الجزائرية الديمقراطية الشعبية الجمهورية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

المدرسة العليا للإعلام الآلي - 08 ماي 1945 – بسيدي بلعباس

Ecole Supérieure en Informatique

-08 Mai 1945- Sidi Bel Abbes

Master Thesis

To obtain the master diploma

Branch: **Informatique**

Specialty: Information System and Web (SIW)

Theme

Time Series Forecasting using Machine Learning and Deep Learning techniques

---

Presented by **:**

- Mr Khiati Khaled Walid
- Mr Hamorune Youcef

Graduated in : **/09/2021** In front of the jury composed of :

- M/Mme/Mlle XXX	Président
- Dr KECHAR Mohamed	Encadreur
- M/Mme/Mlle XXX	Examinateur
- M/Mme/Mlle XXX	Examinateur
- M/Mme/Mlle XXX	Invité

Academic Year: 2020 / 2021

Acknowledgments

First and foremost, praise and gratitude to Allah Almighty for providing us with the patience, bravery, determination, and motivation to complete this work.

We like to express our heartfelt gratitude to our parents, relatives, and friends for their constant support and encouragement.

We would like to express our heartfelt appreciation to our superiors: ***Dr. KECHAR Mohamed****,* ***Dr. CHAIB Souleyman*** *and* ***Mm. RAFAA Houria*** *for their patient instruction, passionate support, and insightful critiques of our research work.*

We would also like to acknowledge the president of the jury as well as the members of the jury for their interest in our research by agreeing to examine it and enrich it with their proposals.

Abstract

Time series forecasting is the behavior of predicting the future after careful consideration and analysis of the past, due to the essential significance of this task in countless areas such as science, business and engineering. Preparing an acceptable model to fit and then forecast the series is a difficult process since each signal/series has unique features and dependence on foreign parameters that are difficult to capture in the model.

There are several Time Series forecasting methods available today, each needing adequate data preparation and analysis to produce a meaningful prediction. The purpose of this paper is to undertake a comparison research on the most widely used Time Series estimators in order to compare their performance on a wide range of series from various areas (economics, finance, meteorology, etc...) using machine learning and deep learning techniques. Some of the implemented models are automated, making hyper-parameter search a component of the model, allowing it to be utilized without any prior knowledge of the models or the datasets on which it will be applied.

Keywords: Time series, Forecasting, Machine learning, Deep learning.

Résumé

La prévision des séries temporelles est le comportement qui consiste à prédire l'avenir après avoir soigneusement examiné et analysé le passé, en raison de l'importance essentielle de cette tâche dans d'innombrables domaines tels que la science, les affaires et l'ingénierie. La préparation d'un modèle acceptable pour ajuster et ensuite prévoir les séries est un processus difficile car chaque signal/série a des caractéristiques uniques et dépend de paramètres étrangers qui sont difficiles à capturer dans le modèle.

Il existe aujourd'hui plusieurs méthodes de prévision des séries temporelles, chacune d'entre elles nécessitant une préparation et une analyse adéquates des données pour produire une prédiction significative. L'objectif de cet article est d'entreprendre une recherche comparative sur les estimateurs de séries temporelles les plus largement utilisés afin de comparer leurs performances sur un large éventail de séries provenant de divers domaines (économie, finance, météorologie, etc...) en utilisant des techniques d'apprentissage automatique et d'apprentissage profond. Certains des modèles implémentés sont automatisés, faisant de la recherche d'hyper-paramètres une composante du modèle, ce qui permet de l'utiliser sans aucune connaissance préalable des modèles ou des jeux de données sur lesquels elle sera appliquée.

Mot Clé: séries temporelles, prévisions, apprentissage automatique, apprentissage profond.

مُلخص

التنبؤ بالسلسلة الزمنية هو سلوك التنبؤ بالمستقبل بعد دراسة متأنية وتحليل الماضي ، بسبب الأهمية الأساسية لهذه المهمة في مجالات لا حصر لها مثل العلوم والأعمال والهندسة. إعداد نموذج مقبول ليناسب ثم توقعه السلسلة عملية صعبة لأن كل إشارة / سلسلة لها ميزات واعتماد فريد على المعلمات الأجنبية التي يصعب التقاطها في النموذج.

هناك العديد من طرق التنبؤ بالسلاسل الزمنية المتاحة اليوم ، كل منها يحتاج إلى إعداد وتحليل كافيين للبيانات لإنتاج تنبؤ ذي مغزى.

الغرض من هذه الورقة هو إجراء بحث مقارنة حول أكثر مقدرات السلاسل الزمنية استخدامًا في لمقارنة أدائهم في مجموعة واسعة من السلاسل من مختلف المجالات (الاقتصاد ، التمويل ، الأرصاد الجوية ، إلخ ...) باستخدام تقنيات التعلم الآلي والتعلم العميق. بعض النماذج المنفذة مؤتمتة ، مما يجعل البحث عن المعلمات المفرطة أحد مكونات النموذج ، مما يسمح باستخدامه دون أي معرفة مسبقة بالنماذج أو مجموعات البيانات التي سيتم تطبيقه عليها.

كلمات مفتاحية السلاسل الزمنية ، والتنبؤ ، والتعلم الآلي ، والتعلم العميق.

Contents

List of Figures

List of Tables

List of acronyms

- **AIC:** The Akaike Information Criteria
- **ANN:** Artificial Neural Network
- **ARIMA:** Auto Regressive Integrated Moving Average
- **ARMA:** Auto Regressive Moving Average
- **AR:** Auto Regressive
- **CNN:** Convolutional Neural Network
- **DL:** Deep Learning
- **GRU:** Gated Recurrent Unit
- **LSTM:** Long Short-Term Memory
- **MAE:** Mean Average Percentage Error
- **MAPE:** Mean Absolute Percentage Error
- **MA:** Moving Average
- **ML:** Machine Learning
- **MSE:** Mean Squared Error
- **NN:** Neural Network
- **RNN:** Recurrent Neural Network
- **SARIMA:** Seasonal Auto Regressive Integrated Moving Average

Part I

Introduction

Chapter 1

Introduction

1.1 Context

Time series modeling is a dynamic study topic that has grabbed the interest of the academic community during the last few decades. The primary goal of time series modeling is to gather, evaluate, and systematically research the past in order to build an acceptable model that represents the series' inherent structure. This model is then used to explain the Time Series and anticipate its future values, i.e. forecasting.

Every day businesses use time series forecasting for a wide variety of purposes such as forecasting daily stock prices, forecasting foreign currency exchange rates, forecasting unemployment rates; meteorologists use it to provide an estimate of wind speeds, daily maximum and minimum temperatures, and to approximate the precipitations.
All of these and many more activities demonstrate the importance of time series and having a solid forecast of the future, as it may be critical for businesses to prepare for anticipated spikes in sales and prepare for it or to avert catastrophe when watching meteorological data.

The purpose of this research is to develop time series forecasting methodologies based on machine learning and deep learning techniques. Then, we fully evaluate these techniques in terms of various features, presenting the benefits and drawbacks of each strategy.

1.2 Thesis outline

Our report is organized as follows:

Part I provides an **introduction** and explanation of the fundamental problem. Part II is a **background** section that offers an overview of the main ideas utilized in the next section. The background section addresses the Time series forecasting area briefly. Then, we provide various forecasting model principles. This section will also showcase machine learning and deep learning techniques, as well as certain detection algorithms and forecasting measures. Part III presents the **state of the art** in time series forecasting methodologies utilizing machine learning/Deep learning techniques, identifying their advantages and shortcomings.

We compare these interesting techniques in detail using a summary table. The comparison is based on several factors such as algorithm settings and produced outcomes. Finally, in Part IV, we wrap off our report by discussing our prospects and future projects.

Part II

Background

Introduction

Time series forecasting is a strategy for predicting future occurrences by examining previous patterns, with the premise that future trends would be similar to past trends. Forecasting is the process of predicting future values using models fitted to previous data. Time series forecasting is required for prediction issues with a time component since it gives a data-driven approach to effective and efficient planning.

All of these and many more activities demonstrate the necessity of time series and having a solid forecast of the future, since it may be critical for businesses to prepare for probable spikes/dips in sales and prepare for it or avert catastrophe while watching meteorological data.

In this section, we will provide some information about our topic. First, in Chapter 2, we will go through the fundamental principles of time series, which will be the major focus of our study. In Chapter 3, Chapter 4 and Chapter 5, we introduce the reader to machine learning, forecasting models, and deep learning techniques, providing an overview of basic concepts in these advanced disciplines.We then briefly cover some forecasting metrics in Chapter 6. Following that, we delve into time series forecasting methods, which will be mostly employed in this study.

Chapter 2

Basic Concepts of Time Series

2.1 What is a Time Series ?

A time series is a succession of data items that occur in **sequential order** over a **given time period** . This is in contrast to cross-sectional data, which represents a single moment in time.

A time series in investing follows the movement of selected data points, such as the price of a securities, over a given period of time, with data points collected at regular intervals [1]. There is no minimum or maximum time requirement, allowing the data to be obtained in a fashion that delivers the information required by the investor or analyst analyzing the activity.

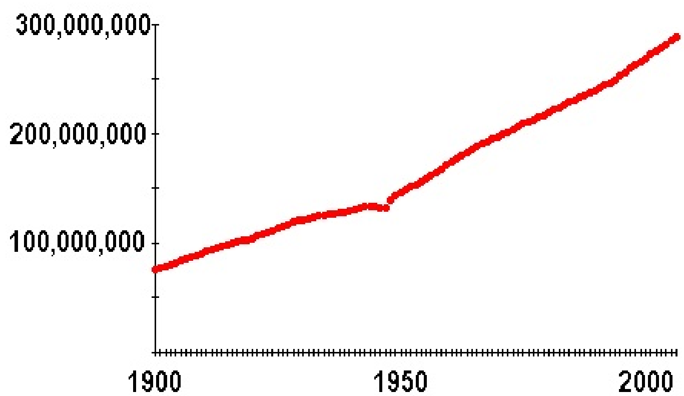


Figure 2.1: A time series graph of the population of the United States from the years 1900 to 2000 [1].

2.2 Types of Time Series

There are several classification criteria for time series; in this section, we will concentrate on **time dependency** and **stationary**.
Time dependence refers to the effect of previous values on newly observed values of the recorded variable/phenomenon. Time series are classified into two types [2]: long memory time series and short memory time series.

- **Long memory time series** are those with a slowly decreasing auto-correlation function; these time series reflect a process with slowly changing behavior. Long memory time series are typically seen in meteorological and geological data; for example, the development of the mean temperature on planets is an example of a long memory time series.
- **Short term time series** represent a process with a high turnover and have an auto-correlation function that decreases rapidly as we move away from the present, making the measure less helpful for the future. Financial data, such as stock prices, are a common example of this time series.

We have stationary and nonstationary time series for the second classification criterion: stationary. **Stationary time series** are processes with statistical characteristics (mean, variance) that are independent of time. Non-stationary time series are those that do not meet the aforementioned definition. These time series are particularly frequent in finance and retail. These time series can be difficult to anticipate without any sort of pre-processing, which is why we utilize a variety of approaches to stationarize these processes in order to obtain better forecasts.

2.3 Why do we do Time Series Analysis ?

Time series analysis is performed for two primary reasons [3]:

- Understanding the process's behavior by analyzing its historical records allows you to model and determine the major parameters that impact the time series and identify its components.
- Forecasting the future values of the series using an appropriate model trained on historical values.

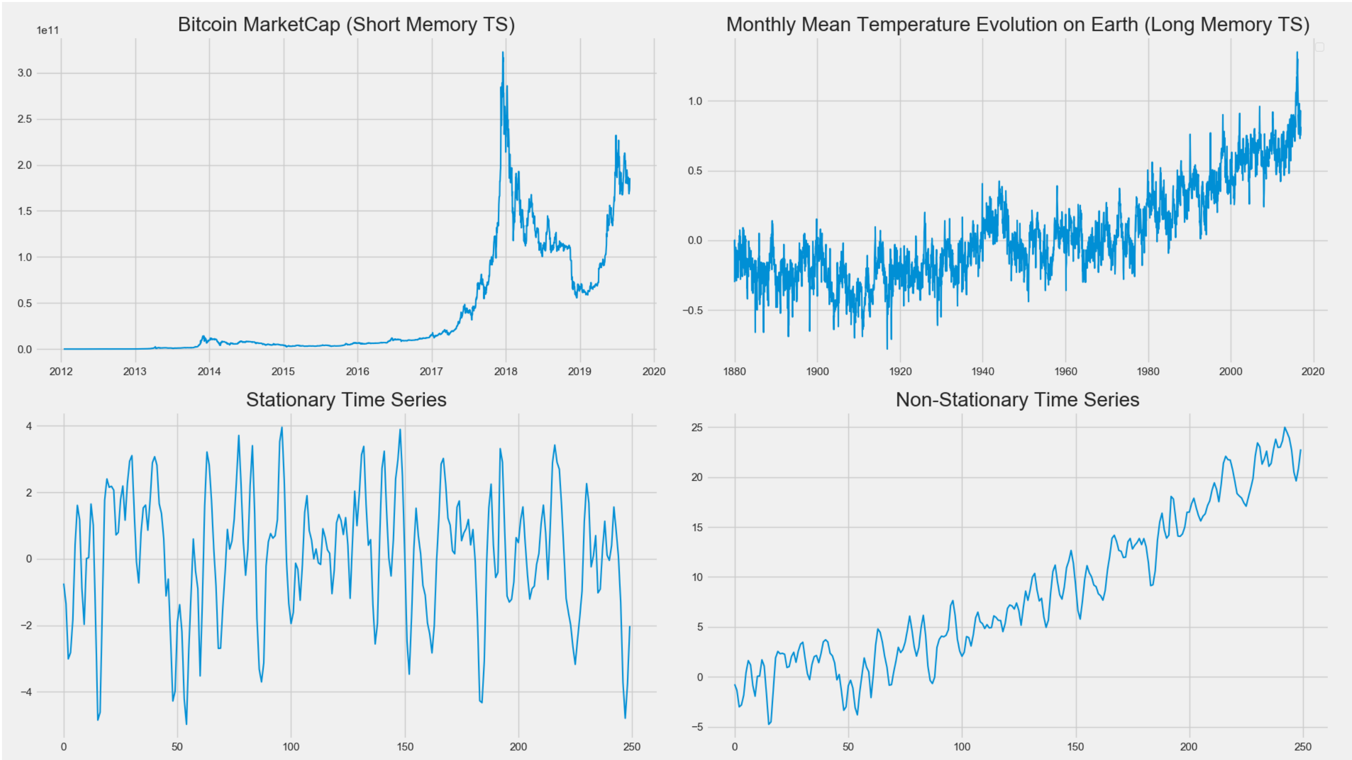


Figure 2.2: Example of Long/Short memory, Stationary/Non-Stationary Time Series [3].

2.4 Time Series Components

$$y(t) = T(t) + S(t) + C(t) + (t) \quad (2.1)$$

$$y(t) = T(t)S(t)C(t)\epsilon(t) \quad (2.2)$$

y(t) represents the measure collected at time step t, T(t) is the general trend of the series, S(t) depicts the seasonal element of the time series, C(t) is the cyclic component of the observation, and (t) reflects the irregular patterns within the series, residuals. Figure 2.3 depicts the sales of a retail store between 2013 and 2015; in this time series, we can see the aforementioned components (Figure 2.4). The trend differs significantly in this example, but the seasonal component may be deduced from the series' behavior and some common sense [1]. We may expect a rise in sales near the end of each year because these periods correlate to the holidays.

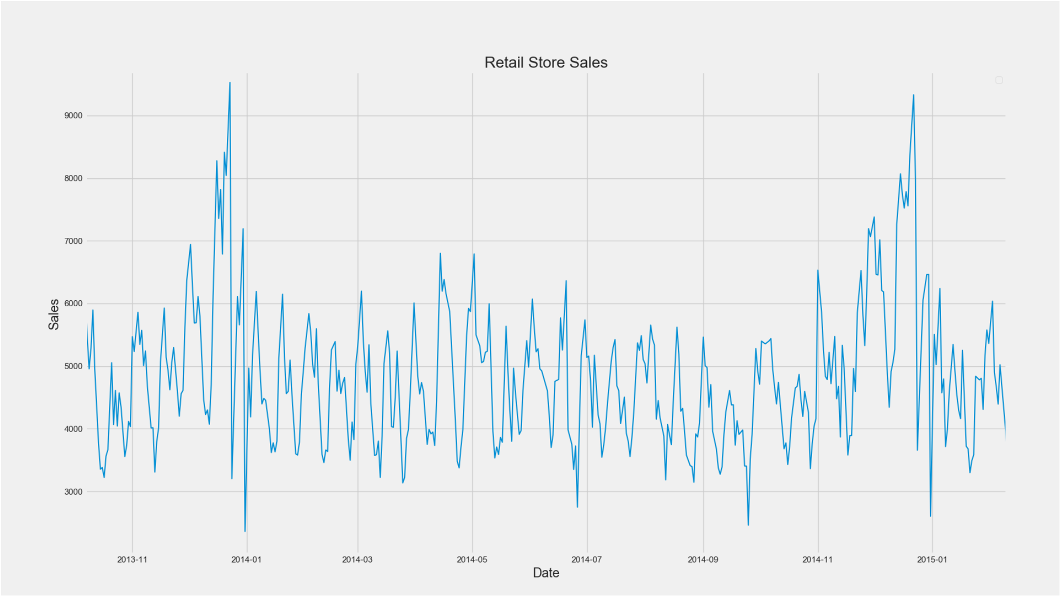


Figure 2.3: Sales of retail store [2].

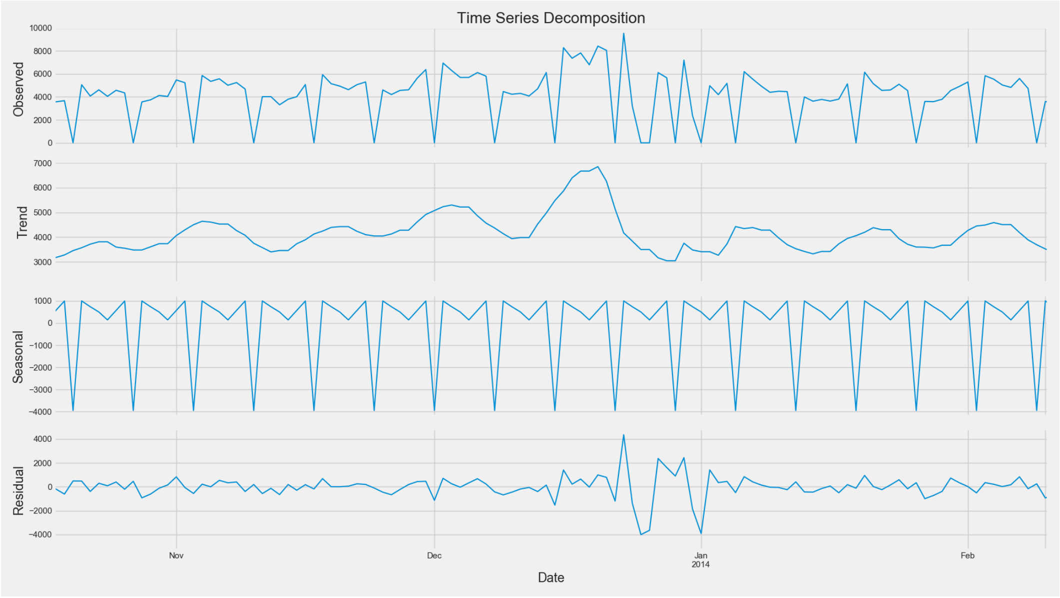


Figure 2.4: Decomposition of the sales of retail store into its components: Trend,Seasonal and Residual [2]

Chapter 3

Machine learning

3.1 What is Machine Learning?

3.1.1 Definition 1

Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed. —Arthur Samuel, 1959 [4].

3.1.2 Definition 2

A more engineering-oriented one:

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E. —Tom Mitchell, 1997 [5].

3.2 Types of Machine Learning Systems

There are so many different types of Machine Learning systems that it is useful to classify them in broad categories based on [6]:

- Whether or not they are trained with human supervision (supervised, unsupervised, semi-supervised, and Reinforcement Learning)
- Whether or not they can learn incrementally on the fly (online versus batch learning)
- Whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model, much like scientists do (instance-based versus model-based learning)

3.2.1 Supervised Learning

In supervised learning , we feed the algorithm with training examples and their desired outputs.

$$Data = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), .., (x^{(n)}, y^{(n)})\} \tag{3.1}$$

The goal is to learn a model/function that maps inputs to outputs for newly-observed data :

- When y is real, we talk about regression.
- When y s discrete, we talk about classification.

3.2.2 Unsupervised Learning

In unsupervised learning, no labels are given to the learning algorithm, leaving it on its own to discover hidden patterns in data.

$$Data = \{x^{(1)}, x^{(2)}, .., x^{(n)}\} \tag{3.2}$$

Unsupervised learning problems include:

- Clustering.
- Density estimation.
- Dimensionality reduction (e.g., ICA, PCA), etc.

3.2.3 Semi-supervised

Semi-supervised algorithms work on small amount of labeled data and large amount of unlabeled data.

These include:

- Self-training
- Generative models
- Graph-Based algorithms
- Multi-view algorithms

3.2.4 Reinforcement learning

The system learns by rewards and punishments given as a feedback to the program's actions in a dynamic environment. Reinforcement learning is usually described in terms of the interaction between some agent and its environment. The agent is the thing that is learning, and the environment is where it is learning, and what it is learning about [7].

Chapter 4

Forecasting Models

4.1 How to Forecast Time Series ?

Time series forecasting differs from typical regression assignments in that we must account for a crucial constraint: **order**. The chronological sequence of the data makes it more difficult for an estimator to develop an overall model that can be used for a long time [8], because patterns may arise for a short period of time and then fade, or the general distribution of the data may change. **The prediction of stock price** [9] is a significant example of this problem since the stock's behavior may change as a result of a new rule, which might affect the entire distribution of the data moving forward.

4.1.1 Time Series Forecasting

Forecasting is the primary reason we perform time series analysis; the basic concept is to try to forecast the future using previous observations. The model that best explains the data will be used to forecast the future based on prior data [8]. A forecasting model is a functional representation of a time series that will be used to anticipate the future. We have two primary variables in univariate time series:

- **Endogenous variables** are the series' previous values.
- **Exogenous variables** (or explicative variables) are external elements that can be associated to the value of the time series; these variables must be predictable (like calendar data) or we will be utilizing other time series to predict our own, as in multivariate time series forecasting.

We shall only use **endogenous factors** in this case study.

We can now begin projecting future values of the series after finding the best model that can fit the data. In this case, we have two options: we may forecast one step ahead or multiple steps ahead [8].

4.1.2 One step ahead

$$y_{t+1} = f(y_0; y_1; ...; y_{t-1}, y_t) \tag{4.1}$$

One step ahead strategy is very straight forward using the past values of the series the task will be to predict the next value of the signal.

4.1.3 Multi-Step

However, with the multi-step approach, things get a little more difficult since we have two options: predict all the values of the series up to y_{t+h} or forecast only the last value of the horizon y_{t+h}.

Iterative

$$\begin{aligned}
\hat{y}_{t+1} &= f(y_0, y_1, ..., y_t) \\
\hat{y}_{t+2} &= f(y_0, y_1, ..., y_t, \hat{y}_{t+1}) \\
&\vdots \\
\hat{y}_{t+h-1} &= f(y_0, y_1, ..., y_t, \hat{y}_{t+1}, .., \hat{y}_{t+h-2}) \\
\hat{y}_{t+h} &= f(y_0, y_1, ..., y_t, \hat{y}_{t+1}, .., \hat{y}_{t+h-2}, \hat{y}_{t+h-1})
\end{aligned}$$

The iterative technique is commonly employed by auto-regressive models, where each prediction is based on prior data, making it impossible for the model to forecast values that are far away in the future directly (as is the case with the AR model), as indicated by the equation above. The drawback of this approach is that it propagates the error made in previous forecasts into the future, potentially making the quality of long-term forecasts uncertain.

Direct

$$y_{t+h} = f(y_0; y_1; ...; y_t) \tag{4.2}$$

The direct approach forecasts a future value directly using past values gathered up to the present; this procedure is explained above in the equation 4.2.

4.1.4 Cross Validation for Time Series

The traditional time series forecasting approach is to train the model/estimator on a subset of the data (Train dataset), forecast on a certain sized horizon, and then compute the estimator's performance on the test dataset. The preceding method is insufficient for time series, particularly ones with a large number of datapoints. This is due to the fact that we cannot assume that the distribution of the series would remain constant throughout time [8]. Over the other hand, the fact that some external/exogenous factors have an influence that we cannot always model/use when making predictions makes forecasting on a reasonably broad horizon inaccurate.

So to prevent such issues, we use an approach called: **Walk Forward Validation**.

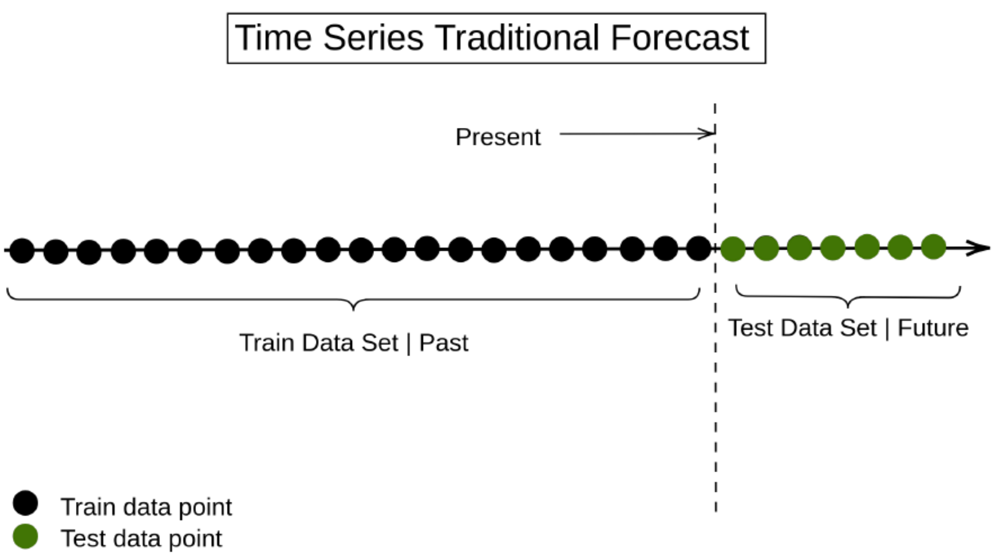


Figure 4.1: Cross Validation for Time Series. [3]

When predicting unknown values, this method considers the **granularity/frequency** of the Time Series. From a business/practical standpoint, this is the most advanced approach/implementation of the Time Series Forecasting issue since it can offer better forecasts (using the most information available up to the 'present'). We presume that our estimators will work well up to a particular sub-horizon, which can be deduced from the granularity of the dataset, and then retrain the model using the same train dataset and the observations that fall inside that sub-horizon that we've previously forecast [10]. This improves the estimators' expertise as they make predictions and addresses a common difficulty in the conventional technique, which is adjusting the model to changes in the overall trend of the series.

4.2 ARMA and its variants

4.2.1 AR(p)

Assumption: The time series is stationary

$$y_t = c + \varnothing_1 y_{t-1} + \varnothing_2 y_{t-2} + .. + \varnothing_p y_{t-p} \tag{4.3}$$

The auto-regressive model is the most basic time series model; it is based on the notion that every value of a time series may be predicted based on previous recorded values. This formulation mathematically indicates that the future value of the time series, y_t, will be the linear combination of the previous values with an intercept as shown in the equation 4.3, where $\varnothing$ is the weight of each observation [11]. The number of delays utilized in forecasting the value of y_t is determined by the parameter p of AR(p).

4.2.2 MA(q)

Assumption: The time series is stationary

$$y_t = \mu + \epsilon_t + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + ... + \theta_q\epsilon_{t-q} \tag{4.4}$$

$$y_t = \mu + \sum_{k=1}^{q} \theta_k\epsilon_{t-k} \tag{4.5}$$

$$\mu = \frac{1}{q}\sum_{t=1}^{q} y_{t-i} \tag{4.6}$$

The moving average model is identical to **the auto-regressive model**, except that the intercept is the average of the most recent q values, and instead of running the auto-regressive model on the observed records, we regress on the residuals (the difference between the average and the observed values). The MA component, as specified in the equations (4.4, 4.5, 4.6), is the linear combination of the most recent q residuals, with θ being the weight of each error correction [11]. The component ϵ is referred to as a random shock because it is expected to have a normal distribution with a zero mean and a constant variance δ^2.

4.2.3 ARMA(p,q)

Assumption: The time series is stationary

$$y_t = c + \epsilon_t + \sum_{i=1}^{p} \varnothing_i y_{t-i} + \sum_{i=1}^{q} \theta_i\epsilon_{t-i} \tag{4.7}$$

The ARMA(p,q) model combines the previously stated **AR(p)** and **MA(q)** models; technically, the model is just the sum of these models as given by the equation 4.7. However, the drawback of these models is that they presuppose that the time series under consideration are stationary, which is not the case in most real-world instances. As a result, we must ensure that any time series modeled with this function is stationary; to do this, we utilize the differencing operation to stationarize the series [11]. This procedure is addressed by the ARIMA model, which generalizes the **ARMA** model for **non-stationary** time series.

4.2.4 ARIMA(p,d,q)

The **ARIMA** model's most essential feature is its ability to **handle non-stationary time series** thanks to the **d** parameter, which controls the amount of differencing operations conducted on the datapoints. As a result, the **ARMA** model will be applied on the differentiated variable rather than to the original record [11]. Even though the **ARIMA** model can manage a wider range of time series (non-seasonal and non-stationary), it cannot handle **seasonal data** since it does not capture this component of the series in its mathematical formulation.

4.2.5 SARIMA(p; d; q)(P;D;Q)^s

The SARIMA model is a generalization of the ARIMA model, developed by **Box and Jenkins in 1970**, that is capable of handling the seasonality component of periodic time series. Box and Jenkins undertake a second differentiation on the seasonal component of the time series in their generalized proposition [11]. As a result, we add four more parameters to the model, which are $(p; d; q)(P; D; Q)^s$.

The **SARIMA** model as shown in Table 4.1 is an extension of all the previously stated models, since we can find a back to any estimator with the correct choice of parameters $(p; d; q)(P; D; Q)^s$:

$SARIMA(p; 0; 0)(0; 0; 0)^0$ describes an AR(p) model

$SARIMA(0; 0; q)(0; 0; 0)^0$ describes an MA(q) model

$SARIMA(p; 0; q)(0; 0; 0)^0$ describes an ARMA(p,q) estimator

$SARIMA(p; d; q)(0; 0; 0)^0$ describes an ARIMA(p,d,q) estimator

	Stationary	Non-Stationary	Seasonality
AR	✓	✗	✗
MA	✓	✗	✗
ARMA	✓	✗	✗
ARIMA	✓	✓	✗
SARMA	✓	✓	✓

Table 4.1: Methodology for Auto-regressive models [11]

4.2.6 Auto-correlation and Partial auto-correlation function

Determining the time dependency/correlation between recent and historical data is a crucial step in time series modeling and forecasting because it allows us to develop an appropriate model that best characterizes the records we have and generates strong forecasts [11].

4.2.7 The Akaike information criterion (AIC)

The Akaike information criterion (AIC) is an estimator of out-of-sample prediction error and thereby relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Thus, AIC provides a means for model selection [12].

In simpler terms, AIC is a single number score that may be used to identify which of several models is most likely to be the best fit for a given dataset. It estimates models in a relative manner, which means that AIC scores are only meaningful when compared to other AIC values for the same dataset. A lower AIC score is preferable.

$$AIC = -2\ln(L) + 2k$$

AlC equation, where L= likelihood and k = # **of parameters.**

In typical machine learning practice, AIC is most commonly employed when it is difficult to assess the model's performance on a test set (small data, or time series). AIC is very useful for time series analysis since the most relevant data in time series analysis is frequently the most recent, which is trapped in the validation and test sets. As a result, training on all data and using AIC can produce better model selection than traditional train/validation/test model selection methods.

A large sample size is often sufficient to approximate, but because using AIC frequently implies a small sample size, there is a sample-size adjusted formula called AICc that adds a correction term that converges to the AIC answer for large samples but gives a more accurate answer for smaller samples. As a general rule, it is best to always use AICc.

4.2.8 Significance of ACF and PACF

ACF is a (full) auto-correlation function that returns auto-correlation values for any series with lagged values. We generate an ACF plot by plotting these values along with the confidence band. Simply said, it describes how well the series' present value is linked to its historical values. Trend, seasonality, and residual can all be components of a time series. ACF takes into account all of these factors when determining correlations, which is why it is referred to be a 'complete auto-correlation plot.'

PACF is an abbreviation for partial auto-correlation function. Essentially, instead of finding correlations of the present with lags as ACF does, it finds correlations of the residuals (which remain after removing the effects which are already explained by the earlier lag(s)) with the next lag value, thus 'partial' rather than 'complete' because we remove previously found variations before finding the next correlation. So, if there is any hidden information in the residual that can be represented by the next lag, we may achieve a strong correlation and preserve that next lag as a feature during modeling [13].

The table below 4.2 can be used to help identify patterns, and what model conclusions we can make about those patterns.

Model	ACF	PACF
MA (q): moving average of order q	Cuts off after lag q	Dies Down
AR (p): autoregressive of order p	Dies Down	Cuts off after lag p
ARMA (p,q): mixed autoregressive-moving average of order (p, q)	Dies Down	Dies Down
ARIMA (p, d, q): Autoregressive Integrated Moving Average of order (p, d, q)	Dies Down	Dies Down

Table 4.2: Role of ACF and PACF in selecting models[14]

4.3 Facebook Prophet

Prophet is a Facebook open source project for time series research and forecasting. It provides a helpful API in Python and R for off-the-shelf forecasting for those who do not have much experience with time series modeling. Prophet combines a semi-automatic forecasting model with the option to modify the model depending on the user's past knowledge and experience [9]; this technique is referred to as "analyst in the loop" in the article that explains the model employed behind Prophet.
Prophet employs a three-component additive time series model:

1. A piece-wise linear or logistic growth function for simulating the series' trend component 4.8.
2. A periodic component represented by the 4.8 s(t) function.
3. A holiday component that must be manually inserted by the user in order to be enabled. This component adds external characteristics (other than holidays) to the time series, allowing the user to make better predictions and accurately predict unusual events. This component enables Prophet to simulate 'conditioned' seasonalities.
4. A forecasting error component that compensates for predicting mistakes.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{4.8}$$

$$g(t) = \frac{C(t)}{1 + \exp(-(k + \mathrm{a}(t)\mathrm{T}\delta)(t - (m + \mathrm{a}(t)\mathrm{T}\gamma)))} \tag{4.9}$$

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right)\right) \tag{4.10}$$

4.3.1 Useful add-ons to the traditional regression additive model

Prophet employs a modified trend detection model that employs a piece-wise linear component to adjust the level of the time series when the trend slope shifts, in order to avoid missing out on local patterns in favor of the overall pattern in the series, and uses the last portion of the trend as the trend for the future. Furthermore, the trend employs a logistic growth-like function to describe the signal's non-periodic development across time. This function contains a handy argument C(t) that allows you to limit the value of the observations to a maximum or a minimum [10].
For example, when forecasting real-time series such as the development of the number of users of a certain service, we know for a fact that this value cannot surpass the number of persons on Earth; or when projecting a stock price, we know for a fact that the price of a

stock cannot be lower than 0 [9]. These little model restrictions can go a long way toward ensuring that the model makes sense and provides explainable projections. Facebook's tool also accommodates multiplicative seasonality and sub-daily frequencies, although these two choices must be supplied to the model manually.

	Sub-daily Frequency		Seasonality		Capped trend	Change-point detection
	Automatic	Manual	Additive	Multiplicative		
Prophet	✗	✓	✓	✓	✓	✓

Table 4.3: A brief Summary of the capability of Prophet[10]

Chapter 5

Deep learning

5.1 Neural Network

Neural Networks are increasingly being utilized in time series forecasting, particularly for lengthy datasets. This is due to NN's ability to simulate complicated nonlinear models and self-adjust to data. NNs are general estimators that may be utilized for a variety of learning tasks.

5.1.1 Artificial Neural networks ANN

Artificial neural networks are a set of machine learning algorithms, and brain inspired techniques that simulate how the biological neuron works.

They were introduced by the neurophysiologist **Warren McCulloch** and the mathematician Walter Pitt In their paper : " A logical calculus of the ideas immanent in nervous activity", in 1943. [15].

5.1.2 From biological to artificial neurons

A neuron, or nerve cell, is an electrically excitable cell [16] that communicates with other cells via specialized links called synapses.

The human brain is composed of large number of interconnected neurons. A network of neurons is a combination of multiple nerve cells that can perform together a complex task like speech or image recognition as show in Figure 5.1.

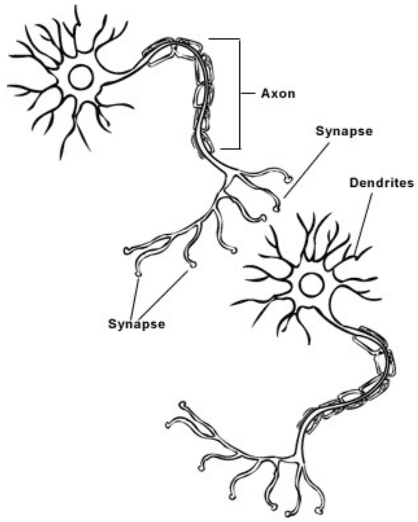


Figure 5.1: Tow connected biological neurons [17].

Biological neurons are fired based on the intensity of the entering signals. This process can be simulation by this activation function :

$$y = \sigma(\sum w_i x_i) \tag{5.1}$$

Where y is the output, σ is the activation function, w_i are the weight of inputs x_i. The activation function has many forms, we discuss them in more detail in the next section.

5.1.3 The activation function

The activation function is a mathematical function attached to each node in the network.It takes the weighted sum and bias of the inputs of that node and after defines its output (should be activated or not).

We use activation function to introduce non-linearity into the output of a neuron in order to perform complex tasks. A network without an activation function is nothing but a linear regression model [18]. Figure 5.2 shows presentation of activation function.

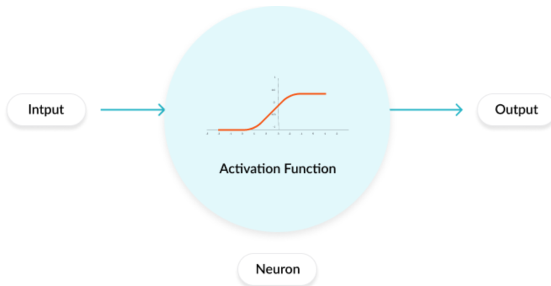


Figure 5.2: Activation function [19].

5.1.4 Types of Activation Functions

1-Sigmoid function

The Sigmoid activation function is sometimes referred to as the logistic function or squashing function in some literature[20]. It is given by the following formula :

$$\text{sigmoid } (x) = \frac{e^x}{1 + e^x} \tag{5.2}$$

It takes the input and map it to a value between 0 and 1. Figure 5.3 presents sigmoid activation function curve.

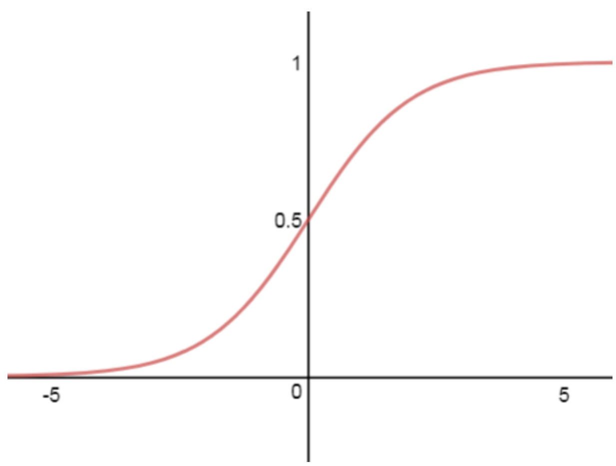


Figure 5.3: Sigmoid function [21].

The common problem with sigmoid activation is Vanishing gradient—for very high or very low values of X, there is almost no change to the prediction, causing a vanishing gradient problem. This can result in the network refusing to learn further, or being too slow to reach an accurate prediction[22].

2-Hyperbolic tangent function (Tanh)

The hyperbolic tangent is another common used activation function. It's defined as following:

It maps the input to a value in the range of [-1,+1] as illustrated in Figure 5.4.

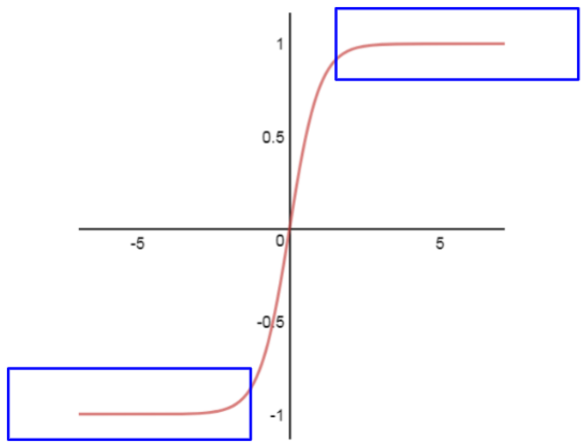


Figure 5.4: Tanh function [21].

It can also cause Vanishing gradient problem.

3-Rectified Linear Unit function (ReLu)

The rectified linear unit (ReLU) is a faster learning activation function proposed by Nair and Hinton in 2010 [23], which has proved to be the most successful and widely used function [24]. The Relu function is given by:

$$f(x) = \max(0, x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{5.3}$$

Figure 5.5 shows ReLu action function. This function rectifies the values of the inputs less than zero forcing them to zero and eliminating the vanishing gradient problem observed in the previous types of activation function.

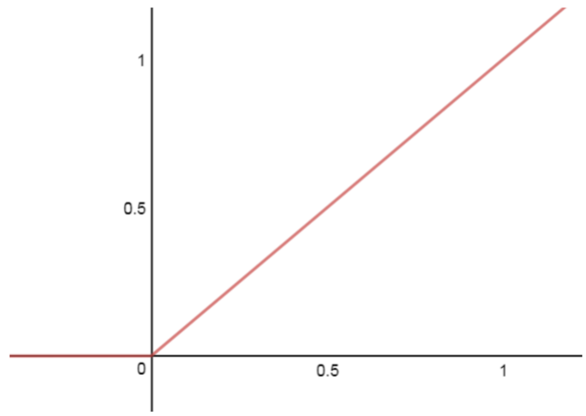


Figure 5.5: ReLU function [21].

however, the ReLU function can cause gradients to die when the inputs approach zero or are negative. In this case, the network can not carry out back-propagation, so it can not learn.

5.1.5 Examples of ANNs

Perceptron

The perceptron is a feed-forward network and the earliest type of the neural network, developed by Rosenblatt [25]. The perceptron was first designed to learn simple shapes and characters on images.

A single hidden-layer perceptron has its limitations: It can solve only linearly separable problems. The classic example is the XOR problem, which cannot be simulated with the single-layer perceptron. The multiple-layer perceptron (MLP), is the most used neural network. It can be used to approximate any continuous functions. A back-propagation algorithm is usually used in the training of MLP.

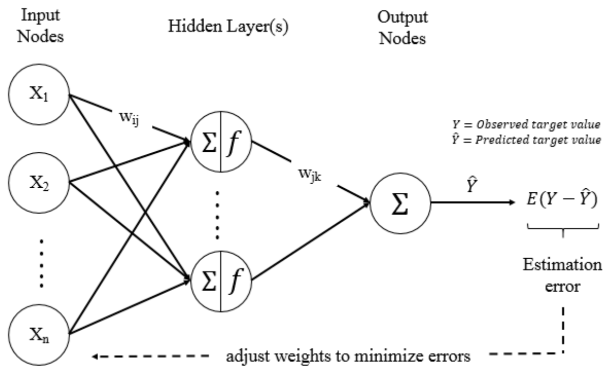


Figure 5.6: A typical multi-layer feed-forward back-propagation ANN [26]

At first,the input is propagated through the network and the output is calculated. Then the error between the calculated output and the correct output, called the cost function, is propagated backward from the output to the input to adjust the weights as shown in the figure 5.6. Mathematically the algorithm minimizes the cost function with a gradient descent method [27].

5.2 Deep Learning (DL)

5.2.1 What is deep learning (DL)

Deep learning (also known as deep structured learning or differential programming) is a special field of machine learning, putting the focus on the representation from the data, and adding successive learning layers to increase the meaningful representation of the input data [28].

Deep-learning methods are representation-learning methods with multiple levels of representation, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level [29]. An example of this representation learning in image processing : Objects are made of local parts, local parts are made of lines and corners, corners and lines are made of edges.

5.2.2 Deep learning applications

Deep Learning has enabled a huge progress in several domains.There are many different applications and this list below is in no way exhaustive :

- Speech recognition
- Object recognition in images
- Machine translation
- Signal processing
- Web mining
- Bioinformatics

5.2.3 Deep learning models

Several Deep Learning models have been proposed, we highlight the following models :

- Auto-encoders (Aes)
- Convolutional neural networks (CNNs).
- Recurrent neural networks (RNNs).
- Gated recurrent unit (GRU)

5.2.4 Recurrent neural networks (RNNs)

RNNs is a type of neural networks that take as their input not just the current example an input, but also the previous output/hidden states, they have an internal loop. Regarding the RNNs architecture, we can find two representations in the literature, one is plotted over a single cell and the second is presented as the sequential cells. Theses illustrations are given in Figure 5.7.

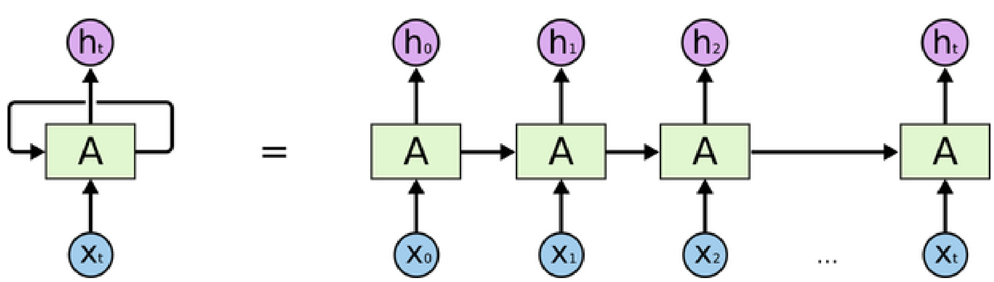


Figure 5.7: Basic architecture of Recurrent Neural Networks [30].

The above figure shows a copies of the RNN cells that are unfolded with different inputs at different time step. At each time step, values of the hidden weights and output are given by:

$$\begin{aligned}
&\mathbf{f}_t = \sigma_g\left(\mathbf{W}_f\mathbf{x}_t + \mathbf{V}_f\mathbf{h}_{t-1} + \mathbf{b}_f\right)\\
&\mathbf{i}_t = \sigma_g\left(\mathbf{W}_i\mathbf{x}_t + \mathbf{V}_i\mathbf{h}_{t-1} + \mathbf{b}_i\right)\\
&\mathbf{o}_t = \sigma_g\left(\mathbf{W}_o\mathbf{x}_t + \mathbf{V}_o\boldsymbol{h}_{t-1} + \mathbf{b}_o\right)\\
&\mathbf{c}_t = \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \sigma_c\left(\mathbf{W}_c\mathbf{x}_t + \mathbf{V}_c\mathbf{h}_{t-1} + \mathbf{b}_c\right)\\
&\mathbf{h}_t = \mathbf{o}_t \otimes \sigma_c\left(\mathbf{c}_t\right)
\end{aligned} \tag{5.4}$$

RNNs use the **Backpropagation through time (BPTT)** that allows calculating the error for each time step, which allows updating the weights.The BPTT algorithm are given as fallows [31]:

1. Present a sequence of time steps of input and output pairs to the network.
2. Unroll the network then calculate and accumulate errors across each time step.
3. Roll-up the network and update weights.
4. Repeat.

Issues with standard RNNs

The standard RNNs is one of the algorithms that have received the most success in deep learning over the past few years. Although, we mention two major issues with standard RNNs :

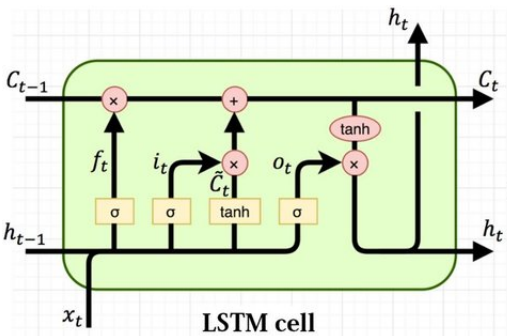
1. Vanishing gradients : very small gradients makes the model stop learning or take way too long to converge.
2. Exploding gradients : high gradients produces high weights, without much reason. Fortunately, this problem can be easily solved by truncating or squashing the gradients [32].

These issues are the main incentive behind the LSTM model that resolve the vanishing gradient problem.

Long Short-Term Memory Units (LSTMs)

LSTMs are an extension for RNNs.They are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn [33]. LSTMs enable RNNs to remember

their inputs over a long period of time by using gates. In LSTMs, we can find three gates input, forget and output gate. The input gate allows to read information from memory, the forget gate deletes the information from the cell state and the output gate writes the information into memory. The gates are activated using **sigmoid functions**.
Below in Figure 5.8 an illustration of structure of the LSTM cell and equations that describe the gates of an LSTM cell.



$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right)$$
$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f\right)$$
$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o\right)$$
$$\tilde{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g\right)$$
$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right)$$
$$h_t = \tanh(C_t) * o_t$$

Figure 5.8: LSTM cell visual representation [34].

To avoid gradient vanishing, LSTM continuously feeds error back to each of the gates until they learn to cut off the value.

LSTMs are used in many domains such as Anomaly detection in times series,Time series prediction,Grammar learning,Web page classification,etc.

5.2.5 Gated recurrent unit (GRU)

GRU is type of neural network and an other solution of problem of gradient vanishing. It is introduced by Cho et al in [35].It is considered as improved version of recurrent network. GRU uses two essential parameters to get an efficient result, namely, update gate and reset gate. A reset gate controls how much of the past information still want to remember whilst the update gate controls how much of old state will be passed in the future. Figure 5.9 illustrates the structure of GRU.

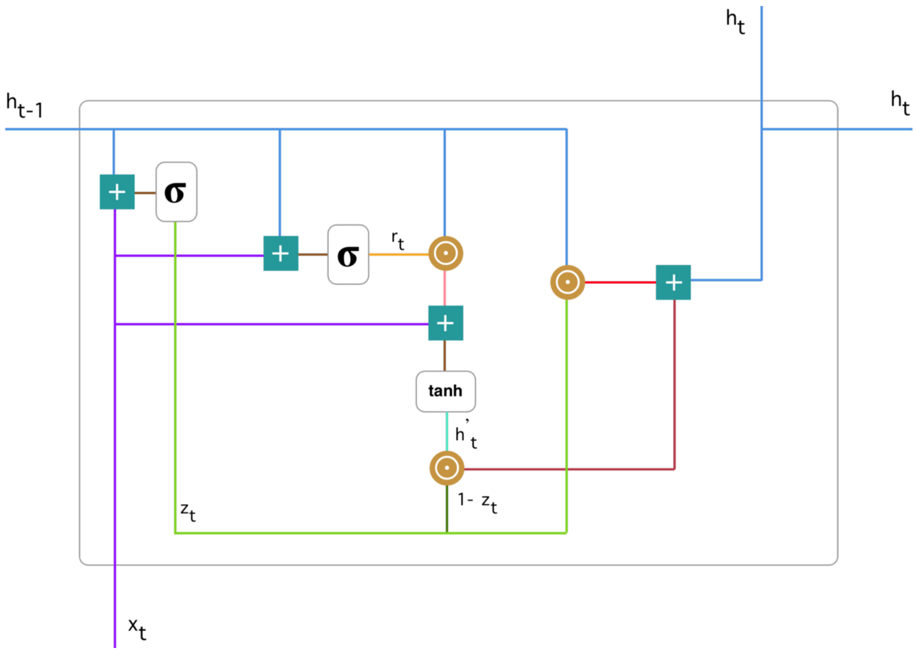


Figure 5.9: Gated Recurrent Unit structure [36].

The Update gate is calculated using the following formula :

$$z_t = \sigma \left(W^{(z)} x_t + U^{(z)} h_{t-1} \right) \tag{5.5}$$

The reset gate is calculated using the following formula :

$$r_t = \sigma \left(W^{(r)} x_t + U^{(r)} h_{t-1} \right) \tag{5.6}$$

where x,w represent the input, weight respectively while $h_-(t-1)$ keep the information for the previous unit and U(z) represents the weight of that unit and σ represents the sigmoid function.

Chapter 6

Forecast Metrics

The selection of forecast metrics is critical since these findings must be compared across various datasets and offer an accurate assessment of **forecast quality**. We will calculate numerous performance measures at the same time in the benchmarking tool, but we will only present the handful that can be compared across all datasets [2]. Some essential characteristics to consider for all forecast measures include comparability across datasets, sensitivity to data transformation, and sensitivity to outliers.

6.1 A Few Forecast Performance Metrics

6.1.1 The Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{N}|y_t - \hat{y}_t| \tag{6.1}$$

Properties:

- It computes the average absolute deviation between forecasted and original data.
- It's also known as **the Mean Absolute Deviation** (MAD).
- Extreme forecast mistakes are not punished by the MAE since they are affected by any data modification.
- For a flawless forecast, MAE should be as near to zero as feasible.

6.1.2 The Mean Absolute Percentage Error (MAPE)

$$\text{MAPE} = \frac{100}{N} \sum_{t=1}^{N} \left| \frac{y_t - \hat{y}_t}{y_t} \right| \tag{6.2}$$

Properties:

- This metric calculates the proportion of average absolute inaccuracy in forecasting.
- It is unaffected by measurement scale (e.g., **MinMaxScaling**), but is impacted by data transformation (e.g. Log Scaling).
- Extreme forecast mistakes are not penalized.
- Provide a metric that is unrelated to the values seen in the series.
- Ineffective in data sets with zeros in the original measurements.
- People with no statistical knowledge will find it simple to interpret.

This problem may be addressed by utilizing the updated MAPE, SMAPE formula [1]; however, the performance metric may diverge if the predicted and real values are both 0.

$$\text{SMAPE} = \frac{100}{N} \sum_{t=1}^{N} \frac{2\,|\hat{y}_t - y_t|}{|y_t| + |\hat{y}_t|} \tag{6.3}$$

6.1.3 The Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2 \tag{6.4}$$

Properties:

- It is a measure of the forecasted values' average squared deviation.
- It penalizes excessive mistakes in forecasting.
- MSE stresses that big individual mistakes have a significant impact on overall **forecast error** (outliers).
- It is affected by changes in size and data transformation.

- It is a difficult measure to evaluate since the value is dependent on the dataset and the values contained in the Time Series (noisy data or the presence of outliers can make assessing a model's performance difficult).
- This measure is not comparable to all datasets.

Another variate of the **MSE** is the **Root Mean Squared Error (RMSE)** which has the same properties as MSE and is formulated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(y_t - \hat{y}_t\right)^2} \tag{6.5}$$

6.1.4 The R2 Score

$$\text{R2} = 1 - \frac{\sum_{t=1}^{N}\left(y_t - \hat{y}_t\right)^2}{\sum_{t=1}^{N}\left(y_t - \bar{y}_t\right)^2} \tag{6.6}$$

Properties:

- Sensitive to extreme values (outliers).
- It is used to compare the goodness of t in a regression model to a baseline approach. R2 $=$ 0 indicates that the performance is comparable to the baseline approach (here the mean value), R2 $>$ 0 indicates that the performance is superior to the baseline, and R2 0 indicates that the performance is inferior to the baseline.
- It cannot tell you whether a regression model offers a sufficient t to your data (a good model can have a low R2 score while a biased model can have a high R2 score, this can be observed mostly Long Time Series with a huge horizon to forecast on) [1].
- Provides a measurement that is independent of the values seen in the series (helpful for evaluating a model's performance on multiple data sets with varying scales/ranges of values).

Conclusion

Preparing an acceptable model to fit and then forecast the series is a difficult process since each signal/series has unique features and dependence on exogenous parameters that are difficult to capture in the model. Academics, statisticians, and economists have suggested a plethora of studies and methods to enhance predicting accuracy over the years. As a consequence, several Time series models have been implemented/improved.

Part III

State of Art

Chapter 7

Time Series Forecasting approaches

7.1 Introduction

Forecasting is the process of utilizing models fit on past data to anticipate future observations. Based on what has already occurred, the future is forecasted or predicted. Time series introduces time order dependency between observations. Several approaches have been proposed for that prediction of the time series in different fields to cover its importance and impact in various applications.

This section discusses six techniques by providing a summary of how they function. These techniques are classified based on the learning method employed, which is either machine learning or deep learning. Following that, we discuss the benefits and drawbacks of each strategy. We offer a summary table for the evaluated machine learning/deep learning approaches at the end of the second portion.

7.2 Machine Learning Based approaches

7.2.1 Application of the ARIMA model on the COVID- 2019 epidemic dataset

Coronavirus disease 2019 (COVID-2019) has been identified as a worldwide danger, and numerous studies are being done to forecast the likely progression of this pandemic using various mathematical models. These mathematical models, which are based on different elements and studies, are susceptible to prejudice. Domenico Benvenuto et al presented in their paper [37] a simple econometric model that may be beneficial in predicting the spread of COVID-2019. they used the Auto Regressive Integrated Moving Average (ARIMA) model predicted using Johns Hopkins epidemiological data to forecast the epidemiological trajectory of COVID-2019 prevalence and incidence.

Data description

COVID-2019 daily prevalence data were obtained from the official website of Johns Hopkins University , and Excel 2019 was used to create a time-series database. The ARIMA model was used on a dataset of 22 number determinations. A descriptive analysis of the data was done to assess the frequency of new confirmed cases of COVID-2019 and to avoid bias.

Materials and methods

The ARIMA model consists of three components: the autoregressive (AR) model, the moving average (MA) model, and the seasonal autoregressive integrated moving average (SARIMA) model. The Augmented Dickey-Fuller (ADF) unit-root test can be used to determine if a time series is stationary. The main techniques for time series stabilization are log transformation and differences. To stabilize the term trend and periodicity, seasonal and nonseasonal deviations were employed.

The ARIMA model's parameters were calculated using an autocorrelation function (ACF) graph and a partial autocorrelation function (PACF) correlogram. ARIMA (1,0,4) was chosen as the best ARIMA model to assess the prevalence of COVID-2019. The statistical program Gretl2019d was used to do statistical analysis on the prevalence and incidence datasets, and the statistical significance threshold was set at 0.05. For the analytical technique, a previous research was used as a reference [38].

Results

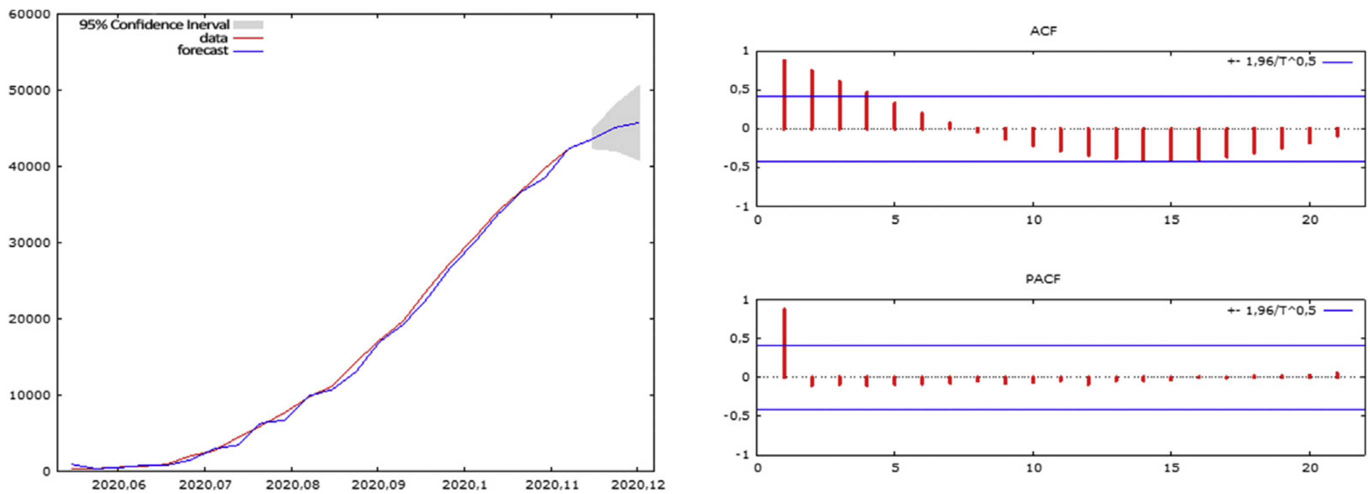


Figure 7.1: Correlogram and ARIMA forecast graph for the 2019-nCoV prevalence [37].

The ACF and PACF correlograms revealed that seasonality had no effect on the prevalence or incidence of COVID-2019. Table 7.1 shows the prediction of prevalence and incidence statistics with relative 95 percent confidence intervals.

	Date	Forecast	95% Confidence Interval
Prevalence	11/02/2020	43599.71	42347.53–44851.9
	12/02/2020	45151.45	42084.88–48218.02
Incidence	11/02/2020	2070.66	1305.23–2836.09
	12/02/2020	2418.47	1534.43–3302.51

Table 7.1: Forecast value for the 2 days after the analysis for the prevalence and for the incidence of the COVID-2019 [37].

Although additional data is required for a more thorough forecast, the virus's transmission appears to be slowing marginally. Furthermore, while the number of confirmed cases continues to rise, the incidence is somewhat reducing. If the virus does not evolve new mutations, the number of reported cases should level out (Fig 7.1 and Fig 7.2). The forecast and estimate obtained are impacted by the definition of the "case" and the mode of data gathering.

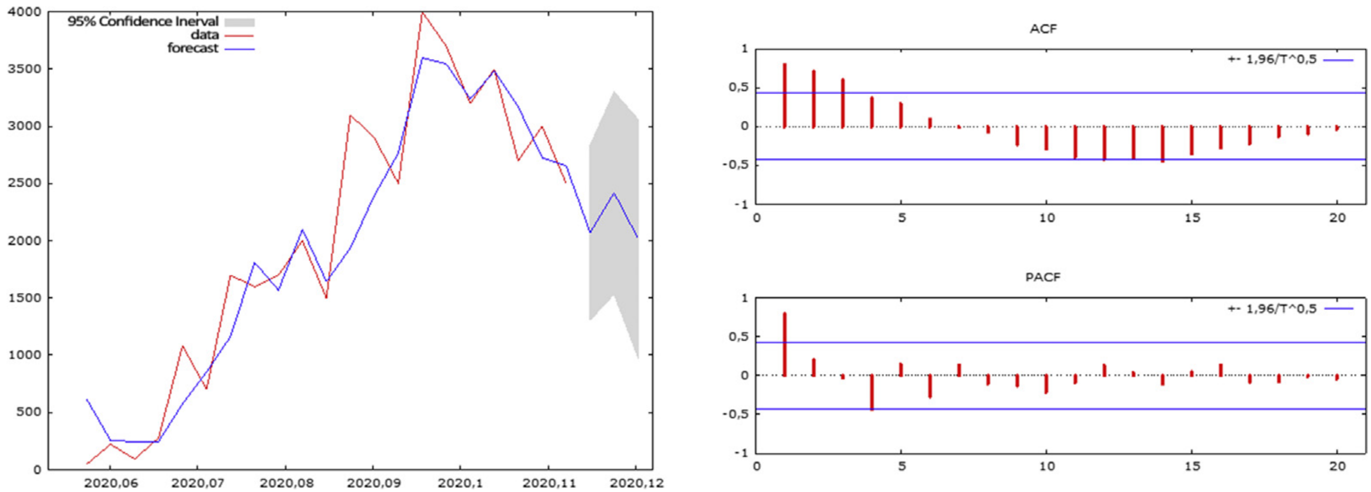


Figure 7.2: Correlogram and ARIMA forecast graph for the 2019-nCoV incidence [37].

7.2.2 Time Series Forecasting Stock Prices ARIMA Approach

Stock price prediction has always drawn attention because of the direct financial benefit as well as the related difficulty. Based on the examination of the literature, **Saptarsi Goswami et al** thought there was a need for a research that included sector-specific analyses and a diverse variety of equities. In this article [14], he investigated the efficacy of the Autoregressive Integrated Moving Average (ARIMA) model on fifty-six Indian equities from various sectors.

He picked the ARIMA model because of its simplicity and wide popularity. He also investigated the influence of several available prior period data on prediction accuracy.The Akaike information criterion was used to compare and parameterize the ARIMA model (AIC).

Data description

The author obtained historical data for fifty-six National Stock Exchange (NSE) businesses from seven sectors, eight companies in each sector, from the NSE India official website. He used twenty-three months of training data and forecasted the next month's data. He has also split the dataset into three distinct time periods: six months, twelve months, and eighteen months.

Materials and methods

Model selection , fitting and forecasting:

ARIMA models are commonly used to evaluate time series data in order to improve understanding and predictions. Initially, the suitable ARIMA model for the specific datasets must be found, and the parameters should have the smallest feasible values so that it can effectively evaluate the data and forecast accordingly. The Akaike Information Criteria (AIC) is a popular statistical model metric. It is used to assess the model's quality of fit. When comparing two or more models, the one with the lowest AIC is typically seen to be more accurate with real data.

MODEL	AICc
0,1,0	-2175.07
0,1,1	-2173.66
1,1,1	-2173.29
2,1,1	-2173.59
2,0,1	-2173.03
2,0,2	-2174.07
2,1,2	-2171.66
1,0,2	-2175.97
2,3,2	-2136.32

Table 7.2: AICc values of dataset of "Emami Limited" for different models [14].

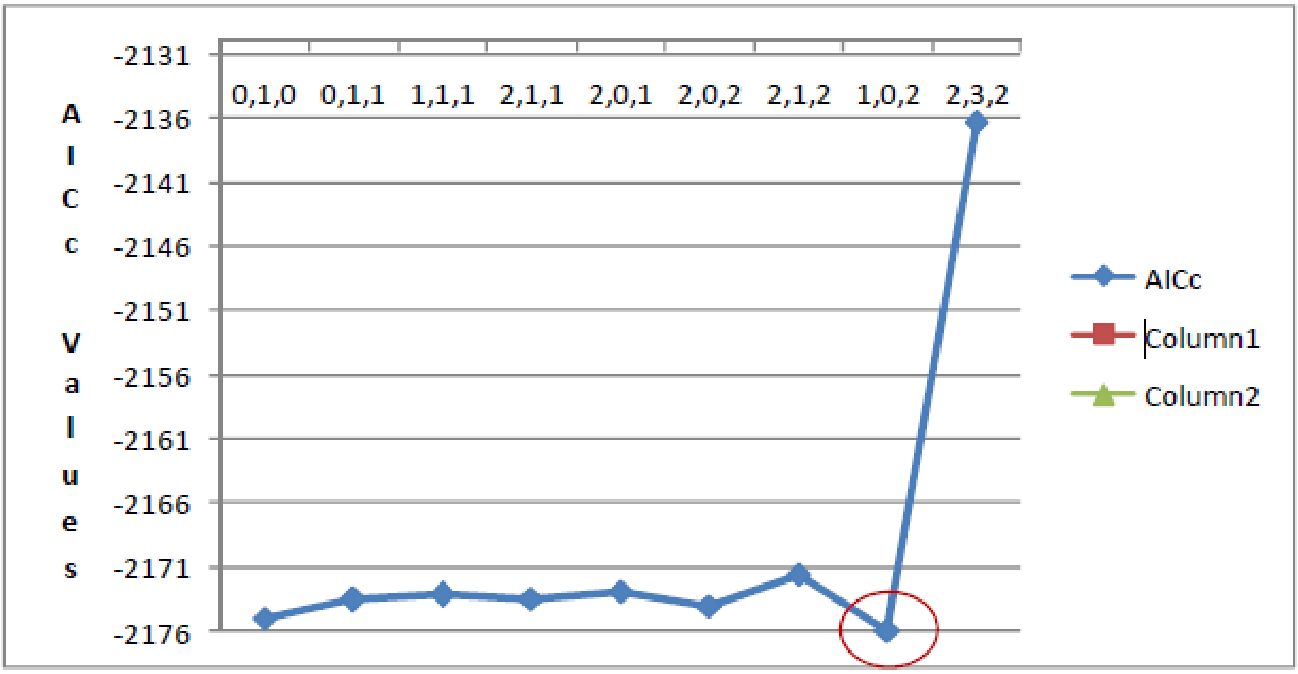


Figure 7.3: AICc values for different models [14].

Depending on AICc, model ARIMA(1,0,2) is selected for the above mentioned stock.

Measuring the accuracy of prediction:
The accuracy was expressed as a percentage. Actual data for 30 days that had previously been forecasted was obtained from the same source and compared to determine accuracy. The accuracy is calculated using the Mean Absolute Error (MAE) technique.

Finally, to obtain exact findings, evaluate the standard deviation of predicting accuracy for each sector. A smaller standard deviation implies that the data members are closer to the mean, whereas a greater standard deviation shows that the data are further deviated from the mean. For implementation, the high price of stocks is taken into account. R is used for all implementation tasks. The series are all stationary.

Results

Sector	Accuracy of prediction (in %) for twenty three months' training data	Accuracy of prediction (in %) for eighteen months' training data	Accuracy of prediction (in %) for twelve months' training data	Accuracy of prediction (in %) for six months' training data
1. Information Technology (IT)	91.06	93.77	93.79	94.03
2. Infrastructure	91.29	91.56	91.58	90.88
3. Bank	90.51	89.37	89.57	88.54
4. Automobile	87.89	85.32	85.78	85.91
5. Power	92.28	92.21	92.21	92.03
6. Fast Moving Consumer Goods (FMCG)	95.93	95.70	95.44	95.85
7. Steel	90.46	89.14	90.29	89.41

Table 7.3: Accuracy of prediction using ARIMA for seven sectors [14].

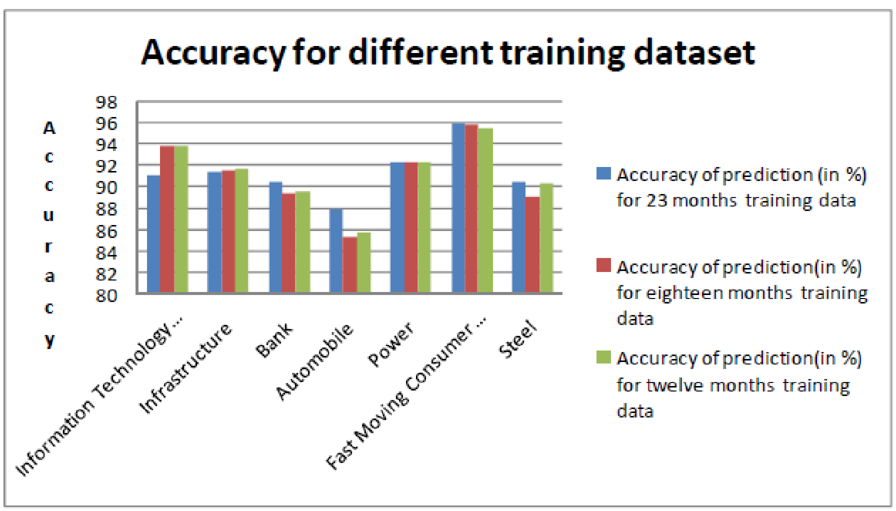


Figure 7.4: Accuracy for different training data sets [14].

In the next table, we show the standard deviation of accuracy of prediction for each sector.

Sector	Standard deviation of twenty three months' training data	Standard deviation of eighteen months' training data	Standard deviation of twelve months' training data	Standard deviation of six months' training data
Automobile	8.401289	16.99188	14.96798	17.47848
Banking	8.238547	15.67122	13.36196	15.77012
Infrastructure	5.073215	3.744663	4.009234	5.63259
Steel	7.619857	8.663484	8.441142	7.718103
Fast Moving Consumer Goods	3.123484	2.468907	2.240957	2.035353
Information Technology	6.310874	6.693157	6.341761	6.193422
Power	4.980284	4.873199	5.168635	5.683261

Table 7.4: Standard deviation of accuracy of forecasting for different sectors [14].

The author performed a research on 56 stocks from seven industries in this paper. All of the stocks chosen are traded on the National Stock Exchange (NSE). He has chosen data from the previous three months for the set empirical investigation. He assessed the ARIMA model's accuracy in forecasting stock prices. The best ARIMA model was chosen using AICc. In his research, he also altered the time period of prior or historic data and investigated the impact on accuracy.

The accuracy of the ARIMA model in predicting stock prices is greater than 85% across all sectors, indicating that ARIMA provides high prediction accuracy. When it comes to individual industries, predicting stocks in the FMCG sector using the ARIMA model yields the highest results in terms of accuracy. However, the accuracy of forecasts for the banking and car industries using the ARIMA model is lower when compared to other sectors.

The standard deviations of accuracy in forecasting of seven sectors show that the automobile, steel, and banking sectors have a high standard deviation, indicating that the values are distributed across a wide range, and there may be some companies for which the ARIMA model does not generate excellent results. The standard deviation for the Information Technology sector is neither too low nor too high, and we obtain above 90% accuracy in forecast for this industry. Stock prices of firms in the IT industry may fluctuate widely owing to changes in the value of the dollar and other reasons.

7.2.3 Time Series Forecasting using Improved ARIMA Approach

There has been a rise in interest in predicting time series databases in recent years. Time series forecasting has been demonstrated to be useful in making appropriate decisions in a variety of sectors. So far, a number of approaches have been presented to achieve the objective of prediction in various directions; in this paper [39] **Soheila Mehrmolaei et al** targeted two purposes. First, give a review by categorizing past important studies that explored time series data forecasting in various application domains. Second, for time series forecasting, present a unique technique to improving the ARIMA model by using a mean of estimation error.

Data description

In this work, three time series data sets are utilized to evaluate the performance of the suggested technique to basic ARIMA. The assessment results show that the proposed technique is suitable and successful for univariate time series forecasting.

- Monthly births time series.
- Annual immigration time series.
- Monthly number of cases of measles time series.

For this part of section we'll be showing only the first dataset from the three described and utilized from the paper.

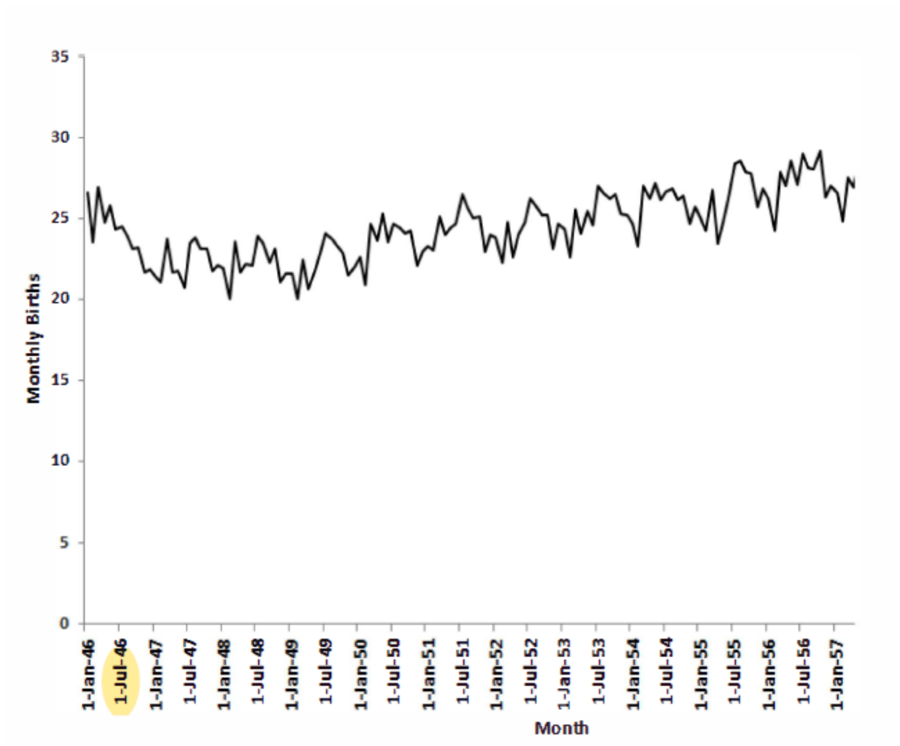


Figure 7.5: Monthly New York city births: Jan 1946- Dec 1959 [39].

Materials and methods

The concept of auto regressive integrated moving average, or ARlMA model, has been widely utilized for estimating future value in the problem of time series forecasting. It is known as the Box Jenkins model, and the future value of a variable is estimated in ARIMA by a linear combination of previous values and errors.
In this section, a strategy for improving the performance of the ARIMA model is proposed. Figure 7.6 depicts an exemplary diagram of the prediction method to the problem of time series forecasting. In suggested technique first divides gathered observations of unknown system into two data sets training and test. T_d, the training data set, is as follows:

$$T_D = \{x(i), i = 1, 2, \ldots\} \tag{7.1}$$

The training data set is then modeled with ARlMA to predict future values. Equation 2 is used to determine estimation error:

$$E(t) = X_T(t) - X_F(t) \tag{7.2}$$

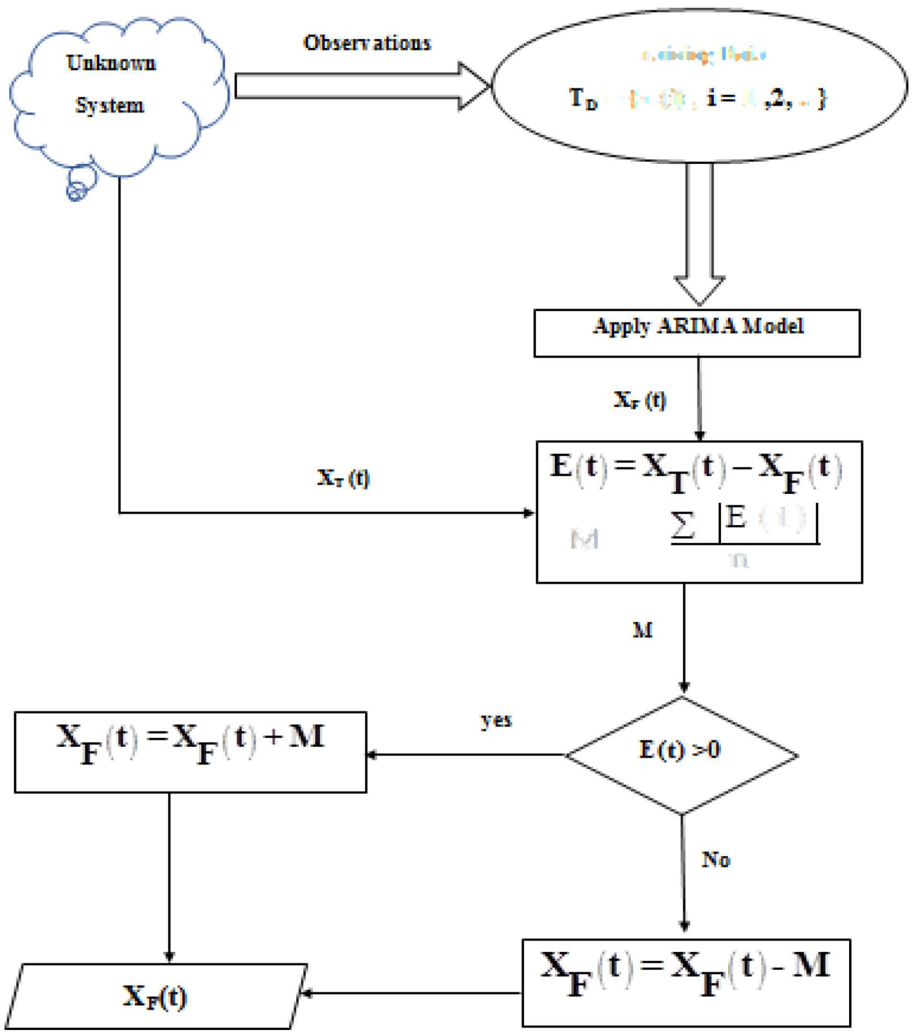


Figure 7.6: Proposed approach for time series forecasting [39].

Where $X_T(t)$ is the desired output and $X_F(t)$ is the expected output. Our proposed method is based on using the mean of estimating error in the forecasting process. Equation 3 calculates the mean of estimate error:

$$M = \frac{\sum |E(t)|}{n} \tag{7.3}$$

Where n is the number of intended output data points. The mean of estimating error is taken into account in the computation of projected values based on E(t) values. E(t) can have a positive or negative value.
Three performance measures are used for determining forecasting efficiency, namely mean square error (MSE), root mean square error (RMSE) and mean absolute error (MAE).

$$MSE = \frac{\sum_{\equiv}^{m} \left(X_{\mathrm{T}}(\mathrm{t}) - \mathrm{X_F}(\mathrm{t})\right)^2}{m} \tag{7.4}$$

$$RMSE = \sqrt{\frac{\sum_{t=1}^{m} \left(X_T(t) - X_F(t)\right)^2}{m}} \tag{7.5}$$

$$MAE = \frac{\sum_{t=1}^{m} \left|(\mathrm{X_T}(\mathrm{t}) - \mathrm{X_F}(\mathrm{t}))\right|}{m} \tag{7.6}$$

Results

The fundamental ARIMA model is used to forecast NYB (New York City births) time series data in this section. It is calculated with several parameter (p,d, q) values and the best model is chosen.
TABLE 7.5 displays the results of performance metrics using the ARlMA model.It shows that the minimal MSE and MAE values for the ARIMA are found (0, 1, 1).

Models	**Performance Measures**	
	MSE	**MAE**
ARIMA(2,1,0)	0.003	0.04
ARIMA(1,0,1)	0.015	0.11
ARIMA(0,1,1)	**0.002**	**0.04**
ARIMA(1,2,1)	0.043	0.16

Table 7.5: Performance measures using basic ARIMA for NYB data [39].

TABLE 7.6 shows the improvement achieved by the suggested technique for predicting NYB time series data.

Techniques	Performance measures		
	MSE	RMSE	MAE
Basic ARIMA	0.002	0.044	0.04
Proposed approach	0.0007	0.028	0.023

Table 7.6: Performance measures of the techniques [39].

According to the findings shown in the table above, the suggested technique outperforms the basic ARlMA model for the NYB time series data set utilized in this work. Figure 7.7 shows the percentage improvement gained by the suggested technique in terms of MSE, RMSE, and MAE values over the standard ARlMA model.

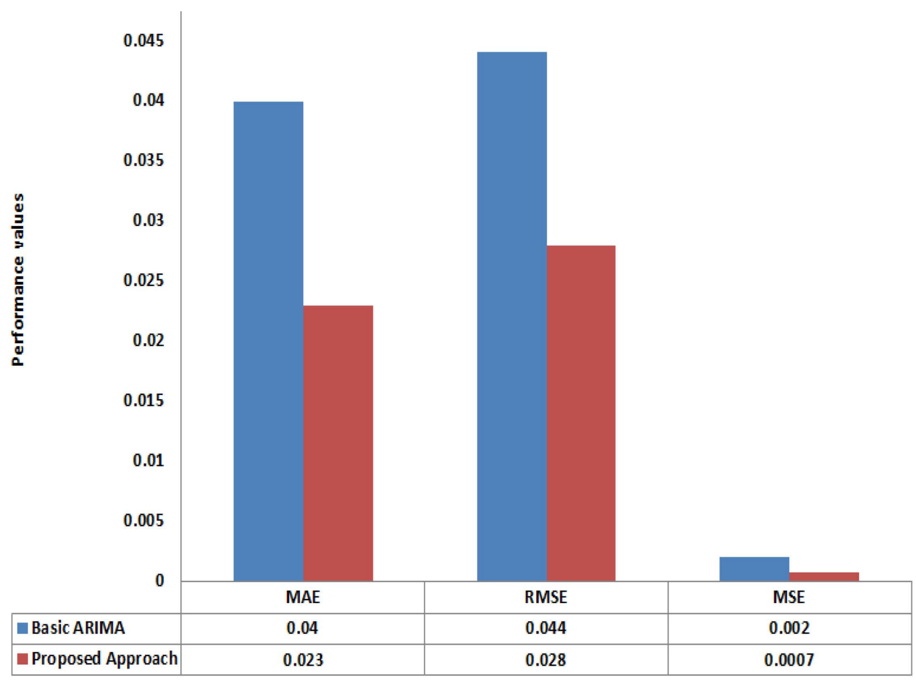


Figure 7.7: Percentage achieved improvement using proposed approach [39].

The result from Figure 7.7 clearly shows that, in terms of performance, the suggested technique outperforms the basic ARlMA model. The suggested approach's prediction responses are depicted in Figure 7.8.

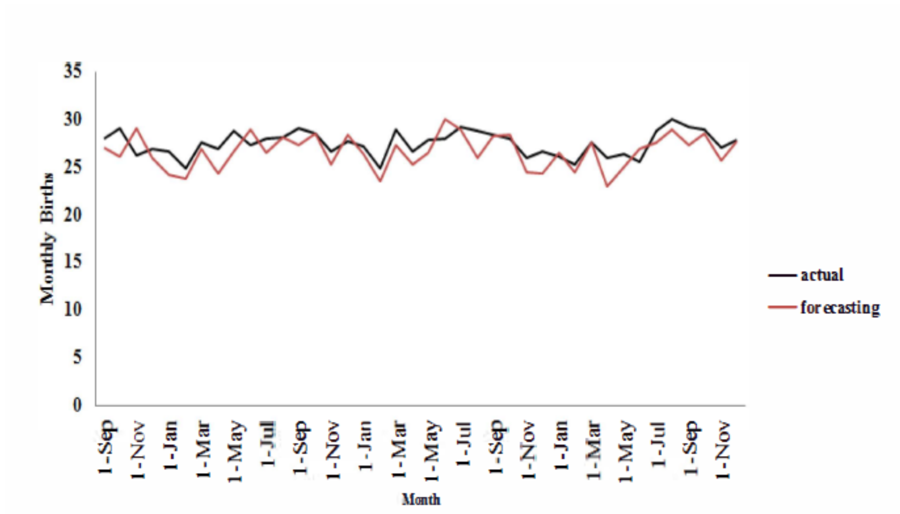


Figure 7.8: Prediction responses using the proposed approach [39].

Forecasting of time series data sets is an essential issue in many application fields such as weather, finance, medicine, and so on. This discipline employs a variety of approaches. Time series forecasting approaches are categorized and divided into two groups in this study depending on predicting duration. Short-term and long-term estimators have been named by several groups.

Furthermore, for time series forecasting in the basic ARlMA model, a technique based on applying a mean of estimation error is presented. As a consequence, the enhanced ARlMA model outperforms the original ARlMA model, and the suggested technique has acceptable accuracy.

7.2.4 Time Series Prediction Based on Facebook Prophet approach

Temperature forecasting is a time series analytic technique that forecasts the condition of the temperature at a certain area in the future. Nowadays, agriculture and industries are heavily reliant on temperature, therefore accurate forecasting is critical since temperature alerts may save lives and property. In this paper [40], **Sabai PhyuThe et al** used Prophet Forecasting Model to forecast Myitkyina's yearly temperature using historical (2010 to 2017) time series data.

Data description

The weather station with the fewest missing data points was chosen for this project. NOAA's 8-year average temperature data for Myintkyina, Myanmar, is used in this work. The weather data obtained is noisy, with a few missing numbers, and it is critical to handle this data. Every lost value is replaced with 0 during the data pre-processing stage. Following that, the data is ready for learning, as seen in Table 7.7.

DATE	TAVG	TMAX	TMIN
1/1/2010	69		50
1/2/2010	67		
1/3/2010	60	81	46
1/4/2010	62	81	45
1/6/2010	65		49
1/7/2010	66	85	48
1/8/2010	63		48
1/9/2010	65		47
1/10/2010	70	83	

Table 7.7: The collected average temperature [40].

Materials and methods

The forecasting algorithm was trained using daily weather data from 2010 to 2017, and it predicts the average temperature for the following three years. Both the projected and actual temperature values are compared on a graph in Figure 7.9 to determine the variance value, and the accuracy has improved considerably over the previous two years. The Root Mean Square Error (RMSE) was 5.7573 in 2012 and 2013 as a measurement of the model's prediction accuracy.

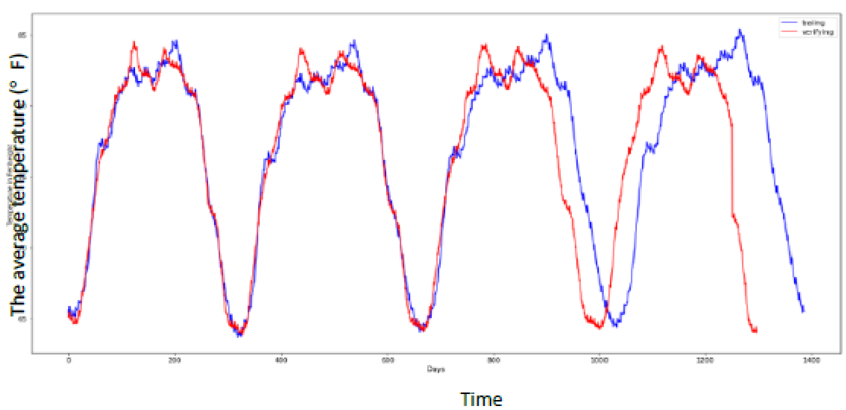


Figure 7.9: Forecasted value vs. real values graph of the average temperature [40].

The result with the appropriate RMSE has demonstrated significant improvements for weather forecasting. Furthermore, the prediction results indicated that the model is well-fitted to the historical data. It is demonstrated via the development of the proposed system that the Prophet Model is capable of producing satisfactory results for temperature prediction and may be utilized as an alternative to standard meteorological approaches.

7.3 Deep Learning Based approaches

7.3.1 Time Series ANN Approach for Weather Forecasting

Weather forecasting is the most challenging problem in the world. There are a variety of reasons for its observed values in meteorology, but it is also a common unbiased time series forecasting challenge in scientific study. Several techniques have been offered by various scientists. The goal of research is to improve prediction accuracy. In this paper [41] **Neeraj Kumar et al** contribute to the same by predicting two crucial meteorological parameters, maximum and lowest temperature, using an artificial neural network (ANN) and MATLAB simulation.

Data description

The Indian Meteorological Department provided weather data for the past hundred years for meteorological variables. The chosen data refers to two sets, namely average maximum and average minimum temperatures for each of the 12 months of the year. Figure 7.10 shows the overall structure of the network's inputs and outputs.

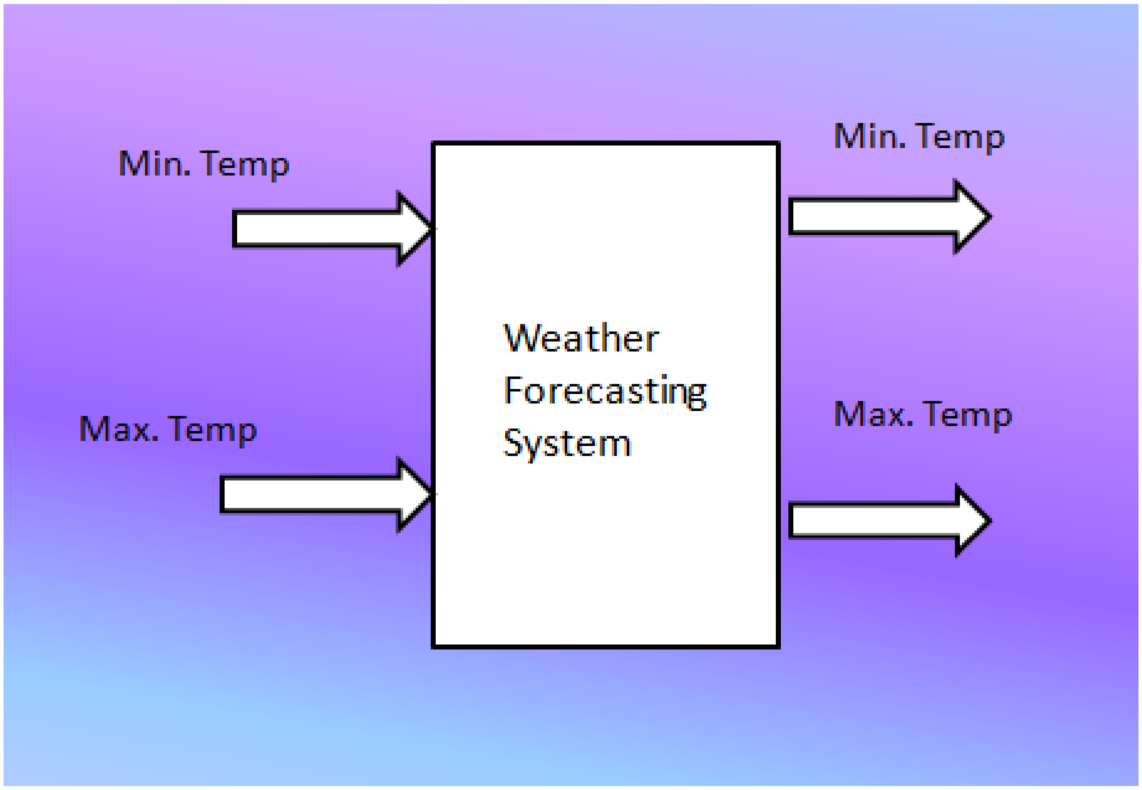


Figure 7.10: General Structure of inputs/outputs for the network [41].

Materials and methods

The worldwide data is separated into two groups, with the training group accounting for 60% of the data and the test group accounting for 40% of the data. Mean Squared Error (MSE) is a measure of the proposed network's error.
The obtained optimal network structure is shown in Table 7.8 below

Network Structure	
Number of Hidden neurons	3
Number of epochs	100
Activation function used in hidden layer	tan-sig
Activation function used in output layer	Pure linear

Table 7.8: The optimal Network Structure [41].

Results

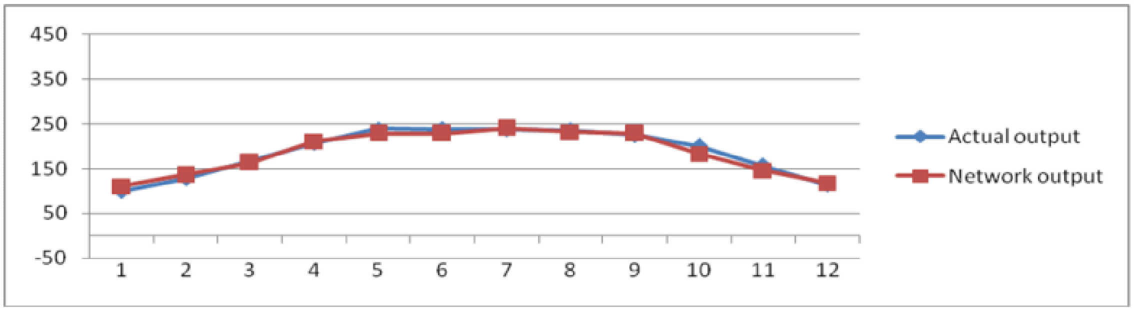


Figure 7.11: Comparison between actual and predicted minimum temperature values for year 2001 [41].

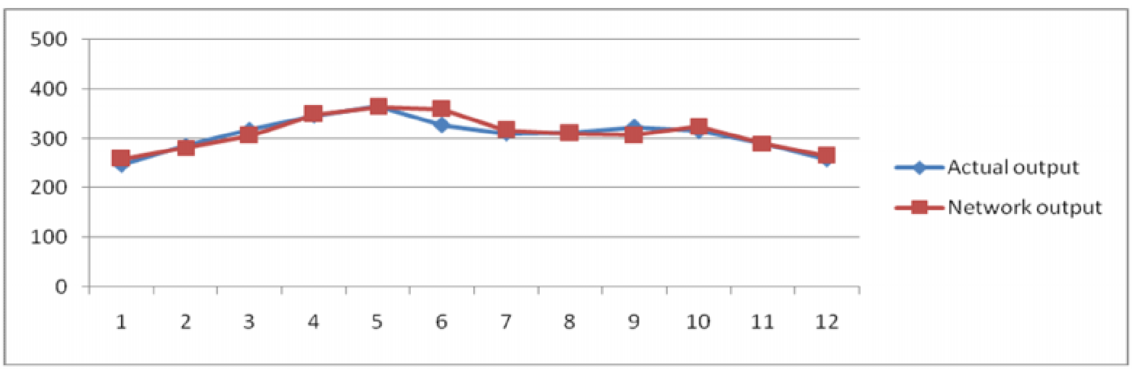


Figure 7.12: Comparison between actual and predicted maximum temperature values for year 2001 [41].

The results indicate that Multilayered Neural Networks might be a useful tool in weather prediction. Using historical data, this sort of network can properly give the mapping between input and output. Once the number of layers and units in each layer have been determined, the network's weights and thresholds must be adjusted to reduce the network's prediction error.

7.3.2 Predicting Computer Network Traffic Hybrid Approach

In This study [42], **Rishabh Madan et al** offers a strategy for forecasting computer network traffic based on the Discrete Wavelet Transform (DWT), the Auto Regressive Integrated Moving Averages (ARIMA) model, and the Recurrent Neural Network (RNN).

On a computer networking equipment that is linked to the internet, computer network traffic is sampled. Initially, the discrete wavelet transform is employed to divide the traffic data into non-linear (approximate) and linear (detailed) components.
Then, using inverse DWT, detailed and approximate components are rebuilt, and predictions are produced using Auto Regressive Moving Average (ARIMA) and Recurrent Neural Networks (RNN), respectively. Internet traffic is a time series that may be used to forecast future network traffic trends.
Numerous computer network administration activities rely significantly on network traffic information. This forecasting is extremely valuable for a variety of applications, including congestion control, anomaly detection, and bandwidth allocation.
This technique is simple to implement and computationally less costly, thus it may be easily applied in data centers to improve quality of service (QoS) while lowering costs.

Data description

	Time Series Datasets		
	Name of the time series	***Total Size***	***Time Interval***
1.	Daily-1	51	1 day
2.	Daily-2	69	1 day
3.	Hourly-1	1231	1 hour
4.	Hourly-2	1657	1 hour
5.	5min-1	14772	5 minutes
6.	5min-2	19888	5 minutes

Table 7.9: Details of time series datasets [42].

The studies were carried out using six time series produced by R.J Hyndman. The data in the time series were captured every hour, every day, and every 5 minutes. Table 7.9 describes the datasets in detail, and Figure 7.10 shows them.

The time series utilized is made up of internet traffic (in bits) collected by an ISP and corresponds to a transatlantic link. The tests were carried out with the help of Anaconda-navigator, which was pre-installed with different Python libraries. The program is free and open source.

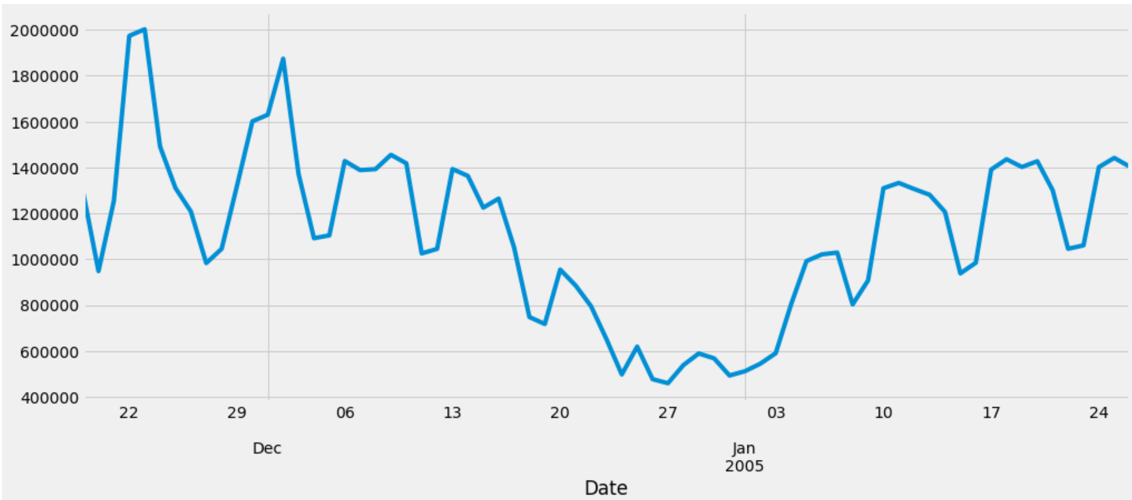


Table 7.10: Time plot of the time series Daily [42].

Materials and methods

This study presents a time series forecasting approach for predicting future computer network traffic that makes use of RNN and ARIMA. It simulates both linear and nonlinear network traffic time series architectures. After decomposition, the linear and nonlinear structures are recovered using the discrete wavelet transform (DWT). Real-world time series, such as data traffic, have both nonlinear and linear patterns. The proposed technique makes advantage of the decomposition power of DWT as well as the prediction capability of ARIMA and RNN for both linear and nonlinear systems. The experiment was carried out using samples of internet traffic time series collected from the DataMarket database.

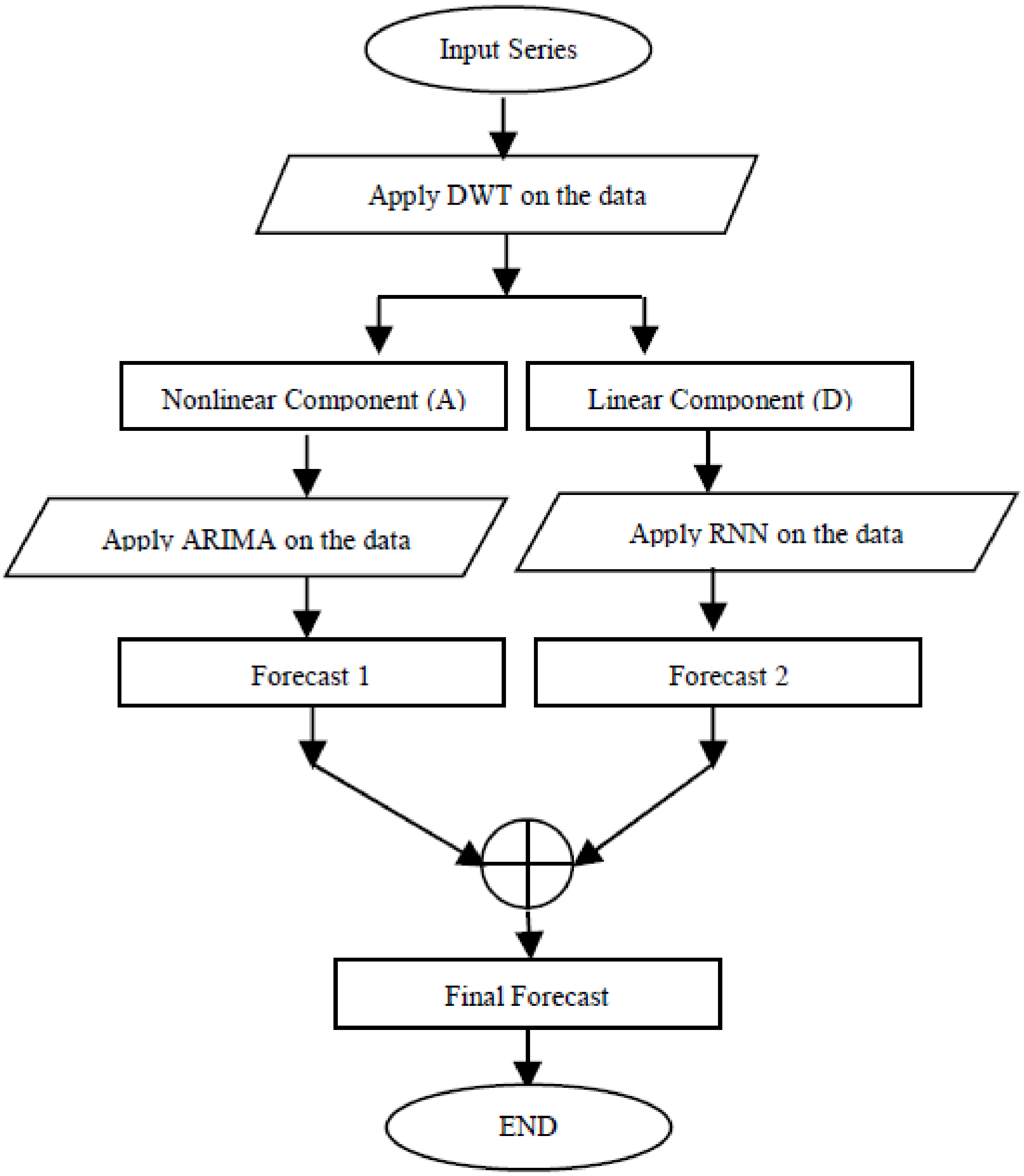


Figure 7.13: Flow Chart showing the proposed technique for network traffic prediction [42].

DWT is initially applied to time series data, resulting in the creation of low frequency and high frequency data components known as Detailed (D) and Approximate (A). Following this, these two components are rebuilt by using the inverse discrete wavelet transform on D and A. After generating the reconstructed components, the ARIMA model is applied to the D portion to create predictions. The RNN is then fitted to the A portion together with the residuals to provide another set of forecasts. Finally, these forecasts are combined component by component to get the final forecasts. The method is depicted in Figure 7.13.

Results

	NRMSE for different models			
			Proposed	
	Name of the time series	*RNN [14]*	*ARIMA*	*ARIMA + RNN*
1.	Daily-1	0.197	0.130	0.115
2.	Daily-2	0.116	0.240	0.191
3.	Hourly-1	0.041	0.030	0.022
4.	Hourly-2	0.027	0.050	0.24
5.	5min-1	0.016	0.020	0.012
6.	5min-2	0.009	0.010	0.008

Table 7.11: NRMSE for the six time series [42].

Following the decomposition of the time series with DWT, the linear and nonlinear parts were modelled with ARIMA and RNN, respectively. The forecast received from each technique was then put together to get the final forecast. Finally, predictions were obtained using the 'db2' wavelet. Table 7.11 summarizes the results, including the Normalized Root Mean Squared Error (NRMSE) for each dataset.

$$
\begin{aligned}
NRMSE &= (MSE)^{1/2} / \left(y_{\max} - y_{\min}\right) \\
MSE &= \sum_{i=1}^{n} e_i^2 / n
\end{aligned} \tag{7.7}
$$

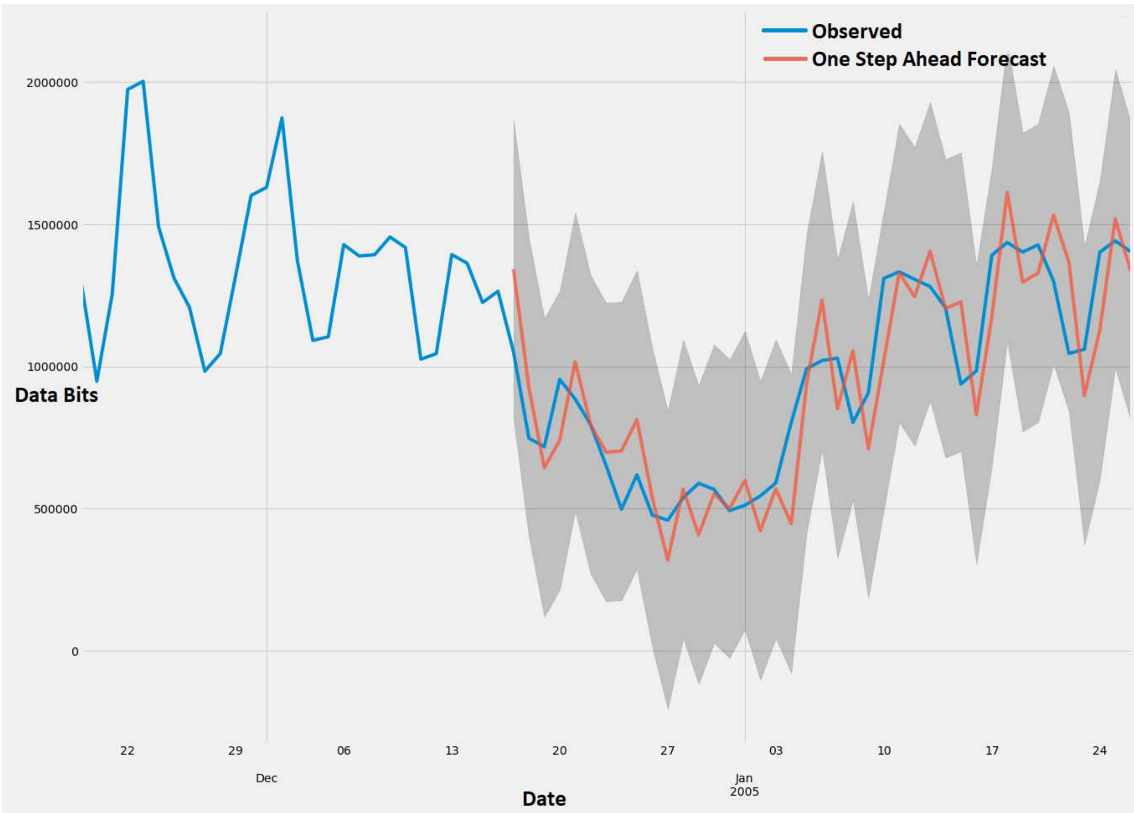


Figure 7.14: Time Series and its forecast Daily [42].

When it comes to operating a computer network and preventing traffic congestion, Internet traffic prediction is critical. In this article, the author suggested a time series forecasting approach for predicting internet traffic using previous data. Many forecasting approaches, such as ARIMA, are widely used in the literature to make forecasts, although they are best helpful for time series that are linear in form. Neural networks, on the other hand, such as RNN, are extremely effective in forecasting nonlinear time series. The proposed approach employs the Discrete Wavelet Transform, as well as a high pass and a low pass filter, to produce linear and nonlinear portions for the time series.

ARIMA and RNN are clearly outperformed by the suggested method (see Table 7.11). Because of the technique's simplicity, it can be readily implemented in data centers.

7.4 Discussion

This section compares time series forecasting techniques by providing the pros and disadvantages of each strategy through a quick examination of time series forecasting approaches.

An application of time series forecasting ARIMA model on the covid-2019 epidemic dataset was proposed in [37] in order to predict the evolution of this last mentioned, a simple econometric model was used for the prediction on the Johns Hopkins epidemiological data.
The Augmented Dickey-Fuller test(ADF) is a type of statistical test called a unit root test helps to determine whether the time series is stationary, ACF and PACF tests estimated the best ARIMA model , which concluded that the dataset is not influenced by the seasonality. Although additional data is required for a more thorough forecast, the virus's transmission appears to be slowing marginally. Furthermore, the authors did not mention whether or not they used any sort of metrics to measure the model efficiency.

The research in [14], the Autoregressive Integrated Moving Average (ARIMA) model was picked to predict Stock prices due to its simplicity and wide popularity, the work primarily concentrated on the degree of accuracy in projecting stock prices for various sectors, which will assist investors in understanding the market and deciding whether or not to engage in the stock market, various period data was taken to measure its effect on the prediction accuracy. AIC is basically used to determine the best values for the model then both ACF and PACF were used to check whether the model selected is appropriate.
The researchers evaluated the standard deviation of predicting accuracy for each sector to get precise results, the accuracy is calculated using the Mean Absolute Error (MAE) technique. Even though the accuracy of the model in predicting stock prices was greater than 85% across all sectors indicating that ARIMA provides high prediction accuracy, the accuracy of forecasts for the banking and car industries is lower when compared to other sectors. The standard deviation is neither too low nor too high, and we obtain outstanding accuracy in forecast. However, the series are all stationary which is the main boost to the model fitting, the authors did not mention any usage of Log transformation or differencing in order to maintain the stationarity required.

The approach [39] suggested a new improved model of ARIMA by using a mean of estimation error, the author gave a review by categorizing past important studies that explored time series data forecasting in various domains and used three time series data (one mentioned) to assess the performance of the proposed approach. The suggested technique indeed outperformed the basic ARlMA model in terms of MSE, RMSE and MAE that was used to determine forecasting efficiency.Nevertheless, the approach violate the principles of the ARIMA model which are based on its simplicity and clarity, the ease of executing SARIMA algorithms made it approachable and familiar for the benefit of forecasting time series, applying more equations in the process of forecasting will add more complexity and analysis in order to maintain the right value out of the algorithm. Furthermore, the suggested approach is yet to be called better than the basic ARIMA model due to lack of usage in different fields and datasets.

In another work [40], Sabai PhyuThe et al used Prophet Forecasting Model to forecast Myitkyina's yearly temperature using historical (2010 to 2017) time series data. Nowadays, agriculture and industries are heavily reliant on temperature, therefore accurate forecasting is critical since temperature alerts may save lives and property. The forecasting algorithm was trained using daily weather data of 7 years and trained on the next 3, The Root Mean Square Error (RMSE) was used as a measurement for the model's prediction accuracy,the prediction results indicated that the model is well-fitted to the historical data. However, the Prophet Forecasting Model used wasn't fully detailed in order to understand the variety of the data used, due to its all set parameters, the model ease the process in a way that makes it difficult to acknowledge the performance of the last mentioned in any specific study field.

The approach [41] suggested temperature forecasting using an artificial neural network (ANN) and MATLAB simulation. Several techniques have been offered by various scientists concerning Weather forecasting,the chosen data refers to two sets, namely average maximum and average minimum temperatures, both separated into train and test groups,with MSE as a measure of the proposed network's error. By comparing real and forecasted temperature values, the forecasting validity was measured and shown promising results using the Multilayered Neural Network model in Weather forecasting. Once the number of layers and units in each layer have been determined, the network's weights and thresholds must be adjusted to reduce the network's prediction error which is the main functionality for a better prediction in the model.

Unlike previous approaches based on machine learning and deep learning, in [42] Rishabh Madan et al offered a strategy for forecasting computer network traffic based on the Discrete Wavelet Transform (DWT), the Auto Regressive Integrated Moving Averages (ARIMA) model, and the Recurrent Neural Network (RNN). Initially, the discrete wavelet transform is employed to divide the traffic data into non-linear (approximate) and linear (detailed) components and then predictions are produced using ARIMA for the detailed data and RNN for the approximate one. Numerous computer network administration activities rely significantly on network traffic information due to its extreme value in the field, the studies were carried out using six time series and took advantage of the decomposition power of DWT as well as the prediction capability of ARIMA and RNN for both linear and nonlinear systems. According to the Normalized Root Mean Squared Error (NRMSE), ARIMA and RNN standing alone are clearly outperformed by the suggested method taking advantage of the successful results and the outstanding performance of ARIMA in linear time series as well as the high accuracy of neural networks like RNN when it comes to nonlinear time series forecasting.

7.5 Comparative analysis

The comparison table 7.12 highlights the examination of the many approaches developed in the state of the art and carried out in accordance with the criteria of:

- Paper
- Algorithms used
- Type of algorithm
- Dataset used
- Parameters
- Results

Paper	**Type**	**Algorithm**	**Dataset**	**Parameters**	**Application area**	**Framework**	**Results**
Domenico Benvenuto et al [37]	Machine learning	ARIMA	Covid-19 (Monthly)	ARIMA(1,0,3)	Health	/	/
Prapanna Mondal et al [14]	Machine learning	ARIMA	Stock price (Monthly)	ARIMA(1,0,2)	Finance	R language	MAE=98.26%
Soheila Mehrmolaei et al[39]	Machine learning	Improved ARIMA	N.Y City births (Monthly)	ARIMA(0,1,1)	Health	/	MAE=0.023 MSE=0.0007 RMSE=0.028
Zar Zar Oo et al [40]	Machine learning	Prophet Facebook	Avg Temperature (Monthly)	Automated	Meteorology	/	RMSE=5.7573
Neeraj Kumar et al [41]	Deep learning	ANN	Weather Temp. (Monthly)	Table 7.8	Meteorology	MATLAB	MSE
Rishabh Madan et al [42]	Hybrid	ARIMA + RNN	Computer Network Traffic (Daily)	Automated	Technology	/	NRMSE=0.115

Table 7.12: Comparative table of the approaches

7.6 Conclusion

We have given several studies on time series forecasting in many fields of research in this section. We have concentrated our efforts on works that make use of machine learning and deep learning techniques. We discovered that the various techniques provided have proven to be successful and well-executed in forecasting future production and providing satisfying outcomes. However, each solution overlooked a criterion or had a flaw in terms of algorithms or data confirmation.

Part IV

Conclusion

Chapter 8

Conclusion and future work

Time series forecasting is a strategy for predicting future occurrences by examining previous patterns, with the premise that future trends would be similar to past trends. Forecasting is the process of predicting future values using models fitted to previous data. Time series forecasting is required for prediction issues with a time component since it gives a data-driven approach to effective and efficient planning.

The purpose of this study is to compare the performance of the most widely used time series estimators. This research was carried out on a number of distinct datasets of varying sizes and from a wide range of areas (financial, business, meteorological,etc...). Furthermore, forecast time was initially a barrier for stochastic model engineering since the grid search was computationally intensive. An analysis of the data reveals that certain candidates outperform the rest of the candidates on average, based on the forecasting criteria employed.

With the addition of exogenous variables, this research might pave the way for the development of an automated time series forecasting library. In addition, based on the findings of this study, we may propose a recommender system that employs the best predicting model depending on the features of the provided datasets (number of seasonal components, number of change-points, monotonicity of the series, etc...).

Bibliography

[1] Yacine Ben Baccar et al. "Comparative Study on Time Series Forecasting." In: (2019).

[2] Jae Hyuk Han. "Comparing Models for Time Series Analysis." In: (2018).

[3] Chris Chatfield. *Time-series forecasting*. CRC press, 2000.

[4] Arthur L Samuel. "Some studies in machine learning using the game of checkers." In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229.

[5] Tom Mitchell. "Machine learning." In: (1997).

[6] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Mar. 13, 2017. 722 pp. ISBN: 978-1-4919-6224-4. Google Books: bRpYDgAAQBAJ.

[7] Stephen Marsland. *Machine Learning: An Algorithmic Perspective, Second Edition*. CRC Press, Oct. 8, 2014. 452 pp. ISBN: 978-1-4665-8333-7. Google Books: 6GvSBQAAQBAJ.

[8] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[9] RJ Hyndman and AB Koehler. "Another look at measures of forecast accuracy", Monash University." In: (2005).

[10] John C Chambers, Satinder K Mullick, and Donald D Smith. *How to choose the right forecasting technique*. Harvard University, Graduate School of Business Administration, 1971.

[11] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[12] Yosiyuki Sakamoto, Makio Ishiguro, and Genshiro Kitagawa. "Akaike information criterion statistics." In: *Dordrecht, The Netherlands: D. Reidel* 81.10.5555 (1986), p. 26853.

[13] Joao Henrique F Flores, Paulo Martins Engel, and Rafael C Pinto. "Autocorrelation and partial autocorrelation functions to improve neural networks models on univariate time series forecasting." In: *The 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2012, pp. 1–8.

[14] Prapanna Mondal, Labani Shit, and Saptarsi Goswami. "Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices." In: *International Journal of Computer Science, Engineering and Applications* 4.2 (2014), p. 13.

[15] Warren S. McCulloch and Walter Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity." In: *The bulletin of mathematical biophysics* 5.4 (Dec. 1, 1943), pp. 115–133. ISSN: 1522-9602. DOI: 10.1007/BF02478259. URL: https://doi.org/10.1007/BF02478259 (visited on 02/12/2020).

[16] P. A. Rutecki. "Neuronal Excitability: Voltage-Dependent Currents and Synaptic Transmission." In: *Journal of Clinical Neurophysiology: Official Publication of the American Electroencephalographic Society* 9.2 (Apr. 1992), pp. 195–211. ISSN: 0736-0258. pmid: 1375602.

[17] *Given That You Have One, Do You Know How It Works? | Louise Behiel*. URL: https://louisebehiel.com/given-that-you-have-one-do-you-know-how-it-works/ (visited on 06/06/2020).

[18] *Activation Functions in Neural Networks*. Jan. 29, 2018. URL: https://www.geeksforgeeks.org/activation-functions-neural-networks/ (visited on 03/10/2020).

[19] *7 Types of Activation Functions in Neural Networks: How to Choose?* URL: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/ (visited on 06/06/2020).

[20] *Quadratic Features and Deep Architectures for Chunking | Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. URL: https://dl.acm.org/doi/abs/10.5555/1620853.1620921 (visited on 03/10/2020).

[21] *Deep Learning Academy*. en. Library Catalog: www.deeplearning-academy.com. URL: http://www.deeplearning-academy.com (visited on 06/06/2020).

[22] *Activationfunction-1.Png (901×354)*. URL: https://missinglink.ai/wp-content/uploads/2018/11/activationfunction-1.png (visited on 02/12/2020).

[23] *Deep Learning | Nature*. URL: https://www.nature.com/articles/nature14539 (visited on 03/10/2020).

[24] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. "Searching for Activation Functions." In: (Oct. 27, 2017). arXiv: 1710.05941 [cs]. URL: http://arxiv.org/abs/1710.05941 (visited on 03/10/2020).

[25] Frank Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1962. 648 pp. Google Books: 7FhRAAAAMAAJ.

[26] *Figure 2. A typical multi-layer feed-forward back-propagation ANN.* en. Library Catalog: www.researchgate.net. URL: https://www.researchgate.net/figure/A-typical-multi-layer-feed-forward-back-propagation-ANN_fig1_318947023 (visited on 06/09/2020).

[27] David J. Livingstone. *Artificial Neural Networks: Methods and Applications.* Humana Press, Oct. 9, 2011. 254 pp. ISBN: 978-1-61737-738-9. Google Books: eVocYgEACAAJ.

[28] Sidahmed Benabderrahmane, Nedra Mellouli, and Myriam Lamolle. "On the Predictive Analysis of Behavioral Massive Job Data Using Embedded Clustering and Deep Recurrent Neural Networks." In: *Knowledge-Based Systems* 151 (July 1, 2018), pp. 95–113. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2018.03.025. URL: http://www.sciencedirect.com/science/article/pii/S0950705118301576 (visited on 02/12/2020).

[29] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning." In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539 (visited on 02/12/2020).

[30] *Basic Architecture of RNN and LSTM.* URL: http://pydeeplearning.weebly.com/1/post/2017/01/basic-architecture-of-rnn-and-lstm.html (visited on 06/07/2020).

[31] *Yanghaocsg/Keras_lstm.* URL: https://github.com/yanghaocsg/keras_lstm (visited on 02/12/2020).

[32] *Recurrent Neural Networks 101: Understanding the Basics of RNNs and LSTM.* URL: https://builtin.com/data-science/recurrent-neural-networks-and-lstm (visited on 02/12/2020).

[33] *Understanding LSTM Networks – Colah's Blog.* URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on 02/12/2020).

[34] *Structure of the LSTM Cell and Equations That Describe the Gates of an... | Download Scientific Diagram.* URL: https://www.researchgate.net/figure/Structure-of-the-LSTM-cell-and-equations-that-describe-the-gates-of-an-LSTM-cell_fig5_329362532 (visited on 06/07/2020).

[35] Kyunghyun Cho et al. "Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* EMNLP 2014. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. URL: https://www.aclweb.org/anthology/D14-1179 (visited on 05/09/2020).

[36] Simeon Kostadinov. *Understanding GRU Networks.* URL: https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be (visited on 06/07/2020).

[37] Domenico Benvenuto et al. “Application of the ARIMA model on the COVID-2019 epidemic dataset.” In: *Data in brief* 29 (2020), p. 105340.

[38] Ya-wen Wang, Zhong-zhou Shen, and Yu Jiang. “Comparison of ARIMA and GM (1, 1) models for prediction of hepatitis B in China.” In: *PloS one* 13.9 (2018), e0201987.

[39] Soheila Mehrmolaei and Mohammad Reza Keyvanpour. “Time series forecasting using improved ARIMA.” In: *2016 Artificial Intelligence and Robotics (IRANOPEN)*. IEEE. 2016, pp. 92–97.

[40] Zar Zar Oo and PHYU Sabai. “Time Series Prediction Based on Facebook Prophet: A Case Study, Temperature Forecasting in Myintkyina.” In: *International Journal of Applied Mathematics Electronics and Computers* 8.4 (2020), pp. 263–267.

[41] Neeraj Kumar and Govind Kumar Jha. “A time series ann approach for weather forecasting.” In: *Int J Control Theory Comput Model (IJCTCM)* 3.1 (2013), pp. 19–25.

[42] Rishabh Madan and Partha Sarathi Mangipudi. “Predicting computer network traffic: a time series forecasting approach using DWT, ARIMA and RNN.” In: *2018 Eleventh International Conference on Contemporary Computing (IC3)*. IEEE. 2018, pp. 1–5.