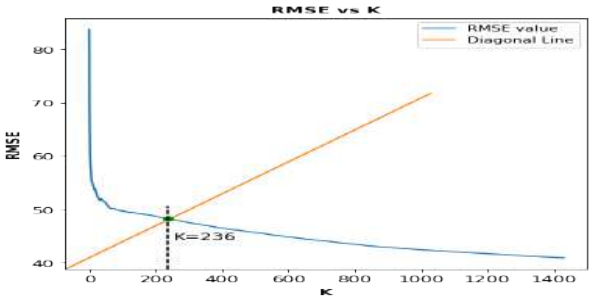
Fig. 6.10 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Cherapalaseri



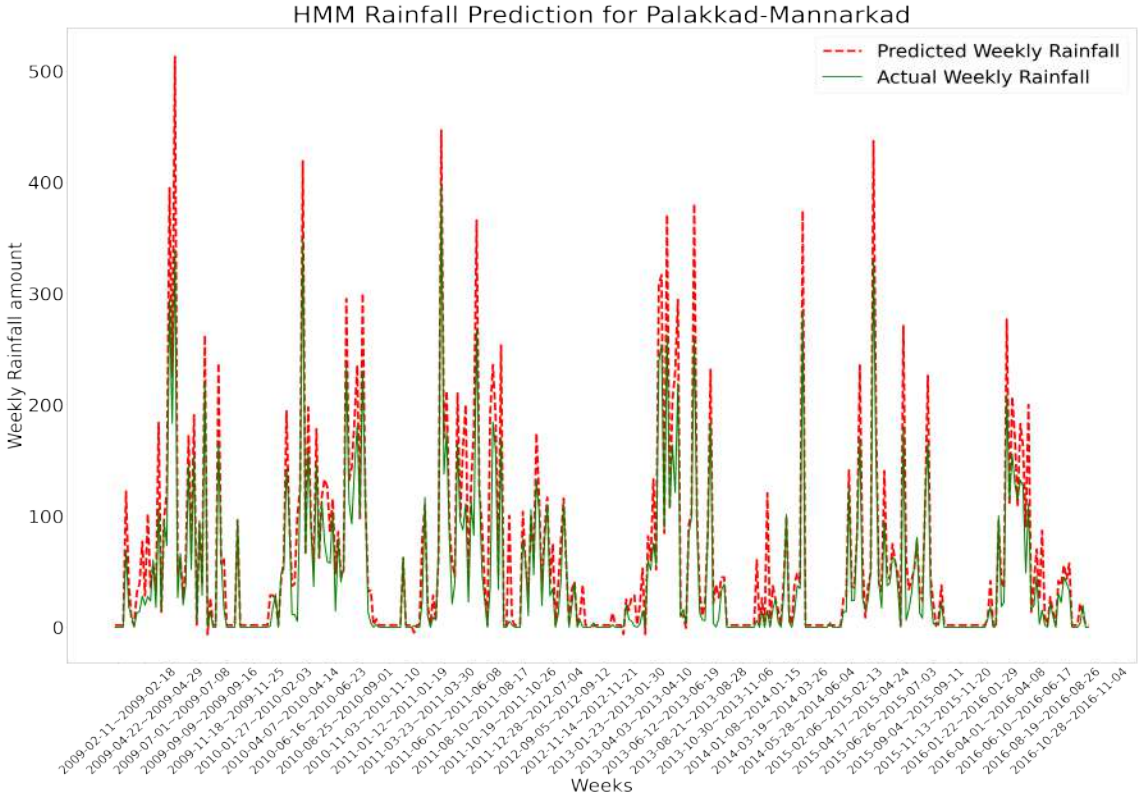
(a)



(b)

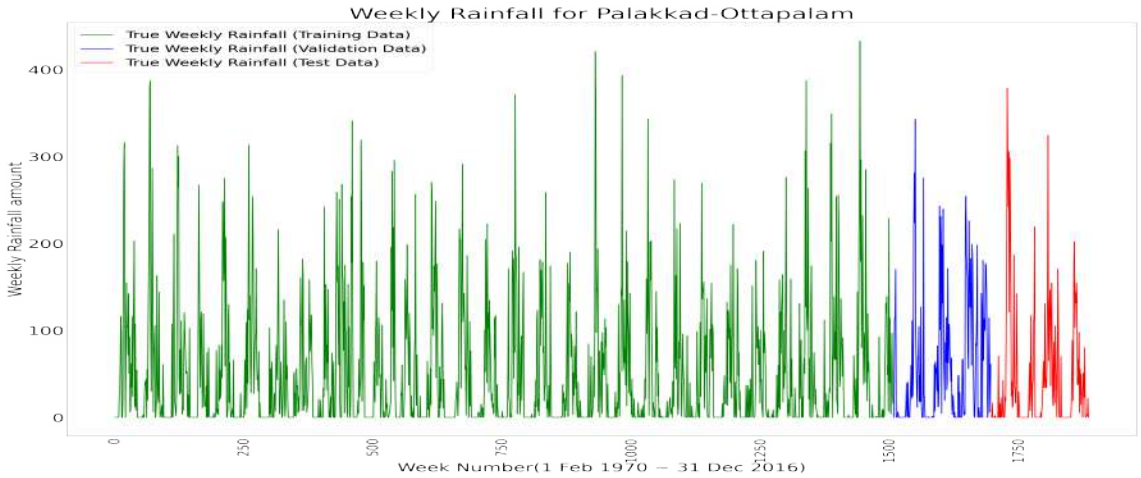


(c)

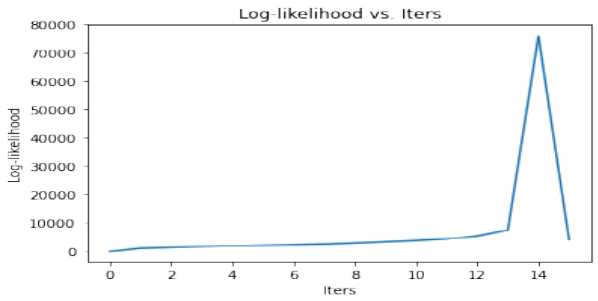


(d)

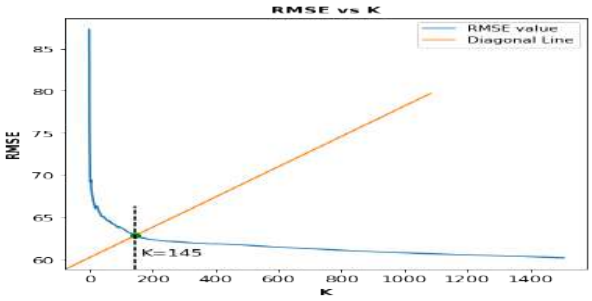
Fig. 6.11 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Mannarkad



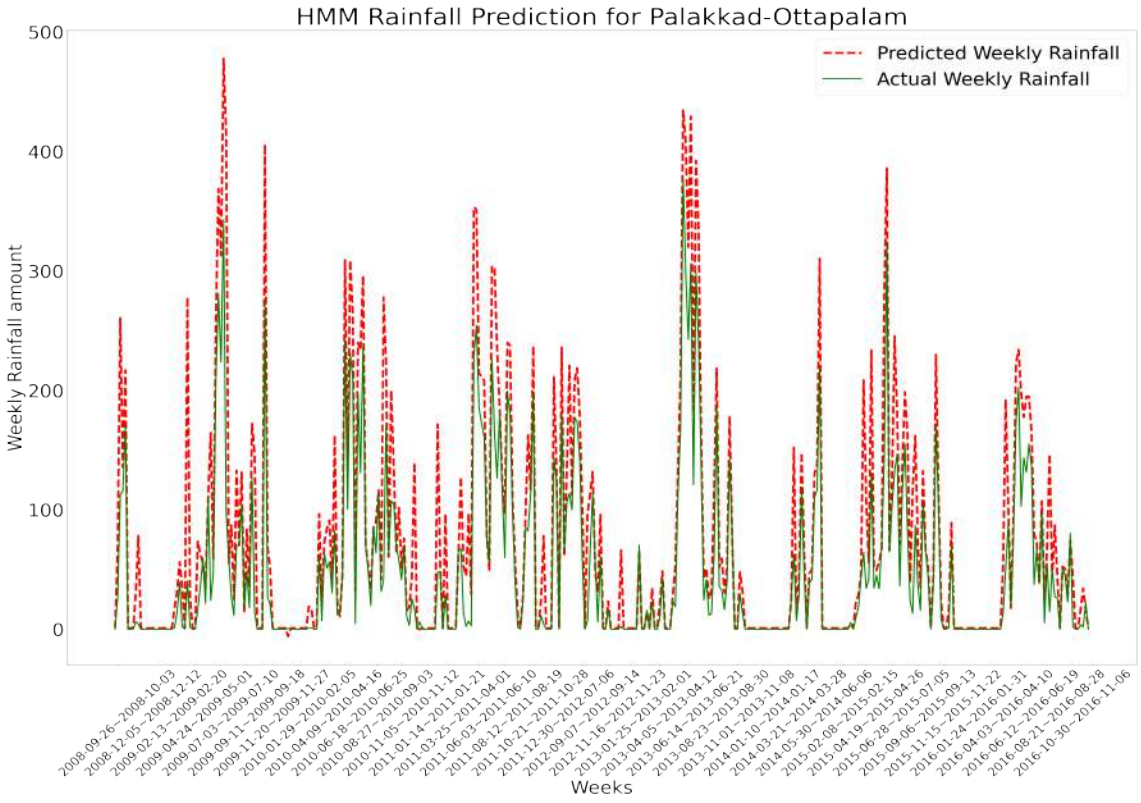
(a)



(b)

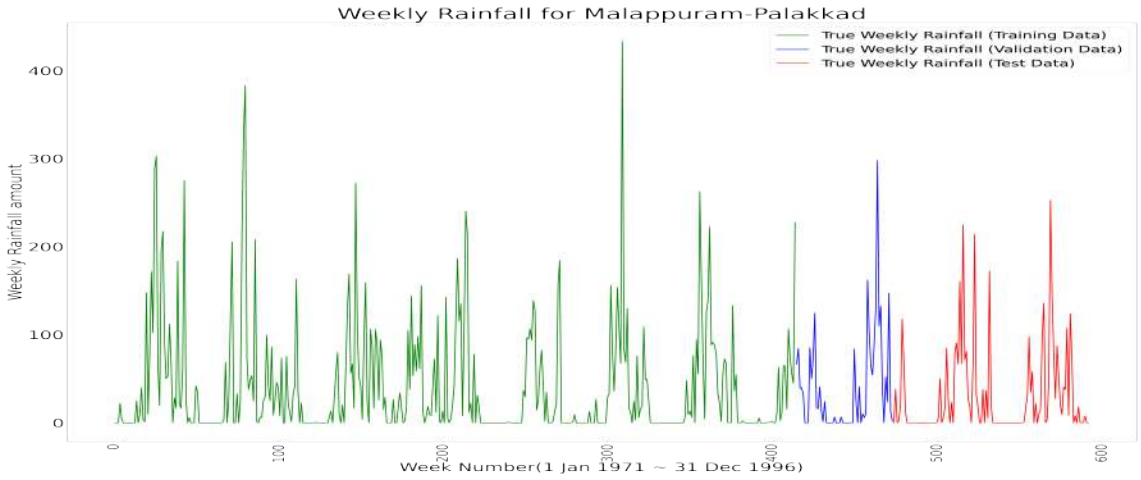


(c)

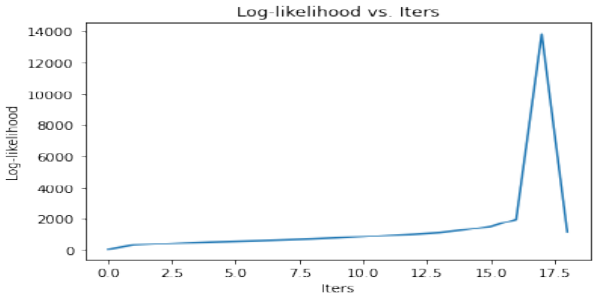


(d)

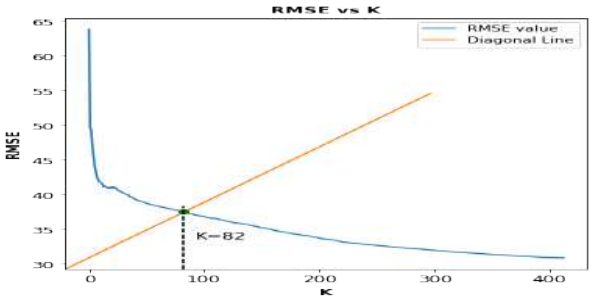
Fig. 6.12 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Ottapalam



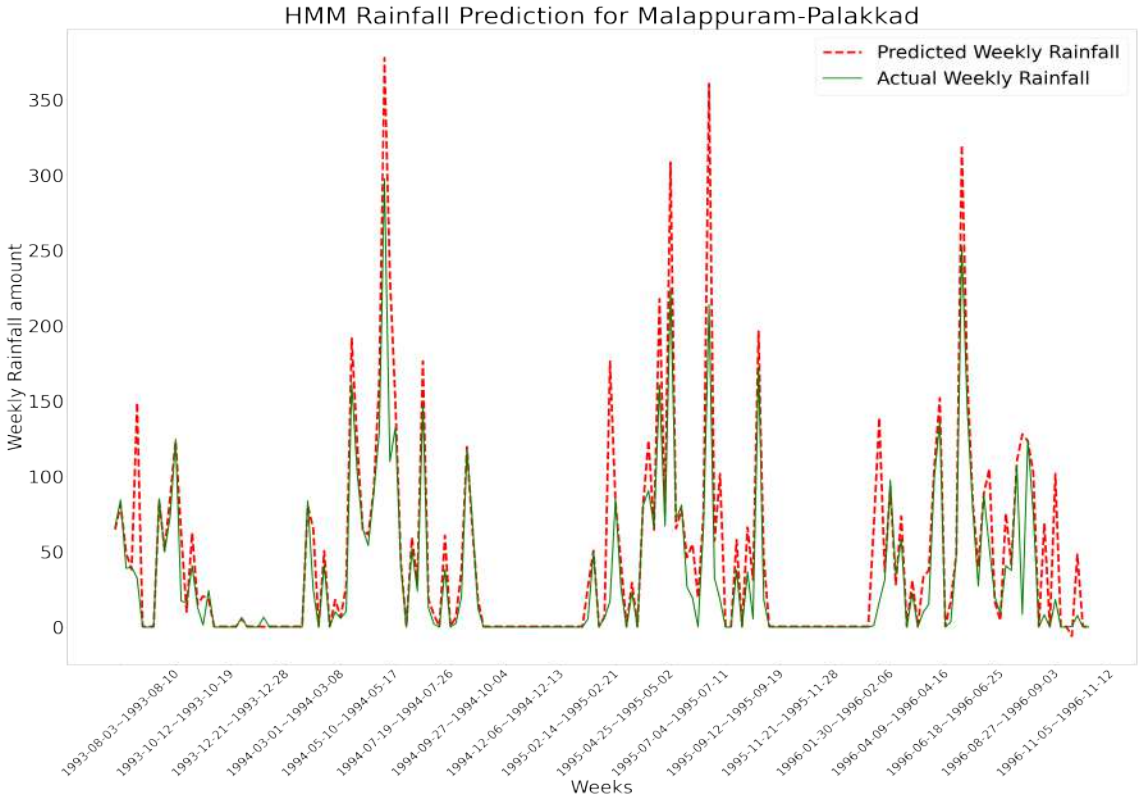
(a)



(b)

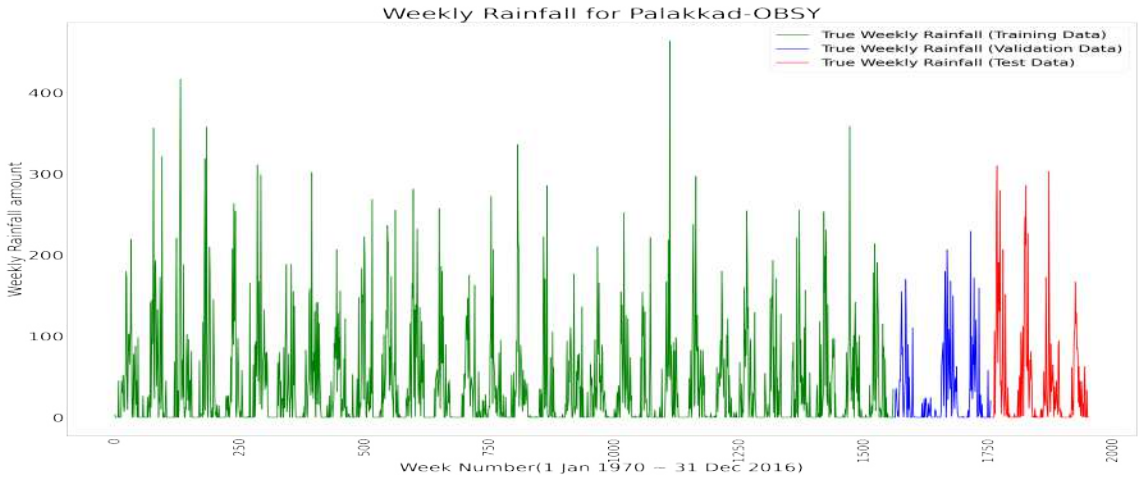


(c)

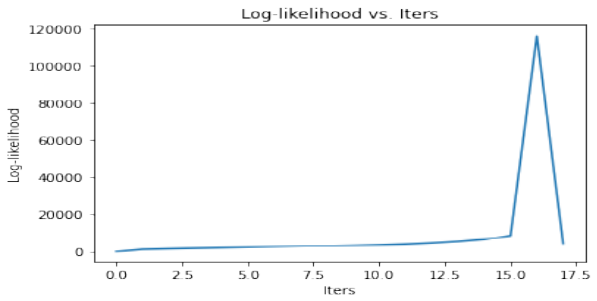


(d)

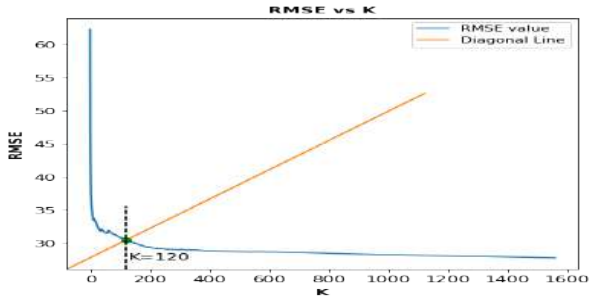
Fig. 6.13 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Palakkad



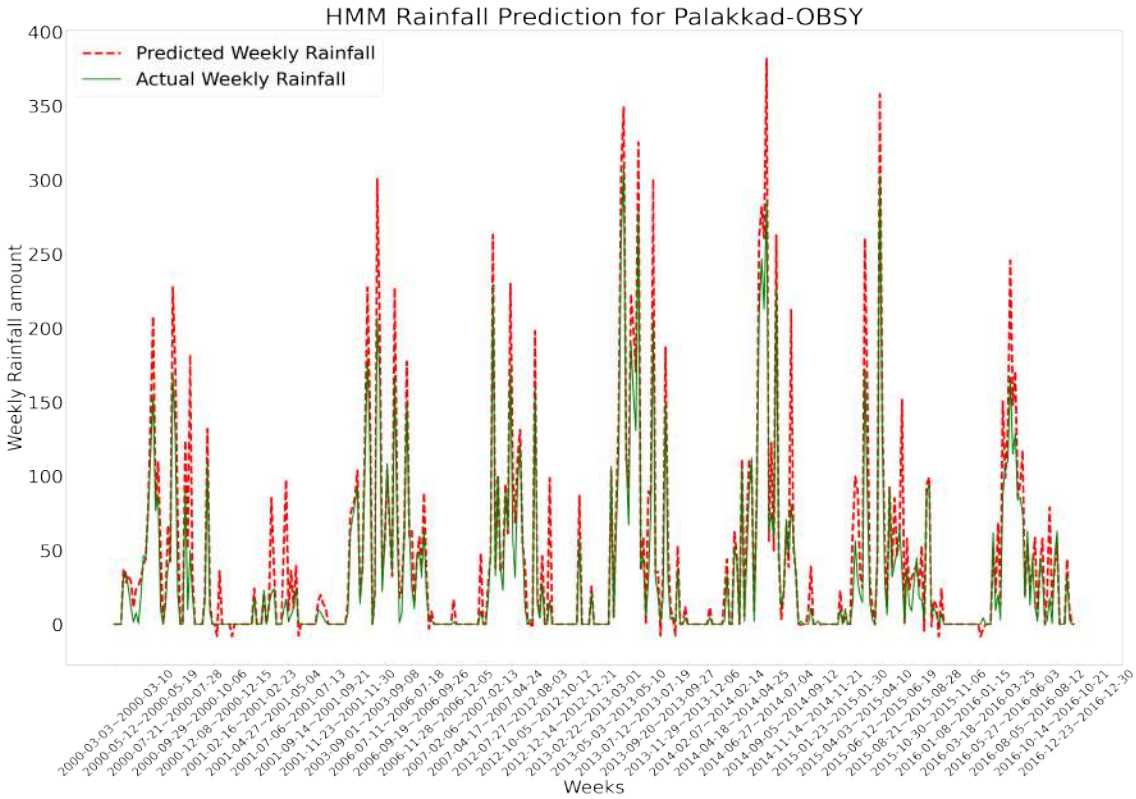
(a)



(b)

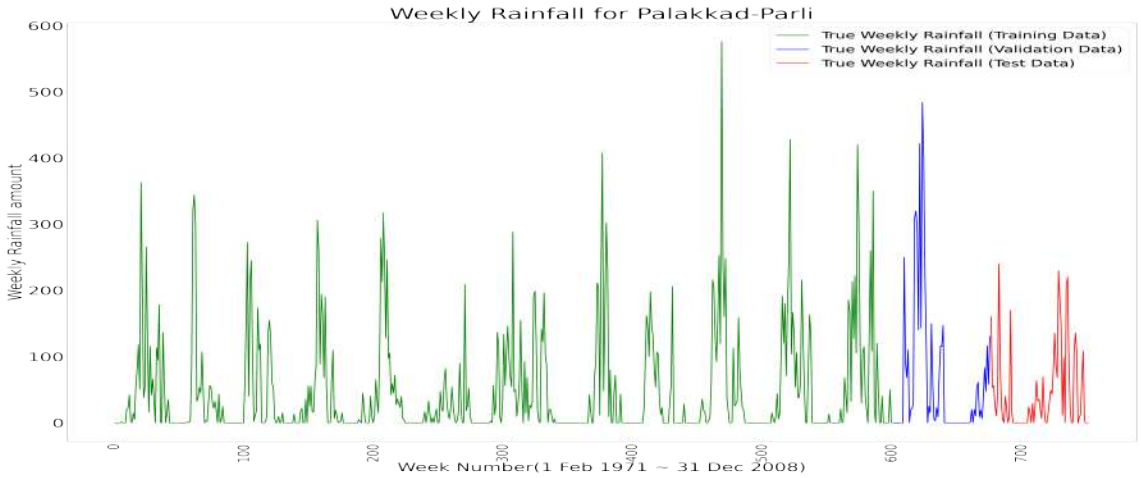


(c)

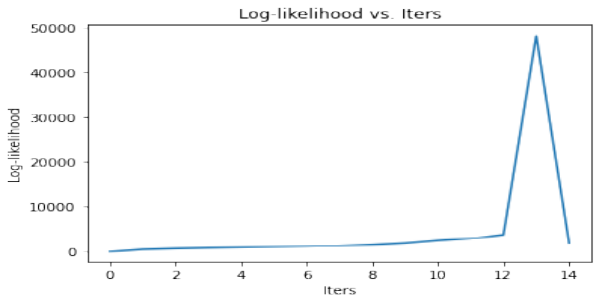


(d)

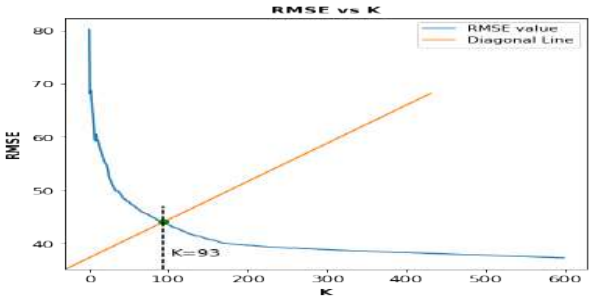
Fig. 6.14 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for OBSY



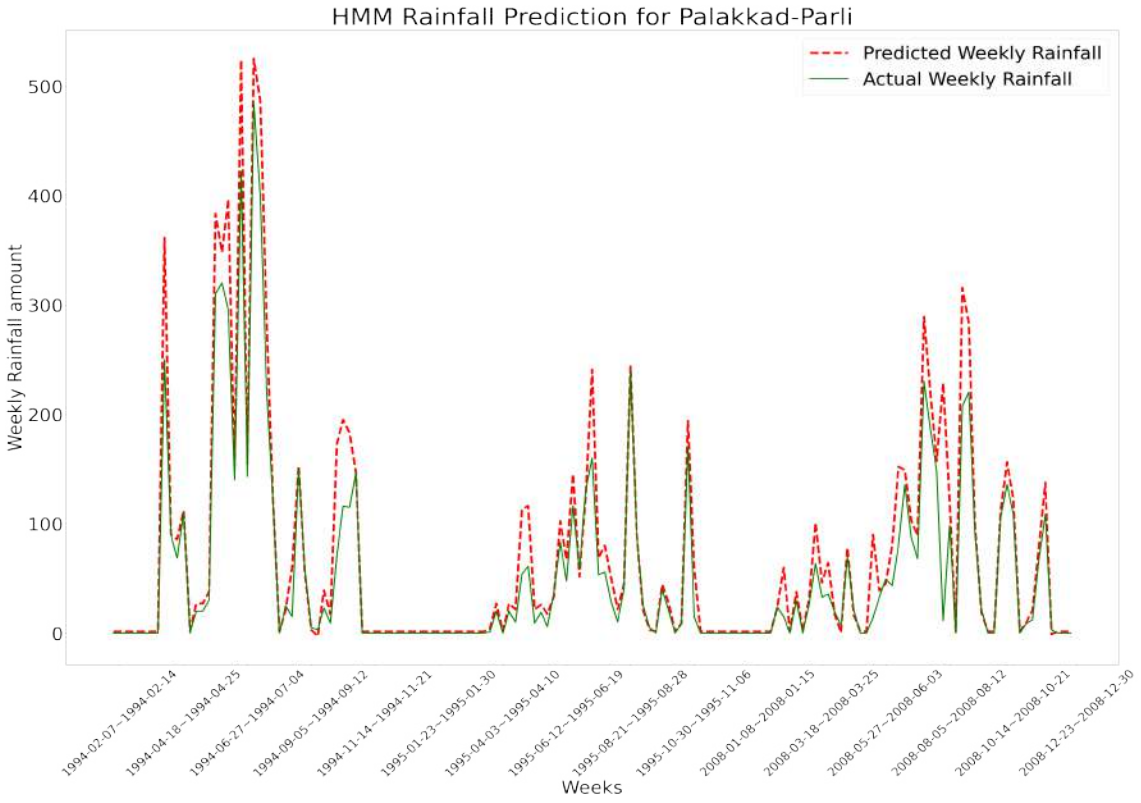
(a)



(b)

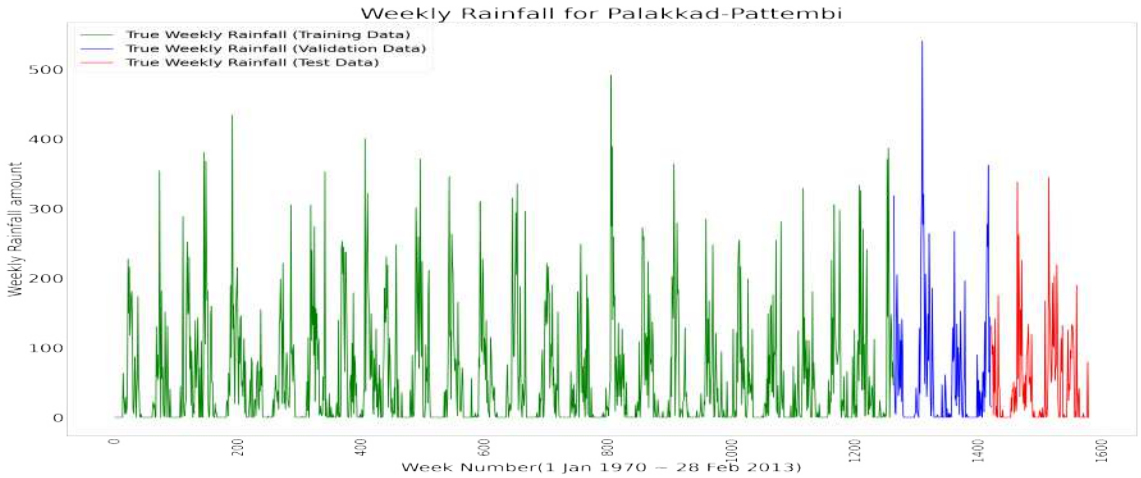


(c)

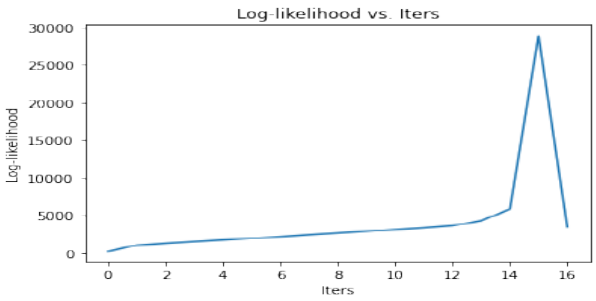


(d)

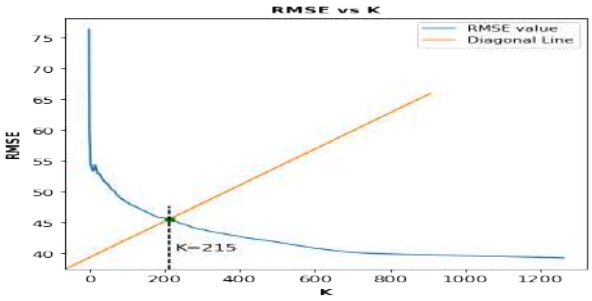
Fig. 6.15 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Parli



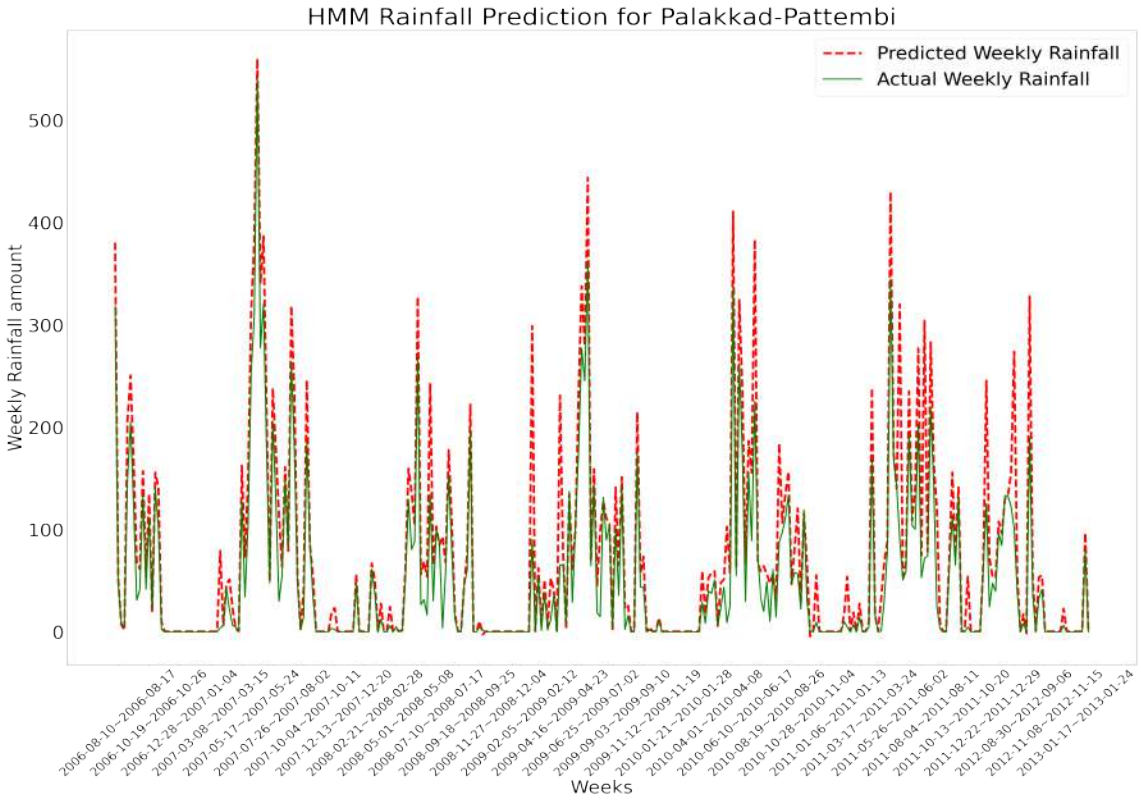
(a)



(b)

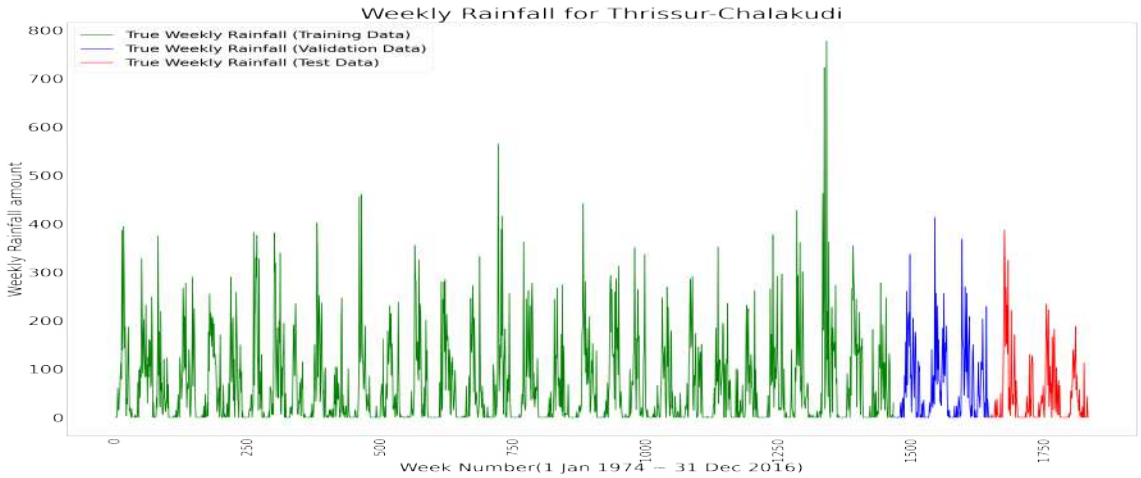


(c)

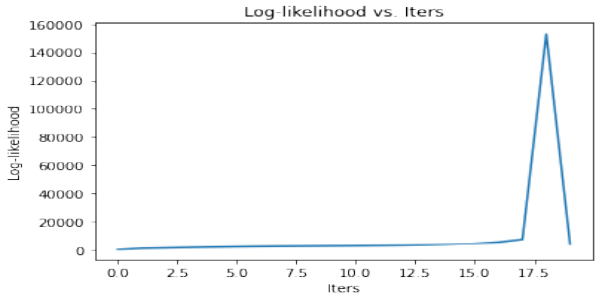


(d)

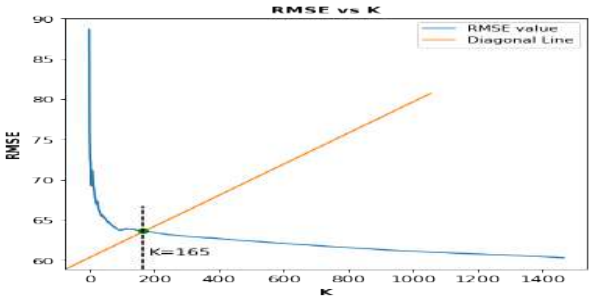
Fig. 6.16 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Pattembi



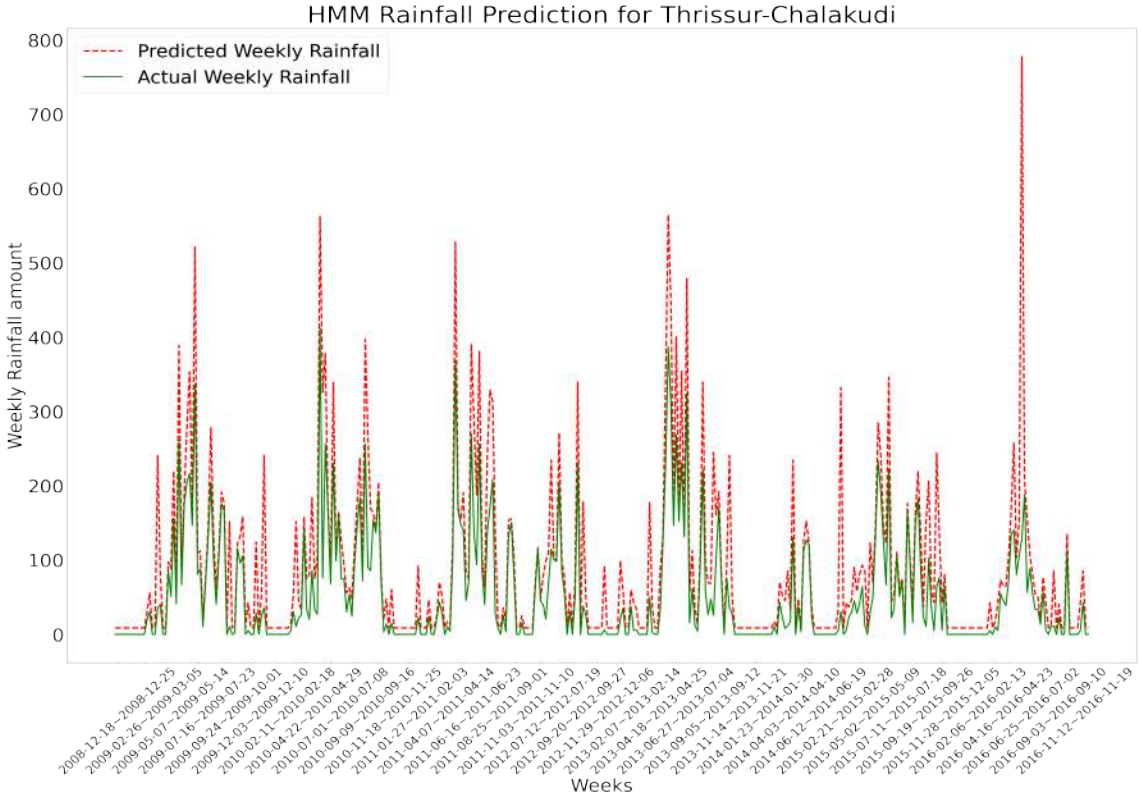
(a)



(b)

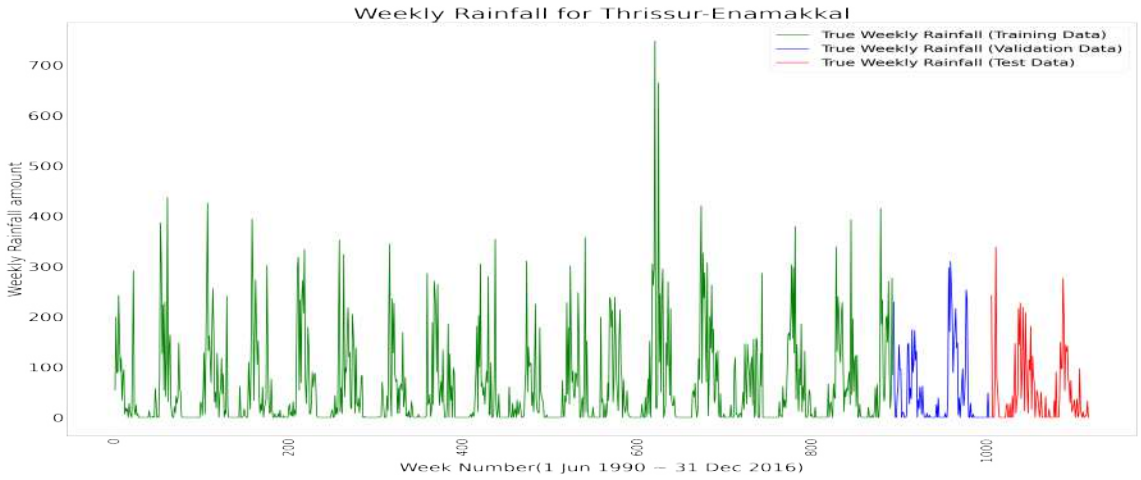


(c)

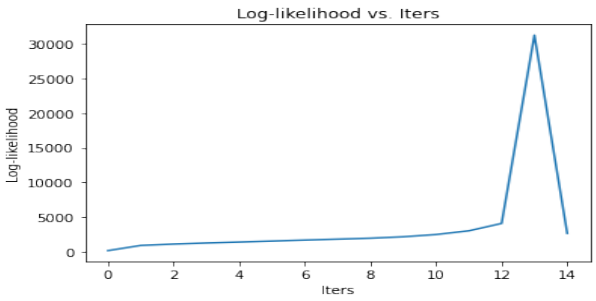


(d)

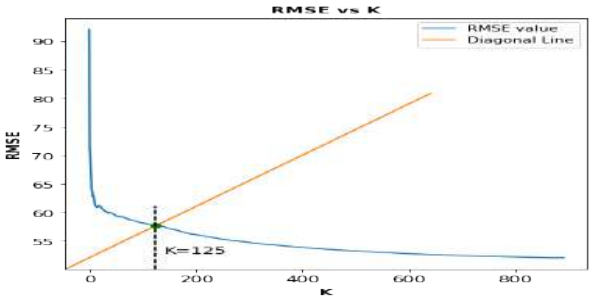
Fig. 6.17 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Chalakudi



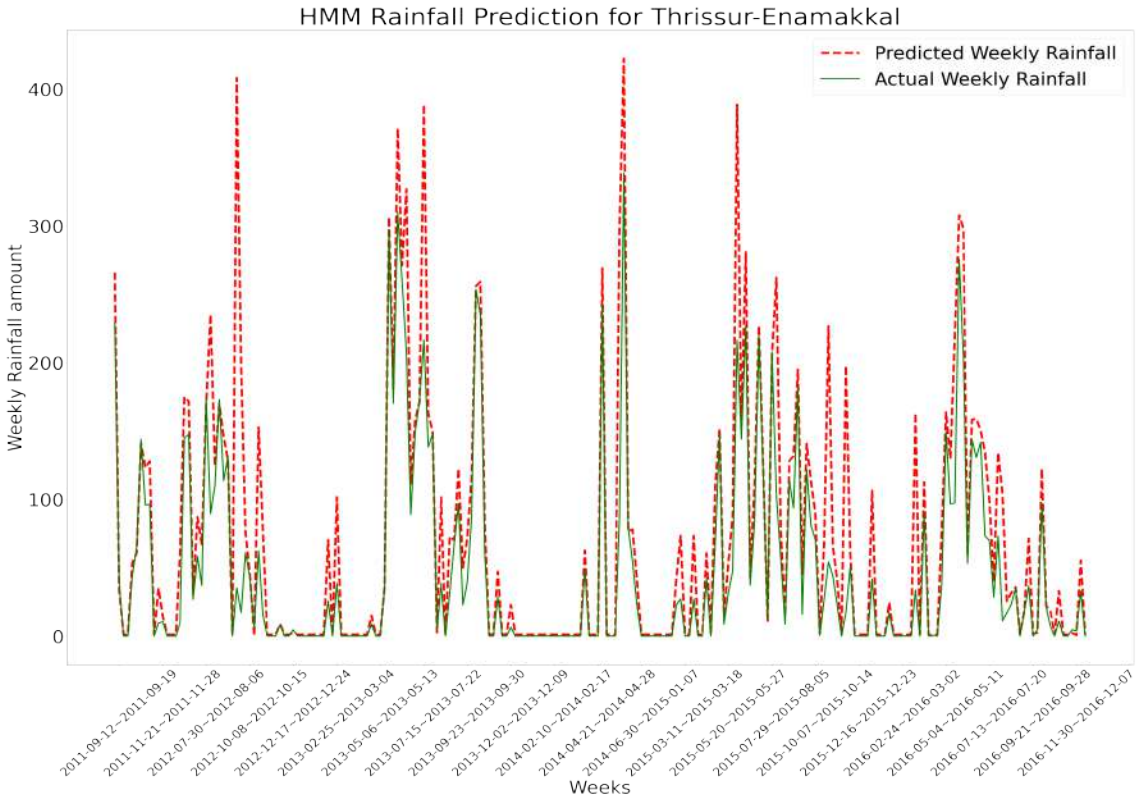
(a)



(b)

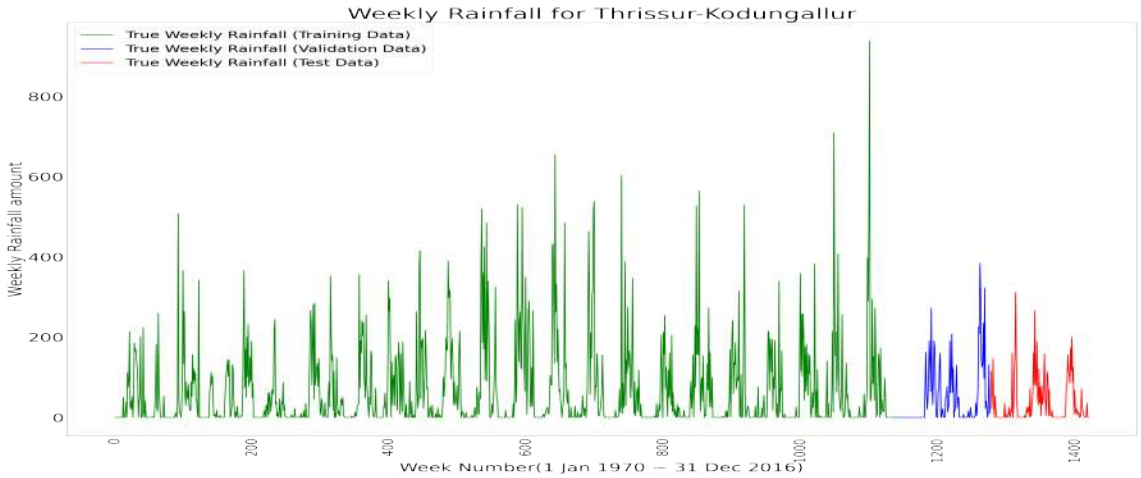


(c)

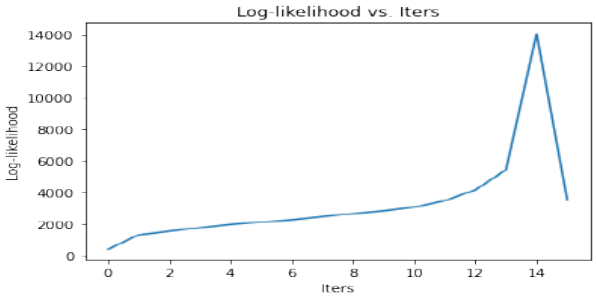


(d)

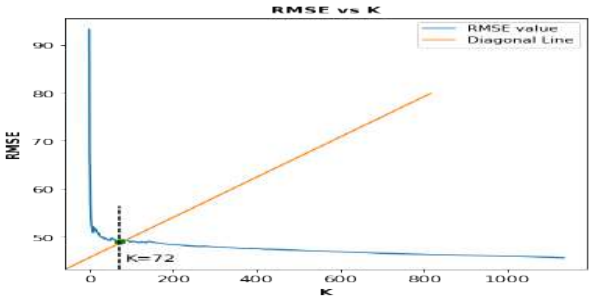
Fig. 6.18 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Enamakkal



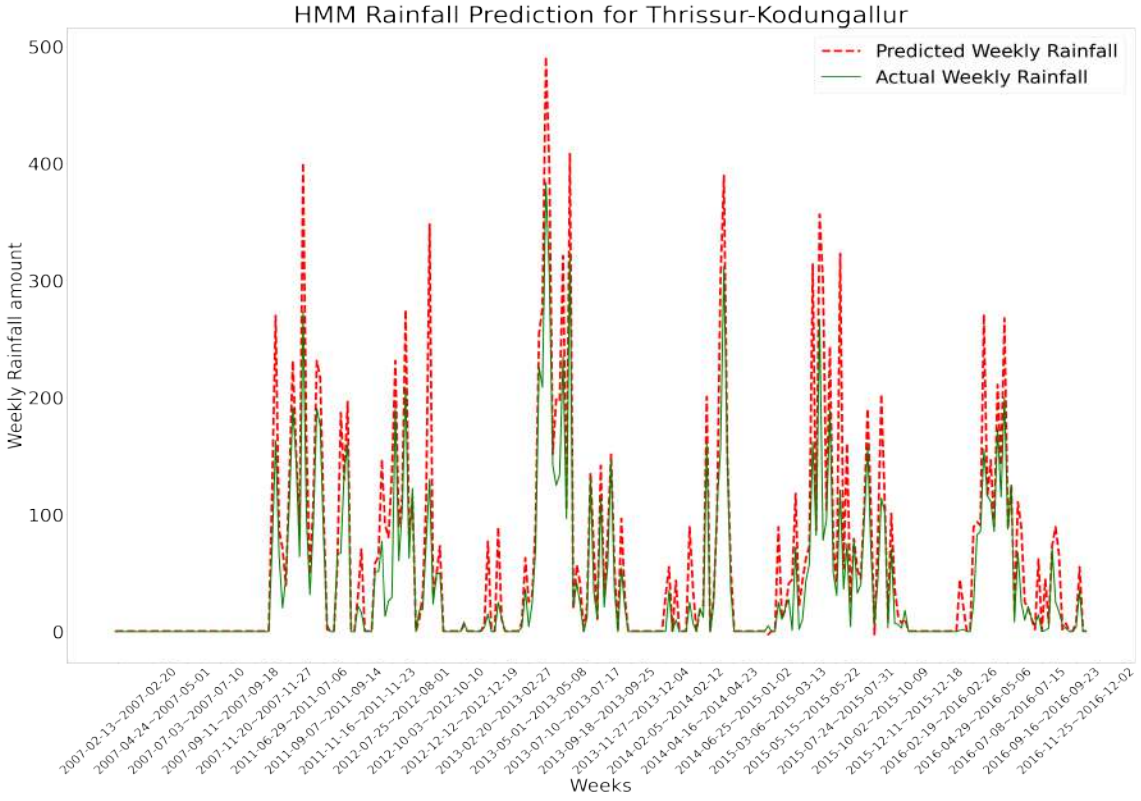
(a)



(b)

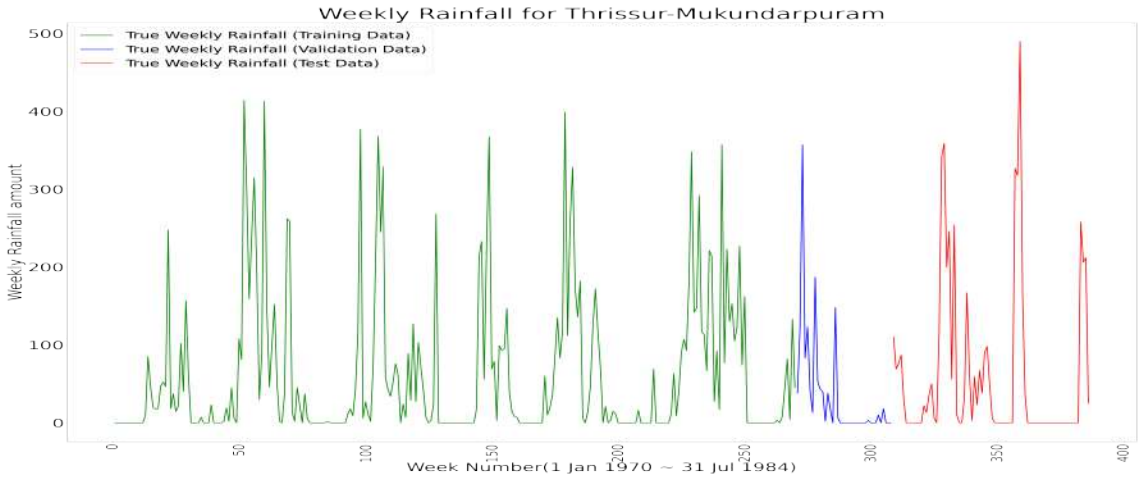


(c)

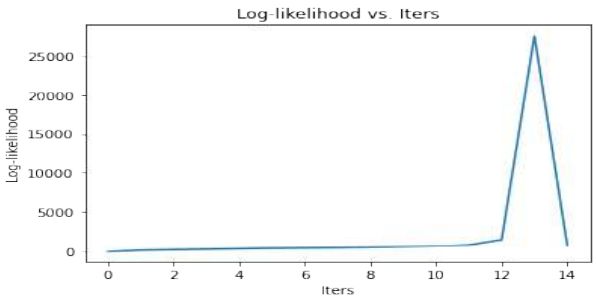


(d)

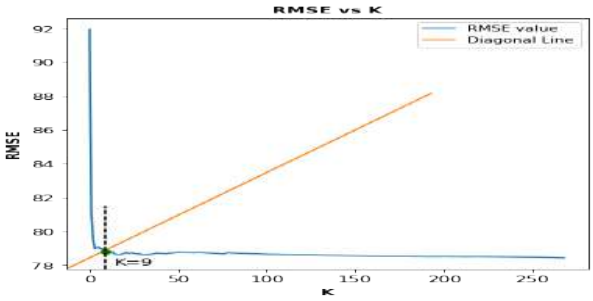
Fig. 6.19 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Kodungallur



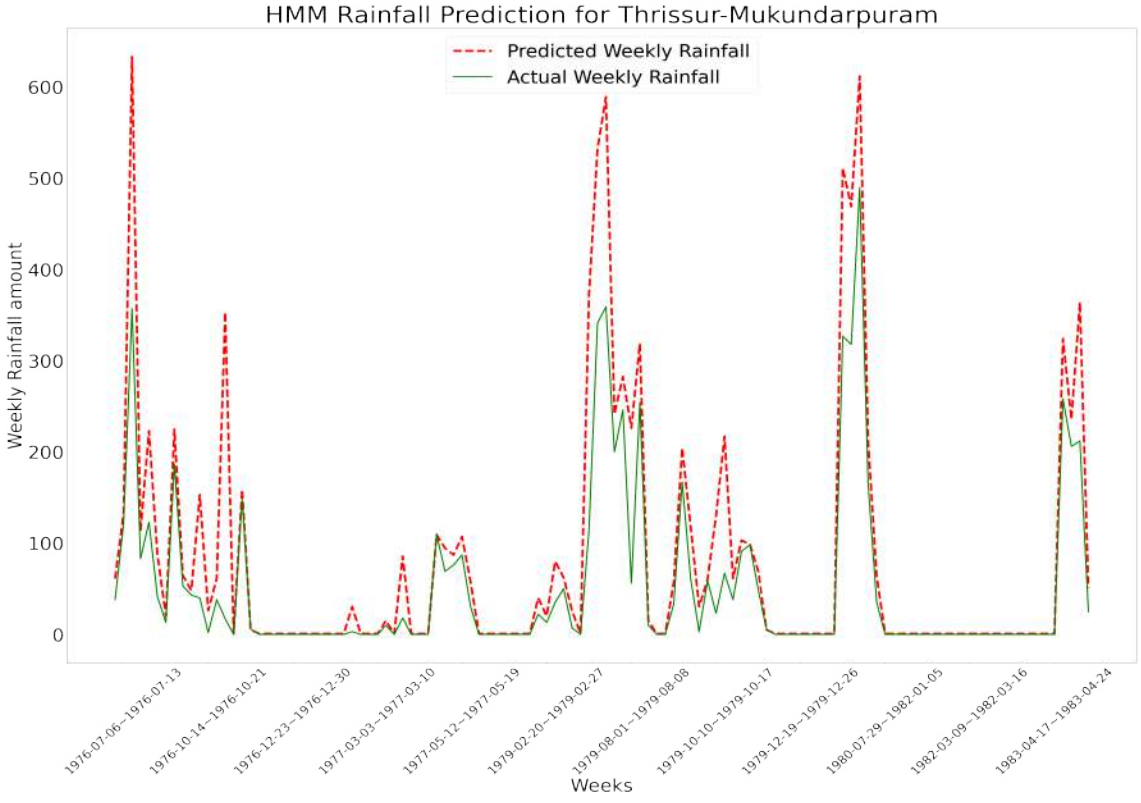
(a)



(b)

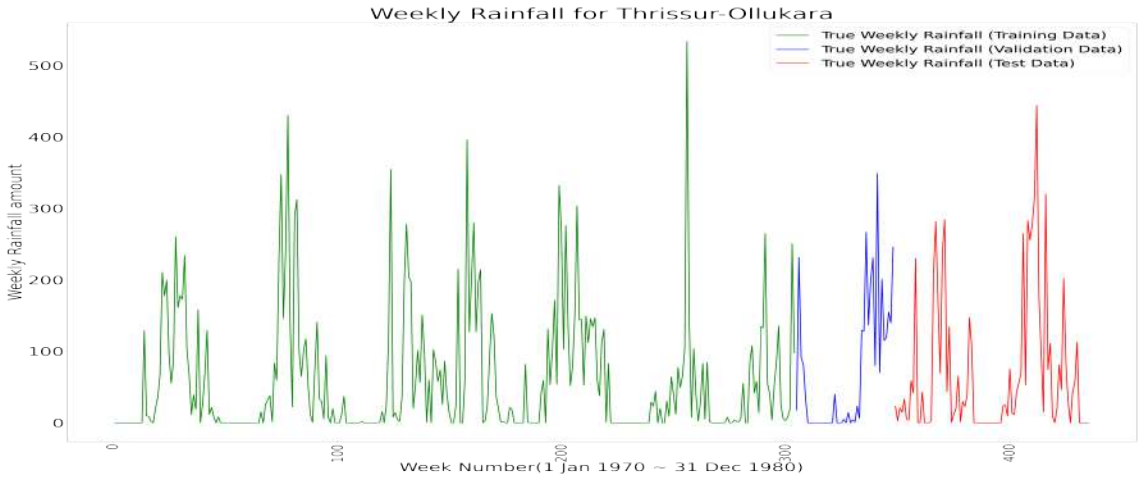


(c)

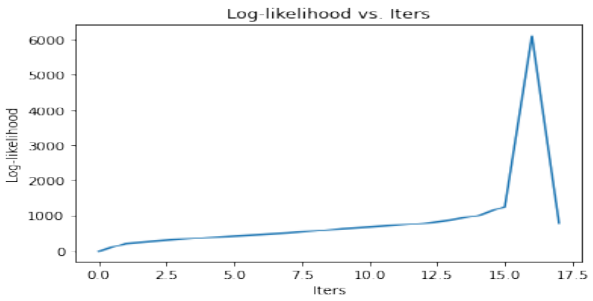


(d)

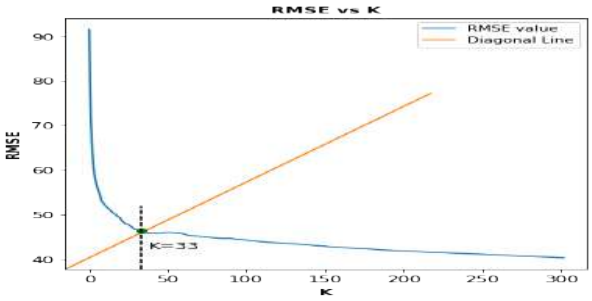
Fig. 6.20 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Mukundarpu-



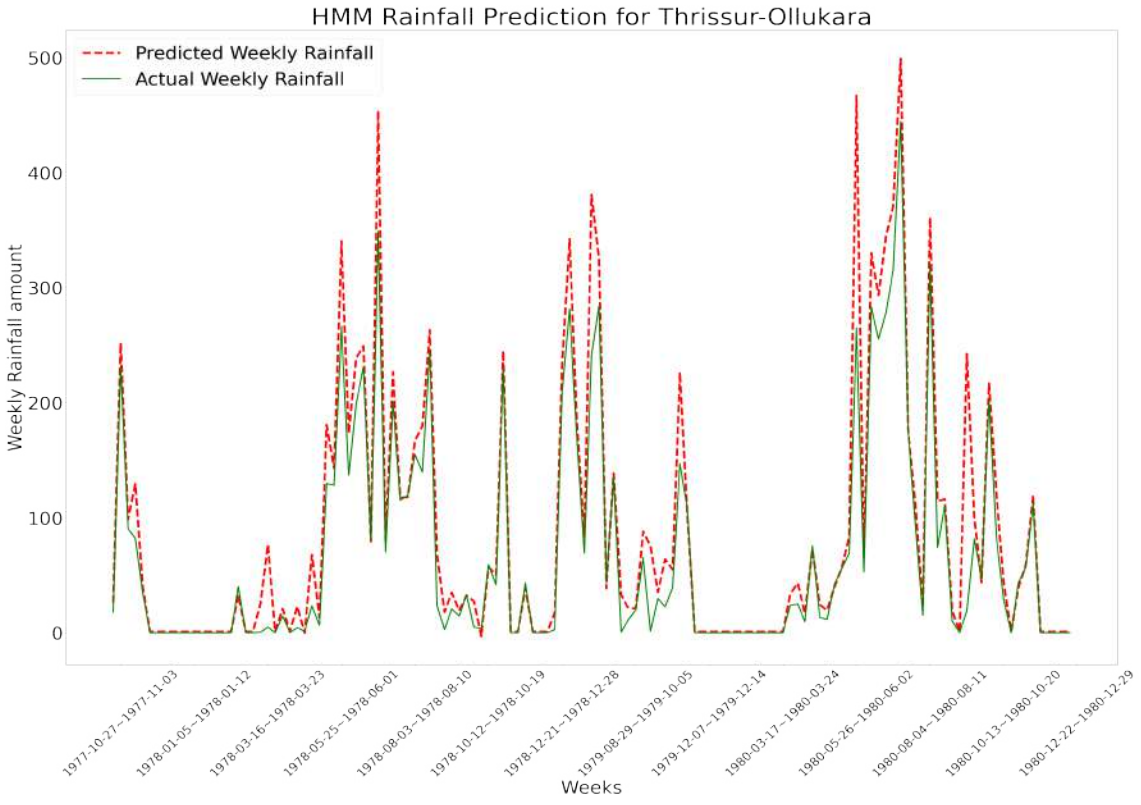
(a)



(b)

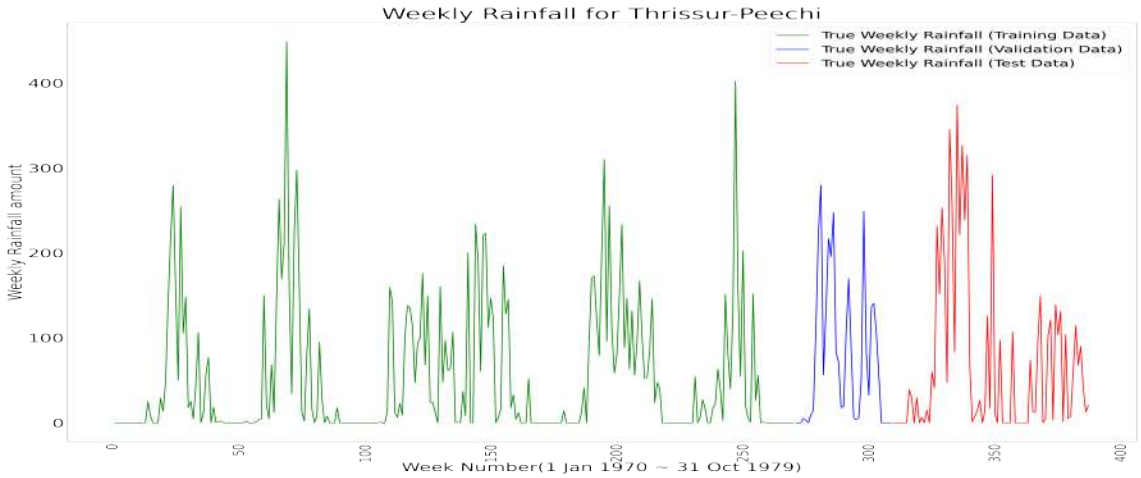


(c)

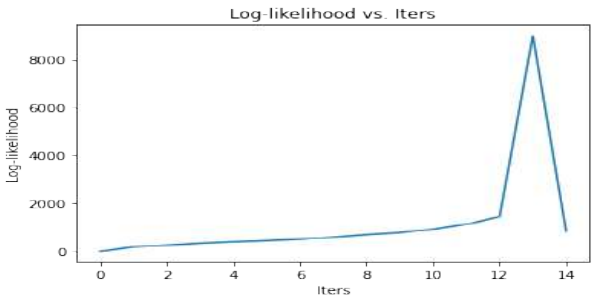


(d)

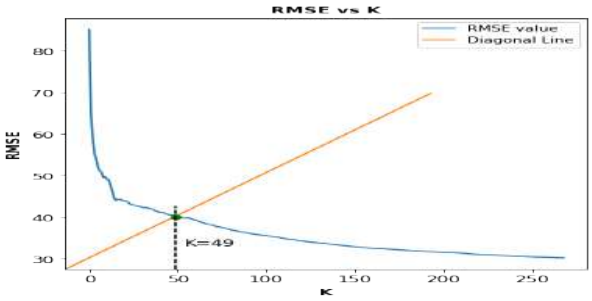
Fig. 6.21 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Ollukara



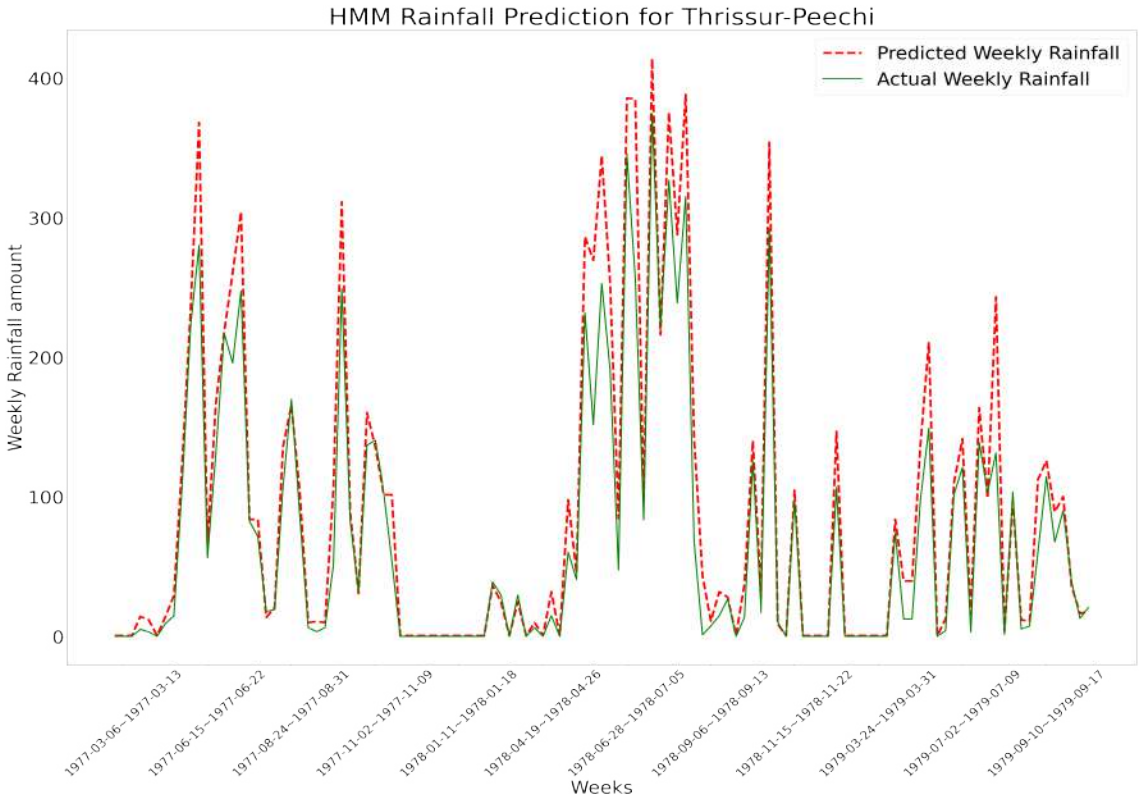
(a)



(b)

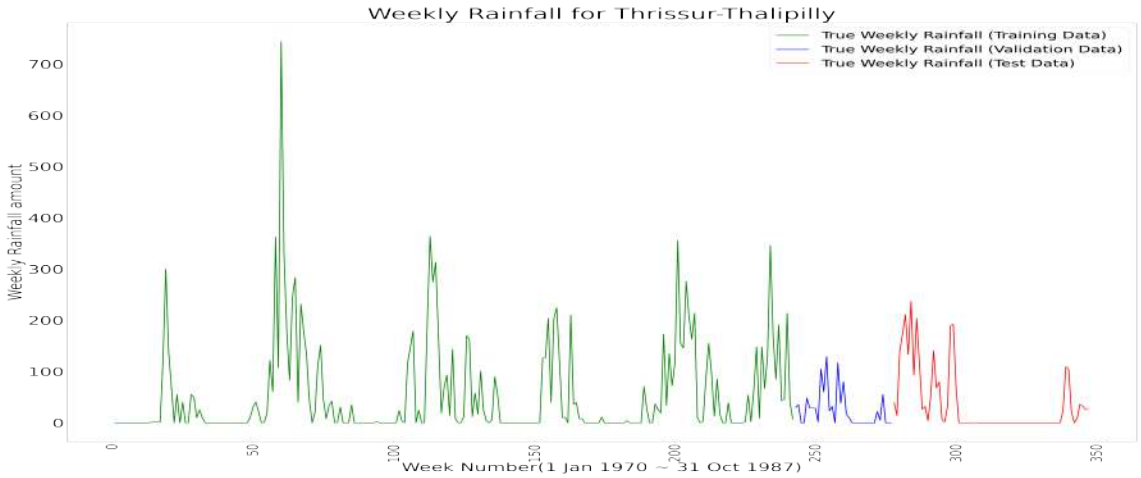


(c)

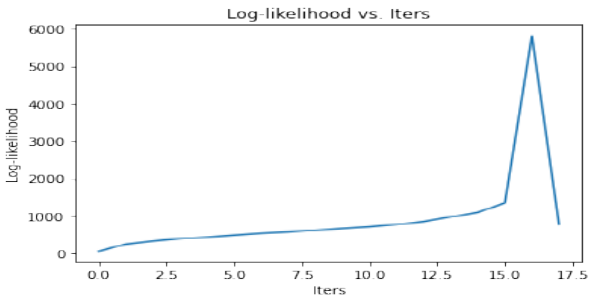


(d)

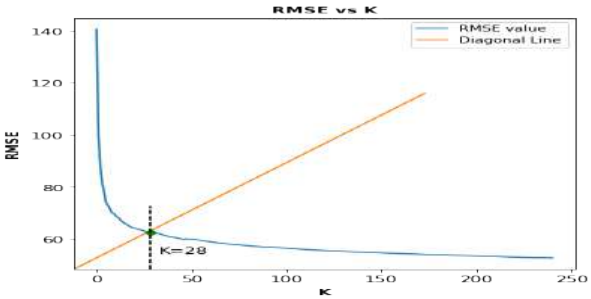
Fig. 6.22 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Peechi



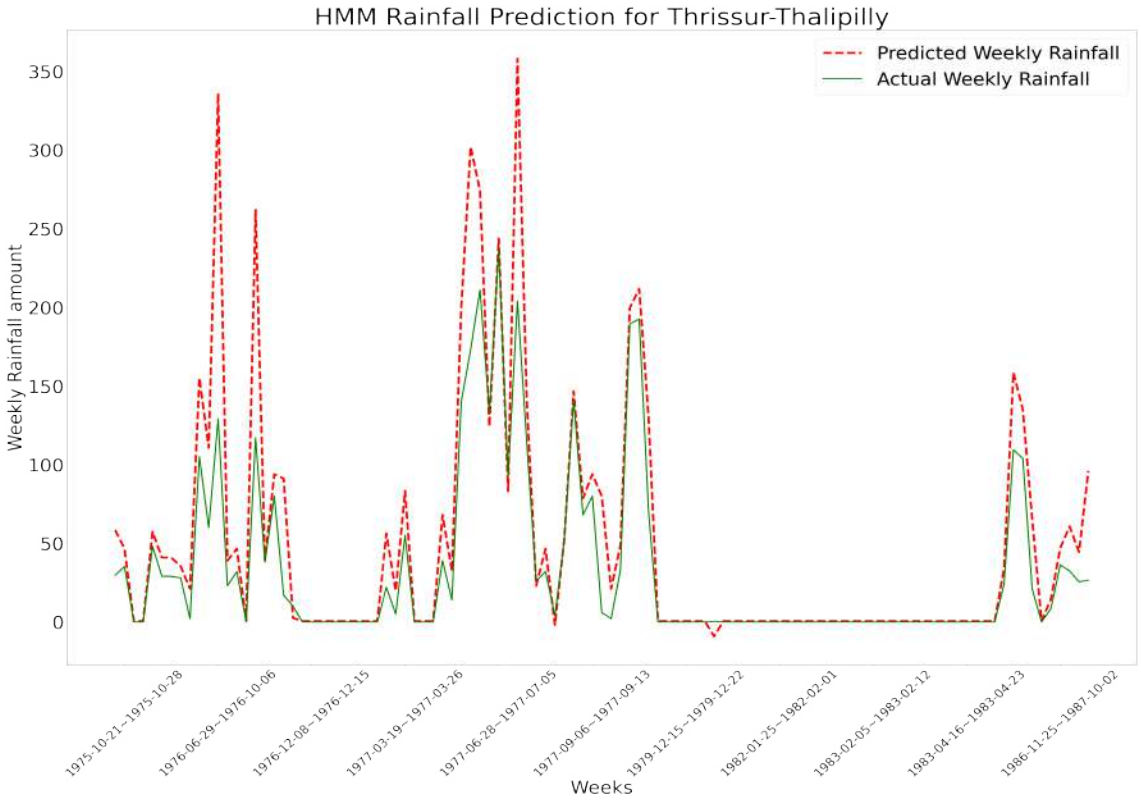
(a)



(b)

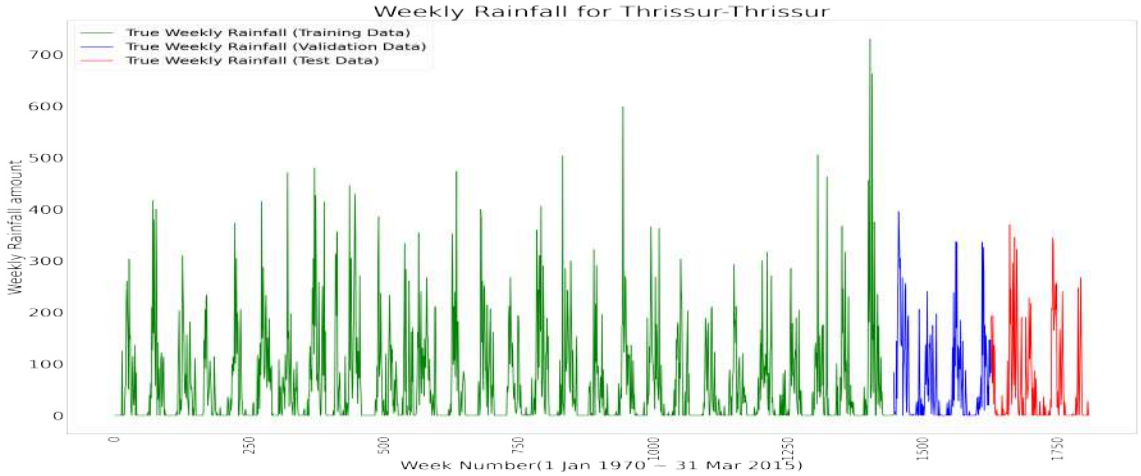


(c)

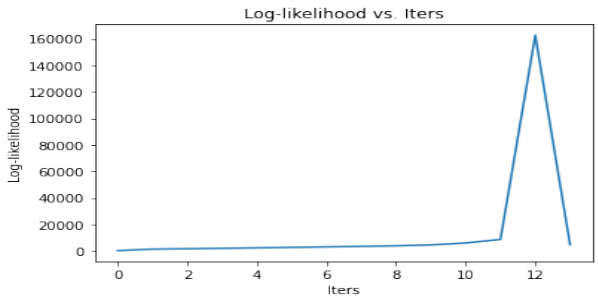


(d)

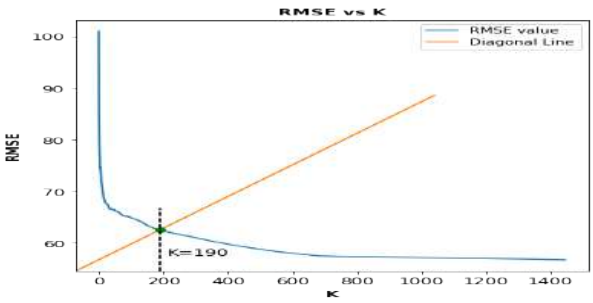
Fig. 6.23 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Thalipilly



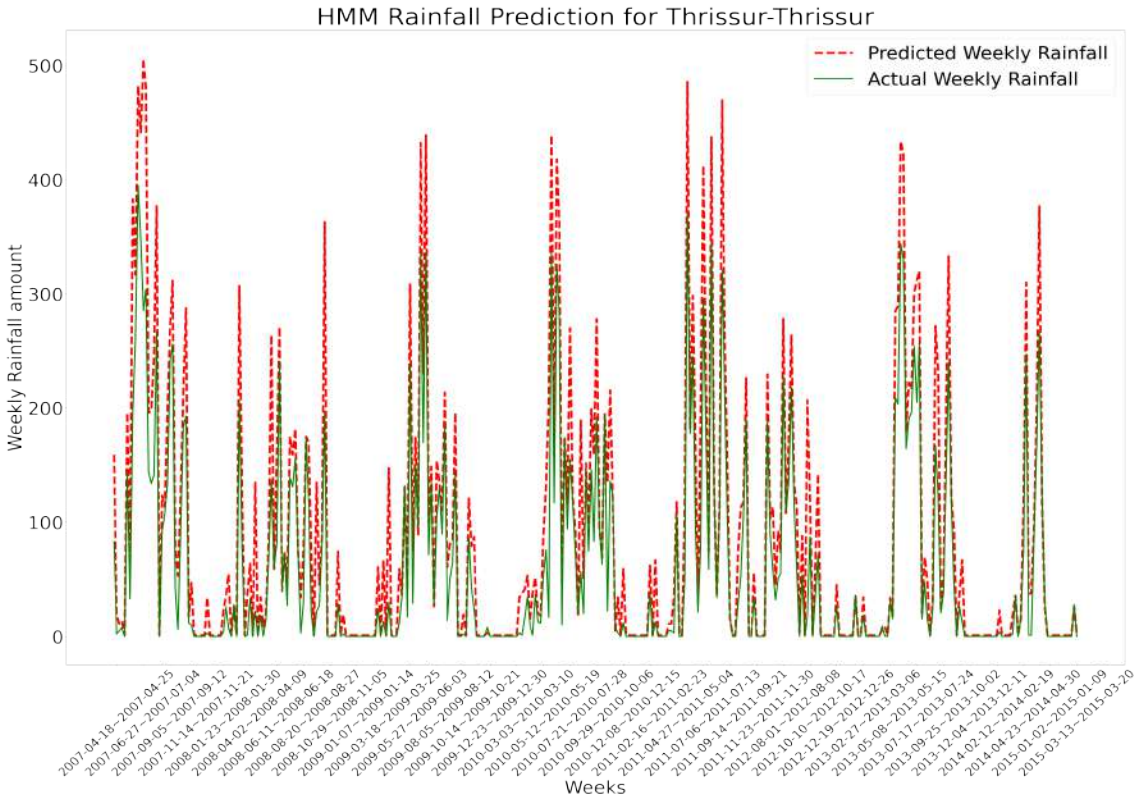
(a)



(b)

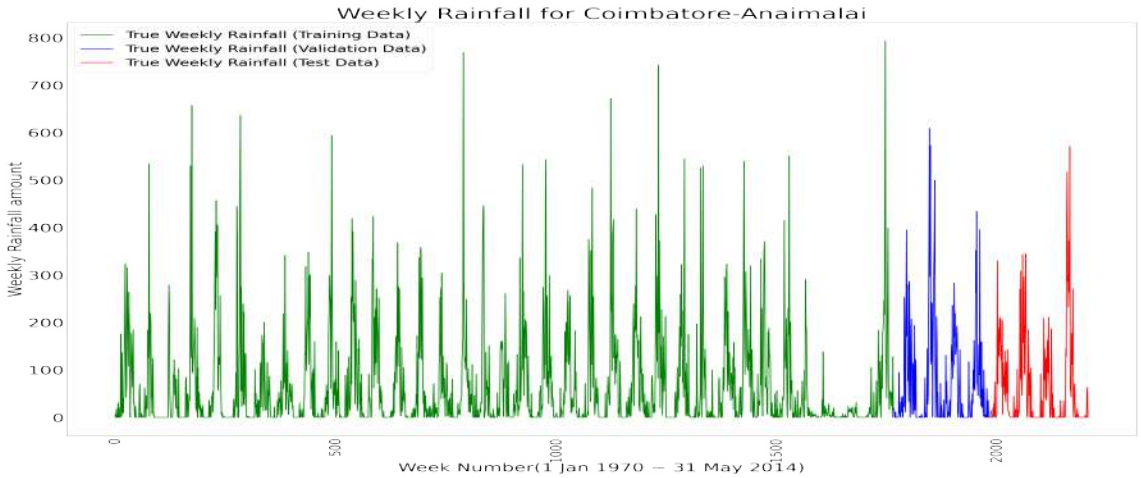


(c)

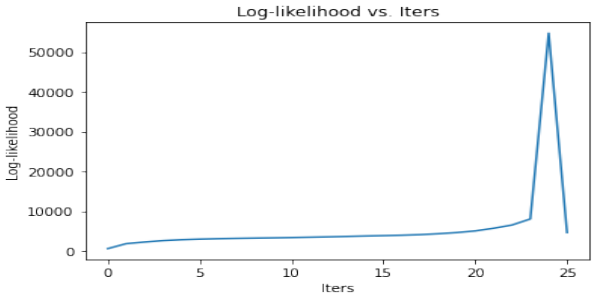


(d)

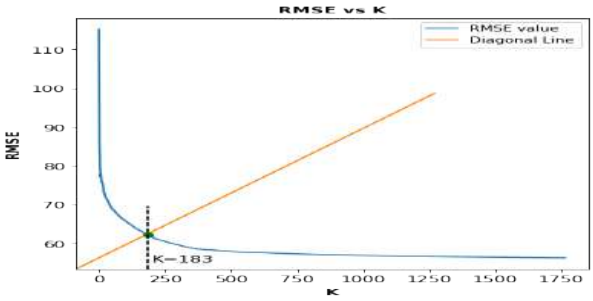
Fig. 6.24 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Thrissur



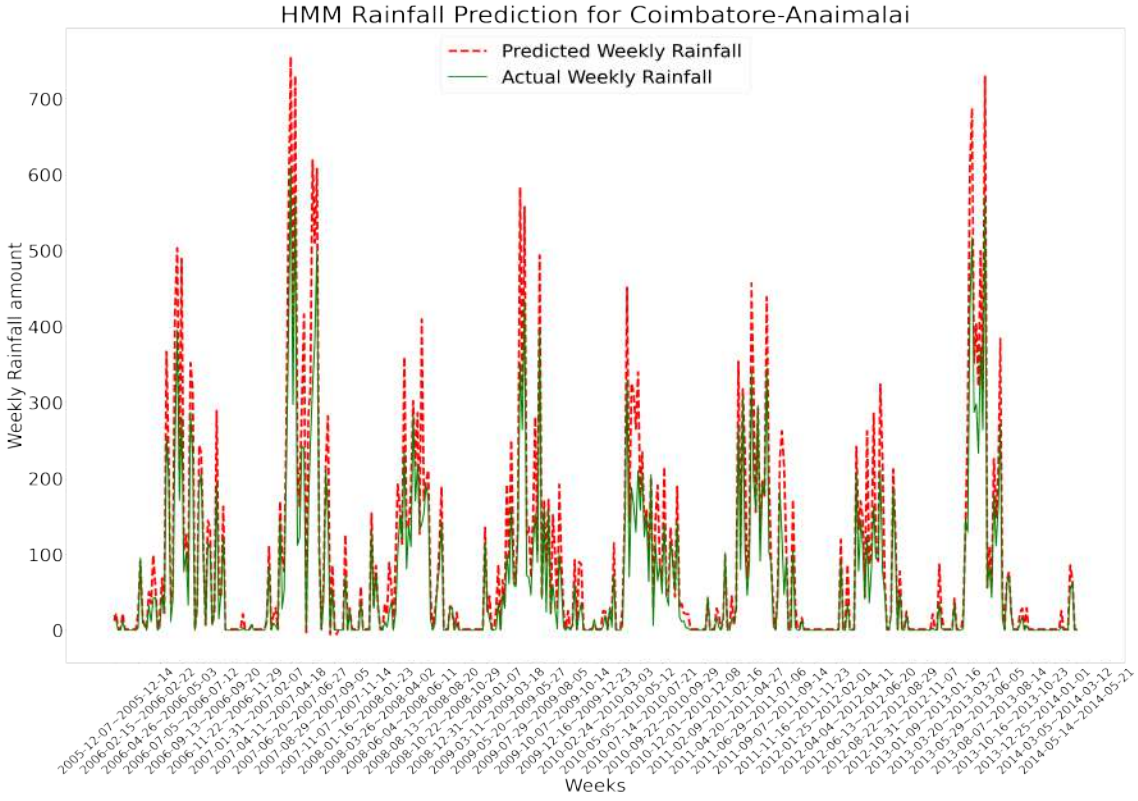
(a)



(b)

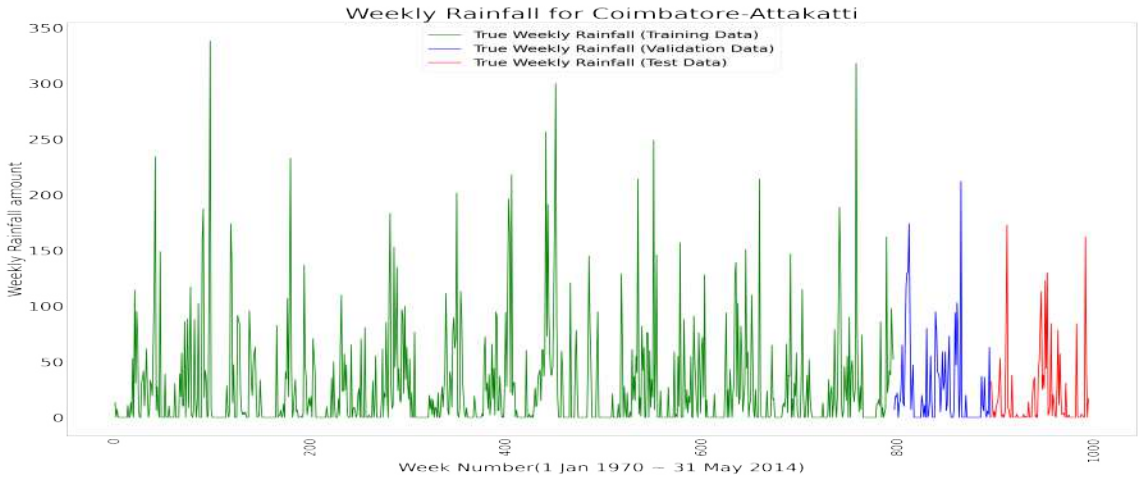


(c)

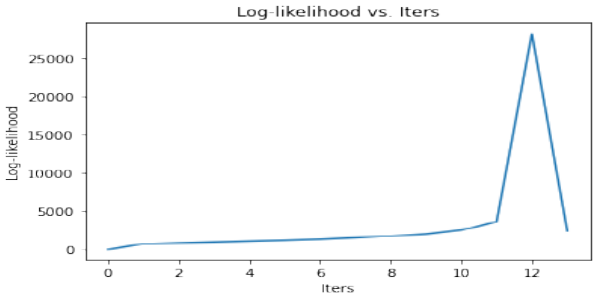


(d)

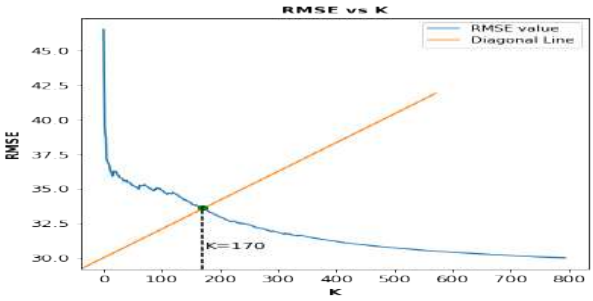
Fig. 6.25 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Anaimalai



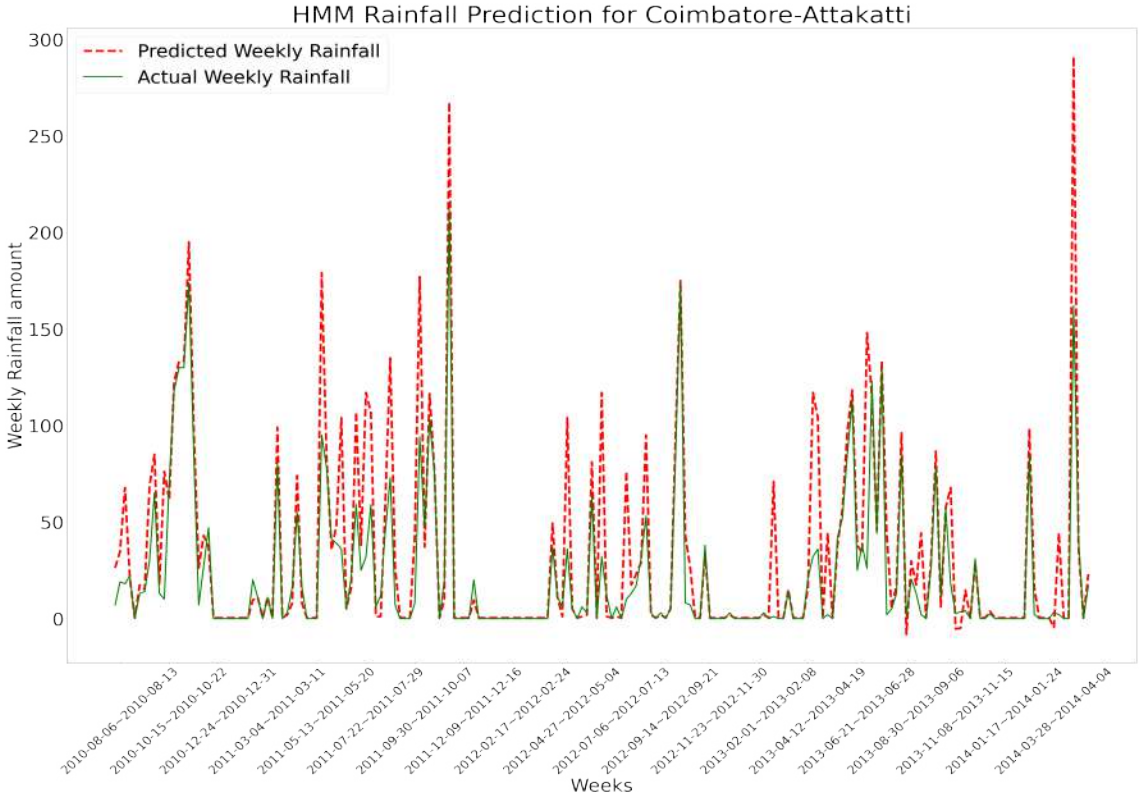
(a)



(b)

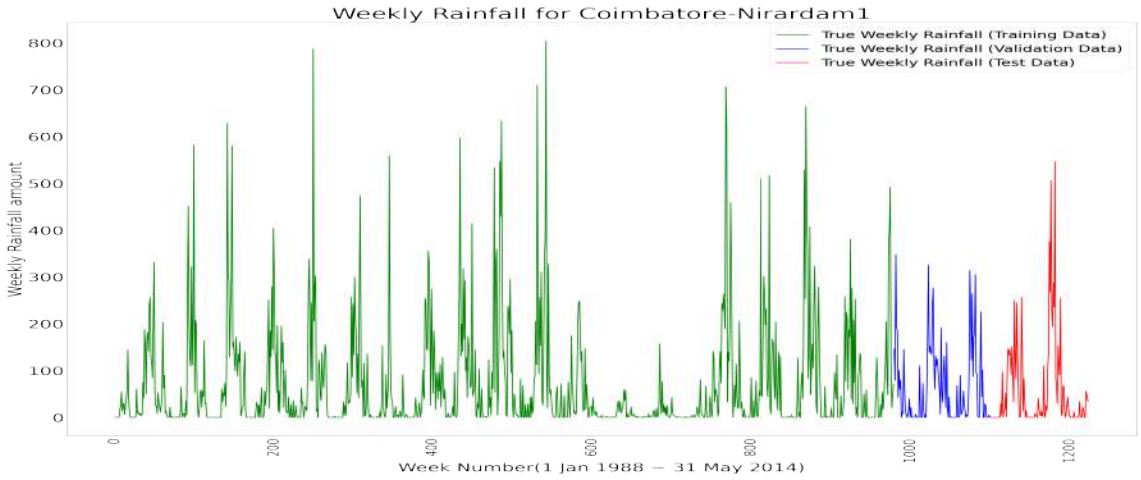


(c)

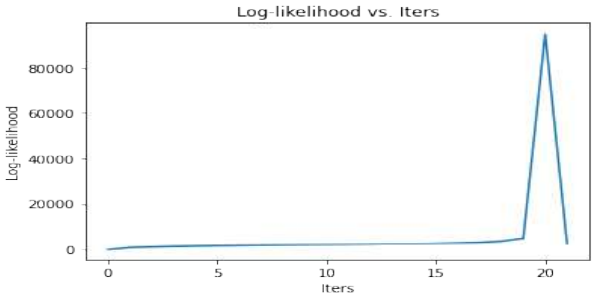


(d)

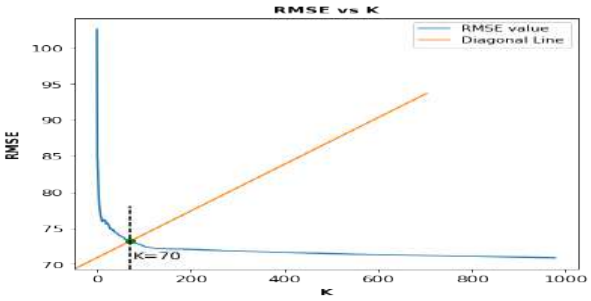
Fig. 6.26 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Attakatti



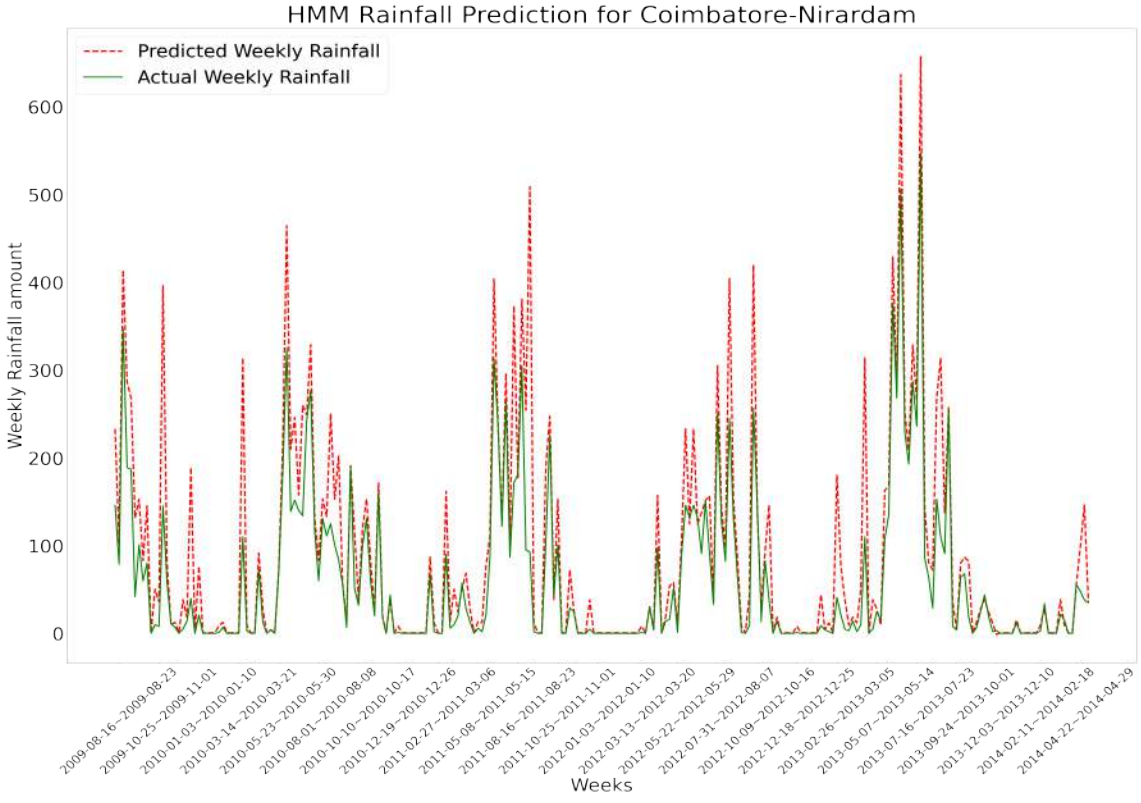
(a)



(b)

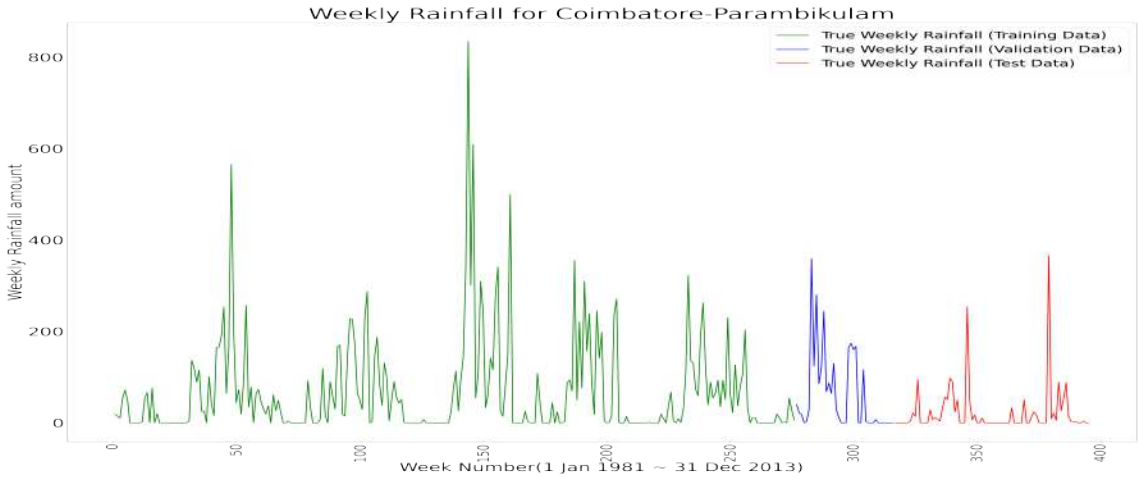


(c)

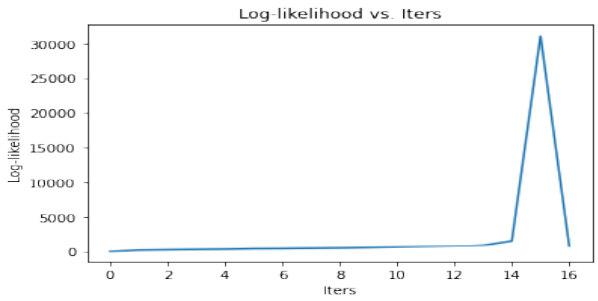


(d)

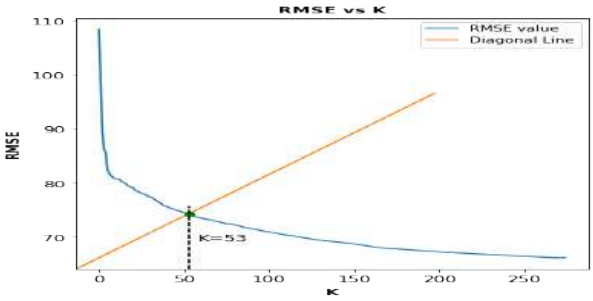
Fig. 6.27 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Nirardam



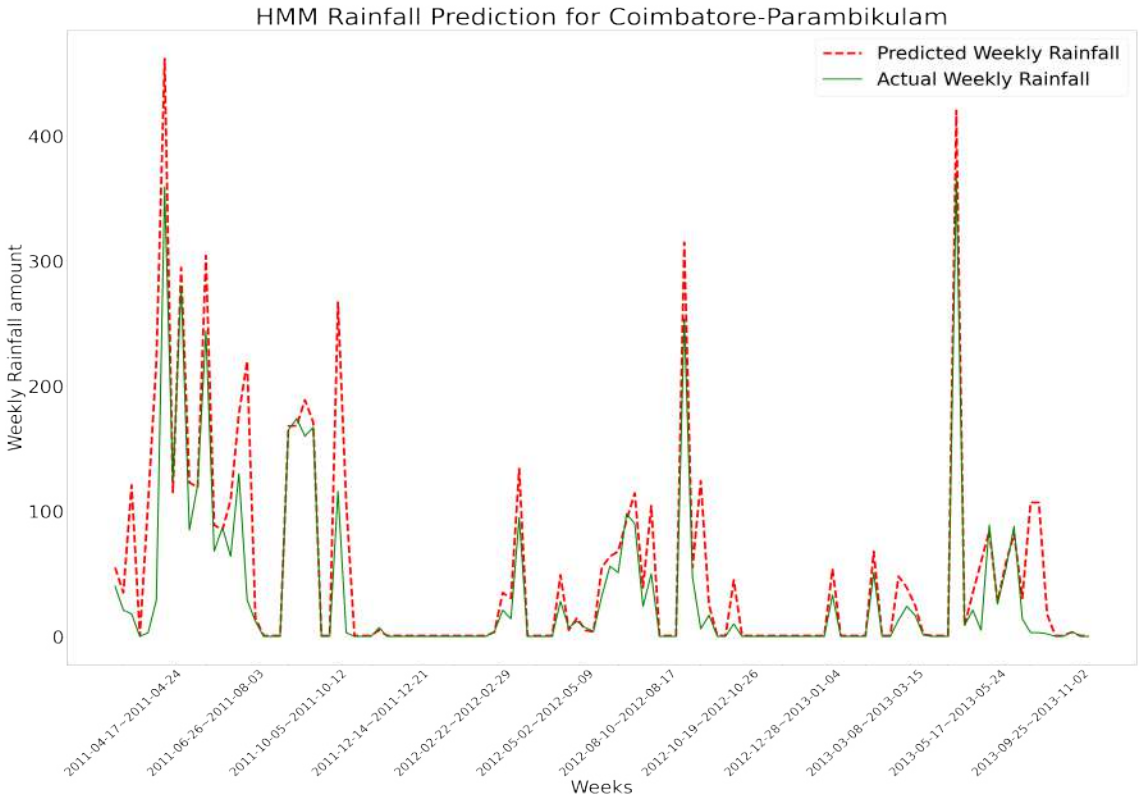
(a)



(b)

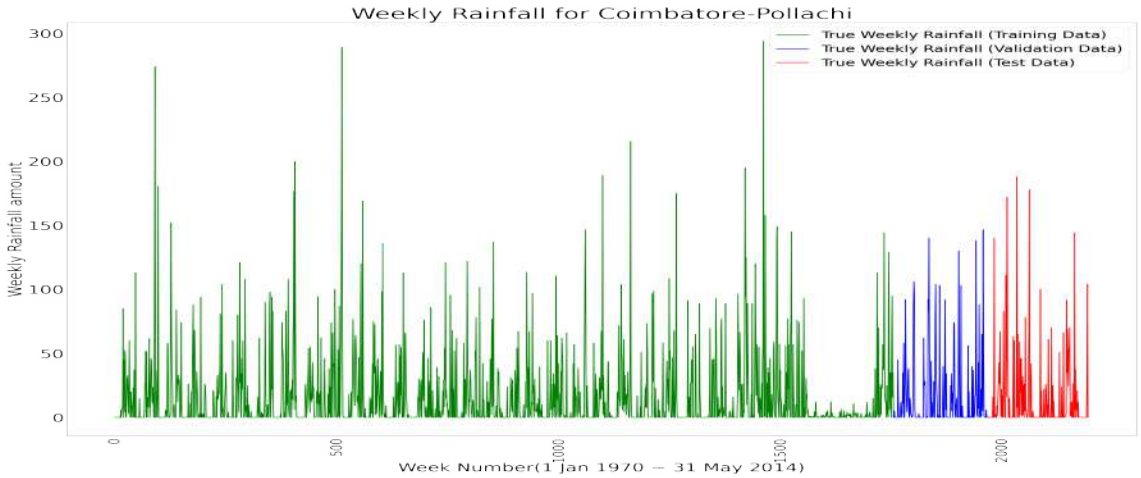


(c)

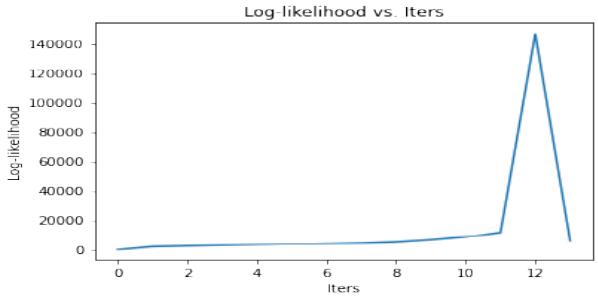


(d)

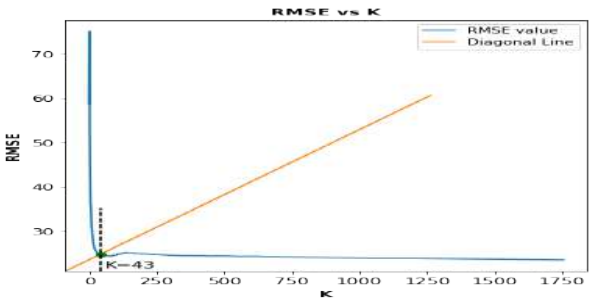
Fig. 6.28 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Parambikulam



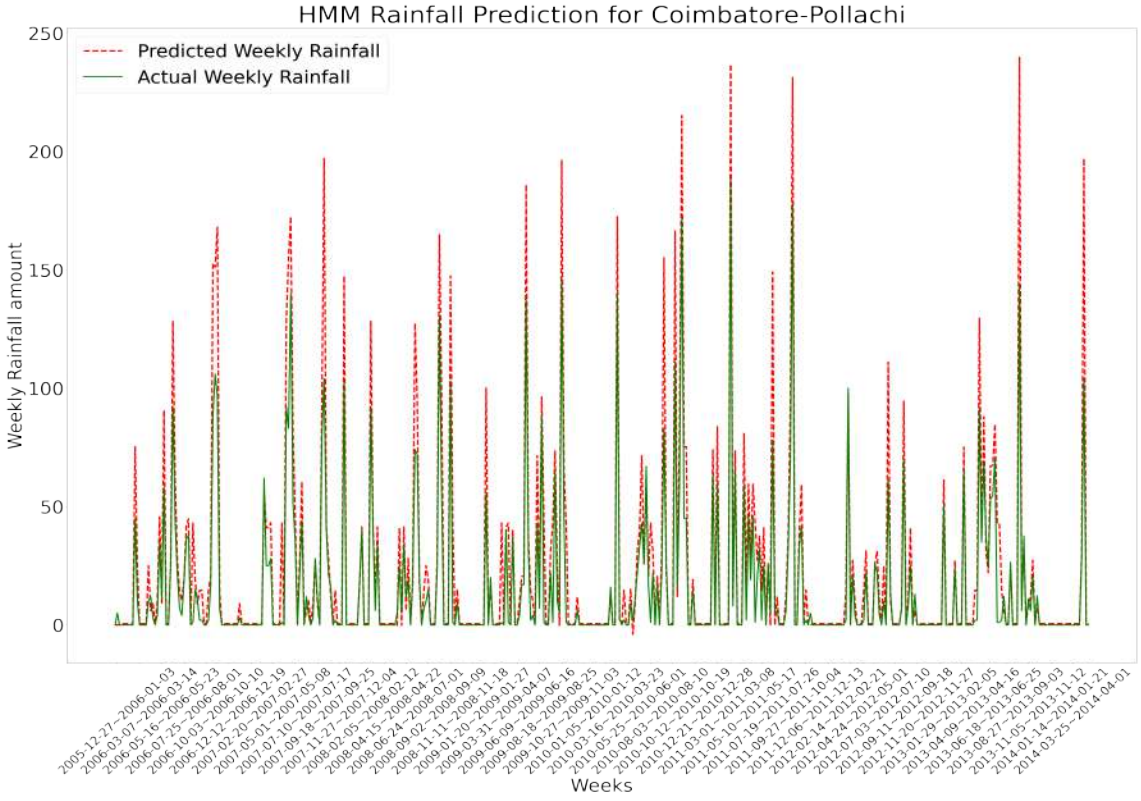
(a)



(b)

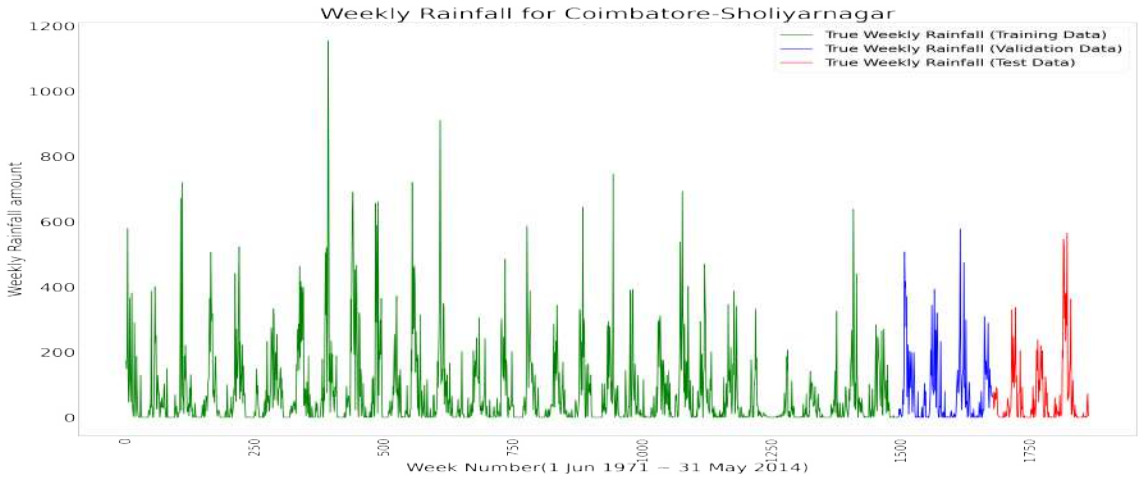


(c)

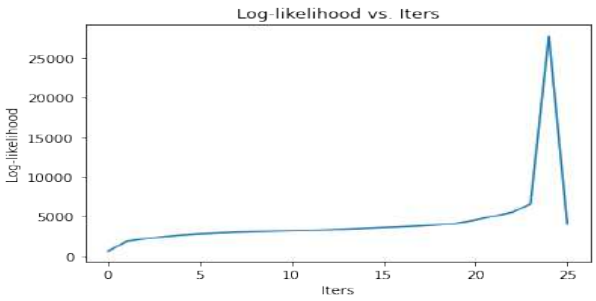


(d)

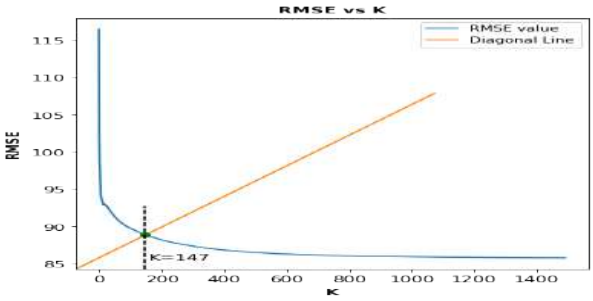
Fig. 6.29 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Pollachi



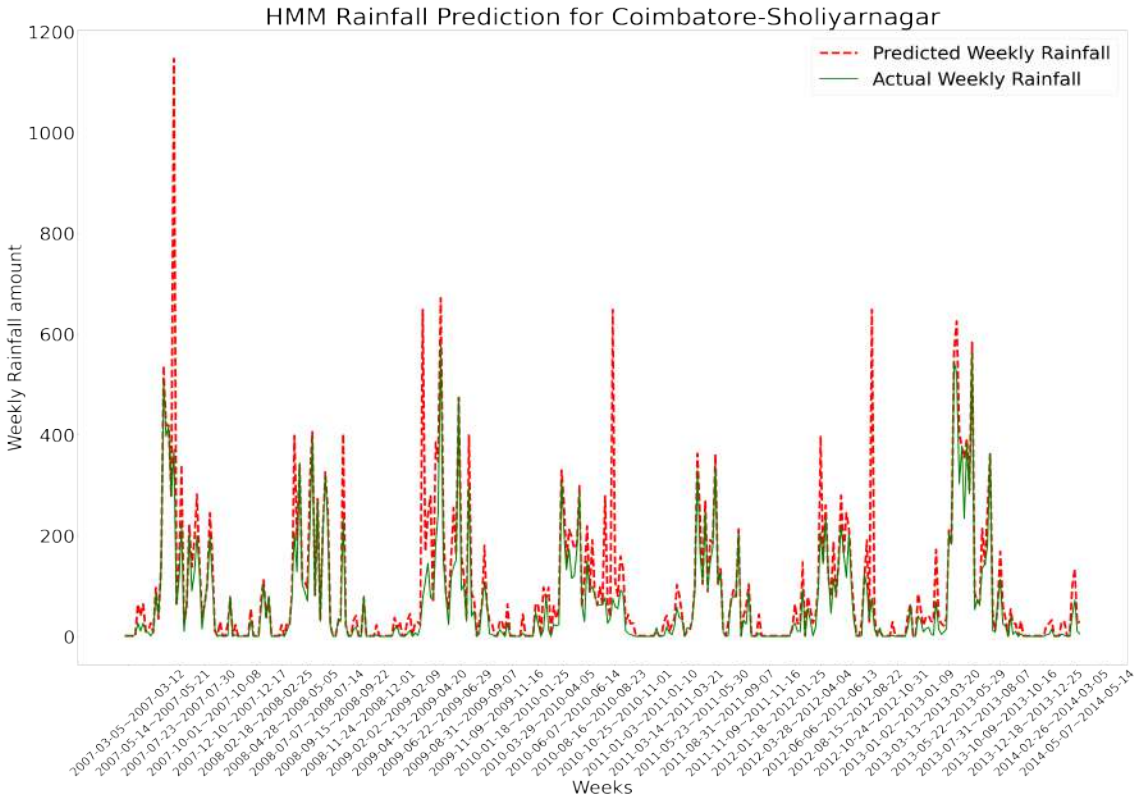
(a)



(b)

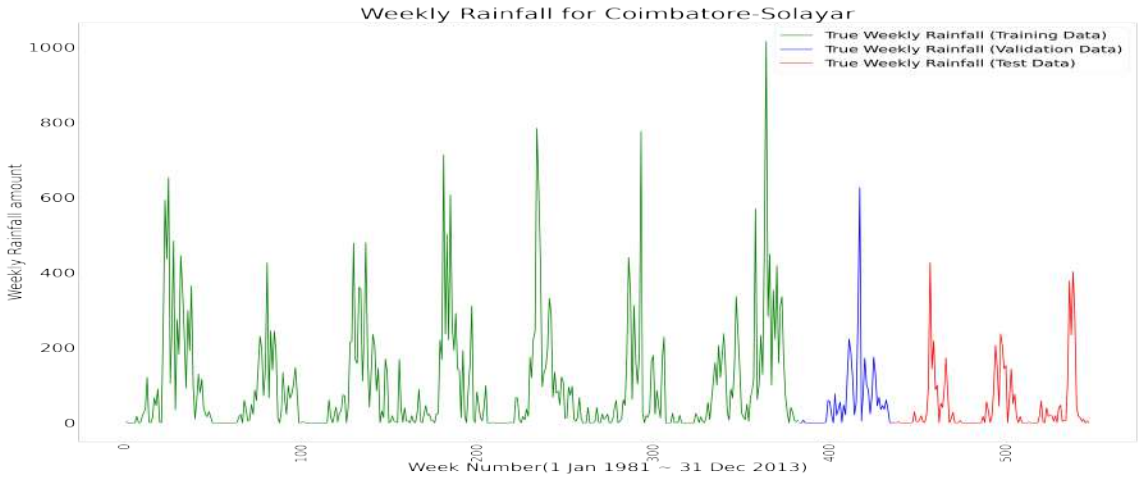


(c)

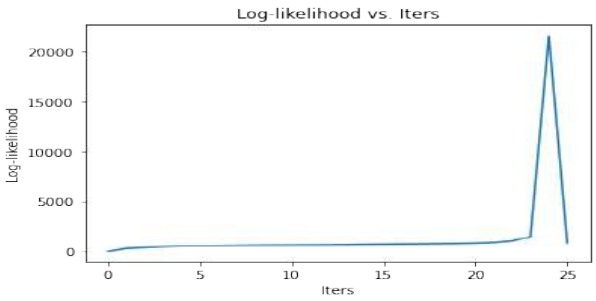


(d)

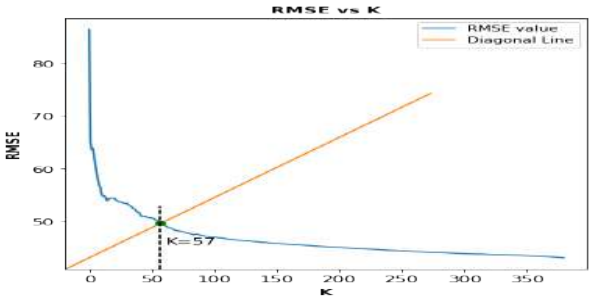
Fig. 6.30 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Sholiyarnagar



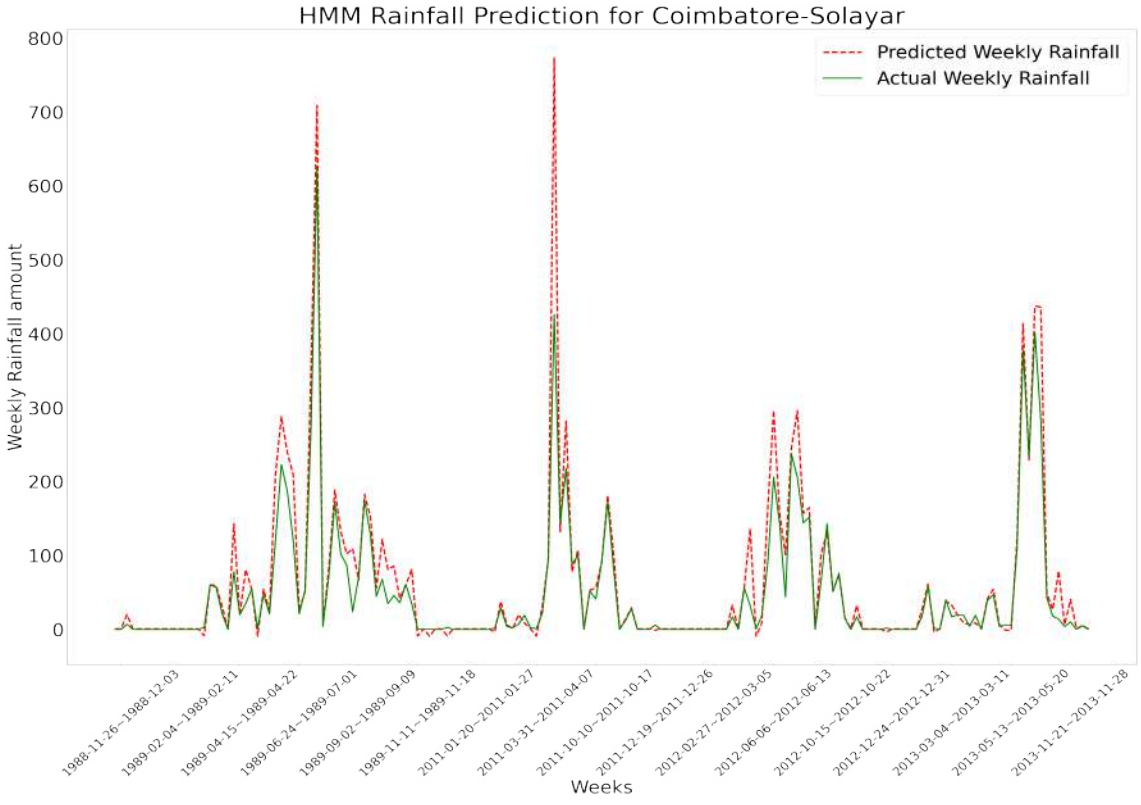
(a)



(b)

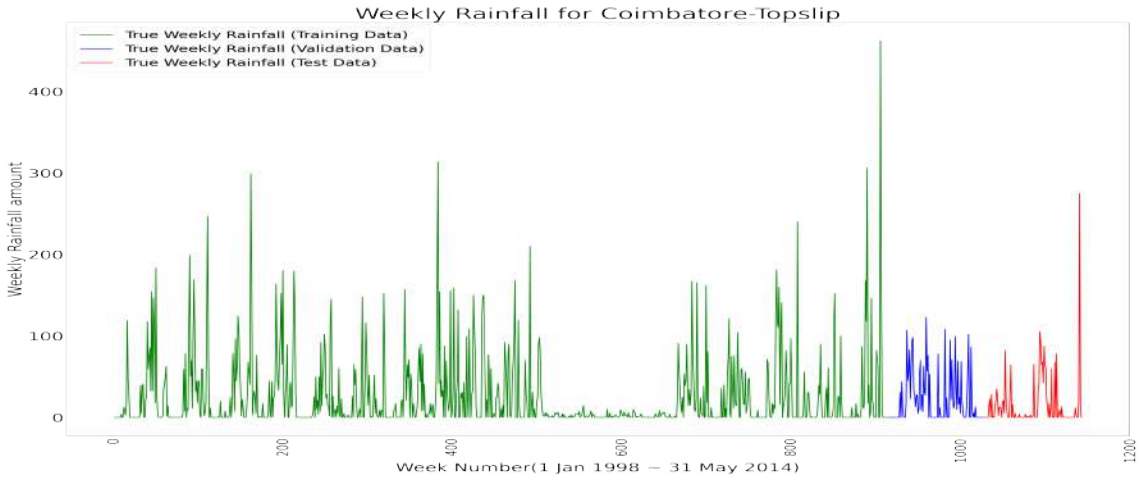


(c)

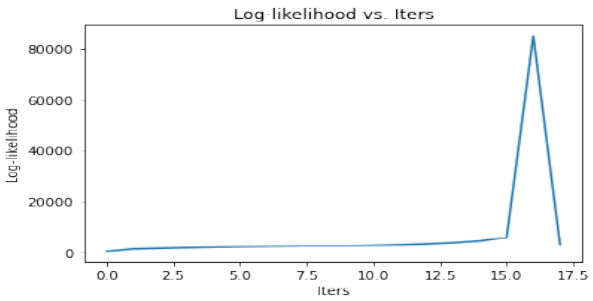


(d)

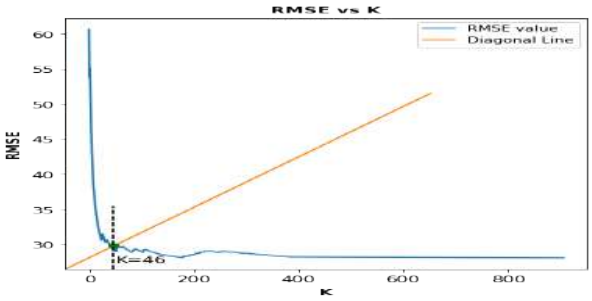
Fig. 6.31 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Solayar



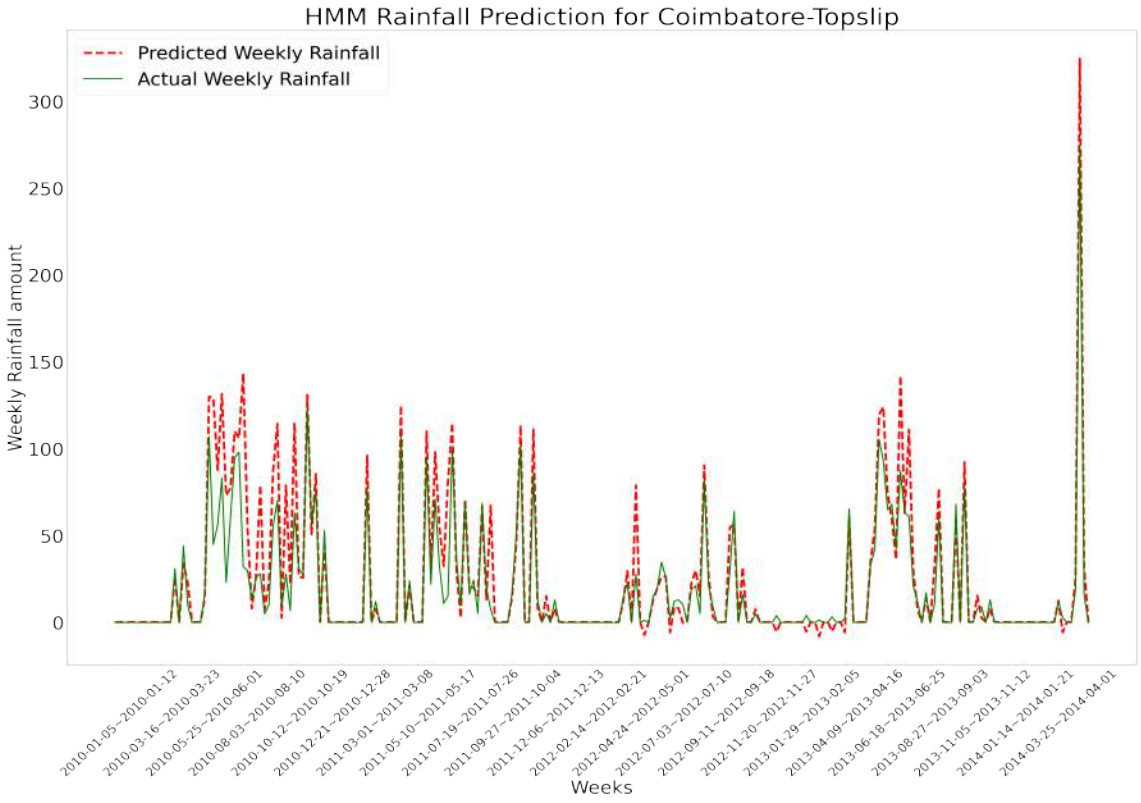
(a)



(b)

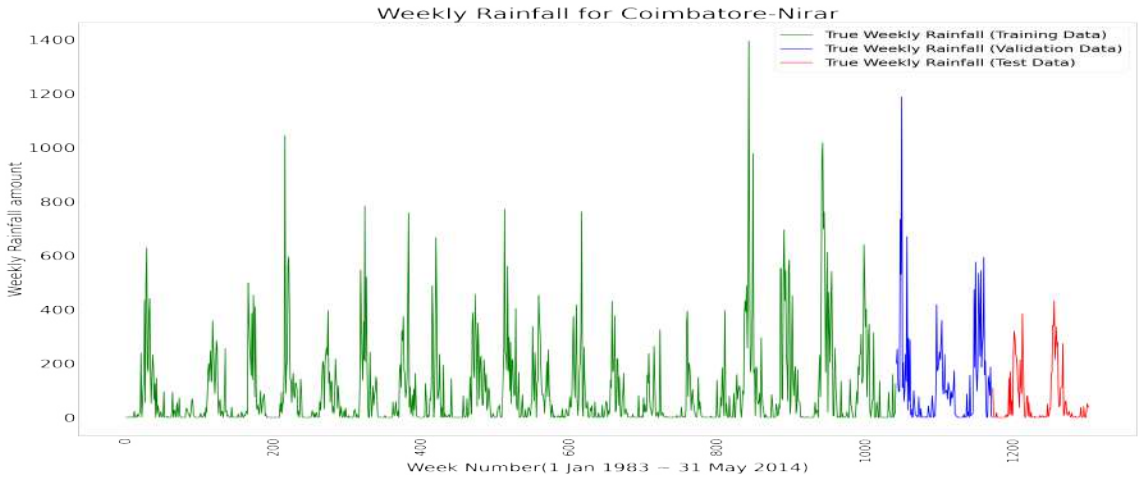


(c)

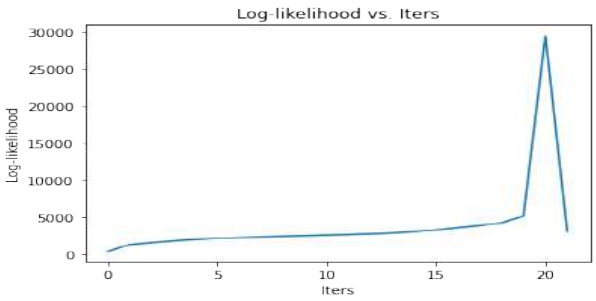


(d)

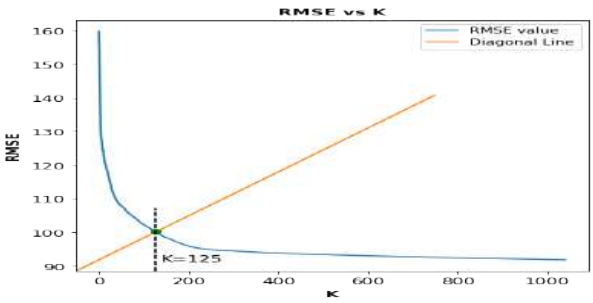
Fig. 6.32 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Topslip



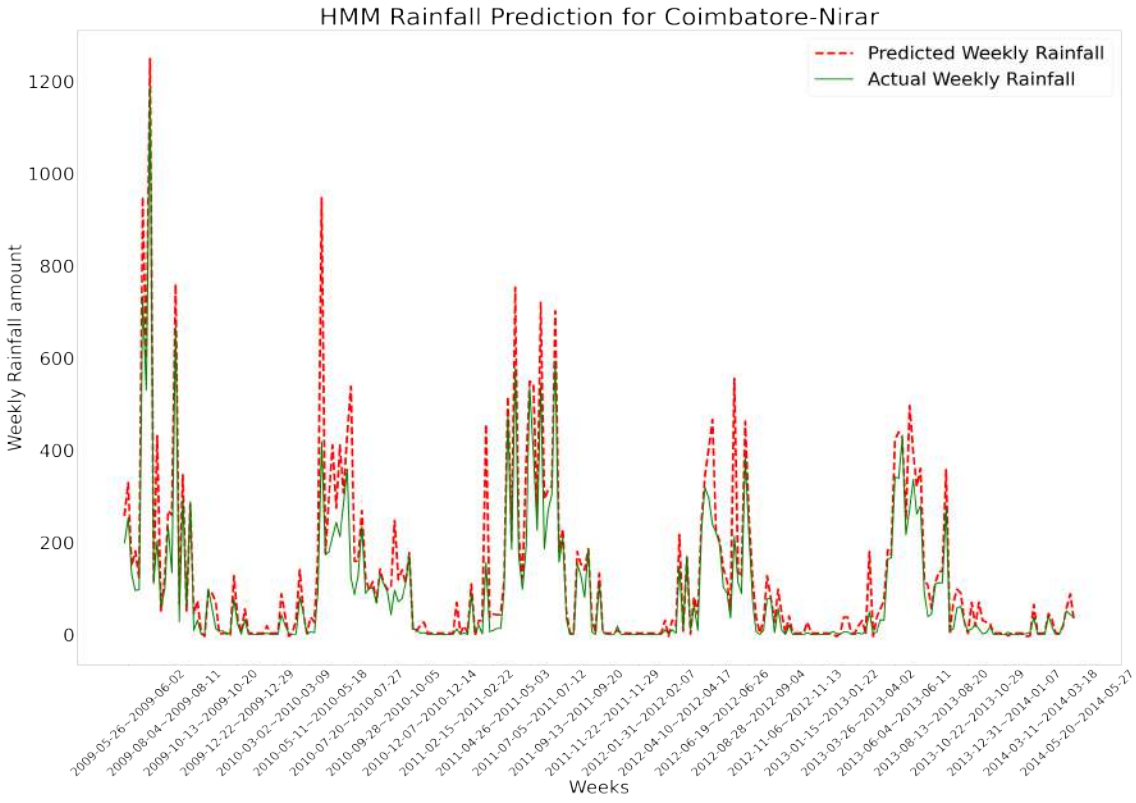
(a)



(b)

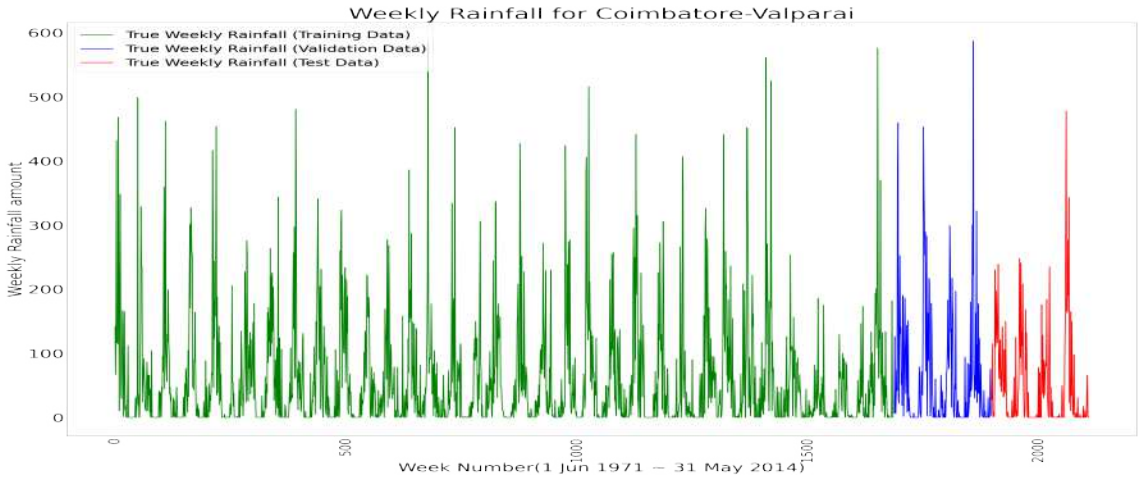


(c)

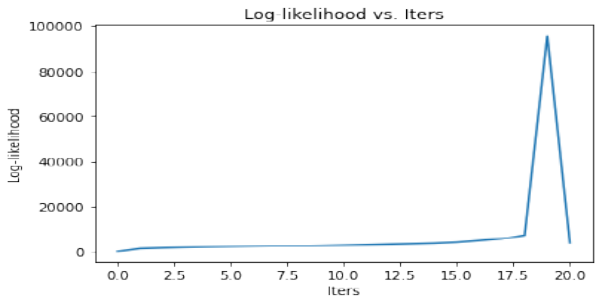


(d)

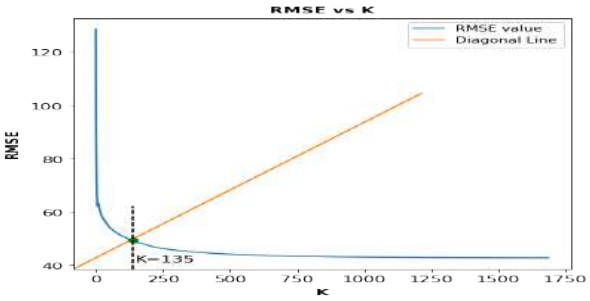
Fig. 6.33 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Nirar



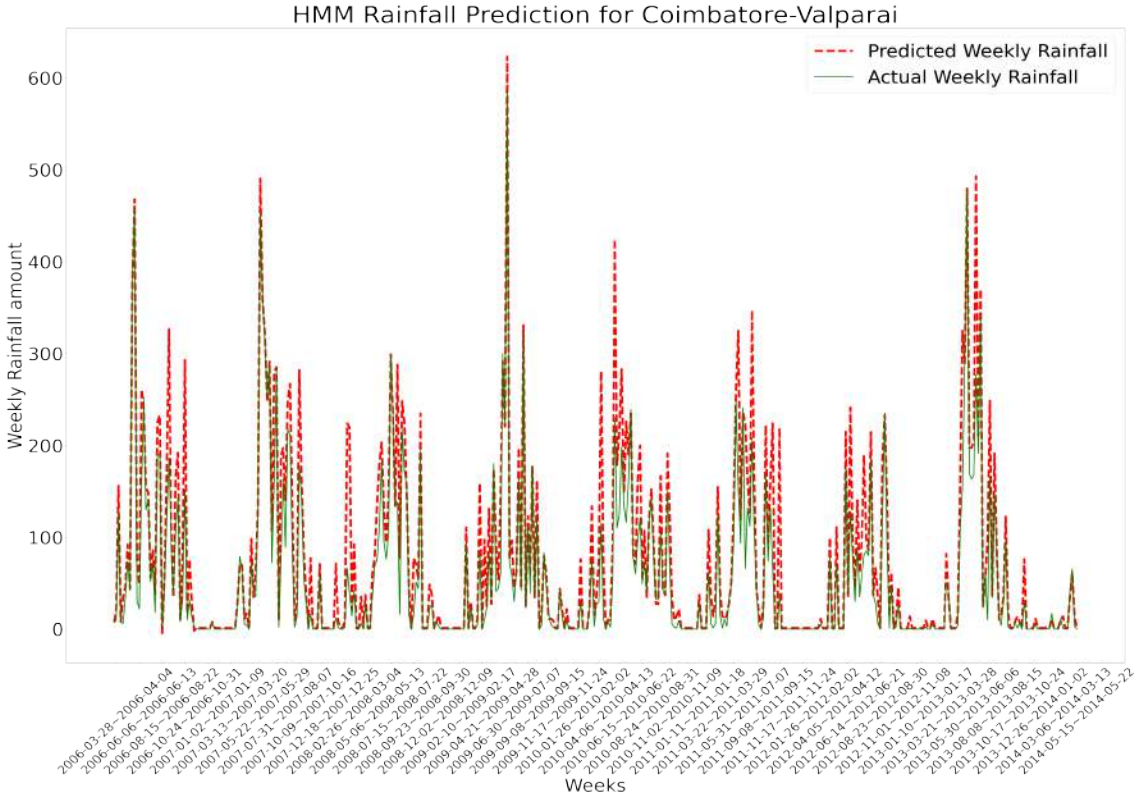
(a)



(b)

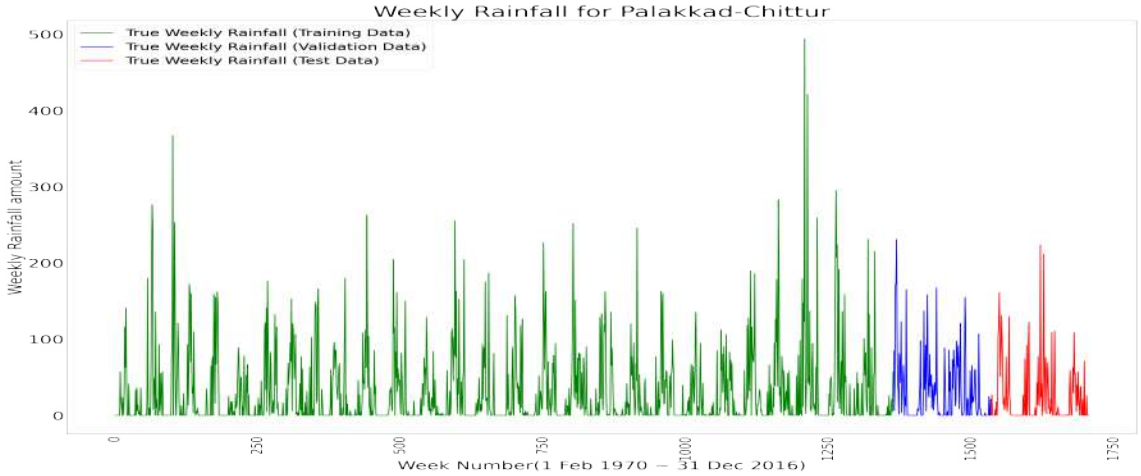


(c)

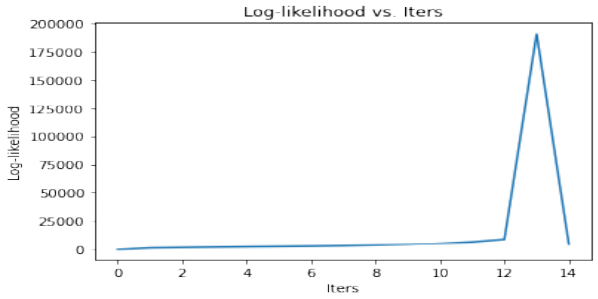


(d)

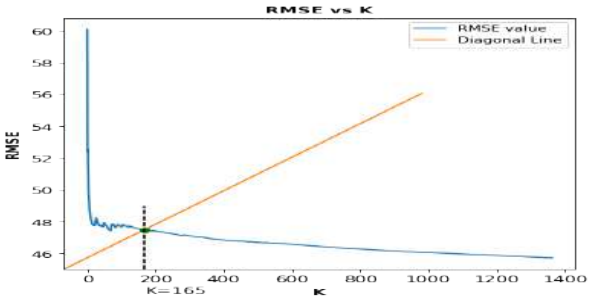
Fig. 6.34 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Valparai



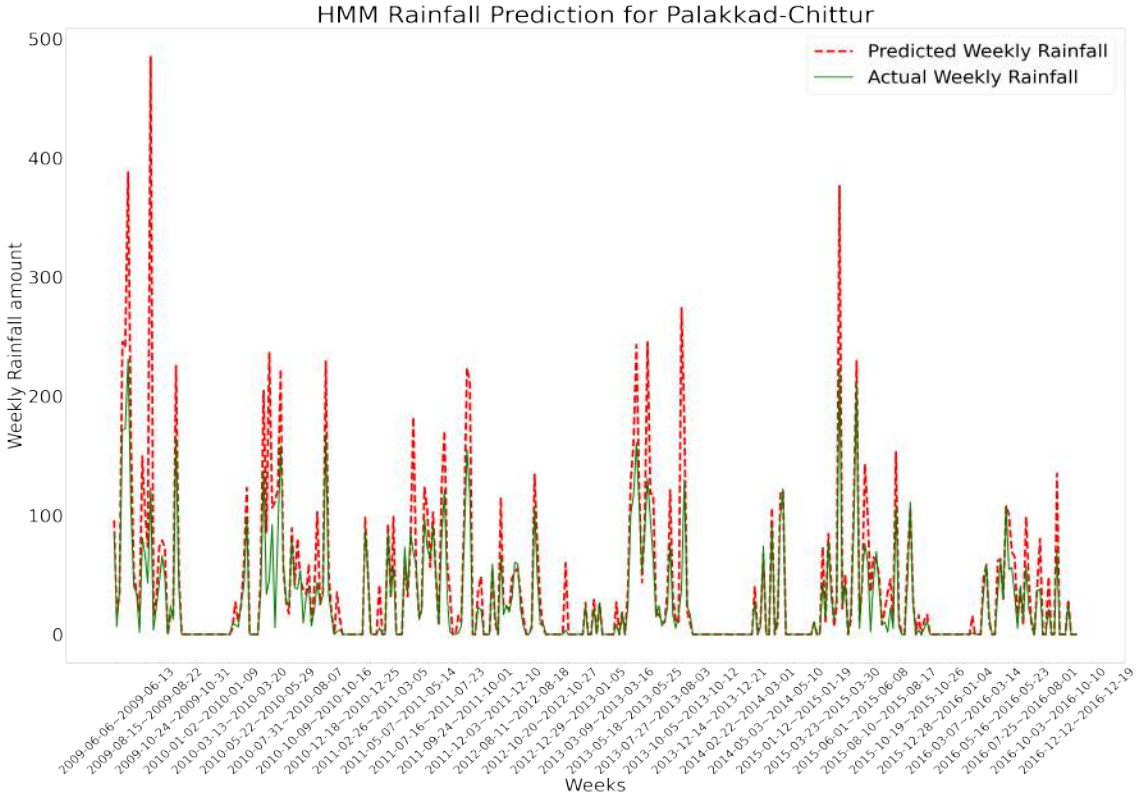
(a)



(b)



(c)



(d)

Fig. 6.35 (a) Total Weekly Rainfall (b) Training of HMM model (c) Choosing the Experience Factor K (d) Prediction of HMM Model for Chittur

Chapter 7

Conclusion and Future Work

In my report firstly we analyzed the data and tried to look into insights into it. We calculated a lot of statistical information about the data. Thereafter we discussed the ARIMA model where we applied the ARIMA model on our rainfall data. The model performed well as it had the option to hold the structure of information yet simultaneously it has failed to catch the limits. Moving further, we looked for literary reference and then we pre-processed the data as referenced in the research paper [1] and changed the rainfall data over to FIG value and further discovering the underlying trend in the data. I stifled the trend in the data and fed it to the ARIMA model after finding the ideal lag from the PACF plot. This time the ARIMA model has indicated great outcomes in catching the limits of data. Yet, stalled out in changing over back from FIG value to rainfall. I attempted to catch this by utilizing a neural network however failed to catch it. I tried to contact the author and sent a request to clarify the defuzzification process and he answered me back expressing that he will explain the things soon. So almost certainly, we will have the option to change the FIG over to rainfall after the author's explanations.

The HMM model which we discussed in Chapter 4 has shown its capability to handle the structure of rainfall data. This got verified when we did prediction on the test. But we saw some shortcomings when we tested it on 2018, 2019, and 2020 data. Then we modified

the prediction algorithm of the model and came up with a new prediction algorithm with another variable called experience factor (K). With the help of performance metrics with got a good K. Next we validated the performance of HMM model with the experience factor on the test and found better results than the original model. We also saw a great performance of the model on 2018, 2019, and 2020 weekly rainfall data. Thereafter we proceeded to find a good value of the experience factor and we found the solution with the properties of a hyperbola. After finding a good K we assessed the model with a huge variety of data to neglect the chance that the model might be a data-driven model. As expected, the model has shown great results on all the station's rainfall data with good performance metrics. Hence, the HMM model is a great model which we got through this project and is very well in predicting weekly rainfall.

References

[1] Pritpal Singh, *Indian summer monsoon rainfall (ISMR) forecasting using time series data: A fuzzy-entropy-neuro based expert system*, 2017, https://www.sciencedirect.com/science/article/pii/S1674987117301408.

[2] *[Online]*, https://www.kaggle.com/sumi25/understand-arima-and-tune-p-d-q.

[3] Lawrence R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, 1988, https://tinyurl.com/s87xvse2.

[4] B. Md. Rafiul Hassan, *Stock Market Forecasting Using Hidden Markov Model:A New Approach*, 2005, http://mleg.cse.sc.edu/edu/csce768/uploads/Main.ReadingList/HMM-stock.pdf.