# Configuration Manual

MSc Research Project
M.Sc. in Data Analytics


Arunendra Choudhary
Student ID: x20192622

School of Computing
National College of Ireland


Supervisor: Dr. Abid Yaqoob

**National College of Ireland**
**Project Submission Sheet**
**School of Computing**

| | |
|---|---|
| **Student Name:** | Arunendra Choudhary |
| **Student ID:** | x20192622 |
| **Programme:** | M.Sc. in Data Analytics |
| **Year:** | 2022 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Dr. Abid Yaqoob |
| **Submission Due Date:** | 15/12/2022 |
| **Project Title:** | Configuration Manual |
| **Word Count:** | XXX |
| **Page Count:** | 8 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

| | |
|---|---|
| **Name:** | Arunendra Choudhary |
| **Date:** | 15th December 2022 |

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies). | ☐ |
| **Attach a Moodle submission receipt of the online project submission**, to each project (including multiple copies). | ☐ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | ☐ |

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Configuration Manual

Arunendra Choudhary
x20192622

## 1 overview

The data set is provided by the National renewal energy laboratory (NREL) which has one-year data, the data is downloaded in csv file. the data is mounted to google drive then the data is read to check the index value.

In order to determine "wind energy prediction and minimizing its effect on birds," this research is based on deep machine learning. A step-by-step tutorial is provided on how to carry out this job.

## 2 Hardware/Software Requirements

### 2.1 Hardware Requirements

the configuration of the machine on which this research is implemented is as follows:

- Operating System:macOS
- Chip: Apple M1
- Storage: 256 GB
- RAM: 8GB

### 2.2 Software Requirement

Software needed for this research:

- Integrated Development Environment: Google Colab
- Programming Language: Python 3.7
- Cloud Storage: Google Drive.
- Overleaf and Excel

## 3 Environment Setup

#### 3.0.1 Google Colab

First, open the google colab as shown in 1 and for faster performance enable the GPU for processing.

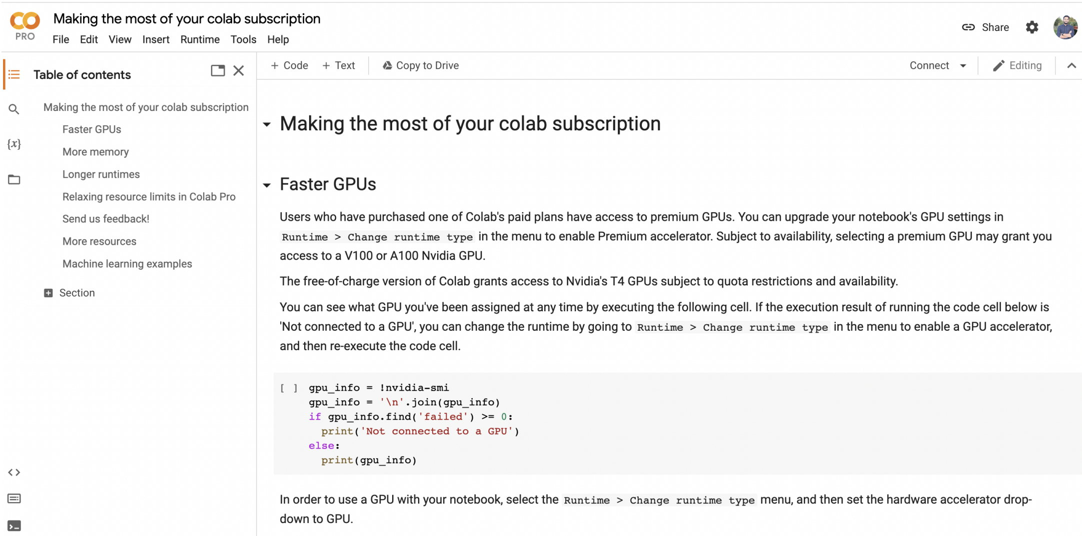


Figure 1: colab homepage

# 4 Data selection

Two datasets are used in this study and both of them can be downloaded following the following links. `https://www.kaggle.com/code/qusaybtoush1990/texas-wind-turbine-accuracy-99 data?select=TexasTurbine.csv` `https://www.kaggle.com/datasets/nelyg8002000/birds-flying`

# 5 Data transforming and Model building

## 5.1 Data Mount

Google drive is used to directly mount the data using the package provided, as shown in figure

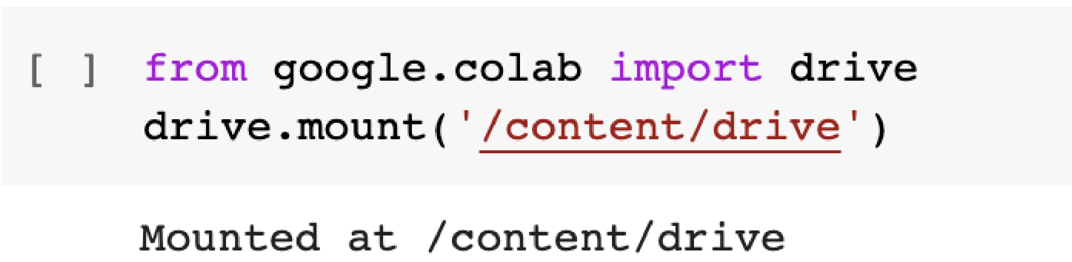


Figure 2: mount drive

## 5.2 package installations and importing libraries

python libraries are required to run and create prediction models. pip install library-name == library-version is used on Anaconda prompt. As shown in figure 3

libraries for Yolo model as shown in figure 12

```
[ ] import numpy as np
    import pandas as pd
    from pandas import read_csv
    from pandas import DataFrame
    from keras.models import Sequential
    from keras.layers import Dense
    from keras.layers import LSTM
    from pandas import Series
    from pandas import concat
    from pandas import datetime
    from math import sqrt
    from matplotlib import pyplot
    from sklearn.metrics import mean_squared_error
    from sklearn.preprocessing import MinMaxScaler
```

Figure 3: mount drive

## 5.3 Data read, transform and splitting

### 5.3.1 Loading data

the data is loaded as shown in figure 5

### 5.3.2 Train and Test

Data is divided into train and test for 70 : 30 as shown in figure 6

# 6 Model Implementation

### 6.0.1 Long short term memory

LSTM model is applied in first model fit on the dataset to see data is best fitted of the model. 10 LSTM are used for 500 entries as shown in figure7

### 6.0.2 Walk forward validation is applied

one step forcasting is done replacing the values for predection, invert scalling and differencing is done as shown in figure 8

### 6.0.3 How to clone YOLO from Github

Yolo v5 has been clone directly from the github as shown in figure 9

For bird image detection small yolo model is used because it have small data set on which it will give better result as shown in figure10

```
### install all requirements

!pip install -r requirements.txt

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting gitpython
  Downloading GitPython-3.1.29-py3-none-any.whl (182 kB)
     |████████████████████████████████| 182 kB 4.3 MB/s
Requirement already satisfied: ipython in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt (line
Requirement already satisfied: matplotlib>=3.2.2 in /usr/local/lib/python3.8/dist-packages (from -r requirements.
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt
Requirement already satisfied: opencv-python>=4.1.1 in /usr/local/lib/python3.8/dist-packages (from -r requiremen
Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt
Requirement already satisfied: psutil in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt (line 1
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.8/dist-packages (from -r requirements.t
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt (
Collecting thop>=0.1.1
  Downloading thop-0.1.1.post2209072238-py3-none-any.whl (15 kB)
Requirement already satisfied: torch>=1.7.0 in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt (
Requirement already satisfied: torchvision>=0.8.1 in /usr/local/lib/python3.8/dist-packages (from -r requirements
Requirement already satisfied: tqdm>=4.64.0 in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt (
Requirement already satisfied: tensorboard>=2.4.1 in /usr/local/lib/python3.8/dist-packages (from -r requirements
Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/python3.8/dist-packages (from -r requirements.txt
Requirement already satisfied: seaborn>=0.11.0 in /usr/local/lib/python3.8/dist-packages (from -r requirements.tx
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=3.2.
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.8/dist-packages
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=3
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-packages (from matplotlib>=3.2.2->-r
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests>=2.23.
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests>=2.23.0
tely or in another tab.   Show diff   sfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.8/dist-packages
sfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests>=2.23.0->-r
```

Figure 4: Libraries

| Time stamp | System power generated \| (kW) | Wind speed \| (m/s) | Wind direction \| (deg) | Pressure \| (atm) | Air temperature \| ('C) |
|---|---|---|---|---|---|
| Jan 1, 12:00 am | 1766.64 | 9.926 | 128 | 1.000480 | 18.263 |
| Jan 1, 01:00 am | 1433.83 | 9.273 | 135 | 0.999790 | 18.363 |
| Jan 1, 02:00 am | 1167.23 | 8.660 | 142 | 0.999592 | 18.663 |
| Jan 1, 03:00 am | 1524.59 | 9.461 | 148 | 0.998309 | 18.763 |
| Jan 1, 04:00 am | 1384.28 | 9.184 | 150 | 0.998507 | 18.963 |

Figure 5: Data file

#### 6.0.4 Download Small Yolo

# 7 Model Evaluation

The model is evaluated on the bases of Mean square error, Root mean square error and the graph plot between two comparing value as shown in figure11

## 7.1 Bird Detection

As we can see our model has detect bird which is shown in figure 13

```
[ ] # scale train and test data to [-1, 1]
    def scale(train, test):
        # fit scaler
        scaler = MinMaxScaler(feature_range=(-1, 1))
        scaler = scaler.fit(train)
        # transform train
        train = train.reshape(train.shape[0], train.shape[1])
        train_scaled = scaler.transform(train)
        # transform test
        test = test.reshape(test.shape[0], test.shape[1])
        test_scaled = scaler.transform(test)
        return scaler, train_scaled, test_scaled
```

```
[ ] # inverse scaling for a forecasted value
    def invert_scale(scaler, X, value):
        new_row = [x for x in X] + [value]
        array = np.array(new_row)
        array = array.reshape(1, len(array))
        inverted = scaler.inverse_transform(array)
        return inverted[0, -1]
```

Figure 6: Libraries

```
# fit an LSTM network to training data
def lstm_fun(train, batch_size, nb_epoch, neurons):
    X, y = train[:, 0:-1], train[:, -1]
    X = X.reshape(X.shape[0], 1, X.shape[1])
    model = Sequential()
    model.add(LSTM(neurons, batch_input_shape=(batch_size, X.shape[1], X.shape[2]), stateful=True))
    model.add(Dense(1))
    model.compile(loss='mean_squared_error', optimizer='adam')
    for i in range(nb_epoch):
        model.fit(X, y, epochs=1, batch_size=batch_size, verbose=1, shuffle=False)
        model.reset_states()
    return model
```

Figure 7: LSTM model is applied

```
[ ] # walk-forward validation on the test data
    predictions = list()
    expectations = list()
    test_pred = list()
    for i in range(len(test_scaled)):
        # make one-step forecast
        X, y = test_scaled[i, 0:-1], test_scaled[i, -1]
        yhat = lstm_ftr(lstm_model, 1, X)#batch to 1

        # Replacing value in test scaled with the predicted value.
        test_pred = [yhat] + test_pred
        if i+1<len(test_scaled):
            test_scaled[i+1] = np.concatenate((test_pred, test_scaled[i+1, i+1:]),axis=0)
        # invert scaling
        yhat = invert_scale(scaler, X, yhat)
        # invert differencing
        yhat = inv_diff(raw_values, yhat, len(test_scaled)+1-i)
        # store forecast
        predictions.append(yhat)
        expected = raw_values[len(train) + i + 1]
        expectations.append(expected)
        print('Hour=%d, Predicted=%f, Expected=%f' % (i+1, yhat, expected))
```

Figure 8: One step forcasting

```
[ ] # walk-forward validation on the test data
    predictions = list()
    expectations = list()
    test_pred = list()
    for i in range(len(test_scaled)):
        # make one-step forecast
        X, y = test_scaled[i, 0:-1], test_scaled[i, -1]
        yhat = lstm_ftr(lstm_model, 1, X)#batch to 1

        # Replacing value in test scaled with the predicted value.
        test_pred = [yhat] + test_pred
        if i+1<len(test_scaled):
            test_scaled[i+1] = np.concatenate((test_pred, test_scaled[i+1, i+1:]),axis=0)
        # invert scaling
        yhat = invert_scale(scaler, X, yhat)
        # invert differencing
        yhat = inv_diff(raw_values, yhat, len(test_scaled)+1-i)
        # store forecast
        predictions.append(yhat)
        expected = raw_values[len(train) + i + 1]
        expectations.append(expected)
        print('Hour=%d, Predicted=%f, Expected=%f' % (i+1, yhat, expected))
```

Figure 9: Yolo installation

```
[ ] !wget https://github.com/ultralytics/yolov5/releases/download/v6.0/yolov5s.pt

    --2022-12-15 03:50:58--  https://github.com/ultralytics/yolov5/releases/download/v6.0/yolov5s.pt
    Resolving github.com (github.com)... 20.27.177.113
    Connecting to github.com (github.com)|20.27.177.113|:443... connected.
    HTTP request sent, awaiting response... 302 Found
    Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/264818686/eab38592-7168-47
    --2022-12-15 03:50:59--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/264818686/e
    Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.111.133, 185.199.108.133, 185.
    Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.111.133|:443... connected.
    HTTP request sent, awaiting response... 200 OK
    Length: 14698491 (14M) [application/octet-stream]
    Saving to: ‘yolov5s.pt’

    yolov5s.pt          100%[===================>]  14.02M  20.1MB/s    in 0.7s

    2022-12-15 03:51:00 (20.1 MB/s) - ‘yolov5s.pt’ saved [14698491/14698491]

[ ] --epochs 50 --data /content/drive/MyDrive/Thesis/birds.yaml --weights /content/yolov5/yolov5s.pt --nosave --cach
```

Figure 10: Small Yolo used

```
# line plot of observed vs predicted
pyplot.plot(raw_values[-prd_val:], label="True")
pyplot.plot(predictions, label="Predicted")
pyplot.legend(loc='upper right')
pyplot.xlabel("Number of hours")
pyplot.ylabel("Power generated by system (kW)")
pyplot.show()
```

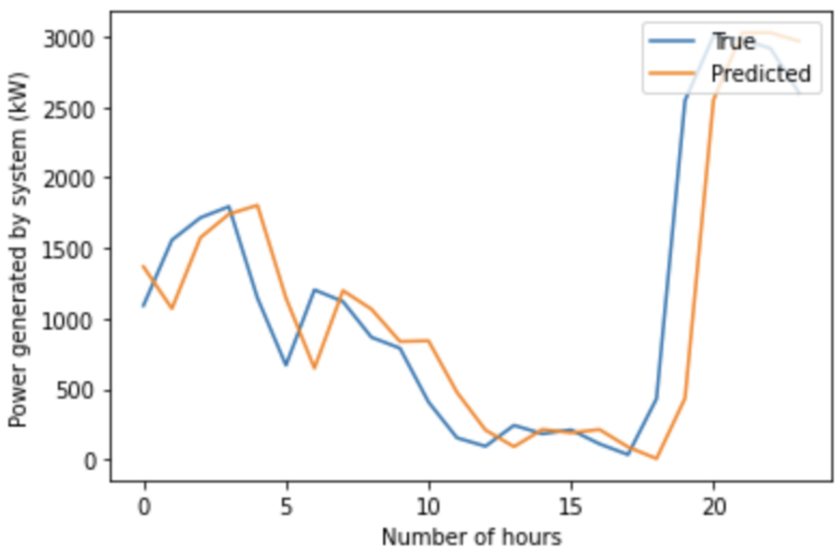


Figure 11: Graph for true and predicted value

```
[ ] print("Model 1 Mean Squared Error: %.3f" % mean_squared_error(testY, prd))

    Model 1 Mean Squared Error: 0.095

[ ] print("Model 1 Root mean squared error: %.3f" % sqrt(mean_squared_error(testY, prd)))

    Model 1 Root mean squared error: 0.308
```

Figure 12: RMSE and MSE

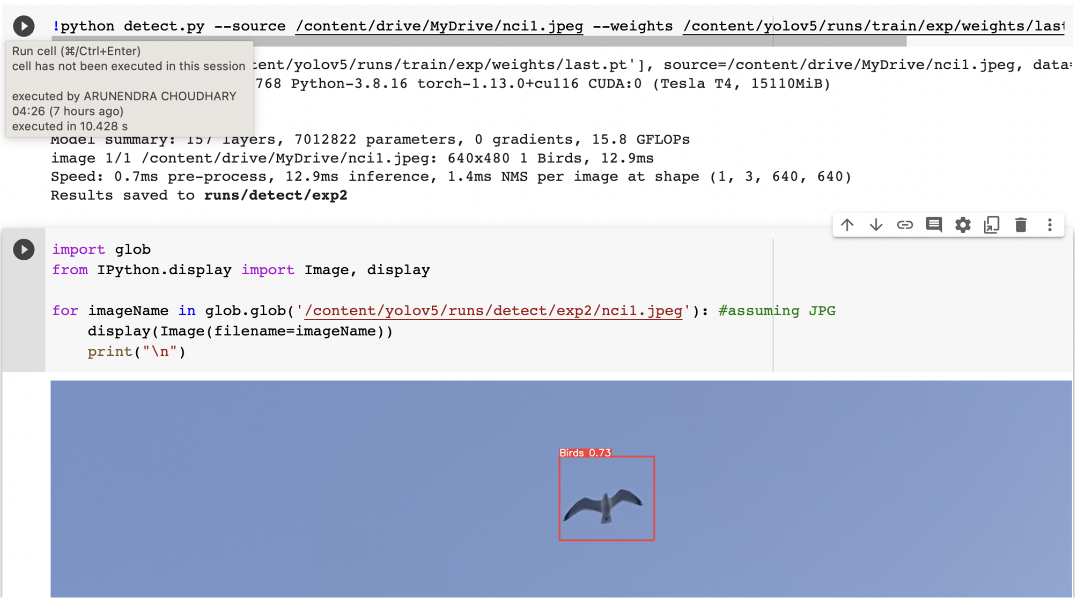


Figure 13: model Accuracy