MIT Academy of Engineering

School of Computer Engineering & Technology

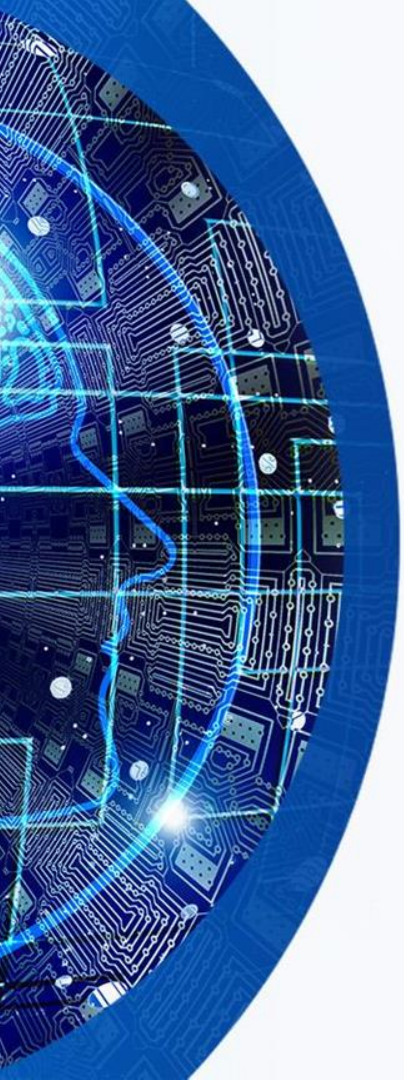
Presentation for FY Minor Project (Final Presentation)

LSTM MODEL OF CYCLONES

Guide:
Mrs. Farhana Shaikh

Veeraj Goudar (F201140)

Aditya Goutam (F201188)



INDEX

INTRODUCTION

- We know that cyclones can produce powerful winds and heavy rainfalls which can cause river floods and submergence of low lying area. So it becomes necessary to predict how many times is the cyclone going to hit in different scenarios.

- For prediction of the data we may take help of python programming language along with some frameworks, libraries and a previous year data set.

- In this project we will take 80% of data, train it so that we may predict the rest 20% of it and then compare it with the original data in the dataset by graphically representing it.

Literature Survey

- Doppler Radar
- Satellite data
- Radiosondes
- Automated surface observing systems
- AWIPS (Advanced Weather Information Processing System)
- SuperComputers

Literature Survey

Pros:

- High Quality of data
- Reliable weather forecasts
- Almost Accurate results
- Locate Precipitation
- Flood Forecasting

Literature Survey

Cons:

- Not entirely reliable
- Requires expertise to analyse
- Relies on intense datasets
- Weather Changes all times
- Needs Monitoring and is human dependent
- More interference

Literature Survey

Convolutional Neural Network:

LSTM Model is part of deep learning algorithm has already been used in weather forecasting and precipitation prediction in past few years.

These LSTM model prediction used in past few years from which the model for cyclone can be created.

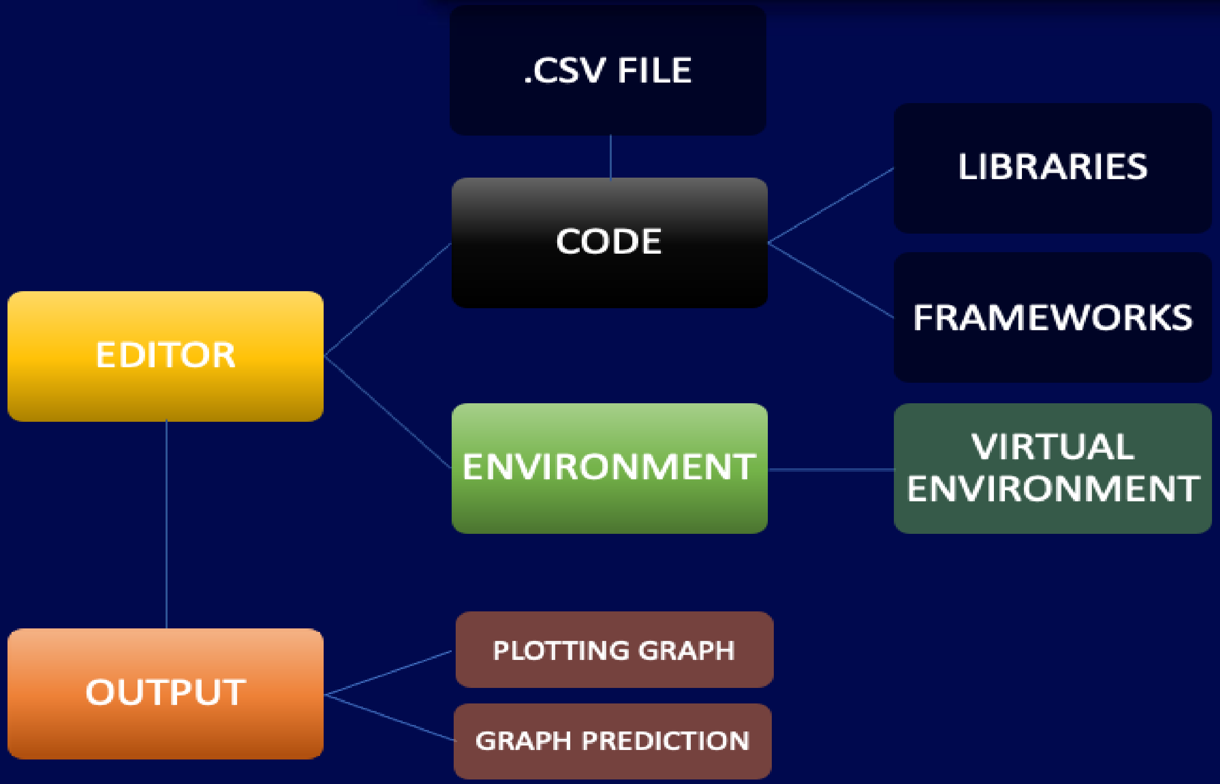
Problem statement

To make an LSTM model of the dataset of Cyclones and Severe Cyclone and create a Prediction Model and check its accuracy.

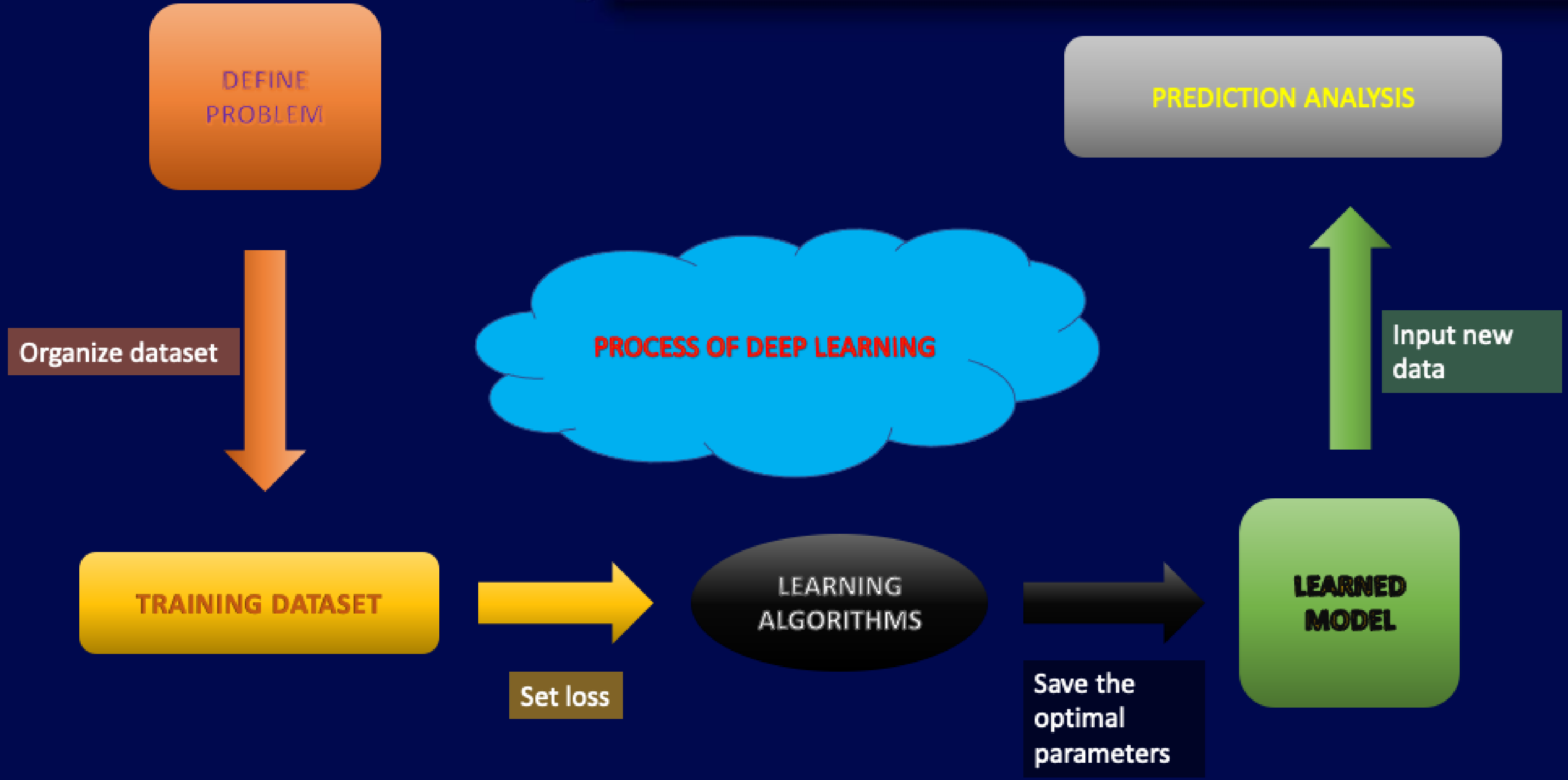
Objectives to be achieved

1. **To attain the dataset of Cyclone from past Few Centuries.**
2. **To convert the dataset into csv file and import and plot the graph.**
3. **To Create an LSTM (Long Short Term Model) model.**
4. **Using Keras as an API (Application Programming Interface).**
5. **To Use LSTM and predict the next dataset.**

SYSTEM ARCHITECTURE DIAGRAM



Block Diagram



Methodology

PLATFORMS

- Visual Studio Code
- Google Colab

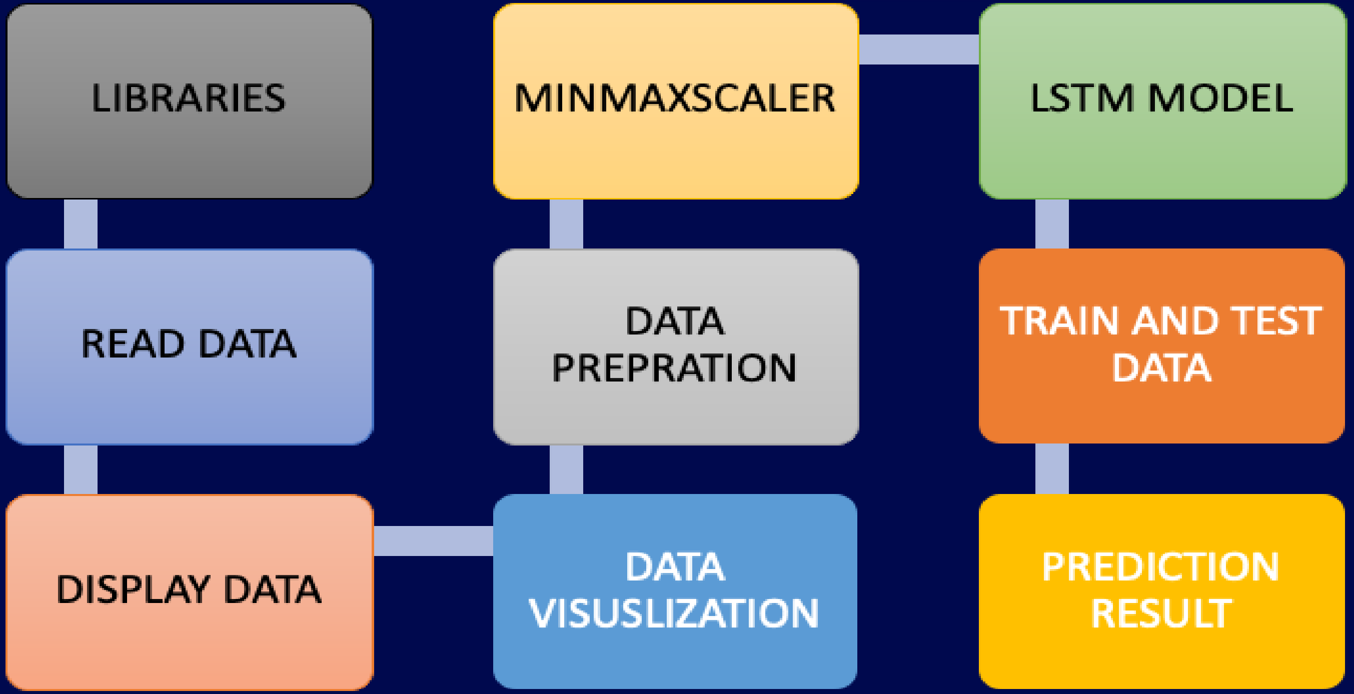
TECHNOLOGY

- Deep Learning
- TensorFlow
- Python
- Keras

Methodology

- Deep Learning contains many algorithms and one of them is LSTM (Long Short Term Memory)
- Deep learning algorithms run data through several “layers” of neural network algorithms.
- Each of which passes a simplified representation of the data to the next layer.
- Most Deep learning algorithms work well on datasets that are huge and contain hundreds and thousands of features, or columns.

Methodology



Complete Implementation

MiniProject.ipynb

+ Code + Markdown | Run All | Clear Outputs | Restart | Interrupt | Variables | ... Python 3.9.4 64-bit ('work': venv)

```python
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
%matplotlib inline

from matplotlib.pylab import rcParams
rcParams['figure.figsize']=20,10
from keras.models import Sequential
from keras.layers import LSTM,Dropout,Dense


from sklearn.preprocessing import MinMaxScaler
```

[5] ✓ 0.5s Python

```python
df=pd.read_csv("/Users/adityagoutam23/Documents/annualFrequency-1769-2019n.csv")
df_CD=pd.read_csv("/Users/adityagoutam23/Documents/annualFrequency-1769-2019n.csv")
df.tail(260)
```

[6] ✓ 0.5s Python

	Year	Cyclonic Disturbances - BOB	Cyclonic Disturbances - AS	Cyclonic Disturbances - TOTAL	Cyclones - TOTAL	Severe Cyclones - BOB	Severe Cyclones - TOTAL
0	1769	9	1	11	7	1	5
1	1770	8	2	14	4	1	5
2	1771	10	1	13	9	3	1
3	1772	8	1	16	8	0	4
4	1773	10	1	12	4	4	4
...	...	...	...	...	...	...	...
246	2015	3	5	12	4	0	2
247	2016	6	2	10	4	1	1
248	2017	8	0	10	3	2	2
249	2018	9	4	14	7	3	6
250	2019	4	8	12	8	2	7

251 rows × 7 columns

Complete Implementation

MiniProject.ipynb

\+ Code + Markdown Run All Clear Outputs Restart Interrupt Variables ...

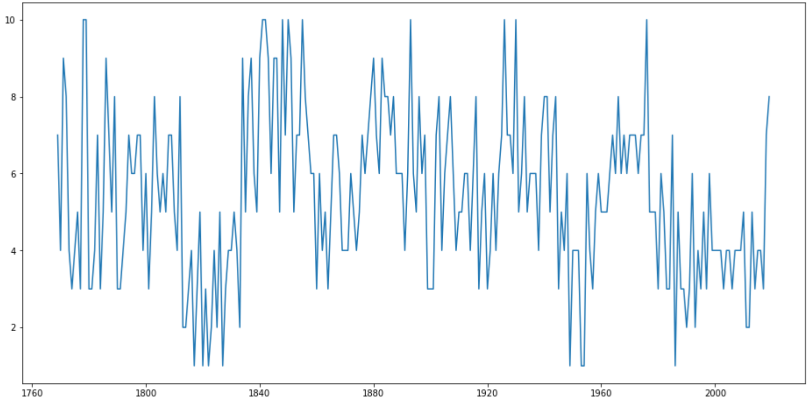
Python 3.9.4 64-bit ('work': venv)

```
df["Year"]=pd.to_datetime(df.Year,format="%Y")
df.index=df['Year']

plt.figure(figsize=(16,8))
plt.plot(df["Cyclones - TOTAL"],label='Cyclones')
```

[10] ✓ 0.9s Python

```
[<matplotlib.lines.Line2D at 0x7fbafa6df8b0>]
```




```
data=df.sort_index(ascending=True,axis=0)
new_dataset=pd.DataFrame(index=range(0,len(df)),columns=['Year','Cyclones - TOTAL'])

for i in range(0,len(data)):
    new_dataset["Year"][i]=data['Year'][i]
```

Complete Implementation

MiniProject.ipynb

+ Code + Markdown | Run All | Clear Outputs | Restart | Interrupt | Variables ... Python 3.9.4 64-bit ('work': venv)

```python
data=df.sort_index(ascending=True,axis=0)
new_dataset=pd.DataFrame(index=range(0,len(df)),columns=['Year','Cyclones - TOTAL'])

for i in range(0,len(data)):
    new_dataset["Year"][i]=data['Year'][i]
    new_dataset["Cyclones - TOTAL"][i]=data["Cyclones - TOTAL"][i]
```

[11] ✓ 0.1s Python

```python
scaler=MinMaxScaler(feature_range=(0,1))
final_dataset=new_dataset.values
train_data=final_dataset[0:200,:]
valid_data=final_dataset[200:,:]
#new_dataset.index=new_dataset.Year
#new_dataset.drop("Year",axis=1,inplace=True)
scaler=MinMaxScaler(feature_range=(0,1))
scaled_data=scaler.fit_transform(final_dataset)

x_train_data,y_train_data=[],[]

for i in range(60,len(train_data)):
    x_train_data.append(scaled_data[i-60:i,0])
    y_train_data.append(scaled_data[i,0])

x_train_data,y_train_data=np.array(x_train_data),np.array(y_train_data)

x_train_data=np.reshape(x_train_data,(x_train_data.shape[0],x_train_data.shape[1],1))
```

[14] ✓ 0.5s Python

```python
lstm_model=Sequential()
lstm_model.add(LSTM(units=50,return_sequences=True,input_shape=(x_train_data.shape[1],1)))
lstm_model.add(LSTM(units=50))
lstm_model.add(Dense(1))

inputs_data=new_dataset[len(new_dataset)-len(valid_data)-60:].values
inputs_data=inputs_data.reshape(-1,1)
inputs_data=scaler.transform(inputs_data)
```

Complete Implementation

MiniProject.ipynb

+ Code + Markdown Run All Clear Outputs Restart Interrupt Variables ... Python 3.9.4 64-bit ('work': venv)

```python
lstm_model=Sequential()
lstm_model.add(LSTM(units=50,return_sequences=True,input_shape=(x_train_data.shape[1],1)))
lstm_model.add(LSTM(units=50))
lstm_model.add(Dense(1))

inputs_data=new_dataset[len(new_dataset)-len(valid_data)-60:].values
inputs_data=inputs_data.reshape(-1,1)
inputs_data=scaler.transform(inputs_data)

lstm_model.compile(loss='mean_squared_error',optimizer='adam')
lstm_model.fit(x_train_data,y_train_data,epochs=1,batch_size=1,verbose=2)
```

[15] ✓ 11.8s Python

```
2021-08-22 22:39:45.532693: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following
CPU instructions in performance-critical operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-08-22 22:39:47.728342: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)

140/140 - 10s - loss: 0.0619

<keras.callbacks.History at 0x7fbadbbc3760>
```

```python
X_test=[]
for i in range(60,inputs_data.shape[0]):
    X_test.append(inputs_data[i-60:i,0])
X_test=np.array(X_test)

X_test=np.reshape(X_test,(X_test.shape[0],X_test.shape[1],1))
predicted_closing_price=lstm_model.predict(X_test)
predicted_closing_price=scaler.inverse_transform(predicted_closing_price)
```

[16] ✓ 2.8s Python

```python
lstm_model.save("saved_model.h5")
```

[17] ✓ 0.1s Python

```python
train_data=new_dataset[:200]
```

Complete Implementation

MiniProject.ipynb

+ Code + Markdown | Run All | Clear Outputs | Restart | Interrupt | Variables ...   Python 3.9.4 64-bit ('work': venv)

[17] ✓ 0.1s   Python

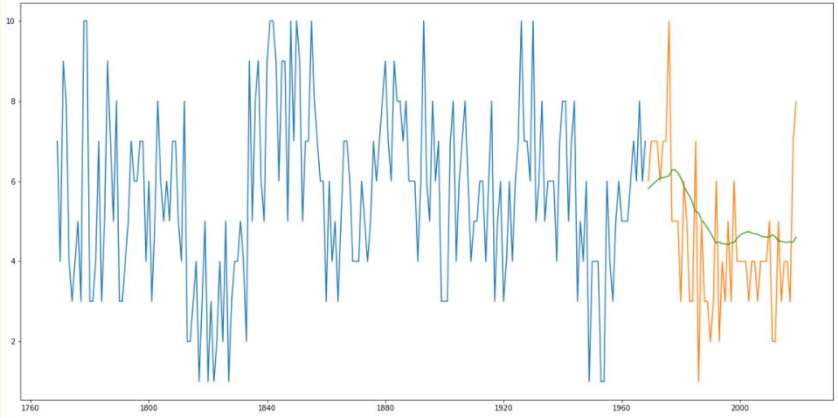
```python
train_data=new_dataset[:200]
valid_data=new_dataset[200:]
valid_data['Predictions']=predicted_closing_price
plt.plot(train_data["Cyclones - TOTAL"])
plt.plot(valid_data[['Cyclones - TOTAL',"Predictions"]])
```

[18] ✓ 1.3s   Python

```
/var/folders/qd/syw98gm56_n6lc6brp28m2ww0000gn/T/ipykernel_3576/2048899411.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  valid_data['Predictions']=predicted_closing_price

[<matplotlib.lines.Line2D at 0x7fbadf8bd490>,
 <matplotlib.lines.Line2D at 0x7fbadf8bd520>]
```



Result/Output

- As the Implementation of Code is completed we have successfully created a LSTM algorithm to predict the cyclone.

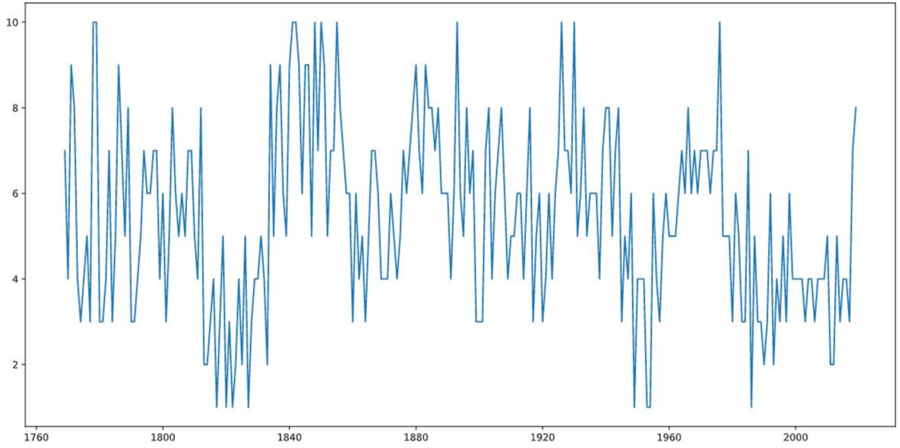
- Now, to check the accuracy of the Prediction of the LSTM
    - We are using 80% as train data for model and rest 20% as test data for comparison with our prediction

Result/Output

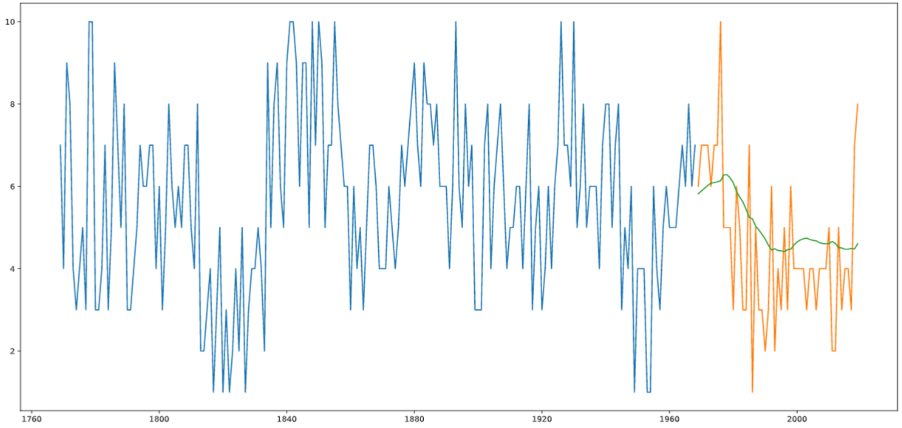
OBJECTIVES	STATUS
1. To attain the dataset of Cyclone from past Few Centuries.	☑
2. To convert the dataset into csv file and import and plot the graph.	☑
3. To Create an LSTM (Long Short Term Model) model.	☑
4. Using Keras as an API (Application Programming Interface).	☑
5. To Use LSTM and predict the next dataset.	☑

Result/Output

Cyclone Total Dataset

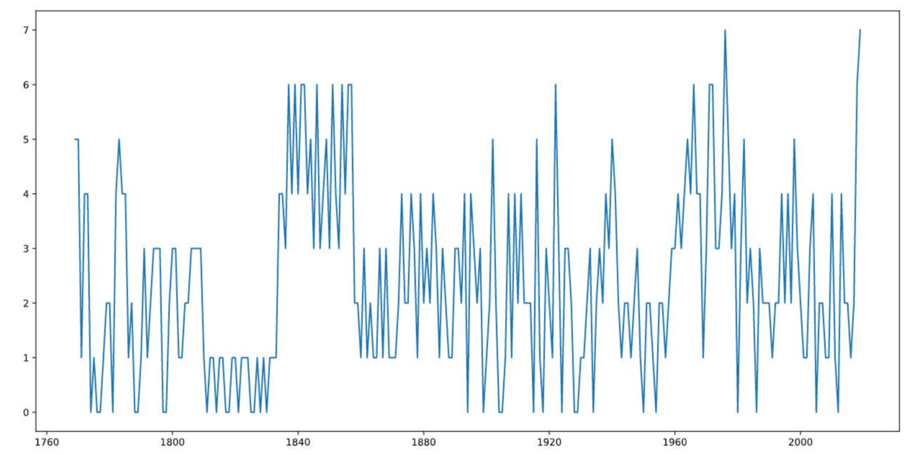


Cyclone Total After Prediction

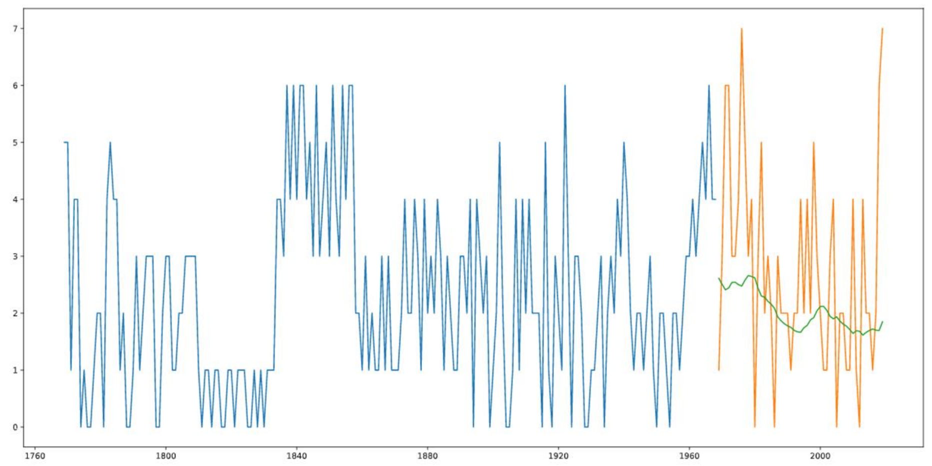


Result/Output

Severe Cyclone Total Dataset



Severe Cyclone Total After Prediction



References

https://towardsdatascience.com/practical-machine-learning-with-keras-19d0e5b2558

https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

https://www.frontiersin.org/articles/10.3389/fdata.2020.00001/full

https://data.gov.in/

https://stackoverflow.com/

https://www.mdpi.com/2073-4433/11/7/676

https://lidarradar.com/info/advantages-and-disadvantages-of-weather-radar

https://analyticsindiamag.com/tensorflow-vs-keras-which-one-should-you-choose/


ANY QUESTIONS?

THANK YOU!!!

☺