Neural Network-Enhanced Decision Support: Investigating Prediction Intervals for Real-Time Digital Marketing Return on Investment Data

Lucas Rabelo de Araujo Morais[1], **Gecynalda Soares da Silva Gomes**[1]

[1]Federal University of Bahia – Institute of Mathematics and Statistics
Salvador – Bahia – Brazil

lucasrabeloufba@gmail.com, gecynaldassg@ufba.br

Abstract. *This work delves into the application of artificial neural network (ANN) models and recurrent neural networks (RNN), for time-series forecasting in the dynamic realm of digital marketing. Focused on a travel company's real-time updated Return on Investment (ROI) data from Google Ads campaigns, the research evaluates the efficacy of prediction intervals (PIs) in capturing forecast uncertainties. The study's contribution lies in the exploration of PIs in ANN models for digital marketing ROI data, providing valuable insights for decision-makers navigating rapidly changing scenarios. The work emphasizes the significance of incorporating intervals in ANN models for robust decision-making in business and digital marketing applications.*

1. Introduction

As noted by [Järvinen and Karjaluoto 2015], consumers now interact with companies primarily through digital channels, such as e-commerce websites and social media. Consequently, it is recommended that organizations utilize web analytics tools to measure the return of their digital marketing campaigns. Some popular digital marketing tools are Google Ads and Meta Ads, which allows companies to create targeted marketing campaigns. The measurement of the return on digital marketing can take various forms, both financial and non-financial (e.g. increased website traffic). Statistical metrics, commonly referred to as Key Performance Indicators (KPIs), are often used for this purpose, with the Return on Investment (ROI) being a widely recognized and utilized metric to measure financial return.

However, predicting these metrics can be challenging, particularly when dealing with non-linear data. This work examines the application of neural network (ANN) and recurrent neural network (RNN) models for time-series forecasting on frequently updated digital marketing ROI data of a travel company. The focus is on exploring parametrical methods for building prediction intervals (PIs), aiming to answer: "How does the use of PIs help us to deal with the uncertainty that is present in the dynamic nature of real-time forecasts of digital marketing performance metrics?", as PIs provide a more realistic representation of uncertainty in the forecasts compared to traditional confidence intervals, as noted by [Carney et al. 1999]. To demonstrate the efficacy of these models, a RNN model is fitted to a complex, frequently updated ROI time-series and two different types of prediction intervals are employed.

The rest of the paper is structured as follows: in Section 2, the paper provides a comprehensive literature review. Section 3 details the methodology employed in this study, including data collection procedures, pre-processing steps, and the implementation of models for time-series forecasting. Section 4 presents the results of the experiments. In Section 5, the findings are discussed. Finally, Section 6 concludes the paper by summarizing key insights and highlighting limitations.

2. Literature Review

[Wong et al. 1997] conducted a review on the applications of neural networks in various business fields, including finance, where a large number of studies have been published on the use of neural networks for stock market forecasting. In the marketing field, [Wong et al. 1997] found 10 papers between 1995 and 1998, with two of them addressing the use of neural networks for predicting airline passenger numbers. A review by [Abiodun et al. 2018] showed that out of the 12 papers focused in the applications of ANNs in marketing, published from 2009 to 2018, 5 focused on the use of neural networks for prediction tasks.

Neural networks have also been utilized in predicting real-time data. [Mei et al. 2019] used Long Short Term Memory (LSTM) Recurrent Neural Networks to predict mobile bandwidth data in real-time, by training LSTM models on temporal patterns from data in different mobile networking scenarios. [Guo et al. 2016] also tackled real-time learning and prediction, proposing an adaptive gradient learning method for RNNs that aimed to make time-series forecasting robust to outliers and change points.

[Armstrong 2001] reported in the literature that neural networks are better than traditional forecasting models for long-term forecasting horizons, but this is not the case for shorter horizons. While neural networks (NNs) provide point-estimate forecasts, they do not offer prediction intervals. According to [Zhu and Laptev 2017], uncertainty estimation has been widely studied for classical forecasting models, but not for ANN models. However, with the advancements in computational power and the growing popularity of neural networks, [Kasiviswanathan and Sudheer 2016] emphasized that the uncertainty in ANN models is a significant issue that cannot be ignored.

[Quan et al. 2014] explored the challenge of dealing with uncertainty in electrical load forecasting using ANN models, acknowledging that ANN-based prediction intervals have some issues such as implementation difficulties, assumptions about the data distribution, and high computational demands. [Nourani et al. 2021] noted that the use of prediction intervals for quantifying uncertainty has grown in recent years for purposes of planning and risk management. However, without understanding the sources of uncertainty, the assessment of ANN modeling quality is impossible, which negatively impacts decision making. This work contributes to literature by investigating the use of PIs associated with ANN modeling in the field of digital marketing, which enhances the model's quality and also proposes a web app to evaluate the model's forecasts, advocating for the usage of PIs as best-case and worst-case scenarios in order to tackle the issue of supporting real-time decision making for marketing campaigns of a travelling agency.

3. Methodology

3.1. Data Analysis

In this work, real-time updated financial data regarding Google Ads marketing campaigns for a Brazilian travel agency was made available by a Brazilian digital marketing company. To preserve their identity, fictional names,"VoaL" and "Otis", were used for the travel agency and the digital marketing company, respectively. The data analysis was performed using the statistical software, `R`.

The digital marketing company, Otis, provided two data sources with information about the travel agency. The first data source was a sheet with daily investment data on Google Ads advertisement campaigns, while the second was a daily updated dataset with revenue data from all marketing channels, including Google Ads, which was used to calculate the daily ROI time series from Google Ads campaigns. The information available in the data and the chosen model's predictions were made available for end users through an web application built with R-shiny, a framework that builds web apps with `R`.

To assess the prediction model, [Hyndman and Athanasopoulos 2021] suggested dividing the dataset into training and validation data, with the test data usually corresponding to 20% of the total and being at least as large as the desired forecasting horizon. In this work, the division between training and validation data followed the approach suggested by [Siami-Namini et al. 2018], with 70% of the data being used for training and fitting the models and 30% being used for validation. ARIMA, LSTM, and RNN models were fitted and compared in different time windows of 3 days (short horizon), 7 days (medium horizon) and 14 days (long horizon) to verify the capacity of generalization of the models when dealing with unseen data.

3.2. Return On Investment - ROI

The metric Return On Investment (ROI) is widely used to evaluate the efficiency of investments or to make comparisons between different investments, in digital marketing, the ROI is used to verify if the marketing channel (or campaign) is returning value to the company. It can be considered a KPI as it is crucial to determine if an investment is yielding a financial return. Although there are various methods to compute ROI, this work was inspired by the approach proposed by [Botchkarev and Andru 2011]. In this formulation, the ROI at each day t, ROI_t is equivalent to equation 1.

$$ROI_t = \frac{\text{Revenue}_t - \text{Investment}_t}{\text{Investment}_t} \tag{1}$$

A ROI of 100% suggests that the revenue generated from the investment was equal to its cost, with no additional return. However, in the proposed formulation, a ROI of 100% implies that the invested money was not only returned, but the profit was equal to the investment. The ROI was calculated by using daily investment data from Google Ads campaigns (e.g. the daily cost of all running marketing campaigns), and daily revenue data from all transactions that were mainly impacted by the Google Ads channel.

3.3. Neural Network Models

Neural Network models are inspired by the architecture of the brain and can be utilized to model complex non-linear relationships between the response and predictor variables. In the context of time series forecasting, these models consist of three layers: an input layer, a hidden layer, and an output layer. Each layer, except for the output layer, can contain multiple neurons. It is also possible to incorporate multiple hidden layers in a Neural Network model.

For an ANN (Artificial Neural Network) with three layers, [Brezak et al. 2012] defines the input of the jth neuron of the hidden layer of the nth learning sample as $net_j(n) = \sum_{i=1}^{I} v_{j,i}x_i(n)$, $j = 1, ..., J-1$, (the bias is always one) where I is the quantity of neurons in the input layer, v are the weights (selected by learning algorithm that minimizes the cost function) and x are the network inputs. In the hidden layer the input is then modified to generate an output following a non-linear activation function, which can be a bipolar sigmoid, given by,

$$y_j(n) = \frac{2}{1 + e^{-c_j net_j(n)}} - 1, j = 1, ..., J-1 \tag{2}$$

c_j is a gain parameter which generally is $0,1$ used to make the neural network robust against outliers, bias' output value is always $y_j = 1$.

The last mth output value of the ANN is obtained by summing all the values given by the neurons in the hidden layer (including bias) and the weights w which are generated in the output, the sum is then given by the following equation:

$$O_m(n) = \sum_{j=1}^{J} y_j(n)w_{mj}, m = 1, ..., M \tag{3}$$

The parameters v, w and c are estimated during the learning process.

3.4. LSTM and Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are an extension of ANNs, the main characteristic of an RNN is the dynamic neuron model with a recurrent structure, this type of neural network was developed because in general ANNs have less memory and cannot deal with temporal dependencies, fundamentally the RNN considers past information to generate an output. [Zeroual et al. 2019] summarizes the architecture of recurrent neural networks, where the neuron A of the neural network receives the input entry x_t and outputs a hidden state information h_t:

$$h_t = g(W_{x_t} + Uh_{t-1}) \tag{4}$$

W is a weight matrix, h_{t-1} the hidden state of the previous time step and U is a matrix known as transition matrix, which is similar to a markov chain and g is a sigmoid function.

However, problems of vanishing gradient that limit the efficiency of simple recurrent neural networks, led to the development of Long Short-Term Memory (LSTM) recurrent neural networks. According to [Zeroual et al. 2019], they are more efficient in handling temporal dependencies. LSTM neural networks can also be implemented in `R` [Wanjohi 2018], these NNs consists of memory blocks called cells connected by layers. Information in the cells is contained in the cell state and hidden states, regulated through mechanisms known as gates, using sigmoid and hyperbolic tangent activation functions. The three main gates controlling the flow of information are called the forget gate, input gate, and output gate.

The gates are formed by sigmoid functions, where I_t is the input gate, F_t is the forget gate, and O_t is the output gate. $\check{C}_t$ and C_t represent the intermediate cell state and the cell state (the next memory input), respectively. Additionally, each sigmoid layer outputs a number between 0 and 1, representing the amount of data to be passed to each cell. 0 means that the information should not be passed, and 1 means it should be fully passed on.

The equations for each gate can be defined as follows:

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i)$$
$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f)$$
$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o)$$

W refers to the weight parameters, and b is the bias parameter.

The forget gate (F_t) determines the type of information that will be deleted from the cell state. The input gate (I_t) chooses which new data should be kept in the cell, where a sigmoid layer called the "input gate layer" first selects the values to be modified. Later, the hyperbolic tangent layer creates a vector of candidate values that can be added to the cell state. The output gate defines what should come out of each cell. The output value is based on the cell state along with the filtered and added data.

The intermediate cell state $\check{C}_t$, the cell state C_t, and the hidden state H_t are defined as:

$$\check{C}_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c)$$
$$C_t = F_t \times C_{t-1} \times \check{C}_t$$
$$H_t = O_t \times \tanh(C_t)$$

3.5. ARIMA

The ARIMA model (Autoregressive Integrated Moving Average model), used in this work, is one of the most widely used methods for time series forecasting. The `auto.arima` function in the statistical software `R` already includes a stationarity test for estimating the differencing parameter and can be directly used for this purpose, [Morettin and Toloi 2004] and [Hyndman and Athanasopoulos 2021] employ two different approaches to explain the model, one being more theoretical and the other computational.

The ARIMA model is a combination of three models: an autoregressive $AR(p)$ where p corresponds to the order of the autoregressive part (the autoregression term indicates that there is a regression of a variable on itself), an integrated $I(d)$ of order d, where d is the degree of differencing. Differencing is used to transform a non-stationary series into a stationary one and can help stabilize the mean of a time series by eliminating or reducing trend and seasonality. The moving average model corresponds to $MA(q)$, where q is the order of the moving average part. The equation of the full model is illustrated below:

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{5}$$

3.6. Error Metrics

The error measures used in the work are the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE), and the Symmetric Mean Absolute Percentage Error (sMAPE). For model selection, those with the lowest RMSE (**Equation** (6)) and MAE (**Equation** (7)) are chosen. Minimizing MAE leads to median predictions, while minimizing RMSE leads to mean predictions. The MAE measure is easier to interpret; however, RMSE is more commonly used.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{n}}, \quad i = 1, ..., N \tag{6}$$

$$MAE = \frac{\sum_{i=1}^{N}|x_i - \hat{x}_i|}{n}, \quad i = 1, ..., N \tag{7}$$

sMAPE (**Equation** (8)) is an alternative measure to the Mean Absolute Percentage Error (MAPE):

$$sMAPE = \sum_{t=1}^{N} 200 \frac{|x_i - \hat{x}_i| / (x_t - \hat{x}_t)}{n}, \quad t = 1, ..., N \tag{8}$$

The sMAPE has some considerations; if x_t is close to zero, $\hat{x}_t$ is likely to be close to zero as well, making the calculation unstable. Another point is that the sMAPE value can be negative, hence not being an absolute percentage measure. Due to these complications, this measure was evaluated together with RMSE and MAE.

3.7. Time Series Non-Parametrical Testing

In time series forecasting, many statistical models assume stationarity. Therefore, it is necessary to check if the series has well-defined trend and seasonality components, otherwise statistical models such as ARIMA won't have a good performance and it's necessary to rely in models that are capable to deal with non linearity in the data. For assessing the trend component, the Mann-Kendall test can be used. The null hypothesis of this test assumes the absence of trend in the series. The test statistic can be defined as the sum of the ranks of the differences between sequential values x_i and x_j with $i < j$. According to [Burn and Elnur 2002], the statistic is defined as $S = \sum_{i=1}^{n-1} \sum_{j=i-1}^{n} sgn(x_j - x_i)$, where $sgn(\theta) \begin{cases} +1 & ,\theta > 0 \\ 0 & ,\theta = 0. \\ -1 & ,\theta < 0 \end{cases}$

The Kwiatkowski-Phillips-Schmidt-Shin test (KPSS) is a non-parametric test with the null hypothesis of stationarity. According to [Nielsen 2005], the test statistic can be calculated assuming no trend in the form of $Y_t = \xi_t + e_t$, where e_t is stationary and $\xi_t = \xi_{t-1} + v_t$ is a random walk with $v_t \sim IID(0, \sigma_v^2)$. If the variance is 0, then $\forall t$ $\xi_t = \xi_0$, and Y_t is stationary. A simple regression $Y_t = \hat{\mu} + \hat{e}_t$ is used to estimate the stochastic component, with the test statistic in **Equation** (9):

$$KPSS = \frac{1}{T^2} \frac{\sum_{t=1}^{T} S_t^2}{\hat{\sigma}_\infty^2} \tag{9}$$

where $S_t^2 = \sum_{s=1}^{t} \hat{e}_s$ is a partial sum of the residuals, and $\hat{\sigma}_\infty^2$ is an estimator of the variance of $\hat{e}_s$.

To verify the seasonal component, the Kruskall-Wallis test can be conducted. As defined in [Morettin and Toloi 2004], each column of a table can be considered a sample from the population, consisting of k samples of size n_j, where Y_{ij} is the i-th observation of sample j, i.e., $j = 1, 2, ..., k$ and $i = 1, 2, ..., n_j$ with $N = \sum_{j=1}^{k} n_j$. Observations Y_{ij} should be replaced by their ranks R_{ij}, obtained by ordering all N observations, with $R_{.j} = \sum_{i=1}^{n_j} R_{ij}$. The test statistic can be calculated as:

$$T_1 = \frac{12}{N(N+1)} \sum_{j=1}^{k} \frac{R_{.j}^2}{n_j} - 3(N+1) \tag{10}$$

The null hypothesis of no seasonality is rejected if $T_1 \geq T_{1c}$ (where T_{1c} is the critical value of the statistic T_1) based on a significance level α.

In addition to analyzing the series behavior, since the goal is to make predictions of future observations, the Shannon spectral entropy can be calculated, which is a measure of how "easy" the series is to predict. According to [Hyndman and Athanasopoulos 2021], a series with high seasonality and trend (well-behaved) will have entropy close to 0, while a series with a lot of noise, hence difficult to predict, will have entropy close to 1. According to [Goerg 2013], Shannon's spectral entropy is defined as in **Equation** (11), and the density is normalized so that $\int_{-\pi}^{\pi} f_x(\lambda)\, d\lambda = 1$. The entropy was calculated using the `feasts` package in `R`.

$$H(f_x) = -\int_{-\pi}^{\pi} f_x(\lambda) \log f_x(\lambda)\, d\lambda \tag{11}$$

3.8. Prediction Intervals

Prediction intervals are a very useful way to represent the uncertainty present in forecasts. A 95% prediction interval contains the values that should include the actual future value with a probability of 95%. According to [Hyndman and Athanasopoulos 2021], considering the assumption that the distribution of future observations follows a normal distribution, a 95% prediction interval for the forecast h can be described as in **Equation** (12), where c depends on the coverage probability of the interval, being a $\alpha/2$ quantile of the normal distribution.

$$[\hat{y}T + h|T - c\hat{\sigma}h, \hat{y}T + h|T + c\hat{\sigma}h] \tag{12}$$

The standard deviation of the distribution of forecasts can be estimated as in **Equation** (13), where K is the number of estimated parameters in the forecasting method, and M represents the number of missing observations in the residuals $y_t - \hat{y}t = et$.

$$\hat{\sigma} = \sqrt{\frac{1}{T-K-M}\sum^{T} t = 1e^2 t} \tag{13}$$

[Hyndman and Athanasopoulos 2021] also propose benchmark methods for calculating $\hat{\sigma}h$, of which two were selected for this work. The naïve benchmark method where $\hat{\sigma}h = \hat{\sigma}\sqrt{h}$ and the mean method with $\hat{\sigma}_h = \hat{\sigma}\sqrt{1+1/T}$. This is done for comparison, respectively, between an interval where the amplitude of the upper and lower limits increases over the forecast horizon (with naïve being the simplest of those presented by the authors) and another that keeps the amplitude constant on the average of the series.

4. Results

The RNN model was optimized to be included in the web application built with r-shiny, since data was frequently updated it was necessary to input new training data in the model, while maintaining the aspect ratio of 70% for training and 30% for validation, where the RNN took 3.59 seconds to process the training function compared to 64.61 seconds for the LSTM model and 0.31 seconds for the ARIMA model, since the RNN and the LSTM model had the lowest error measures in the validation and test sets, the most efficient model was chosen to be the RNN (**Table 1**), where the MAE means that on average, its predictions are off by approximately 1.03 units from the actual ROI values, The SMAPE of 0.43 suggests that, on average, its predictions have an average error of approximately 43% and the RMSE suggests that on average the predictions in validation set are off by approximately 1.31 units from the actual values.

The LSTM neural network had the best error metrics with lower MAE (on average the model's predictions are off by approximately 0.83 units) and RMSE (predictions off by 1.05 units) than the two compared models (**Table 1**) and a SMAPE of 38%. The ARIMA model had the worst error metrics, with a SMAPE of 57%, which is 14% higher than the recurrent neural network model and 19% higher than the LSTM.

Table 1. Error Measures in Validation Set.

Models	MAE	SMAPE	RMSE
ARIMA	1,59	0,57	1,82
RNN	1,03	0,43	1,31
LSTM	0,83	0,38	1,05

In addition to the test set, the predictions for different time periods in **Table 2** show that the model with the best forecasts for a long time period (14 days) was the recurrent neural network, with an average percentage error of 27% (SMAPE) and the predictions are deviating by 0.68 units from the real ROI values considering MAE and 0.84 units considering RMSE. Furthermore, for all time periods, in the RNN, both RMSE and MAE were below 1 for the model, which is a good indicator. If predictions were off by 1 on average, it would mean that the model's forecasts would significantly deviate from the real ROI values in most cases; however, the ARIMA model showed slightly better metrics for the 7-day period. The performance of the LSTM model was only slightly better than the RNN model for the 7-day period in terms of MAE and RMSE metrics.

Table 2. Error Measures in Test Sets.

Forecasting Window	3 Days			7 Days			14 Days		
Models	MAE	SMAPE	RMSE	MAE	SMAPE	RMSE	MAE	SMAPE	RMSE
ARIMA	1.11	0.43	1.14	0.54	0.21	0.68	0.72	0.28	0.85
RNN	0.82	0.32	0.93	0.58	0.23	0.71	0.68	0.27	0.84
LSTM	1.04	0.40	1.08	0.57	0.21	0.84	0.79	0.32	1.11

Prediction intervals were also calculated for the RNN model to quantify the uncertainty of the predictions. With a coverage probability of 95%, the benchmark methods used were the Naïve and Mean methods, and the number of parameters used to estimate the standard deviation of the predictions were 3, which was equal to the number of parameters estimated by the RNN model. The main difference between the Naïve prediction interval (**Figure 1**) and the Mean interval (**Figure 2**) is that the former increases with the forecast horizon while the latter remains constant with the mean.

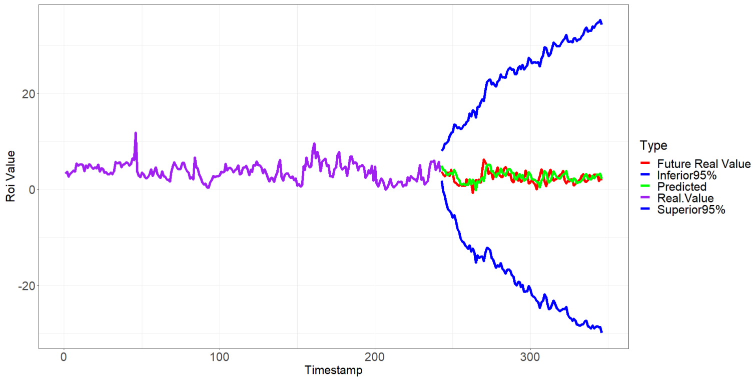

Figure 1. RNN 95% Prediction Interval with Naïve Benchmark Method

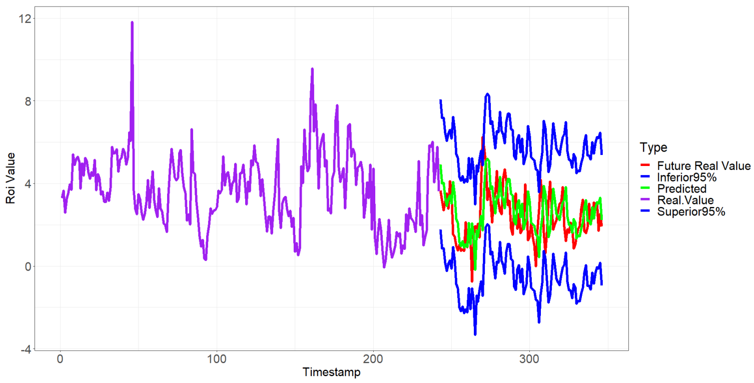

Figure 2. RNN 95% Prediction Interval with Mean Benchmark Method

The mean benchmark method was selected to estimate $\hat{\sigma}_h$ and calculate the prediction intervals for the real-time forecasts in the web application. The mean benchmark was preferred over the naïve method due to its advantages in visualization and well-defined upper and lower bounds. Additionally, the mean benchmark provided more credible intervals for the 14-day prediction window, as compared to the wider prediction intervals produced by the naïve method.

The results shown in this section were obtained in a specific time frame, however, given the real time nature of the data it's important to periodically check if the model is still making reliable forecasts, so since the model receives new data as training input, a section called Error Metrics was developed in the web app (**Figure 3**), where basically the error metrics of the last 14 days (since that was the chosen forecast horizon) are used to continuously evaluate the model's forecasting quality, when one of the error metrics starts to grow continuously the model needs to be reviewed. Considering the SMAPE for instance, in (**Figure 3**), the model is predicting the next two weeks with an average error of 21%, if this error grows, the model's architecture should be reviewed.

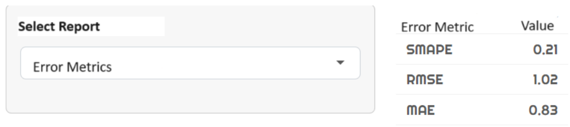

Figure 3. Last 14 Days Error Metric Report in Web App.

5. Discussion

Since the ROI time series has a daily frequency, the decomposition of its components followed the recommendation of [Hyndman and Athanasopoulos 2021]. The work states that if the data is observed more than once a week, there may be more than one seasonal pattern, and daily data usually exhibits weekly seasonality. Therefore, for additive decomposition, an initial frequency of 7 was used. The additive decomposition of the series in **Figure 4** shows that, despite the absence of a trend, the seasonal component was present in the time series.

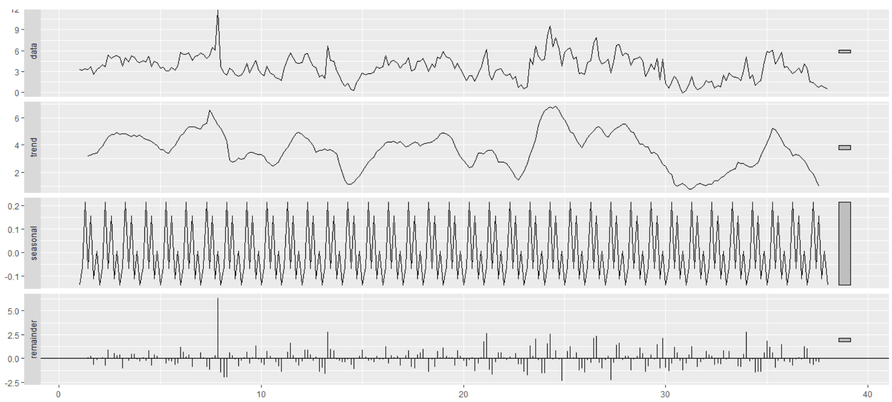

Figure 4. Additive Decomposition of Time Series

In addition to graphical evidence of seasonality, the Kruskall-Wallis test provided evidence to reject the null hypothesis of non-seasonality at a 5% significance level (p-value < 0.0001). However, the Mann-Kendall test provided evidence to reject the null hypothesis of a lack of trend in the series (p-value < 0.0001), contradicting the additive decomposition analysis, which means that even though there is graphical evidence to state that there is no trend in the data, there is no statistical evidence to back this statement.

Furthermore, the Kwiatkowski-Phillips-Schmidt-Shin test (or KPSS test) revealed that at a significance level of 5%, the null hypothesis of stationarity could be rejected (p-value = 0.0239). The Shannon spectral entropy was 0.85, i.e., very close to 1, considered a high value, highlighting the difficulty of forecasting the time series. Additionally, the Keenan non-linearity test revealed that the series exhibits non-linear behavior (p-value $<$ 0.0001). Therefore, a model from the ARIMA class would not be sufficient. Since the series is challenging to predict, non-stationary, and demonstrates a non-linear relationship, taking neural network models in consideration for ROI forecasting is a must, these findings were crucial in considering the two neural network models, both of which resulted in accurate ROI predictions.

The selection of the simple RNN model for the real-time forecasts in the web application (**Figure 5**) was based on its good performance compared to the ARIMA and its lower computational intensity compared to the LSTM. The RNN model showed a 94% decrease in the processing time of the training function, and its prediction intervals were updated with new data. The approach of presenting results through an web application and adjusted models used in this work was similar to that of [Ristow et al. 2021], but the authors used an ARIMA model and did not specify if their predictions were updated.

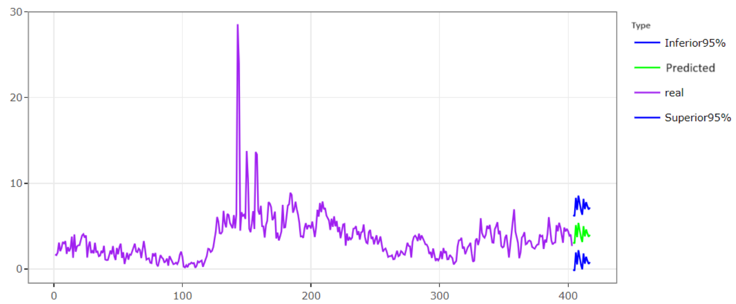

Figure 5. Real Time Updated 14 days prediction for ROI In Web App.

This study adopted a parametric method to estimate the standard deviation and calculate prediction intervals, which required the estimation of the parameters of a probability distribution. However, even though assuming the normality of the probability distribution for future predictions may help to capture uncertainty in data, it can be a strong assumption, as noted in [Quan et al. 2014]. They presented a nonparametric method, the LUBE (Lower Upper Bound Estimation) method, which utilized an ANN model with two outputs for the upper and lower bounds. While this method has been shown to be more robust compared to traditional methods, it requires training two models, one for point forecasts and another for PIs, which can be computationally intensive.

[de Sousa et al. 2020] trained a model to present point forecasts and used the LUBE method to construct PIs, but this approach may not be feasible in this study due to the computational demand. To reduce the computational demand and stay within RAM usage in the cloud server limits, this study chose to work with a parametric method and train only the point forecast model, which reduces training time, since the model needs to train every time the web app is opened. As for the assumptions of a probability distribution, [Hyndman and Athanasopoulos 2021] do not specify the use of a normal distribution, and statistical tests or empirical knowledge can be used to estimate prediction intervals and select a distribution.

The importance of prediction intervals in forecasting for enhancing decision making is also a topic in [Goodwin et al. 2010], however the authors conducted the study in a laboratory, rather than a commercial environment, since fast decisions are usually common in business, bringing forecasts with a dynamic nature, whilst dynamically training the model and making prediction intervals available for the end user was achieved in this work by proposing the use of a web application (a friendly User Interface). A reduced version (hiding sensitive information of the companies) of the web app in a frozen timeframe is also available and can be checked in [de Araujo Morais 2024].

Among the forecasting models for ROI, the LSTM model yielded the best results in terms of error metrics compared to the test set, outperforming the ARIMA model. This outcome aligns with the findings of studies where neural networks are also superior than ARIMA [Siami-Namini et al. 2018] [Lou et al. 2022]. The importance of deep learning methods for complex time series forecasting was also verified in this study in predicting ROI, as both the simple recurrent neural network and the LSTM had smaller errors than the ARIMA model for the test set. Additionally, the ARIMA model extrapolated predictions to the mean in the test set.

6. Conclusion

In the field of business and marketing, the application and discussion of prediction intervals in artificial neural network (ANN) models is limited. Previous studies in this area, such as [Abiodun et al. 2018] and [Wong et al. 1997], have not specifically addressed the use of prediction intervals. However [Nourani et al. 2021] explored this subject. This paper makes a unique contribution by providing a comprehensive examination of the most efficient method for measuring uncertainty in frequently updated digital marketing ROI data, which enables decision makers to rapid plan their actions in crisis like cenarios, optimizing social media campaigns or in setting more realistic goals to be achieved.

After a careful evaluation of available methods, the mean benchmark prediction interval was selected as the most appropriate for the current study. This decision was based on the computational efficiency of the method and the better defined prediction intervals it provided for the 14-day forecast horizon. The inclusion of prediction intervals in the analysis of ROI data is critical for informed decision-making, as the lower and upper bounds of the interval can provide a worst-case and best-case scenario, respectively.

One of the limitations of this study is the lack of computational power available in the cloud service hosting the web application, even though the study tackles the problem of dealing with the dynamic nature of the real-time data, since the structure of the time series may change over time, it can be necessary to review the architecture of the RNN model in case the error metrics get worse and the model stops reflecting reality. Furthermore, computational power was also a constraint, as the lack of computational power prevented the implementation of the LUBE method for forecasting the ROI and fitting a model for point predictions.

For future research in the business field utilizing web applications with more resources available, it is recommended to employ the LUBE method to estimate prediction intervals, if not possible the parametric prediction intervals are less computationally intensive, easier to implement and can capture some uncertainty in the predictions. In conclusion, this study highlights the importance of incorporating prediction intervals in ANN models for business and digital marketing applications on data collected through social media and websites, providing valuable insights for future research in this area.

References

Abiodun, O., Jantan, A., Omolara, A., Dada, K., Mohamed, N., and Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11).

Armstrong, J. (2001). *Principles of Forecasting: A Handbook for Researchers and Practitioners*.

Botchkarev, A. and Andru, P. (2011). A return on investment as a metric for evaluating information systems: Taxonomy and application. *Interdisciplinary Journal of Information, Knowledge, and Management*, 6:245–269.

Brezak, D., Bacek, T., Majetic, D., Kasac, J., and Novakovic, B. (2012). A comparison of feed-forward and recurrent neural networks in time series forecasting. In *IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr)*.

Burn, D. and Elnur, M. (2002). Detection of hydrologic trends and variability. *Journal Of Hydrology*, 255:107–122.

Carney, J., Cunningham, P., and Bhagwan, U. (1999). Confidence and prediction intervals for neural network ensembles. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339)*, volume 2, page 1215–1218 vol.2.

de Araujo Morais, L. R. (Last accessed on January 13, 2024). Relatório dinâmico - voal - otis / dynamic report - voal - otis. Author's personal website. `https://marreapato.shinyapps.io/otis_voal/` (In Portuguese).

de Sousa, A., Furtado, H., Macêdo, W., and Meneses, A. (2020). Analysis of artificial neural network point forecasting models and prediction intervals for solar irradiance estimation. *American Journal of Engineering and Applied Sciences*, 13(3):347–357. doi: 10.3844/ajeassp.2020.347.357.

Goerg, G. (2013). Forecastable component analysis (foreca). Available from: arXiv:1205.4591 [stat.ME].

Goodwin, P., Önkal, D., and Thomson, M. (2010). Do forecasts expressed as prediction intervals improve production planning decisions? *European Journal of Operational Research*, 205(1):195–201. doi: 10.1016/j.ejor.2009.12.020.

Guo, T., Xu, Z., Yao, X., Chen, H., Aberer, K., and Funaya, K. (2016). Robust online time series prediction with recurrent neural networks. page 816–825.

Hyndman, R. and Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 3rd edition.

Järvinen, J. and Karjaluoto, H. (2015). The use of web analytics for digital marketing performance measurement. *Industrial Marketing Management*, 50:117–127.

Kasiviswanathan, K. and Sudheer, K. (2016). Comparison of methods used for quantifying prediction interval in artificial neural network hydrologic models. *Model. Earth Syst. Environ.*, 22(2).

Lou, H., Wang, X., Gao, Y., and Zeng, Q. (2022). Comparison of arima model, dnn model and lstm model in predicting disease burden of occupational pneumoconiosis in tianjin, china. *BMC Public Health*, 22:2167. `https://doi.org/10.1186/s12889-022-14642-3`.

Mei, L., Hu, R., Cao, H., Liu, Y., Han, Z., and Li, F. e. a. (2019). Realtime mobile bandwidth prediction using lstm neural network. page 34–47.

Morettin, P. and Toloi, C. (2004). *Análise de Séries Temporais [Time Series Analysis]*. Edgard Blucher.

Nielsen, H. (2005). Non-stationary time series and unit root tests. Accessed January 11, 2024.

Nourani, V., Paknezhad, N., and Tanaka, H. (2021). Prediction interval estimation methods for artificial neural network (ann)-based modeling of the hydro-climatic processes, a review. *Sustainability*, 13(4).

Quan, H., Srinivasan, D., and Khosravi, A. (2014). Uncertainty handling using neural network-based prediction intervals for electrical load forecasting. *Energy*, 73:916–925.

Ristow, D., Henning, E., Kalbusch, A., and Petersen, C. (2021). Models for forecasting water demand using time series analysis: a case study in southern brazil. *Journal of Water, Sanitation and Hygiene for Development*, 11(2):231–240.

Siami-Namini, S., Tavakoli, N., and Siami Namin, A. (2018). A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, page 1394–1401.

Wanjohi, R. (2018). Time series forecasting using lstm in r. Last Access: 11/01/2024. http://rwanjohi.rbind.io/2018/04/05/time-series-forecasting-using-lstm-in-r/.

Wong, B., Bodnovich, T., and Selvi, Y. (1997). Neural network applications in business: A review and analysis of the literature (1988–1995). *Decision Support Systems*, 19(4):301–320.

Zeroual, A., Harrou, F., Dairi, A., and Sun, Y. (2019). Deep learning methods for forecasting covid-19 time-series data: A comparative study. *Chaos, Solitons Fractals*, 6(47):110121.

Zhu, L. and Laptev, N. (2017). Deep and confident prediction for time series at uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, page 103–110.