Analysis of Time Series Forecasting for Reactive Power of a Transformer

Sai Sanjeev Kumar Reddy Boppidi[1], Sai Annanya Sree Vedala[2]

[1]Student, Dept. of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Telangana, India
[2]Student, Dept. of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Telangana, India

---

Abstract - *This paper investigates the implementation of the models ARIMA [3] and Prophet to forecast the inconsistencies in reactive power. We propose the usage of two models for prediction and go on to analyse which works more efficiently with the help of performance metrics MSE and RMSE. We use the data received periodically from a transformer over a period of twelve months. Using the aforementioned data, we prioritise the ARIMA model over the Prophet model since it portrays highly promising results in comparison.*

Key Words: ARIMA, Prophet, Time Series Forecasting, Autoregression, Power Grid, SARIMA

1.INTRODUCTION

In an electrical power grid, the power load is not consistent. The inconsistency might result from various factors like the seasons, hour of the day, and climate. The perpetual variation in energy consumption necessitates the power grid to be flexible. Transformers play a vital role in the power grid. Hence, monitoring of the transformers is crucial to prevent the failure of the power grid. By monitoring the transformers continuously, the transformer's life can be improved, be utilized to their total potential, and reduce the maintenance cost. The rapidly progressing requirement for power supply adds substantial pressure on the asset owners to look after their more essential transformers. In this paper, we address this requirement with the help of two approaches of time series forecasting.

1.1 Time-Series Forecasting

A time series [2][9] is essentially a sequence where a metric is recorded over periodic time intervals. A time series can be yearly, quarterly, monthly, weekly, daily, hourly, and even seconds wise, depending on how often the data is obtained. An example of a yearly time series would be an annual budget and a second wise time series would be that of web traffic. Time series forecasting [1] is where we want to predict the future values that a time series is likely to take. Forecasting a time series, for instance, demand and sales becomes highly essential since it associates with a tremendous commercial value. In many companies belonging to sectors like manufacturing, finance, real estate, etc., it drives the fundamental business planning, procurement, and management activities. Any errors in the forecasts would be very expensive since they would affect the entire supply chain. It is thus highly important to get accurate predictions to not end up in losses.

Time series forecasting is divided into two umbrellas. These are Univariate Time Series Forecasting [4] and Multivariate Time Series Forecasting [5]. If only the previous values of the time series are utilized to predict the next values, it is Univariate Time Series Forecasting. If other predictors are used other than the previous value of the series or the series itself to make predictions, it is Multivariate Time Series Forecasting. Forecasting and prediction sometimes are used interchangeably, however, there is a notable distinction. In some fields, data at a specific future point in time can be referred to as forecasting, future data in general is referred to as prediction. Series forecasting is often used along with time series analysis. Time series analysis includes the development of models to attain an understanding of the data to understand the underlying trends.

1.2 ARIMA

ARIMA stands for 'Auto-Regressive Integrated Moving Average'. It is actually a model that forecasts a given time series based on the past values of its own, that is, its own forecast errors, so what is used to forecast future values is its equation itself. ARIMA models can be used for any time series that exhibits patterns and is not random white noise. This ARIMA acronym is descriptive as it captures the important aspects of the model. These are AR, I, and MA which stand for Autoregression, Integrated, and Moving Average. Autoregression [6] refers to a model in which the dependent relationship between some lagged observations and observation is utilized. Secondly, Integrated refers to the use of subtracting raw observations in order to make the time series seem stationary. Moving Averages [7] is a model that puts to use the dependence between a residual error and an observation from a similar model applied to the observations which are lagged. When we combine seasonality to ARIMA, it results in the Seasonal ARIMA model [13].
The aforementioned parameters of the ARIMA model can be described as follows:

- p(autoregressive order): Lag observations' number included in the model.

- d(differencing order): Count of the times the raw observations are subtracted.
- q(moving-average order): The size of the moving average window.

A linear regression [8] model generally includes a specified number and type of terms. The data is further prepared with the help of a degree of differencing to remove trend and seasonal structures that are unfavourable ie., make it stationary. 0 is often used for a parameter which should not be used as an element of the model. In this manner, ARIMA models can be configured to perform the function of an ARMA model, an AR, I, or MA model as well. When an ARIMA model is used for a time series forecasting, it is assumed that the process that generated the observations in the first place is an ARIMA process. This helps to motivate the requirement to assert the assumptions of the model in the raw observations and residual errors of forecasts.

1.3 Prophet

Prophet is an open-source tool developed by Facebook which utilizes a Bayesian-based curve fitting method to forecast the time series data. In Prophet, trends are fit with yearly, weekly, and daily seasonality, and holiday effects. It uses Fourier series and a weekly seasonal component modeled using dummy variables to include a yearly seasonal component. It is an entirely automated procedure that enables users to develop forecasts quickly and easily. The Prophet procedure provides users with many possibilities to tweak and modify forecasts. Prophet works most efficiently time-series with strong seasonal data and a considerable number of seasons of historical data

2. LITERATURE SURVEY

We investigated two of the most significant databases, the IEEE and ACM databases. We looked for the term "time series forecasting" and found similar forecasting methods in most articles and reports. In the process of our literature survey, we did not specify "time series forecasting for a smart grid" since we wished to perform more general based search and get a birds-eye view of the field of time series forecasting. We will use this survey and the documents as the base for our methodology and progress. We have decided to separate the method into three groups. Machine learning, ARIMA, and other techniques. In a report from 2012, Mehdi Khashei and Mehdi Bijari [15], proposed a hybrid of an Artificial Neural Network and ARIMA model to overcome some of the limitations of the ARIMA models. 3 different data sets with different characteristics were used to test the model. According to them, ARIMA models generally tend to have low accuracy for predicting non-linear data. They observed that their hybrid solution outperformed the results that came when the models were separately implemented. The Mean Absolute Error (MAE) and Mean Squared Error (MSE) for the short as well as the long-term forecasting showed better results with the hybrid model. Bangzhu Zhu and Yiming Wei [16] suggested a hybrid model that uses a Least Squared Vector Model with the ARIMA models, which is a neural network model used to forecast the prices of carbon in the EU. Three different kinds of hybrid models are proposed that were implemented on a dataset that is non-stationary and the results show a clear advantage of combining the models. Mehdi Khashei et al. [17], compare performance for the ANN, SARIMA, and Fuzzy logic models both individually and in combinations. Their results show that it is often better to combine different models than to just go with one model. Many other models in combination make up for the fact that SARIMA does not favor non-linear data. A set of previous data is required for the SARIMA and ARIMA models, which is not often available in the real world. Usually, the data available is less, ARIMA might not be the best use for such a forecast.

2.1 ARIMA Models Related Work

The field of time series forecasting has been used for multiple different tasks that require planning often due to restrictions in adaptability. ARIMA is a very popular algorithm for time series forecasting. ARIMA models were introduced in the 1950s initially but were popularised by George E. P. Box as well as Gwilym Jenkins. To get a clear overview of what ARIMA is, we must first break it down into smaller pieces. An Autoregressive model (AR) represents a random process. In the autoregressive model, the output depends on previous inputs and a stochastic term. AR models take past steps into consideration when calculating the next step. The issue with the AR model is that temporary or single shocks affect the whole output indefinitely. To avoid this, lag values exist. The lag value refers to the number of the previous steps that should be used to contribute to the output more extensively than the others. The AR model can be non-stationary as it can be represented by a unit root variable. The MA (Moving Average) model, however, is always stationary. Moving average is a linear regression model considering the shock values, contrary to the AR model which is the linear regression to non-shock values. There is no reason why you cannot combine these models and that is where the ARMA process comes in. The ARMA model combines the two models to make an accurate prediction which is more than the existing accuracy. The ARMA model compares the results for both models and makes a prediction based on the results [18]. The drawback of the ARMA process is it assumes a stationary time series. Thus it will not take seasonality into account. That is where ARIMA comes in handy. The I in ARIMA stands for integrated, which is the subtracting of previous observations in the series. Nonseasonal ARIMA is often described as ARIMA (p, d, q) where p is the number of lags in the AR model, d is the grade of differentiation (the number of times the data has previous values subtracted) and the q is the order of the MA model. As the ARIMA models are a combination of models, ARIMA (1, 0, 0) is the same as an AR (1) model as the integrated part and the moving average are not used. Lag refers to the delay in steps of time between two data set points that are being compared. The order of MA is how many of the previous shocks that the model will consider when predicting. A shock is an external influence or something that changes the value to an extreme point, both high and low. In the same way any ARIMA (p, d, q) where p is 0 is equal to an ARMA (p, q) model. Lastly, we aim to investigate different machine learning techniques as machine learning has been more and more prevalent in many areas in the last years. The concept of machine learning is not just to instruct the program on how to solve a problem, but instead to give the problem and let the computer solve it in its own way. One of the most prevalent solutions regarding machine learning and time series forecasting is the usage of neural networks. A neural network is a network of nodes that are connected and communicate with each other to solve a task. The network has one or more layers that are not visible but have neurons. These are the nodes with the information that helps calculate the result for the given problem. The result is not a definitive result, but rather an estimated result and the accuracy of the result depends on the number of hidden "neurons" and layers in the network. The more neurons and layers, the more calculations/operations, and the more accurate output.

2.2 Other Techniques

There are multiple different techniques used for time series forecasting, but the two previously mentioned methods are by far the more popular options today. There are different mathematical models that can be used for forecasting such as state-space models. The idea of space state models is that any dynamic system can be defined by differential equations. This means that the current state of the system is defined by the previous state and a state variable that changed the state. By this observation, we can define any system as a function of its states and external inputs. By knowing some of the observed data, we can calculate the optimal estimation for a selected state. The ARIMA models can also be converted into space state models [19], as they are to some degree differential equations as well. There are many variations on SSM but the most used is the linear SSM [19]. Other works tend to use a combination of models to create hybrid models to eliminate the inherent flaws of some models [20] and many of the works compare their hybrid solution to the preexisting models. A study from 2013 uses an Exponential Smoothing Space State model comparing it with other algorithms to forecast solar irradiance. The used data set had data collected monthly and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) stationary test proved the time series to be non-stationary. The results concluded that the ESSS model out-performed ARIMA, Linear Exponential Smoothing, and Simple Exponential 14 smoothing except for two months, but was only 0.5% behind, which means that the ESSS is a reliable model for their data set.

3. DATASET DESCRIPTION

The dataset we used is recorded by IoT sensors every 15 minutes over a period of 12 months from a transmission grid in Tripura, India. The dataset consists of various parameters as described below. We worked on the parameter "Reactive Power"(KVAR) to learn the working of Time-Series Forecasting algorithms.

	DeviceImei	DeviceTimeStamp	KWH	KVARH	KW	KVA	KVAR
0	867322031102411	2021-11-29 20:47:08	856412.3	865450.4	99.652	100.099	9.456
1	867322031102411	2021-11-29 20:46:36	856411.4	865449.6	102.500	102.909	9.169
2	867322031102411	2021-11-29 20:31:34	856385.4	865423.4	108.789	109.577	13.123
3	867322031102411	2021-11-29 20:15:18	856356.8	865394.6	104.330	104.893	10.857
4	867322031102411	2021-11-29 20:01:33	856331.6	865369.2	113.927	114.495	11.381

Fig -1: Snippet of the data frame which consists of 7 columns.

Given above is a snippet of the data that was used for this project. The dataset had 7 columns and 60134 rows. The columns are as follows:

- **DeviceImei:** The IMEI number of the transformer from which we are receiving the data.
- **DeviceTimeStamp:** The dataset recorded data every 15 minutes as mentioned above. The time of the data being procured is recorded and displayed in this column.
- **KWH:** The energy that a device consumes is measured in KWH(Kilowatt-Hour). The kilowatt-hour consumption records how many watts are used and how often they are used. In our dataset, the KWH signifies the measurement of the transfomer's wattage *and* the amount of time it was used.
- **KVARH:** KVARH stands for Kilo Volt Amperes Reactive Hours. It is a unit of reactive energy consumption, most commonly used in industries. Here, this column refers to the same unit of the transformer every 15 minutes.
- **KW:** It stands for Kilo Watt. It comprises 1000 Watts. It is a unit of electric power. It refers to the power dissipated by the transformer.
- **KVA:** A KVA refers to 1,000 volt amps. A volt is a unit of electrical pressure and an ampere is a unit of electrical current. Apparent power is the term used for the product of the volts and amps. This column contains this unit of power of the transformer.
- **KVAR:** It stands for Kilo Watt Amp Reactive. This unit is usually used to express power in all forms, but is most commonly used to express real power. This column describes this unit of the transformer.

Given above is the description of the different columns of the dataset.

Column name	Unit name
KWH	Kilowatt-hour
KVARH	Kilovolt Amperes Reactive Hours
KW	Kilowatt
KVA	Kilovolt Amperes
KVAR	Kilovolt Amperes Reactive

Table -1: Columns and units

Above table gives the units of different parameters which are given in the data frame

4. METHODOLOGY

4.1 Stationarity Analysis

One of the essential properties to analyze when examining data sets or time series is the stationarity of the time series. A stationary time series [9][12] is not affected by seasonality, i.e., the trend is consistent. To evaluate the stationarity of our dataset, we adopted Augmented Dickey-Fuller Test [10]. This test is based on an autoregression model, which optimizes for different lag values. Lag is the delay in time between two values in the time series that you are comparing. The mathematical formula of Augmented Dickey-Fuller test can be given as:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \delta_1 \Delta y_{t-1} + \cdots + \delta_{p-1} \Delta y_{t-p+1} + \epsilon_t,$$

α = constant

β = coefficient on time trend

p = lag order of the autoregressive process

y_t = variable of interest

t = time index

Δ = first difference operator

The Augmented Dickey-Fuller test uses a lag variable to eliminate autocorrelation from the results. Supporting the null hypothesis that γ=0 against the alternative hypothesis γ<0, the unit test is carried out. In Python, the Augmented Dickey-Fuller test is available in the Statsmodels package as adfuller(). The Table-2 shows the critical values for the Dickey-Fuller Test.

The Table-2 shows the critical values for the Dickey-Fuller Test.

Sample Size	1%
T=25	-4.38
T=50	-4.15
T=100	-4.04
T=250	-3.99
T=500	-3.98
T=501+	-3.96
Calculated Value:	-3.705

Table -2: Critical values for Augmented Dickey-Fuller Test and our results from the root test.

By observing the critical values we can see that the calculated value is not lower than the critical value than the critical value, which means that we cannot reject the null hypothesis and the time series is not stationary.

4.2 Implementation Of ARIMA

Our previous results showed that the time series is not stationary, this affects the selection of the ARIMA models. Since our time series show both seasonality and trend, the Seasonal ARIMA model is the best choice. H. Matsila et al. [11] used the Seasonal ARIMA model for successfully forecasting over their small data set. They had a weekly seasonality and did short forecasts with very high accuracy. Hence, we chose Seasonal ARIMA Model to weigh in the seasonality and produce results with higher accuracy. The Seasonal ARIMA model requires 7 parameters usually denoted by SARIMA (p,d,q)(P,D,Q)m, where (p,d,q) are trend order and (P,D,Q)m represent the seasonal order. The trend order is the same values which we would be using for fitting an ARIMA model, i.e., p is autoregressive order, d is differencing order and q is moving-average order as mentioned above. The seasonal elements also have the same parameters, however, m is the number of time steps for each season. We used a grid search to get the most fitting parameters for the ARIMA forecasting models. We defined a set of model combinations to test which forecast model has the lowest error regarding our data set. To evaluate which model is the best fit, we assessed based on multiple one-step forecasts, compared the actual value, and calculated the RMSE. The model that the grid search found out to be the best is the ARIMA(1,0,2)(2,1,0)100 with no trend, linear trend, and constant with a linear trend.

4.3 Implementation Of Prophet

Facebook's Prophet Model has a few limitations on how its data should be represented. It presumes to have an input of two columns, namely 'ds' for Date-Time Stamp and 'y' for target variable. The model does not accept any other names for columns. In our data the target variable is 'KVAR'. A Prophet() object is defined and configured in order to use Prophet for forecasting. Then it is fit on the dataset by calling fit() function and passing the data to it. The Prophet() object takes seasonality as an argument. In our case, we took the seasonality to be weekly. The model figures out almost everything else automatically.

By calling the predict() function and passing the data frame that contains the column named 'ds' and all the rows with date-times for which the forecasts are to be made. The result of the predict() function is a data frame which consists of numerous columns, out of which the most important are the columns 'ds' and 'yhat', where 'yhat' is the predicted value. The columns 'yhat_upper' and 'yhat_lower' give the uncertainty of the forecasted value. The forecasted values can be visualized with the help of plot().

5. PERFORMANCE METRICS

To evaluate the individual forecasting models prediction ability, we made use of commonly used performance metrics for measuring prediction errors. Prediction errors are the difference between the predicted and the actual values. MSE and RMSE give the error in regard to median of the data.

5.1 Mean Squared Error

The mean squared error is a measure that refers to how close a set of points is to the structured regression line. It is measured by calculating the distances from each of the points to the regression line and squaring them. These distances are essentially the errors. The squaring of the distance is done to remove the negative values. It gives higher importance and weight to larger differences. Since we are finding the average of a set of errors, it is referred to as the mean squared error. We prefer a lower MSE, this signifies a better forecast.

$$MSE = \frac{\sum_{i=1}^{n}(x_i - x'_i)^2}{n}$$

In the above formula,
MSE = Mean Squared Error
n = Number of data points
x_i = Actual Values
x' = Predicted Values

5.2 Root Mean Squared Error

A commonly used metric for evaluating results of predictions is the Root mean square error, also referred to as root mean square deviation. Similar to mean squared error, it shows how far the true values are from the predictions. RMSE uses Euclidean distance to measure the distance. To calculate RMSE, we have to calculate the residual which is the difference between the truth and the prediction for every data point. We then have to compute the residual norm at all data points, and further calculate the mean of these residuals to finally take the square root of the obtained mean. RMSE needs true measurements at each predicted data point, thus, it is commonly used in supervised learning problems.
We can express RMSE as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x'_i)^2}{n}}$$

where,
RMSE = Root Mean Squared Error
n = Number of data points
x_i = Actual Values
x_i' = Predicted Values

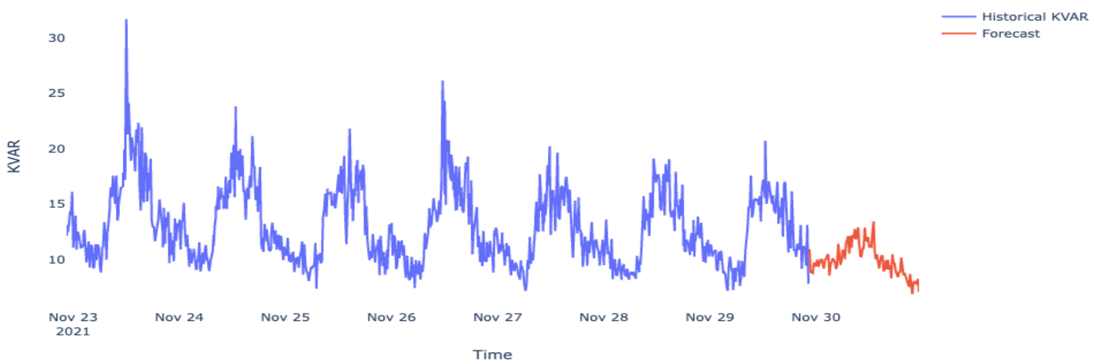

Fig -2 : Visualization of forecasted values of reactive power using ARIMA.

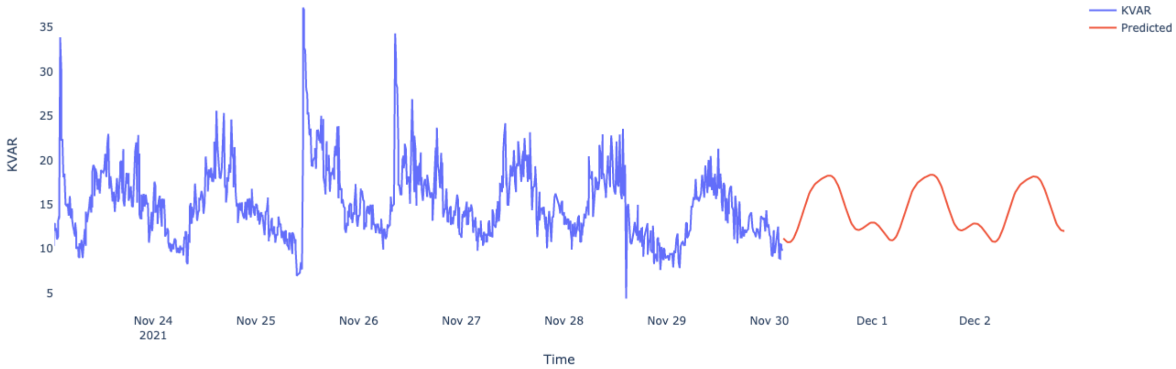

Fig -3 : Visualization of forecasted values of reactive power using Prophet.

6. RESULTS

We analysed our data set and investigated the implementation of traditional Seasonal ARIMA and Prophet time-series forecasting methods. As mentioned earlier, the difference between the real value and the estimated value of the data is found with the MSE (Mean Square Error) and RMSE (Square Root of the Mean Square Error) methods. The MSE value for the forecasting using ARIMA Model is 14.006 and the RMSE value is 3.743. When we used the Prophet Model the MSE and RMSE turned out to be 18.915 and 4.349 respectively.

7. DISCUSSION AND FUTURE WORK

From the results gathered, in our case for this data set and the algorithms we used, the Seasonal ARIMA model had the best prediction accuracy of the tested methods. It can be clearly observed that the ARIMA model shows best results as it's applied over a short period of time. Whereas, the Prophet couldn't show satisfactory results.

As a prospect of the future work, the same dataset can see the implementation of a different set of models including exponential smoothing and LSTM to check the results. Additionally, the data that was obtained from the transformer was only power related, the next step could be to apply the similar models to different sets of data obtained in real time from the transformer. This could lead to better results of time series forecasting.

8. CONCLUSION

In conclusion, we were successful in perfectly laying out the differences in the performances of the ARIMA model and the Prophet model. We also went on to explain the highly likely reasons for the same. In data of the sort that we used, ARIMA gives the best results since it doesn't need data of a very long time span. It successfully works on data with a short time span too.

REFERENCES

[1] D.Webby, R. and O'Connor, M.Webby, Richard, and Marcus O'Connor. "Judgemental and Statistical Time Series Forecasting: A Review Of The Literature". International Journal of Forecasting, vol 12, no. 1, 1996, pp. 91-118. Elsevier BV.

[2] Bryan Lim and Stefan Zohren"Time Series Forecasting With Deep Learning: A Survey(2021) Available at: https://arxiv.org/pdf/2004.13408.pdf

[3] Fattah, Jamal & Ezzine, Latifa & Aman, Zineb & Moussami, Haj & Lachhab, Abdeslam. (2018). Forecasting of demand using ARIMA model. International Journal of Engineering Business Management. 10. 184797901880867. 10.1177/1847979018808673.

[4] Moritz, S. et al. (2015) Comparison of different Methods for Univariate Time Series Imputation in R, arXiv.org. Available at: https://arxiv.org/abs/1510.03924.

[5] Chakraborty, K., Mehrotra, K., Mohan, C., & Ranka, S. (1992). Forecasting the behavior of multivariate time series using neural networks.

[6] Hannan, E., & Kavalieris, L. (1986). REGRESSION, AUTOREGRESSION MODELS. Journal Of Time Series Analysis.

[7] "Generalized Autoregressive Moving Average Models". 2021. Journal Of The American Statistical Association.

[8] "Applied Linear Regression". 2021. Google Books. https://books.google.co.in/books?hl=en&lr=&id=xd0tNdF0OjcC&oi=fnd

[9] P.J.BrockwellandR.A.Davis,IntroductiontoTimeSeriesandForecasting,3rded. Springer, 2016.

[10] W. A. Dickey, D. A.; Fuller, "Distribution of the estimators for autoregressive time series with a unit root," Journal of the American Statistical Association, vol. 74, p. 427–431, 1979.

[11] H. Matsila and P. Bokoro, "Load Forecasting Using Statistical Time Series Model in a Medium Voltage Distribution Network," in IECON 2018 - 44th Annual

Conference of the IEEE Industrial Electronics Society. IEEE, 2018.

[12] Whittle, P. 1953. "Estimation And Information In Stationary Time Series". Arkiv För Matematik 2.

[13] Baldigara, Tea, and Maja Mamula. 2015. "Modelling International Tourism Demand Using Seasonal ARIMA Models".

[14] Rougier, Jonathan. 2016. "Ensemble Averaging And Mean Squared Error". Journal Of Climate 29 (24): 8865-8870. doi:10.1175/jcli-d-16-0012.1.

[15] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and arima models for time series forecasting," Expert systems with applications, vol. 39, pp. 4344–4357, 2012.

[16] B. Z. . Y.Wei, "Carbon price forecasting with a novel hybrid arima and least squares support vector machines methodology," Omega, vol. 41, pp. 517–524, 2013.

[17] M. B. Mehdi Khashei and S. R. Hejazi, "Combining seasonal arima models with computational intelligence techniques for time series forecasting," Soft Computing, vol. 16, pp. 1091–1105, 2012.

[18] P. J. Brockwell and R. A. Davis, Introduction to Time Series and Forecasting, 3rd ed. Springer, 2016.

[19] F.D.K.S.K.V.M.J.W.W. Paninski L, Ahmadian Y, "A new look at state- space models for neural data," Journal of Computational Neuroscience, vol. 29,pp. 107–126, 2006.

[20] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," Neurocomputing, vol. 50, p. 159 – 175, 2003.