Hybrid Architecture for Long Short-Term Memory and Autoregressive: Application in Time Series Data for Birth Prediction

SAKAYO TOADOUM SARI (toadoum.sari@aims-cameroon.org)
African Institute for Mathematical Sciences (AIMS)
Cameroon

Supervised by: Dr. Marcellin ATEMKENG
Centre for Radio Astronomy Techniques & Technologies (RATT), Department of Mathematics, Rhodes University, SA

December 20, 2019

Submitted in Partial Fulfillment of a Cooperative Masters Degree in Industrial Mathematics at AIMS-Cameroon

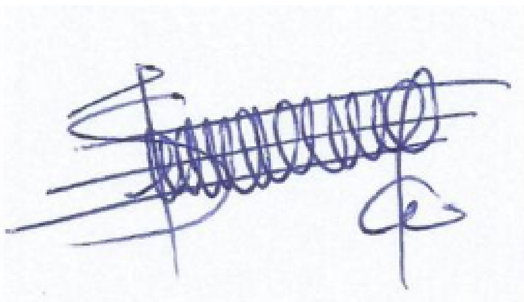
Abstract

An Autoregressive (AR) and Autoregressive Integrated Moving Average (ARIMA) architectures have been successfully applied as linear models in time series forecasting. Today, Recurrent Neural Networks (RNN); a sub-field of Artificial Neural Networks (ANN) are showing promising results in predicting time series forecasting when compared to ARIMA or AR. In many cases, our data comes from random distribution processes which makes it difficult to predict the time series data using a linear architecture such as ARIMA or AR that also considers that the data comes from a normal distribution. RNN draw their potential from the latter. That said, instead of considering the behavior of time series as linear, they are capable to understand the random process via non-linear systems. A type of RNN are Long Short-Term Memory (LSTM). LSTM is capable to mimic the behavior of randomly generated data over time via complex units with various components such as weights, activation functions, backward and forwards neurons connections, and connections between neurons of the same layer.

In this work, we show-cast the applicability of ARIMA, AR, and LSTM on time series birth data to predict future births based on archive birth data. The ARIMA, AR, and LSTM show a Root Mean Square Error (RMSE) of 0.63, 0.585, 0.469 respectively. The RMSE shows that the LSTM outperforms the AR then the AR outperforms the ARIMA. We then propose a new sequential hybrid architecture that draws its benefit from the LSTM and AR resulting in the lowest RMSE of 0.15.

Declaration

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



SAKAYO TOADOUM SARI, December 20th, 2019.

Contents

Abstract i

List of Figures v

1 INTRODUCTION 1

1.1 Problem . . . 1

1.2 Methods . . . 1

1.3 Contribution . . . 2

1.4 Thesis Organization . . . 2

2 Literature review 3

2.1 General machine learning leading . . . 3

2.2 Decision Modeling . . . 5

2.3 Model Generalization . . . 6

2.4 Mathematical Model for Time Series Analysis . . . 7

2.5 Regression Algorithms . . . 8

2.6 Hyper-parameters selection . . . 12

3 Methodology 13

3.1 Data Preparation and Data Preprocessing . . . 13

3.2 Exploratory Data Analysis . . . 14

3.3 Data Preparation . . . 16

3.4 Time-series forecasting models used . . . 17

3.5 Train and Test Set . . . 18

3.6 Performance Evaluation Metrics for Regression . . . 18

4 Results and Discussions 20

5 Important links 33

Acknowledgements 34

References 35

Glossary

AI Artificial Intelligence.

AIC Akaike Information Criteria.

ANN Artificial Neural Networks.

AR Auto Regressive.

ARIMA AutoRegressive Integrated and Moving Average.

BUNEC Bureau National de l'Etat civil du Cameroun.

FFNN Feed Forward Neural Networks.

KDE Kernel Density Estimation.

LSTM Long Short-term Memory.

MA Moving Average.

MAE Mean Absolute Error.

ML Machine Learning.

RMSE Root Mean Square Error.

RNN Recurrent Neural Networks.

List of Figures

2.1 The flowchart training, testing, and predicting from an ML model [9]. 3

2.2 The flowchart describing the process with all steps of applying supervised ML to a real-world problem [8]. 5

2.3 Supervised learning classification techniques with different algorithms using in this methods [8]. 6

2.4 Examples of under-fitting(left), appropriate-fitting(middle) and over-fitting(right) [10]. . 7

2.5 A neuron in an artificial neural network with inputs, weights, and output [6]. 9

2.6 An illustration of FFNN, RNN and LSTM memory block: (a) FFNN; (b) RNN; (c) LSTM memory block [13]. 11

3.1 Flowchart showing the workflow of the "project" (from data analysis up to model evaluation). 14

3.2 Time series of birth count per day showing some irregularities in the "daily count" column. 15

3.3 Distribution plot of our dataset showing that the data has right-skewed distribution. . . 16

3.4 Diagram of the total training process of the LSTM model. 17

3.5 Flowchart of the Sequential Hybrid model. The residual of the AR becomes the input of the output of the LSTM. The output of the model combines the output of AR and LSTM. 18

4.1 The Plot of MVA. MVA2 is a moving average of 2 weeks with a shift of 1 week, MVA4 is a moving average of 4 weeks with a shift of 1 week, MVA6 is a moving average of 6 weeks with a shift of 1 week, and MVA8 is a moving average of 8 weeks with a shift of 1 week. 20

4.2 Histogram and density plot of the data after transformation (feature MVA8). 22

4.3 A plot of data (green curve), training set (red curve) and prediction (blue curve) for the AR model. 23

4.4 A plot of the test (blue curve) and predicted data (red curve) for the AR model. 24

4.5 The residual plots for the ARIMA model. Top left is the plot of residual error, top right the density plot, bottom left the Q-Q-plot, and the bottom right the correlogram. 25

4.6 The plot of data (green curve), training (red curve) and prediction (blue curve) for the ARIMA model. 26

4.7 A plot of test data (blue curve) and predicted data (red curve) for the ARIMA model. . 26

4.8 A plot of data in green, training set in red and predicted set in blue. 27

4.9 Plot of losses(training set in blue and validation test in orange). 28

4.10 A plot of test data (red curve) and predicted set (blue curve). 29

4.11 A plot of data (green curve), training (red curve) and predicted (blue) set of hybrid AR&LSTM model. 30

4.12 Plot of test data (red curve) and predicted (blue curve) set of hybrid AR&LSTM model. 30

1. INTRODUCTION

Throughout history, technology has been the driving force of change. From movable type, television, up to the Internet, technology has been embraced and incorporated into our daily lives. Within the evolution of civilized society, the vast rewards of technological innovations have far outweighed the negatives. The digital revolution has altered conceptions of time and distance. It has created a wealth of information that is available at the stroke of a key. As an integrand part of this spectacular success, digitalization has proved to be quite game-changing in terms of improvement of the process, consistency, and quality of quite a lot of businesses. Thanks to the age of the digital era in which we currently live, almost every field of science that results in studying population needs data as a piece to bring evidence of facts. This can help get precise information about the dynamic of a population and get estimates of things like the growth rate, the infant mortality rate. Thus the importance of digitalization of civil status documents and archival storage system. Infantile mortality which, sadly is one of the challenges still encountered in Africa, especially in Central Africa, as well as the improvement of medical services, could be addressed using such data coupled with an appropriate method of analysis. Speaking of methods, powerful tools such as Machine learning and deep learning can show or predict the number of population in the future based on those data.

1.1 Problem

The department of Civil Status of the Limbe City Council has been always responsible for making and delivering the birth certificate, death certificate, and marriage certificate. However, the process through which these certificates are established is time-consuming, and well not tracked. Indeed, the civil Status department uses booklets provided by the Bureau National de l'Etat civil du Cameroun (BUNEC) in Buea (Cameroon) to follow up with the process. BUNEC supervises, controls, regulates and evaluates the national civil status system. That process though obsolete is still used by the city council despite a lot of drawbacks such as weak traceability, costly in time and money for both citizens and the city council, unreliability of the archival storage system, etc. It is then important to digitize the process of certificates establishment, in such a way that citizens could effortlessly request for certificates and services online, and via the digitized workflow system and help to save data for more research. Furthermore, to understand the dynamics of a population as well as improving the life quality of a population, it's very critical to conduct studies regularly about births, marriages, and deaths of habitants of a given location and keep records of them.

1.2 Methods

In this work, we conduct the following studies:

- We will develop a database management system with a website as an interface that will help Limbe City Council to record data about birth, death, and marriage.
- We will conduct a comparative study for time series prediction using ARIMA, AR, LSTM and Hybrid AR&LSTM to predict the future birth patterns of the Limbe City Council data.
- We propose a new hybrid architecture based on LSTM and AR that outperforms ARIMA.

1.3 Contribution

Although ARIMA, AR essentially for linear update models plus some noise thrown in and LSTM essentially for a nonlinear time series model are well known in the literature we have applied them in the data. Based on the fact that the LSTM outperforms the AR then the AR outperform the ARIMA according to their results, we have come-out with an idea of combining the two best models AR and LSTM in single architecture that will take advantage of the two models and capture linear and nonlinear phenomena because it has both linear and nonlinear modeling capabilities.

1.4 Thesis Organization

This piece of work is divided into three parts. The first part presents the literature review about machine learning. We will explain briefly decision modeling, how to model from the data, generalized model and the mathematics behind each model used in this project.

The next part is about the methodology used to handle the problem.

The last section will round-up with some description of the methodology. This will be followed by the results as well as some perspectives.

2. Literature review

2.1 General machine learning leading

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use in performing a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence(AI). ML algorithms build a mathematical model based on sample data, known as "training data", to make predictions or decisions without being explicitly programmed to perform the task.

ML field seeks to understand at a precise mathematical level what capabilities and information are fundamentally needed to learn different kinds of tasks successfully and to understand the basic algorithmic principles involved in getting computers to learn from data and to improve performance with feedback [4].

Machine Learning algorithm is trained using a training data set to create a model. When new input data is introduced to the ML algorithm, it predicts based on the model and Figure 2.1 described the whole running process.

The prediction is evaluated for accuracy and if the accuracy is acceptable, the ML algorithm is deployed. If the accuracy is not acceptable, the ML algorithm is trained again and again with an augmented training data set.

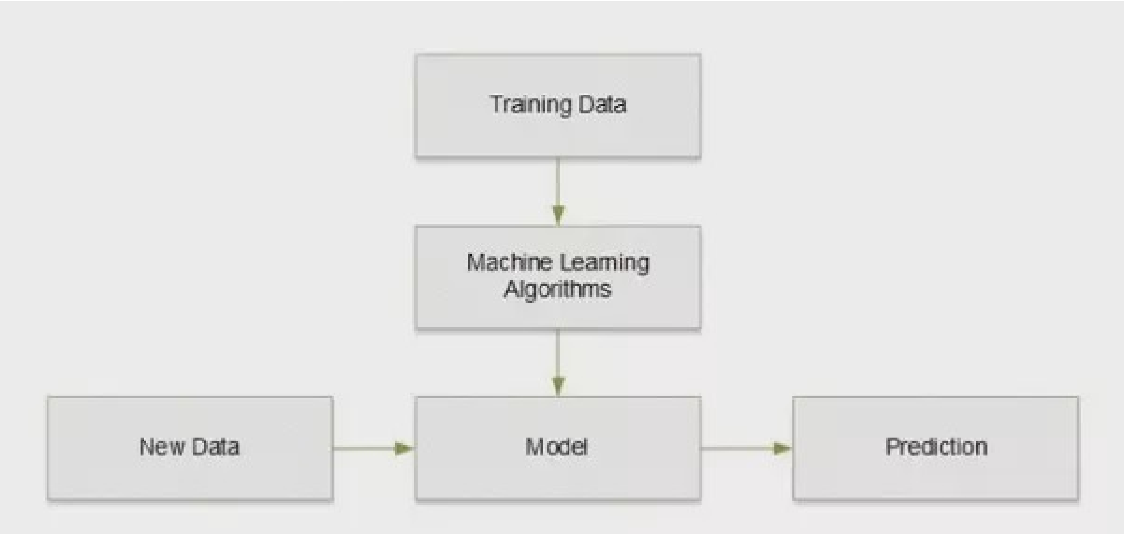


Figure 2.1: The flowchart training, testing, and predicting from an ML model [9].

ML algorithms are classifying into three categories:

- Supervised learning.
- Unsupervised learning.
- Reinforcement learning.

This stream especially supervised learning is explained in detail in the following section.

2.1.1 Supervised Learning.

Suppose that $\mathcal{F}$ is a class of parametric function and y the target variable with x input data. The objective is to find $f \in \mathcal{F}$ such that the expected risk

$$Rick(f) = E_p[L(x, y, f)] \tag{2.1.1}$$

is the weakest. In this notation, L is the loss function (or error function) which quantifies the extent to which $f(x)$ corresponds to the target value y. E_p stands for the expectation concerning the unknown distribution p. The choice of a loss function depends on the nature of the modeling problem as we will see in the section methodology. Figure 2.2 is an organigram of supervised ML with a classification task

2.1.2 Unsupervised Learning.

Let $\mathcal{G}$ be a class of parametric function and x an input data. The objective is to find $g \in \mathcal{G}$ such that the expected risk

$$Rick(f) = E_p[L(x, g)] \tag{2.1.2}$$

is the weakest. In this notation, L is the loss function (or error function) which quantifies the extent to which $g(x)$ corresponds to the best model to find without any knowledge of the target. The data in unsupervised learning is neither classified nor labeled and allowing the algorithm to act on that data(information) without guidance. E_p stands for the expectation for the unknown distribution p. The choice of a loss function depends on the nature of the modeling problem as we will see in the section methodology.

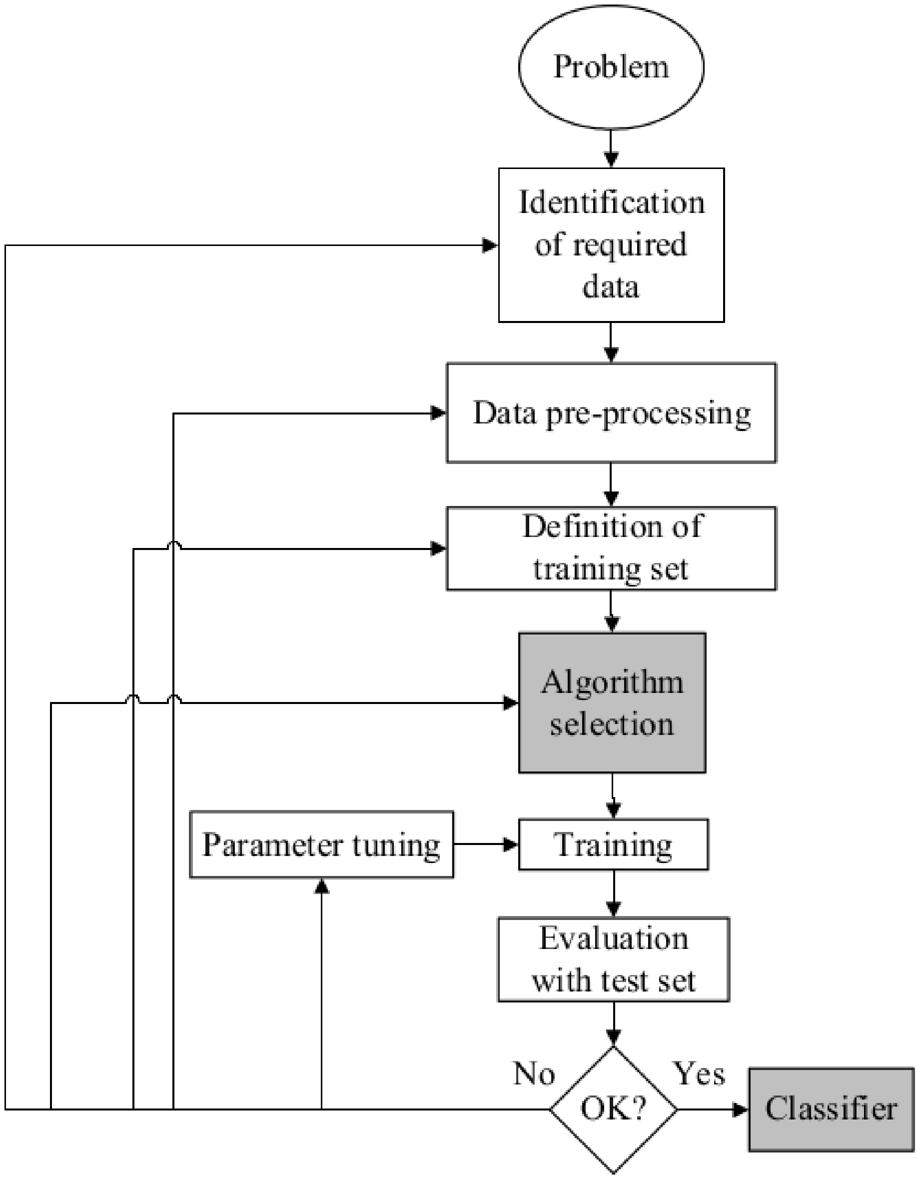


Figure 2.2: The flowchart describing the process with all steps of applying supervised ML to a real-world problem [8].

2.2 Decision Modeling

To construct decision model from data, it is necessary to have certain number of observations concerning all the considered variables. Each problem is described by a set of "explanatory" variables and a variable to explain. A decision model will be validated on observations of the same values of the explanatory variables and those associated with the explained variable. Once validated, however, this model will be used for observations that are limited to the values of the explanatory variables to obtain an estimate of explained variables.

2.2.1 Types of decision problems.

There are three types of decisions problems:

- **Classification:** The variable explained is the nominal variable, each observation is associated with a modality (class) and only one. Figure 2.3 is an organigram of supervised ML with a classification task.
- **Regression:** The explained variable is a quantitative variable that takes values in a sub-domain of the set of real numbers.
- **Structured prediction:** The explained variables take values in a structured set of data.

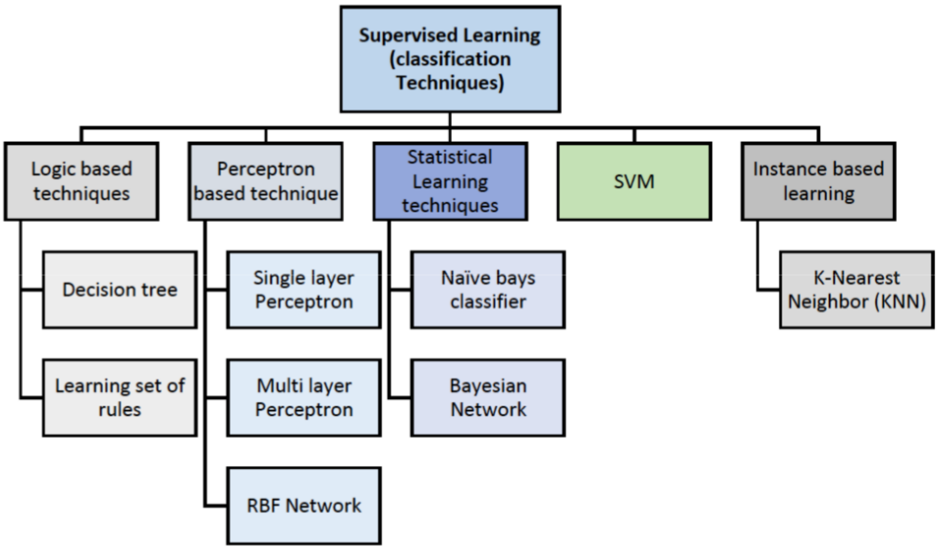


Figure 2.3: Supervised learning classification techniques with different algorithms using in this methods [8].

2.3 Model Generalization

The construction of decision models from data exploits a set of observations for which the supervision information is present (the values of the explained variable are know). Once constructed, these models are used to make prediction, i.e, to estimate the value of the explained variable on new observations for which the supervision information is absent. the ability to generalize a model is the ability of the model to make good predictions about new observations not previously seeing by the models.

Generalization refers to how well the concepts learned by a ML model applied to specific examples not seen by the model when it was learning.The goal of a good ML model is to generalize well from the training data to any data from the problem domain. This allows us to make predictions in the future on data the model has never seen.

There is a terminology used in ML when we talk about how well a ML model learns and generalizes to new data, namely over-fitting and under-fitting. Figure 2.5 shown an over-fitting and under-fitting and they are the two biggest causes for the poor performance of ML algorithms [5].

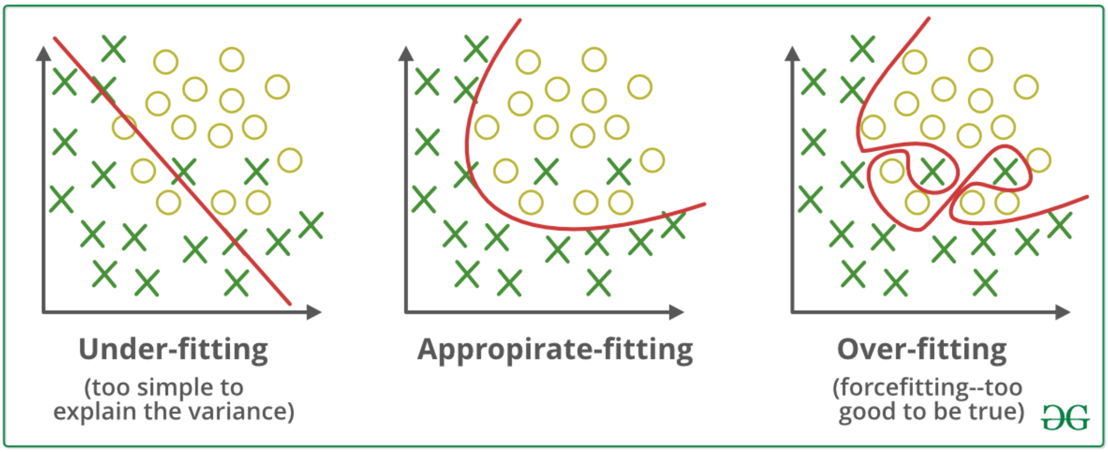


Figure 2.4: Examples of under-fitting(left), appropriate-fitting(middle) and over-fitting(right) [10].

2.3.1 Over-fitting and Under-fitting.

Over-fitting is a modeling error that occurs when a function is too closely fitted to a limited set of data points. Over-fitting the model generally takes the form of making an overly complex model explain particularities in the data under study.

We have under-fitting when the model cannot capture the underlying trend of the data or does not fit the data well enough. It usually happens when we have fewer data to build an accurate model and also when we try to build a linear model with nonlinear data.

2.4 Mathematical Model for Time Series Analysis

Mathematically, a time series is given as

$$y_t = f(t), \tag{2.4.1}$$

where y_t is the time series value at time t. According to the Additive Model, a time series can be decomposed as

$$y_t = T_t + S_t + C_t + R_t, \tag{2.4.2}$$

where T_t, S_t, C_t, and R_t are the trend value, seasonal, cyclic and random fluctuations at time t respectively. This model assumes that all four components of the time series act independently of each other [2].

The multiplicative model assumes that the various components in a time series operate proportionately to each other. According to this model, we have

$$y_t = T_t \times S_t \times C_t \times R_t. \tag{2.4.3}$$

Different assumptions lead to different combinations of additive and multiplicative models as shown in [2]

$$y_t = T_t + S_t + C_tR_t. \tag{2.4.4}$$

Among those others, we note that time-series analysis can also be decomposed:

$$y_t = T_t + S_t \times C_t \times Rt \quad \text{or} \quad y_t = T_t \times Ct + S_t \times R_t. \tag{2.4.5}$$

2.5 Regression Algorithms

Regression algorithms is used to predict the output values based on input features from the data fed in the system.

2.5.1 Moving average (MA) models.

Rather than using past values of the forecast variable in a regression, a moving average model uses past forecast errors in a regression-like model.

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + ... + \theta_q \varepsilon_{t-q}, \tag{2.5.1}$$

where ε_t is white noise. We refer to this as an $MA(q)$ model, a moving average model of order q. Of course, we do not observe the values of ε_t, so it is not a regression in the usual sense.

2.5.2 Autoregressive (AR) model.

AR is a time series model that uses observations from previous time steps as input to a regression equation to predict the value at the next time step. an autoregressive model of order p can be written mathematically as:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \varepsilon_t, \tag{2.5.2}$$

where ε_t is white noise. This is like a multiple regression but with lagged values of y_t as predictors. We refer to this as an $AR(p)$ model, an autoregressive model of order p.

2.5.3 AutoRegressive Integrated Moving Average (ARIMA) model.

ARIMA is actually a class of models that explains a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values. The full model can be written as:

$$y'_t = c + \phi_1 y'_{t-1} + ... + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + ... + \theta_q \varepsilon_{t-q} + \varepsilon_t, \tag{2.5.3}$$

where y'_t is the differenced series (it may have been differenced more than once). The "predictors" on the right-hand side include both lagged values of y_t and lagged errors. We call this an $ARIMA(p, d, q)$ model, where :
$p =$order of the autoregressive part,
$d =$degree of first differencing involved,
$q =$order of the moving average part.

2.5.4 Artificial Neural Network for Time Series Data.

Artificial neural networks (ANN) or connectionist systems are computing systems that are inspired by, but not identical to, biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, without being programmed with task-specific rules.

In a nutshell, a typical ANN is composed of an input layer, hidden layers, and an output layer. The input layer is the first layer of the neural network which are one or many hidden layers brings the initial data into the system for further processing by subsequent layers of artificial neurons. The hidden layers are layers between the input and output layers, in which artificial neurons receive a set of weighted inputs and produce an output through an activation function. The output layer is the last layer of neurons that produces the outputs from the ANN modeling, any of the layers is a collection of neurons [3].

Neuron

A neuron is a computational unit that takes the inputs and makes calculations and produces an expected output. For example if, we have as inputs, $x_1, x_2, ..., x_n$ each one of input is multiplied by a specific weight, $w_1, w_2, ..., w_n$. A sum of this weighted input is x_i is then computed at each neuron given a bias b. Figure 2.6 is an example of neuron with inputs, weight and then output.

$$z = \sum_{i=0}^{n} w_i x_i + b \tag{2.5.4}$$

The z is then passed through a non-linear function f to produce the output $y = f(z)$. This output is then transmitted to other neurons.

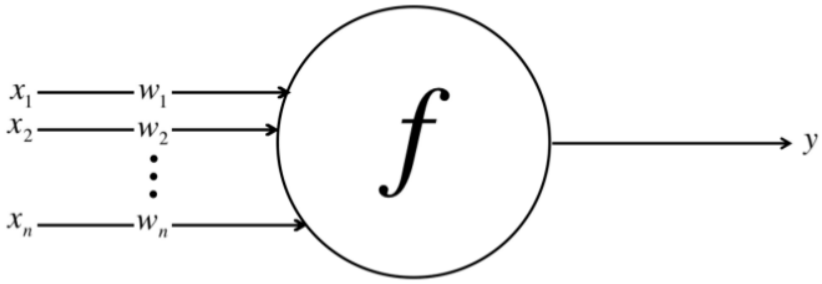


Figure 2.5: A neuron in an artificial neural network with inputs, weights, and output [6].

Sigmoid neuron

The sigmoid activation function is mathematically defined by :

$$f(z) = \frac{1}{1 + e^{-z}}. \tag{2.5.5}$$

The function means that when the logit is very small, the output of a logistic neuron is very close to 0 and when the logit is very large, the output of the logistic neuron is close to 1.

$Tanh$ neuron

$Tanh$ neuron uses a similar kind of S-shaped nonlinearity, but instead of ranging from 0 to 1, the output of tanh neurons range from -1 to 1. The function f is defined:

$$f(z) = tanh(z). \tag{2.5.6}$$

ReLU neuron

ReLu is a nonlinear function in nature and linear combinations of ReLus are also nonlinear. Relu is a suitable activation function due to its capability to approximate any random behavior by segments of Relu. ReLu is defined mathematically as:

$$f(z) = max(0, z). \tag{2.5.7}$$

The first initiative of neural networks (single-layer) was proposed by Rosenblatt and this model is limited to linearly separable patterns. That's why we will have the development of a powerful neural network called Multilayer Perceptron that will overcome the limitations of the single layer.

Later, based on the limitation of the Multilayer Perceptron to operate a proper adjustment that would enable the model to best generalize, we will have the born of backpropagation algorithm in Multilayer Perceptron that will help to handle the issue. Backpropagation has forward and backward phases.

2.5.5 Back Forward propagation.

Forward phase is where the activations propagate from the input layer to the output layer [1].

$$\begin{cases} z_k = \sum_i w_{i,k} x_k + b_k, \\ x_k = f(z_k). \end{cases} \tag{2.5.8}$$

Here x is the input, b is the weighted value from a bias that always has an output of 1, $w_{i,k}$ is the weight, i and k run across layers.

Backward phase is where the predicted actual value and the true nominal value in the output layer are propagated backward so it can modify the weights and bias values to improve accuracy.

$$L(w, b) = \sum_k \sum_i (y_{ik} - f_k(x_i))^2 \tag{2.5.9}$$

Given that $L(w, b)$ is quadratic, there exists at least one minimum, so gradient descent is used to find this global minimum. We are using the quadratic error in this case because we have a regression problem and for the regression problem, we use sum-of-squared errors as our measure of fit (error function). Hence, backpropagation relies on the gradient descent to minimize the error function $L(w, b)$. At the iteration, $(l + 1)$ the gradient descent is defined by:

$$\begin{cases} w_{kn}^{(l+1)} = w_{kn}^{(l)} - \gamma_l \sum_i \frac{\partial L_i}{\partial w_{kn}^{(l)}}, \\ b_{kn}^{(l+1)} = b_{kn}^{(l)} - \gamma_l \sum_i \frac{\partial L_i}{\partial b_{kn}^{(l)}}, \end{cases} \tag{2.5.10}$$

where γ_l is the learning rate it gives the step speed towards convergent.

2.5.6 Long short-term memory neural network model.

LSTMs, are a special RNNs that are suitable for learning long-term dependencies. The key part that enhances LSTMs' capability to model long-term dependencies is a component called memory block. As illustrated in Figure.3.3(c), the memory block is a recurrently connected subnet that contains functional modules called the memory cell and gates. The memory cell is in charge of remembering the temporal state of the neural network and the gates formed by multiplicative units are responsible for controlling the pattern of information flow [13].

- x_t : Input vector;

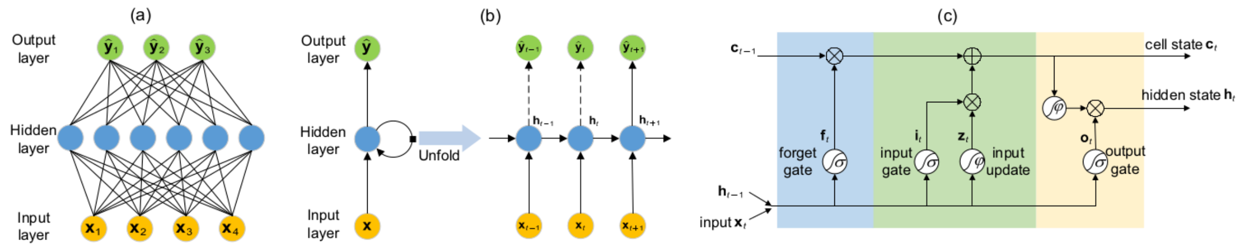


Figure 2.6: An illustration of FFNN, RNN and LSTM memory block: (a) FFNN; (b) RNN; (c) LSTM memory block [13].

- h_{t-1} : Previous cell Output;
- c_{t-1} : Previous cell Memory;
- h_t : Current cell Output;
- c_t : Current cell Memory.

The first step in constructing an LSTM network is to identify information that is not required and will be omitted from the cell in that step. This process of identifying and excluding data is decided by the sigmoid function, which takes the output of the last LSTM unit (h_{t-1}) at time $t-1$ and the current input (x_t) at time t. Additionally, the sigmoid function determines which part from the old output should be eliminated. This gate is called the forget gate (or f_t); where f_t is a vector with values ranging from 0 to 1, corresponding to each number in the cell state, c_{t-1} [12].

$$f_t = \sigma\left(w_f[h_{t-1}, x_t] + b_f\right), \tag{2.5.11}$$

where σ is the sigmoid function, w_f, and b_f are the weight matrices and bias, respectively, of the forget gate. The following step is deciding and storing information from the new input (x_t) in the cell state as well as to update the cell state. This step contains two parts, the sigmoid layer and second the $tanh$ layer. First, the sigmoid layer decides whether the new information should be updated or ignored (0 or 1), and second, the $tanh$ function gives weight to the values which passed by, deciding their level of importance (-1 to 1). The two values are multiplied to update the new cell state. This new memory is then added to old memory c_{t-1} resulting in c_t.

$$\begin{aligned} i_t =& \sigma\left(w_i[h_{t-1}, x_t] + b_i\right), && (2.5.12)\\ z_t =& tanh\left(w_n[h_{t-1}, x_t] + b_n\right), && (2.5.13)\\ c_t =& c_{t-1}f_t + z_t i_t. && (2.5.14) \end{aligned}$$

Here, c_{t-1} and c_t are the cell states at time $t-1$ and t, while w and b are the weight matrices and bias, respectively, of the cell state. In the final step, the output values (h_t) is based on the output cell state (o_t) but is a filtered version. First, a sigmoid layer decides which parts of the cell state make it to the output. Next, the output of the sigmoid gate (o_t) is multiplied by the new values created by the $tanh$ layer from the cell state (c_t), with a value ranging between -1 and 1 [12].

$$\begin{aligned} o_t =& \sigma\left(w_o[h_{t-1}, x_t] + b_o\right), && (2.5.15)\\ h_t =& o_t tanh(c_t). && (2.5.16) \end{aligned}$$

Here, w_o and b_o are the weight matrices and bias, respectively, of the output gate.

2.6 Hyper-parameters selection

In ML, hyperparameters optimization or tuning is finding a set of optimal hyperparameters for a learning algorithm that result in an acceptable accuracy.

For LSTMs, learning rate, optimizers, dropout rate, number of hidden layers as well as the number of units in those layers define the full structure of the network[7]. The traditional way of performing hyperparameter optimization has been grid search, or a parameter sweep, which is simply an exhaustive searching through a manually specified subset of the hyperparameter space of a learning algorithm. A grid search algorithm must be guided by some performance metric, typically measured by cross-validation on the training set or evaluation on a held-out validation set.

In this project, our regression model will be the Recurrent Neural Networks long short-term memory neural networks(LSTM) because it is the most appropriate for time series data. And to confirm this statement, Rao.S.S shows in 2009 that Implementing Time Series Analysis techniques, LSTMs perform better than simple FFNNs [11]. In the same way, Rahayu at al. (2008) draws the same conclusion for complex problems.

3. Methodology

In this study, we aim to build a model that could predict the number of births in the future (number of births certificates drawn up per day) based on the "birth count per day" of previous years. Since the expected outcome or the value we try to predict is a continuous variable, we definitely have a regression problem.

3.0.1 Flowchart Diagram.

The flowchart below Figure 3.1, explains the method used to handle the problem in this project. Started by an observation of the raw data, a preprocessing will be completed. Subsequently, we will transform our data using MA whose role is instrumental in eliminating some noise and random variations. After obtaining a smooth /clean version of the data, we will apply /fit the AR, ARIMA, LSTM and hybrid AR&LSTM models to those. The last step will be the comparative analysis between different models and the selection of the best one based on some specific criteria of performance.

3.1 Data Preparation and Data Preprocessing

The data was collected at Limbe City Council's archive because the database management system is not yet available, we have to create the data manually. The data were recorded on papers across different booklets from 2010 to 2019, we have structured a way of storing those pieces of information.

Though the process was a bit hard in terms of handling missing data and sometimes wrongly entered observations, we had to rely on the help of some of the Limbe City Council workers from Civil Status and Archive Department.

As a result of this, a pieces of information that could be used for analysis was finally recorded. More details will be given in the next chapter.

3.1.1 Missing Data Handling.

In the case of missing data, a decision has to be made on how to handle those. Some of the commonly used methods for dealing with missing data are deletion and imputation. In the case of our dataset, we use imputation because the dataset is small and we are also interested in using consecutive values in the future, especially for feature engineering.

3.1.2 Features Engineering.

Feature engineering can be defined as a process of generating new features from the existing ones. New features were introduced to be used for training the model to improve its performance. MA was used to develop new features. The MAs are used to get an overall idea of the trends in a dataset. They are very important in forecasting long-term trends especially in the case the data is noisy or the data has high variances among the data points. MA is used to balance the high variance. The MA is much more important because it reduces the effect of random variations in data (centralize the data points around the same mean and standard deviation). It makes the data smooth to show the trend clearly.

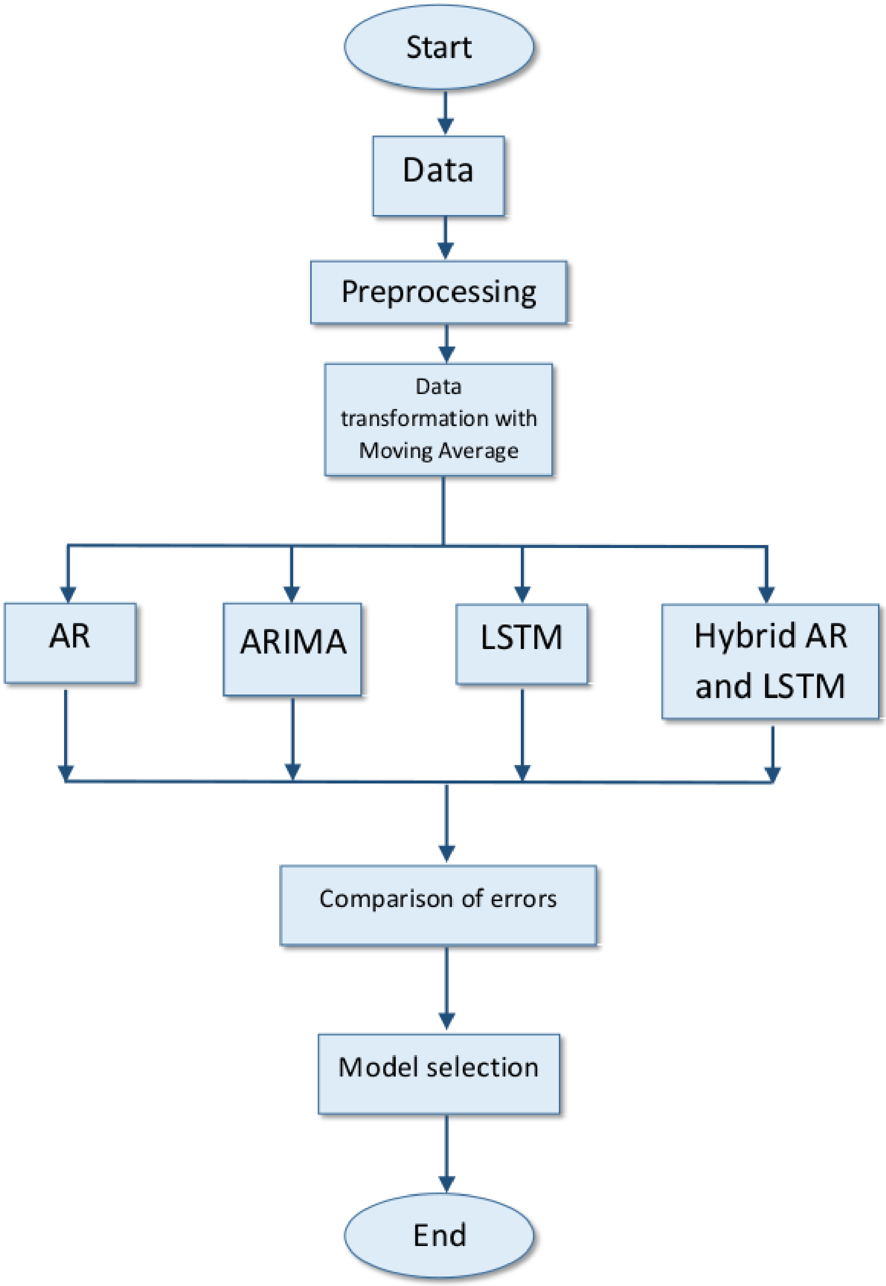


Figure 3.1: Flowchart showing the workflow of the "project" (from data analysis up to model evaluation).

3.2 Exploratory Data Analysis

The dataset submitted to our attention comprises several variables: name of the child, date of birth, sex, place of birth, occupation of father, occupation of the mother, place of birth of the father, place of birth of the mother, residence of parents, draw update.

The draw update is representing the date that each birth certificate is filled. Leveraging the use of some powerful functions contained in the Pandas library in Python, we aggregate our data to sort the data frame by unique dates.

Table 3.1: Summary table of the "daily count" column.

	daily_count
count	457.000000
mean	3.105033
std	2.942787
min	1.000000
25%	1.000000
50%	2.000000
75%	4.000000
max	23.000000

Table 3.1 shows a descriptive summary of birth count per day in our dataset. The dataset contains 457 observations of daily birth draw up from January 2010 to June 2019. We have an average of 3.10 cases per day and a standard deviation of 2.94.

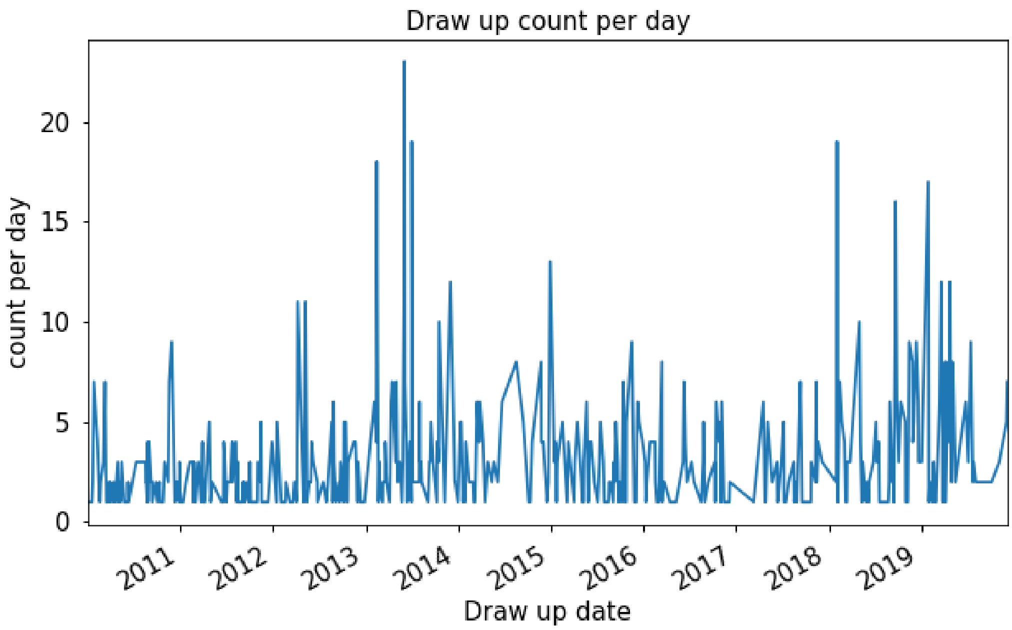


Figure 3.2: Time series of birth count per day showing some irregularities in the "daily count" column.

3.2.1 Distribution of the data.

From the summary table, we observe that the first quartile(25%) is 1 birth recorded, the third (75%) is 4 birth recorded, and the interquartile will be 3 births, so there is variability in the data and some values are far from the mean according to the max value that's why we have this distribution (figure 3.3). We need to transform the data in such a way that we will have a normal distribution which will represent more stability when training the models.

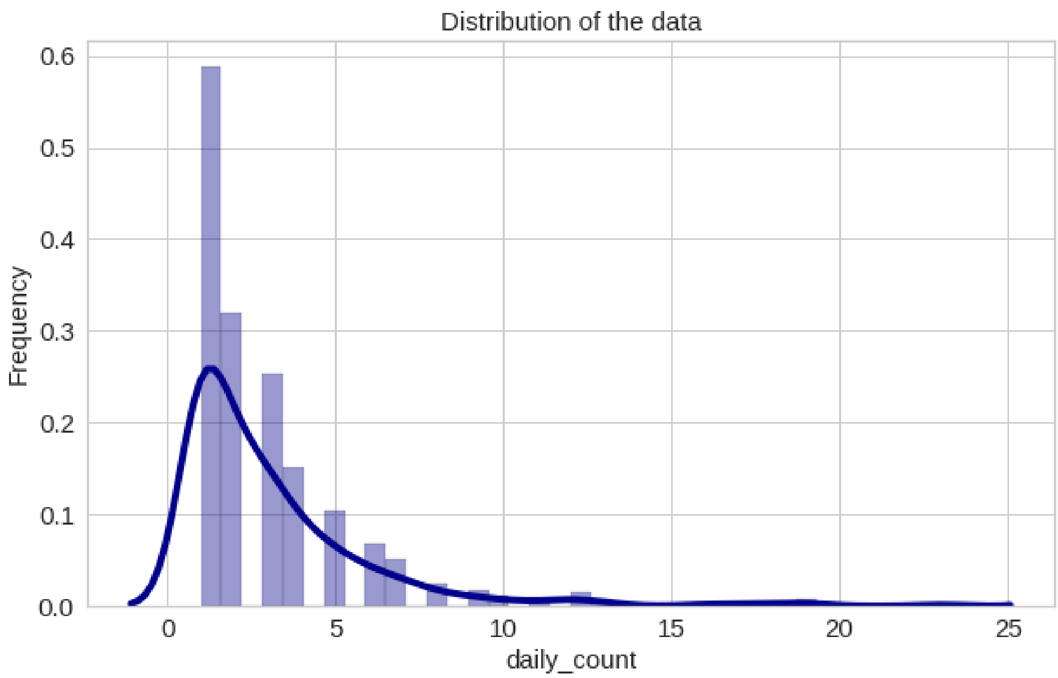


Figure 3.3: Distribution plot of our dataset showing that the data has right-skewed distribution.

3.3 Data Preparation

The original dataset has random variations. To avoid weak performance prediction of our models, because they will consider those high variances as noise, we have used MA to make a smooth or more informative version of our original dataset. MA is useful as a data preparation technique as it can help reduce the random variation and therefore, some noise in the observations. Here we are taking different cases, including two, four, six and eight weeks to compare and take the best feature to generate by MA and use for prediction.

Following this principle, five new variables /features have been generated from the input variables, namely **Baseline, MVA2, MVA4, MVA6, MVA8**. For example, the baseline will only have the first value as an empty and last eight values non-empty because of the shift of one (shift(1)) and MVA8 (a moving average of 8 with a shift 1) will have first eight values as empty values and one non-empty value as the ninth value. In these new variables, we are only interested in the last value from all the five variables. This method will help us to have a data centralized around the same mean and standard deviation. We will make a prediction and take the best moving average model.

3.4 Time-series forecasting models used

After making our data smooth, In this project, we will use Autoregressive (AR), Autoregressive Integrated Moving Average, LSTM and hybrid (AR&LSTM) models for prediction.

3.4.1 LSTM model.

The LSTM model in this project is, first of all, worked sequentially. The model has:

- four layers(input layer, two hidden layers, and an output layer);
- units=10 (10 neurons for each layer and 1 for the output layer);
- batch size=16;
- number of epochs=100;
- The optimizer used for the model is Adam.

Those hyper-parameters(units, epoch, batch size,optimizer) and 20% for validation give good results base on the Gridsearch and Figure 3.4 shows the total training process of the LSTM.

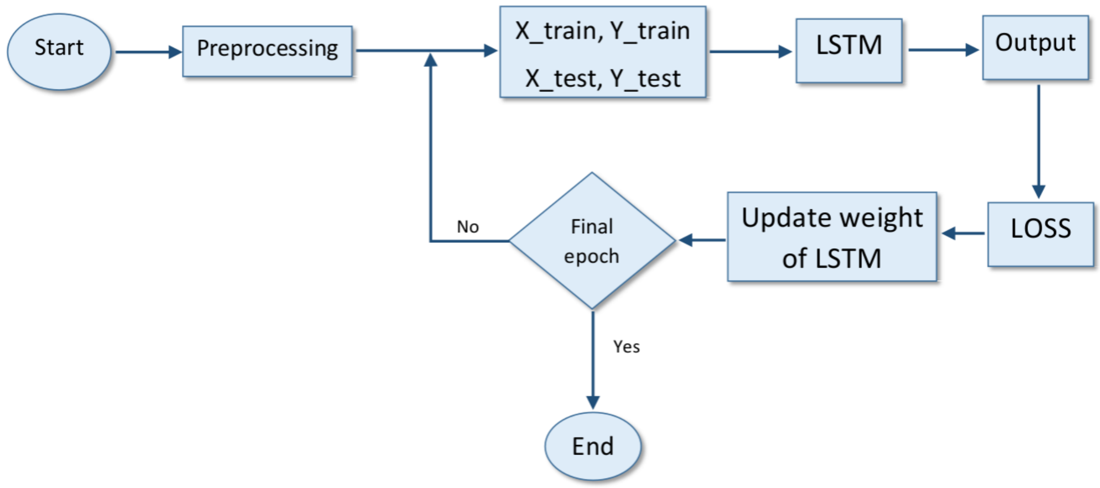


Figure 3.4: Diagram of the total training process of the LSTM model.

3.4.2 Hybrid AR and LSTM model.

This section concern our main contribution to this project. It is about the new approach that will fusion the AR and LSTM model capabilities to have a new architecture that will detect linear and nonlinear phenomena in our data. Time series is considered to be composed of a linear autocorrelation structure and a nonlinear component. That is,

$$Y_t = L_t + N_t, \tag{3.4.1}$$

where L_t denotes the linear component and N_t the nonlinear component and Y_t the input (time series data). These two components have to be estimated from the data. First, we let AR model the linear component, then the residuals from the linear model will contain only the nonlinear relationship. e_t is the residual at time t from the linear model, then $e_t = Y_t - \hat{L}_t$; where $\hat{L}_t$ is the forecast value for time t [14]. Figure 3.4 explains the process of this sequential hybrid model. The residuals of AR will be model using LSTM to detect the non-linearity part of the time series.

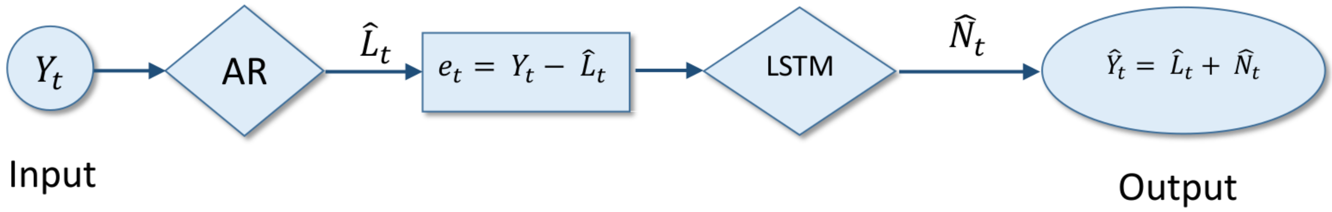


Figure 3.5: Flowchart of the Sequential Hybrid model. The residual of the AR becomes the input of the output of the LSTM. The output of the model combines the output of AR and LSTM.

The hybrid model should be able to pick up the advantages of each of AR and LSTM.

3.5 Train and Test Set

The dataset was then split into 80% (380 observations) for the train set and 20% (69 observations) for the test set so as to avoid under-training the model.

3.6 Performance Evaluation Metrics for Regression

Evaluation metrics explain the prediction performance of a model. An essential aspect of evaluation metrics is their ability to distinguish model outcomes. The choice of metric entirely depends on the model and the implementation method of the model. In this work, we will use MAE and RMSE.

3.6.1 Mean Absolute Error (MAE).

In MAE, the error is calculated as an average of absolute differences between the target values and the predictions. The MAE is a linear score which means that all the individual differences are weighted equally in the average. MAE is represented mathematically as follows:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}, \tag{3.6.1}$$

where y_i represents the actual value, $\hat{y}_i$ is the predicted value. Its properties are:

- The measure of how far the predictions were from the actual output.
- No precise idea about the direction of the errors.

- The smaller the MAE, the better the prediction.
- No penalization of extreme deviations.

3.6.2 Mean Squared Error (MSE).

MSE is the most simple and common metric for regression evaluation. MSE basically measures the average the squared error of our predictions. For each point, it calculates square difference between the predictions and the target and then averages those values. MSE is represented mathematically as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3.6.2}$$

where y_i target value, $\hat{y}_i$ is the predicted value. Its properties are:

- The opposite signed errors do not offset one another.
- It penalizes extreme errors that occurred while forecasting.
- MSE emphasizes the fact that the total forecast error is in fact much affected by large individual errors, i.e. large errors are much expensive than small errors.
- MSE does not provide any idea about the direction of overall error.
- MSE is sensitive to the change of scale and data transformations.

3.6.3 Root Mean Squared Error (RMSE).

Root Mean Square Error(RMSE) is the square root of MSE. The square root is introduced to scale the errors to be the same as the scale of targets.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} = \sqrt{MSE} \tag{3.6.3}$$

where y_i represents the actual value, $\hat{y}_i$ is the predicted value.

4. Results and Discussions

Throughout this chapter, we will discuss experiments and present the results obtained from the different techniques. We will start by visualizing our features create in the data to get a flavor of it. Then, performing AR, ARIMA, LSTM and Hybrid AR&LSTM models. We will use also performance evaluation metrics to select an efficient model with high prediction performance.

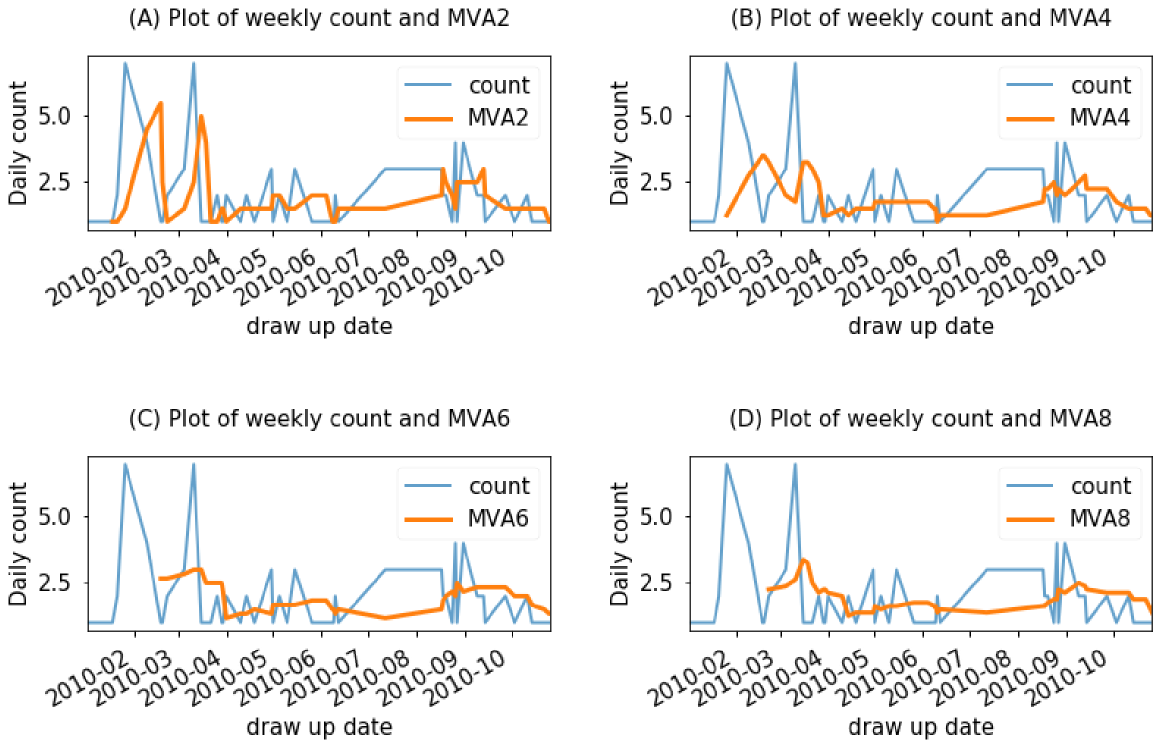


Figure 4.1: The Plot of MVA. MVA2 is a moving average of 2 weeks with a shift of 1 week, MVA4 is a moving average of 4 weeks with a shift of 1 week, MVA6 is a moving average of 6 weeks with a shift of 1 week, and MVA8 is a moving average of 8 weeks with a shift of 1 week.

Figure 4.1 shows the plots of weekly counts for different values of MVA. The top left shows a loss of two first weeks because we are doing the moving average of two weeks and a forward shift of 1 week. The moving average of two weeks which will make week one have an empty value and a forward shift will push this further into week three. Hence, MVA2 we will have two first weeks with empty values. The top right expresses a loss of four first weeks because we are doing the moving average of four weeks and a forward shift of 1 week. The moving average of four weeks which will make three weeks have empty values and a forward shift will push this further into week five. Then, MVA4 will have four first weeks with empty values and the same process is used for others. The bottom left reveals an MVA6 with six first weeks with empty values and The bottom rights shows an MVA8 with eight first weeks with empty values. With the use of the drop function inside the Pandas Library in Python, the first eight rows will be dropped leaving us with only 449 observations instead of 457 observations.

Table 4.1: Moving averages with a shift of 1.

Draw.up.date	daily_count	Baseline	MVA2	MVA4	MVA6	MVA8
2010-01-02	1	NaN	NaN	NaN	NaN	NaN
2010-01-11	1	1.0	NaN	NaN	NaN	NaN
2010-01-18	1	1.0	1.0	NaN	NaN	NaN
2010-01-21	2	1.0	1.0	NaN	NaN	NaN
2010-01-26	7	2.0	1.5	1.25	NaN	NaN
2010-02-09	4	7.0	4.5	2.75	NaN	NaN
2010-02-18	1	4.0	5.5	3.50	2.666667	NaN
2010-02-19	1	1.0	2.5	3.50	2.666667	NaN
2010-02-22	2	1.0	1.0	3.25	2.666667	2.25

Table 4.2: Moving averages with a shift of 1 after dropping missing values.

Draw.up.date	daily_count	Baseline	MVA2	MVA4	MVA6	MVA8
2010-02-22	2	1.0	1.0	3.25	2.666667	2.250
2010-03-05	3	2.0	1.5	2.00	2.833333	2.375
2010-03-11	7	3.0	2.5	1.75	3.000000	2.625
2010-03-16	1	7.0	5.0	3.25	3.000000	3.375
2010-03-19	1	1.0	4.0	3.25	2.500000	3.250
2010-03-22	1	1.0	1.0	3.0	2.500000	2.500
2010-03-26	2	1.0	1.0	2.50	2.500000	2.125
2010-03-29	1	2.0	1.5	1.25	2.500000	2.250

New features create using moving averages such as Baseline, MVA2, MVA4, MVA6, and MVA8 were taken as a prediction and based on its prediction performance we will choose our best feature for the remainder models. The performance prediction for each variable is shown in the table below.

Performance prediction of new features		
	MAE	RSME
Baseline	2.2	3.4
MVA2	2.12	3.21
MVA4	2.06	3.01
MVA6	2.05	2.99
MVA8	1.98	2.96

As a complement of the result displayed in the above table, MVA8 has the best performance and it will be only used as the predictor for our model.

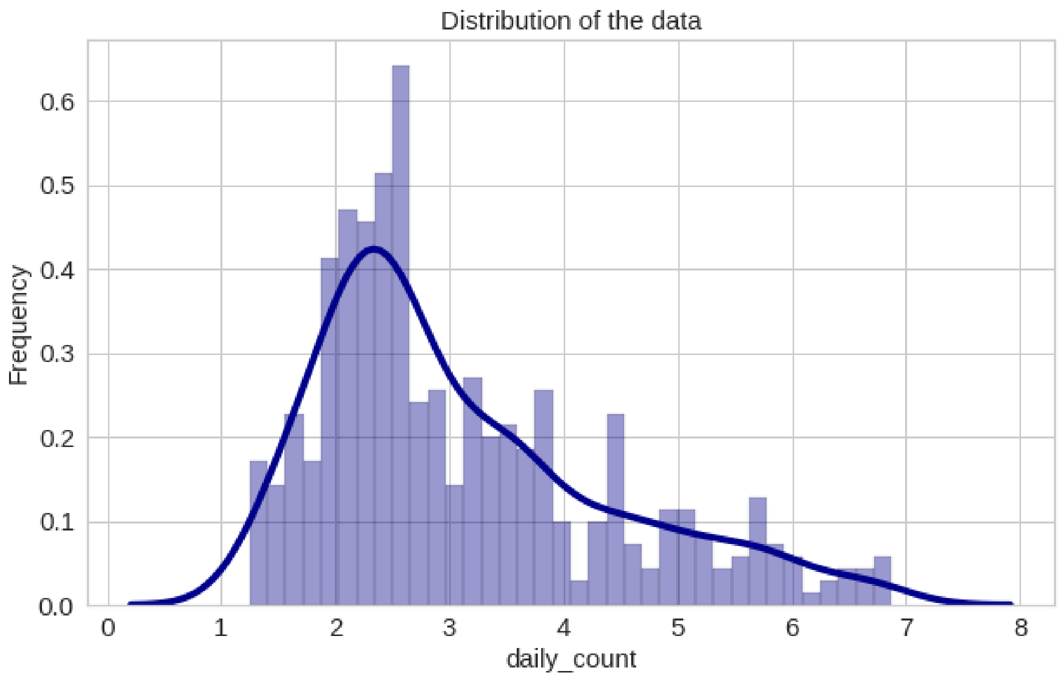


Figure 4.2: Histogram and density plot of the data after transformation (feature MVA8).

This distribution is much better than the first one (figure 3.2).

In the forecasting process, future data is predicted. For the evaluation of prediction of the number of birth in the future, miscellaneous prediction methods were used in the past. We use time-series prediction with AR, MA, and ARIMA models which predict future data (number of births) based on previous and current data. Here in Figure 4.3, the x-axis shows the number of days taken for forecasted the values and the y-axis shows the number of birth (MVA8).

In Figure 4.3, the red curve represents the forecasted (predicted) and the blue curve represents the number of births for 69 days, we have used 380 days to train our model in red and the real data is about 449 days of the number of births for the AR model. Those plots are showing the capability of the AR model to make predictions out of the training set. Figure 4.4 is a subset of Figure 4.3 and helps for good visualization. The prediction of the number of births for 69 is made in the part of our data that the model was not trained, this is to confirm that the model generalizes well. Our model here is AR(1).

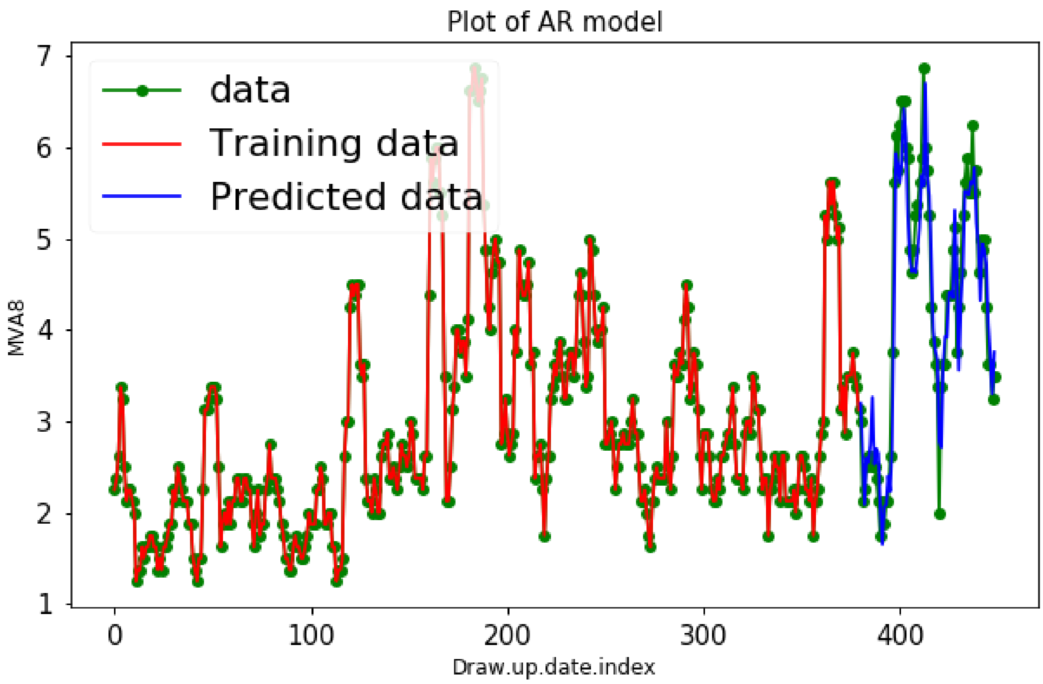


Figure 4.3: A plot of data (green curve), training set (red curve) and prediction (blue curve) for the AR model.

The table below shows prediction performances for the autoregressive model. The AR model gives a MAE=0.47 and RMSE=0.61.

Figure 4.4 shows that the predicted value almost matches the real value. In conclusion, our model was well trained.

Performance prediction of AR model	
MAE	RSME
0.47	0.61

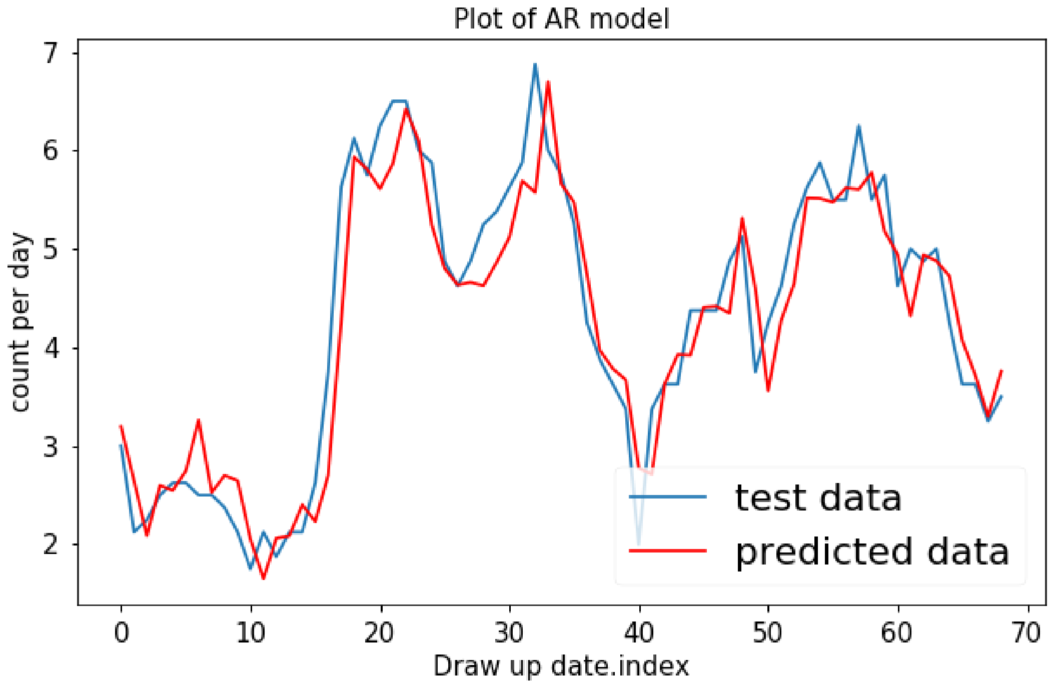


Figure 4.4: A plot of the test (blue curve) and predicted data (red curve) for the AR model.

The $auto\_arima$ is a function in $pdarima$ that uses a stepwise approach to search multiple combinations of p, d, q parameters and chooses the best model that has the least Akaike Information Criteria (AIC). The best model given by this method is $ARIMA(1, 0, 2)$ with AIC$= 592.159$. Our primary concern is to ensure that the residuals of our model are uncorrelated and normally distributed with zero-mean. If the seasonal ARIMA model does not satisfy these properties, it is a good indication that it can be further improved.

In this case, our model diagnostics suggest that the model residuals are normally distributed based on the following:

- In the top-right (Figure 4.5) plot, we see that the Kernel Density Estimation(KDE) line follows closely with the $N(0, 1)$ line (where $N(0, 1)$ is the standard notation for a normal distribution with mean 0 and a standard deviation of 1). This is a good indication that the residuals are normally distributed.
- The $q - q$ $plot$ (Figure 4.5) on the bottom left shows that the ordered distribution of residuals (blue dots) follows the linear trend of the samples taken from a standard normal distribution with $N(0, 1)$. Again, this is a strong indication that the residuals are normally distributed.
- The residuals over time (top left plot of Figure 4.5) don't display any obvious seasonality and appear to be white noise. This is confirmed by the autocorrelation (i.e. correlogram) plot on the bottom right, which shows that the time series residuals have low temporal correlation(one) with lagged versions of itself.

Those observations lead us to conclude that our model produces a satisfactory fit that could help us understand our time series data and forecast future values. Overall we can then use this model for forecasting.

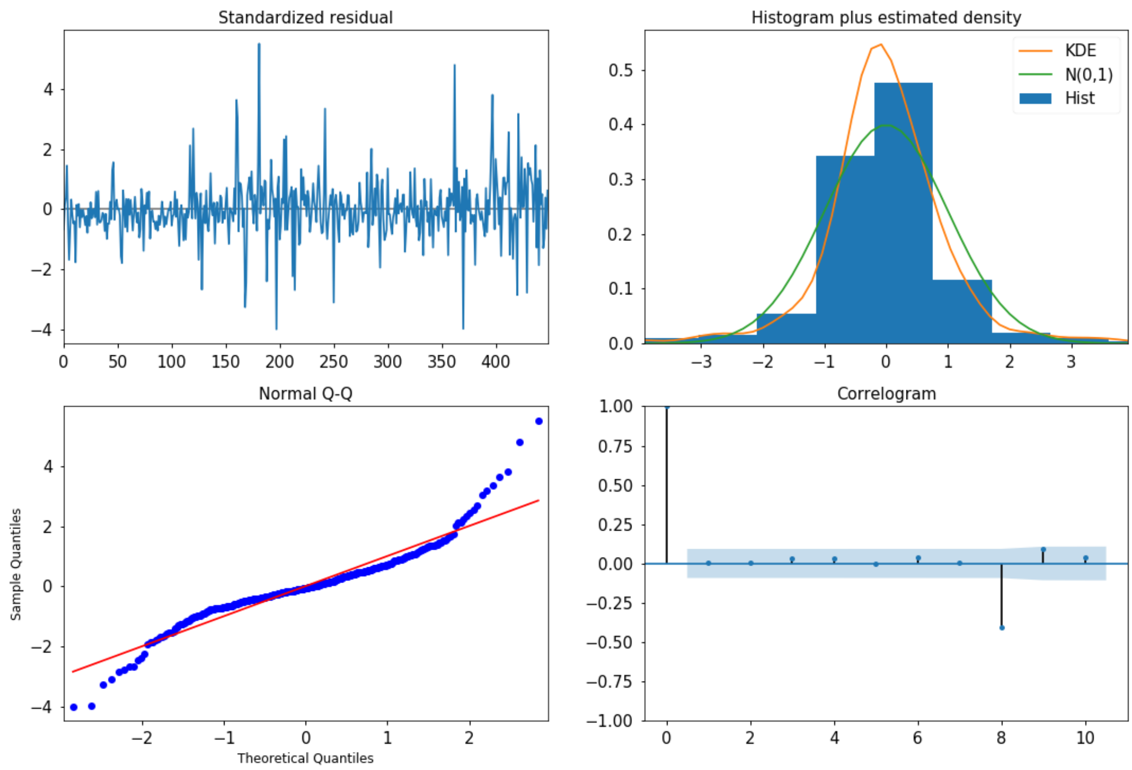


Figure 4.5: The residual plots for the ARIMA model. Top left is the plot of residual error, top right the density plot, bottom left the Q-Q-plot, and the bottom right the correlogram.

Figures 4.6 and 4.7 are showing the forecasting using the ARIMA model out of the training set. The x-axis is the number of days using for forecasting(prediction) and the y-axis is the number of births. Figure 4.7 is a subset of Figure 4.6. Those plots show that our model is learning very well and have the capacity to make a prediction in the new set of data. From Figure 4.6, the green curve is the real data, the red one is the set of data that we have used to train our model, and the blue curve is the part of our data that the model was not trained on but we have used for prediction to confirm that the model generalizes well. In figure 4.7, the blue curve is for test data and the red curve is the prediction of the test data.

Figure 4.7 plot shows the precision of our model, how the predicted model (red curve) is closed to the real test data (blue curve).

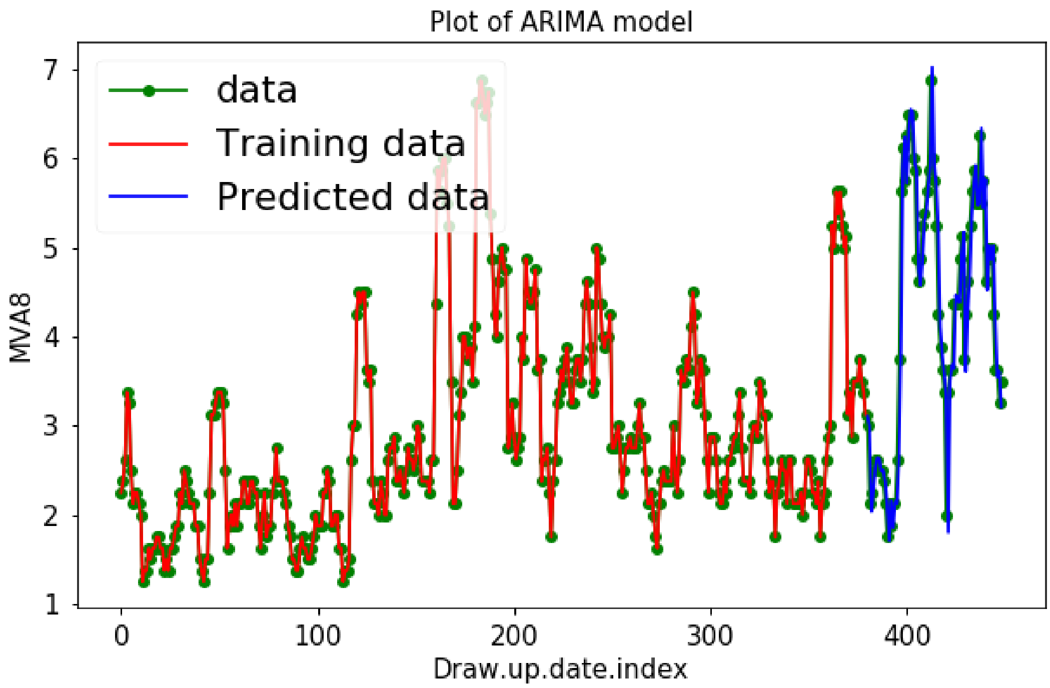


Figure 4.6: The plot of data (green curve), training (red curve) and prediction (blue curve) for the ARIMA model.

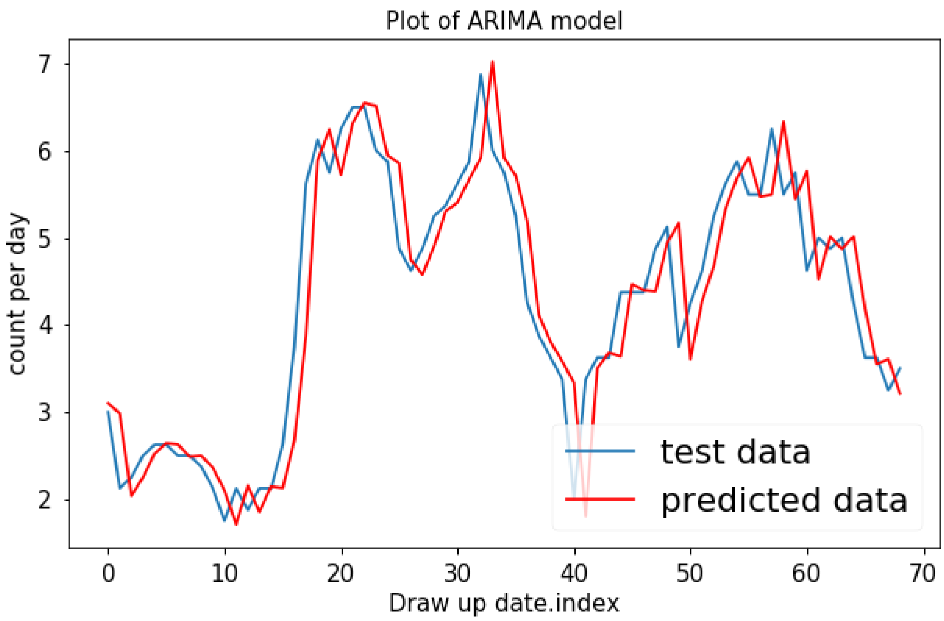


Figure 4.7: A plot of test data (blue curve) and predicted data (red curve) for the ARIMA model.

Performance prediction of ARIMA model	
MAE	RSME
1.6	0.63

This table shows prediction performances for the autoregressive integrated moving average (ARIMA). Our ARIMA model has a MAE=1.6 and RMSE=0.63.

The nonlinear model for the time series forecasting using in this project is the LSTM. Figure 4.8 shows how our LSTM model is learning and predicting out of the range of the training set. We can observe that predicted values closely match actual values of data and Figure 4.10 is a subset of Figure 4.8 it is for a good visualization for the prediction performance. From Figure 4.8, the green curve is the data, the red one is the set of data that we have used to train our model, and the blue curve is the part of our data that the model was not trained on but we have used for prediction and to confirm that the model generalizes well. Like other models before (AR&ARIMA), we have used for training 380 days and 69 days to predict the number of birth. In figure 4.10, the blue curve is for test data and the red curve is the prediction of the test data. The LSTM model outperforms the AR and ARIMA models.

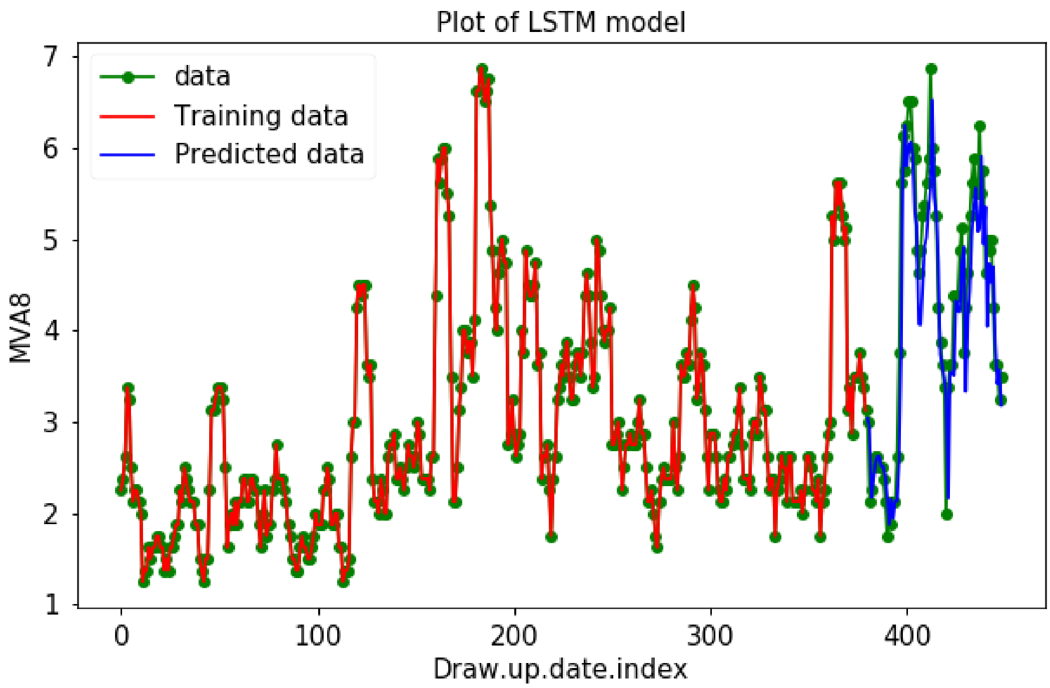


Figure 4.8: A plot of data in green, training set in red and predicted set in blue.

The Learning curve calculated from the training dataset that gives an idea of how well the model is learning and the validation curve calculated from the validation dataset and shows how well the model is generalizing. The loss of the model will almost always be lower on the training dataset than the validation dataset. This means that we should expect some gap between the train and validation loss learning curves. This gap is referred to as the "generalization gap". From Figure 4.9 below, we can observe that training and validation loss converge after forty-five epoch we can then say that we have a good fit because the validation loss curve is below the training curve and then there is a mixed of curves (converge) at a certain epoch. Figure 4.10 show the closeness of prediction of number of birth

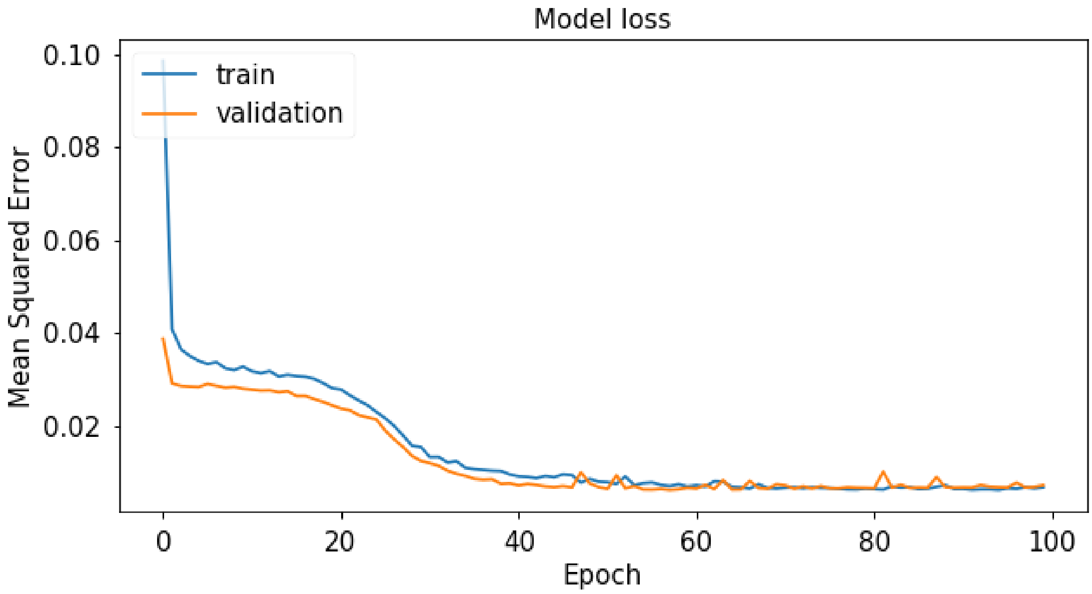


Figure 4.9: Plot of losses(training set in blue and validation test in orange).

(curve in blue) with the test data (curve in red).

Performance prediction of LSTM model	
MAE	RSME
0.4	0.469

This table shows the prediction performance of our LSTM model. MAE=0.4 and RMSE=0.469.

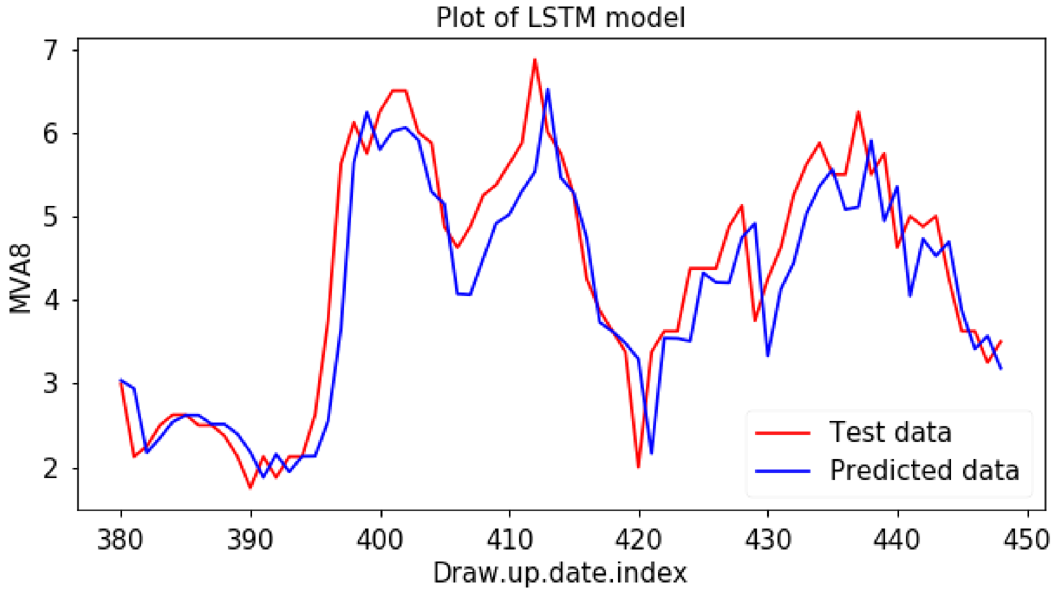


Figure 4.10: A plot of test data (red curve) and predicted set (blue curve).

The aim of building a hybrid model comes from the fact that a linear model is not sufficient since there are still some non-linear correlation structures left in the residuals. However, residual analysis is not able to detect any nonlinear patterns in the data. There is currently no general diagnostic statistics for nonlinear autocorrelation relationships. Therefore, even if a model has passed diagnostic checking, the model may still not be adequate in that nonlinear relationships and have not been appropriately modeled. Any significant nonlinear pattern in the residuals will indicate the limitation of the AR. By modeling residuals using LSTM, nonlinear relationships can be discovered [14].

Those plots below (Figure 4.11 and Figure 4.12) show us how our hybrid model is learning and predicting out of the range of the training set. We can observe that predicted values closely match actual values of data and Figure 4.12 is a subset of Figure 4.11 that will help for good visualization for the prediction performance. Figure 4.11 curve in green is for the data, the red curve is for the training set and the blue one is for prediction of the number of birth.

Performance prediction of the Hybrid model	
MAE	RSME
0.17	0.15

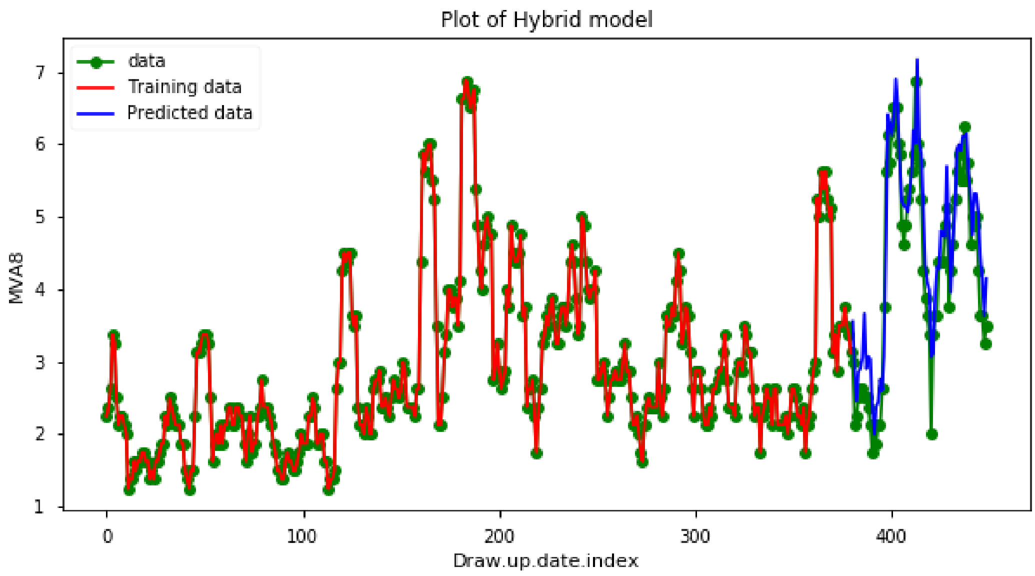


Figure 4.11: A plot of data (green curve), training (red curve) and predicted (blue) set of hybrid AR&LSTM model.

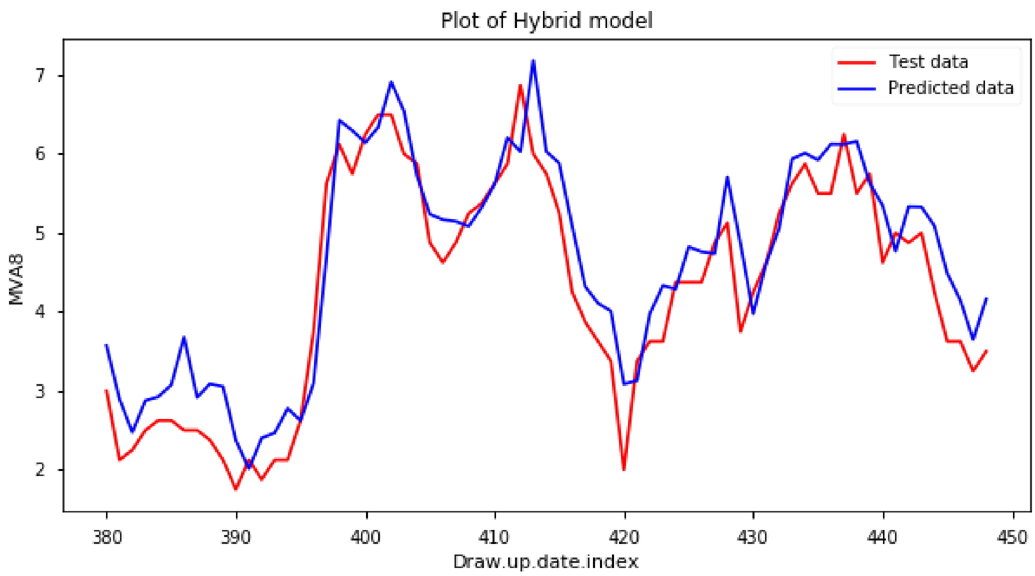


Figure 4.12: Plot of test data (red curve) and predicted (blue curve) set of hybrid AR&LSTM model.

The hybrid model gives the smallest error for prediction MAE=0.17 RMSE=0.15 than the prediction of the AR model and the LSTM model separately.

From all the results of the experiments in this project, it was found that Hybrid AR&LSTM model outperformed AR, ARIMA, and LSTM with MAE 0.17 and RMSE 0.15.

Performance prediction of models		
Models	MAE	RSME
Hybrid Model	0.17	0.15
LSTM	0.4	0.469
AR Model	0.47	0.61
ARIMA Model	1.6	0.63
MVA8 Model	1.98	2.96

This proposed Hybrid model improves on the moving average prediction on MAE and RMSE.

CONCLUSION

To avoid archaic methods used in Cameroon for delivering civil status documents, it is important to digitize the process in such a way that citizens could effortlessly request for certificates and services online, and via the digitized workflow system and help to save data for more research. Furthermore, to understand the dynamics of a population as well as improving the life quality of a population, it's very critical to conduct studies regularly about births, marriages, and deaths of habitants of a given location and keep records of them.

This project shows the importance of saving those data with an emphasis on the following:
Finding an efficient technique for birth prediction based on time series data.
Ability to anticipate the future number of birth and improve life quality. Four prediction models techniques were developed namely: AR, ARIMA, LSTM, and Hybrid AR&LSTM. Some of the key findings captured in this work are as follows:

- Forecasting time series need first of all an informative data (normal distribution), some transformation to eliminate high variances for noisy data is very important. This will help with stability when we will train our models.
- Building an efficient LSTM model, we need to run a Grid search for the best hyper-parameters (batch_size, epoch, units, and dropout). Grid search will give us the best model and very close to the one that we need and convergent.
- The effectiveness and accuracy of these methods in prediction performance were analyzed by using a comparison of MAE and RMSE. Based on those metrics, hybrid AR&LSTM have come to be the best model to predict the number of birth with a MAE=0.17 and RMSE=0.15.
- The model after hybrid AR&LSTM in terms of performance of prediction is the LSTM model with MAE=0.4 and RMSE=0.469.

(Perspectives: As future work, we advise to use statistical methods other than ARIMA and AR and then fusion them with LSTM and compare their performance with the hybrid AR&LSTM. We will also develop the parallel hybrid model of AR and LSTM and make a comparison with the sequential one that we have developed so far. We will also use larger data sets and more features that can help in having more training data points and give more interesting results, increasing efficiency of the recurrent neural network.)

5. Important links

https://github.com/Toadoum/Time-series-forecasting-AR-ARIMA-
https://github.com/Toadoum/LSTM-model-with-python
https://github.com/Toadoum/Hybrid-AR-LSTM-model-for-prediction-of-birth

Acknowledgments

This essay would not have been completed without the grace of God who gave me health and the will.

I dedicate this work to my father Ngarhoroum Sari and mother Allasra Odile. First of all, I would like to thank the founder of AIMS, Prof. Neil Geoffrey Turok for creating this great pan-African network in which I have acquired a lot of scientific research resources. I warmly thank all the AIMS staff especially Prof.Dr. Mama Foupouagnigni, who allowed me to fulfill my studies at AIMS-Cameroon with welcome and harmony.

I sincerely thank the Academic Director, Professor Marco Andrea Garuti, for welcoming me to the AIMS community and giving me the chance to a year full of experience with all his contributions.

At the same time, I would also like to thank particularly my supervisor Dr. Marcellin ATEMKENG and Mr. Theophilus Anash Narh for agreeing to lead this work and help me work on this topic which allowed me to embark on a new field. Despite a busy schedule, they always found a moment for me. I would like to thank the Limbe City Council for providing a 6 months internship for my training and my supervisor Mme BONA ETE EWALISENE for her availability and support.

I offer my appreciation to all the AIMS Cameroon tutors who followed my studies with all their help for my success, especially to Rockefeller to whom I express all of my gratitude for his presence especially during the realization of this memory with many helpful conversations and comments after reading drafts of this essay.

Finally, I want to end by thanking from the bottom of my heart those I can count on every day during this year of study at AIMS, especially my dear friend, Angelina Hasina, my brothers, and my sisters, who have never ceased to bring me their moral, emotional and financial support; my friends and classmates for their spirit of collaboration and sharing, more especially to Momo Junior, Ntsafack Guy, Patricia Gaelle, Nessma Adil, Shadrack Kibet, Emmanuel, Glorie, Joel and Vianney who brought me help and encouragement to all those who in one way or another, contributed to the realization of this essay. Many thanks.

References

[1] https://www.sciencedirect.com/topics/computer-science/backpropagation, 2016.

[2] https://www.toppr.com/guides/business-mathematics-and-statistics/time-series-analysis/components-of-time-series/, April, 2019.

[3] Suzanne Lacasse Zhongqiang Liu Beibei Yang, Kunlong Yin. Time series analysis and long short-term memory neural network to predict landslide displacement. *Springer-Verlag GmbH Germany*, 2019.

[4] Avrim Blum. Machine learning theory, 2014.

[5] Jason Brownlee. *Overfitting and Underfitting With Machine Learning Algorithms*. August 12, 2019.

[6] Nikhil Buduma and Nicholas Locascio. Fundamentals of deep learning: Designing next-generation machine intelligence algorithms, 2017.

[7] M Taner Eskil K.K. Emre Aki, Tugba Erkoc. *Subset Selection for Tuning of Hyper-parameters in Artificial Neural Networks*.

[8] S. B. Kotsiantis. Supervised machine learning : a review of classification techniques, 2007.

[9] Web page. https://medium.com/@venkateshpnk22/an-introduction-to-machine-learning-theory-2b09a4748187, April 27, 2014.

[10] Web page. https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/, November 12, 2019.

[11] S. S Rao. A course in time series analysis, technical report. 2008.

[12] Giha Lee 1 Xuan-Hien Le, Hung Viet Ho and Sungho Jung. Application of long short-term memory (lstm) neural network for flood forecasting. 2019.

[13] Rongpeng Li Xianfu Chen Zhiming Liu Yuxiu Hua, Zhifeng Zhao and Honggang Zhang. Deep learning with long short-term memory for time series prediction. 2018.

[14] G. Peter Zhang. Time series forecasting using a hybrid arima and neural network model, 2003.