Erasmus School of Economics

ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Bachelor Thesis Econometrics and Operations Research

Hybrid ANN and RNN to forecast retail sales

Supervisor :

Utku Karaca

Second assessor :

Kathrin Gruber

Name student and student ID number :

Louis Lacombe (466759)

Abstract

This paper looks into the different methods of forecasting monthly retail sales in the US. It compares many of the techniques such as linear model, Holt-Winters, Box-Jenkins but also neural networks. More precisely it looks into how a hybrid between the more classical econometrics techniques and neural networks can lead to a stronger model with more accurate forecasts. In general, we will find that most combinations of hybrids are better than the classical econometrics models. We will discover that by combining a linear model with either neural network, we find the lowest Mean Absolute Percentage Error (MAPE) values. Additionally, we won't find a significant difference between the use of a Recurrent Neural Network (RNN) as oppose to Artificial Neural Network (ANN) and find they have similar characteristics.

Date final version: July 4, 2020

Contents

1 Introduction

Retail sales are a very important component of the industry whether it is on a macro or microeconomic level. From a macro perspective, it accounts for two-thirds of the GDP which is an essential measure in order to evaluate the health of the economy. In microeconomics, retail sales could be an opportunity for businesses to anticipate sales, possibly by applying the algorithms on their own individual data, thereby making their supply chain model more efficient. These two scenarii show the importance, and some of the practical aspects of correctly forecasting retail sales. Since retail sales are a good example of time series data, we will discuss the different forecasting methods that have been used over the past years.

Often in time series data, the issue of non-stationary and seasonal data arises. The Holt-Winters procedure (Winters, 1960) helps resolve this issue as it smooths the data. While linear model techniques often explore the relationship between explanatory and the independent variable, with time-series data, an important aspect is the past behavior of the latter as it can help forecast future values. An example of that is the Box-Jenkins method which applies an auto-regressive integrated moving average (ARIMA) (Box et al., 1970). This method requires various steps such as parameter identification, estimation, and residual diagnostics.

Whereas the aforementioned methods assume a linear process to estimate the data, neural networks do not make this assumption. Hence, they offer a different perspective, a new way to interpret the world as opposed to the traditional econometrics methods. The idea of neural networks was first introduced by McCulloch & Pitts in 1943, they developed the idea that the nervous activity in our brains could be represented mathematically, starting the idea of using artificial neural networks (ANN). While ANN are not built for time series data, they have proven to lead to promising results (Kihoro et al., 2004; Alon et al., 2001; Ansuj et al., 1996; Kohzadi et al., 1996; Hill et al., 1996; Kuo & Reitsch, 1995). Specifically, in the paper by G. Zhang et al. they mention clearly that neural networks perform better than traditional techniques when there is a non-linear structure and if the data is more volatile and in large sample. Additionally, it is important to note that the parametrization for neural networks is crucial as otherwise, it could lead to traditional econometrics methods having a better forecast power (Sharda & Patil, 1992; Tang & Fishwick, 1993).

While there are a large number of different neural networks, one that has a specific application to past observations is recurrent neural network (RNN). As explained in (Elman, 1990) paper, the output of the neural network is used back as input. This allows for the neural network to possibly

find further patterns between the previous observation and the new forecast. As shown through various papers such as Kumar et al. (2004), the recurrent neural network seems to work better with time series problems. It also seems to outperform the more traditional techniques (Ho et al., 2002). Especially in volatile conditions, neural networks seem to be better as can be shown through the use of a separation algorithm, combing ARIMA and RNN in the most volatile cases (Shui-Ling & Li, 2017). As data can often have some linear and non-linear components, an interesting combination would be to use a hybrid model. This is often made of a linear method such as linear model or Box-Jenkins and a non-linear method such as ANN or RNN. Note that there are different kinds of hybrid models, some which combine the different approximations and return a weighted average (in parallel) (J.-J. Wang et al., 2012; Luxhøj et al., 1996) or those which use the residuals of one model to then try to predict the error terms (in series) (L. Wang et al., 2013; Tseng et al., 2002; Faruk, 2010). Finally, (L. Wang et al., 2013) discuss whether to use a multiplicative or additive method, that is, once having performed the ARIMA, using directly the error terms (additive method) or a fraction (multiplicative method), the error terms over the estimated values. The results show that the multiplicative method performs better on nearly all the evaluation criterion expect in short term forecasting.

Throughout the research we have highlighted some key findings from the past exploration. It has been a heavy topic of discussion over the past years, especially focusing on different combination of hybrid models. In this paper, we will try to understand which hybrid model works best for time series data, combining different linear models with ANN or RNN. Hence, our research question will be, "To what extent does changing the underlying models of a hybrid structure affect the forecasting accuracy of time series data?"

The paper will be structured as followed by first getting a better understanding of what has been done in previous papers, then giving some insights into the data and explaining how this data is suitable to answer the research question. Afterwards, we will look at methods discussed throughout the literature review, explaining the approach and parameters used. Additionally, in this section, we will elaborate on the multiple evaluation metrics used to compare the forecast. Subsequently, we will describe the results obtained, providing an understanding of them, explaining the different underlying ideas and the main outcomes from those results. Finally, we will answer the research question, explain the limitations and propose further research. In this paper, we will firstly replicate the paper by Alon et al. (2001), then we will add extensions such as creating hybrid models using linear models and neural networks in order to predict forecast retail sales.

2 Literature

To have a better understanding of the hybrid methods, we will have a look at one of the founding papers on hybrid models by G. P. Zhang in 2003. In this paper, the hybrid approach described is done in series. At first an ARIMA model estimates the values, from which we then get the residuals and the neural networks attempts to predict the residuals. They perform the results on three different data sets and compare them to a simple ARIMA or ANN. It is clearly found that the hybrid model not only improves the prediction but also has a lower variance. Additionally, by fitting the ARIMA model first, it solves the issue of overfitting the data.

In order to have a better understanding of the different types of hybrid methods, we will look into the paper written by Khashei & Hajirahimi in 2017 which discusses and compares the different approaches to a hybrid model. In this paper, the authors focus mainly on a neural network combined with ARIMA. Using the approach proposed by G. P. Zhang, they try different outcomes such as first passing the model through the neural network and predicting the residuals with ARIMA. They also make a parallel approach which consists of taking a weighted average of each model's prediction. In this case, multiple methods are also used to find the optimal weights such as a simple average, linear regression or algorithmic models. It is found that in general both methods improve the simple models of ARIMA and neural network. In addition, Khashei & Hajirahimi find that the model in series seems to work better than the parallel ones. This result is consistent over both data sets. Hence, we choose to use the series structure approach where we will try different combinations.

3 Data

To be able to replicate the paper by (Alon et al., 2001), we will therefore use the same data set. This data is the monthly retail sales in the United States from the US Central Bureau. This type of data is of great importance for organizational purposes and to be able to understand the economy better. Additionally, this is the exact same set of data as used in (Alon et al., 2001) hence, this will allow us to have a stronger comparison with the paper. It's important to note that two different sets of data will be analyzed, from Jan-1978 to Dec-1985 (period one) and Jan-1986 to April-1995 (period two). The first period is more volatile as it is during a time of supply push inflation, high unemployment and interest rates but also two recessions. In comparison the second period is much more stable with less fluctuations in the data. We can see a representation of that in the Figure 1 as we can clearly observe a higher fluctuation in difference of retail.

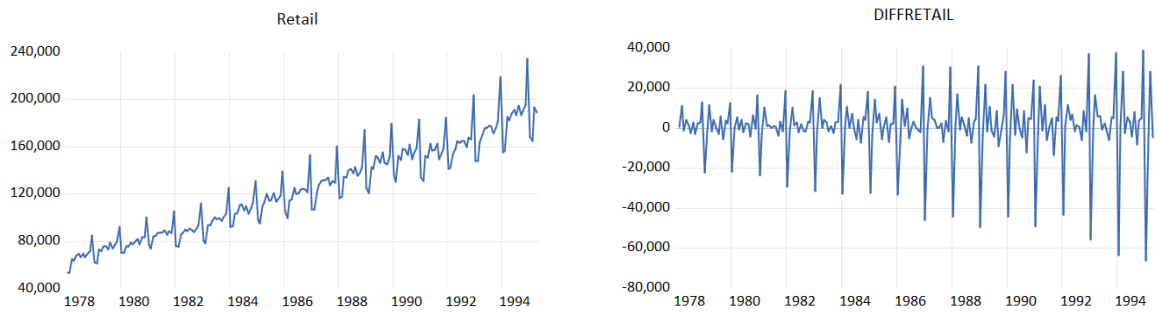

Figure 1: Plot of all observation on left and difference on right

When looking into the two periods, on a macroeconomic perspective, during period one we had two recessions versus only one in period two. Notably, the recession, from Jan-1980 to Jul-1980 due to a raise in interest rate in response to the inflation in 1970. This recession is very similar to the one that occurred during the second period, in 1991. The bigger difference between the two periods is due to the recession which occurred from Jul-1981 to Nov-1982 due to the energy crisis of 1979 and tight government monetary policies (Labonte et al., 2002; Walsh, 1993; Knoop, 2009). This dip was slightly stronger than the ones previously mentioned, with a GDP decline of 2.7% and higher peak unemployment of 10.8%. Hence, as we will split the data into two difference periods, we will make a data analysis of both separately.

3.1 Period one: Jan-1978 to Dec-1986

When we perform a simple model with just $y_t = \alpha + \beta t + \varepsilon$, we get an R^2 of 0.8105 which already a good performing model with coefficients $\hat{\alpha} = 62407$ and $\beta = 571.3512$ which is significantly different from zero, hence, indicating the presence of a trend. As there seems to be a clear upward trend, we perform an Augmented Dickey-Fuller (ADF) (based on automatic SIC, lag length 13, using constant and trend) to check our intuition, with a p-value of 0.791, so we do not reject the null hypothesis of a stochastic trend.

We check for seasonality by regressing the difference in retail on dummies for every month, we get an R^2 of 0.9429. This clearly shows that we have a seasonal pattern in the series as dummies alone manage to explain quite well the differences. This is confirmed by the Wald test: 115.1501 with p-value = 0.0000 which restricts that every dummy coefficient to be equal to zero. Additionally, we can clearly notice that between December and January there is a huge difference as can be observed in the Figure 1 with the large down spike.

Finally, to try and understand the parametrization needed for ARIMA models we look at the autocorrelations and partial autocorrelations (see Figure 5 in Appendix). From the autocorrelations, we can notice it could probably be interesting to take twice the difference. This is quite interesting as we can observe (see Figure 2) a very strong 2^{nd} and 11^{th} autocorrelation, which is probably strongly linked to the December-January difference we previously observed. Additionally, and non-surprisingly, the 12^{th} component is highly linked to the first while other months are not as important, this would also be why using a seasonal aspect could prove to be useful as then you would find the difference between each period. These would correspond to the autoregressive components, hence having a specification for an AR model. When looking at the partial autocorrelation, we actually check which could be the component for the MA model, in this case it would seem that a model with 8 moving average terms could be good.

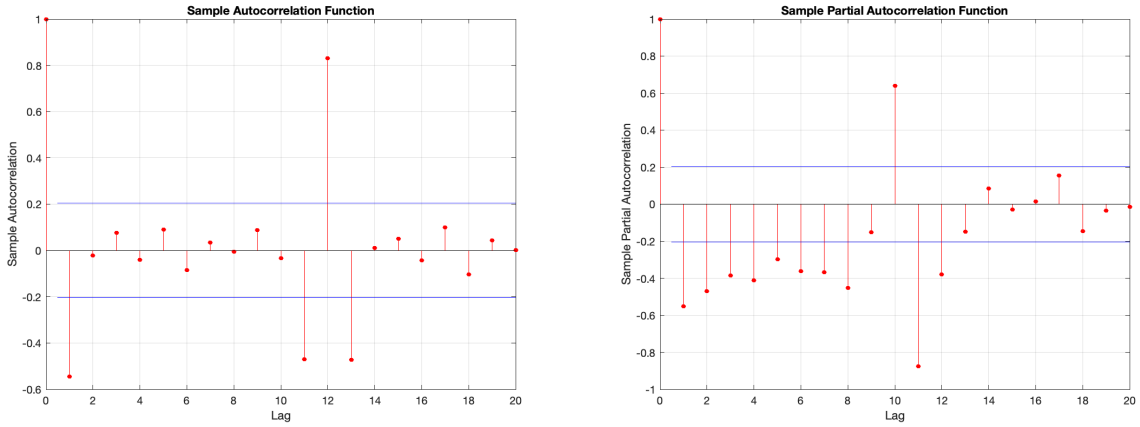

Figure 2: Twice differentiated retail first period autocorrelation and partial-autocorrelation

3.2 Period two: Jan-1986 to Apr-1995

When we perform a simple model with just $y_t = \alpha + \beta t + \varepsilon$, we get an R^2 of 0.7135 with coefficients $\hat{\alpha} = 53904$ and $\beta = 652.41$ which is significantly different from zero and when combined with the high R^2, would indicate a the presence of a trend. As there seems to be a clear upward trend, we perform an ADF (based on automatic SIC, lag length 13, using constant and trend) to check our intuition, with a p-value of 0.9297, so we do not reject the null hypothesis of a stochastic trend. This can also be recognized when looking at the autocorrelation of retails (see Figure 6 in Appendix). It's interesting to note in comparison to period one, period two seems to have a stronger trend as can be seen from the β coefficient or a higher non-rejection for the ADF.

When checking for seasonality, we check by regressing the difference in retail on dummies for every month, we get an R^2 of 0.9469. This clearly shows that we have a seasonal pattern in the series as dummies alone manage to explain quite well the differences. This is confirmed by the Wald test: 148.6060 with p-value = 0.0000 which restricts that every dummy coefficient to be equal to zero. Here we again notice the difference in the large month of December. As can be observed in Figure 7 in the Appendix, from the autocorrelations and partial autocorrelations, we see a very similar pattern to period one, hence, the same conclusions can be made.

4 Methodology

In order to forecast time series data, we will use different methods stating with the linear model then moving to Holt-Winters method which takes into account some of the basic properties of this type of data. We will then look into ARIMA and SARIMA which are both very specific to time series data as they make use of past observations. Finally, we will investigate neural networks, first applying ANN and RNN to then implement the hybrid methods.

4.1 Classical Econometric Models

Linear model A normal linear regression using time trend and seasonal dummies. This would simply be able to account for the non-stationarity using the trend and try to differentiate between each month using dummies. The Ordinary Least Squares (OLS) regression is used as a base model to provide a method of comparison to other methods.

Holt-Winters Exponential smoothing is an extension of a simple moving average window as it weights observations such that more recent observations have a higher value than previous ones. This is very useful in time series as intuitively, the last value has a stronger correlation than a value two years ago. Winters (1960) expanded upon this concept to include for a linear trend and seasonal changes. Note that there are multiple methods such as additive or multiplicative (see equation below), those will be chosen based upon the ones that provides the best answer.

$$
\begin{aligned}
\hat{y}_{t+h|t} &= (\ell_t + hb_t)s_{t+h-m(k+1)} \\
\ell_t &= \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\
s_t &= \gamma\frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m}
\end{aligned}
$$

where l_t represents the level with $l_0 = y_0$, b_t the trend with initial estimate $b_0 = \frac{1}{L}\Big(\frac{y_{L+1}-y_1}{L} + \frac{y_{L+2}-y_2}{L} + \cdots + \frac{y_{L+L}-y_L}{L}\Big)$ and s_t the seasonal component where

$$
s_i = \frac{1}{N}\sum_{j=1}^{N}\frac{y_{L(j-1)+i}}{A_j} \quad \forall i = 1, 2, \cdots, L \quad \text{where} \quad A_j = \frac{\sum_{i=1}^{L} y_{L(j-1)+i}}{L} \quad \forall j = 1, 2, \ldots, N
$$

Note here that L is the cycle length and that N represents the number of complete cycles present in the data.

Box-Jenkins The autoregressive integrated moving average (ARIMA) model is an extension of the ARMA which includes a differentiation step in order to fix the possible problem of non-stationarity. Whereas the autoregressive model (AR) part of the model links to auto-regressive components which are represented by the ϕ_i coefficients, using lagged values, the moving average model (MA) part uses the residuals from a moving average window (θ_i). The model can be observed below:

$$
y_t = c + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t
$$

Box-Jenkins then introduced an approach to building the ARIMA model (Box et al., 1970). It follows three steps; identification, estimation and diagnostic checking. In the identification step, we aim to understand the underlying traits of the data, this will be done by using autocorrelations and partial-autocorrelations plots of the data. We will then be able to identify if differencing is needed but also the MA and AR terms. An interesting estimator for parameter selection is the Akaike information criterion (Akaike, 1974) as it is an in-sample fit to estimate likelihood of a model to predict future values, giving a good tradeoff between a model that fits the data well without overfitting. In the estimation step, we use maximum likelihood to estimate the parameters of the model. Finally, in the diagnostic step, we look mostly at the residuals and check different plots of them to identify if aspects such as serial correlation are still present in the error terms. The model is then names as follows: ARIMA(p,d,q).

Another use of the ARIMA is by expending it to make it more specific to the seasonal effect, hence, if we believe that the autocorrelation is strong with the seasonality of the series. It works in the same way as the ARIMA but includes only the twelfth term. The model is then names as follows: SARIMA(p,d,q)(P,D,Q)$_{12}$.

$$\underset{\uparrow}{(1-\phi_1 B-\cdots-\phi_p B^p)}\ \underset{\uparrow}{(1-\Phi_1 B^{12}-\cdots-\Phi_P B^{12})}\ \underset{\uparrow}{(1-B)^d y_t}\ =\ \underset{\uparrow}{(1+\theta_1 B+\cdots+\theta_q B^q)}\ \underset{\uparrow}{(1+\Theta_1 B^{12}+\cdots+\Theta_Q B^{12})\varepsilon_t}$$

$$\underset{\text{AR}(p)}{}\qquad\qquad\underset{\text{SAR}(P)}{}\qquad\qquad\underset{d\text{ differences}}{}\qquad\qquad\underset{\text{MA}(q)}{}\qquad\qquad\underset{\text{SMA}(Q)}{}$$

4.2 Neural networks

Artificial neural networks Artificial neural networks (ANN) are models based on the biological representation of the neurons in the brain with the aim of recognizing patterns.

In Figure 3, we can see all the input data, each following circle is known as a perceptron. The idea is that each level of perceptrons work as layers, and the more layers are included the more sophisticated the ideas conveyed by the network are, it's important to note however that the adding layers has a very large computational impact on the model. In most research it is generally agreed to use a three-layer feed forward network, hence a single hidden layer with eight perceptrons for sake of replicability of (Alon et al., 2001). The general approach for a neural network is one where at each iteration, we give the network the inputs. All the inputs are connected (by weights) to the perceptrons which themselves are connected to the output. These weights are updated after each iteration by finding the mean squared error ($MSE = \frac{1}{n}\sum_{t=1}^{n}(e_t)^2$) and trying to minimize it for the next iteration. This is done through the learning algorithm which finds the minimum value of the gradient for the function. Hence, returning the update to each weight to improve the prediction performance. It's important to note that usually, a learning rate is applied to avoid overfitting, in this case we will check which learning rate gives the best results. The model would therefore have the following representation:

$$y_t = \alpha_0 + \sum_{j=1}^{q} \alpha_j f\left(\sum_{i=1}^{p} w_{ij} x_{it} + w_{0j}\right) + \varepsilon_t$$

In the equation above, p is the number of input nodes ($i = 0, 1, 2, ..., p$). In our case we will use the input data to be an X matrix that holds all dummy variables and the trend, hence the matrix will hold 12 values at each time t. In each neural network, we can add a large number of hidden layer, q represents that number ($j = 0, 1, 2, ..., q$) and finally, y_t is the output. Note that the weights (w_{ij}) and bias terms α can be randomly distributed between -0.5 and 0.5 and by performing the neural network, they will be adjusted throughout the process in order to fit to the observation given, hence,

providing more accuracy to the outputs. However, in this paper we will use, we will perform the Nguyen & Widrow (1990) method which consists of adjusting the weights so that they are set in their own interval. The center of that interval is found at $x = -w_{0j}/w_{ij}$. It is useful to use as it reaches the target error faster (Wayahdi et al., 2019) and get faster accuracy (Christyaditama et al., 2019), this is due to the fact that when randomly assigning errors, it would be possible to get extremely small values for some weights and by using the Nguyen-Widrow, we allow the weights to be set in a more identical manner (Mishra et al., 2014).

The function $f()$ shown in the previous equation represents the sigmoid activation function for the outcomes of each perceptron such that they range between 0 and 1. $f(x) = \frac{1}{1+exp(-x)}$

This therefore means that, the neural network performs a nonlinear function using the past observations and the weights.

$$y_t = f(y_{t-1}, y_{t-2}, \cdots, y_{t-p}, w) + \varepsilon_t$$

In order to replicate the paper by Alon et al., when training the algorithm, we used a process called Bayesian regularization backpropagation to update the weights and bias values. This makes use of the Levenberg-Marquardt optimization function in order to minimize the error. However, compared to the standard optimization process, the addition of the Bayesian regularization is to minimize the linear combination of the squared errors with the weight while making sure that the neural network keeps good generalization properties (MacKay, 1992; Foresee & Hagan, 1997).

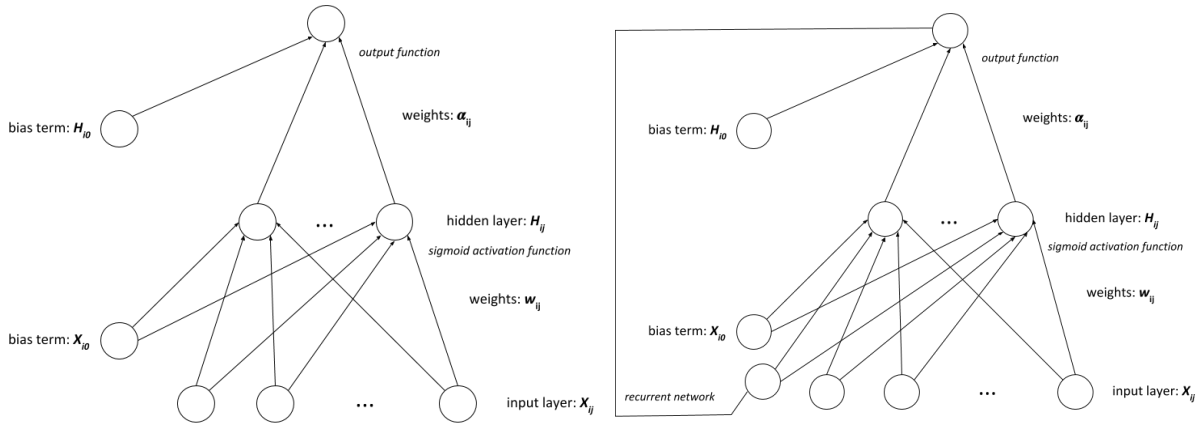

Figure 3: Left: ANN structure, Right:RNN structure

Recurrent neural networks The recurrent neural networks (RNN) work in a similar fashion to that of the ANN. The main difference is in the fact that the output given by the previous run of the RNN will be used as one of the inputs into the network (see Figure 3). This means that the neural

network learns from its output as it is included back into the inputs for the network meaning that the weights are established and that possibly the previous output could give indications of how well the model is working.

Hybrid In literature, the hybrid method was first introduced by G. P. Zhang in 2003 and much of the methodology is based on it. The aim of making a hybrid model is to combine both assets of each method. First of all, by performing an ARIMA, we would be able to understand the underlying linear aspects, find the trend and seasoning. The errors should in that case be much less correlated between each other and follow less of a pattern. For that reason, we then apply a neural network to try and explain the residuals. This approach will be done with both the ANN and RNN. With a special focus on trying to understand if applying an RNN has advantages over using an ANN. The model will therefore be split: $y_t = L_t + N_t$ where L_t denotes the linear component and N_t the non-linear which would also be the residuals of the linear component.

We will then use the residuals of the linear component and see if those can be predicted using the non-linear model. As such, the neural networks will be the following function: $(e_t = f(e_{t-1}, e_{t-2}, \cdots, e_{t-p}, w) + \varepsilon_t)$. In this case, the residuals will be those from the ARIMA model. The final prediction will be made by: $\hat{y_t} = \hat{L_t} + \hat{N_t}$. Note that in here the $\hat{L}$ refers to the estimate of the ARIMA and $\hat{N}$ to that of neural network.

This specific method is called the additive hybrid as to find the N_t component, we use $N_t = y_t - \hat{L_t}$, however, there are other methods such as the multiplicative in which the residuals of the linear model, component you later estimate with the neural network which are calculated by $N_t = y_t/\hat{L_t}$. We will compare both approaches because the standard usage seems to be of the additive methods, however, in L. Wang et al. (2013) they use the multiplicative method which outputs better predictions.

Finally, we will also use a hybrid which consists of including the residuals from the linear method as inputs in the neural network. This would be done by including the residual term for each time period, hence, we would therefore have 13 inputs. The idea with this is to see if by including the residual, there is still a strong link between the residual of the econometric model and the final prediction. This will be interesting to use with different techniques such as linear model, ARIMA or SARIMA as they we will be able to understand if a strong link can be made from the error term to the true output value.

4.3 Model evaluation measures

In general, the most common method of evaluation for forecast accuracy is the mean absolute percentage error (MAPE). However, we will try to add some additional techniques that circumvent some of the issues linked with MAPE (Chen et al., 2017). Throughout the evaluation measure we will use y_t to denote the true values and $\hat{y}_t$ to denote the predicted output. The error will be calculated as followed: $e_t = y_t - \hat{y}_t$.

MAPE & sMAPE Using the mean absolute percentage error (MAPE) allows comparison between data sets as the errors are proportional to their original value. Despite being a popular measure, MAPE is a measure which has some quite special properties and can go terribly wrong when specific values are allowed. For example, if the true value is 0, this would create an error for the metric as one cannot divide by zero. Additionally, while there is no upper bound on the metric, there is a clear lower bound at 0. Hence, to remedy this problem, we can use symmetric MAPE (sMAPE) which is also more resistant to outliers (Chen et al., 2017).

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{\mid (e_t) \mid}{\mid y_t \mid}, \quad sMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{2 \mid (e_t) \mid}{(\mid y_t \mid + \mid \hat{y}_t \mid)}$$

MBRAE & UMMBRAE When trying to evaluate the performance of a model, metrics such as RMSE or MAPE are often very useful, however, it's usually good to have another model to be able to evaluate how well the model actually estimated. Using the same, idea, relative metrics allow to directly compare two model, hence, to understand which one is better. This is also used with simple models for time series such as the naive or random walk. Hence, if we use mean relative absolute error (MRAE), we directly compare the errors of each model. A new method we will use mean bounded relative absolute error (MBRAE) and UMBRAE from (Chen et al., 2017) as explained through the papers, this metric is more robust. Note that as MBRAE lacks in interpretability, UMBRAE is the the metric used to remedy this issue.

$$MRAE = \frac{1}{n}\sum_{t=1}^{n}\frac{\mid e_t \mid}{\mid e_t^* \mid}, \quad MBRAE = \frac{1}{n}\sum_{t=1}^{n}\frac{\mid e_t \mid}{\mid e_t \mid + \mid e_t^* \mid}, \quad UMBRAE = \frac{MBRAE}{1 - MBRAE}$$

where y_t is the true value, $\hat{y}_t$ and $\tilde{y}_t$ are the predicted values by the models we compare and $e_t = y_t - \hat{y}_t$ and $e_t^* = y_t - \tilde{y}_t$. If $UMBRAE < 1$, then the y_t performs $(1 - UMBRAE)100\%$ better than the y_t^* whereas if $UMBRAE > 1$ then it performs $(UMBRAE - 1)100\%$ worse. If it's equal to 1, then the methods are relatively similar.

In the next following tests we will use $d_t = \varepsilon_t^2 - \varepsilon_t^{*2}$ where ε_t^2 and ε_t^{*2} represents the percentage forecast error of the two models being compared.

Wilcoxon signed-rank test Wilcoxon's signed-ranks test is based on the number of times there is a positive difference versus a negative difference in the values. The null hypothesis is that the difference between the pairs follows a symmetric distribution around zero, hence, that the forecasts are not statistically different. The test statistics is calculated as follows:

$$SR = \sum_{t=1}^{n} I_+(d_t) rank(\mid d_t \mid) \text{ where } I_+(d_t) = \begin{cases} 1, & \text{if } d_t > 0 \\ 0, & \text{otherwise} \end{cases}$$

When scaled as shown in the equation below, we assume the test follows a normal distribution.

$$\frac{SR - N(N+1)/4}{\sqrt{N(N+1)(2N+1)/24}} \sim N(0,1)$$

Diebold Mariano test The Diebold Mariano (DM) test also checks between two forecasts if one is more accurate than the other under the null hypothesis that $\bar{d} = 0$. In the equation below P is the number of forecast observations.

$$DM = \frac{\bar{d}}{\sqrt{V(\hat{d}_{t+1})/P}} \sim N(0,1), \quad V(\bar{d}_{t+1}) = \frac{1}{P-1} \sum_{t=T}^{T+P-1} (d_{t+1} - \bar{d})^2$$

5 Results

The results in the following section were performed on Matlab version R2020a using the Econometrics, Deep Learning and Optimization Toolbox expect the Holt-Winters procedure which was executed on Eviews 10. Additionally, we will specify the specific parametrization of each method and when looking into each method we will look at both period one and two as they often have similar attributes. Note that Table 3 in the Appendix note the specifications of the results and Table 4 in the Appendix represents all the results.

5.1 Classical Econometric Models

Linear model This model is the most basic one, the explanatory variables all 11 dummies for the months, the trend variable and a constant. In both period one and two, we have an R^2 around 0.97. This model already has a very good accuracy and results in MAPE values of 3.2820 and 3.9583 respectively. This model already has better accuracy than the random walk which is a very strong model generally in time series predictions. When looking into rolling window observations, we can

notice a good improvement in the MAPE values of 2.922 and 3.483. However, this is expected as we give it more data to create a more accurate model, hence, this is not too surprising.

Holt-Winters When performing the Holt-Winters procedure, there are multiple aspects to check for model optimization, we choose the optimize with Log-likelihood objective and have a convergence of 0.0001 while limiting the number of iterations to 500. Then, as we can recall from methodology, we have either additive or multiplicative model. For period one, the best specification was a model with additive error, trend and multiplicative season which gave a MAPE value of 1.5247 whereas for the second period, the best model was when all the components were set to additive which had a MAPE value of 2.3459. Note that as we performed the rolling window, we had a slightly different performance for the first period with a MAPE of 1.6210 and for the second 2.171. The accuracy of the forecasts using Holt-Winters are much better than those of the linear model, however, this makes sense as the Holt-Winters procedure is specialized for time series data that has the characteristics described in the Data section. It is however interesting to note that the period one has better forecast despite being a less stable period.

Box-Jenkins This procedure is much more appropriate to time series than linear model and we tested different combinations also due to further research to for the hybrid components. As found in the Data section, we noticed some clear autocorrelations in the variable, additionally we found some possible patterns that we could explore further. However, in order to find the best possible predictions, the models have been chosen according to the best MAPE values possible.

For the ARIMA models, we allowed the model to vary by allowing AR and MA terms up to 12 and a difference up to 4. For the SARIMA models, we allowed the same changes in AR, MA and difference. However, for season specific, we limited to one differentiation, SAR and SMA term. Hence, for the first period, we chose the following ARIMA(11,2,9) which has a MAPE of 1.8653 compared to 1.4597 for the SARIMA$(11,2,3)(1,1,1)_{12}$. This already shows the improvements of using a seasonal model. For both models, when analyzing the residuals, we can note that both models seem to be correctly specified as we cannot observe any significant autocorrelations in the residuals or square of residuals which therefore follows the assumption of homoskedasticity. Finally, when performing the Jarque-Bera test, they both seem to not reject the assumption of a normal distribution with p-values at least greater than 0.5. These tests therefore confirm that the model seems well specified and follows some of the assumptions required. When observing the second period, we can first observe that the results are slightly lower in the second period as oppose to the

first. With a MAPE of 1.7382 for ARIMA(12,1,5) and 1.1839 for SARIMA$(7,0,2)(1,1,1)_{12}$. When conducting an analysis on the residuals like on period one, we arrive at similar conclusions and therefore can assume that the models are correctly specified, and assumptions validated.

It is interesting to see here the SARIMA models always beat the Holt-Winters procedure whereas the ARIMA is worse in period one and better in the second. This clearly shows that adding the seasonal component is important. Additionally, as opposed to the two previous methods where period one had a lower MAPE than period two, in the ARIMA and SARIMA methods, the forecasts are much more accurate for period two, which follows the assumptions we could have made since period two being more stable.

Note that the models chosen for each specific period will be kept fixed throughout the rest of the paper, hence, in the hybrid methods, when using the ARIMA of period one, it will be the same one as describes in this section.

5.2 Neural Networks

Artificial neural networks In order to choose which neural network specifications we would determine for the rest of the models, we decided to first perform some tests in order to find the optimal settings for each model. First of all, the training versus validation period, we decided to opt for an 80-20 split across both period one and two and a maximum number of epochs set to 2000 in order to keep the results as comparable as possible. Additionally, as mentioned in the methodology, we only have one hidden layer made of 8 perceptrons and also making use of the sigmoid functions. To find the best learning rate, we tried from 0.01 to 0.91 increasing by 0.1 at each iteration. The only difference in the two setups of the networks between both periods is the leaning rate. Whereas in period one we use a learning rate of 0.21, in period two, the optimal results were with a learning rate of 0.01. Throughout the rest of the research, these exact specifications will be used each time for each period.

Finally, as we know, since the weights are set randomly at the start, in order to find more truthful results, each time we performed ten iterations in order to find an average MAPE and minimum MAPE value. We will discuss further the implications of the average versus minimum performance in a later section.

For the first period we got a minimum and average MAPE value of 1.3926 and 2.2594 and for the second period one of 1.5826 and 2.7765. These are noticeably better than the ARIMA models previously discussed but perform quite poorly compared to SARIMA.

RNN The same specification for the setup of the RNN are used as in the ANN. This is done in order to have better to keep the models as comparable to one another. The only setting which we adapted is the learning rate, 0.41 for period one and 0.31 for period two. It is interesting to note that those are for both periods exactly 0.2 higher, however, whether this is pure coincidental is up to further research.

When looking at period one, we have a minimum and average MAPE value of 1.4087 and 2.7239 and for period two MAPE values of 1.6928 and 2.5665. These results are a bit surprising as we would have expected the RNN to work better than the ANN as it is supposed to be better adapted for time series due to the fact it uses the last output back as input. However, we can notice that the average RNN MAPE values are lower. This could indicate that indeed the RNN is normally better and possibly the weights were just better adjusted for one of the iterations and therefore allowed the network to find a better fit for the observations. It will be interesting to see if this pattern repeats when comparing between the hybrid models using ANN or RNN.

Hybrid In the hybrid section, we will look at all the different possible model combinations we made Table 1. In the hybrid models, if we took one econometric method and one neural network, we would have three different possibilities to combine them together. They are presented by mentioning the name of the econometric model which was combined with the neural network, where a "+" and "∗" afterwards denotes the additive and multiplicative hybrid and the additional "resid" would be the model in which the residuals are introduced in the inputs. The values presented are the minimum MAPE values of each model.

As we can observe from the results, it seems quite clear that making use Linear Model is worse when combines with the additive or multiplicative method. However, when combining it with by including the residuals as inputs, we get outstandingly good results with the lowest MAPE values of any model close to 0. This would imply that while the linear model does not get very good results, the error terms still have quite enough meaning behind them such that the neural network can pick up on pattern and therefore the neural networks in these situation makes up for the worse econometric model. We will delve into comparison of each method respectively later on, specifically comparing the two neural networks.

Table 1: Hybrid MAPE results

		ANN								
		LM +	LM *	LM resid	ARIMA +	ARIMA *	ARIMA resid	SARIMA +	SARIMA *	SARIMA resid
Period one	min MAPE	2.0170	3.2201	0.0006	1.7989	1.7210	1.3082	1.3589	1.4347	1.1828
	avg MAPE	2.4274	3.3190	0.0250	2.1278	1.8044	1.5930	1.8486	1.4602	1.4104
Period two	min MAPE	1.4162	3.9184	0.0005	1.4651	1.6237	1.7650	1.1834	1.1812	2.0255
	avg MAPE	3.3030	3.9543	0.0008	1.6509	1.7126	3.3437	1.4674	1.1960	2.7150

		RNN								
		LM +	LM *	LM resid	ARIMA +	ARIMA *	ARIMA resid	SARIMA +	SARIMA *	SARIMA resid
Period one	min MAPE	1.6142	2.2543	0.0014	1.8575	1.7326	1.2947	1.4112	1.4068	1.2156
	avg MAPE	2.3745	3.1941	0.0260	2.1326	1.8390	2.3821	1.7650	1.4466	2.9791
Period two	min MAPE	1.4198	3.8442	0.0007	1.4815	1.5413	1.5660	1.2168	1.1824	1.8527
	avg MAPE	2.6082	3.9106	0.0014	1.8176	1.6570	2.5181	1.5457	1.2063	2.4788

5.3 Comparison of models

5.3.1 Robustness

When using neural network, there is a large dependency of the final outcome upon the initial weight used. Despite the use of the Nguyen-Widrow method, we can still observe large differences at each new estimation of the neural networks. In order to see the differences between the model with the best outcome and the average of the ten iterations, we will take the $1 - \frac{min.\ MAPE}{avg.\ MAPE}$, meaning that the smaller the value, the better the robustness as there is a smaller difference between the average and minimum value. As we can observe in Figure 4, most of the time we will get similar idea in period one or two. As we saw in the hybrid section, we get the best results when including the residuals, however, it also turns out that those are when we have the highest ratio between the two, hence making them the least robust method. On the other hand, we can clearly observe that in the multiplicative method, we often have values that are close to being the same each time as the average does not seem to be very different from the minimum and this can clearly be observe whether we use the linear model, ARIMA, SARIMA or ANN and RNN.

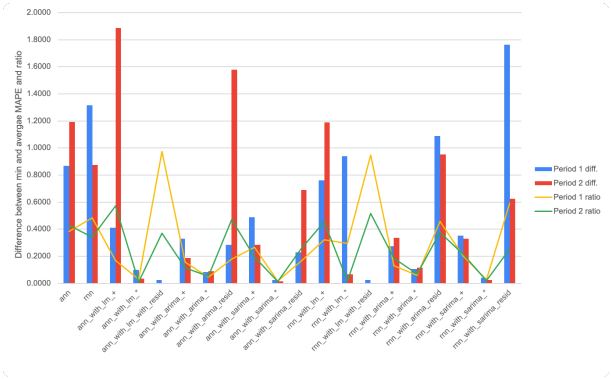

Figure 4: Bar chart of robustness of models

5.3.2 Comparing each method as groups

LM When observing all the results that include the linear model (see Figure 8 in Appendix), we can clearly notice that the hybrid including the residuals in the model has the best performance, which is significantly better than the additive model when comparing period one as we have a p-value of 0.0094 for DW and 0.0022 for SR, both rejecting the hypothesis of equal forecasts. Additionally, here, we can observe some possible overfitting as the R^2 when using the additive methods is quite high except for period one with the RNN whereas its predictive accuracy is noticeably better when compared to the ANN.

ARIMA While for period one, we can observe that including the residuals seems to clearly improves the prediction values, for period two, the best model is the additive one (see Figure 9 in Appendix). When comparing the ANN additive method to the simple ARIMA for the second period, we still find that the hybrid is significantly better with a p-value of 0.0316 and 0.0499 for DW and SR respectively, hence, rejecting null hypothesis of equal forecasts. Notice that when looking at period two including residuals, they have a really high R^2, however, suffer in the forecast in comparison to the additive method. It seems in this case they suffer from an overfitting issue. Furthermore, as we look into some specifics, we can see that the MAPE and UMBRAE values sometimes differ in the ranking of which methods are better, specifically for period one where the UMBRAE metric sometimes even prefers the simple ARIMA model compared to the hybrid specification.

SARIMA When using the SARIMA residuals to create a hybrid model, it would seem that we encounter similar problems as with ARIMA. Whereas period one confirms the general aspect that using residuals are often the better method, in period two, the forecasts are much worse (see Figure

10 in Appendix). This also seems to follow the pattern we observed with ARIMA as higher R^2 often leads to worse forecasts, hence, indicating that there could be overfitting.

Additive models and Multiplicative models The results in the additive and multiplicative seem to indicate similar patterns, in the additive we can notice that SARIMA seems to give the best accuracy, especially in period two as in period one, the UMBRAE metric indicates that the best forecast model is ANN hybrid with ARIMA for additive (see Figure 11 in Appendix). This can also be observed as while both SR and DW do not reject the hypothesis of equal forecasts, the sign of both values being different with -0.5491 and 1.1456 respectively indicates that the SR test does believe the SARIMA hybrid to be better while DW believe the ARIMA hybrid to be. When comparing the ANN or RNN, it's very difficult to draw a clear conclusion as whereas in the multiplicative method, the RNN outperform the ANN all the time, in the additive method, it's more indecisive.

Residual models The graphs observed in Figure 13 in the Appendix are very representative, we can clearly see that using the linear model in the hybrid context strongly improves the accuracy of forecasts compared to using more complex econometric models. Whereas for period one, we could make a better use of an ANN model, for period two, it would seem an RNN is more appropriate, however, the measures are quite similar.

Results summary To have a good understanding of all the results, Table 2 in Appendix shows which methods out of all three hybrid had the best MAPE value. Whereas LM, Holt-Winters, and ARIMA seem to be easily outperformed by the neural networks, SARIMA has strong performance and notably even clearly outperforms both ANN and RNN. When looking into the hybrids, a clear wining technique seems to be using the residuals of the linear model with a neural network. For each period, these outperform any other method and give MAPE values which are extremely close to zero. However, as we have seen in the paragraph about robustness, these methods also vary a lot depending on the initial weights. So even though these methods perform really well, they might need to be iterated multiple times to get the best results. Additionally, we noticed these iterations usually took slightly longer to run, which could be understood as the neural network was finding more patterns, and therefore was better able at explaining the residuals. This is also the reason we believe it performed best as since in the linear model with MAPE values averaging 3.6197, we clearly were not capturing all the different changes and we believe this is what the neural network

was able to pick up upon.

When observing more advanced econometric techniques, we noticed that using residuals would not always give the best result. Specially in period two, we found that in this more stable period, the best forecasts were actually made by the additive or multiplicative methods which gave noticeable improvements to ARIMA. When we observe specifically the improvements made on the hybrid SARIMA, they are very minor and done using the multiplicative method. This could be explained as this method is very consistent and always give slightly better forecasts. We can also observe that with some exceptions, usually the additive method has more accurate forecasts compared to the multiplicative method. This goes against the results found in L. Wang et al. (2013), however, it is noted that for short term forecasts, the multiplicative method does not work as well. It would therefore be interesting to increase the number of forecast observations to determine if that holds true. Hence, it might not always be useful to have a hybrid when the initial estimation is already very strong, but if improvements were highly required, the use of the multiplicative method could prove to improve slightly the results.

Finally, to answer the research question, we will now look into the differences between the two neural networks. Our aim in this paper was to try and find a better forecast for retail sales, however, we also wanted to see if using an RNN would improve on the forecasts. As can be seen throughout the results, and even more noticeably in Table 2, the ANN outperforms slightly (yet not significantly, as when performing DW or SR test, the null hypothesis was never rejected) the RNN nearly all the time. This is a bit surprising as the RNN uses an additional input, the output of the previous iteration, hence, we would have expected a better result from RNN. We believe this might have happened due to the fact that we used little data for both the estimation but also the number of predictions we made, hence, possibly not allowing the networks to fully make use of the additional inputs.

6 Conclusion

Overall throughout this paper we have been able to significantly improve the forecasting performance. This is very important for business owners, for the entire market and economy as being able to understand these concepts allow for better planning. What we have noticed is that in the econometrics methods, using the SARIMA allowed to get very strong results in both periods. Additionally, in more stable times, for the second period, it even outperformed the neural networks

which could be predictable as this would better fit the model assumptions. As noticed in the robustness analysis, one of the main issues was the differences between each result when running the same network. This came about due to initial values of the weights and bias. Despite the use of the Nguyen-Widrow method, values differed greatly hence why we conducted a robustness analysis. This allowed us to see how each method performed differently and which were more consistent. Whereas the hybrid models were generally better than the simple econometric methods they did not always outperformed the neural networks. However, when combining the best methods, clearly using hybrids gave a much better outcome. The method which clearly surpassed any other was the use of a linear model, using those residuals as inputs into the network and trying to predict the true value of the forecasts. This gave MAPE values averaging 0.0033 for all periods and different neural networks as opposed to an average of 1.1905 for the best SARIMA hybrid methods, so about 361 times better. We believe this method performs this well as when doing a linear regression, it does not capture all the correlations and possible connections within the data, hence, in the residuals, there is still links that are strong enough such that the neural networks are able to correctly predict the true values. In this method is also where we have the highest differences between the minimum and average MAPE which shows that the network is therefore highly dependent on the initial value, and if set well, could lead to a very accurate model.

Limitations One of the many issues we have found throughout this paper is that we only have 208 observations and we also split the data as two different period emerge. Neural networks usually require a large number of observations as the model learn by fitting to all the cases, hence, the more data we have, the more complex understanding the network can make of the inputs. We believe this is partly why the neural networks do not always outperform the econometrics models, but also why at times, the difference between minimum and maximum MAPE is so large. Possibly, this also helps to explain why the RNN wasn't always able to outperform the ANN, despite its special property which should be of use in time series data.

For further research, it would be interesting to see how these methods perform on different data sets, still in time series, but possibly with more observations and less seasonal data. It would also be interesting to see if we increase the number of iterations, how close to a MAPE of zero the hybrid of linear model with neural network can get to. It's important to note here that by adding more data, the latter may start taking much longer to estimate as it was already one of the slowest models and adding new data may lengthen the training stage.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, *19*(6), 716–723.

Alon, I., Qi, M., & Sadowski, R. J. (2001). Forecasting aggregate retail sales:: a comparison of artificial neural networks and traditional methods. *Journal of retailing and consumer services*, *8*(3), 147–156.

Ansuj, A. P., Camargo, M., Radharamanan, R., & Petry, D. (1996). Sales forecasting using time series and neural networks. *Computers & Industrial Engineering*, *31*(1-2), 421–424.

Box, G. E., Jenkins, G. M., & Reinsel, G. (1970). Time series analysis: forecasting and control holden-day san francisco. *BoxTime Series Analysis: Forecasting and Control Holden Day1970*.

Chen, C., Twycross, J., & Garibaldi, J. M. (2017). A new accuracy measure based on bounded relative error for time series forecasting. *PloS one*, *12*(3).

Christyaditama, I. G. P., Candiasa, I. M., & Gunadi, I. G. A. (2019). *Optimization of artificial neural networks to improve the accuracy in predicting the selection of competency competencies of vocational students using nguyen-widrow* (Tech. Rep.). EasyChair.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Faruk, D. Ö. (2010). A hybrid neural network and arima model for water quality time series prediction. *Engineering applications of artificial intelligence*, *23*(4), 586–594.

Foresee, F. D., & Hagan, M. T. (1997). Gauss-newton approximation to bayesian learning. In *Proceedings of international conference on neural networks (icnn'97)* (Vol. 3, pp. 1930–1935).

Hill, T., O'Connor, M., & Remus, W. (1996). Neural network models for time series forecasts. *Management science*, *42*(7), 1082–1092.

Ho, S.-L., Xie, M., & Goh, T. N. (2002). A comparative study of neural network and box-jenkins arima modeling in time series prediction. *Computers & Industrial Engineering*, *42*(2-4), 371–375.

Khashei, M., & Hajirahimi, Z. (2017). Performance evaluation of series and parallel strategies for financial time series forecasting. *Financial Innovation*, *3*(1), 24.

Kihoro, J., Otieno, R., & Wafula, C. (2004). Seasonal time series forecasting: A comparative study of arima and ann models.

Knoop, T. A. (2009). *Recessions and depressions: understanding business cycles: understanding business cycles*. ABC-CLIO.

Kohzadi, N., Boyd, M. S., Kermanshahi, B., & Kaastra, I. (1996). A comparison of artificial neural network and time series models for forecasting commodity prices. *Neurocomputing*, *10*(2), 169–181.

Kumar, D. N., Raju, K. S., & Sathish, T. (2004). River flow forecasting using recurrent neural networks. *Water resources management*, *18*(2), 143–161.

Kuo, C., & Reitsch, A. (1995). Neural networks vs. conventional methods of forecasting. *The Journal of Business Forecasting*, *14*(4), 17.

Labonte, M., Makinen, G., Government, & Division, F. (2002). The current economic recession: How long, how deep, and how different from the past?..

Luxhøj, J. T., Riis, J. O., & Stensballe, B. (1996). A hybrid econometric—neural network modeling approach for sales forecasting. *International Journal of Production Economics*, *43*(2-3), 175–192.

MacKay, D. J. (1992). Bayesian interpolation. *Neural computation*, *4*(3), 415–447.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115–133.

Mishra, K., Mittal, N., & Mirja, M. H. (2014). Image compression using multilayer feed forward artificial neural network with nguyen widrow weight initialization method. *International Journal of Emerging Technology and Advanced Engineering*, *4*(4).

Nguyen, D., & Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *1990 ijcnn international joint conference on neural networks* (pp. 21–26).

Sharda, R., & Patil, R. B. (1992). Connectionist approach to time series prediction: an empirical test. *Journal of Intelligent Manufacturing*, *3*(5), 317–323.

Shui-Ling, Y., & Li, Z. (2017). Stock price prediction based on arima-rnn combined model. *DEStech Transactions on Social Science, Education and Human Science*(icss).

Tang, Z., & Fishwick, P. A. (1993). Feedforward neural nets as models for time series forecasting. *ORSA journal on computing*, *5*(4), 374–385.

Tseng, F.-M., Yu, H.-C., & Tzeng, G.-H. (2002). Combining neural network model with seasonal time series arima model. *Technological Forecasting and Social Change*, *69*(1), 71–87.

Walsh, C. E. (1993). What caused the 1990-1991 recession? *Economic Review-Federal Reserve Bank of San Francisco*(2), 33.

Wang, J.-J., Wang, J.-Z., Zhang, Z.-G., & Guo, S.-P. (2012). Stock index forecasting based on a hybrid model. *Omega*, *40*(6), 758–766.

Wang, L., Zou, H., Su, J., Li, L., & Chaudhry, S. (2013). An arima-ann hybrid model for time series forecasting. *Systems Research and Behavioral Science*, *30*(3), 244–259.

Wayahdi, M., Zarlis, M., & Putra, P. (2019). Initialization of the nguyen-widrow and kohonen algorithm on the backpropagation method in the classifying process of temperature data in medan. In *Journal of physics: Conference series* (Vol. 1235, p. 012031).

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management science*, *6*(3), 324–342.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, *14*(1), 35–62.

Zhang, G. P. (2003). Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, *50*, 159–175.

7 Appendix

7.1 Data

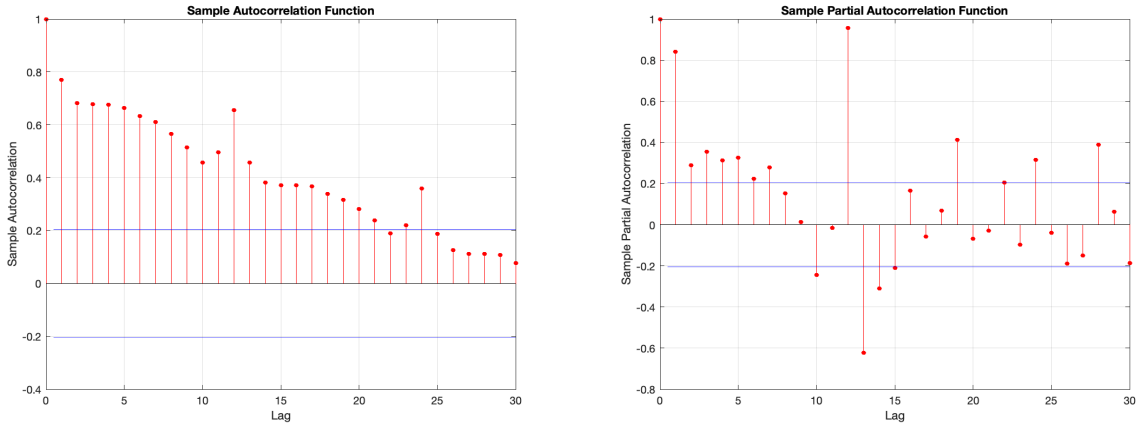

Figure 5: Retail first period autocorrelation and partial-autocorrelation

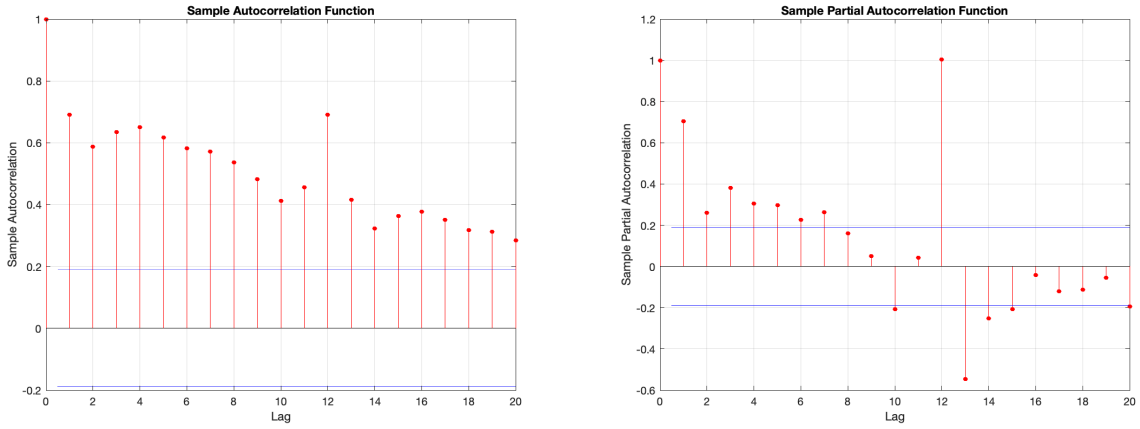

Figure 6: Retail second period autocorrelation and partial-autocorrelation

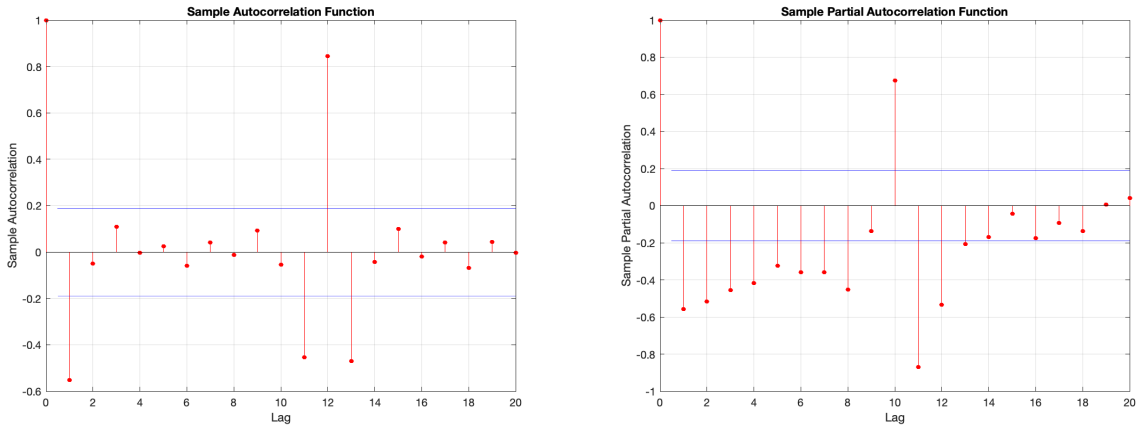

Figure 7: Twice differentiated retail second period autocorrelation and partial-autocorrelation

7.2 Methodology

Holt-Winters additive

$$\hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$$
$$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$$
$$b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}$$
$$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m},$$

7.3 Results

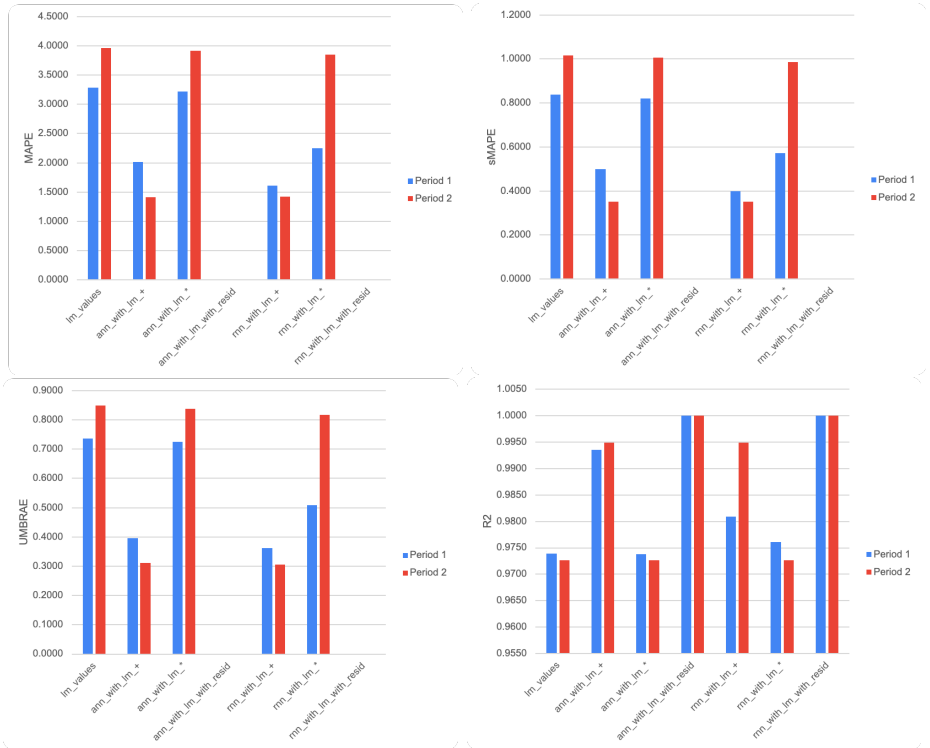

Figure 8: Using Linear Model all methods and metric evaluations

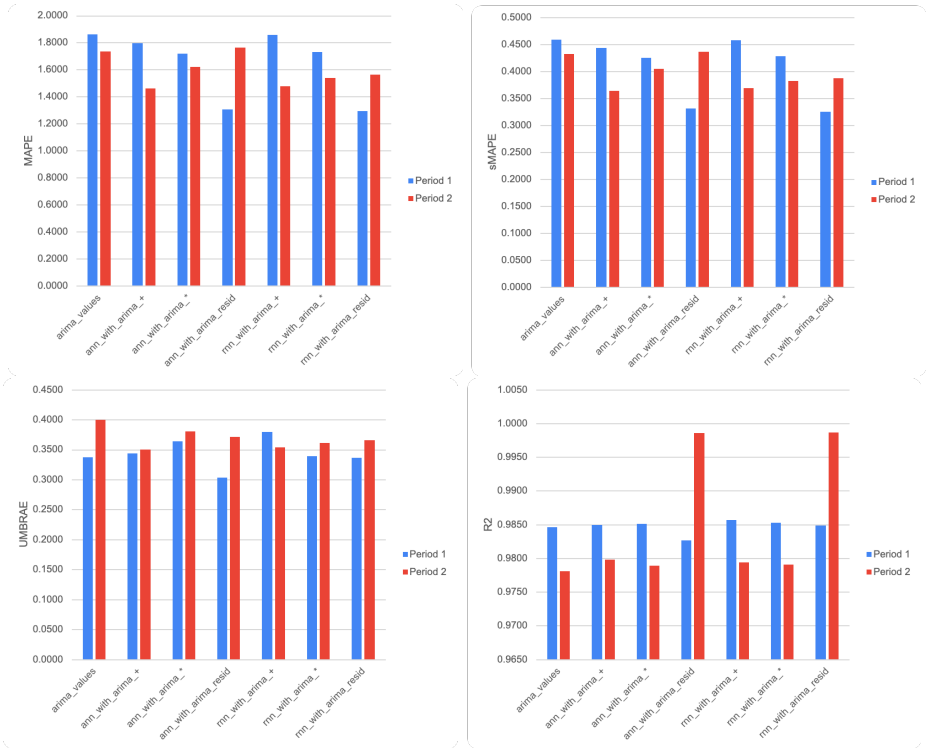

Figure 9: Using ARIMA all methods and metric evaluations

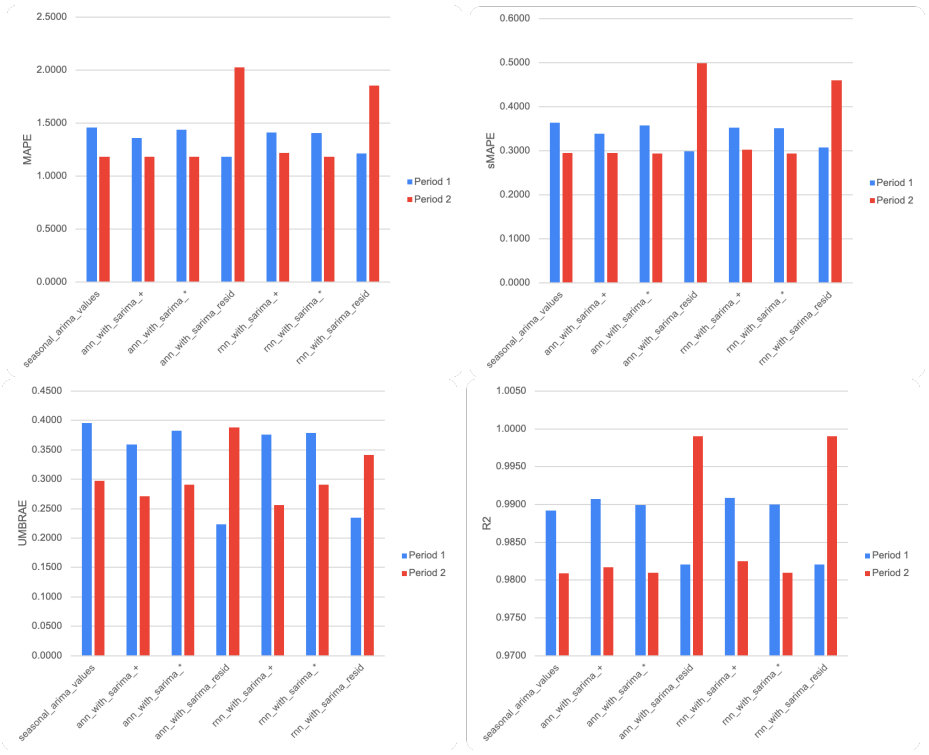

Figure 10: Using SARIMA all methods and metric evaluations

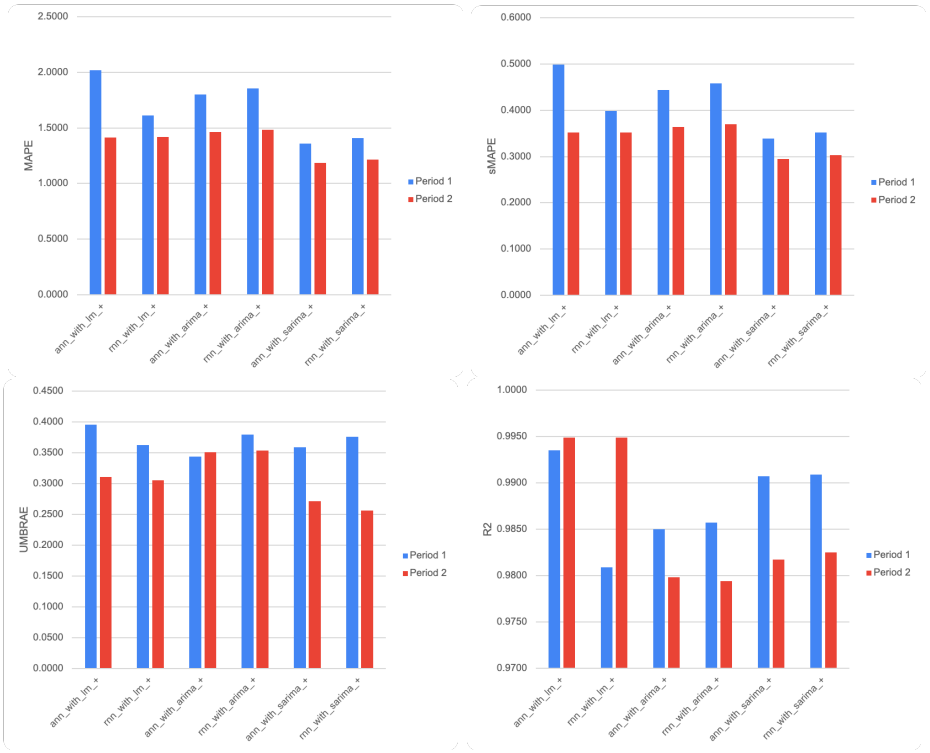

Figure 11: Using additive methods and metric evaluations

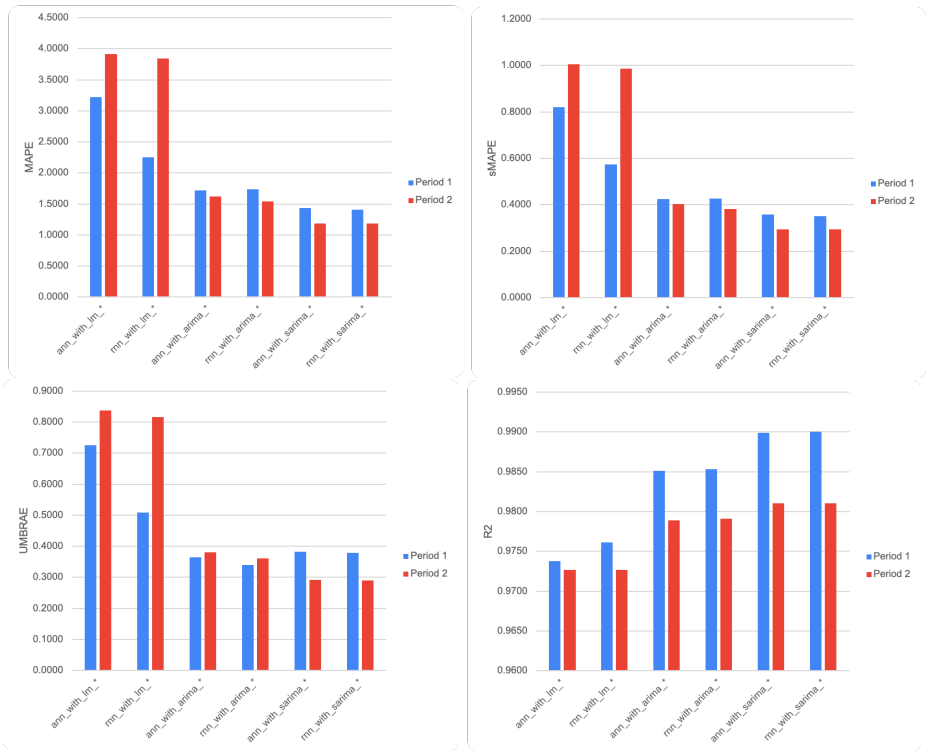

Figure 12: Using multiplicative methods and metric evaluations

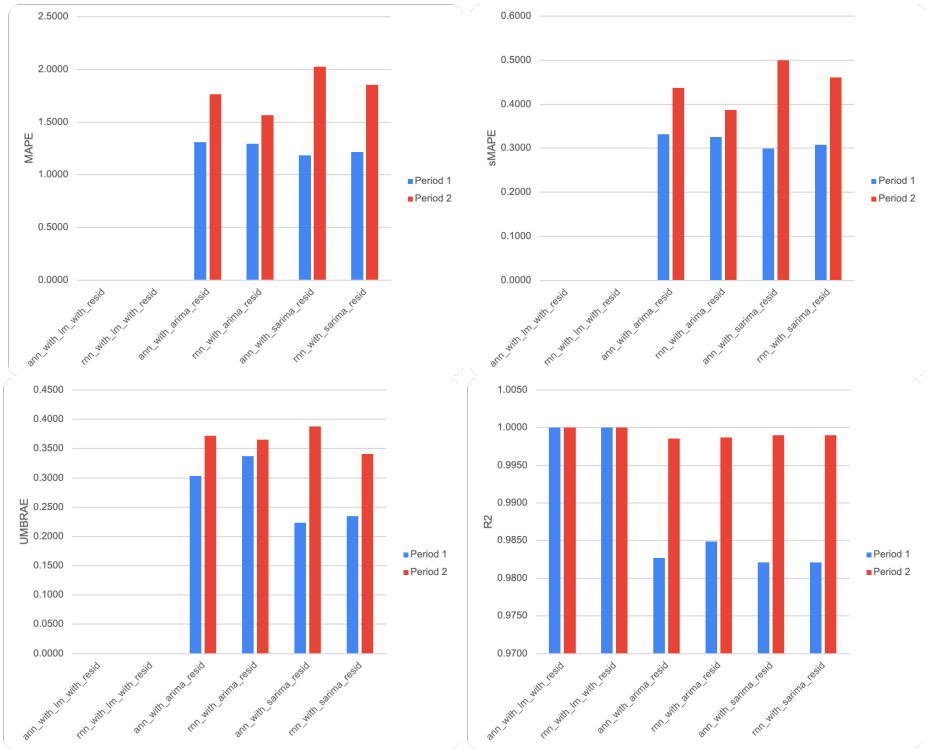

Figure 13: Using the residuals methods and metric evaluations

Table 2: Summary of best results

		LM	Holt-Winters	ARIMA	SARIMA	ANN	RNN	ANN_LM_best	ANN_ARIMA_best	ANN_SARIMA_best	RNN_LM_best	RNN_ARIMA_best	RNN_SARIMA_best
Period one	MAPE	3.2810	1.5247	1.8653	1.4597	1.3926	1.4087	0.0006	1.3082	1.1828	0.0014	1.2947	1.2156
	method							*ann_with_lm_with_resid*	*ann_with_arima_resid*	*ann_with_sarima_resid*	*rnn_with_lm_with_resid*	*rnn_with_arima_resid*	*rnn_with_sarima_resid*
Period two	MAPE	3.9583	2.3459	1.7382	1.1839	1.5826	1.6928	0.0005	1.4651	1.1812	0.0007	1.4815	1.1824
	method							*ann_with_lm_with_resid*	*ann_with_arima_+*	*ann_with_sarima_**	*rnn_with_lm_with_resid*	*rnn_with_arima_+*	*rnn_with_sarima_**

Table 3: Model specifications

	LM	Holt-Winters	ARIMA	SARIMA	ANN	RNN
Period one	no specifics	Error: Additive Trend: Additive Season: Multiplicative	ARIMA(11,2,9)	SARIMA(11,2,3)(1,1,1)	- Learning rate: 0.2100 - Max epochs: 2000 - TrainID: 1:67, ValID: 68:84	- Learning rate: 0.4100 - Max epochs: 2000 - TrainID: 1:67, ValID: 68:84
Period two	no specific	Error: Additive Trend: Additive Season: Additive	ARIMA(12,1,5)	SARIMA(7,0,2)(1,1,1)	- Learning rate: 0.0100 - Max epochs: 2000 - TrainID: 1:80, ValID: 81:100	- Learning rate: 0.3100 - Max epochs: 2000 - TrainID: 1:80, ValID: 81:100
General	On Matlab using "fitlm"	- On Eviews - Optimize with log-likelihood objective - Convergenceof 0.0001 - Max. number of iterations: 500	On Matlab using "arima"	On Matlab using "arima"	- On Matlab using "feedforwardnet" - 1 hidden layer of 8 perceptrons	- On Matlab using "layrecnet" - 1 hidden layer of 8 perceptrons - 1 ouput used as input

	ANN LM +	ANN LM *	ANN LM resid	ANN ARIMA +	ANN ARIMA *	ANN ARIMA resid	ANN SARIMA +	ANN SARIMA *	ANN SARIMA resid
Period one									
Period two									
General	Same specifications as for ANN	Same specifications as for ANN	Same specifications as for ANN	Same specifications as for ANN and ARIMA	Same specifications as for ANN and ARIMA	Same specifications as for ANN and ARIMA	Same specifications as for ANN and SARIMA	Same specifications as for ANN and SARIMA	Same specifications as for ANN and SARIMA

	RNN LM +	RNN LM *	RNN LM resid	RNN ARIMA +	RNN ARIMA *	RNN ARIMA resid	RNN SARIMA +	RNN SARIMA *	RNN SARIMA resid
Period one									
Period two									
General	Same specifications as for RNN	Same specifications as for RNN	Same specifications as for RNN	Same specifications as for RNN and ARIMA	Same specifications as for RNN and ARIMA	Same specifications as for RNN and ARIMA	Same specifications as for RNN and SARIMA	Same specifications as for RNN and SARIMA	Same specifications as for RNN and SARIMA

Table 4: Summary of all results

		LM	Holt-Winters	ARIMA	SARIMA	ANN	RNN
Period one	MAPE	3.2810	1.5247	1.8653	1.4597	1.3926	1.4087
	sMAPE	0.8371	0.3863	0.4589	0.3643	0.3516	0.3556
	RMSE	4516.8000	2420.9000	2768.7000	2118.3000	2274.8000	2282.5000
	MBRAE	0.4238	0.2503	0.2523	0.2834	0.2423	0.2465
	UMBRAE	0.7355	0.3339	0.3375	0.3954	0.3198	0.3271
	R2	0.9739	0.9885	0.9846	0.9892	0.9809	0.9810
Period two	MAPE	3.9583	2.3459	1.7382	1.1839	1.5826	1.6928
	sMAPE	1.0154	0.5760	0.4320	0.2950	0.3897	0.4187
	RMSE	9174.2000	5956.1000	3868.2000	2822.7000	4447.1000	4030.1000
	MBRAE	0.4592	0.2804	0.2856	0.2293	0.2129	0.2672
	UMBRAE	0.8492	0.3897	0.3998	0.2975	0.2705	0.3646
	R2	0.9727	0.9828	0.9781	0.9809	0.9963	0.9960

		ANN LM +	ANN LM *	ANN LM resid	ANN ARIMA +	ANN ARIMA *	ANN ARIMA resid	ANN SARIMA +	ANN SARIMA *	ANN SARIMA resid
Period one	MAPE	2.0170	3.2201	0.0006	1.7989	1.7210	1.3082	1.3589	1.4347	1.1828
	sMAPE	0.4988	0.8212	0.0002	0.4432	0.4254	0.3311	0.3393	0.3576	0.2989
	RMSE	2699.2000	4446.6000	0.9529	2660.6000	2507.7000	2122.3000	2008.0000	2103.3000	2126.6000
	MBRAE	0.2832	0.4204	0.0003	0.2558	0.2671	0.2329	0.2640	0.2769	0.1827
	UMBRAE	0.3951	0.7252	0.0003	0.3438	0.3645	0.3036	0.3588	0.3830	0.2236
	R2	0.9935	0.9738	1.0000	0.9850	0.9851	0.9827	0.9907	0.9899	0.9821
Period two	MAPE	1.4162	3.9184	0.0005	1.4651	1.6237	1.7650	1.1834	1.1812	2.0255
	sMAPE	0.3516	1.0050	0.0001	0.3636	0.4045	0.4366	0.2945	0.2941	0.4991
	RMSE	3142.2000	9107.2000	1.2591	3408.6000	3575.3000	3976.7000	2861.4000	2830.5000	5119.4000
	MBRAE	0.2370	0.4559	0.0002	0.2597	0.2757	0.2711	0.2136	0.2256	0.2795
	UMBRAE	0.3107	0.8378	0.0002	0.3509	0.3806	0.3719	0.2716	0.2913	0.3879
	R2	0.9949	0.9727	1.0000	0.9798	0.9789	0.9986	0.9817	0.9810	0.9990

		RNN LM +	RNN LM *	RNN LM resid	RNN ARIMA +	RNN ARIMA *	RNN ARIMA resid	RNN SARIMA +	RNN SARIMA *	RNN SARIMA resid
Period one	MAPE	1.6142	2.2543	0.0014	1.8575	1.7326	1.2947	1.4112	1.4068	1.2156
	sMAPE	0.3987	0.5733	0.0004	0.4580	0.4279	0.3249	0.3524	0.3517	0.3073
	RMSE	2429.4000	3367.9000	1.9449	2727.0000	2528.3000	1728.4000	2102.4000	2074.8000	2149.0000
	MBRAE	0.2662	0.3370	0.0005	0.2752	0.2532	0.2520	0.2734	0.2745	0.1902
	UMBRAE	0.3628	0.5082	0.0005	0.3797	0.3391	0.3370	0.3762	0.3784	0.2349
	R2	0.9809	0.9761	1.0000	0.9857	0.9853	0.9849	0.9909	0.9900	0.9821
Period two	MAPE	1.4198	3.8442	0.0007	1.4815	1.5413	1.5660	1.2168	1.1824	1.8527
	sMAPE	0.3524	0.9857	0.0002	0.3694	0.3824	0.3871	0.3028	0.2944	0.4603
	RMSE	3165.5000	8983.2000	1.5323	3327.5000	3577.2000	3547.4000	2958.4000	2839.1000	4713.1000
	MBRAE	0.2340	0.4495	0.0002	0.2613	0.2653	0.2678	0.2040	0.2252	0.2543
	UMBRAE	0.3055	0.8165	0.0002	0.3538	0.3611	0.3658	0.2563	0.2906	0.3410
	R2	0.9949	0.9727	1.0000	0.9794	0.9791	0.9987	0.9825	0.9810	0.9990

7.4 Code

We will now explain the use of each file uploaded in the zip file, mentioning its inputs and outputs. Additionally, the file data.csv contains the data used for this paper.

analysis_normality: Input an array, return skewness, kurtosis and Jarque-Bera test.

dw_test_simple: Input the true value and both predictions, returns the Diebold-Mariano test.

forecast_accuracy: Input the true value, y estimates and forecasts, returns the MAPE, sMAPE, RMSE, MRAE, MBRAE, UMBRAE and R^2.

sr_test: Input the true value and both predictions, returns Wilcoxon signed-rank test.

Note there are some files for period 1 and 2 however they follow the same naming structure, hence, in the list below we will just use "i".

simple_linear_i: LM model.

arima_i: ARIMA model.

sarima_i: SARIMA model.

ann_i: ANN model.

rnn_i: RNN model.

ann_linear_additive_1: ANN LM + hybrid model.

ann_linear_1_multiplicative: ANN LM * hybrid model.

ann_linear_i_including_resid_in_X: ANN LM resid hybrid model.

ann_hybrid_additive_1: ANN ARIMA + or ANN SARIMA + hybrid model.

ann_hybrid_1_multiplicative: ANN ARIMA * or ANN SARIMA * hybrid model.

ann_hybrid_i_including_resid_in_X: ANN ARIMA resid or ANN SARIMA resid hybrid model.

rnn_linear_additive_1: RNN LM + hybrid model.

rnn_linear_1_multiplicative: RNN LM * hybrid model.

rnn_linear_i_including_resid_in_X: RNN LM resid hybrid model.

rnn_hybrid_additive_1: RNN ARIMA + or RNN SARIMA + hybrid model.

rnn_hybrid_1_multiplicative: RNN ARIMA * or RNN SARIMA * hybrid model.

rnn_hybrid_i_including_resid_in_X: RNN ARIMA resid or RNN SARIMA resid hybrid model.

rolling_period_i: Can be used as a rolling window method when changing the method.

arima_optimization_i: Optimization of ARIMA model.

ann_optimization_i: Optimization of ANN model.

rnn_optimization_i: Optimization of RNN model.