A Time-Series Forecasting Performance Comparison for Neural Networks with State Space and ARIMA Models

Milton Soto-Ferrari
Marketing and Operations Department
Indiana State University, Terre Haute IN 47807, USA
milton.soto-ferrari@indstate.edu

Odette Chams-Anturi
Department of Economic Science
Universidad de la Costa, Barranquilla, Colombia
ochams@cuc.edu.co

Juan P. Escorcia-Caballero
Department of Entrepreneurship and Management
Universidad del Norte, Barranquilla, Colombia
juane@uninorte.edu.co

Abstract

This research focuses on the development of an automated forecasting procedure that implement State Space (SS), Auto Regressive Integrated Moving Average (ARIMA), and Neural Networks (NN) to identify the best forecasting strategy for time series with numerous patterns. The proposed approach is applied on multiple time series exhibiting different series patterns from the M4 competition. Based on our study, the performance of ARIMA models showed superior results when compared to the ETS performance for seasonal data. In addition, NN and ARIMA showed a higher performance for cyclical and non-stationary data. NN performance was competitive in all types of data patterns. ARIMA stepwise selection procedure proved to be the most accurate in general for all the series. This delimited development is part of a comprehensive application that will encompass a dashboard tool designed to automatize forecasting procedures of different types of time series presented in the industry.

Keywords
Forecasting, State Space, ARIMA, Neural Networks.

1. Introduction

The growing demand and increasing competitive global economy has grasped a lot of attention for researchers, analysts, and companies towards time series forecasting in the past few years (Hyndman et al., 2002). A time series is a sequence of observations taken sequentially in time. Many data sets on the industry are by definition time-series (e.g., yearly orders in a factory for clothing, weekly series of sales, or hourly observations of manufacturing goods) (Seifert et al., 2015) . Ample effort had been devoted over the past years to the development and escalation of the precision in time series forecasting models. Therefore, recently several time series forecasting models have been created, developed, and evolved in the literature (Mentzer and Bienstock, 1998; Widiarta et al., 2008). Due to the vital importance of time series forecasting in numerous fields such as business, economics, finance, sales, and engineering; appropriate methods should be applied on the relative typology of the data and the existent forecasting conditions (Mentzer and Bienstock, 1998).

Forecasting conditions should consider time horizons, inputs available, data patterns, and many other aspects to select an appropriate method. These methods can be simple, such as using the most recent observation as a forecast (naïve method), or exceedingly complex, such as neural networks predictions. The best model to use depends on several factors such as; historical data, the connections between the forecasting attribute with the predictors, time horizons, and implementation. It is common to compare two or three potential models. Each model is itself an artificial architype that is based upon a set of hypothesis and assumptions that include factors that are estimated using historical data (Hyndman and Athanasopoulos, 2018).

On this research, we proposed a procedure that develop forecasts of time series with different patterns using state space (SS), auto regressive integrated moving average (ARIMA), and neural networks (NN) models. The proposed approach is applied on six time series selected from the M4 Competition. Currently the M-Competitions, now in its four version, have attracted great interest in providing objective evidence of the most appropriate way of forecasting various variables of interest. In our application, we would follow a similar structure as detailed in Soto-Ferrari et al. (2019), in which the forecasts for each of the series are calculated individually using ETS, ARIMA, and NN. Forecasts are calculated for each technique and the performance is determined by calculating the root mean squared error (RMSE), the mean absolute error (MAE) and the mean absolute percentage error (MAPE) in a defined testing set, focusing on the identification of patterns for each series (e.g., stationary, seasonal, cyclical).

This procedure is a delimited development of a comprehensive application that will encompass a dashboard tool designed to automatize forecasting procedures for different types of time series presented in the industry. Our emphasis is the designing of an automated forecasting instrument that could be implemented extensively in the industry. The structure of this work is as follows. In Section 2, we describe the forecasting methods and our proposed procedure. Section 3 describes the data and the assessment of forecasting performances. Section 4 shows the results of the application, and Section 5 concludes the paper.

2. Forecasting Models Description

2.1 State Space Models (ETS)

In the ETS context, the components of the exponential smoothing models are decomposed into three categories, the trend component, the seasonal component, and the remainder or error component. The trend component refers to the direction of the series, the seasonal component refers to the recurring elements of a series with a certain periodicity, and the remainder or error component refers to the unpredictable elements of the series (Hyndman et al., 2008). For uncertainty assessment, in the ETS context we can obtain both point forecasts and prediction intervals.

There are three basic variations in exponential smoothing models: simple exponential smoothing, double exponential smoothing, also called trend-corrected exponential smoothing or Holt's model with additive damped trend errors, and the Holt-Winters additive and multiplicative methods with a probability for damped trend errors (Brown, 1959; Holt, 1957; Winters, 1960). Each deterministic exponential smoothing model can be reformulated as two stochastic ETS models, one that includes additive errors, and one that includes multiplicative errors (Ramos et al., 2015). Differentiating between these two alternatives is only relevant for prediction intervals, not point forecasts. In the ETS notation, E stands for type of error (additive or multiplicative), T stands for the modeling options in the trend (non-existent (N), additive (A), multiplicative (M), damped additive (Ad) or damped multiplicative (Md)), and S stands for the modeling options in the seasonality (non-existent (N), additive (A), multiplicative (M)) (Hyndman and Khandakar, 2007). Table 1 provides the recursive formulas to apply for progressive calculations and point forecasts. Where:
y_t denotes the estimation of the forecast at time *t*, l_t denotes the estimation of series level at time *t*, b_t denotes the estimation of the slope (trend) at time *t*, s_t denotes the estimation of seasonality at time *t, m* denotes the number of seasons in a year. The constants α, β, γ are the smoothing parameters that are constrained between 0 and 1 for the purpose of interpreting the data sets as moving averages (Makridakis et al., 1998).

Table 1. State space equations for models structures.

ADDITIVE ERROR MODELS			
Trend	**Seasonal**		
	N	***A***	***M***
N	$y_t = l_{t-1} + \varepsilon_t$ $l_t = l_{t-1} + \alpha\varepsilon_t$	$y_t = l_{t-1} + s_{t-m} + \varepsilon_t$ $l_t = l_{t-1} + \alpha\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = l_{t-1} + s_{t-m} + \varepsilon_t$ $l_t = l_{t-1} + \alpha\varepsilon_t / s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t / l_{t-1}$

A	$y_t = l_{t-1} + b_{t-1} + \varepsilon_t$ $l_t = l_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$	$y_t = l_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $l_t = l_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (l_{t-1} + b_{t-1})s_{t-m} + \varepsilon_t$ $l_t = l_{t-1} + b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(l_{t-1} + b_{t-1})$
$\boldsymbol{A_d}$	$y_t = l_{t-1} + \phi b_{t-1} + \varepsilon_t$ $l_t = l_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$	$y_t = l_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $l_t = l_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (l_{t-1} + \phi b_{t-1})s_{t-m} + \varepsilon_t$ $l_t = l_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = \phi b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(l_{t-1} + \phi b_{t-1})$
MULTIPLICATIVE ERROR MODELS			
Trend	**Seasonal**		
	N	***A***	***M***
N	$y_t = l_{t-1}(1 + \varepsilon_t)$ $l_t = l_{t-1}(1 + \alpha\varepsilon_t)$	$y_t = (l_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $l_t = l_{t-1} + \alpha(l_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(l_{t-1} + s_{t-m})\varepsilon_t$	$y_t = l_{t-1}s_{t-m}(1 + \varepsilon_t)$ $l_t = l_{t-1}(1 + \alpha\varepsilon_t)$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
A	$y_t = (l_{t-1} + b_{t-1})(1 + \varepsilon_t)$ $l_t = (l_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(l_{t-1} + b_{t-1})\varepsilon_t$	$y_t = (l_{t-1} + b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $l_t = l_{t-1} + b_{t-1} + \alpha(l_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = b_{t-1} + \beta(l_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(l_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (l_{t-1} + b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $l_t = (l_{t-1} + b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(l_{t-1} + b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$
$\boldsymbol{A_d}$	$y_t = (l_{t-1} + \phi b_{t-1})(1 + \varepsilon_t)$ $l_t = (l_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(l_{t-1} + \phi b_{t-1})\varepsilon_t$	$y_t = (l_{t-1} + \phi b_{t-1} + s_{t-m})(1 + \varepsilon_t)$ $l_t = l_{t-1} + \phi b_{t-1} + \alpha(l_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = \phi b_{t-1} + \beta(l_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(l_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (l_{t-1} + \phi b_{t-1})s_{t-m}(1 + \varepsilon_t)$ $l_t = (l_{t-1} + \phi b_{t-1})(1 + \alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(l_{t-1} + \phi b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1 + \gamma\varepsilon_t)$

2.2 Auto Regressive Integrated Moving Average (ARIMA)

ARIMA is an extensive class of prediction models that can represent autocorrelated and stochastic seasonal and non-seasonal time series, including autoregressive (AR), moving average (MA) and mixed AR or MA processes with differentiated or integrated (I) baselines (Hyndman and Athanasopoulos, 2018). The autoregression terms indicate a dependency of a current value against its own previous observations. The integrated term indicates the number of times differentiation has to occurred (i.e. subtracting an observation from an observation at the previous time step) to make the time series stationary. The moving average terms indicate a dependency between current value and the residual error of previous observations.

The notation in these types of models is formally denoted as ARIMA(*p,d,q*), where *p* represents the number of lagged values to consider for autoregression, *d* represents the number of times the series have been differentiated to achieve stationarity, and *q* represents the number of moving average parameters that are accounted in the model for random jumps in the time series (Series, 1970). For seasonal ARIMA the notation consider the following: ARIMA(*p,d,q*)(*P,D,Q*), where the uppercase letters have the same meaning as the lowercase letters, but these are referred only to seasonal parameters. For illustration, let us denote the formulation for the multiplicative seasonal ARIMA model *(p,d,q) x (P,D,Q)m*:

$$\emptyset_p(B)\emptyset_P(B^m)(1 - B)^d(1 - B^m)^D y_t = c + \theta_q(B)\Theta_Q(B^m)\varepsilon_t$$

Where:

$$\emptyset_p(B) = 1 - \emptyset_1 B - \cdots - \emptyset_p B^p, \emptyset_P(B^m) = 1 - \emptyset_1 B^m - \cdots - \emptyset_P B^{Pm}$$
$$\theta_q(B) = 1 + \theta_1 B + \cdots + \theta_q B^q, \Theta_Q(B^m) = 1 + \Theta_1 B^m + \cdots - \Theta_Q B^{Qm}$$

With *m* as the seasonal frequency, *B* is the backward shift operator, *d* is the degree of ordinary differencing, and *D* is the degree of seasonal differencing. $\emptyset_p(B)$ and $\theta_q(B)$ are the regular autoregressive and moving average polynomials of orders p and q, respectively, $\Theta_Q(B^m)$ and $\Theta_Q B^{Qm}$ are the seasonal autoregressive and moving average polynomials of orders P and Q, respectively, $C = \mu(1 - \emptyset_1 - \cdots - \emptyset_p)(1 - \emptyset_1 - \cdots - \emptyset_p)$where μ is the mean of $(1 - B)^d(1 - B^m)^D y_t$ process and ε_t is a zero mean Gaussian white noise process with variance σ2.

2.3 Neural Networks (NN)

Neural networks are a collection of nodes interconnected between themselves with different processing layers of neurons that allow for the analysis of different scenarios through the combination of the parameters or cases pertaining to each node. A neural network is typically composed by three parts or layers which are the input layer, which contains the predictors; the hidden layer where the data from the predictors is processed and analyzed; and the output layer which contains the forecasts from the data (Han et al., 2011).

The simplest form of neural networks come when there are no intermediate hidden layers in the model, yielding an output equivalent to a linear regression. Thus, when the hidden layers are added, the model automatically becomes non-linear. For time series forecasting, the predicted outputs are generated by the weighted combination of the inputs. Learning algorithms that minimize the cost function of the total operation of the model, such as the mean square error, are used to select the weights in neural networks and the hidden layers are required when the network has to predict data that is complicated or non-obvious. When dealing with time series data, inputs to the neural network may come from lagged values pertaining to the time series. This type of models are known as Neural Network Auto Regression (NNAR) (Hyndman and Athanasopoulos, 2018).

In this study, feed-forward network models with only one hidden layer were considered and the notation $NNAR(p, k)$ was used to denote the p lagged inputs and the k nodes in the hidden layer. Moreover, for seasonal data the last observed values from the seasonal inputs were added and considered in the model. Therefore, as a more general expression, we considered the model $NNAR(p, P, k)_m$ with inputs $(y_{t-1}, y_{t-2}, \dots y_{t-p}, y_{t-m}, y_{t-2m}, y_{t-Pm})$ and k neurons in the hidden layer. In order to develop $NNAR(p, P, k)_m$ models, the optimal number of lags for non-seasonal time series were selected according to the default number established by AIC for a linear $AR(p)$ model; whilst for seasonal time series the default values were chosen as $P = 1$ and p adjusted according to the optimal linear model fitted to the adjusted seasonal data. If not specified, then $k = (p + P + 1)/2$ rounded to the nearest integer.

3. Data Selection and Forecasting Procedure Description

All data is selected from the M4 competition (MOFC, 2020). We consider the monthly dataset in which the data is separated into training and testing sets. The selected series are composed by the following six components: M10, M11, M12, M13, M14, M15 (names as described in the M4 competition). The total number of observations varies in size however, the testing set for each have an equal size of 18 months. We would consider an ending date of all series including training and testing as December 2019 (12/2019), based on this we would calculate the beginning date of each set. These data sets are independently evaluated using ETS, ARIMA and NN models; we would find a forecast for a 95% prediction interval for each testing set series. In order to evaluate forecast performance for the different strategies, RMSE, MAE, and MAPE are calculated and presented for the testing set. Where:

$$\text{RMSE} = \sqrt{\frac{1}{n-m}\sum_{t=m+1}^{n}(y_t - \hat{y}_t)^2} \qquad \text{MAE} = \frac{1}{n-m}\sum_{t=m+1}^{n}|y_t - \hat{y}_t|$$

$$\text{MAPE} = \frac{1}{n-m}\sum_{t=m+1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right|\text{x100}$$

For these formulations, *m* is the periodicity; y_t is the true value of the time series *y* at time *t*, and $\hat{y}_t$ is the forecast. For this research, state space models are estimated using the forecast package for R statistical software described in (Hyndman and Khandakar, 2007). The automatic ETS function is used to estimate the state space model form parameters. The AUTO.ARIMA function implemented in the same package is also used to identify and estimate the ARIMA models. This function conducts a stepwise selection over possible models and returns the best ARIMA model for the series, we would activate the trace setting to perform an extensive and continuous evaluation of the models (i.e. auto.arima(training series ,trace = TRUE, stepwise=FALSE, approximation=FALSE)) . For neural network forecasts, the NNETAR function is implemented. For this approach, we would do an iterative application with a total of 100 networks repeats fitted to the series (i.e. nnetar (train.C1.ts, repeats = 100)). Every repeat would start with different random weights on the inputs that are then averaged when computing the forecasts. The network is trained for one-step forecasting and multi-step forecasts are computed recursively. Figure 1 presents the graphical representations of the selected series with their specific pattern description.

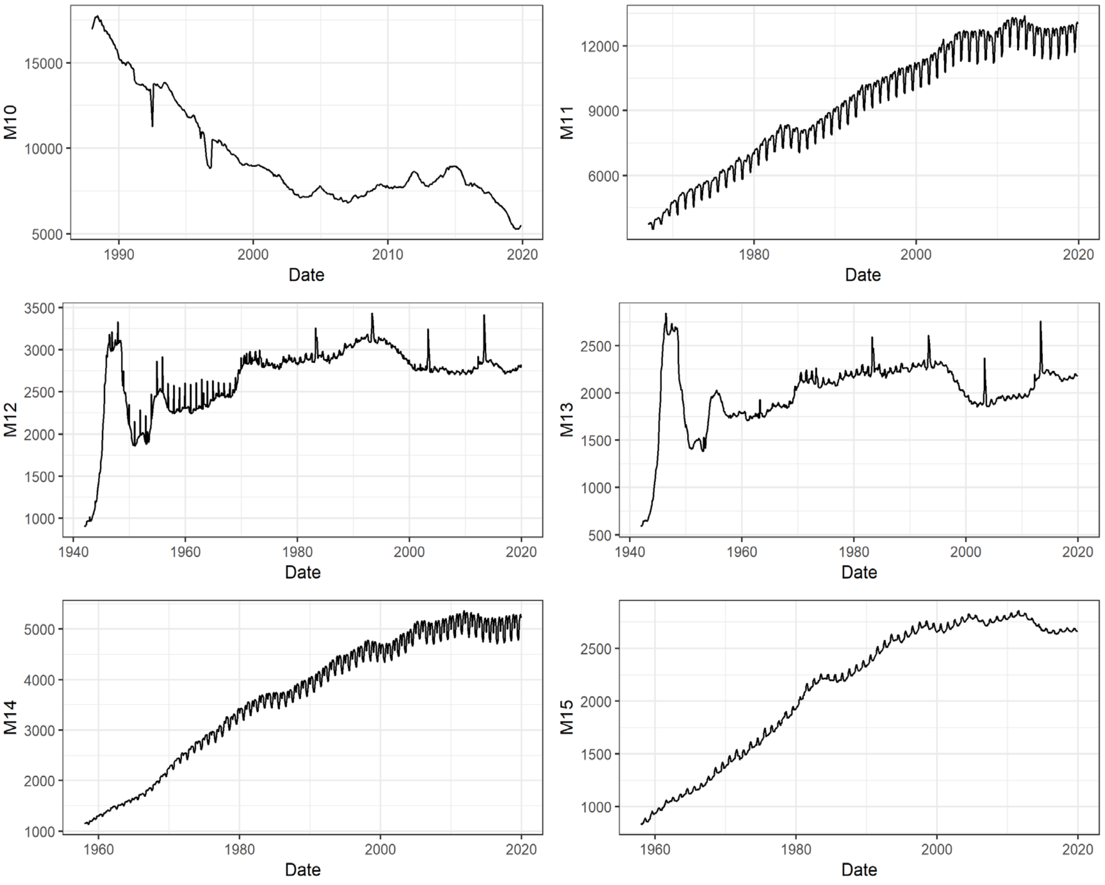
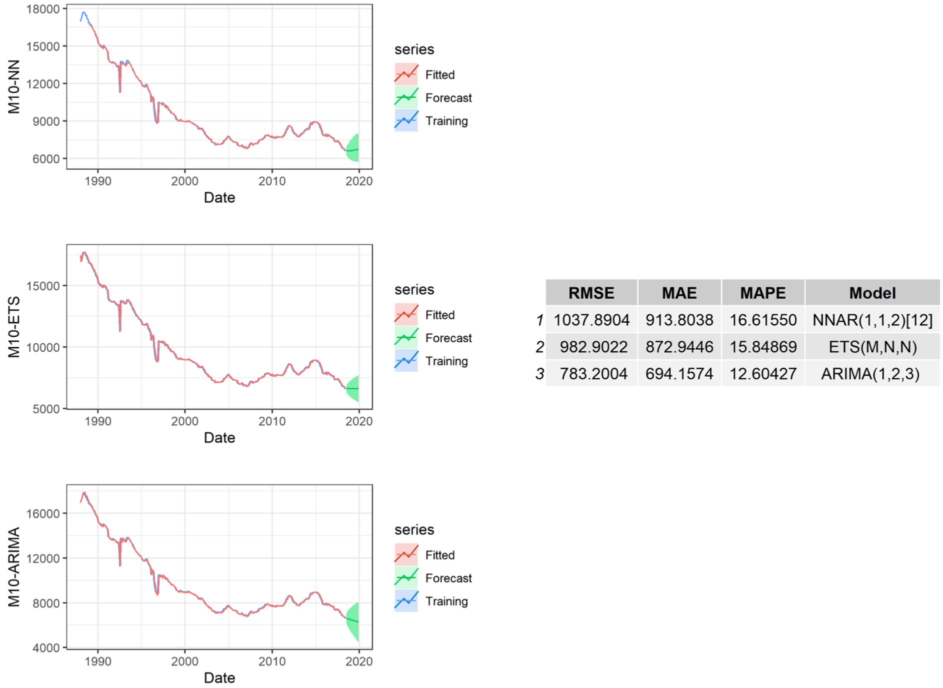


Figure 1. Selected Data Sets

From the components, we can observe that the series M11, M14, M15 present a monthly seasonal pattern over time with a positive trend (upward). While series M12 and M13 appear to be cyclical after the year 1980. However, the values appear to be non-stationary decreasing and increasing randomly in each cycle. Series M10 is a negative trend series with a shift and a slight stationary period between years 2009 and 2015. The six components are selected for analysis because these represent a considerable spectrum of patterns that could be found in a real context. We developed the forecasts for all the series with the calculation of their RMSE, MAE, and MAPE for each method (i.e. ETS, ARIMA, NN). The following section presents the forecast results for each of the series. Forecast performance is calculated for the testing set in each series.

4. Results

The following figures present the graphical and numerical results of the performance measures for each of the series. Each figure shows the fitted forecast for the training set (red) and the forecast including the prediction intervals for the testing set (green). For series M10 the best overall method is ARIMA (1,2,3) with an estimated MAPE of 12.6%. Figure 2 presents the results for M10.

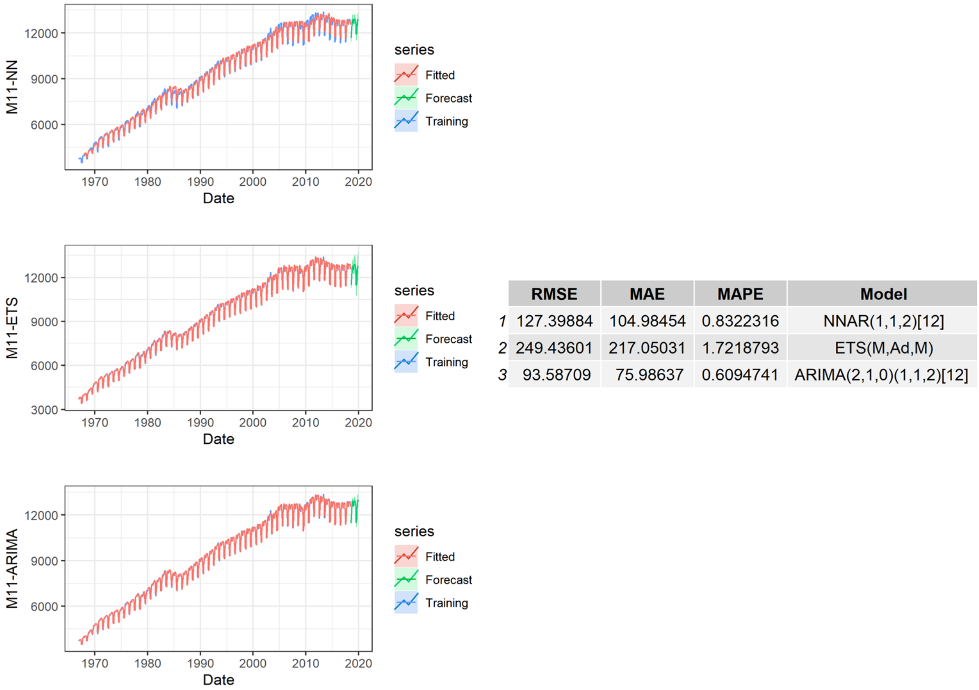


	RMSE	MAE	MAPE	Model
1	1037.8904	913.8038	16.61550	NNAR(1,1,2)[12]
2	982.9022	872.9446	15.84869	ETS(M,N,N)
3	783.2004	694.1574	12.60427	ARIMA(1,2,3)

Best Method: ARIMA

Figure 2. M10 Forecasts

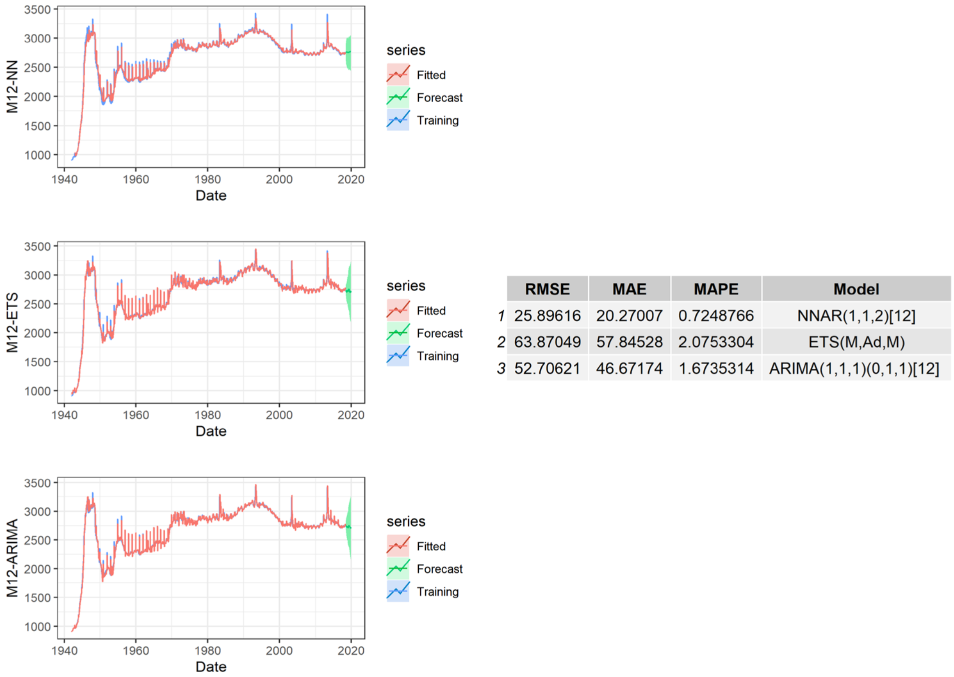
For series M11 the best overall method is ARIMA(2,1,0)(1,1,2)[12] with an estimated MAPE of 0.61%. For series M12 the best overall method is NNAR(1,1,2)[12] with an estimated MAPE of 0.72%. Both M11 and M12 series as described present seasonal patterns that are best represented by ARIMA and NNAR models respectively. Figure 3 presents the results for series M11 and Figure 4 for series M12.



	RMSE	MAE	MAPE	Model
1	127.39884	104.98454	0.8322316	NNAR(1,1,2)[12]
2	249.43601	217.05031	1.7218793	ETS(M,Ad,M)
3	93.58709	75.98637	0.6094741	ARIMA(2,1,0)(1,1,2)[12]

Best Method: ARIMA

Figure 3. M11 Forecasts

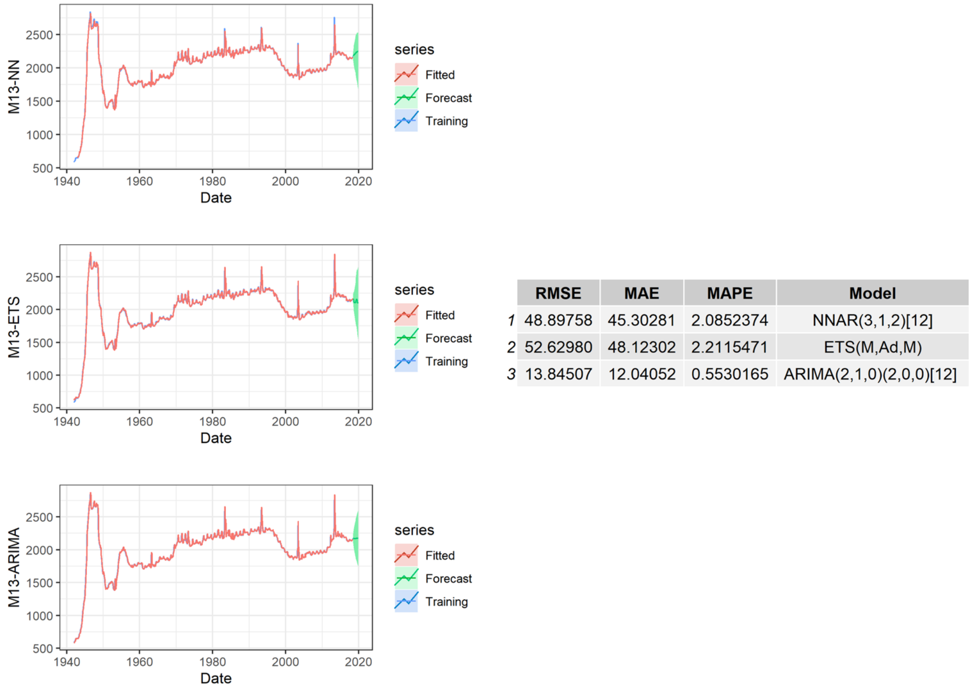


	RMSE	MAE	MAPE	Model
1	25.89616	20.27007	0.7248766	NNAR(1,1,2)[12]
2	63.87049	57.84528	2.0753304	ETS(M,Ad,M)
3	52.70621	46.67174	1.6735314	ARIMA(1,1,1)(0,1,1)[12]

Best Method: NN

Figure 4. M12 Forecasts

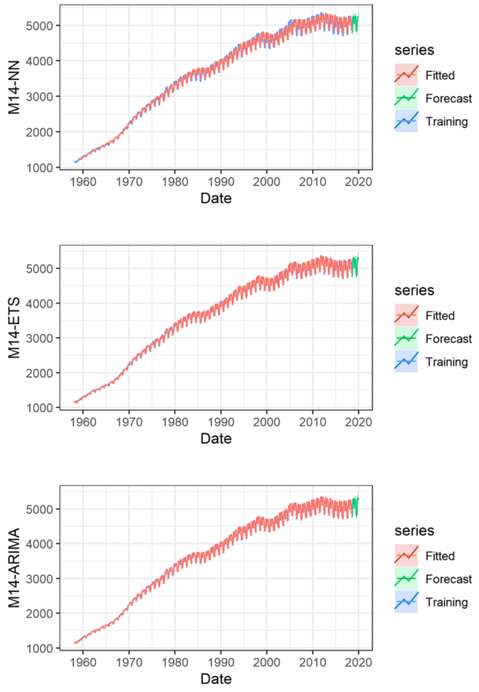
For series M13 the best overall method is ARIMA(2,1,0)(2,0,0)[12] with an estimated MAPE of 0.55%. For series M14 the best overall method is ETS(M,A,M) with an estimated MAPE of 0.20%. M13 presents an irregular seasonal pattern that increases the complexity on the prediction. M14 series present a seasonal pattern that is best represented by the Holt-Winters' multiplicative method (i.e. ETS(M,A,M)). Figure 5 presents the results for series M13 and Figure 6 for series M14.

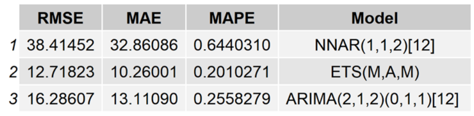


	RMSE	MAE	MAPE	Model
1	48.89758	45.30281	2.0852374	NNAR(3,1,2)[12]
2	52.62980	48.12302	2.2115471	ETS(M,Ad,M)
3	13.84507	12.04052	0.5530165	ARIMA(2,1,0)(2,0,0)[12]

Best Method: ARIMA

Figure 5. M13 Forecasts



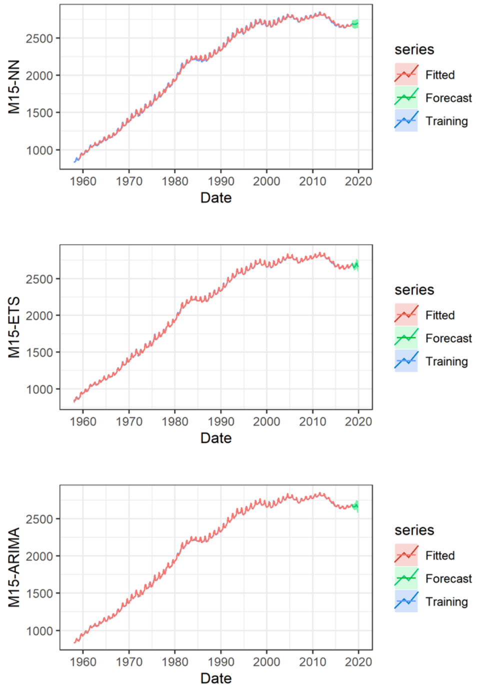


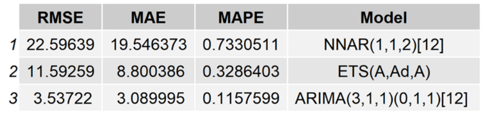
	RMSE	MAE	MAPE	Model
1	38.41452	32.86086	0.6440310	NNAR(1,1,2)[12]
2	12.71823	10.26001	0.2010271	ETS(M,A,M)
3	16.28607	13.11090	0.2558279	ARIMA(2,1,2)(0,1,1)[12]

Best Method: ETS

Figure 6. M14 Forecasts

For series M15 the best overall method is ARIMA(3,1,1)(0,1,1)[12] with an estimated MAPE of 0.12%. M15 series as described present a seasonal pattern that is best represented by ARIMA. Figure 7 presents the results for series M15.





	RMSE	MAE	MAPE	Model
1	22.59639	19.546373	0.7330511	NNAR(1,1,2)[12]
2	11.59259	8.800386	0.3286403	ETS(A,Ad,A)
3	3.53722	3.089995	0.1157599	ARIMA(3,1,1)(0,1,1)[12]

Best Method: ARIMA

Figure 7. M15 Forecasts

5. Discussion and Conclusions

Based on our study, the performance of ARIMA models showed superior results when compared to the ETS performance for seasonal data. In addition, NN and ARIMA showed a higher performance for cyclical and non-stationary data. NN performance was competitive in all types of data patterns with the remarkable characteristic that the NN automated method with repeats, adjusted a similar conceptual model of layers for different patterns of data by modifying the weighs of the inputs. ARIMA trace evaluation procedure proved to be the most accurate in general for all the series. If we would adapt the forecasting procedure into a real context NN and ARIMA, because of the settings of continuous trace and multiple repeats to identify the best model will require a longer running time. However, results of performance are expected to improve given this approach.

All the procedures for forecasting is automated to run with the R libraries "forecast", "zoo", "ggplot2", and "dplyr". We used the *ggplot2* library in combination with the *autoplot* function to run and create the graphical representations for the forecasts. The developed procedure is flexible and can be replied to any time series. We intent to apply our developments into projects developed in cooperation with companies or businesses in need of an automated forecasting dashboard. Our purpose is to apply and continuous to include efficient methods of automated forecasting practices such as bagging models and complex neural networks predictions into actual industry issues.

6. References

Brown, R. (1959), "Statistical forecasting for inventory control", *New York: McGraw Hill.*

Han, J., Pei, J. and Kamber, M. (2011), "Data mining: Concepts and techniques", *Elsevier.*

Holt, C. (1957), "Forecasting trends and seasonal by exponentially weighted averages", *International Journal of Forecasting*, Vol. 20 No. 1, pp. 5–13.

Hyndman, R. and Athanasopoulos, G. (2018), "Forecasting: Principles and practice", *OTexts. (Hydmand Book)*.

Hyndman, R. and Khandakar, Y. (2007), "Automatic time series for forecasting: The forecast package for R", *Clayton VIC, Australia: Monash University, Department of Econometrics and Business Statistics.*, Vol. 6/7.

Hyndman, R., Koehler, A., Ord, J. and Snyder, R. (2008), "Forecasting with exponential smoothing: The state space approach", *Springer Science & Business Media*.

Hyndman, R., Koehler, A., Snyder, R. and Grose, S. (2002), "A state space framework for automatic forecasting using exponential smoothing methods", *International Journal of Forecasting*, Vol. 18 No. 3, pp. 439–454.

Makridakis, S., Wheelwright, S. and Hyndman, R. (1998), "Forecasting: Methods and applications", *3rd Ed, John Wiley & Sons, New York.*

Mentzer, J. and Bienstock, C. (1998), "Sales forecasting management: Understanding the techniques", *Systems and Management of the Sales Forecasting Process. Sage Publications,ThousandOaks,CA*.

MOFC. (2018), "M4 Competition", available at: https://mofc.unic.ac.cy/m4/ (accessed 25 July 2020).

Ramos, P., Santos, N. and Rebelo, R. (2015), "Performance of state space and ARIMA models for consumer retail sales forecasting", *Robotics and Computer-Integrated Manufacturing*, Vol. 34, pp. 151–163.

Seifert, M., Siemsen, E., Hadida, A. and Eisingerich, A. (2015), "Effective judgmental forecasting in the context of fashion products", *Journal of Operations Management*, Vol. 36 No. 1, pp. 33–45.

Series, B.G.J.G.T. (1970), "Analysis: Forecasting and control", *San Francisco: Holden Day Press*.

Soto-Ferrari, M., Chams-Anturi, O., Escorcia-Caballero, J.P., Hussain, N. and Khan, M. (2019), "Evaluation of bottom-up and top-down strategies for aggregated forecasts: State space models and arima applications", *In International Conference on Computational Logistics. Springer, Cham*, pp. 413–427.

Widiarta, H., Viswanathan, S. and Piplani, R. (2008), "Forecasting item-level demands: An analytical evaluation of top–down versus bottom–up forecasting in a production-planning framework", *Journal of Management Mathematics*, Vol. 19 No. 2, pp. 207–218.

Winters, P. (1960), "Forecasting sales by exponentially weighted moving averages", *Management Science*, Vol. 6, pp. 324–342.

7. Biographies

Milton Soto-Ferrari is an Assistant Professor of the Operations and Supply Chain Management program in The Scott College of Business at Indiana State University. He has published in recognized journal and research conference proceedings. His research interests include business and healthcare analytics, supply-chain management, logistics, and industry application of forecasting policies. Currently, he is the director of a research project related to the automation of advanced forecasting techniques in the industry.

Odette Chams Anturi is a professor at the Science Economics Department, Universidad de la Costa (Colombia). Her research interests are Supply Chain Management, Organizational Learning, Innovation and Ambidexterity. She has worked in public and private companies of mass consumption and healthcare entities. Currently she is part of a research project about Quality Management and Ambidexterity in the Organic Agro-Food Industry.

Juan P. Escorcia Caballero is a professor at the Entrepreneurship and Management Department, Business School, Universidad del Norte (Colombia). His research interests are Supply Chain Quality Management, Innovation, and Ambidexterity. He has published several journal and conference papers. He has participated in three research projects funded by the Administrative Department of Science, Technology and Innovation of Colombia, and currently he is part of a research project about Quality Management and Ambidexterity in the Organic Agro-Food Industry.