Predicting the Trajectory of Tropical Cyclones and Storms using Recurrent Neural Networks and other Emerging Learning Machines

Mahamady Ouedraogo (mahamady.ouedraogo@aims.ac.rw)
African Institute for Mathematical Sciences (AIMS) Rwanda

Supervised by: Professor Ernest Fokoué
School of Mathematical Sciences, Rochester Institute of Technology Rochester, New York, USA

May 2020

Submitted in partial fulfilment of the requirements of a Master of Science in Mathematical Sciences at AIMS Rwanda

African Institute for Mathematical Sciences
RWANDA

Abstract

Tropical cyclones characterized by a strong wind with average speed between 118 and 165 km/h, they originate above tropical and subtropical waters. These cyclones constitute a permanent danger for human and his environment. Accurate forecasting of their trajectories is essential to reduce economic losses and save human lives. Given the complexity and non-linearity of meteorological system, a Recurrent Neural Network (RNN) which is one of machine learning techniques, it could be beneficial for modeling these cyclones behavior. In this essay we will propose the application of a fully connected Recurrent Neural Networks (Long Short Term Memory (LSTM)) to predict the trajectory of Tropical Cyclones in Madagascar, located in South East Africa . We were employed RNN over a grid system to reduce typical truncation errors. RNN and grid system are known to handle non-linearity and complex problem like our current study.

Using information about wind speed and pressure at points of given latitude and longitude, from the Australian agency (Bureau of Meteorology) and the National Hurrican Center (NHC) of the US , we will try to predict the path of a given cyclone at 6-hour intervals using keras package of python3. The results show that this proposed technique is competitive in accuracy compared to current methods which have been used since longtime in weather forecasting.

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

Mahamady Mohagenito OUEDRAOGO, May 2020

Contents

Abstract **i**

1 Introduction **1**
1.1 Literature Review 2

2 Tropical Cyclone **4**
2.1 Definition 4
2.2 Tropical Cyclone Structure 4
2.3 Formation of Tropical Cyclones 5
2.4 Dissipation of Tropical Cyclone 6
2.5 Effects 7

3 Methodology **8**
3.1 Recurrent Neural Networks 8
3.2 Model and Implementation 15

4 Results & Interpretation **18**
4.1 Presentation of Data 18
4.2 Descriptive Analysis 20
4.3 Prediction of Trajectories 24

5 Conclusion **26**

References **30**

1. Introduction

Even in the modern world we live in, much of the life on this planet, including our ability to grow food and even our basic safety, is affected by the weather. Therefore, to improve their chances of survival and their well-being, human beings have always tried to predict the weather, and in particular of course to predict the extremes of the weather conditions and events. One kind of such extreme events are tropical cyclones, which are a strong storm characterized by high speeds and very violent winds, in part by swirling headwinds. These storms wreak havoc on both land and sea, they very often cause economic damage and loss of life. In 2017, cyclone Enawo caused in Madagascar, 81 losses of life, approximatively 40,520 Damaged Houses, 1,8000 Classrooms totaly destroyed and 16 Health Centres destroyed (Le Bellec, 2018). In Mozambique, more than 200 persons lost their lives and 1,400 persons have been injured. At least 17,000 houses have been destroyed, displacing thousands of families. In Zimbabwe, more than 344 deaths and hundreds of missing people have been reported (Webots, a) due to destructive winds, torrential rainfall, and powerful storm surges. To minimise the damage it is essential to be able to predict the paths of tropical cyclones as accurately as possible.

Over the years, a large amount of work has been done in predicting of trajectories of this natural event, but most of the current methods for predicting tropical cyclones tracks are statistical in nature. These methods are limited due to the complexity and non-linearity of atmospheric systems. However, recently, with the ever increasing power of computers and technological advances, such as machine learning and artificial intelligence, new approaches have been applied the most recent being recurrent neural networks.

Recurrent neural networks are part of the family of artificial neural networks that have recurrent connections. Artificial neural networks are systems made up of units called neurons, connections and interacting in a non-linear way. A neural network is recurrent if there is at least one cycle in the structure. These recurrent neural networks, capable of efficiently modeling complex non-linear temporal relationships, could play a very important role in increasing the accuracy of future predictions of tropical cyclones tracks.

The choice of this topic is justified on the one hand by the fact that I have a passion for new technology, in particular machine learning and artificial intelligence; on the other hand, due to the fact that there has been no study carried out on this subject in the last few years namely the prediction of tropical cyclones in Africa using recurrent neural networks.

This essay endeavours to demonstrate the usefulness of the techniques of machine learning and artificial intelligence in the field of climate research, and in particular in the modelling of complex non-linear temporal relationships, by using recurrent neural networks to predict the trajectories of tropical cyclones and other emergent learning machines.

The essay has three main parts: in the first part we will describe the characteristics of tropical cyclones in general, their structure and their movement, then we will introduce the concept of fully recurrent neural networks on a grid model and other emergent learning machines, and finally we will apply these concepts to the particular case of predicting the paths of cyclones.

1.1 Literature Review

Despite the difficulties in weather forecasting, scientists are still making progress in research. In the case of the tropical cyclone forecasting, for example where scientists are working to find ways to improve the predictive capacity of models in order to understand in advance the movements of the tropical cyclone for the well-being of individuals.

Given the damage that the tropical cyclone can cause, it is necessary to understand these phenomena. There are a multitude of models for predicting the path of the tropical cyclone. These models differ from one another in their predictive power and the parameters that can be taken into account by each given model. The National Hurricane Center (NHC) of the National Oceanic and Atmospheric Administration (NOAA) based in the United States use four types of models in their path predictions: dynamic, statistical, statistical-dynamic and ensemble or consensus models.

Dynamic models are complex, they allow interactions between variables over time and they can be linear or nonlinear. These models are derived from data and use computational power. The models use a multiply nested movable mesh system to depict the interior structure of tropical cyclones.

Statistical models are simple models used to describe a given phenomenon, they use only statistical formulas to find out the behavior patterns of storms from historical data. These models are much simpler than dynamical models. The relationship used to predict cyclone paths are based on many characteristics specific to the storms collected, such as the location and date of the tropical cyclone. Basically, because of their simplicity, statistical models are used as a reference to establish the performance of the most complex models. Dynamic systems with statistical relationships both of them combined allow models to use large-scale variables as a set of predictors for tropical cyclone forecasting schemes.

In Wang et al. (2009) a dynamical-statistical model is developed for predicting Atlantic seasonal hurricane activity. So the model is built based on the empirical relationship between the observed interannual variability of hurricanes and also the variability of sea surface temperatures and vertical wind shear in 26 years (1981-2006) hindcasts from the National Centers for Environmental Prediction (NCEP) Climate Forecast System.

Basically, the lack of observations introduces uncertainty about the true initial state of the atmosphere. To account for this uncertainty, ensemble forecasting is proposed. By definition ensemble or consensus models are a combination of forecasts from given different models and different physical parameters, or varying model initial conditions. These models, with their abilities to identify situational uncertainty and provide probabilistic forecasting information, have an important role to play in cyclone forecasting.

With all these prediction models developed, there are still challenges to be met for nonlinear modeling of spatio-temporal systems. Nowadays with new technology we are witnessing new emerging techniques. For data mining techniques are also used in hurricane path prediction and time series analysis (Lee and Liu, 2000). In Moradi Kordmahalleh et al. (2016), the auther explain that, despite the availability of data and advanced prediction techniques, there is still a need for more advanced techniques with more accuracy in prediction. The neural network therefore appears to be particular and suitable for predicting the trajectory of a cyclone with complex systems and unknown dynamics. Dynamic Time Warping (DTW) which is used to make all the cyclones uniform, can allow Recurrent Neural Network to equally learn from each cyclone. However, the downside with the use of DTW is that it does not allow cyclones for whose path is not monotonous. What we see with the complexity of atmospheric systems is precisely the stochastic nature of the trajectory of the hurricane. Recurrent Neural Networks is one method which could benefit from learning from all cyclone paths.

Faced with the limitations of previous approaches a recurrent neural network, with its capacity for modeling any complex non-linear temporal behaviour of a hurricane, could likely increase the accuracy of forecasting future hurricane tracks as desired. For this reason, we have chosen to predict the paths of hurricanes and storms using recurrent neural networks and other emergent learning machines.

First of all, we are going to describe in the following section what a tropical cyclone is, How it is formed, the three main parts which compose it namely Eye, Eyewall and Rainbands and the effects associated to these cyclones.

2. Tropical Cyclone

2.1 Definition

Tropical cyclone, also known as a hurricane or typhoon, is an intense, circular-shaped storm that usually begins over warm tropical oceans and is characterized by low atmospheric pressure and strong winds accompanied by heavy rain. It draws its energy from the sea surface and retains its shape as long as it is above the warm water. Basically, a tropical cyclone very often brings winds that exceed at least 17m/s^{-1}. These strong winds are accompanied by torrential rains and a very devastating phenomenon known as storm surge which is dangerous to life. Sea surface elevation can reach about 6 metres above normal levels. This combination of water and wind causes cyclones that are very dangerous for coastal areas in tropical and subtropical regions of the world. At the end of summer, precisely from July to September in the northern hemisphere and from January to March in the southern hemisphere, cyclones hit remote areas such as the Gulf Coast of North America, northwestern Australia, eastern India and Bangladesh in Asia.

2.2 Tropical Cyclone Structure

It should be noted that different countries have different names for tropical cyclones. Tropical cyclone is specific in India or Australia, in the North Atlantic Ocean and the North East Pacific they are called hurricanes and also in the North West Pacific around the Philippines, in Asia, they are called typhoons. These storms are about 320 kilometres in diameter with localized swirling winds around a central region of low atmospheric pressure. The winds are usually due to the low pressure core and the rotation of the earth. These factors challenge the trajectory of the wind through Coriolis, which is a natural phenomenon. The direction of rotation of the cyclone differs according to the hemisphere; in the north hemisphere the direction is counterclockwise i.e. cyclonic and it is clockwise or anticyclonic in the southern hemisphere.

Basically, there are three main parts which constitute tropical cyclone namely: rainbands, eye, and eyewall; see below the description of each part.

2.2.1 The Eye.

The eye of cyclone is located in the center, this region is mostly calm and it is surrounded by another part called eyewall. Basically the lowest pressure of tropical cyclone occurs in the eye, comparatively this pressure is approximatively 15 percent lower than outside the storm. The eye has circular form which is ranged between 30 and 65 kilometers in diameter with warm temperature.

2.2.2 The Eyewall.

The Eyewall is the part that surrounds eye, it is the most dangerous part of tropical cyclone. This part is characterized by strong winds, heaviest rainfall and the deepest convective clouds which rise to the surface of the Earth with a height of about 15,000 meters. The strong winds are caused by changes in atmospheric pressure near the eye and create the large pressure gradient force.

2.2.3 The Rainbands.

This part is curved bands of clouds and thunderstorms and far from eye with a spiral shape. These bands are sometimes responsible for heavy rain, strong winds and tornadoes. Most of the heavy rain occurs near the center of the storm, and outside through the rainbands. See below the representative image of the structure of the tropical cyclone

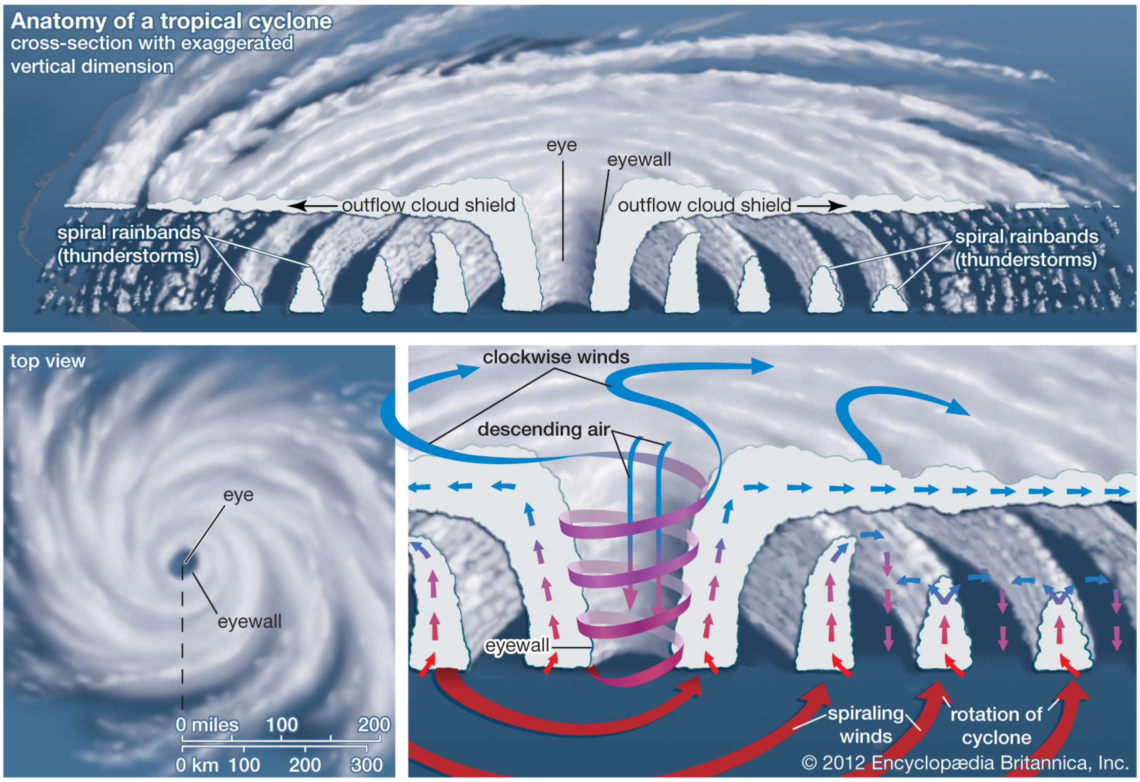


Figure 1: Anatomy of Tropical Cyclone(Webots, f)

2.3 Formation of Tropical Cyclones

Tropical cyclones form in the tropics between 5° to 30° and south over warm ocean, the ocean temperature can be approximately 27°C. The ocean provides source of moisture for evaporation, this moisture helps the cyclone to evaporate and to form cumulonimbus cloud afterwards; during the condensation process, latent heat is released, this latent heat helps to sustain the tropical cyclone. In addition there will be a force called coriolis force which is necessary for the deflection of wind into the cyclones, notice that coriolis force is zero at the equator so there is no possibility for tropical cyclones to form. In total there are four stages for the formation of a tropical cyclone.

• First stage namely formative stage

This stage is characterized by:

– Strong updraughts of warm, moist air

- Low pressure intensifies
- Pressure gradient steepens
- Wind deflects and spirals into the low pressure
- Gale force winds
- Cumulus cloud developing

• Second stage also called immature stage

At this step we notice:

- Bands of cumulonimbus cloud develop further
- Pressure drops below 1000 hpa
- Heavy rain
- Cyclone strength winds approximativey 50 kilometers from eye
- No clear eye

• Third stage or mature stage

This stage as its name suggests is characterized among other things by:

- Wind speeds reach tropical cyclone strength
- Cyclone strength winds up to 300 kilometers from eye
- Torrential and widespread rain
- Clear eye has formed
- Pressure has dropped below 950 hpa

• Fourth stage also called decaying stage

- Pressure starts to rise
- Surface friction from land and the lack of moisture cause cyclone to dissipate
- Heavy rains continue but start to clear up
- Winds decrease in speed

2.4 Dissipation of Tropical Cyclone

Cyclones dissipate when they can no longer extract enough energy from the warm ocean water. They can contribute to their own disappearance by moving to the deeper and cooler ocean waters. In addition, a storm that moves over land will suddenly lose its fuel source and quickly lose intensity. A cyclone that moves to very high elevations will change its structure and become extra-tropical to the extent that it encounters cold water. This transformation is marked by an increase in the diameter of the storm and the form change from circular to V-shaped or comma as the rain bands reorganize. An extratropical cyclone is generally characterized by higher central pressure and consequently lower wind speeds. These cyclones are fuelled by a variation in temperature from north to south, they weaken and disappear within a few days.

2.5 Effects

Having seen the various processes involved in the formation of the hurricane, it should be noted that this is not without consequences; these consequences include horizontal wind, tornadoes, gusts and eddies, storm surge and rainfall. Indeed, every year storms and cyclones affect dozens of countries around the world. The loss of life and damage to property is significant because of high winds, heavy rain, heavy swells and storm surges. This damage does not only affect localized areas on islands or coastlines, these cyclones also cause inland damage through flooding and landslides sometimes even hundreds of kilometres from the ocean. In the last 50 years, nearly one million tropical cyclones have been recorded, this is justified by the increase in population in exposed areas, due to the attraction of the sun in rich countries and population growth in other countries.

Once we have an idea of what a tropical cyclone is and the damage it can cause to humans, we are going to look at the methodology we will adopt to be able to predict the trajectory of these cyclones in order to somehow minimize the damage. In the next section we will talk about artificial neural networks, specifically recurrent neural networks. We will see how the algorithm behind these recurrent neural networks works, but before that we will see in figure 2 below the particularity of these recurrent neural networks compared to normal neural networks.

3. Methodology

In this chapter, we will talk about statistical perspective on machine learning. Given a data:

$$\mathscr{D}_n = \{(X_i, Y_i) \overset{iid}{\sim} \mathscr{P}_{xy}(x, y),\ X_i \in \mathscr{X}, Y_i \in \mathscr{Y},\ i = 1, \cdots, n\}$$

statistical learning is framed as the learning of a mapping function $f(\bullet)$ of an input space $\mathscr{X}$ associated to the output space $\mathscr{Y}$. The main purpose of this learning system is to learn a generalized mapping between an input and an output for given $\mathscr{D}_n$ such that we can be able to predict for new instances from $\mathscr{D}_n$ where the output variable $\mathscr{Y}$ is unknown, this is the main idea behind feedforward neural network, Recurrent Neural Networks and so on. Basically, the methods used in this essay fall in that class of mathematical objects.

3.1 Recurrent Neural Networks

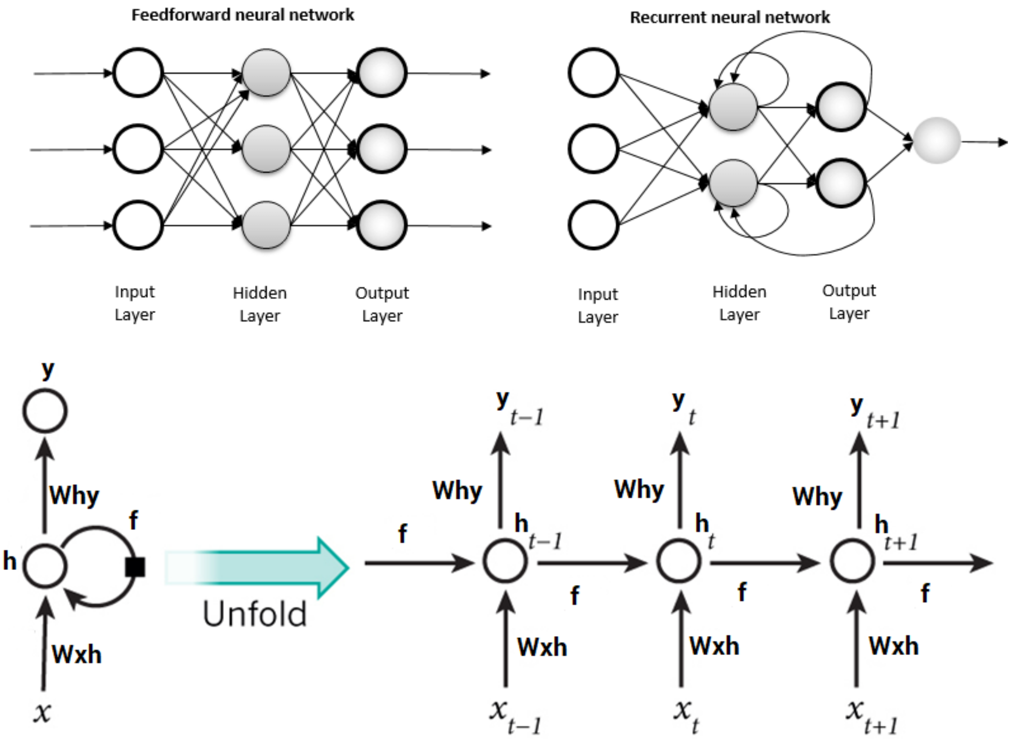


Figure 2: Recurrent Neural Network Architecture (Webots, e)

Recurrent Neural Networks (RNNs) are special type of Artificial Neural Networks (ANN) that feature connections between hidden layers that are propagated through time in order to learn sequences. A Sequence is a stream of data, it could be finite or infinite, which are interdependent like times series data, or a conversation (words). Basically, Recurrent Neural Network are used in times series prediction for example prediction of the stocks over the time, speech recognition, speech synthesis, robot control, and machine translation like Natural Language Processing. The term recurrent is because they perform in the same way for every element in the sequence. RNNs are known as nonlinear dynamical models, they have in a sense some memory about what happenend earlier in the sequence of data that's why

RNNs can handle historical data. RNNs have emerged from the limits of the Feed Forward Network (no future scope, only based on current input, difficulty in remembering the past state...)

Basically, RNN is one of the variants of the Kalman filter. We can define the Kalman filter as an iterative mathematical algorithm that uses over time a series of observed measurements that contain noises and other inaccuracies, and produces estimates of unknown variables more accurately than those based on a single measurement, by estimating what we call a joint probability distribution on the variables for each timeframe.

The filter means simply that the best estimator is obtained by means of processing observation data containing errors. In other terms to obtain somehow some unknown parameters, some observations must be recordered.The observation value is dependent on certain parameters and the observation value contains noises. The Kalman filter is somehow a recursive filtering method. It uses the previous estimated value or the recent observation value to estimate the current value.

The Kalman filter is used in a wide range of technological fields like radar, electronic vision and communication. For example in the case of radars where one wishes to follow a target, data on its position, speed and acceleration are measured at all times but with some of disturbance due to noise or measurement errors. Kalman filter uses the dynamics of the target which defines its evolution over time to obtain good data, by eliminating the effect of noise. This data can be calculated for the present moment (filtering), in the past (smoothing), or on a future horizon (prediction).

The Kalman Filter is used in other fields which are different from electronics, for example in meteorology and oceanography, for data assimilation in digital model, in finance or in navigation and it is even used in the estimation of road traffic conditions in the case of control by access ramp where the insufficient number of magnetic loops on the road.

The idea about RNN is that,the output of one instance is taken as input for the next instance for the same neuron. The way the data is kept in memory and flows at different time periods makes RNNs powerful and successful in modeling sequential or historical data. The current state is given by:

$$h_t = f(h_{t-1}, x_t) \tag{3.1.1}$$

where x_t is the input state, h_{t-1} the previous state, h_t the current state and $f(\bullet, \bullet)$ a given function. The function $f(\bullet, \bullet)$ is a mathematical function, which creates the mapping between the output and the hidden states. The hidden state performs the processing, it is very important and the magic to convert has the input to the desired output based on three necessary things: weights, bias and activation functions.

Weight is same as the slope in linear regression, it is a numerical parameter which determines how strongly each of the neurons affects the other. The weight can be responsable of gradient descent problem, so while training a RNN, the slope can be either too small or very large and this makes training very difficult. When the slope is to small, the problem is known as vanishing gradient; when the slope is too large the problem is called exploding gradient.

The bias is like the intercep in linear equation, it is used to adjust the output and the weighted sum of the inputs to the neuron.

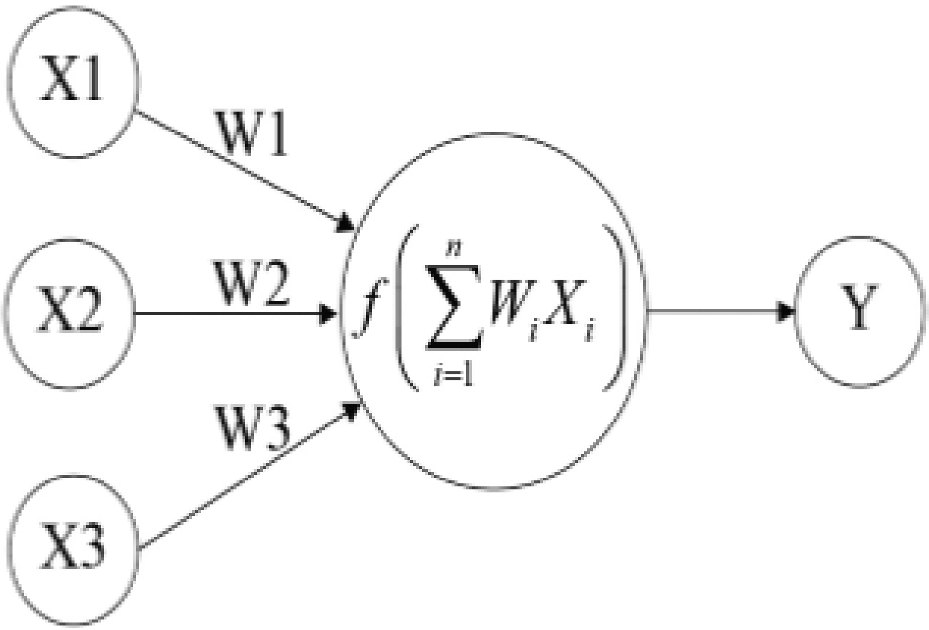


Figure 3: Mapping between an input and an output

So, the hidden state vectors can be defined by:

$$h_t = f(\boldsymbol{w_{hh}} \cdot h_{t-1} + \boldsymbol{w_{xh}} \cdot x_t) \tag{3.1.2}$$

Where $\boldsymbol{w_{hh}}$ is a $n_h \times n_h$ weight matrix at recurrent neuron, $\boldsymbol{w_{xh}}$ is $n_h \times n_x$ weight matrix at the input neuron and f the activation function for the hidden state (Karpathy et al., 2015).

The output at time t, $\boldsymbol{Y}_t$ is given by:

$$\boldsymbol{Y}_t = f(\boldsymbol{w_{yh}} \cdot h_t) \tag{3.1.3}$$

$\boldsymbol{w_{yh}}$ is $n_h \times n_y$ weight at the output layer. Basically, the main key which gives to RNNs models the ability to handle the nonlinear or complex problem in the nature is the activation function. It can be Rectified Linear Unit function (ReLU), hyperbolic tangent function (tanh) or sigmoid function.

The error for time step t is given by the following equation:

$$Error = (\hat{\boldsymbol{Y}}_t - \boldsymbol{Y}_t) \tag{3.1.4}$$

where $\hat{\boldsymbol{Y}}_t$ is the predicted value and $\boldsymbol{Y}_t$ the actual value.

The overall loss which is the summation of time step specific losses found in $[t, T]$ is given by:

$$\mathcal{L}(\hat{\boldsymbol{Y}}, \boldsymbol{Y}) = \sum_{t=1}^{T} \mathcal{L}(\hat{\boldsymbol{Y}}_t - \boldsymbol{Y}_t) \tag{3.1.5}$$

The extended Kalman filter can be viewed as a nonlinear version of the basic Kalman filter that linearize the models about a current estimate. The Extended Kalman Filter started to appear in the neural network training like RNN due to the increasing power of computer system development.

In the extended Kalman filter, the state transition also called process model and observation model or data model are given by the equations below:

$$\boldsymbol{X}_t = f(\boldsymbol{X}_{t-1}, \boldsymbol{u}_t) + \boldsymbol{w}_t \tag{3.1.6}$$

$$\boldsymbol{Z}_t = h(\boldsymbol{X}_t) + \boldsymbol{v}_t \tag{3.1.7}$$

where:

• $\boldsymbol{X}_t$ is the state transition model

• $\boldsymbol{Z}_t$ is observation model

• t denote time step

• $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are the process and observation noises which are both assumed to be zero mean multivariate Gaussian noises.

• $\boldsymbol{u}_t$ is the control vector or process input.

The function $f(\bullet, \bullet)$ can be used to compute the predicted state from the previous estimate and the function $h(\bullet)$ can be used to compute the predicted measurement from the predicted state.

RNN can be describe mathematically by the following random equations based on the Extended Kalman filter state transition and observation models:

$$\boldsymbol{X}_t = \boldsymbol{X}_{t-1} + \boldsymbol{\delta}_{t-1} \tag{3.1.8}$$

$$\boldsymbol{Y}_t = f(\boldsymbol{X}_t, \boldsymbol{U}_t, \boldsymbol{V}_{t-1}) + \mathcal{E}_t \tag{3.1.9}$$

Where:

• $\boldsymbol{X}_t$ expresse the state of Neural Network at time t as a stationary proccess which is corrupted by the noise $\boldsymbol{\delta}_{t-1}$; in other the state $\boldsymbol{X}_t$ of Neural Network consist of network weights

• $\mathcal{E}_t$ is the error associated to the output

• $\boldsymbol{Y}_t$ expresse the desired output of the network as nonlinear function of given input vector

• $\boldsymbol{U}_t$ is the input vector of the weight vector $\boldsymbol{X}_t$

• $\boldsymbol{V}_{t-1}$ is the recurrent neuron from previous step.

Basically the idea of Extended Kalman Filter lies in linearization of data model at each step around the last state estimate $\hat{\boldsymbol{X}}_t$. The training of Neural Network can be seen as minimization problem i.e find the state estimate $\boldsymbol{X}_t$ which minimizes least square error based on all the previous data model. The equations are obtained by means of the maximum posterior estimation or the minimum variance estimation. the solution of this problem can be expressed below.

$$\boldsymbol{K} = \boldsymbol{\Sigma_{i|i-1}} \boldsymbol{H}_i^T (\boldsymbol{H}_i \boldsymbol{\Sigma_{i|i-1}} \boldsymbol{H}_i^T + \boldsymbol{R}_i)^{-1} \tag{3.1.10}$$

Where:

- $\boldsymbol{K}$ is called Kalman gain matrix
- $\boldsymbol{i}$ denote the time
- $\boldsymbol{\Sigma_{i|i-1}}$ is error covariance matrix of the state
- $\boldsymbol{H}$ is a matrix which contains partial derivatives of $i's$ output respect to $N's$ weight.

$$\boldsymbol{H} = \begin{bmatrix} \frac{\partial Y_1}{\partial W_1} & \frac{\partial Y_1}{\partial W_2} & \cdots & \frac{\partial Y_1}{\partial W_N} \\ \frac{\partial Y_2}{\partial W_1} & \frac{\partial Y_2}{\partial W_2} & \cdots & \frac{\partial Y_2}{\partial W_N} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ \frac{\partial Y_i}{\partial W_1} & \frac{\partial Y_i}{\partial W_2} & \cdots & \frac{\partial Y_i}{\partial W_N} \end{bmatrix}$$

$$\hat{\boldsymbol{X}}_{t+1} = \hat{\boldsymbol{X}}_t + \boldsymbol{K}_i(\boldsymbol{y}(i) - f(\hat{\boldsymbol{X}}_t, \boldsymbol{U}_t, \boldsymbol{V}_{t-1})) \tag{3.1.11}$$

Where:

- $\hat{\boldsymbol{X}}_t$ is a vector of all the weights
- $f(\bullet, \bullet, \bullet)$ is a function returning a vector of actual outputs
- $\boldsymbol{y}$ is a vector of desired outputs,

$$\boldsymbol{\Sigma_{i|i}} = \boldsymbol{\Sigma_{i|i-1}} - \boldsymbol{K}_i \boldsymbol{\Sigma_{i|i-1}} \boldsymbol{H}_i + \boldsymbol{Q}_i \tag{3.1.12}$$

$$\boldsymbol{Q}_i = E[\boldsymbol{e_i} \boldsymbol{e_i}^T] \tag{3.1.13}$$

Where:

- $\boldsymbol{Q}_i$ is zero mean multivariate Gaussian noises associated to the covariance matrix of the state
- $\boldsymbol{e_i}$ is the gaussian noise.

During the training process using RNN, the weights are updated using Extended Kalman Filter, so we have three steps:

- Forward pass: compute $\boldsymbol{Y}$ the output vector
- Compute the matrix $\boldsymbol{H}$
- Compute the new networks weight $\boldsymbol{W}_{i+1}$ and the covariance matrices $\boldsymbol{\Sigma_{i|i}}$

$$\boldsymbol{K} = \boldsymbol{\Sigma_{i|i}} \boldsymbol{H}_i^T (\eta \times \boldsymbol{I} \times (\boldsymbol{H}_i \boldsymbol{\Sigma_{i|i}} \boldsymbol{H}_i^T) + \boldsymbol{R}_i)^{-1} \tag{3.1.14}$$

$$\boldsymbol{R}_i = E[\boldsymbol{V}\boldsymbol{V}^T] \tag{3.1.15}$$

Where:

- $\boldsymbol{Ri}_i$ is zero mean multivariate Gaussian noises associated to the Kalman gain matrix
- $\boldsymbol{V}$ is the first derivative of activation function in RNN

$$\boldsymbol{\Sigma_{i|i+1}} = \boldsymbol{\Sigma_{i|i}} - \boldsymbol{K}\boldsymbol{\Sigma}_{i|i}\boldsymbol{H}^T + \boldsymbol{Q} \tag{3.1.16}$$

$$\boldsymbol{\hat{W}_{i+1}} = \boldsymbol{W_i} + \boldsymbol{e}_i \tag{3.1.17}$$

$$\boldsymbol{W_{i+1}} = \boldsymbol{\hat{W}_{i+1}} + \boldsymbol{K} \times \mathcal{E} \tag{3.1.18}$$

Where η is the learning rate value of RNN and $\boldsymbol{I}$ is identity matrix, the value of $\mathcal{E}$ is given by the difference of the output (prediction) system with the actual data.

Below the Kalman filter diagram schematizing the different steps of the algorithm.

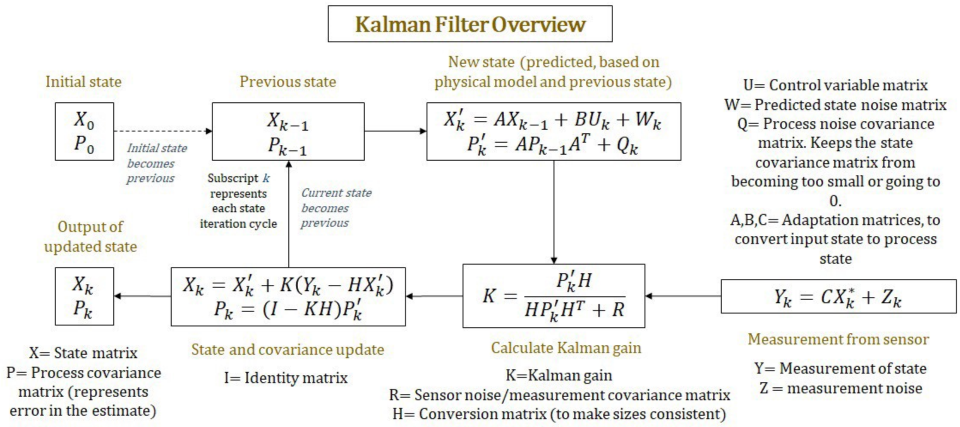


Figure 4: Processing flow of Kalman filter (Webots, d)

3.1.1 Gradient Descent.

Generally gradient descent problem is one of the issues encountered during the training of Neural Network. Below an idea about how the gradient can vanish or explode during the training of RNN.

Basically, training Recurrent Neural Network is done by accrossing time step using Backpropagation where the overall error gradient is equal to the summation of individual error gradient at each time step. The total error of time step T can be computed by:

$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial W} \tag{3.1.19}$$

So applying chain rule to compute the overall error gradient our equation becomes:

$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial Y_t} \cdot \frac{\partial Y_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_k} \cdot \frac{\partial h_k}{\partial W} \tag{3.1.20}$$

By observation, $\frac{\partial h_t}{\partial h_k}$ is the derivative of the hidden state at time t with respect to the hidden state at time k which involves the product of jacobian $\frac{\partial h_i}{\partial h_{i-1}}$. We can rewrite $\frac{\partial h_t}{\partial h_k}$ as:

$$\frac{\partial h_t}{\partial h_k} = \frac{\partial h_t}{\partial h_{t-1}} \cdot \frac{\partial h_{t-1}}{\partial h_{t-2}} \cdots \frac{\partial h_{k+1}}{\partial h_k} = \prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}} \tag{3.1.21}$$

So based on hidden state vector equation (3.1.2), $\frac{\partial h_t}{\partial h_{t-1}}$ yields to $\boldsymbol{W}^T[f^{'}(h_{t-1})]$, hence by replacing in (3.1.21) we get:

$$\frac{\partial h_t}{\partial h_k} = \prod_{i=k+1}^{t} \boldsymbol{W}^T diag[f^{'}(h_{t-1})] \tag{3.1.22}$$

By performing eigendecomposition on the Jacobian matrix $\boldsymbol{W}^T[f^{'}(h_{t-1})]$ we get the eigenvalues

$|\lambda_1| > |\lambda_2| > |\lambda_3| > \cdots > |\lambda_n|$ associated to eigenvectors $v_1, v_1, \cdots, v_n$.

So any change on the hidden state vector Δh_t in the direction of given eigenvectors v_i will affect the change with eigenvalue associated with eigenvector $\lambda_i \Delta h_t$.

Equation (3.1.21) implies that subsequent time steps will result in scaling the change with λ_i^t which is the i^{th} eigenvalue raised to the power of the current time step t.

Taking the sequence $\lambda_i^1 \Delta h_1, \lambda_i^2 \Delta h_2, \cdots, \lambda_i^t \Delta h_t$ we realize that λ_i^t will end up by dominating Δh_t because λ_i^t grows exponentially if $\lambda_i > 1$

$$\text{if} \begin{cases} \lambda_i > 1 & \text{Gradient explodes} \\ \lambda_i < 1 & \text{Gradient vanishes} \end{cases}$$

So consider:

$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\| \leq \left\| \boldsymbol{W}^T \right\| \cdot \left\| diag[f^{'}(h_{i-1})] \right\|$$

and $\Gamma \boldsymbol{W}$ the largest eigenvalue assosiated to $\left\| \boldsymbol{W}^T \right\|$, Γh associated to $\left\| diag[f^{'}(h_{i-1})] \right\|$. Notice that the value of Γh depends on the type of activation function which we used.

So we have this inequality

$$\left\| \frac{\partial h_i}{\partial h_{i-1}} \right\| \leq \left\| \boldsymbol{W}^T \right\| \cdot \left\| diag[f^{'}(h_{i-1})] \right\| \leq \Gamma \boldsymbol{W} \cdot \Gamma h$$

$$\left\| \frac{\partial h_t}{\partial h_k} \right\| = \left\| \prod_{i=k+1}^{t} \boldsymbol{W}^T diag[f^{'}(h_{t-1})] \right\| \leq (\Gamma \boldsymbol{W} \cdot \Gamma h)^{t-k}$$

So if the distance between t and k becomes large, the value of Γ and $\boldsymbol{W}$ will tell if it is vanishing gradient problem or exploding gradient.

$$\text{if} \begin{cases} \Gamma > 1 \quad \text{or} \quad \boldsymbol{W} > I & \text{The problem is known as exploding gradient} \\ \Gamma < 1 \quad \text{or} \quad \boldsymbol{W} < I & \text{The problem is known as vanishing gradient} \end{cases}$$

The issues of this gradient problem is that we can have:

- Poor performance of the model
- Bad accuracy
- Long time during the training

Below, some solutions depending on the gradient problem we have. If the problem is known as vanishing gradient we have to:

- Initialize the weight
- Choose the the right activation function
- Use Long Short Term Memory networks (LSTM)

The solutions to deal with exploding gradient problem are:

- Initializing the identity
- Truncated Backpropagation
- Gradient clipping

Fascinated by the temporary displacement of tropical cyclone we will apply the mathematical notions above to predict the trajectory of some cyclones of Madagascar.

3.2 Model and Implementation

To implement our model, we will use Python3 precisely keras. Keras is an Application Programming Interface (API) which integrates with basic level of deep learning languages such as TensorFlow. As we said we will employe RNN over a grid model, we need to fix a good value of hyperparameter because grid search is traditional technique for implementing hyperparameter and the value of this hyperparameter affects the training algorithm. So by default, keras provides 0.001 as the value of learning rate hyperparameter. The learning rate by definition is the rate at which the RNN updates the weights for hidden layer and is modified using Stochastic Gradient Descent.

Grid model is one of methods used in climate to handle nonlinear or complex problems, this method has proven itself in numerical weather prediction (NWP). In our case, a Recurrent neural network will be trained to learn more about the trajectories of cyclones base on grid locations. The idea is that the grid system can reduce the number of possible truncation errors and contribute to improve the accuracy of our model by allowing us to control the amount of loss used by prediction (Birchfield, 1960)

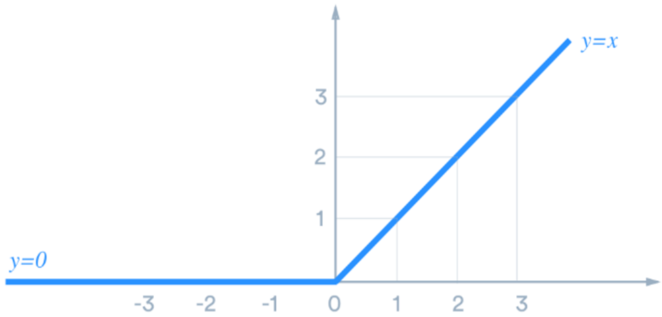
We will need also a reccurent dropout. It is just a regularization method where input and recurrent connections to LSTM units are excluded probabilistically from given activation and weight updates while training a network. This has the effect of avoiding overfitting and improving the performance of the model.

Sequence prediction problems like our case are challenging because the length of the input sequence can vary and neural networks in particular, are designed to work with fixed length inputs. Another challenge sequence data is temporal ordering of the observations, it can be a challenge to extract features suitable for use as input to supervised learning models. Recurrent neural networks, such as the Long Short-Term Memory, or LSTM, network are specifically designed to deal with sequences of input data. LSTM also called Long Short-Term Memory Cells is a special kind of RNN capable of learning long term dependencies remembering information for long periods of time by preventing vanishing or exploding weights throughout the RNN.

We will stack a Dense layer after the LSTM layer, because it's actually the layer where each neuron is fully connected to all neurons. In other, given neurons of the layer N will be connected to neurons of the layer $N+1$. Also, we will use three hidden layers and **ReLu** (Rectified Linear Unit) as activation function. The number of hidden layer in RNN contribute to the complexity of the model.

The **Rectified Linear Unit** function is defined by:

$$\text{f}(x) = \begin{cases} x & \text{when} \quad x \geq 0 \\ 0 & \text{when} \quad x < 0 \end{cases}$$



Using ReLU as activation function in training the model, the output is between [0, ∞). The choice of the activation function has an impact on the result, this function must be sufficiently robust, differentiable, simple and fast for processing, it should not be zero centered, also it should not cause gradient vanish. The sigmoid function, once widely used, has certain limitations: the computations are time consuming and complex, it causes gradients to vanish and no signals pass through the neurons at some point of time, it is slow in convergence and it is not zero centered (Giuseppe Ciaburro, September 2017).

Why ReLU in this essay because it is simpler and faster also it doesn't have vanishing gradient problem.

Before training our model, we normalized our data using this formula $\mathcal{Z} = \frac{\mathcal{X}-\mu}{\sigma}$ where $\mathcal{X}$ is the actual value, μ the mean and σ the standard deviation. The idea behind that normalization is to avoid gradient vanishing. Without normalize the data, the gradient tends to be very high. A normalized value represents somehow the probability that the value could appear in the given historical data, so therefore the RNN can modify the weight vectors at each state easily.

To summarize, we have 5 essential steps for the implementation of our prediction model.

• Define Network

We will construct our LSTM neural network with a one input, one output, 2 memory units in the LSTM hidden layer, and 2 neurons in the fully connected output layer with ReLU activation function.

• Compile Network

We will use rmsprop optimization which is a gradient-based optimization technique proposed by Geoffrey Hinton with default configuration and the mean squared error loss function. Rmsprop is considered to be an efficient way to solve the problem. This technique is based on the moving average of square gradients to normalize the gradient itself. This has the effect of balancing the step size - decrease the step for a large gradient to avoid exploding in case of high weight, and increase the step for a small gradient to avoid disappearing in case of small weight.

- **Fit Network**

We will fit the network for 100 epochs and use a batch size equal 50. We will also turn one all verbose output.

- **Evaluate Network**

We will evaluate our network based on the training dataset, also we will evaluate the model on a test or validation set.

- **Make Predictions**

We will make predictions for the training input data, so typically we will make predictions on test or validation dataset.

As they say, theory is super fuzzy very often in certain circumstances without practice, but that doesn't mean it's useless. To do this, we will see in the next section, the wonders after the application of our method, namely the results obtained after analysis.

4. Results & Interpretation

4.1 Presentation of Data

The data that we will use in our tropical cyclone track prediction study from two different sources, the Bureau of Meteorology (BoM) and the National Hurricane Center (NHC). The BoM was established in the 1906's under the Meteorological act. It is an executive agency of the Australian Government and its purpose is to provide meteorological services to Australia and the surrounding areas. From its inception to the present day, the Bureau of Meteorology has also brought together the weather services of the United States that existed previously. The formal transfer of responsibilities between The States in the field of meteorological recording took place on January 1, 1908, say 2 years after the creation of the Bureau of Meteorology. We were able to download the data via the link (Webots, c). The database contains the following variables: Longitude, Latitude, Pressure, Sequence, Name and Datetime; this database covers the period from 1986/1987 to 2016/2017.

The second source of our data is the National Hurricane Center (NHC), a division of the U.S. National Oceanic and Atmospheric Administration (NOAA). NHC's objective is to track and predict tropical weather systems between the Prime Meridian and the 140th West Meridian to 30 North Parallel in the Northeast Pacific Ocean and 31 North Parallel in the North Atlantic Ocean. This agency is co-located with the Miami branch of the National Weather Service and it is located on the campus of the Florida International University at Park University in Florida. We were able to access the data through this link (Webots, b) for the same period as before, the database contains the variables Pressure, Wind Speed, Date and Category. In this database we were only interested in the variables Wind Speed and Category because the other database we had downloaded contained the variables Date and Pressure. In total our database contains 9 variables and 761 observations of data recorded every 6 hours, the data was not clear due to that we did some preprocessing like removing some missing values and some variables using some packages of python (pandas, numpy also matplotlib).

These data concern the tropical cyclones of Madagascar located in South East Africa with current population estimated by 27,691,018 based on Worldometer elaboration of the latest United Nations data.

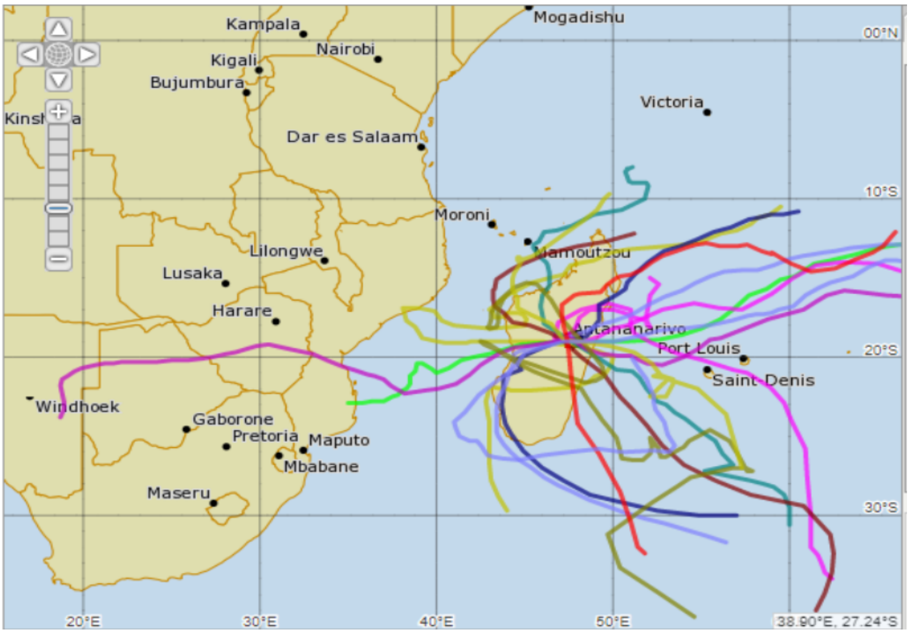


Figure 5: Map of different cyclones crossing Madagascar (Webots, c)

The main variables we utilized are longitude, latitude, pressure (in millibar) and wind speed (in knot where one knot equal to 1.15 mile per hour). So given the longitude and latitude, we derive the distance traveled and the direction of each point recodered six hourly by cyclone based on **Haversine formula**. These variables are useful because they provide a measure of relation also the reprentational sense to learn for paths prediction. The variables distance, direction, pressure, wind speed and grid point will be considered in our prediction. Below **Haversine formula**

The Haversine formula for the distance along a great circle between two points on a sphere is:

$$\boldsymbol{a} = (\sin(dlat/2))^2 + \cos(lat1) \times \cos(lat2) \times (\sin(dlon/2))^2 \tag{4.1.1}$$

$$\boldsymbol{c} = 2 \times atan2(\sqrt{a}, \sqrt{1-a}) \tag{4.1.2}$$

$$\boldsymbol{d} = R \times c \tag{4.1.3}$$

where:

- $dlon = lon2 - lon1$ and $dlat = lat2 - lat1$
- $lon1$ and $lon2$ are the longitudes of the two points
- $lat1$ and $lat2$ are the latitudes of the two points
- $atan2$ is arctangent function
- $\boldsymbol{R}$ is the radius of Earth (3956 miles)
- $\boldsymbol{c}$ is the great circle distance in radians
- $\boldsymbol{d}$ is the great circle distance in miles

Below an overview of initial data without any preprocessing.

```
df = pd.read_csv('Base.csv')
df.head()
```

	year	seq	name	datetime	lat	lon	pressure	winSpeed	Cat
0	1987	5	CALIDERA	1/11/1988 12:00	-11.0	59.6	1000	24.0	TD
1	1987	5	CALIDERA	1/11/1988 18:00	-11.0	58.9	1000	20.0	TD
2	1987	5	CALIDERA	1/12/1988 0:00	-11.2	58.2	1000	20.0	TD
3	1987	5	CALIDERA	1/12/1988 6:00	-11.3	57.7	997	25.0	TD
4	1987	5	CALIDERA	1/12/1988 12:00	-11.8	56.7	997	25.0	TD

After some preprocessing:

```
# Extact date, day, month, year, hour, from datetime
df['Date'] = pd.DatetimeIndex(df['datetime']).date
df['Day'] = pd.DatetimeIndex(df['datetime']).day
df['Month'] = pd.DatetimeIndex(df['datetime']).month
df['Year'] = pd.DatetimeIndex(df['datetime']).year
df['Hour'] = pd.DatetimeIndex(df['datetime']).hour
# delete columns
del df['year']
del df['seq']
del df['Cat']
df.head()
```

	name	lat	lon	pressure	winSpeed	Day	Month	Hour	Distance	Direction	gridID
0	CALIDERA	-11.0	59.6	1000	24.0	11	1	12	47.476776	180.231363	3.0
1	CALIDERA	-11.0	58.9	1000	20.0	11	1	18	49.431384	209.248548	3.0
2	CALIDERA	-11.2	58.2	1000	20.0	12	1	0	34.580192	315.900988	3.0
3	CALIDERA	-11.3	57.7	997	25.0	12	1	6	75.999715	315.961428	3.0
4	CALIDERA	-11.8	56.7	997	25.0	12	1	12	70.702642	311.259139	3.0

4.2 Descriptive Analysis

As the name suggests, in this subsection we have made a descriptive study of our different variables just to have a clearer idea on our database. Below some results obtained.

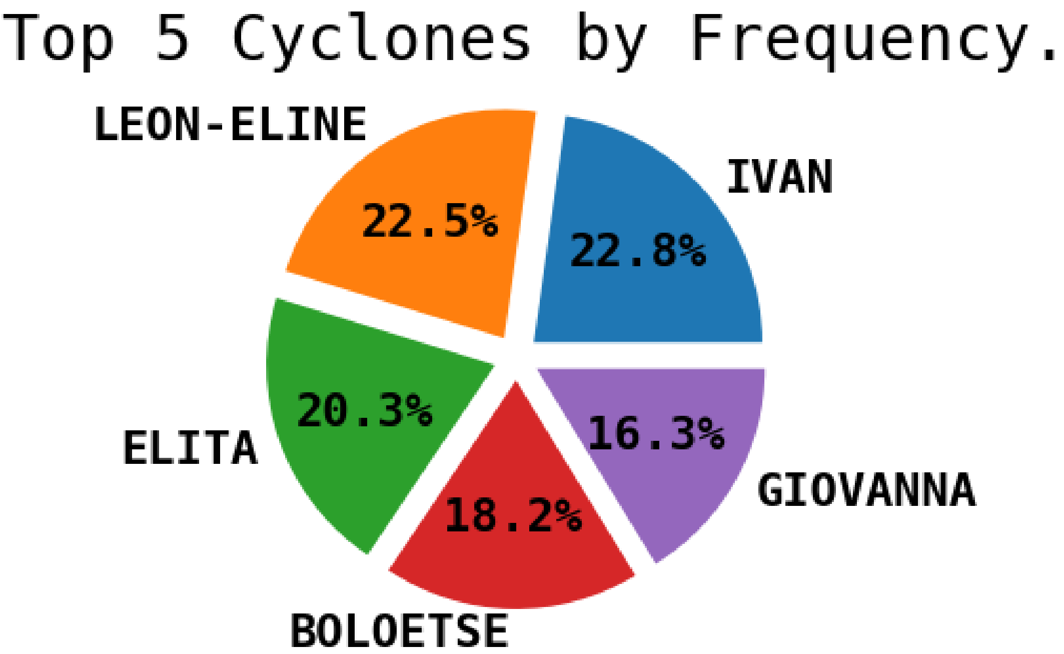
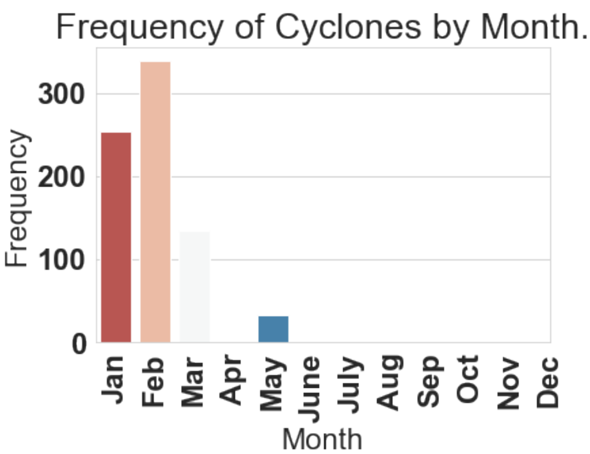


figure 6: Top five cyclone by frequency

Figure 6 shows that among the cyclones recorded in total 17 cyclones, cyclone IVAN is the most observed of all followed by cyclone LEON-ELINE, ELITA, BOLOETSE and GIOVANNA



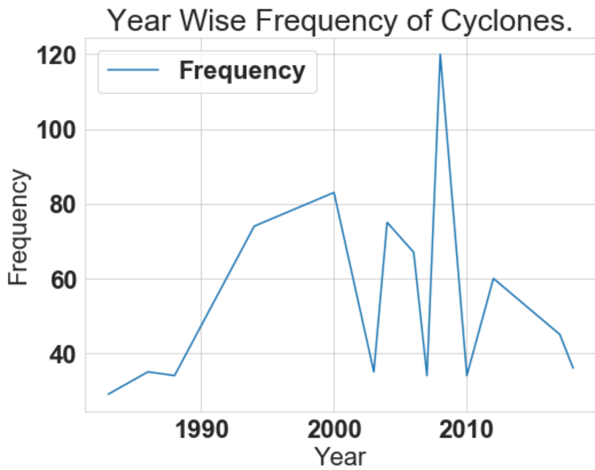


Figure 7: Frequency of cyclones recorded by month and by year

In terms of frequency per month figure 7 tells us that the month of February is the month when we recorded more cyclones in Madagascar over the period from 1987 to 2017. So, this gives us an idea about the period we should avoid visits in Madagascar, we can see on the figure 5 that this period is between January to March because of high risk of cyclones in the east and northeast. From September to December for example we can enjoy visits in Madagascar because that period we don't have high risk of cyclones.

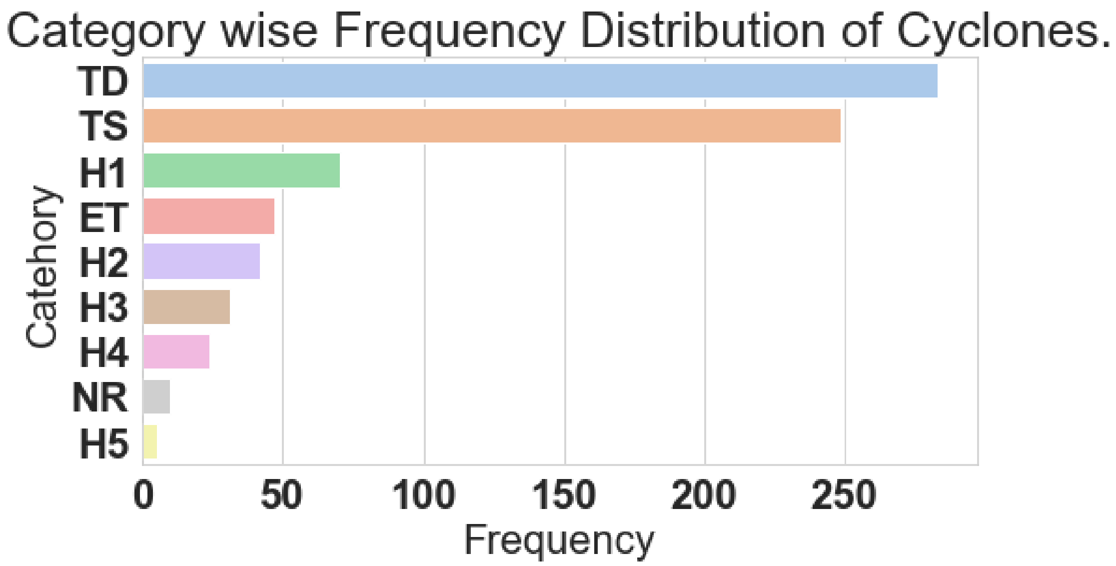


Figure 8: Frequency of cyclones recorded by category

Among the types of cyclones encountered in Madagascar, tropical depression (TD) is the most common of all. Tropical cyclone depression is a cyclone whose wind speed is less than 34 knots approximately, followed by the Tropical Storm (TS) whose wind speed is between 34 and 63 knots.

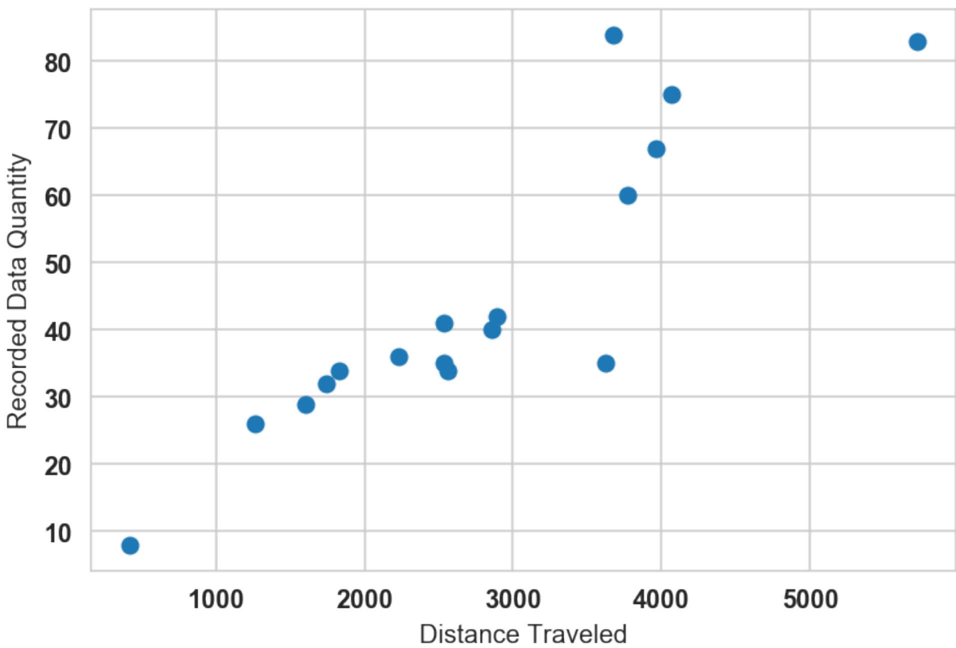


Figure 9: Distance Traveled vs Recordered Data

From figure 9, we notice that there is a correlation between the number of data points collected and the distance traveled. We can conclude that cyclones are continuously traveling therefore, direction and angle of travel continuously provide the recurrent neural network with information about its trajectory behavior. We can see also that cyclone LEON-ELINE (1999) traveled approximatively 5724.323 miles and cyclone FAMI (2009) traveled approximatively 416.825 miles.

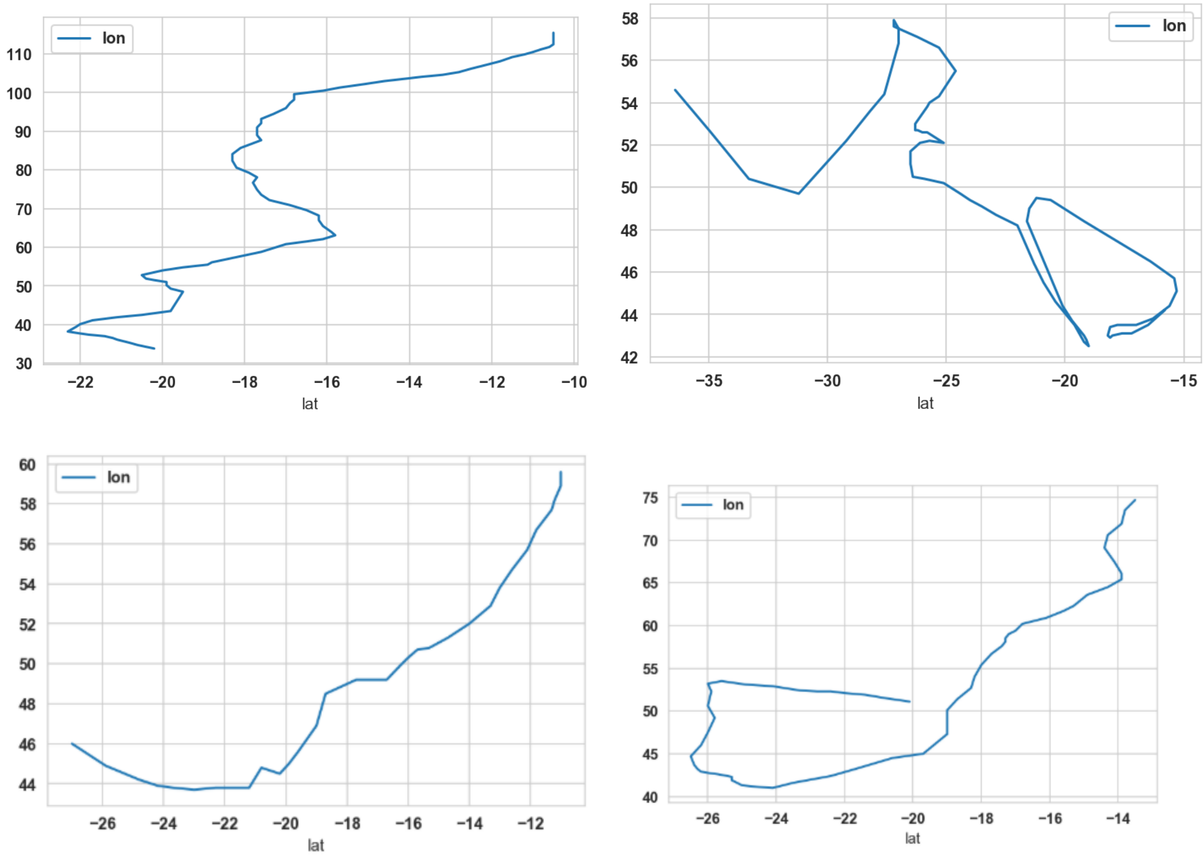


Figure 10: Trajectories of the four cyclones slected randomly base on latitude and longitude

Figure 10 shows the simulation of four cyclones selected randomly among seventeen based on longitude and latitude in Madagascar.

4.3 Prediction of Trajectories

After removing some outliers that contain too little or too much information to keep a more normal distribution, we obtain as valid sample 380 observations. So we divided this sample in two parts: 85% of the data were used for training the model and 15% for testing the accuracy of the model.

Each data tuple containing a sequence of features of size 5 comprised of wind speed, pressure, distance, direction, and grid identification number. As a result, validation of the training set was completed on 10% of the 85% training set. In total we have five layers such as one input layer, three hidden layers and one output layer. We consider a hundred iterations, the accuracy of our Recurrent Neural Network using ReLU as activation function employed over a grid system is shown for the trajectory predictions of four cyclones selected randomly in figure 10 below the piece of code that we used for bulding our model.

```
def build_model(layers):
    model = Sequential()

    for x in range(0,5):
        model.add(LSTM(input_dim=layers[0], output_dim=layers[1], return_sequences=True))
        model.add(Dropout(0.1))

    model.add(LSTM(layers[2], return_sequences=False))
    model.add(Dropout(0.1))

    model.add(Dense(output_dim=layers[2]))
    model.add(Activation("relu"))

    start = time.time()
    model.compile(loss="mse", optimizer="rmsprop",metrics=['accuracy'])
    print("Compilation Time : ", time.time() - start)
    return model
```

```
model = build_model([feature_count, seq_len, 1])
history=model.fit(X_train, y_train, batch_size=50, epochs=100, validation_split=0.1, verbose=1)
```

After building our model we check the performance of our model which is common practice using any machine learning algorithm. We evaluate the score of our model using the piece of code below.

```
trainScore = model.evaluate(X_train, y_train, verbose=0)
print('Train Score: %.4f MSE (%.4f RMSE)' % (trainScore[0], math.sqrt(trainScore[0])))

testScore = model.evaluate(X_test, y_test, verbose=0)
print('Test Score: %.4f MSE (%.4f RMSE)' % (testScore[0], math.sqrt(testScore[0])))
```

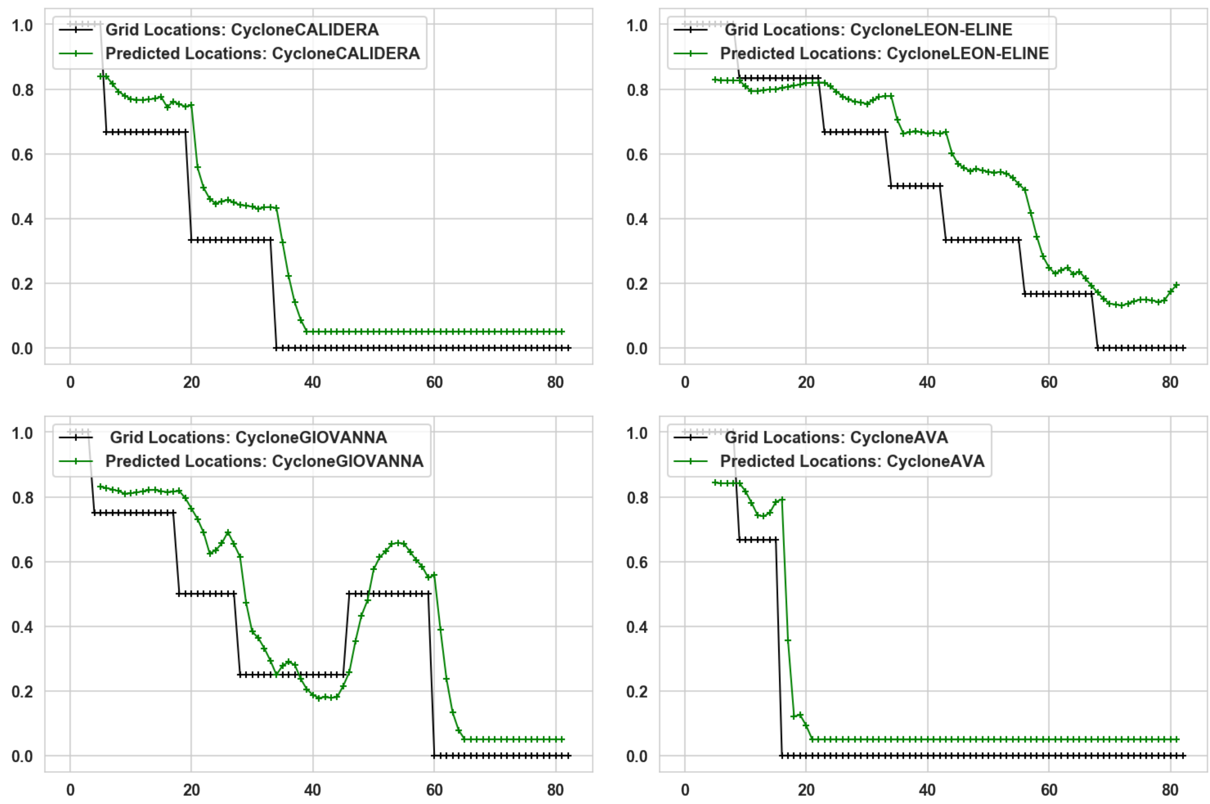


Figure 11: Trajectory of four cyclones selected randomly

We made a comparison between three activation functions respectively Rectified Linear Unit (ReLU), Hyperbolic tangent (tanh) and Sigmoid function just to justify our affirmations above about ReLU. See in the table below the results.

Function	Compilation Time (second)	Train Score (MSE)	Test Score(MSE)	Accuracy
ReLU	0.015	0.0307	0.0293	0.9464
tanh	0.022	0.0352	0.0322	0.9464
Sigmoid	0.022	0.0404	0.0392	0.9464

Table 1: Comparison of different activation functions

5. Conclusion

The end goal of this study was to predict the paths of cyclones using Recurrent Neural Networks over a grid system. This has been achieved in figure 11. From the results of our experiments performed, some parameters affect the training performance of RNN over grid model. Selection of the activation function and the number of neurons in the hidden layer impact the accuracy of our predictions.

In the table 1 above, we can see that in term of complilation time, the function ReLU is fast than hyperbolic tangent and simoid function. This confirms the assertions made above regarding the choice of the Rectified Linear Unit function (ReLU).

Figure 10 shows that our model RNN was able to successfully encapsulate and forecast the future trajectory paths of **Madagascar** cyclone at six-hour intervals. Figure 10 shows also that the grid locations predicted followed the similar trajectory behavior as the real cyclone trajectory.

One problem, the model has much less data to train on, which likely can affects the accuracy of the predictions. However, it is reasonable that there is less data for cyclones and storms, especially in Africa. The problem with using LSTM is that it can learn sequences, but cannot capture dependencies between correlating sequences.

To summarize, the choice of recurrent neural network employed over a grid system in this essay is to encapsulate the nonlinearity and complexity behind forecasting cyclones trajectories and potentially to increase the accuracy which is real issue of predicting for given event. We turned our model to predict the next cyclone location at six hours, as the data was collected at each six hours by the Bureau of Meteoology (BoM) Australian agency and National Hurricane Center (NHC). The error associated to the prediction mean-squared error (MSE) and root-mean-squared error (RMSE) were 0.03 and 0.18, for the training and 0.0276 MSE (0.1661 RMSE) for testing, our model predicted with 94.64% accuracy.

The idea of recurrent neural networks over grid-based models are not new, the combination of both to model complex nonlinear temporal relationships is a novel contribution. The advantage of this model is that it can be trained or predict cyclones of any type.

The high accuracy using Recurrent Neural Network over grid system can be explained by the fact that the grid method tends to reduce truncation errors and RNN learns the behavior of our cyclones from one grid location to the next and not general cyclone trajectories which can differ highly for given the different nonlinear and dynamic features. As a result,this model seems to be ideal for RNNs to handle the complexity of cyclone trajectories optimally.

In our current work, as we use longitude and latitude to derive the grid location, our future work will consist of using another deep learning process, such that we can reduce the conversion weight from grid locations to latitude and longitude coordinates independently of the size of the data.

We can also implement Bayesian neural network which is based on the probability and combined with our grid-based RNN that could increase the accuracy as Bayesian models could quantify the uncertainty of our prediction. Uncertainty should be considered in weather forcasting, in particular tropical cyclones forecasting. This story of uncertainty is very important in all decision making processes because it dictates decision making.

Appendix

The following are a list of abbreviations

ANN: Artificial Neural Networks

RNN: Recurrent Neural Networks

LSTM: Long Short Term Memory

BoM: Bureau of Meteorology

NHC: National Hurricane Center

MSE: Mean-Squared Error

RMSE: Root-Mean-Squared Error

ReLU: Rectified Linear Unit

tanh: Hyperbolic tangent

Epoch: One iteration of providing the networks with an input and updating the network's weights

Batch-size: Number of samples per gradient update.

Rmsprop: Root Mean Square Propagation

NOAA: National Oceanic and Atmospheric Administration

NWP: Numerical Weather Prediction

API: Application Programming Interface

DTW: Dynamic Time Warping

NCEP: National Centers for Environmental Prediction

TD: Tropical cyclone of tropical depression intensity ($<$ 34 knots)

TS: Tropical cyclone of tropical storm intensity (34-63 knots)

H1: Hurricane 1 (wind Speed between 64-82 knots)

H2: Hurricane 2 (wind Speed between 83-95 knots)

H3: Hurricane 3 (wind Speed between 96-112 knots)

H4: Hurricane 4 (wind Speed between 113-135 knots)

H5: Hurricane 5 (wind Speed $>$135 knots)

ET: Extratropical cyclone, a cyclone of any intensity for which the primary energy source is baroclinic.

Acknowledgements

I would like to thank my supervisor Professor Ernest Fokoué, who has guided me through this project. Thank you to all of AIMS staff, for granting me the opportunity to work on this project, and for a very memorable visit to Rwanda.

Thank you Dr Alexandra Rogers my personal tutor, it was a pleasure to work with you.

Thanks to Dr Hermann SORE research professor at Nazi Boni University in Bobo Dioulasso (Burkina Faso) for his advice.

To my father Alassane OUEDRAOGO and to my deceased mother Maimouna OUATTARA may the almighty grant you eternal rest. May they find in this modest work the expression of my deep gratitude!

References

Bergstra, J., Yamins, D., and Cox, D. D. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference*, pages 13–20. Citeseer, 2013.

Birchfield, G. E. Numerical prediction of hurricane movement with the use of a fine grid. *Journal of Meteorology*, 17(4):406–414, 1960.

Collins, S. N., James, R. S., Ray, P., Chen, K., Lassman, A., and Brownlee, J. Grids in numerical weather and climate models. *Climate Change and Regional/Local Responses*, pages 111–128, 2013.

Giuseppe Ciaburro, B. V. *Smart models using CNN, RNN, deep learning, and artificial intelligence principles*, volume 314. Packt Publishing Ltd, September 2017.

Gwimbi, P. The effectiveness of early warning systems for the reduction of flood disasters: some experiences from cyclone induced floods in zimbabwe. *Journal of Sustainable Development in Africa*, 9(4):152–169, 2007.

Hall, T. M. and Jewson, S. Statistical modelling of north atlantic tropical cyclone tracks. *Tellus A: Dynamic Meteorology and Oceanography*, 59(4):486–498, 2007.

Karpathy, A., Johnson, J., and Fei-Fei, L. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

Krishnamurti, T., Kumar, V., Simon, A., Bhardwaj, A., Ghosh, T., and Ross, R. A review of multimodel superensemble forecasting for weather, seasonal climate, and hurricanes. *Reviews of Geophysics*, 54 (2):336–377, 2016.

Kurihara, Y., Tuleya, R. E., and Bender, M. A. The gfdl hurricane prediction system and its performance in the 1995 hurricane season. *Monthly weather review*, 126(5):1306–1322, 1998.

Le Bellec, A. North-eastern madagascar and cyclone enawo. *The State of Environmental Migration*, 2018.

Lee, R. S. and Liu, J. N. Tropical cyclone identification and tracking system using integrated neural oscillatory elastic graph matching and hybrid rbf network track mining techniques. *IEEE Transactions on Neural Networks*, 11(3):680–689, 2000.

McDermott, P. L. and Wikle, C. K. An ensemble quadratic echo state network for non-linear spatio-temporal forecasting. *Stat*, 6(1):315–330, 2017.

Michel-Kerjan, E. O. Catastrophe economics: the national flood insurance program. *Journal of economic perspectives*, 24(4):165–86, 2010.

Moradi Kordmahalleh, M., Gorji Sefidmazgi, M., and Homaifar, A. A sparse recurrent neural network for trajectory prediction of atlantic hurricanes. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 957–964, 2016.

Tannehill, I. R. *The hurricane*, volume 94. US Department of Commerce, Weather Bureau, 1956.

Wang, H., Schemm, J.-K. E., Kumar, A., Wang, W., Long, L., Chelliah, M., Bell, G. D., and Peng, P. A statistical forecast model for atlantic seasonal hurricane activity based on the ncep dynamical seasonal forecast. *Journal of climate*, 22(17):4481–4500, 2009.

Webots. Cyclone idai's effect on southern africa. Webots, https://www.mercycorps.org/blog/cyclone-idai-facts, Accessed April 2020a.

Webots. Bureau of meteorology. Webots, http://www.bom.gov.au/cyclone/tropical-cyclone-knowledge-centre/history/tracks, Accessed April 2020b.

Webots. National hurricane center. Webots, https://coast.noaa.gov/hurricanes, Accessed April 2020c.

Webots. Kalman filter. Webots, https://medium.com/@siddheshzanj/extended-kalman-filter-94fe07fd5c79, Accessed March 2020d.

Webots. Recurrent neural network. Webots, https://www.researchgate.net/figure/Feed-forward-and-recurrent-ANN-architecture_fig1_315111480, Accessed March 2020e.

Webots. Tropical cyclone. Webots, https://www.britannica.com/science/tropical-cyclone, Accessed March 2020f.

Webots. recurrent neural network. Webots, https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/, Accessed March 2020g.

Wikipedia. Black scholes. Wikipedia, the Free Encyclopedia, http://en.wikipedia.org/wiki/Black%E2%80%93Scholes, Accessed April 2012.

Zehnder, J. A. The interaction of planetary-scale tropical easterly waves with topography: A mechanism for the initiation of tropical cyclones. *Journal of the atmospheric sciences*, 48(10):1217–1230, 1991.

Zehnder, J. A. and Gall, R. L. Alternative mechanisms of tropical cyclone formation in the eastern north pacific. *Atmósfera*, 4(1):37–51, 1991.