EGYPTIAN SPACE AGENCY

EGYPTIAN UNIVERSITIES TRAINING SATELLITE PROJECT

EUTS

ANOMALY DETECTION FOR SATELLITE TELEMETRY DATA USING MACHINE LEARNING

Prepared by

Helwan University

Computer and Artificial intelligence–Computer Science T42

Introduction

- **Project Idea**: detect of telemetry data that come from satellites with using of machine learning algorithms,
- **Telemetry data** is set of measurement and reading of embedded device Across time interval
- **In this project** ,we use two dataset consist of Date and Temperature records ,after preprocessing dataset we use time series and machine learning algorithm to detect future temp

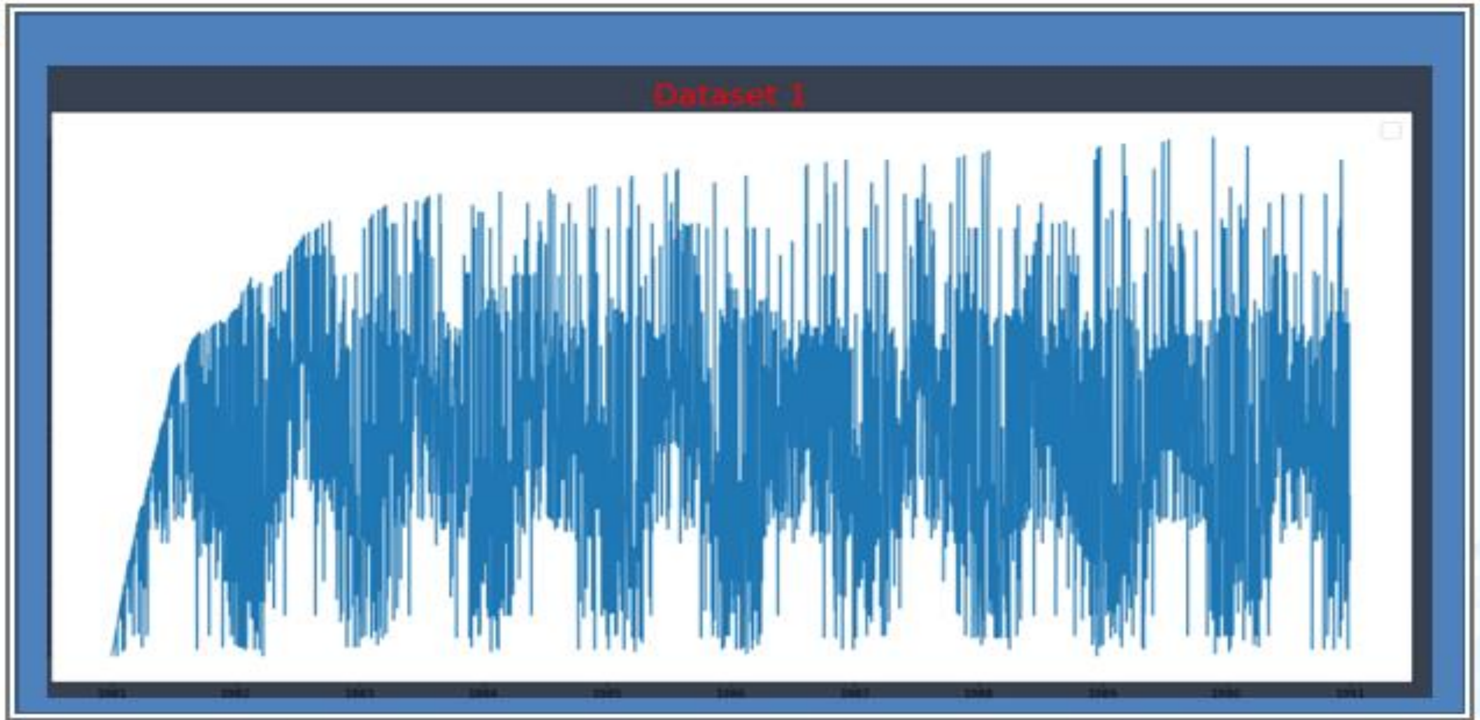
Egyptian Universities Training Satellite

Data Analysis

EUTS
Egyptian Universities Training Satellite

Dataset 1

- dataset 1 consists of one column "Date","Temp" and we try to separate them into two columns ‘Date’ , ‘temp’



	"Date","Temp"
0	"1981-01-01",20.7
1	"1981-01-02",17.9
2	"1981-01-03",18.8
3	"1981-01-04",14.6
4	"1981-01-05",15.8

Dataset 1 Preprocessing

- Dataset 1 consists of one column "Date","Temp" and we try to separate them into two columns ‘Date’ , ‘temp’
- Convert temp column data type from string to float
- Convert Date column data type from string to Datetime
- Drop Null values and check for Duplicated
- Finally data after Cleaned

```
"1986-02-04"" at position 1859
```

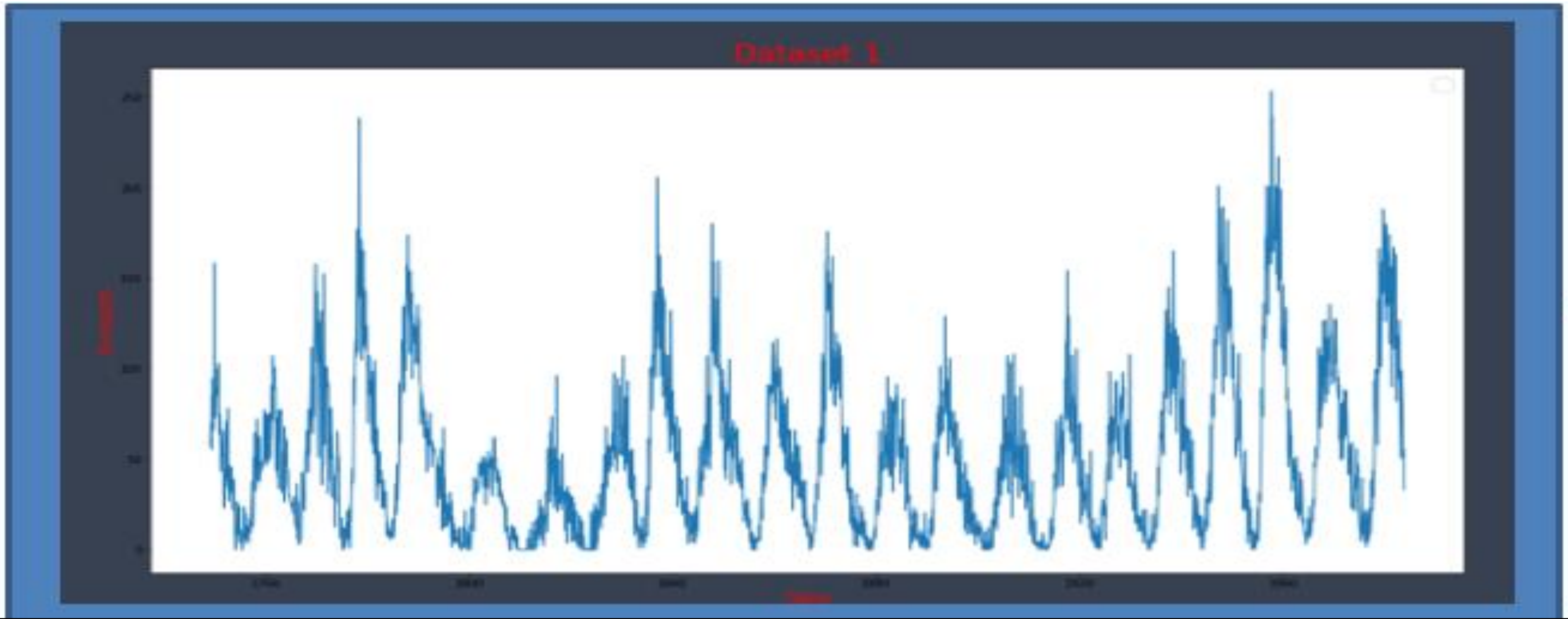
	Date	Temp
0	1981-01-01	20.7
1	1981-01-02	17.9
2	1981-01-03	18.8
3	1981-01-04	14.6
4	1981-01-05	15.8

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3647 entries, 0 to 3649
Data columns (total 2 columns):
Date    3647 non-null datetime64[ns]
Temp    3647 non-null float64
dtypes: datetime64[ns](1), float64(1)
memory usage: 85.5 KB
```

Egyptian Universities Training Satellite

Dataset 2

dataset 2 it consists of one column "Month","Sunspots" "Date","Temp" we try to seprate this column into two column Date and Sunspots



	"Month","Sunspots"
0	"1749-01",58.0
1	"1749-02",62.6
2	"1749-03",70.0
3	"1749-04",55.7
4	"1749-05",85.0

وكالة الفضاء المصريه
Egyptian Space Agency

Dataset 2 Preprocessing

- dataset 2 it consists of one column "Month","Sunspots" "Date","Temp" we try to separate this column into two column Date and Sunspots
- convert data type for Sunspots from String to float
- Convert Date column data type from string to Datetime
  - we found two issue at index 7 where year and month are swapped ,at index 489 invalid month
- Drop Null values and check for Duplicated
- Finally dataset look like

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2815 entries, 0 to 2818
Data columns (total 2 columns):
Date         2815 non-null datetime64[ns]
Sunspots     2815 non-null float64
dtypes: datetime64[ns](1), float64(1)
memory usage: 66.0 KB
```

	Date	Sunspots
0	1749-01-01	58.0
1	1749-02-01	62.6
2	1749-03-01	70.0
3	1749-04-01	55.7
4	1749-05-01	85.0

EUTS

Egyptian Universities Training Satellite

ARIMA Model

INTRODUCTION TO ARIMA

- ARIMA is an that stands for Auto Regressive Integrated Moving Average.
- ARIMA is an acronym that stands for Auto Regressive Integrated Moving Average. It is a generalization of the simpler Auto Regressive Moving Average and adds the notion of integration
- An ARIMA model is a class of statistical models for analyzing and forecasting time series data.
- Any 'non-seasonal' time series that exhibits patterns and is not a random white noise can be modeled with ARIMA models.
- The statsmodels library provides the capability to fit an ARIMA model.
- An ARIMA model can be created using the statsmodels library as follows:
    - Define the model by calling ARIMA() and passing in the p, d, and q parameters:
        - p is the order of the AR term(It refers to the number of lags of Y to be used as predictors.)
        - q is the order of the MA term(It refers to the number of lagged forecast errors that should go into the ARIMA Model.)
        - d is the number of differencing required to make the time series stationary

- The model is prepared on the training data by calling the fit() function.
- Predictions can be made by calling the predict() function and specifying the index of the time or times to be predicted

implemented in project

1-train data

2- We use Auto ARIMA and Get
Best model(Dataset1): ARIMA(3,0,1)
Best model(Dataset2): ARIMA(3,0,2)

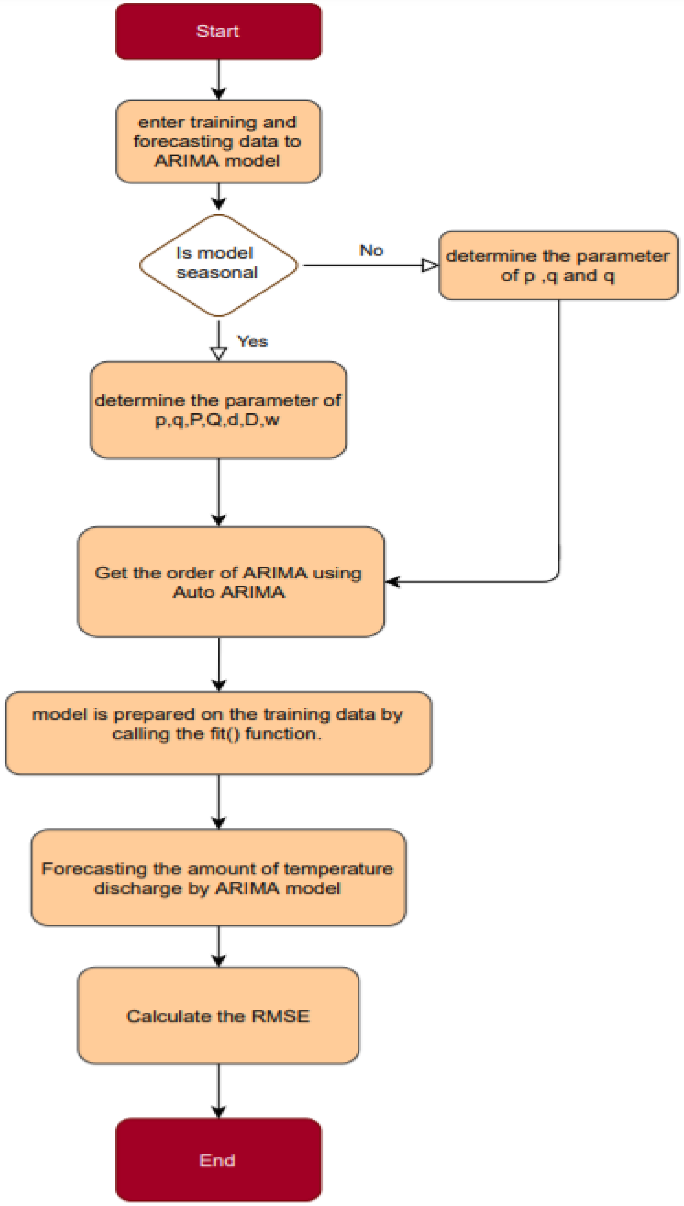
3- Create ARIMA Model

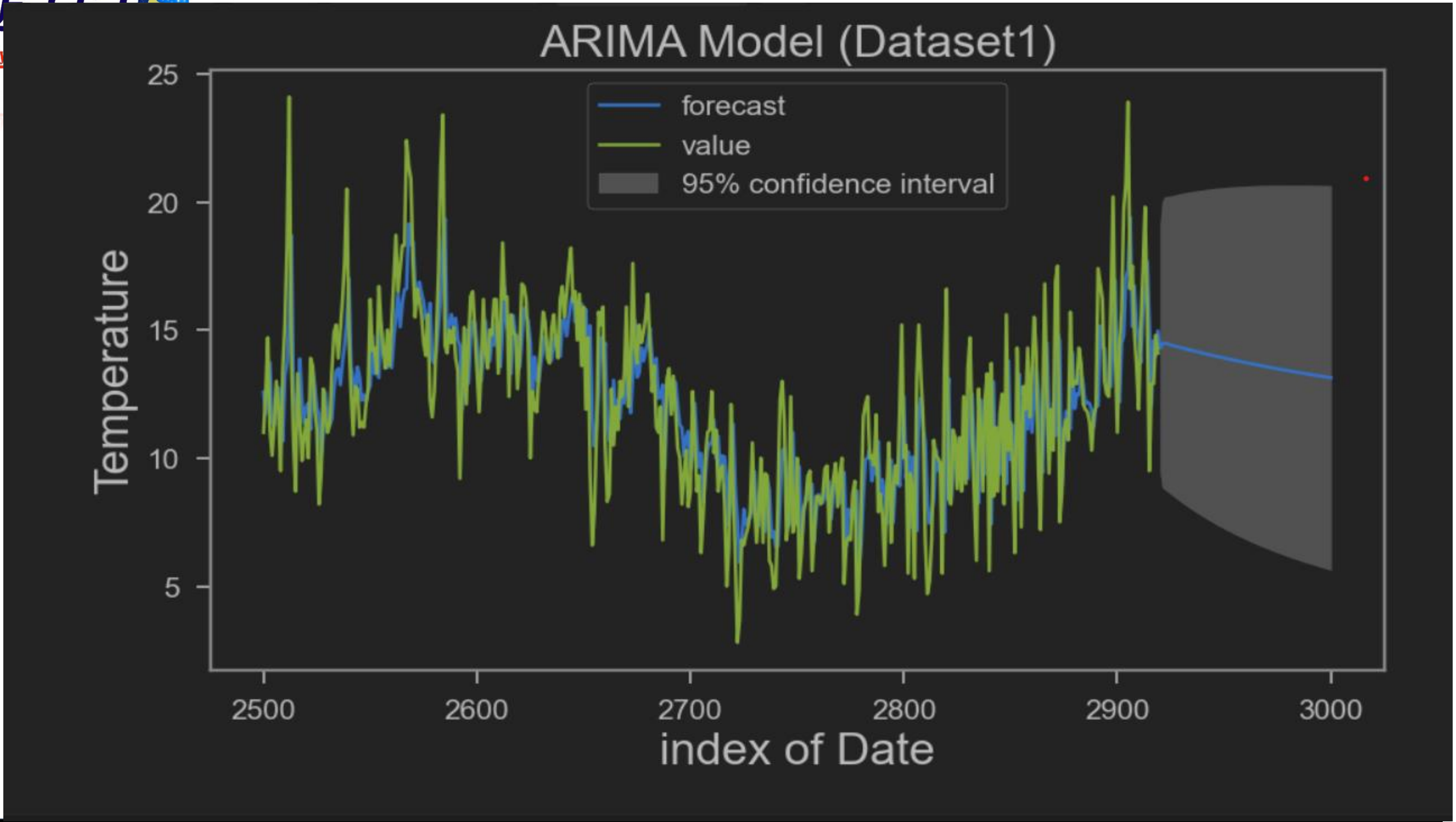
4-Fit the model

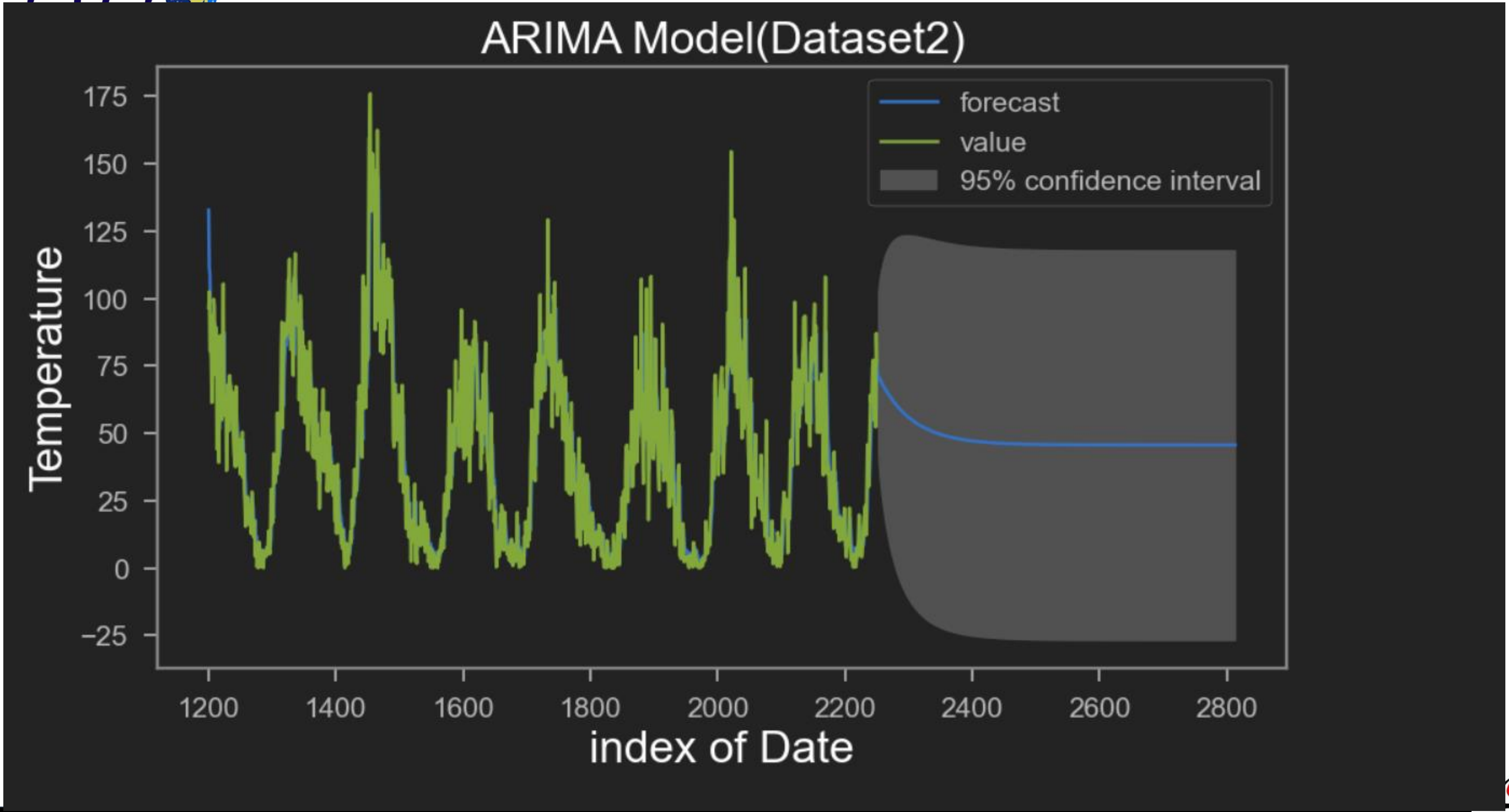
5-predict

6-calculate the mean square error

Algorithm Name	Mean Square Error on Dataset 1	Mean Square Error on Dataset 2
RMSE	4.019	62.041



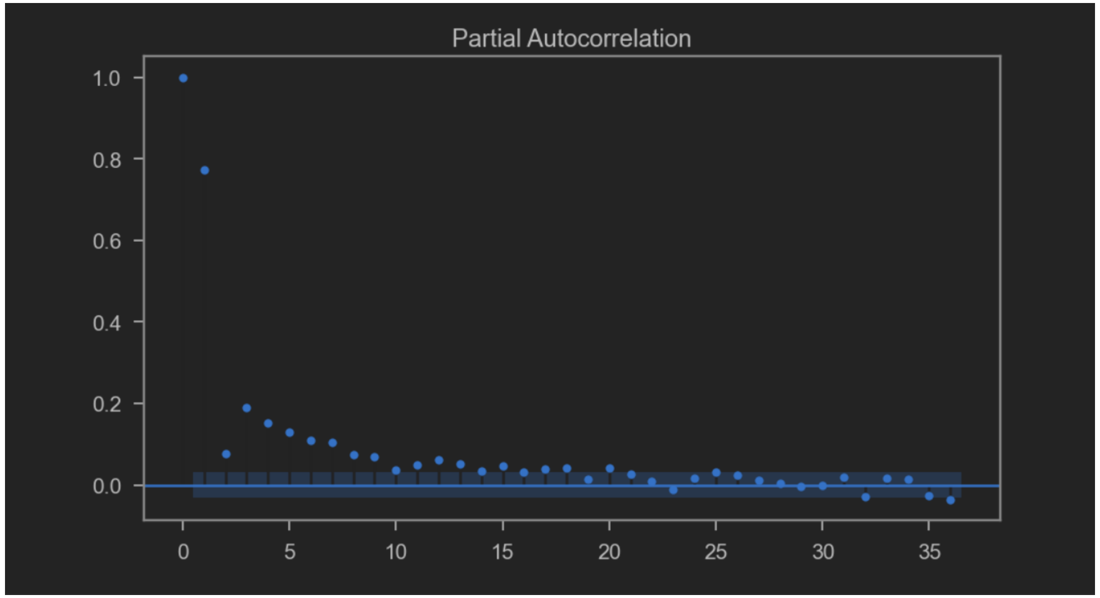

ARIMA Model (Dataset1)
forecast
value
95% confidence interval
Temperature
25
20
15
10
5
2500
2600
2700
2800
2900
3000
index of Date

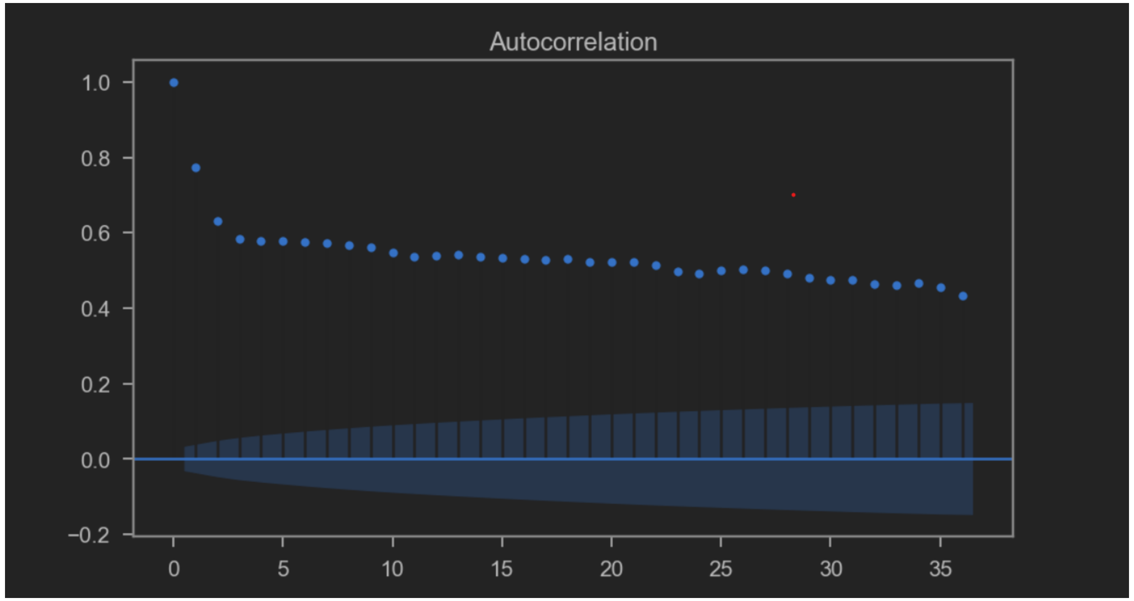

ARIMA Model(Dataset2)
forecast
value
95% confidence interval
Temperature
175
150
125
100
75
50
25
0
−25
1200
1400
1600
1800
2000
2200
2400
2600
2800
index of Date

ARMA Model

INTRODUCTION TO ARMA

- ARMA stands for Auto Regressive Moving Average.
- An ARMA model is a class of statistical models used to describe weakly stationary time series.
- We use ACF & PACF plots to check parameters and see if there’s a need to use differencing.

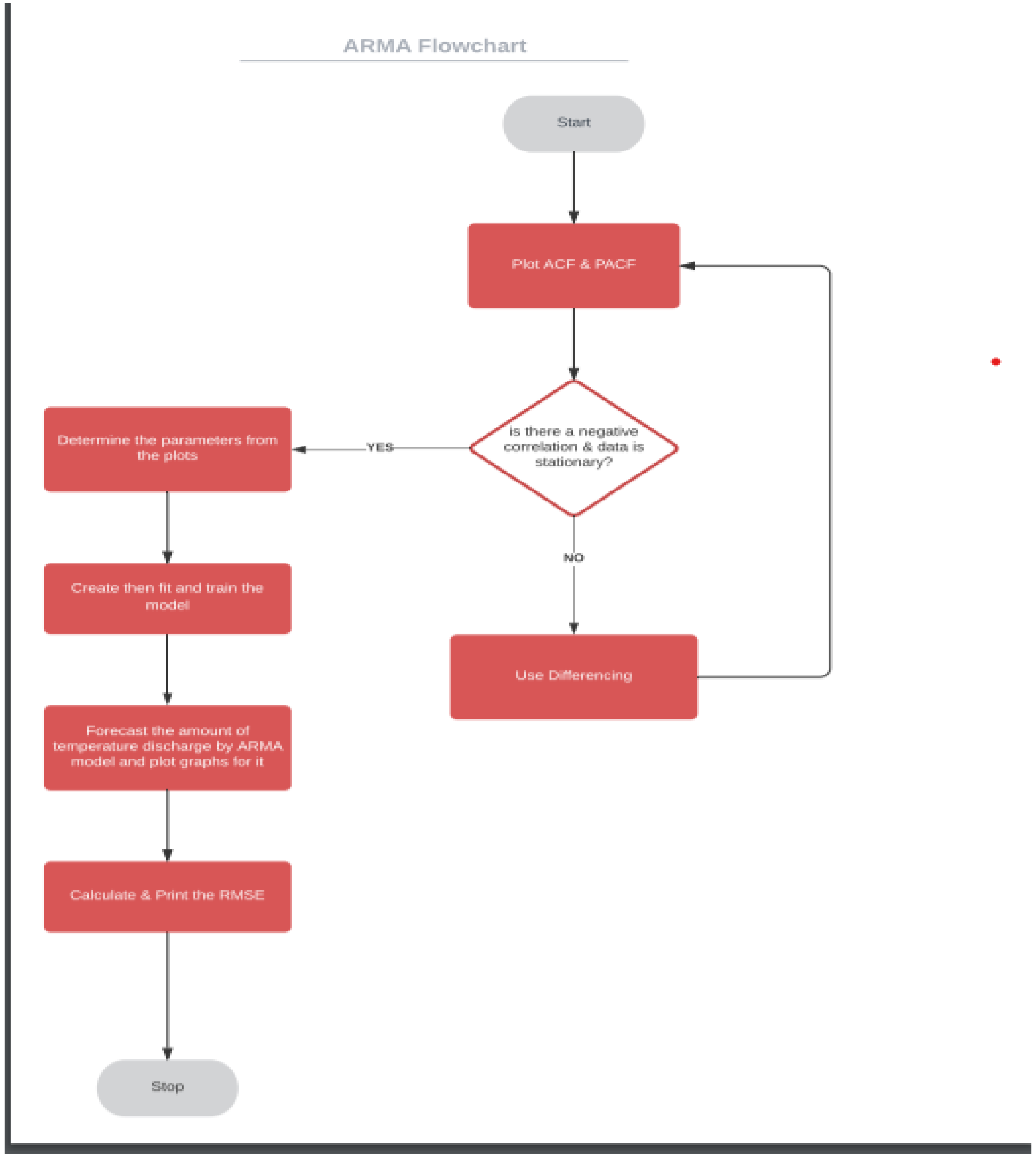




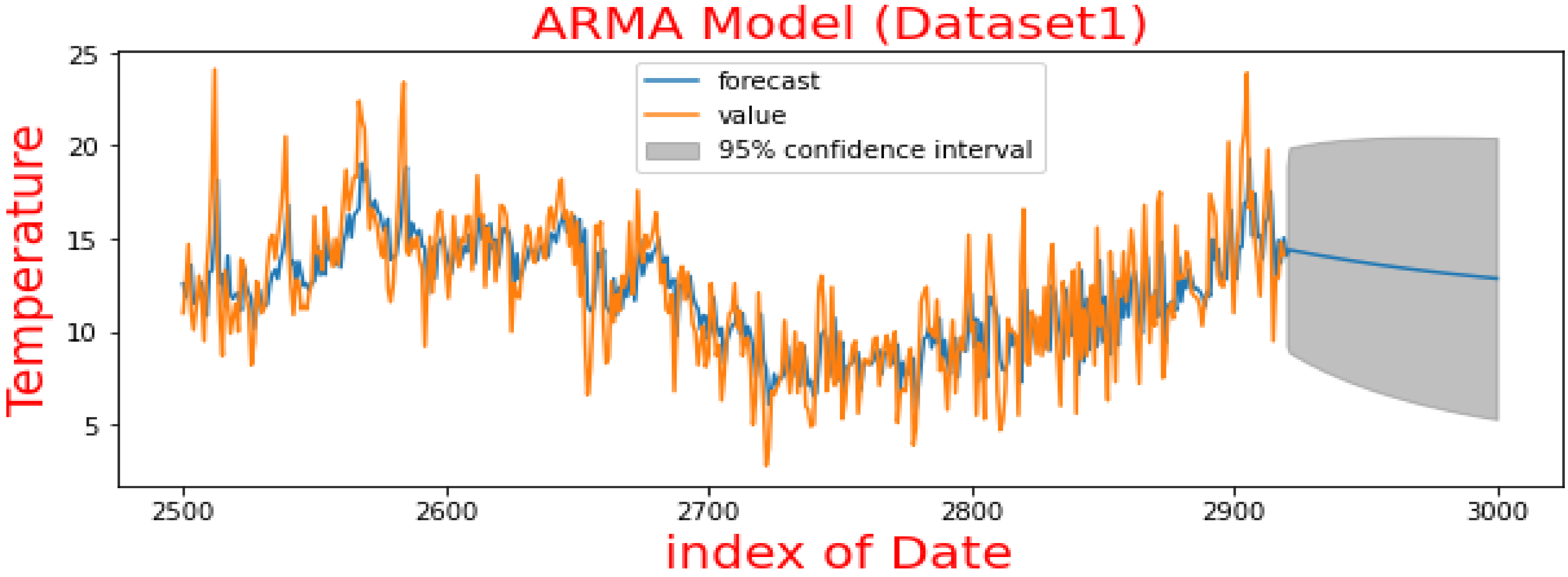
INTRODUCTION TO ARMA

- An ARMA model can be created using the statsmodels library as follows:
  - Define the model by calling ARMA() and passing in the p, d, and q parameters.
  - The model is prepared on the training data by calling the fit() function.
  - Predictions can be made by calling the predict() function and specifying the index of the time or times to be predicted
- The parameters of the ARMA model are defined as follows:
  - p: The number of lag observations included in the model, also called the autoregressive 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.

Flow chart

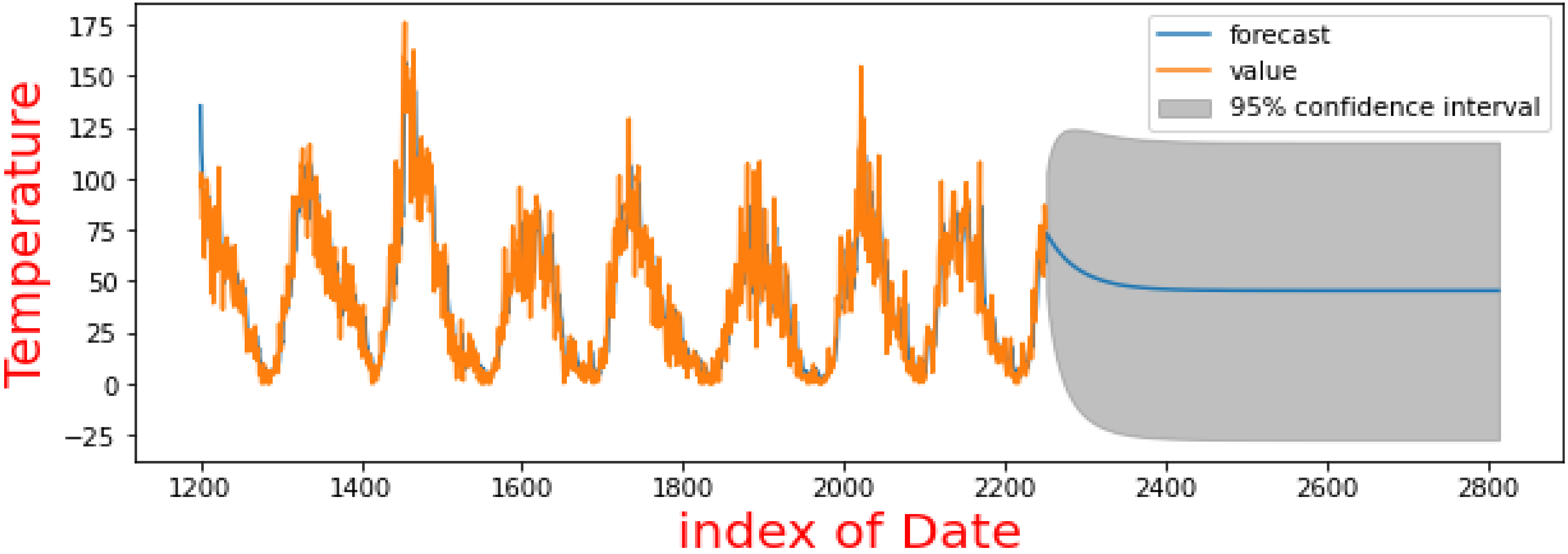


EgSA
وكالة الفضاء المصرية
Egyptian Space Agency



Result “Mean Square Error:3.987

ARMA Model(Dataset2)



Result “Mean Square Error:62.247

EUTS
Egyptian Universities Training Satellite

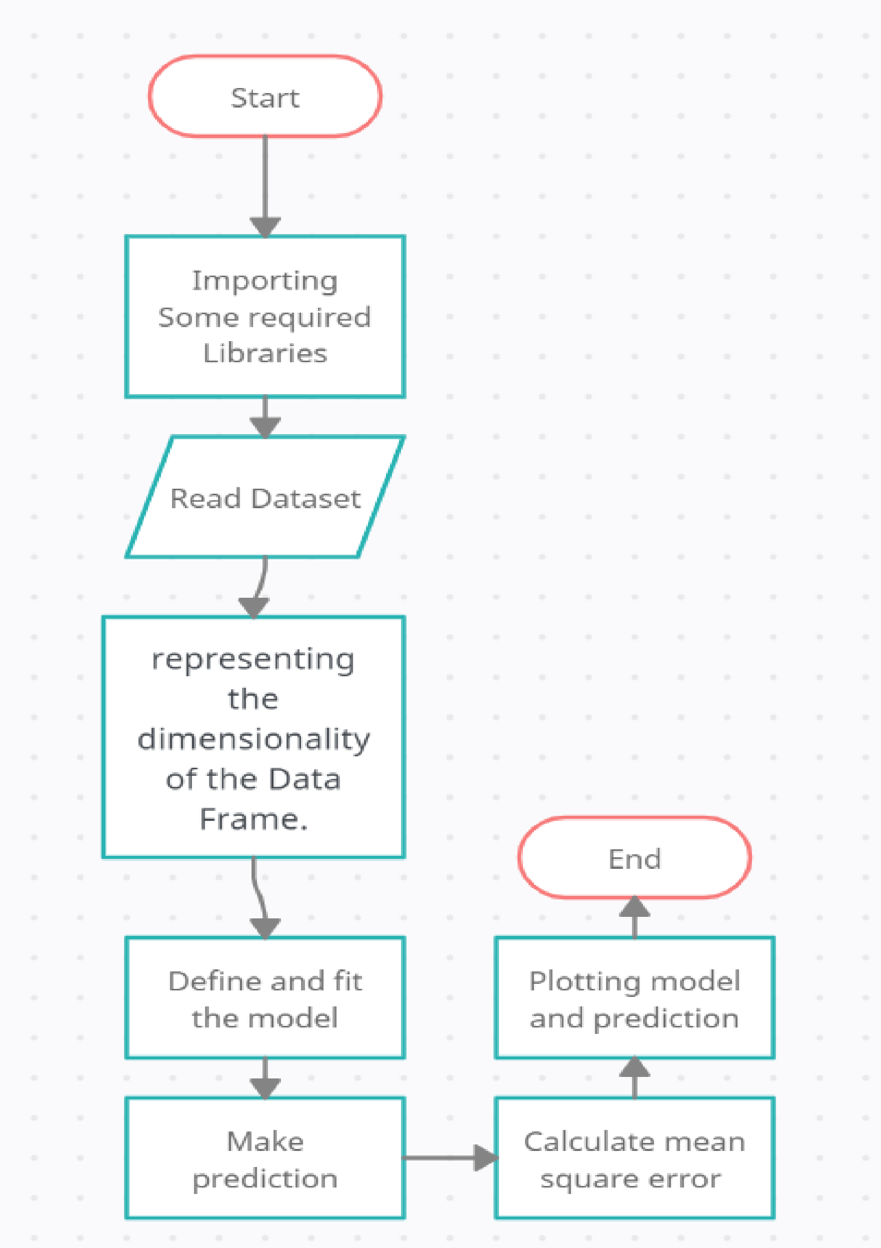
SES Model

INTRODUCTION TO SES

- Exponential Smoothing is an elementary and pragmatic technique used for forecasting where the forecast is made through the exponentially weighted average of prior observations.
- It analyzes data from a specific period of time via providing more importance to recent data and less importance to former data
- This method produces “smoothed data”, the data that has a noise removed, and allows trends and patterns to be more clearly visible.

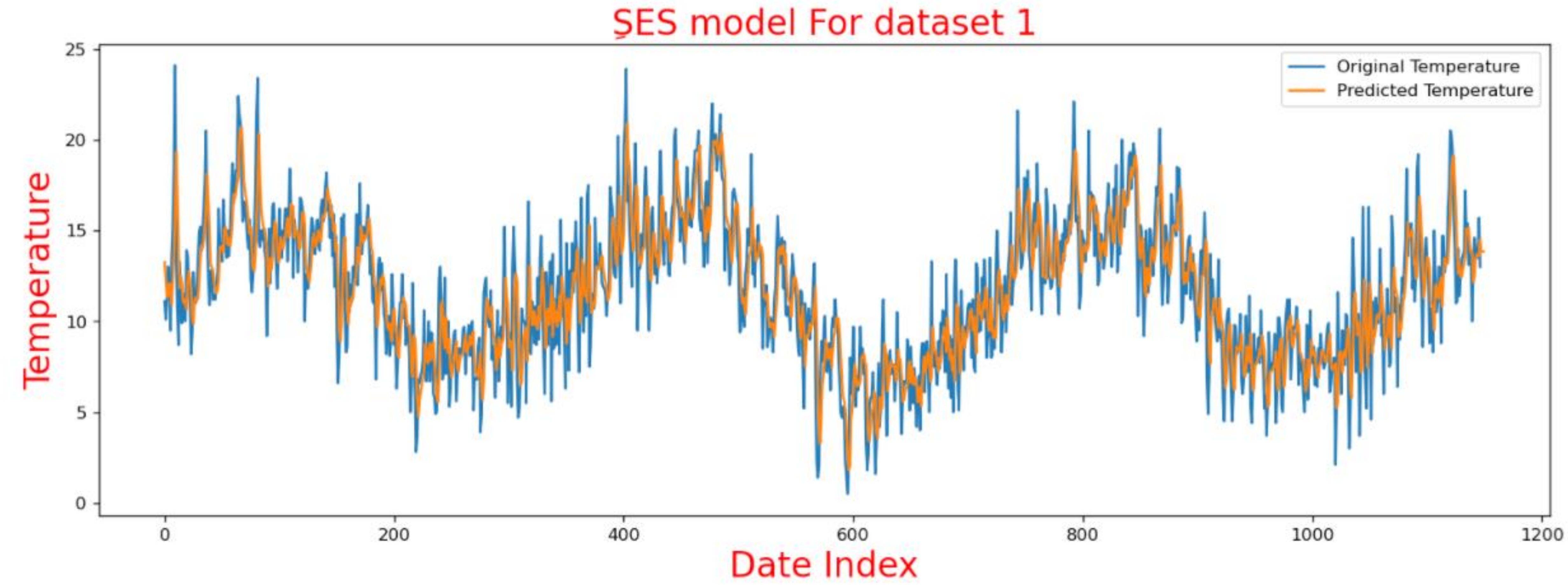
- The weight of each parameter, or decrease in weight is always determined by smoothing parameter, called as α (alpha - single parameter/hyperparameter).
- The value of α(alpha) lies between 0 to 1 such that;
- **α(alpha)=0:** signifies that future forecasted values are the average of historical data (giving more weights to historical data)
- **α(alpha)=1:** signifies that future forecast values are the results of the recent observation (giving more weights to recent observations).

- **Idea:-**Forecast future values using a weighted average of all previous values in the series.
- **Uses:-**Forecast a series with no trend and no seasonality
- **Types:-** 1-Simple exponential Smoothing (For series with no trend or seasonality) Good for our case!
- 2-Holt's Exponential Smoothing (For series with trend but no seasonality)
- 3-Winter's Exponential Smoothing (For series with trend and seasonality)
- **Key concepts:-**Smoothing Constant
- **Advantages:** Simple, Popular and adaptive0


Start
Importing Some required Libraries
Read Dataset
representing the dimensionality of the Data Frame.
Define and fit the model
Make prediction
Calculate mean square error
Plotting model and prediction
End

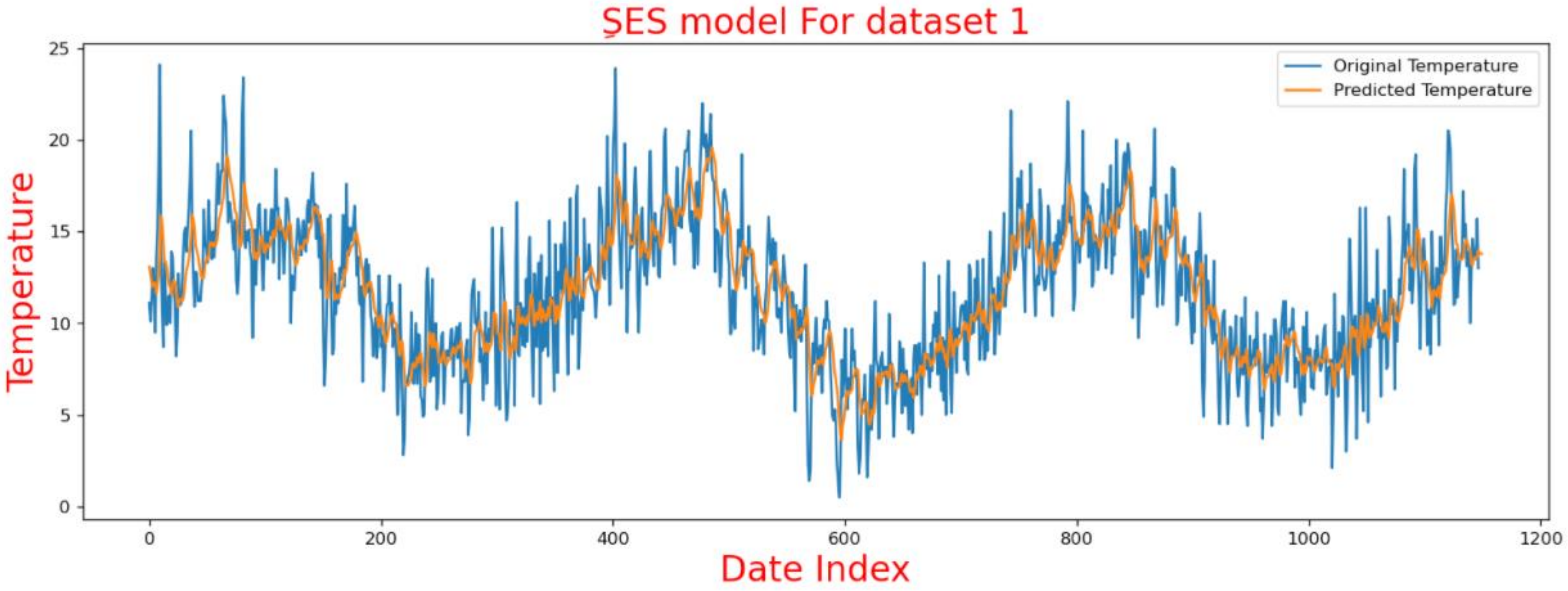
Plotted graph of first dataset

Mean_sqr_error=2.16876874275847



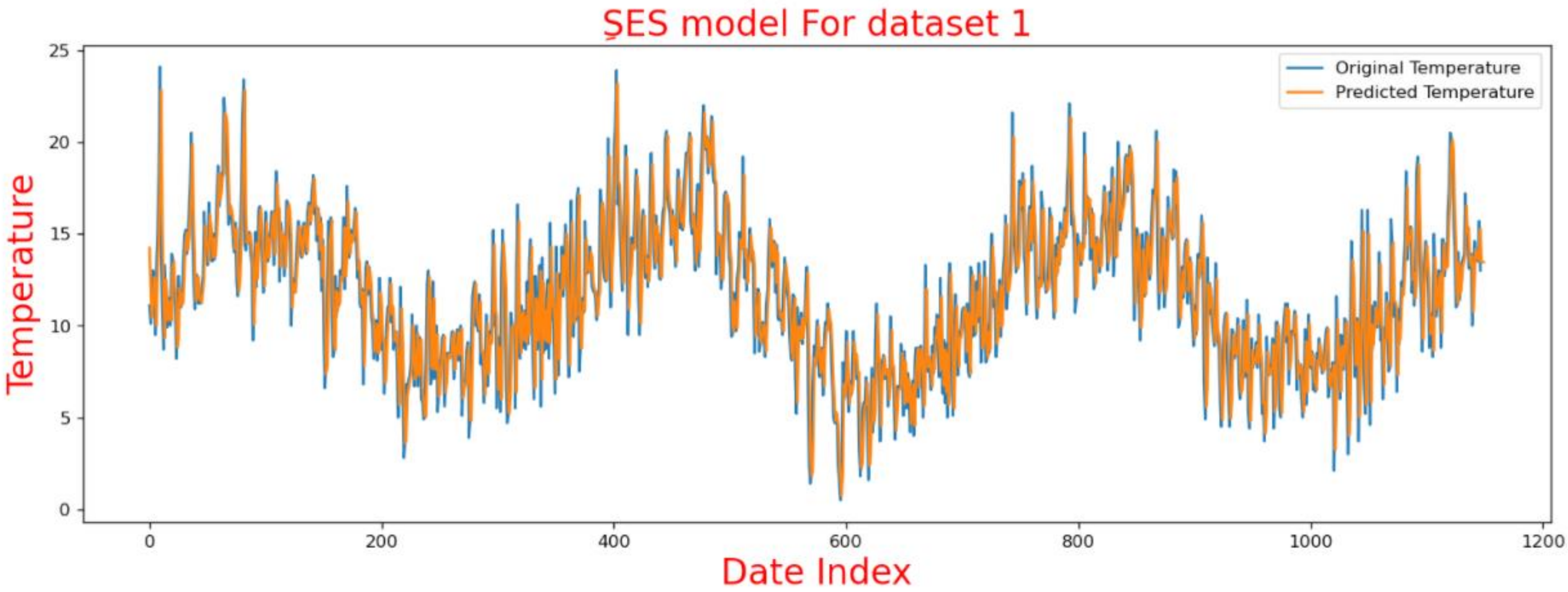
Plotted Graph of first dataset with alpha=0.2

Mean_sqr_error=2.010276443080201



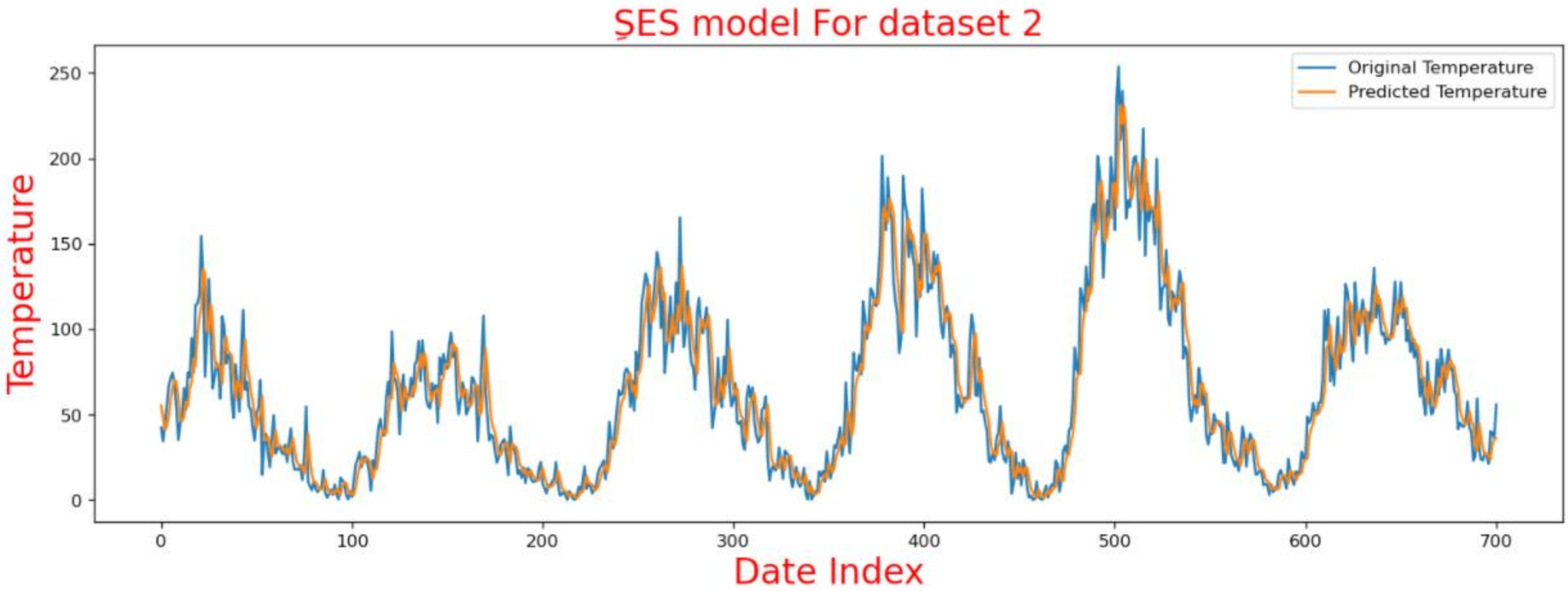
Plotted Graph of first dataset with alpha=0.8

Mean_sqr_error=2.943970631017748



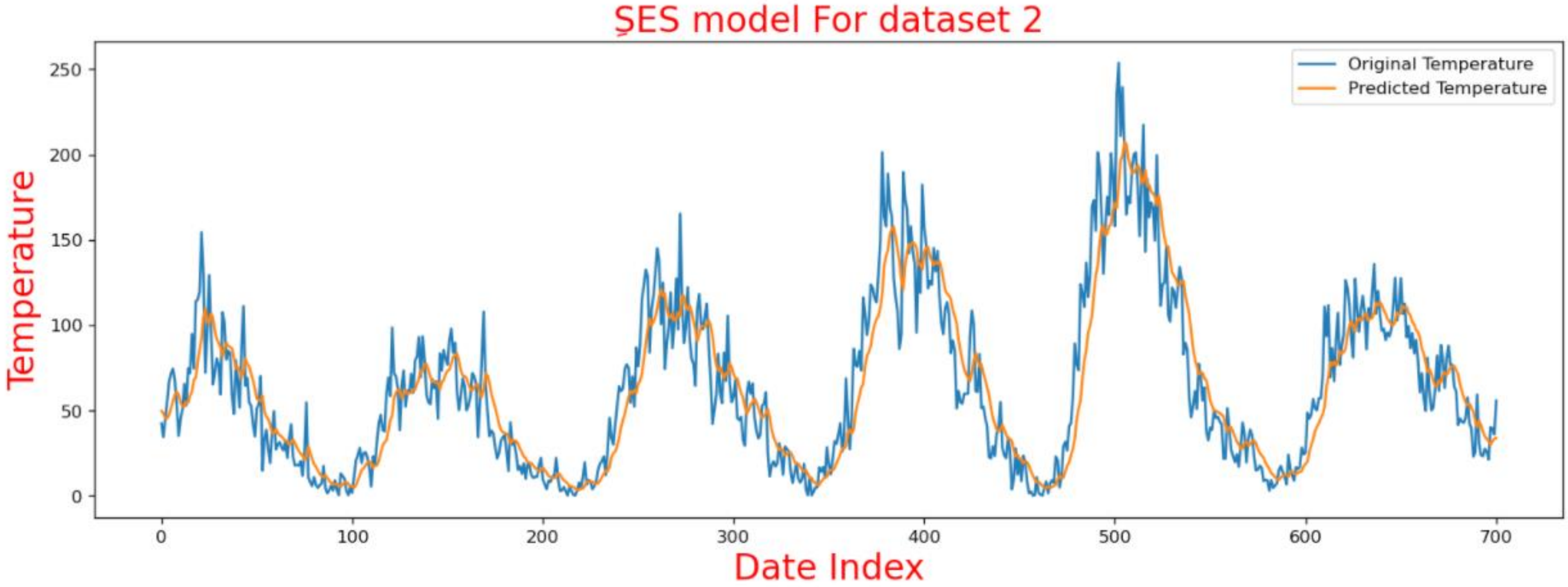
Plotted graph of second dataset

Mean_sqr_error=17.21472624307182



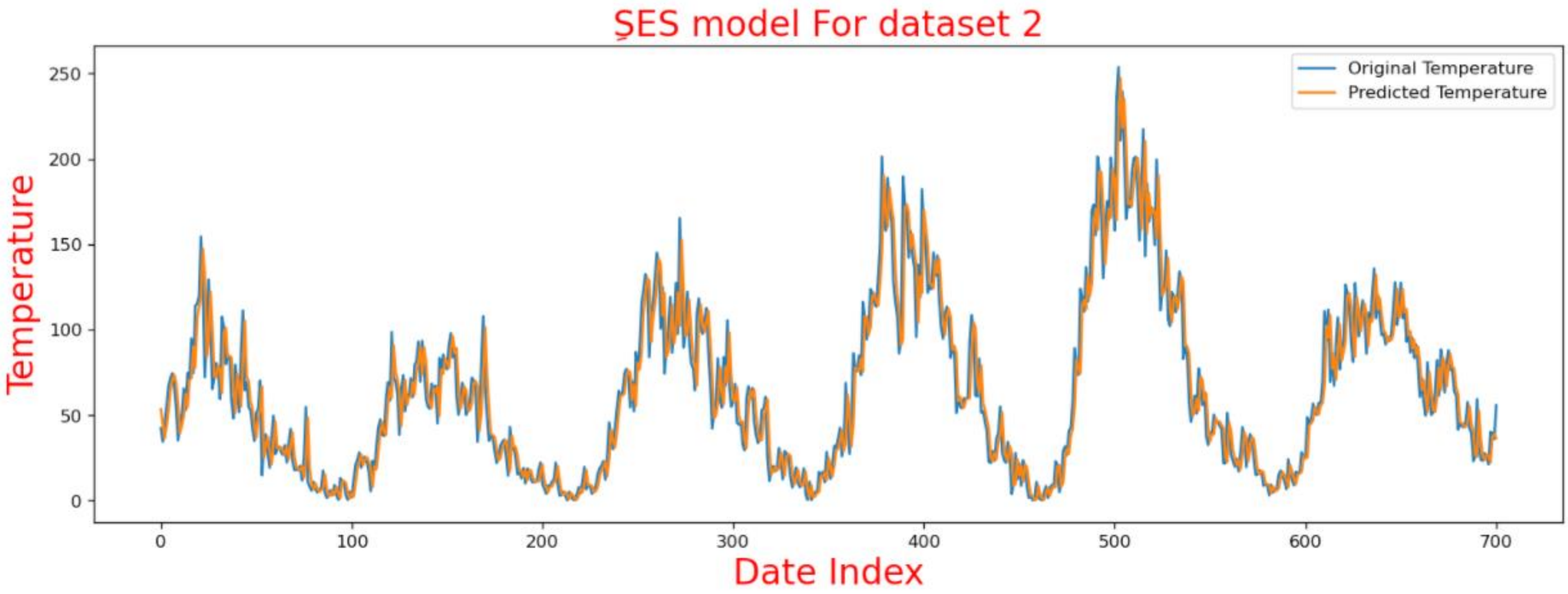
Plotted graph of second dataset with aplha=0.2

Mean_sqr_error=20.349995504549714



Plotted graph of second dataset with aplha=0.8

Mean_sqr_error=17.374454299005553



LSTM Model

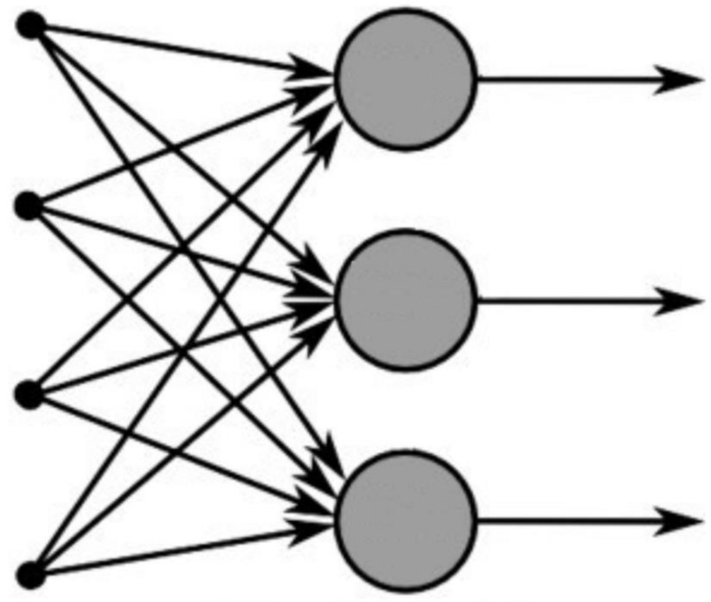
Basic Long Short Term Memory (LSTM)

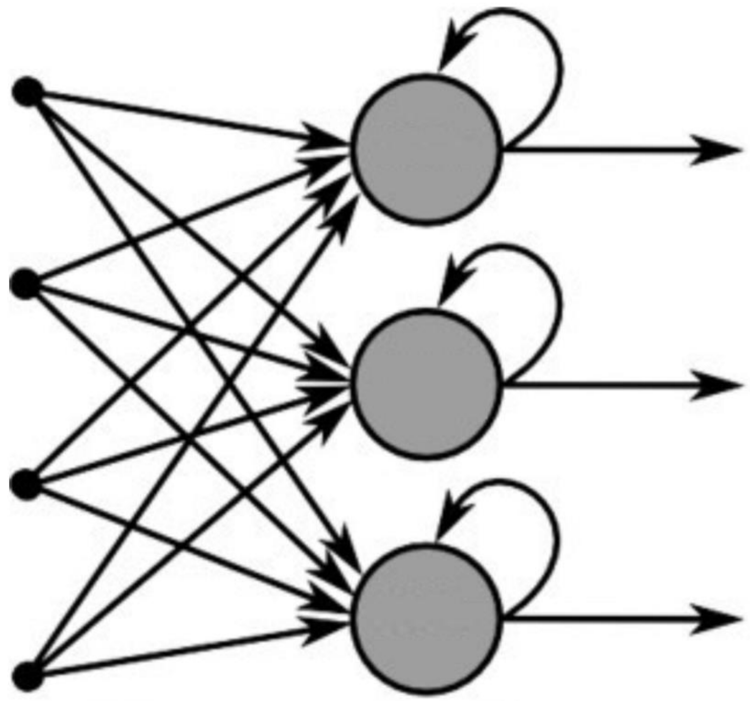
Model type	application
LSTM is a type of RNN deep learning Units that is mostly used for analysis of sequential data (time series data prediction).	There are different application areas that are used: Language model, neural machine translation, music generation, time series prediction, financial prediction, etc.

EgSA
وكالة الفضاء المصرية
Egyptian Space Agency

Standard NN

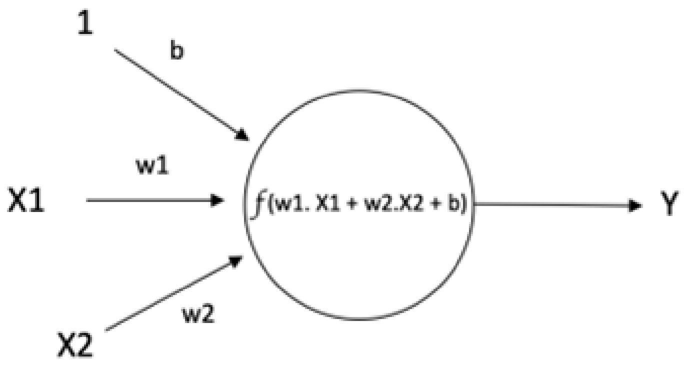
RNN





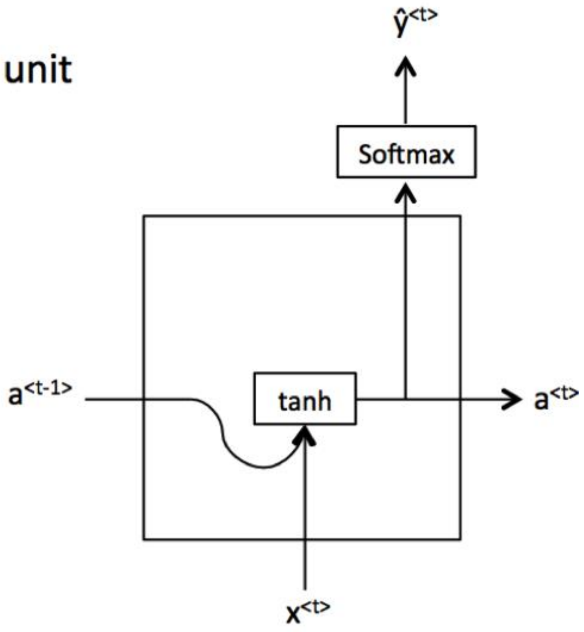
Types of NN

Standard NN | RNN | RNN with LSTM unit

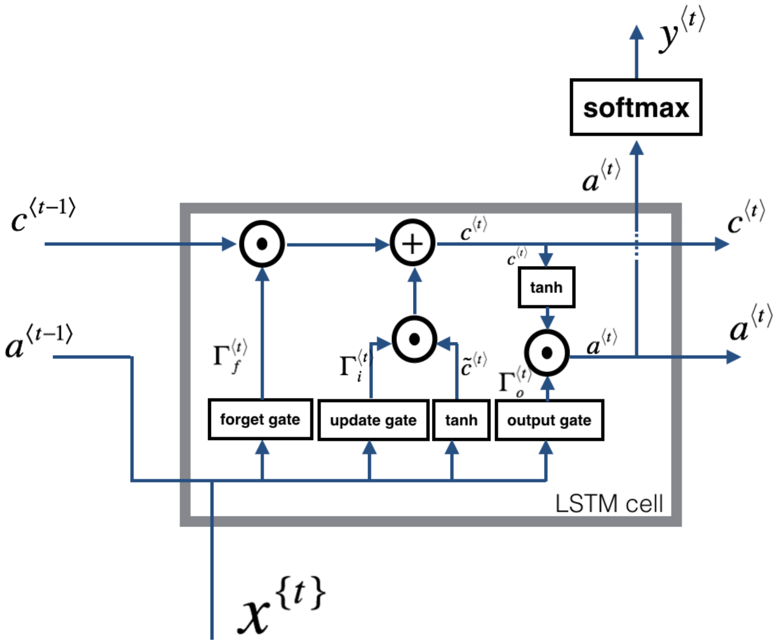


Y= f(w1. X1 + w2.X2 + b)

RNN unit



$$a^{<t>} = \tanh(W_a[a^{<t-1>}, x^{<t>}] + b_a)$$



LSTM Cell

Type of RNN units

$$\Gamma_f^{\langle t\rangle} = \sigma(W_f[a^{\langle t-1\rangle}, x^{\langle t\rangle}] + b_f)$$

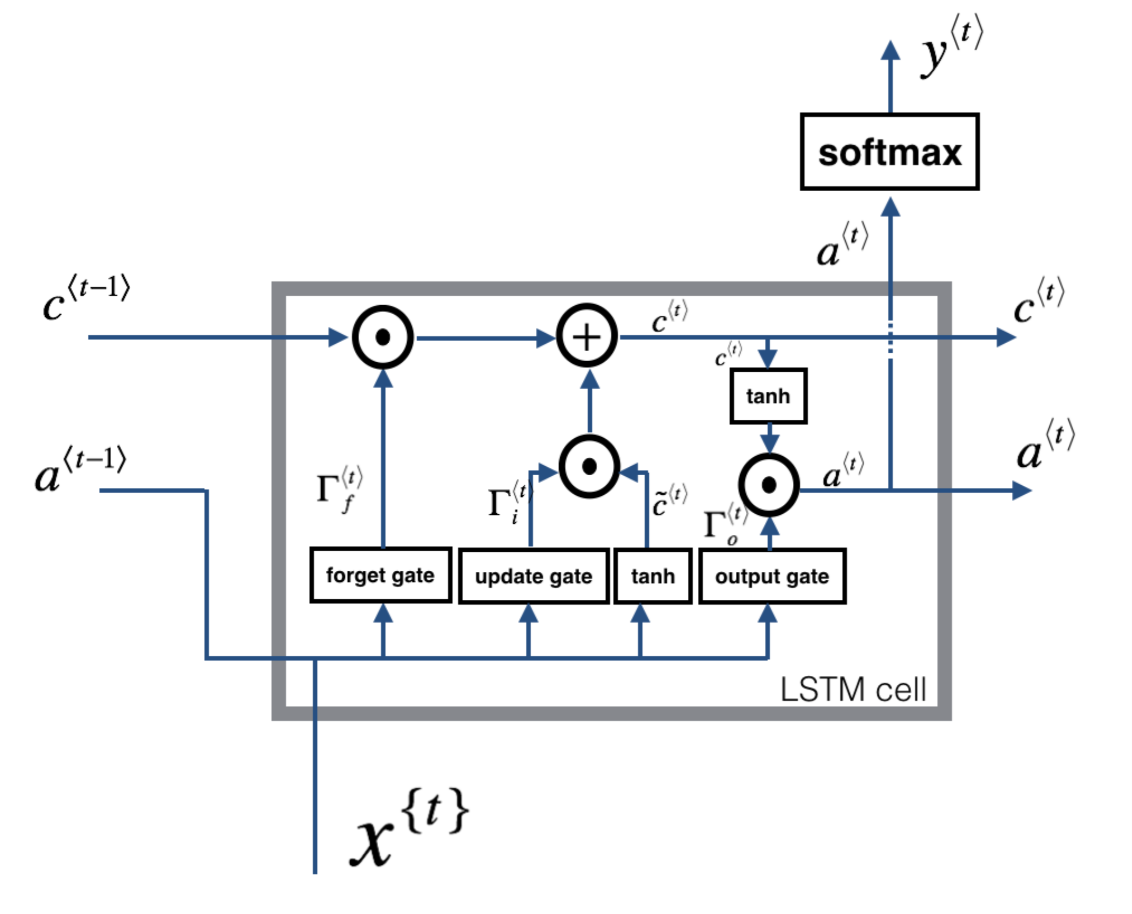
$$\Gamma_u^{\langle t\rangle} = \sigma(W_u[a^{\langle t-1\rangle}, x^{\langle t\rangle}] + b_u)$$

$$\tilde{c}^{\langle t\rangle} = \tanh(W_C[a^{\langle t-1\rangle}, x^{\langle t\rangle}] + b_C)$$

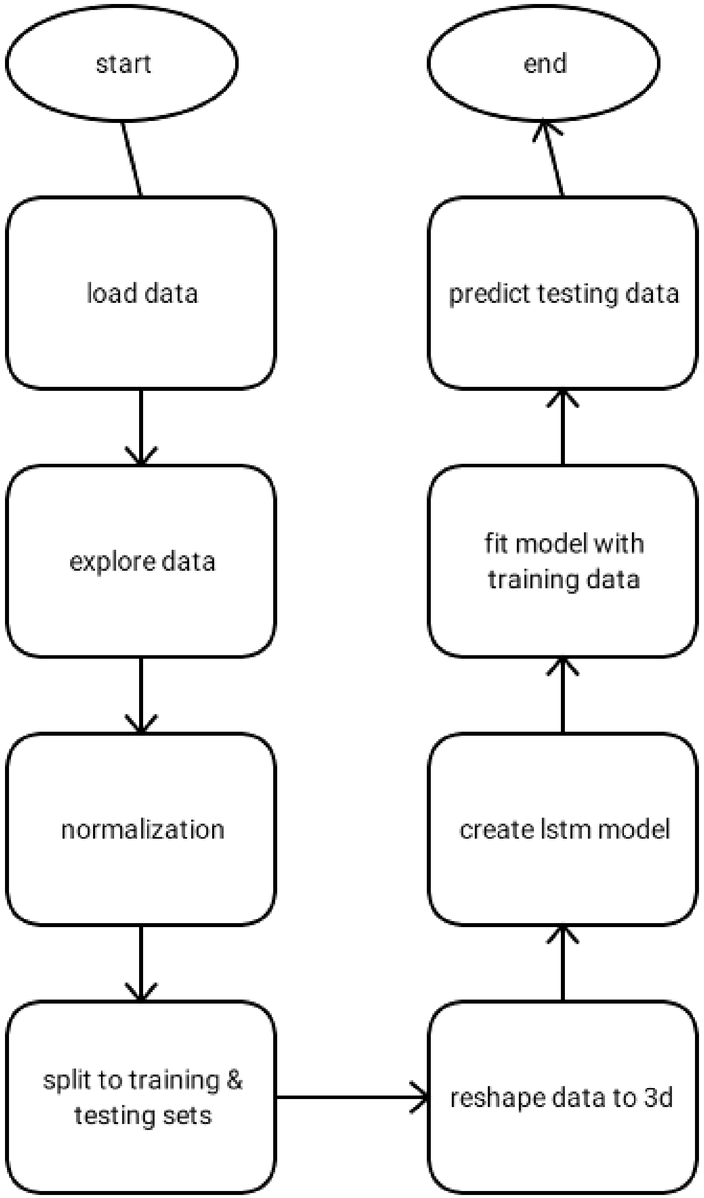
$$c^{\langle t\rangle} = \Gamma_f^{\langle t\rangle} \circ c^{\langle t-1\rangle} + \Gamma_u^{\langle t\rangle} \circ \tilde{c}^{\langle t\rangle}$$

$$\Gamma_o^{\langle t\rangle} = \sigma(W_o[a^{\langle t-1\rangle}, x^{\langle t\rangle}] + b_o)$$

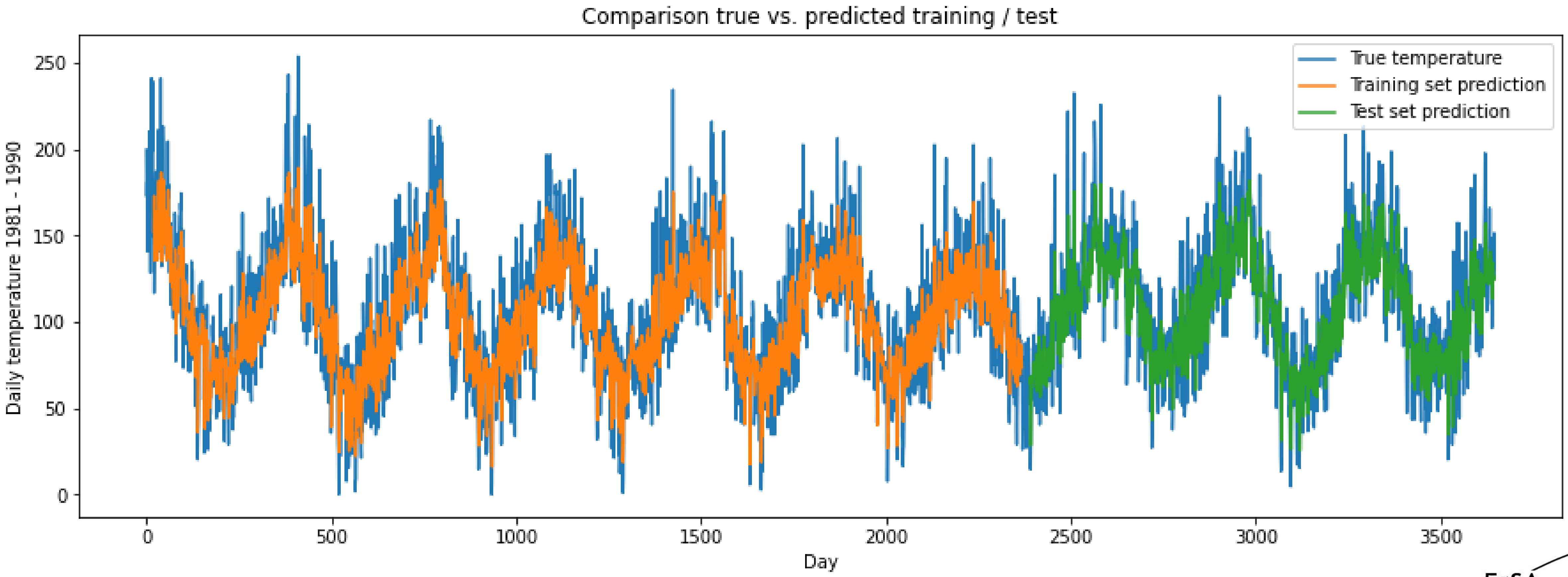
$$a^{\langle t\rangle} = \Gamma_o^{\langle t\rangle} \circ \tanh(c^{\langle t\rangle})$$



Code flow chart



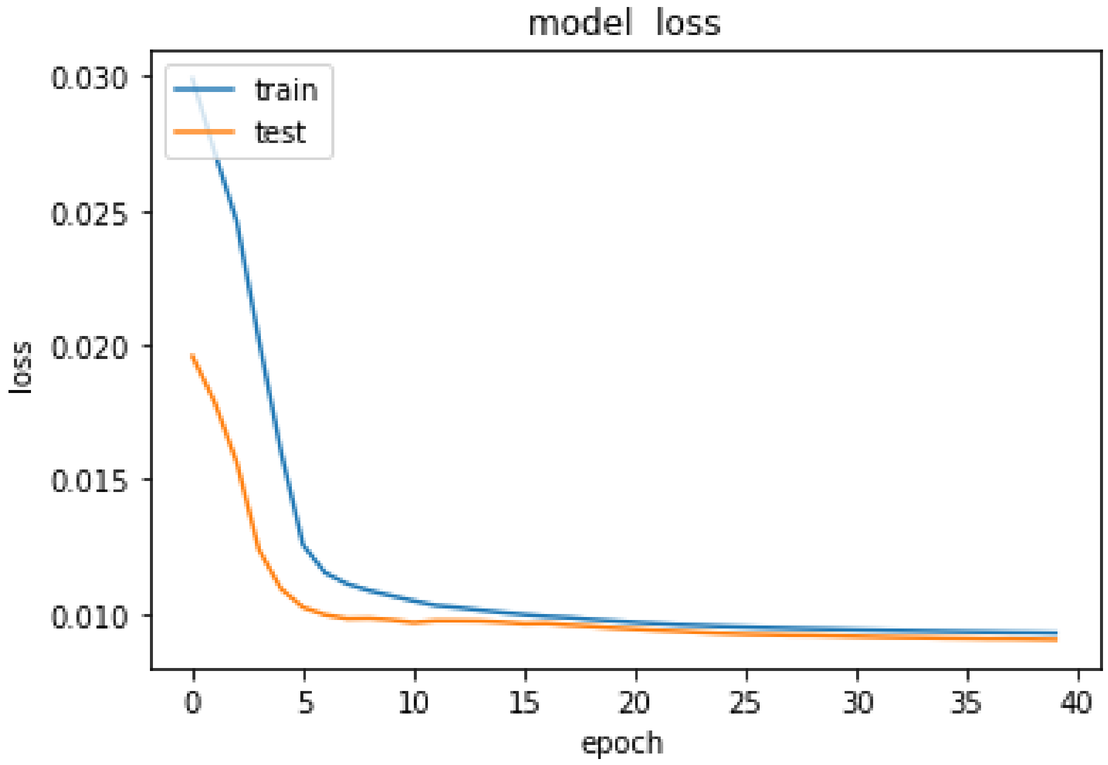
Results Dataset 1



EgSA
وكالة الفضاء المصرية
Egyptian Space Agency

Loss

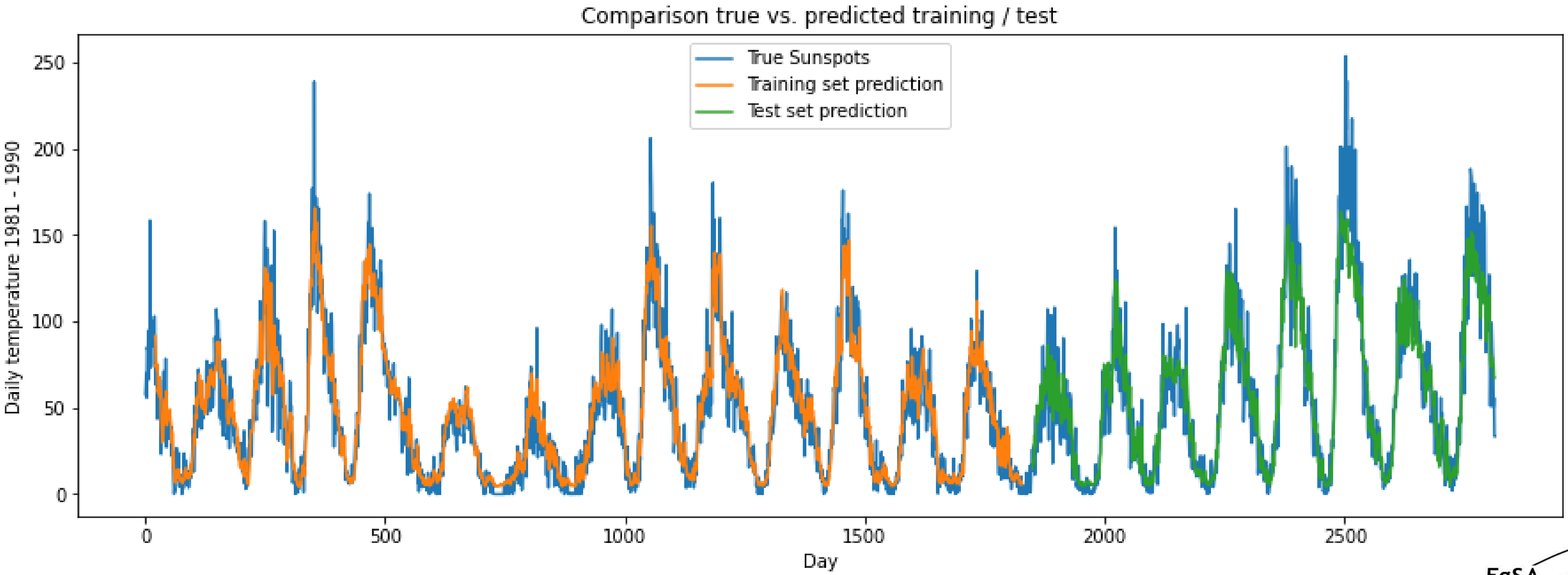
RMSE



Training data score= 23.87 RMSE

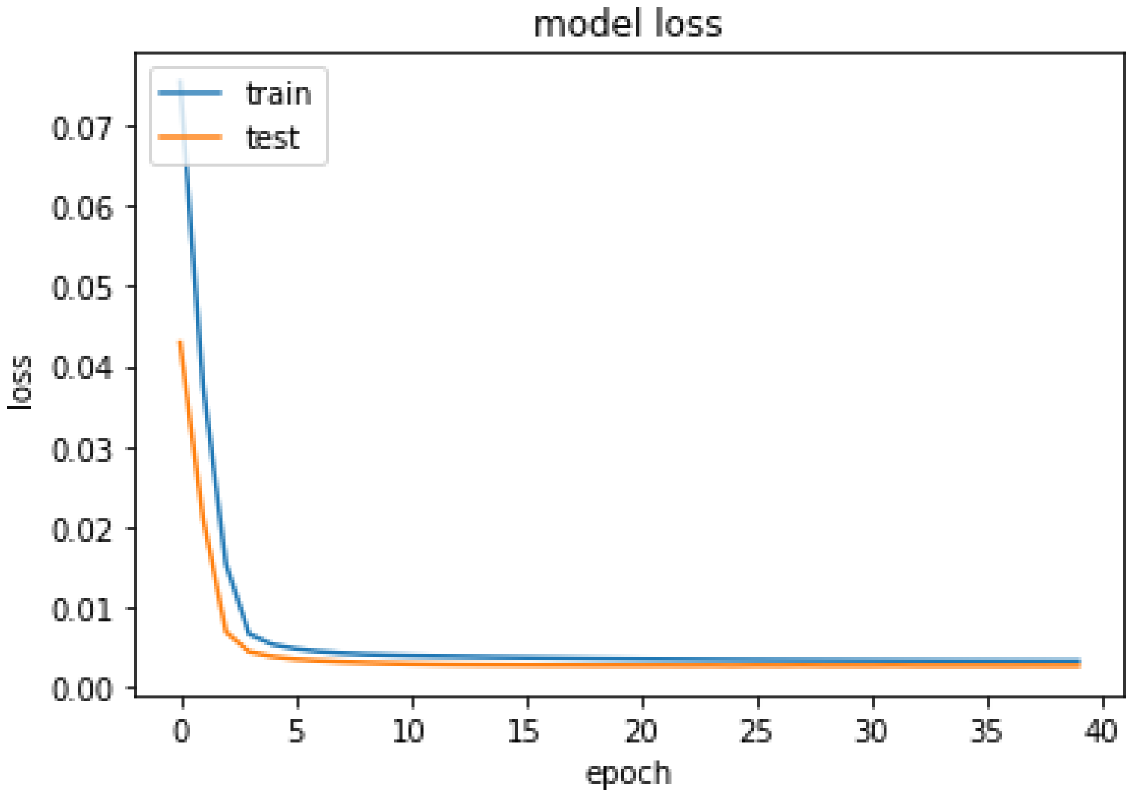
Testing data score= 23.15 RMSE

Results dataset 2



EgSA
وكالة الفضاء المصرية
Egyptian Space Agency

Loss



RMSE

Training data score= 13.86 RMSE

Testing data score= 18.49 RMSE

XGBOOST Model

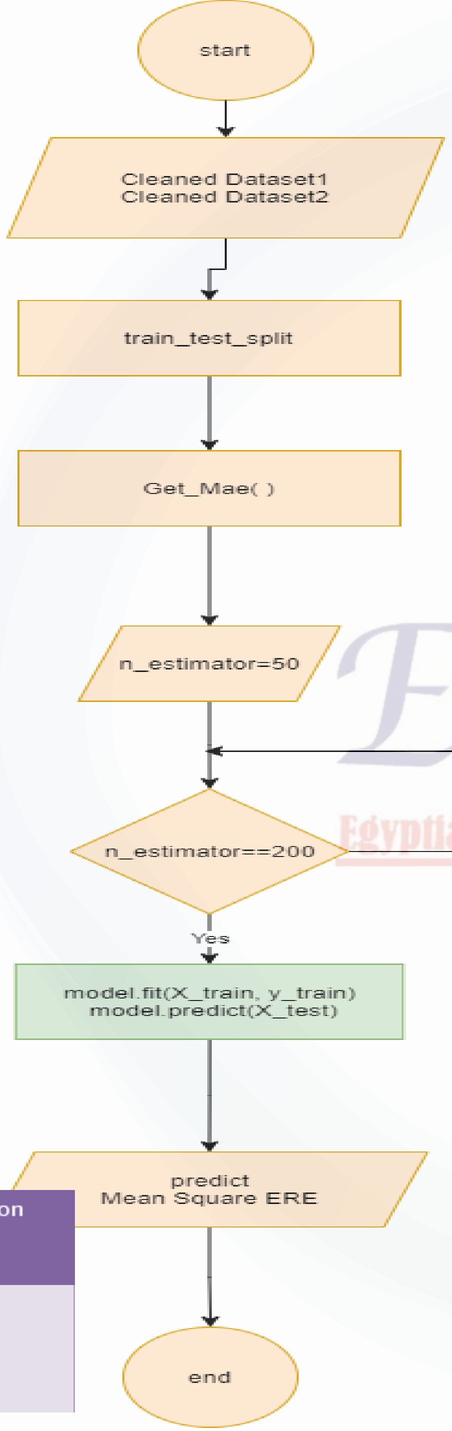
Description

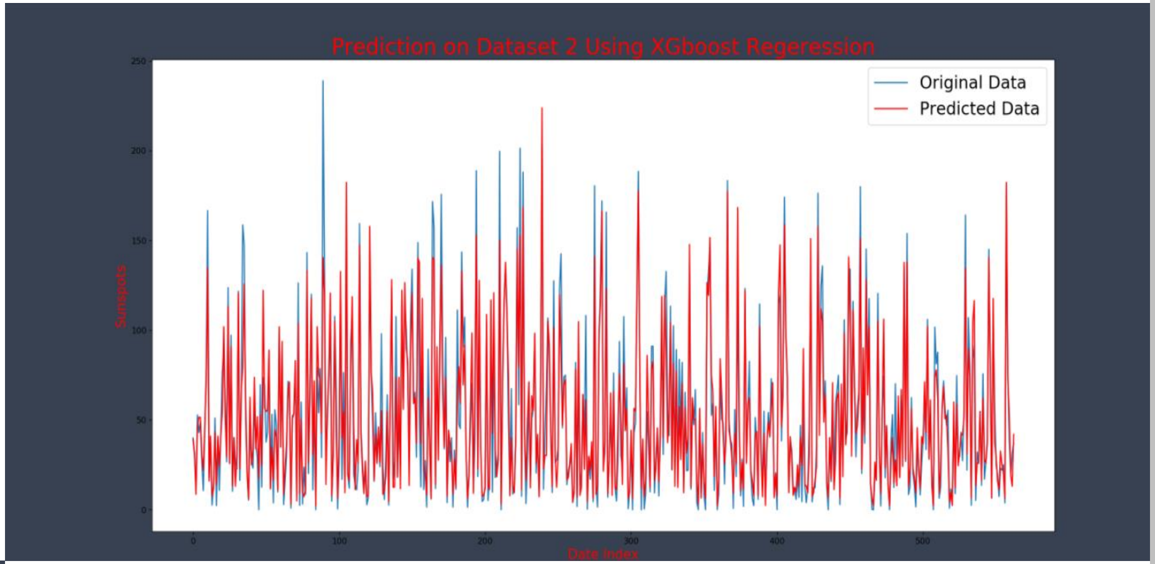
- XGBoost is an efficient implementation of gradient boosting for classification and regression problems.
- XGBoost can also be used for time series forecasting, although it requires that the time series dataset be transformed into a supervised learning problem first.

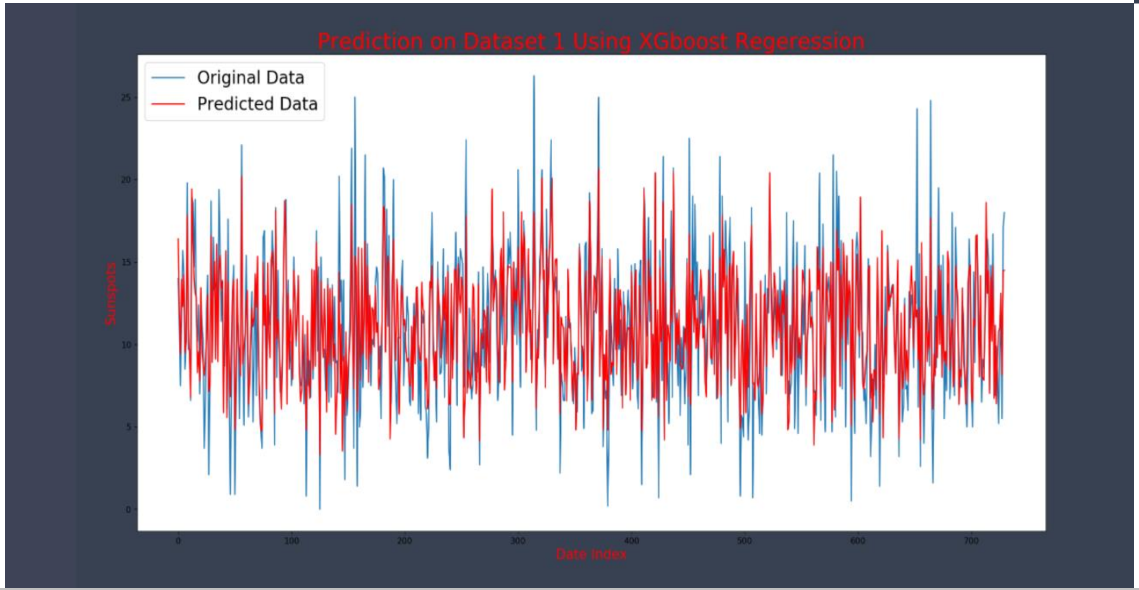
implemented in project

- -First thig we Give Index Instead of Data to use XGBoost
- -then, we split data to train and test with size 0.8 to train and without random state
- - we try to find the best n_estimator through mean square error calculation then we select Best n_estimator for Dataset 1 is 100 and Dataset 2 is 70
- -after that we predict our data and here is the prediction result on plot:

Algorithm Name	Mean Square Error on Dataset 1	Mean Square Error on Dataset 2
XGBosst Regression	1568.689	2091.9695







Egyptian Space Agency

EUTS

Egyptian Universities Training Satellite

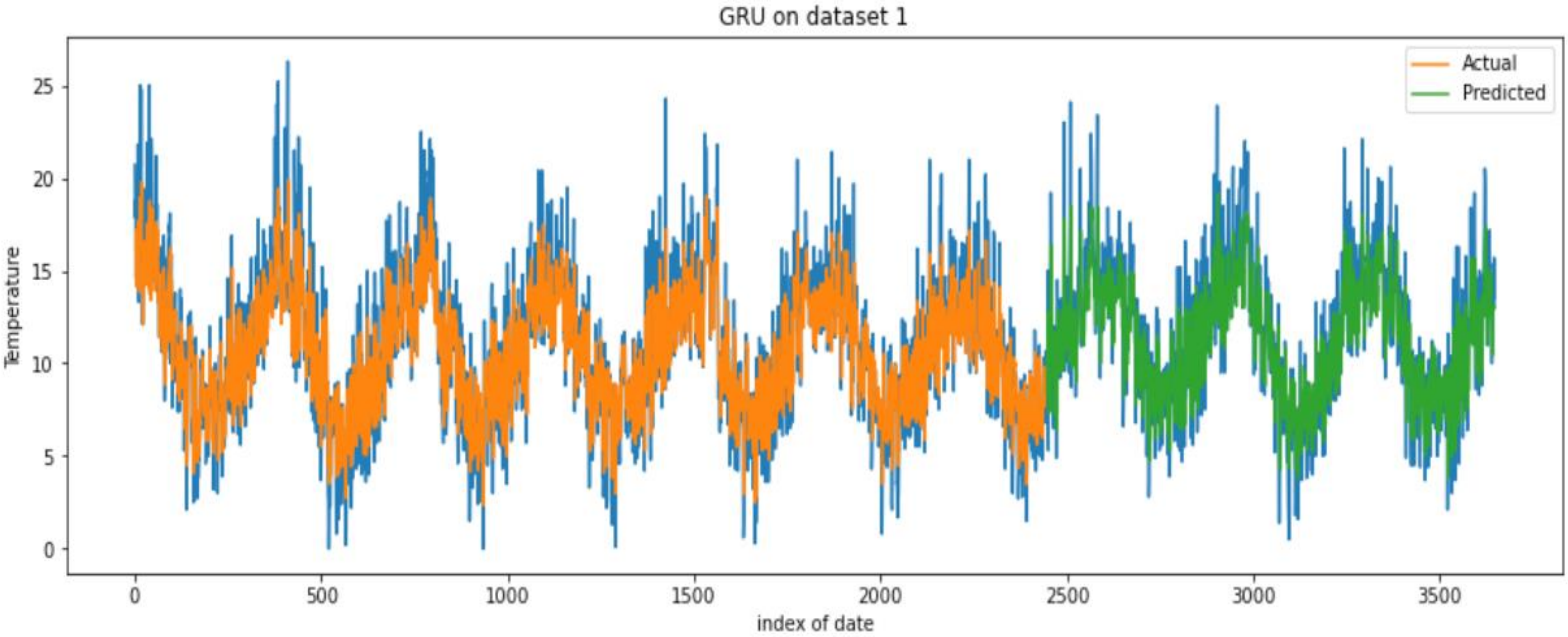
GRU Model

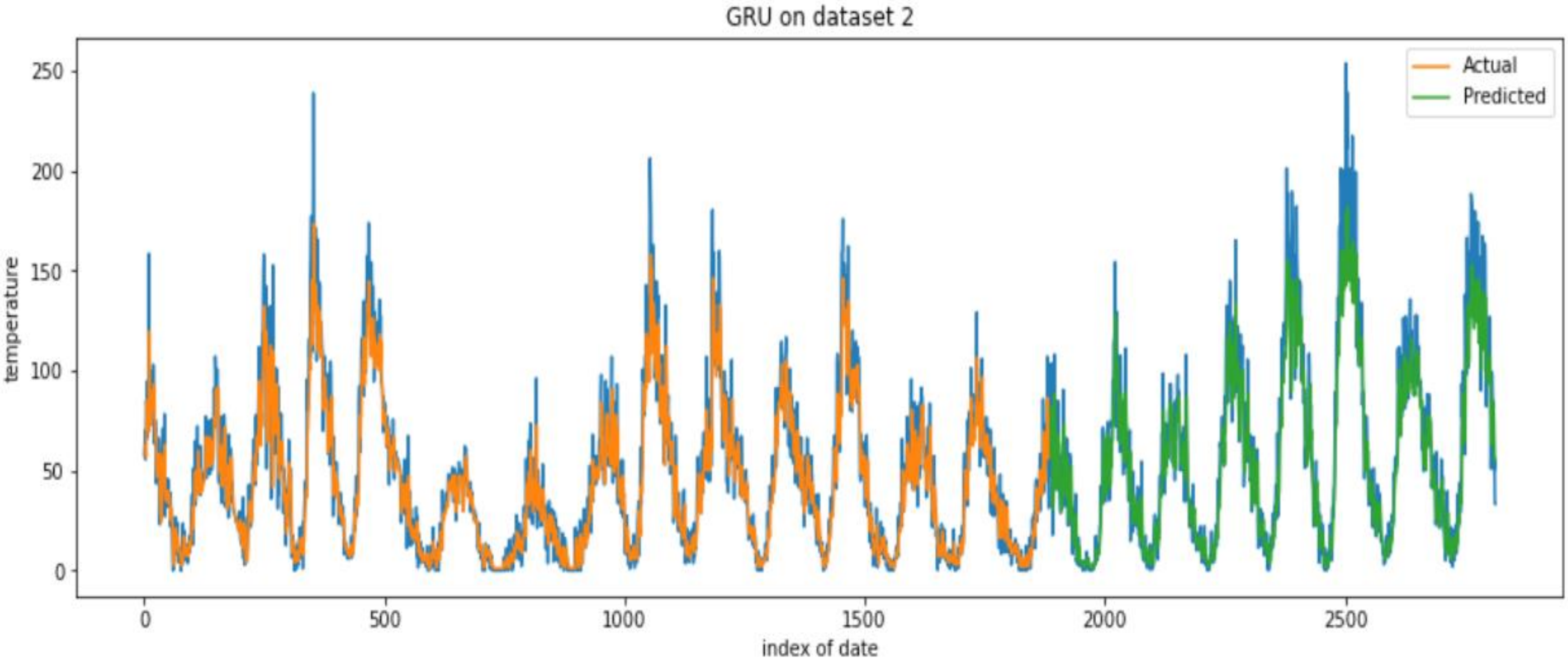
What is GRU

- Gated recurrent units (GRUs) are a gating mechanism in recurrent neural networks
- The GRU is like (LSTM) with a forget gate, but has fewer parameters than LSTM, as it lacks an output gate.
- GRUs are improved version of standard recurrent neural network to solve the vanishing gradient problem of a standard RNN

How it works

- GRU uses update gate and reset gate. Basically, these are two vectors which decide what information should be passed to the output.
- they can be trained to keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction.
- The update gate helps the model to determine how much of the past information (from previous time steps) needs to be passed along to the future.
- The reset gate is used from the model to decide how much of the past information to forget.


GRU on dataset 1
Actual
Predicted
Temperature
index of date
0
5
10
15
20
25
0
500
1000
1500
2000
2500
3000
3500


GRU on dataset 2
Actual
Predicted
temperature
index of date
0
50
100
150
200
250
0
500
1000
1500
2000
2500

Algorithm Name	Mean Square Error on Dataset 1	Mean Square Error on Dataset 2
ARIMA	4.01	62.04
ARMA	**3.987**	**62.247**
Simple Exponential Smoothing "SES"	**2.16**	**17.21**
Long Short Term Memory 'LSTM'	23.15	18.49
XGBosst Regression	**1568.689**	**2091.9695**
Gated recurrent Unit "GRU"	2.56	15.42