GPC 208 RADIOMETRIC METHODS

ENHANCING SHORT-TERM RAIN FORECASTING ACCURACY USING RADIOMETRIC BRIGHTNESS TEMPERATURE DATA

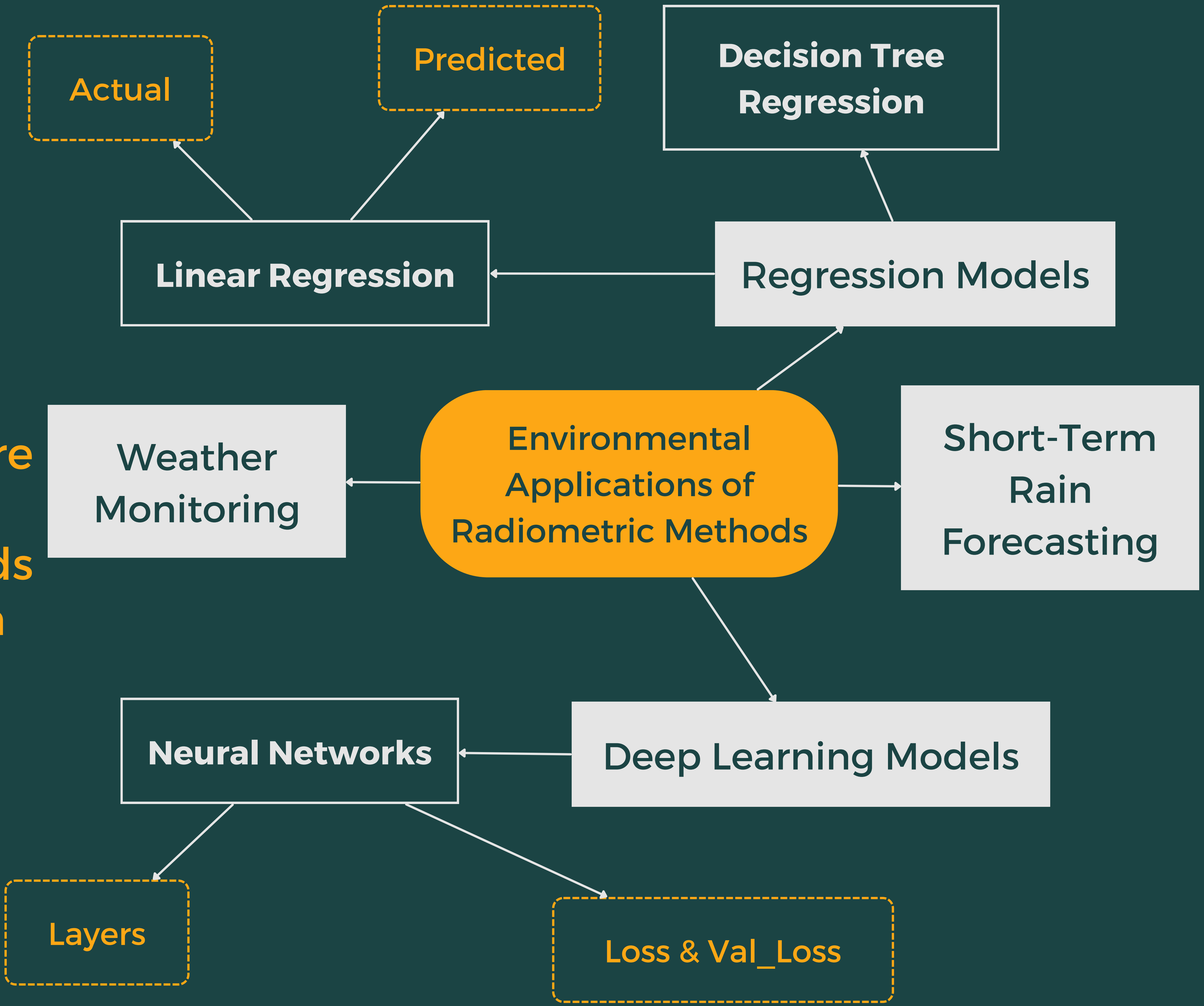
PRESENTED BY

Raj Sahu, Drishti Sen

CONTENTS

INTRODUCTION

- Important for analyzing atmospheric phenomena, short-term weather forecasting, and air pollution monitoring.
- Microwave radiometers provide continuous measurements.
- The radiometric brightness temperature outputs at 23.834 and 30 GHz are used to establish a relation where data trends which are precursors to rain events can be identified using this parameter.
- Linear regression, decision tree regression and neural networks can be used for estimation of rainfall with different levels of accuracy.



MOTIVATION & OBJECTIVE

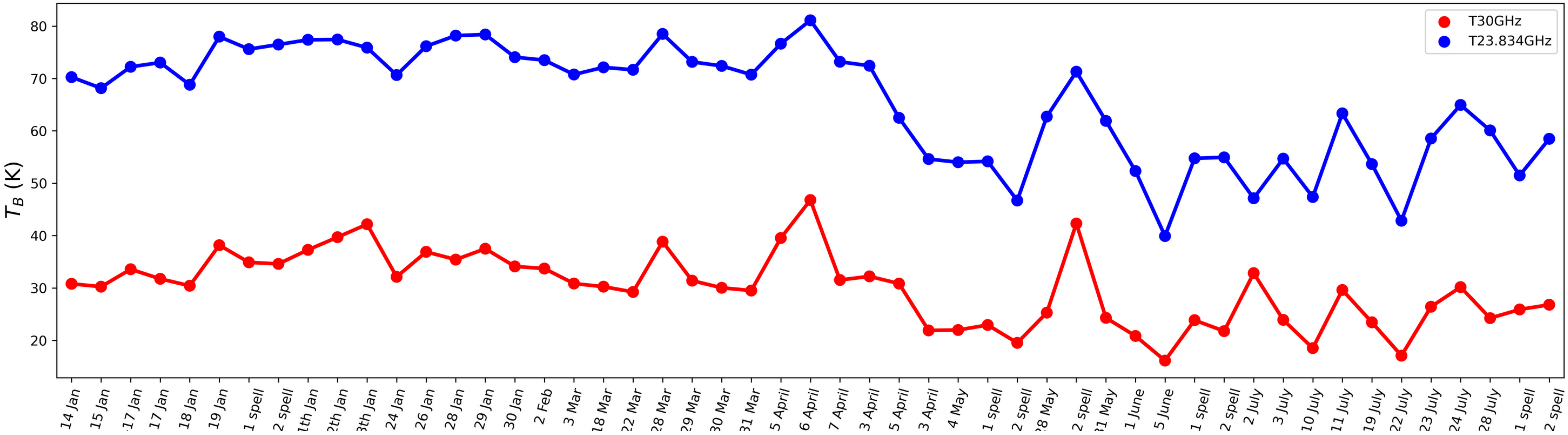
- The underlying drive behind this project is to demonstrate that radiometric data holds significant value not just for mineral exploration but also for environmental purposes.
- Additionally, this project places emphasis on the groundbreaking potential of machine learning and deep learning models in shaping our future.

- The objective of this project is to enhance short term rain prediction accuracy with the help of Radiometric Brightness Temperature data. We have used the following - Linear Regression, Neural Networks, and Decision Tree Regression to find out the most accurate model.

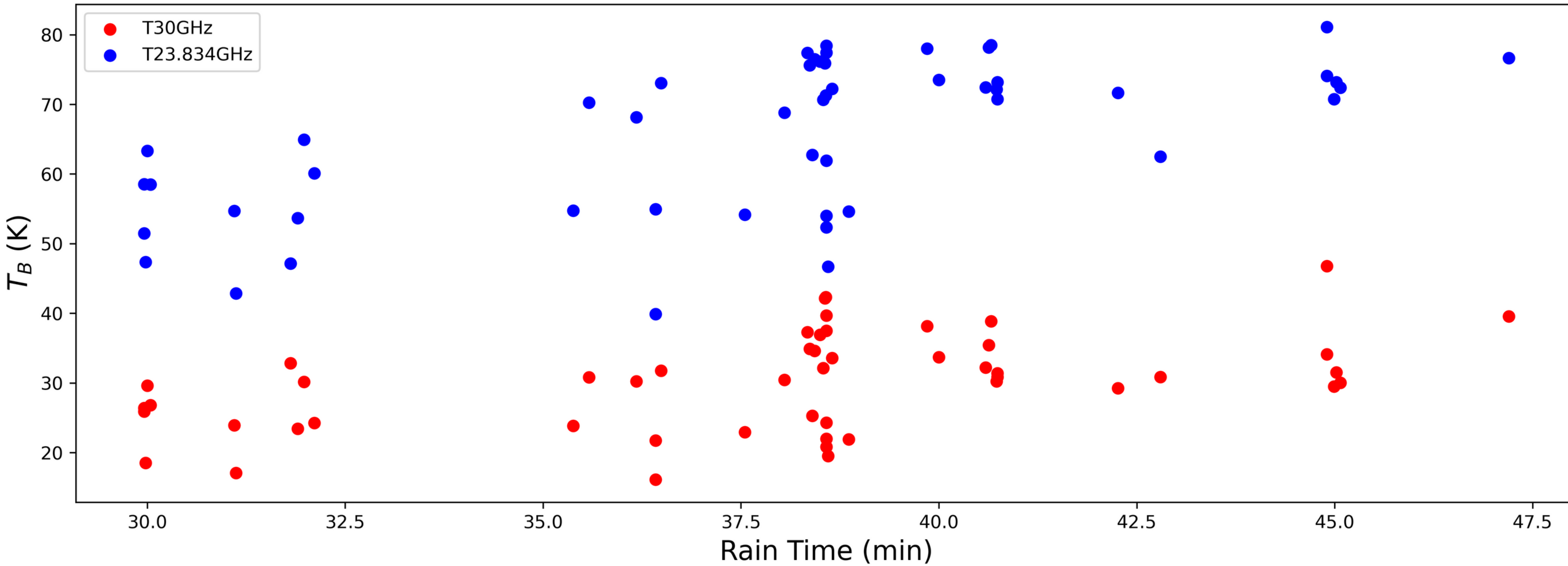
DATA

Date	T(K) at 23.834 GHz	T(K) at 30 GHz	Precursor time of rain event(min)
14 Jan	70.26	30.80	35.58
15 Jan	68.15	30.25	36.18
16 - 17 Jan	73.04	31.75	36.49
18 Jan	68.80	30.43	38.05

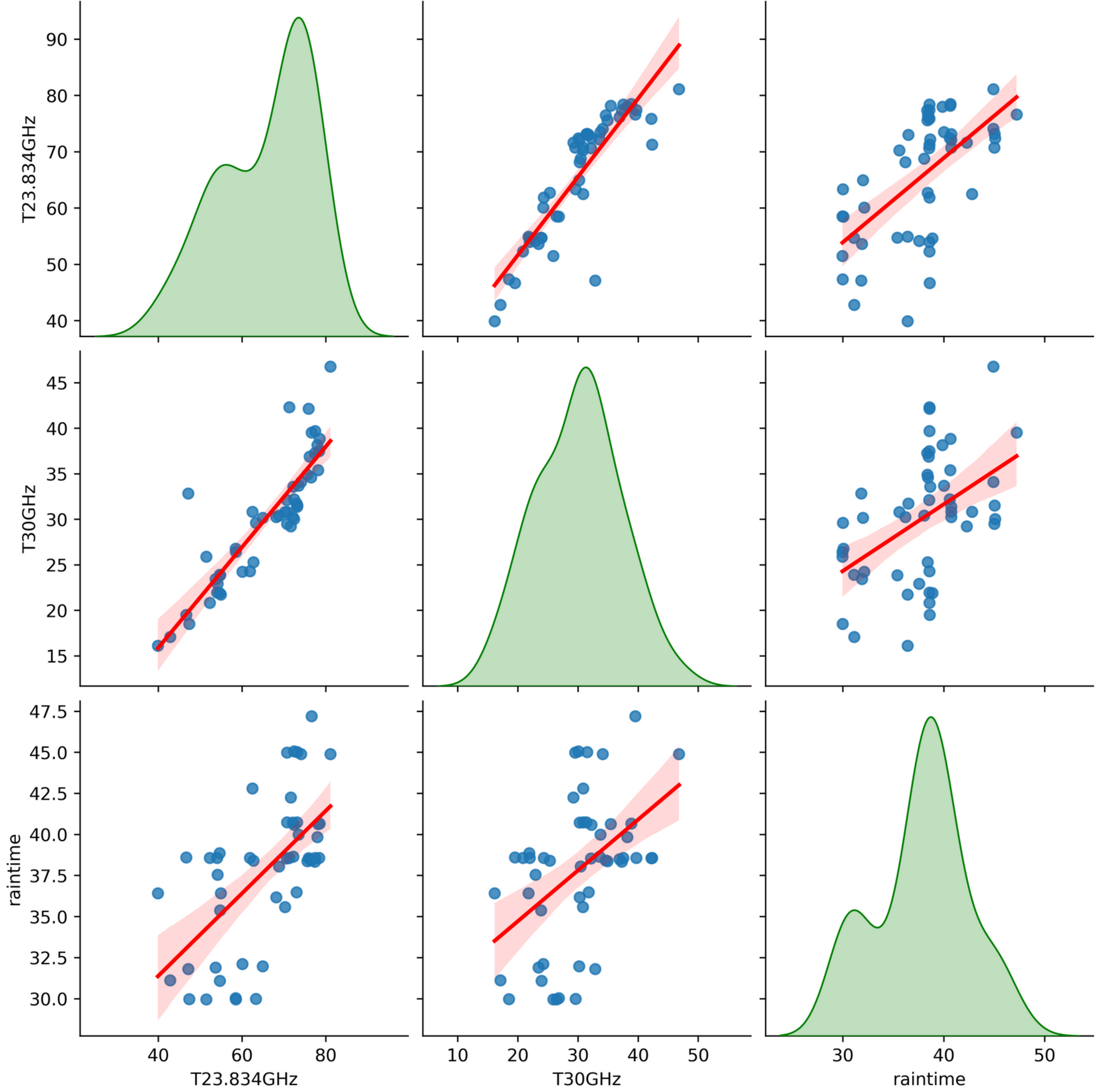
DATA ANALYSIS



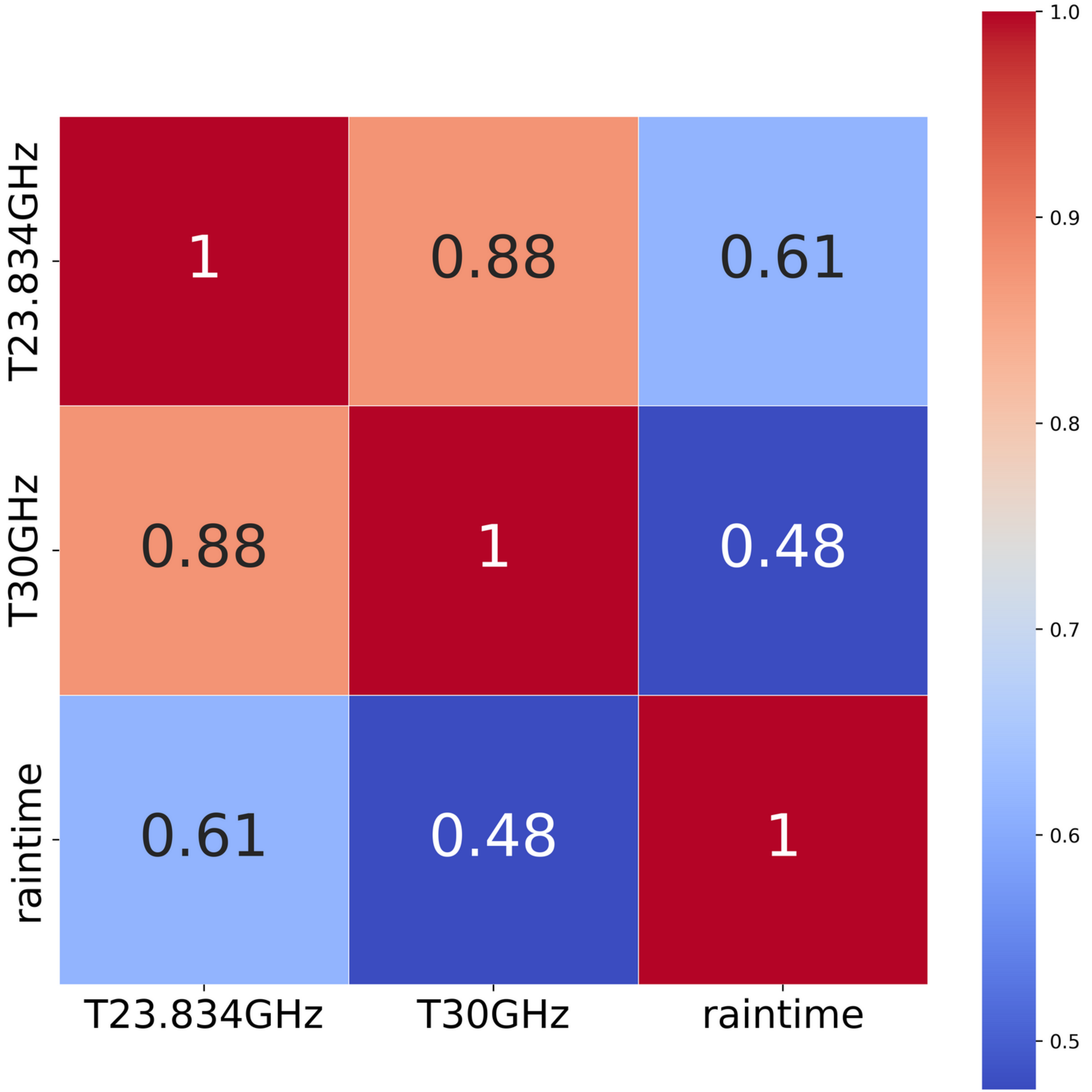
DATA ANALYSIS



Pair Plot



Heat Map



1
0.88
0.61
0.88
1
0.48
0.61
0.48
1
T23.834GHz
T30GHz
raintime
T23.834GHz
T30GHz
raintime
1.0
0.9
0.8
0.7
0.6
0.5

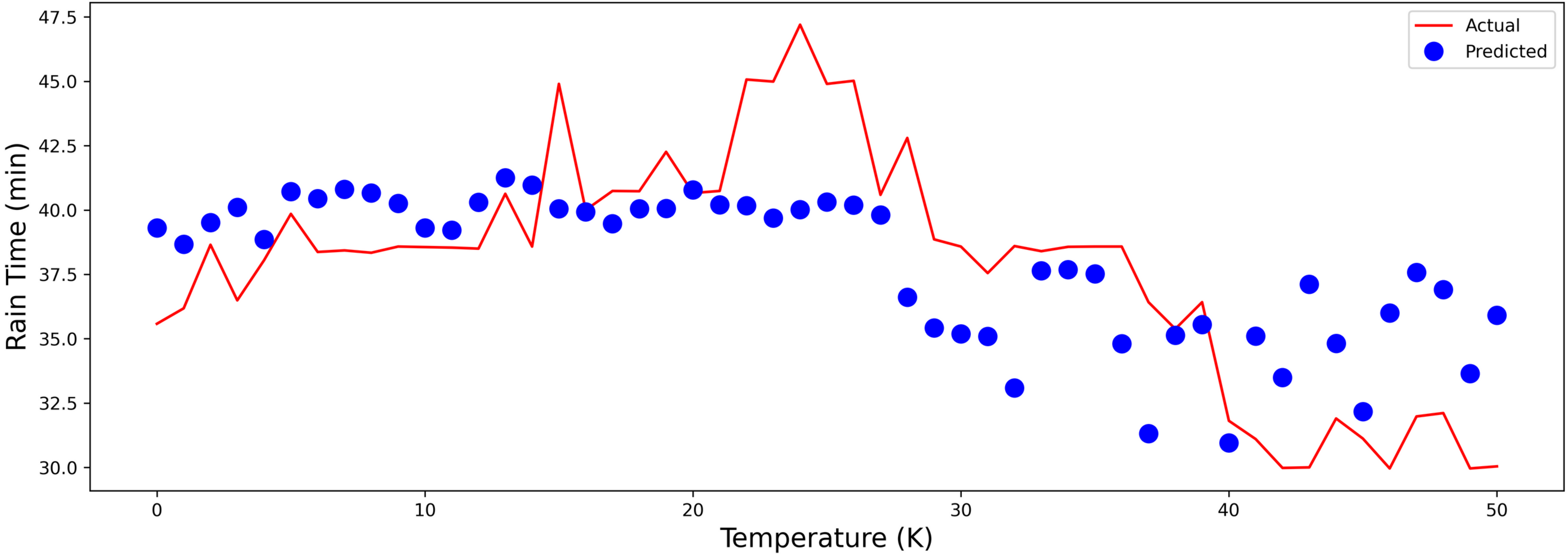
LINEAR REGRESSION


```
xTrain = data[['T23.834GHz','T30GHz']].values
yTrain = data['raintime']
```

```
from sklearn.linear_model import LinearRegression
```

```
linearModel = LinearRegression()
linearModel.fit(xTrain,yTrain)
linearPrediction = linearModel.predict(xTrain)
plt.figure(figsize=(15,5))
plt.plot(yTrain,color='red',label='Actual')
plt.plot(linearPrediction,'o',label='Predicted',color='blue',markersize=10)
plt.legend()
plt.savefig('plots/LinearRegression.png',dpi=450)
plt.show()
```

INTERPRETATIONS

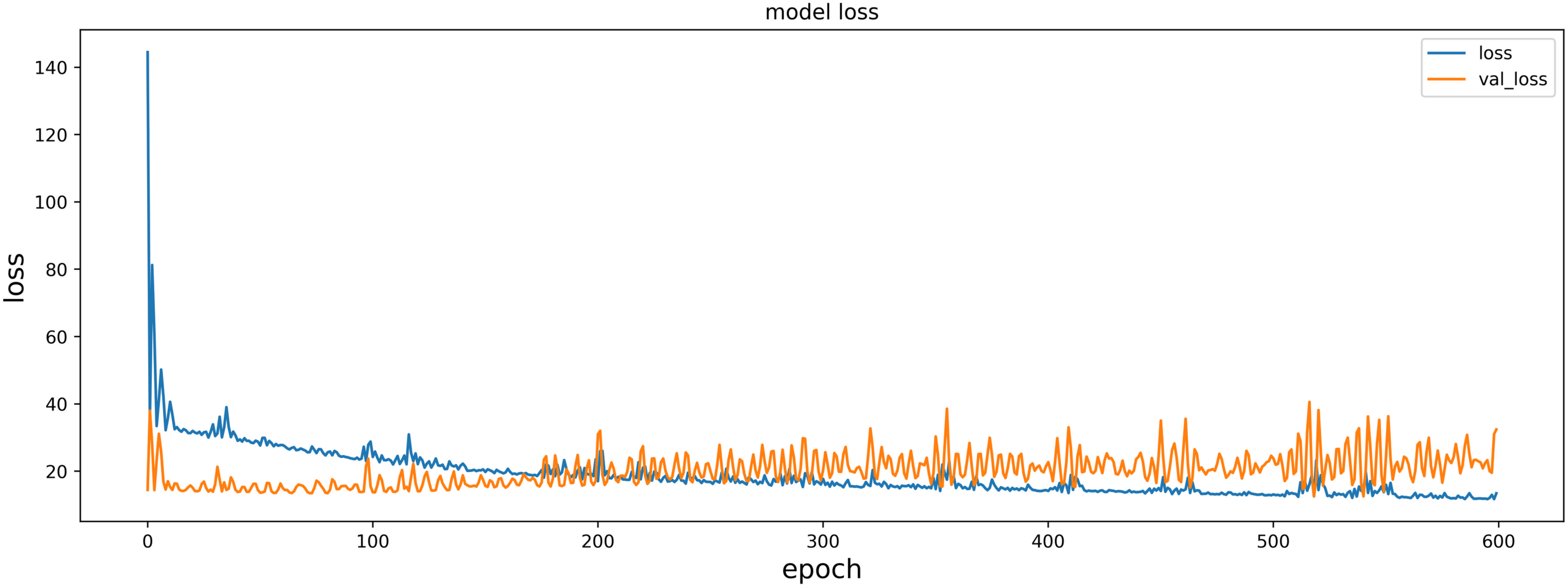


NEURAL NETWORKS

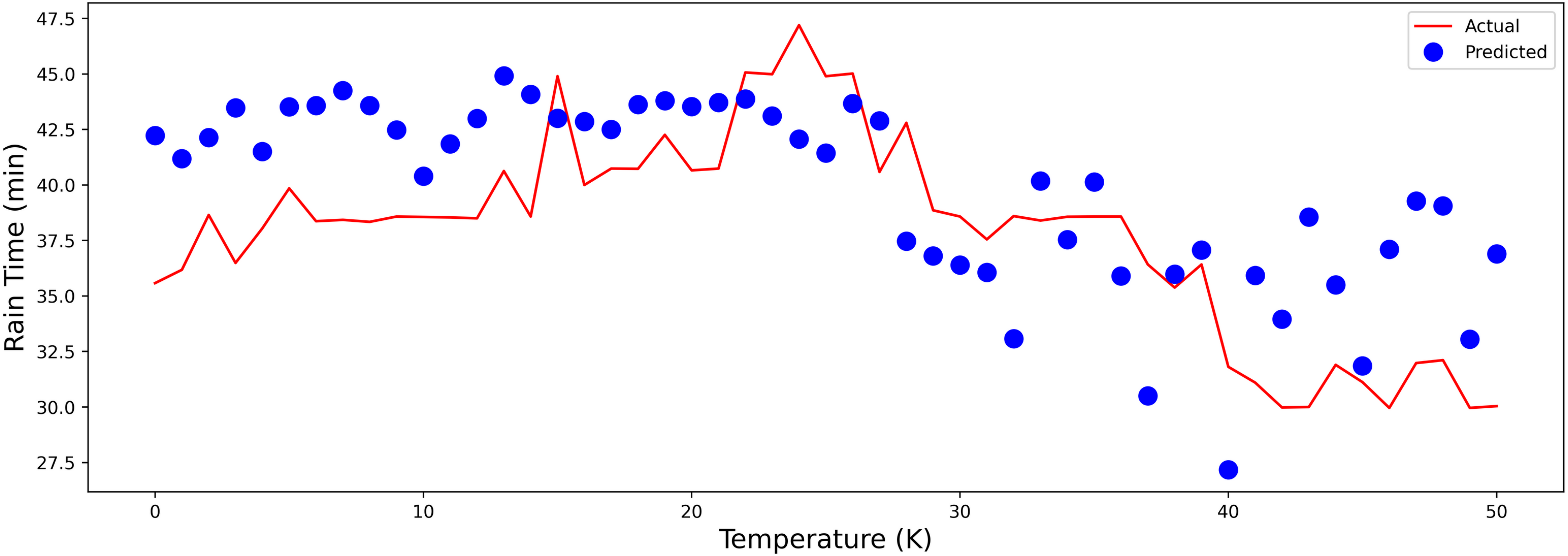

```python
from keras import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
```

```python
neuralModel = Sequential([
    Dense(2,activation='linear',input_shape=(2,)),
    Dense(1,activation='relu')
    ]
)
neuralModel.compile(
    optimizer=Adam(learning_rate=0.02),
    loss='mean_squared_error'
)
neuralHistory = neuralModel.fit(
    xTrain,
    yTrain,
    epochs=1000,
    batch_size=32,
    validation_split=0.2
)
```

INTERPRETATIONS



INTERPRETATIONS

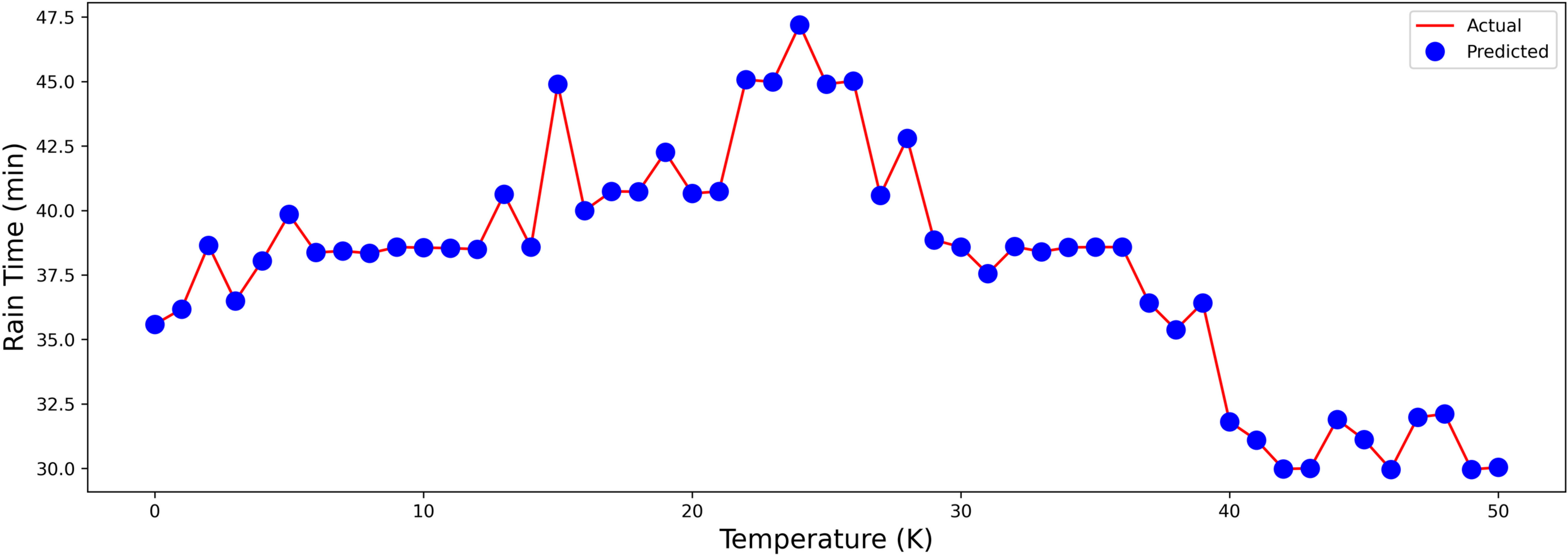


DECISION TREE REGRESSION

```
from sklearn.tree import DecisionTreeRegressor

decisionModel = DecisionTreeRegressor()
decisionModel.fit(xTrain,yTrain)
decisionPrediction = decisionModel.predict(xTrain)
plt.figure(figsize=(15,5))
plt.plot(yTrain,label='Actual',color='red',markersize=10)
plt.plot(decisionPrediction,'o',label='Predicted',color='blue
',markersize=10)
plt.legend()
plt.savefig('plots/DecisionRegression.png',dpi=450)
plt.show()
```

INTERPRETATIONS



DEPLOYMENT


```python
import modelbit
mb = modelbit.login()
```

You're connected to Modelbit as sahuraj457@gmail.com. Workspace: rajsahu. Branch: main

```python
mb.deploy(
    decisionModel,
    'DecisionTreeRegression',
    python_version='3.11',
    python_packages=["scikit-learn==1.2.2"]
)
```

Deploying DecisionTreeRegression

Uploading dependencies...

Uploading 'DecisionTreeRegression': 2.24kB [00:01, 1.81kB/s]

Success!

Deployment DecisionTreeRegression will be ready in a few seconds!

View in Modelbit

API REQUEST

```
import requests
url = 'https://rajsahu.app.modelbit.com/v1/DecisionTreeRegression/latest'
T23_834Ghz = 25
T30Ghz = 35
query = {"data": [T23_834Ghz,T30Ghz]}
r = requests.post(url, json=query)
value = r.json()['data']
print(f'{value} Rain Time (min)')
```

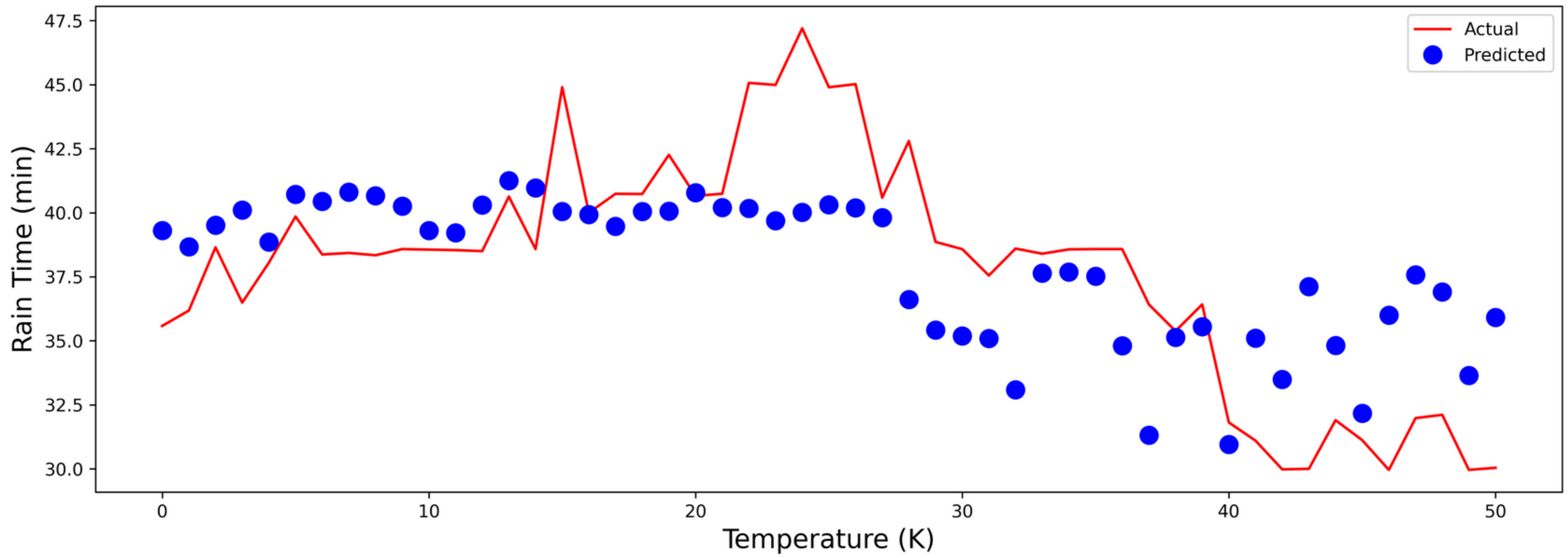
[13] 11.5s

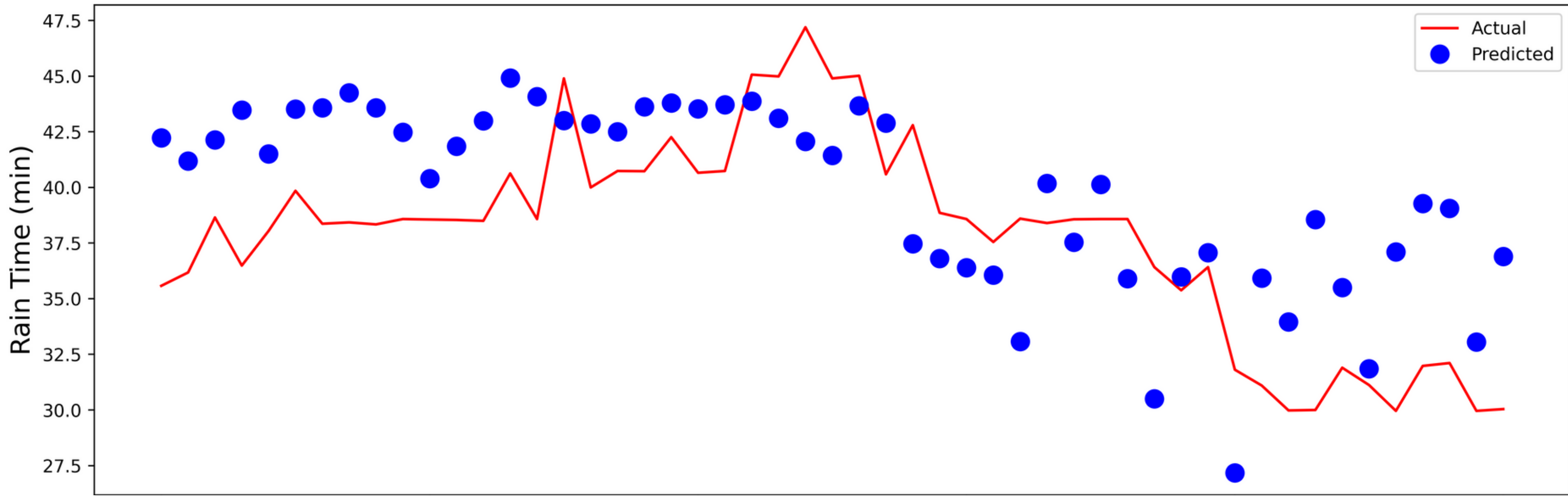
```
31.81 Rain Time (min)
```

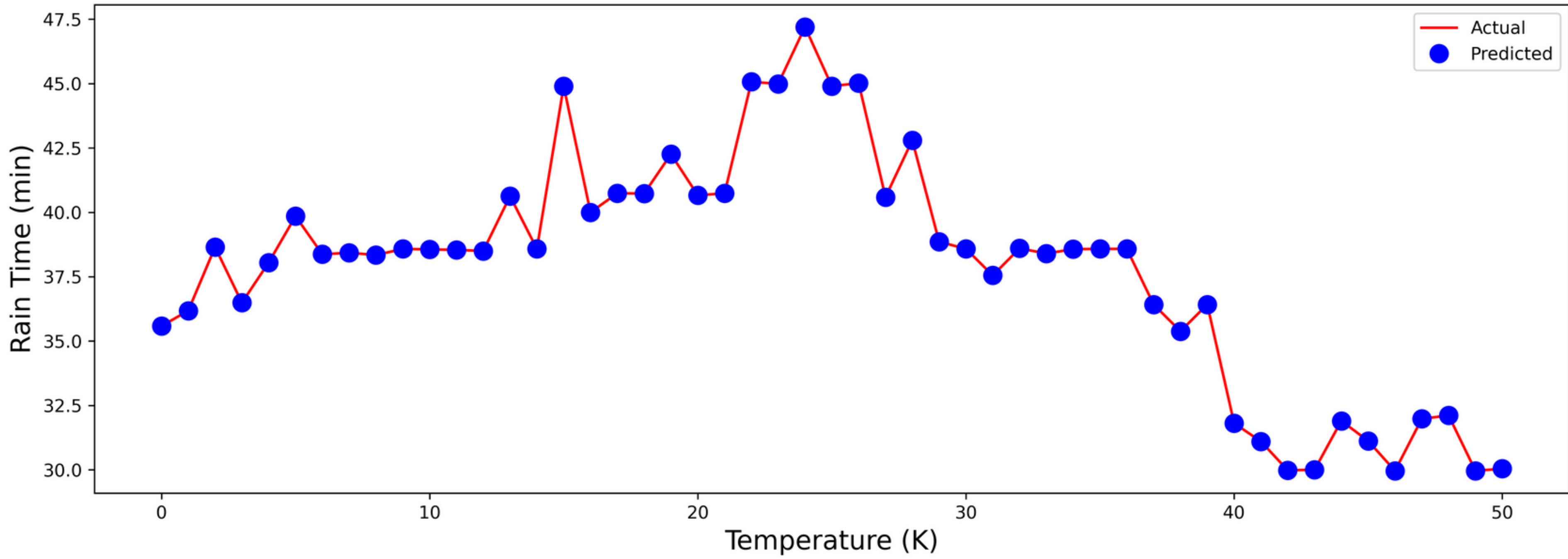
DISCUSSIONS

In our linear regression and neural network models, the predicted rain time is close to the actual rain time. However certain deviations are observed.

The decision tree regression model shows 100 per cent accuracy with the actual rain time. However, this has data overfit issues.







CONCLUSIONS

- Radiometric measurement is extremely significant for short-term weather forecasting due to its all-weather capability.

- The variations observed can be said to be indicators of rain initiation. It can be further analyzed for evaluating different climatic situations, especially in temperate climates to check the variability of the time interval before the initiation of precipitation.

- The accuracy of prediction achieved in these models can be further enhanced by applying other machine learning techniques.

REFERENCES

Kausik Bhattacharya et al. [2020], Short Term Rain Forecasting From Radiometric Brightness Temperature Data, Journal of Mechanics of Continua And Mathematical Sciences, Vol-15, No.-2,pp 70-83

https://github.com/rajsahu2004/Radiometric-Method-Project