Change History

<table>
<thead>
<tr>
<th>Issue</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0/02/2019</td>
<td>First Issue</td>
</tr>
</tbody>
</table>

Table of Contents

1. Introduction 3
2. Project Objective 4
3. Methodology 5
   3.1. Long Short-Term Memory of Recurrent Neural Networks
   3.2. Programming Tools: Python libraries
4. Project Progress 10
5. Limitations 15
6. Future Plan 16
7. Reference 17
1. **Introduction**

Tropical cyclone originates over tropical oceans where surface water is directed heated by solar radiation. The warm and moist air over the water surface rises up and condenses at the upper position to release latent heat. At the same time, the escape of air forms a low pressure region over the water surface of tropical oceans, which attracts the inflow of cooler and denser air from the surroundings. A convection effect is formed since the inflow air will be directly heated by sunlight to rise, and condense at the upper position. Such effect accumulates air mass and energy to favor the formation of a tropical cyclone. Coriolis Force due the rotation of the Earth spins the typhoon system and makes it propagates along the tropical oceans.

When tropical cyclone lands, it may bring negative effect to surrounding areas, such as rainstorm, flooding, and even serious destructions, injuries or deaths due to highly intense wind. After a tropical cyclone lands, it may eventually lose strength because of the absence of energy apply from the ocean. A tropical cyclone event must be carefully monitored and accurately predicted in order to reduce its effect by taking safety measures.

This project, a “Typhoon Track Predictor”, is proposed to be an application of machine learning techniques. “Long Short-Term Memory” (LSTM) version of Recurrent Neural Networks (RNNs) will be developed as the predictor model. Using the predictor model, it is proposed that when a set of typhoon track data over a fixed number of timestamps is input, prediction of future typhoon track over the next several timestamps will be given as output.

Hong Kong is settled at the one of the basins in Pacific Ocean where tropical cyclone usually occurs at. The developing predictor model is supposed to take past track data of the historical typhoon events which occurred at Pacific basins. The best track database which is provided by meteorological centers such as NOAA International Best Track Archive for Climate Stewardship (IBTrACS), Regional Specialized Meteorological Centre (RSMC), Joint Typhoon Warning Center (JTWC) etc. are consulted.

The following is an intermediate progress report of the project. Details of preliminary research and prototype works will be reported, including choices of methodologies and their justifications, description of current progress, limitations of the currently complete works and future plan.
2. Project Objective

The fundamental objective of the project is to develop a predictor model which takes a sequence of typhoon track data over a fixed number of timestamps and return prediction data of typhoon feature(s) over a fixed number of future timestamps as output, where time interval between each neighboring row of data is fixed and uniform.

For examples, with 6-hour intervals between data rows, a model is able to takes an input typhoon track dataset over the latest 16 timestamps (i.e. the latest 4 days) and then gives prediction of typhoon feature (e.g. latitude) to the following 8 timestamps (i.e. 2 days).

The developing model targets to takes in numeric data (in text format). To make the predictor model applicable in track prediction during a real typhoon event, the features selected should be available in real-time forecasting report which is provided by a professional meteorological center.

For examples, RSMC Tokyo gives information of typhoon features including center position, maximum sustained speed, radii of wind areas during a time interval in its real-time typhoon forecasting. Such data are all open to the public so that the collection of input data is convenient [1].

Another propose of the project is to analyze the effect to accuracy by including different typhoon features in a prediction model. The effect to accuracy by including a typhoon feature may be evaluated by its availability and measurement accuracy in real-time forecasting, and its actual effect to the model’s performance in real tests. Moreover, best track data from different data sources, where the data are available in real-time forecasting, will be consulted and compared.
3. Methodology

The project is an application of machine learning techniques where neural networks are proposed to be developed as the predictor model. It is a statistical model that does not explicitly consider the physics rules in the real world but developed by learning the data from many of the historical events [2].

Generally, a neural network model is trained by feeding a large number of true sample data. Provided input A and sample output B in a training, a neural network iterates through a large number of data batches to adjust an artificial weight matrix W such that it can calculate $B' = f(WA)$ with minimum difference with true output B. In other words, it is trying to simulate a relationship that best-fits the real potential relationship between A and B. A typhoon predictor model is trying to find out the relationship between the “previous typhoon states” and the “future typhoon states”. Neural networks in real life applications are variable more complicated which depend on proposes.

3.1. Long Short-Term Memory (LSTM) of Recurrent Neural Networks (RNNs)

Typhoon track data are time-series data that each data row in the dataset represents the state of typhoon during a certain time interval, and a sequence of track data should be in continuous time span. A number of previous typhoon states may affect the prediction of a number of future states.

The Long Short-Term Memory version of Recurrent Neural Networks is a type of neural network that is widely applied to deal with the prediction of sequential data. RNN are designed to be able to “stack” the memory when it walks through a sequential type data. It is suitable for RNN to perform prediction of sequential data such as text sentences or time-series dataset such as stock prices or weather features etc.
General Introduction to Recurrent Neural Networks

![General structure of an RNN.](image)

The major characteristic of an Recurrent Neural Network is that the prediction of the previous state \( t - 1 \) is also the one of the input parameters to the prediction in the next state \( t \). By figure 1, RNN at a certain state (timestamp) generally takes two type of input: the prediction done in previous state \( s[t - 1] \) (which is a prediction to the current state), as mentioned, and the value at current timestamp \( x[t] \). The prediction done in current state (which is a prediction to the next state) will also be the input parameter to the next state.

In prediction of future data, a RNN model is able to consult data at a number of previous steps. The training of an RNN is the find out the connection between data at previous steps and those at future steps [3].
Long Short-Term Memory

The type of neural network that the project is proposed to implement is not traditional RNNs but Long Short-Term Memory, a varied version of RNNs.

Figure 2. General Structure of an LSTM state.

By figure 2, different with traditional RNNs, a LSTM state takes an additional input: forget cell. There is a forget gate built in the logic circuit of a LSTM that selects whether the prediction result in one certain state should be passed to the next state. This modification of the structure of RNN is to reduce the problem that some irrelevant, short-term memory may corrupt the long-term memory stored in the model [3].

As the model is going to simulate a connection/relationship between the input data and output prediction, the model should be able to consult more previous timestamps instead of the last timestamp only. The addition of forget gate makes the weight of the last timestamp data not dominantly high.
3.2. **Programming Tools: Python libraries**

In the project, programming codes are written in Python to develop LSTM models. The following are the major Python software libraries those are used in development of the models and related functions:

**Tensorflow [4] and Keras [5]**
Developed by Google, Tensorflow is the one of the major software tools for developing machine learning models. It provides a platform for manipulations or applications of machine learning model, including major actions like building, training, saving and loading and performing predictions with the models.

Running on the top of Tensorflow, Keras is a high-level software API for neural network models development. Keras provides user-friendly functions which are easy to understand and manipulate. It also allows fast prototyping with analytical statistics of the models which favors evaluation of the machine learning works.

Keras provides fundamental models of different types of neural networks including LSTM such that the fundamental architecture of predictor models can be developed quickly.

**NumPy [6]**
In the project, the main function of the software library NumPy is supporting the manipulation of data type n-dimensional array (“ndarray”) which is used to store the time-series dataset of typhoon track data. Assume each row typhoon track data is with c features (columns) and there is r timestamps (rows) in the track dataset of one typhoon event. Such that each typhoon event occupies an \((r \times c)\) array. Let each typhoon event array be one single batch and there are n typhoon event datasets to train in one training. Therefore the input to model training function will be a NumPy ndarray with shape \((n \times r \times c)\). In preliminary model development r and c are fixed because fixed timestamps and number of features are required. The input ndarray is like:

NumPy has direct cooperation with Keras libraries since the ndarray inputs can be directly fitted into Keras functions. Another convenience of NumPy is that arithmetic operations such as data normalization can be intelligently applied to all elements in an ndarray object to favor the ease of data pre-processing.
Pandas [7]
The main function of Pandas libraries is to deal with the input and output of in-memory data. In the project all input datasets are sourced from or pre-processed as CSV (Comma Separated Values, can be opened with Excel Spreadsheets) files. The major data type to manipulate in Pandas is “DataFrame” which has a similar logic to Python “dictionaries”. A standard CSV file can be read in as a DataFrame object and a DataFrame object can be output as a CSV file. DataFrame can be conveniently manipulated with columns’ labels and indices.

The read in DataFrame object is further converted to NumPy ndarray for fitting the data in the Keras models.

Matplotlib [8]
The Matplotlib library is used for the visualization of statistics. It can be used to plot graphs of the prediction results with the true sample data together, along a continuous time flow (i.e. y-axis as the values of typhoon feature to predict and x-axis as the timestamp). Such that a prediction can be directly compared with the true data at the same timestamp.

After a model is trained and used to perform predictions with test input data, the result can be immediately visualized in order to favor fast evaluation to the performance of the trained model. In application of the model, the prediction result will also be visualized and displayed in a similar format.

The Matplotlib library provides additional functions that assist research and analyze of the model such as rescaling of the graph window and saving the graph in a common image format (e.g. PNG).
4. **Project Progress**

Up to the date of submission of this interim report, the project has accomplished:
- Basic architecture of LSTM model
- Design of operation flow to apply the LSTM models into prediction

Different strategies of model training (e.g. selection of typhoon features and data sources) will be kept testing and assessing in the next stage. The following is a brief demonstration of the logic flow of training a LSTM typhoon track predictor model:

**Data Source**

Different data sources that provide best track data of historical typhoon events are consulted during the project development, such as RSMC, JTWC, IBTrACS etc, and IBTrACS provides collection of best track data that are issued by meteorological centers over the world (which are the members of World Meteorological Organization (WMO)).

In this demonstration, the best track data provided by RSMC Tokyo (which is available in the website of Japan Meteorological Agency (JMA)) are used [9]. The RSMC is responsible to forecast and analyze typhoons that occur in Pacific Basins so that the data consulted are also applicable to the typhoon forecasting in Hong Kong.

**Data Pre-processing**

The source data downloaded from the JMA website is in TXT format. It is first converted to CSV format by a simple program. By contrast, the dataset provided in the IBTrACS website is in CSV format (and others) that needs no format conversion.

Typhoon events that occurred during the period between 1977 and 2019 are selected.

Four features (columns) are extracted from the dataset as the input features:
1. Latitude (of center position), with degree as unit.
2. Longitude (of center position), with degree as unit
3. Maximum Sustained Speed in 10 minutes, with knot (kt) as unit
4. Central Pressure, with hPa as unit

The above features represent part of a typhoon’s state during a time interval.

Output features: center position, where latitude and longitude are trained in two different models
Time interval = 6 hours
Input time-steps: 15 steps = 90 hours = 3.75 days
Output time-steps = 6 steps = 36 hours = 1.5 days
That means the model will take the typhoon track data at the latest 3.75 days and perform predictions of typhoon track at the following 1.5 days.

The data extracted will be normalized to favor training of weight matrix.
The program applies min-max normalization with the equation:

\[ x := (x - \text{dataset. min})/(\text{dataset. max} - \text{dataset. min}) \]

The following figures show how a portion of dataset changes after normalization.

<table>
<thead>
<tr>
<th>Date Time</th>
<th>Max Speed</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td>18/8/1977 0:00</td>
<td>35</td>
<td>20.0</td>
<td>123.0</td>
<td>998</td>
</tr>
<tr>
<td>18/8/1977 6:00</td>
<td>35</td>
<td>20.6</td>
<td>127.2</td>
<td>992</td>
</tr>
<tr>
<td>18/8/1977 12:00</td>
<td>35</td>
<td>21.0</td>
<td>128.6</td>
<td>992</td>
</tr>
<tr>
<td>18/8/1977 18:00</td>
<td>35</td>
<td>21.5</td>
<td>125.8</td>
<td>992</td>
</tr>
<tr>
<td>19/8/1977 0:00</td>
<td>36</td>
<td>22.0</td>
<td>126.0</td>
<td>992</td>
</tr>
<tr>
<td>19/8/1977 6:00</td>
<td>36</td>
<td>22.0</td>
<td>123.9</td>
<td>992</td>
</tr>
<tr>
<td>19/8/1977 12:00</td>
<td>36</td>
<td>21.7</td>
<td>122.7</td>
<td>992</td>
</tr>
<tr>
<td>19/8/1977 18:00</td>
<td>40</td>
<td>21.3</td>
<td>121.6</td>
<td>992</td>
</tr>
<tr>
<td>20/8/1977 0:00</td>
<td>40</td>
<td>21.2</td>
<td>121.1</td>
<td>992</td>
</tr>
<tr>
<td>20/8/1977 6:00</td>
<td>40</td>
<td>21.2</td>
<td>120.5</td>
<td>992</td>
</tr>
<tr>
<td>20/8/1977 12:00</td>
<td>40</td>
<td>21.2</td>
<td>119.9</td>
<td>985</td>
</tr>
<tr>
<td>20/8/1977 18:00</td>
<td>40</td>
<td>21.2</td>
<td>118.2</td>
<td>985</td>
</tr>
<tr>
<td>21/8/1977 0:00</td>
<td>35</td>
<td>21.5</td>
<td>119.0</td>
<td>985</td>
</tr>
<tr>
<td>21/8/1977 6:00</td>
<td>40</td>
<td>22.0</td>
<td>119.0</td>
<td>985</td>
</tr>
<tr>
<td>21/8/1977 12:00</td>
<td>40</td>
<td>22.6</td>
<td>119.0</td>
<td>985</td>
</tr>
<tr>
<td>21/8/1977 18:00</td>
<td>40</td>
<td>22.6</td>
<td>119.4</td>
<td>985</td>
</tr>
<tr>
<td>22/8/1977 0:00</td>
<td>40</td>
<td>22.5</td>
<td>119.7</td>
<td>985</td>
</tr>
<tr>
<td>22/8/1977 6:00</td>
<td>40</td>
<td>24.3</td>
<td>120.5</td>
<td>990</td>
</tr>
<tr>
<td>22/8/1977 12:00</td>
<td>40</td>
<td>26.3</td>
<td>122.4</td>
<td>990</td>
</tr>
<tr>
<td>22/8/1977 18:00</td>
<td>40</td>
<td>26.7</td>
<td>123.2</td>
<td>990</td>
</tr>
<tr>
<td>23/8/1977 0:00</td>
<td>40</td>
<td>28.3</td>
<td>124.0</td>
<td>990</td>
</tr>
<tr>
<td>23/8/1977 6:00</td>
<td>50</td>
<td>28.6</td>
<td>125.4</td>
<td>985</td>
</tr>
<tr>
<td>23/8/1977 12:00</td>
<td>60</td>
<td>31.0</td>
<td>127.0</td>
<td>985</td>
</tr>
<tr>
<td>23/8/1977 18:00</td>
<td>60</td>
<td>31.8</td>
<td>128.2</td>
<td>985</td>
</tr>
<tr>
<td>24/8/1977 0:00</td>
<td>50</td>
<td>32.3</td>
<td>129.3</td>
<td>985</td>
</tr>
<tr>
<td>24/8/1977 6:00</td>
<td>50</td>
<td>32.1</td>
<td>131.0</td>
<td>985</td>
</tr>
<tr>
<td>24/8/1977 12:00</td>
<td>40</td>
<td>32.6</td>
<td>132.6</td>
<td>992</td>
</tr>
</tbody>
</table>

Figure 3. Raw data before normalization (with “Date Time” as index).
The dataset is organized in batches, where each batch is occupied by one single typhoon event only. The model training requires two set of input data: the first acts as the input data and the second acts as the target data of prediction. Since the model is proposed to predict the future 6 timestamps, the last 6 rows in a batch will be selected as the target data and the latest 15 rows above the target data will be selected as the input data.
Model Foundation and Training

Model: "sequential"

<table>
<thead>
<tr>
<th>Layer</th>
<th>Output Shape</th>
<th>Param #</th>
</tr>
</thead>
<tbody>
<tr>
<td>masking</td>
<td>(None, 15, 4)</td>
<td>0</td>
</tr>
<tr>
<td>lstm</td>
<td>(None, 15, 32)</td>
<td>4756</td>
</tr>
<tr>
<td>lstm_1</td>
<td>(None, 16)</td>
<td>3156</td>
</tr>
<tr>
<td>dense</td>
<td>(None, 6)</td>
<td>102</td>
</tr>
</tbody>
</table>

Total params: 7,974  
Trainable params: 7,974  
Non-trainable params: 0

Figure 5. Summary of a LSTM model.

Figure 5 shows the basic architecture (layers) of a LSTM model developed in the project.

The first layer is a masking layer that is used to deal with row with missing data. A timestamp row with missing data will be padded with zeros in each field. This will inform the masking layer to ignore such rows in training so as to reduce the effect of missing data.

The following are 2 LSTM models those are used to deal with the LSTM operations. The last layer is a Dense (fully connected layer) that is responsible for outputting the prediction.

After pre-processing data. The source data are processed into batches of typhoon events. The batches are further developed into 2 groups: the first group is for training and the second group is for validation and testing.

After iterations of model training, the model will perform test prediction to one batch from the training data as a demonstration.
Figure 6 is a demonstration of the performance of a trained Latitude model. In the graph the first 2 time-steps are zero-padded because of missing data. The red and blue points at the last 6 time-steps are the prediction and true future data (latitude of central position) respectively.

The y-axis is latitude with degree as unit and the x-axis is timestamp.

The prediction and true sample data are plotted together along the same time series such that they can be compared directly to evaluate the performance of the trained model. This demonstration shows how Matplotlib library are applied for model evaluation and testing, and how prediction results are delivered in real applications (the “True Future” should be absent in real-time forecasting).
5. Limitations

During development of the predictor model, two major shortcomings are currently investigated. The following briefly discuss the problems found in designing the training strategy for LSTM typhoon track predictor models.

Small Batch Size
There are about 20 to 30 typhoon events occurring in one year and the life of each typhoon varies. For the purpose of favoring international cooperation, the time interval set in each best track must be fixed and uniform so it is usually 6 hours between 2 neighboring timestamps. In best track datasets there are at most only 4 tracks available in the forecast of a day.

Some typhoons only shortly sustained so that their track data might not provide enough number of rows to support an accurate prediction. In the above demonstration (training with RSMC best track data), only 960 typhoon events were recorded during 1977 to 2019. And not all of them get enough timestamps and available feature values to meet the requirement of a LSTM model.

The numbers of timestamps to input and predict in a model must be carefully considered and tested to make sure that the model get sufficient training data and it is useful in real-time application.

Selection of Typhoon Features
There are many potential features or information of a typhoon to be selected in a predictor model. When selecting a feature, it should make sure that a feature is available in both best track data archives and information center of real-time forecasting. The feature must be reasonably relevant to the motion of a typhoon. The input data in model training and application should be issued by the same meteorological organization to maintain source consistency.

However in most of the best track data archives not every row is fulfilled. It may represent the real-life condition that the measurement of some sort of data might not be available as expected. The availability of data is then another factor for considering whether a feature is suitable to be included in a predictor model.
6. Future Plan

<table>
<thead>
<tr>
<th>Period</th>
<th>Deliverables of Phase 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>February - April 2020</td>
<td>• refinement on trained model</td>
</tr>
<tr>
<td></td>
<td>• Model test and evaluation</td>
</tr>
<tr>
<td></td>
<td>• Conduct instructions or simple UI to apply the model</td>
</tr>
<tr>
<td></td>
<td>• Preparation of final report</td>
</tr>
<tr>
<td>19 April 2020</td>
<td>Deliverables of Phase 3 (Construction)</td>
</tr>
<tr>
<td></td>
<td>• Finalized tested implementation</td>
</tr>
<tr>
<td></td>
<td>• Final report</td>
</tr>
<tr>
<td>20-24 April 2020</td>
<td>Final presentation</td>
</tr>
<tr>
<td>5 May 2020</td>
<td>Project exhibition</td>
</tr>
</tbody>
</table>

Figure 6. Schedule of Final Year Project.

Figure 6 shows the schedule of the project which is based on the time table provided by the department of Computer Science. The period between February and mid-April 2020 is the final phase of finishing the product and the dates after the period is the time for delivering the final product (presentation, project submission, poster exhibition).

In the final phase of project development, effort will be focused on completing the predictor model. The selection of typhoon features and data sources will be further researched and tested. The setting of number of time-steps required by a LSTM model should also be looked into to make sure that model training is feasible (input data are sufficient) and its application is reasonable (taking reasonable number of input time-steps and output prediction in a number of time-steps that is useful).
7. **Reference**


5. Keras. Available: [https://keras.io/](https://keras.io/)


