ECONOMIC IMPLICATIONS OF DEEP MACHINE LEARNING FOR TOURISM TIME SERIES FORECAST

Ivanka Petrova Vasenska[1,a,*]

[1]Tourism Department, Faculty of Economics, SWU "Neofit Rilski", Blagoevfrad, 66 Ivan Michaiylov, Str., Bulgaria
[a]ivankav@swu.bg, ivankav@abv.bg
[*]Corresponding author

Cite as: *Vasenska, I. P. (2024). Economic implications of deep machine learning for tourism time series forecasting, Ekonomicko-manazerske spektrum, 18(1), 90-101.*

Available at: *dx.doi.org/10.26552/ems.2024.1.90-101*

Received: 10 March 2024; Received in revised form: 19 May 2024; Accepted: 6 June 2024; Available online: 30 June 2024

Abstract:
Research background: Predicting the flow of inbound tourism accurately has always posed a significant challenge for all parties involved in the industry. The complex nature of the tourism product, which is directly and indirectly affected by various risks, disasters, and crises, further highlights its susceptibility to disruptions and fluctuations. As a result, there has been a growing interest in forecasting inbound tourism flows using contemporary data science methods and artificial neural network (ANN) methods. This paper, therefore, seeks to explore the AI forecasting techniques by employing a deep machine learning (DML) approach and comparing various Python libraries for time series forecasting within a Jupyter Notebook computing environment. The data from domestic and international tourism stays registered in Bulgaria for the period 2002 to 2023 has been utilized to construct an advanced deep neural network using multiple Python libraries.
Purpose of the article: The purpose of the current paper is to establish which time series forecasting model - Exponential Smoothing, TBATS, Auto ARIMA, Theta or LSTM has better accuracy estimation and may be applied for similar tasks in the future by for research and practical economic purposes.
Methods: The applied methodology was based on the classical scientific method. As to the main findings they could help in daily operation planning, managing and relocating resources in tourism on micro and macro level as well as with insights on the drawbacks and limitations on this research area related to analysis and novelty implications.
Findings & Value added: What is more, the results obtained reaffirmed that ANN can be applied for an accurate forecasting, especially in the case of Bulgaria, where such models have not been applied yet neither by the tourism related academics nor by the business and the policymakers.

Keywords: Bulgaria tourism flows; timeseries forecast; artificial neural network; method accuracy

JEL Classification: C22; C45; Y1; Z32

1. Introduction

Deep machine learning (DML), a subset of artificial intelligence and a specialized field within machine learning (ML), utilizes highly sophisticated artificial neural network (ANN) structures. These structures find applications in various domains such as computer vision, speech recognition, and natural language processing domain (Goodfellow et al., 2016; Geron, 2017). ANNs are complex models that employ network architectures akin to the biological neural networks (NN) used by the human brain, consisting of numerous interconnected processing layers. These models emulate the operational principles of the brain. Specifically, when the brain encounters new information, it seeks to relate it to pre-existing knowledge to comprehend it. Consequently, the brain deciphers the information by labelling and categorizing the elements, a process that deep machine learning models replicate for training purposes

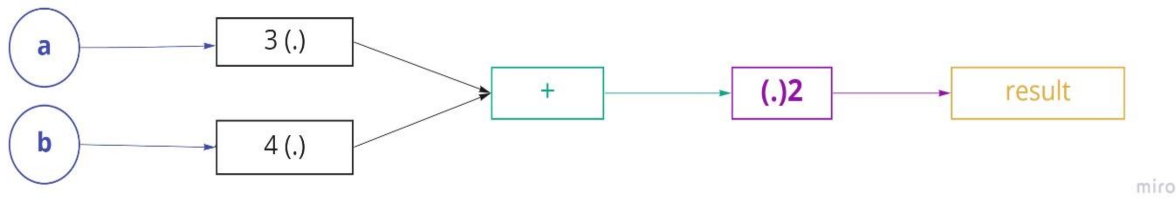
The term "deep" in the context of machine learning is technical and refers to the number of layers in an ANN. A standard ANN consists of three types of layers: the input layer (which receives the data), the output layer (which produces the result of data processing), and the hidden layer (which identifies patterns within the data). While a single-layer neural network can exist, a deep ANN is distinguished from a shallow ANN (with a single hidden layer) by its multiple hidden layers, enabling it to perform more complex estimations (Martin et al., 2018; Jakhar and Kaur, 2019; Nguyen et al., 2019; An and Moon, 2019). As data moves from one hidden layer to another, simpler features are recombined and recomposed into complex features. The layers of the NN consist of organized nodes that are not interconnected. Each node from one layer is connected to each node from the next layer, and each one from the input. The number of connections between layers equals the number of parameters in a layer plus one – for the bias term. The bias neuron is a unique neuron added to each layer in the neural network, which stores the value of 1. This allows for the shifting or "translation" of the activation function left or right on the graph. In simple terms, DML performs exceptionally well on unstructured data and has higher accuracy than traditional machine learning (ML), but it requires a large database, or in other words – big data. The most common methodology for building NN architecture structure is by applying the production-ready computer language Python and its libraries across potentially thousands of multi-GPU servers via computational graphs. Some of these libraries include:

- Scikit-Learn, which is user-friendly and implements many supervised and unsupervised ML algorithms through a consistent interface in Python.
- TensorFlow, a more complex library for distributed numerical computation using data flow graphs.
- Keras, a Python wrapper library, which can be built independently on top of TensorFlow;
- Statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models as well as for conducting statistical tests and statistical data exploration. The results are tested against existing statistical packages to ensure that they are correct (Lazzeri, 2021);
- Darts is a Python library for user-friendly forecasting and anomaly detection on time series.

All ANN run on the principle of the computational or dataflow graphs. A computational or dataflow graph is a form of a directed graph where vertices or nodes describe operations, while edges represent data flowing between these operations. If an output variable represents the result of applying a binary operation to two inputs a and b, then we draw directed edges from x and y to an output node representing the result (z) and annotate the vertex with a label describing the performed computation via Jupyter Notebook and python. In the example above

(Figure 1), the input data in the circles (a, b) are transformed via the black, blue and purple rectangles representing functions (computational nodes) which compute the example equation $(3a + 4b)^2 = \text{z}$ (1).

Figure 1: Computational graph



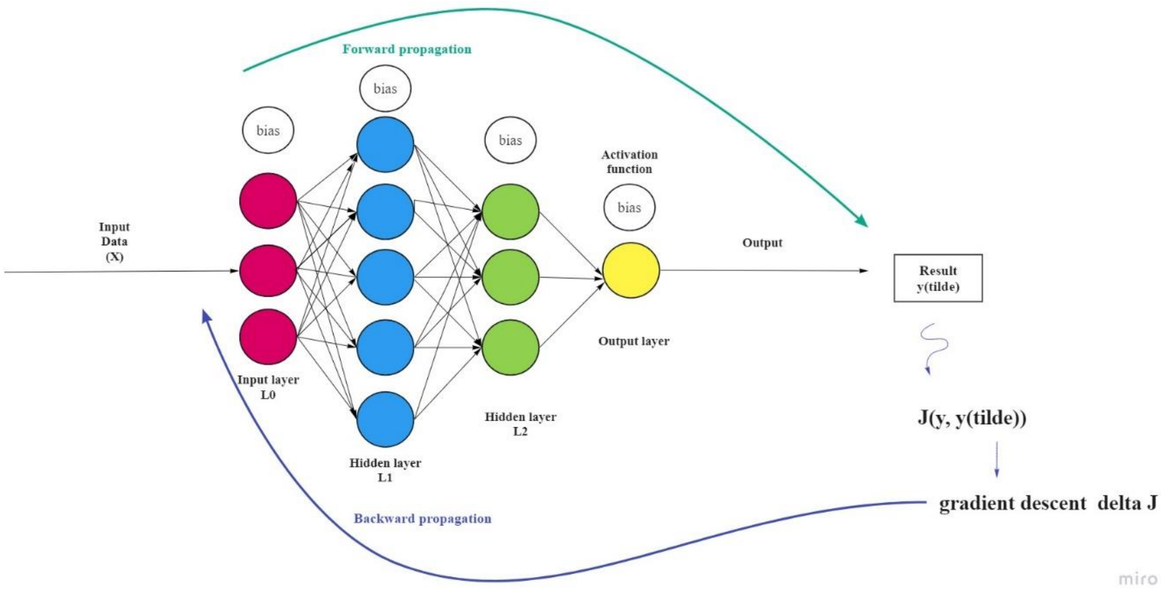
Source: author's own elaboration

Python, a programming language developed by Guido van Rossum and first introduced in 1991 (Nguyen et al., 2019), has been effectively utilized by major corporations such as Google, Microsoft, Facebook, Tesla, YouTube, Airbnb, and others. Researchers have adopted Python for ML and DML in the fields of tourism and hospitality for tasks such as property management, big data analysis, market segmentation, customer targeting, optimal price calculation, and determining preferences and behaviours. Jupyter Notebook is an open-source tool that supports computations for DML using Python and can function offline (Nguyen et al., 2019).

ANN can be single layer and multilayer. The single-layer ANN, has a single layer of nodes. Each node in the single-layer connects directly to an input variable and contributes to an output variable (similar to the computational graph depicted on Figure 1). Such ANN can be applied for regression models mainly for solving linear ordinary differential equations. Whereas, the multilayered ANNs could be applied for complex regression or classification tasks. A standard multi-layer ANN has three types of layers: an input layer (receives the input data), an output layer (produces the output of the data processing), and a hidden layer (extracts the patterns from the dat. A deep Artificial Neural Network (ANN) is distinguished from a shallow ANN (which has only one hidden layer) by its numerous hidden layers, enabling it to perform more intricate tasks (Martin et al., 2018; Jakhar and Kaur, 2019; Nguyen et al., 2019; An and Moon, 2019; Oreshkin et al., 2020). The layers of an ANN are composed of structured nodes, with no interconnections between them. In a fully-connected neural network, every neuron in each layer is linked with every neuron in the subsequent layer, as well as with all inputs. The total number of connections between layers equals the number of parameters in the layer, plus one additional for the bias element. This bias element is a unique neuron added to each layer in the neural network that simply holds a value of one, allowing for the activation function to be shifted or "translated" along the graph.

Fully-connected neural network (FC**)** is a type of multi-later ANN, where each neuron in a layer is connected to all neurons in the next layer. The input data (X) passes via the layers on the NN (forward propagation) and after the output layer with the help of the activation function, we have a result. This result has to be compared with the total cost function/loss function $J(y, \tilde{y})$, where y is the expected result and $\tilde{y}$ is the real result. This loss functions (ready to use the function from the API of TensorFlow library) have their gradients and they allow us to make the opposite process – backward propagation. Here TensorFlow is very helpful because when a model based on gradient descent is been applied and only forward propagation has been formed (the combination of layers) the library calculates the gradients and applies the backward propagation by itself. A loss function can be applied by preferences or calculated bearing in mind that this is the most important feature of a neural network. FC can be implemented in the industry of tourism and hospitality for market segmentations, arrivals predictions, classifying tourists' behaviour, etc.

Figure 2: Fully-connected neural network



Source: author's own elaboration

Neural networks are constructed from individual components that process incoming signals to generate a corresponding output value, which is then disseminated to the connected elements. There are three primary components:

- A collection of connections, or synapses, each characterized by a weight factor wij, which can assume either positive or negative values.
- An adder that aggregates the input signals, each multiplied by its respective weight factor. The linear adder is the most commonly used type.
- An activation function, or output function, which transforms the total received input into the neuron's output signal.

In mathematical terms, the operation of a neuron can be described as follows. Let us denote (x_1, x_2, x_3,...,x_i) as the input signals, w_{ij} as the synaptic weights of the neuron, and w_i as the connection between neurons. x_i are the input signals to the neuron and w_o represents the bias weight. So the output data z of the neuron is defined accordingly:

$$z = w_0 \sum_{i=1} (w_i x_i) \tag{1}$$

The most common types of activation functions that determine the output signal of the neuron are a single jump function or a threshold function:

$$\mathrm{f(z)} = \begin{pmatrix} 1: & z \geq 0 \\ 0: & z < 0 \end{pmatrix} \tag{2}$$

and the sigmoid function, which can be described as follows:

$$\mathrm{y(x)} = \frac{1}{1 + \epsilon^{\beta z}} \tag{3}$$

 ISSN 1337-0839 (print) / 2585-7258 (online)

The sigmoidal is most commonly used in error backpropagation networks, where the fast-growing function works to maintain the balance between linear and non-linear behaviour. To overcome certain disadvantages of the sigmoidal function, a hyperbolic function (*tanh*) can be used, which functions in a larger space [−1, 1]:

$$(\mathrm{z}) = \begin{pmatrix} 1: & z > 0 \\ 0: & z = 0 \\ -1: & z < 0 \end{pmatrix} \tag{4}$$

Convolutional Neural Networks (CNNs) are networks specifically designed to work with images (or translation invariant data more generally) (Nguyen et al., 2019). Current applications include a wide variety of image classifiers and have recently been applied in the field of tourism and hospitality – for forecasting and forecasting (Lu et al., 2020a), for personalized travel information (Han et al., 2018), for sentiment analysis for a recommender system (An and Moon, 2019) and sentiment analysis through eWOM (Martin et al., 2018), for cultural information (Hirotsu et al., 2020) and thus, helping businesses understand customer perception and improve their offerings.

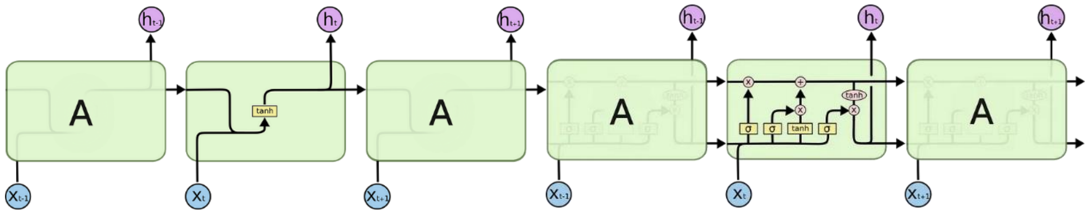
Recurrent Neural Networks (RNN) are specially designed to deal with sequential data (Nguyen et al., 2019). They are widely used by scientists in NLP in the sphere of tourism and hospitality for translations, time series analysis, sentiment analysis, tourist flows and demand forecasting. Furthermore, when RNNs are used in conjunction with CNNs they can be applied for applications such as video classifications. Namely on such ANN is based the method of the current research paper, which is further elaborated in the Methodology section.

DML excels at handling unstructured data and offers greater accuracy than traditional ML, but it necessitates a large database, or in other words, big data (Stylos et al., 2021). Neural networks can exhibit both linear and non-linear characteristics. The ability to handle non-linearity is distributed throughout the network and influences how the input signal is shaped, which can also be non-linear. One of the primary benefits of neural networks is supervised learning, where the input information is mapped to the output. To accomplish this, the synaptic weights are adjusted based on a specific set of training examples. Each training example comprises an input signal and its corresponding desired response. During the training phase, the network alters the synaptic weights to minimize the discrepancy between the desired output signal and the actual output, according to predetermined statistical criteria. Supervised learning can also be viewed as an optimization task.

2. Methodology

For many years, accurately predicting tourism forecast, like overnight stays has been a challenge. To address this, many researchers developed new automated methods that combine traditional forecasting techniques with modern DML models (Frechtling, 2001; Crone and Graffeille, 2004; Akin, 2015; Claveria et al., 2016; Chenguang et al., 2017; Li and Cao, 2018; Brownlee, 2018; Golshani et al., 2018; Cheng et al., 2019; Dwyer et al., 2020; Bi et al., 2021; Egger, 2022; Herzen et al., 2022; Dowlut and Gobin-Rahimbux, 2023; Lu et al., 2020b; Athanasopoulos et al., 2024; Bufalo and Orlando, 2024). Li and Cao (2018) propose the following RNN model (Figure 3): one input unit, one output unit, and one recurrent hidden unit expanded into a full network. x_t is the time step input, o_t is the output of a time step $t.$ s_t is the hidden state at a time step and is the "memory" of the network. W, U, V are parameters in different network layers.

Figure 3: Stanart RNN and LSTM model



Source: adapted by the author based on Li and Cao (2018)

Unlike traditional deep neural network that uses different parameters at each layer, RNN shares the same parameters (W, U, V) in all steps. This greatly reduces the total number of parameters we need to learn. In the forward propagation process, the current layer calculates the input data and outputs to the next layer according to the network connection and weight $s_j(t)$ is the output of the hidden layer at time *t*. The calculation process is:

$$s_j(t)=f\left(\sum_{i}^{l} x_i(t)v_{ji} + \sum_{h}^{m}(t\text{-}1)u_{jh} + b_j\right) \tag{5}$$

where $\sum_i^l x_i(t)v_{ji}$ is the input of the input layer at the moment

$t. \sum_{h}^{m}. (t-1)u_{jh} + b_j)$ is the input of the currently hidden layer

$(t-1). b_j$ is the bias

$f(\cdot)$ is a map function, which applies a given function to each element of a collection that is typically non-linear as tanh or ReLU

According to Li and Cao (2018), the forward pass of a RNN is the same as that of a multilayer perceptron with one hidden layer, except those activations arrive at the hidden layer from both the current external input and the hidden layer activations from previous time steps. Considering, an input sequence of length Tx, which is introduced to RNN with I input units, H hidden units and K output units. $x_t$is the input value i at the moment t, a $a_h^t b_h^t$ is the network input to a unit h at the moment t and unit activation h at the moment t. The hidden units are:

$$a_h^t = \sum_{i=1}^{1} \omega_{ih} x_i^t + \Sigma_{h'}^{H} = 1\omega_{h'h} b_{h'}^{t-1} \tag{6}$$

Non-linear, differentiable activation functions are then applied:

$$b_h^t = \Theta_h(a_h^t) \tag{7}$$

The network inputs to the output units can then be calculated by:

$$a_\kappa^t = \sum_{x=1}^{H} \omega_{hk} b_h^t \tag{8}$$

The subtlety is that for RNN the loss function depends on the activation of the hidden layer not only through its influence on the source layer, but also through its influence on the hidden layer at the next time step. Therefore:

$$\delta_h^t = \omega'(a_h^t) \sum_{k=1}^{K} \delta_k^t \omega_{hk} + \sum_{h=1}^{H} \omega_{h'}^{t+1} \omega_{hh'} \tag{9}$$

Long-Short Term Memory (LSTM) networks were specifically engineered to tackle the issue of long-term dependencies that recurrent neural networks (RNNs) struggle with due to the vanishing gradient problem. Unlike traditional feedforward neural networks, LSTMs possess feedback connections. This unique feature allows LSTMs to handle whole data sequences, like in our case time series, without processing each data point separately. Instead, they preserve valuable information from past data in the sequence to aid in the processing of new data points.

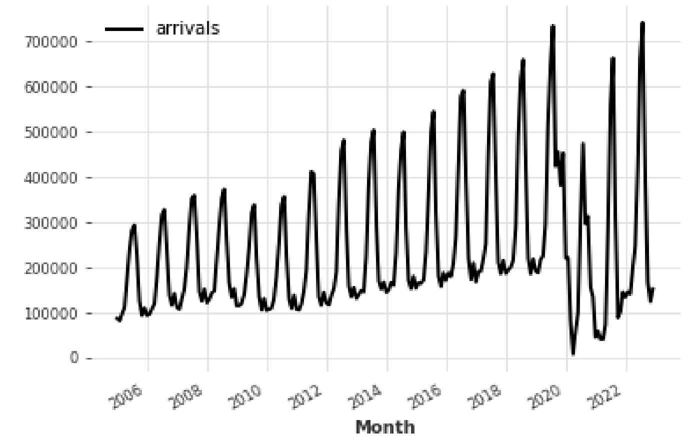
Gated Recurrent Unit (GRU) is an improved model of LSTM, which controls the switch of memorizing and forgetting by setting multiple threshold gates (Lu et al., 2020). Both LSTM and GRU can solve the limitation problem of handling long-term dependencies well. which led to their successful application on various sequence learning problems, such as forecasting.

The method was deployed on a user-friendly Python platform (Jupyter Notebook) which uses a question-based framework to guide users. We compared various forecasting methods, including basic, classical, machine learning, and deep learning approaches. Our goal was to identify the most effective method with the best settings for predicting monthly overnight stays in Bulgarian accommodations from 2002 to 2023. The data was gathered via the statistical office of the European Union (Eurostat, 2023), the Bulgarian National Statistical Institute (National Statistical Institute, 2024) and an emailed paper report from the Ministry of Tourism. To ensure accurate results, we utilized number of python libraries as well as well-regarded DARTS library in order to establish whish time series forecasting model performs better and may be applied for similar tasks in the future by for research and practical purposes.

3. Results

From the python environment a TensorFlow environment was activated and then Jupyter Notebook wrapper was created where via the Darts libraries the data was deployed, pre-processed and observed (Figure 4).

Figure 4: Total number room nights

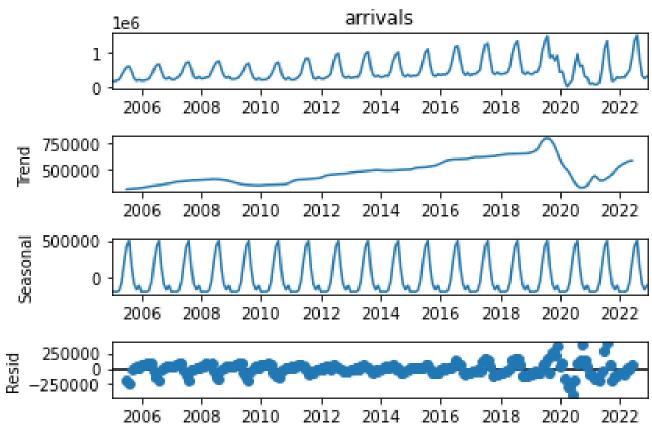


Source: author's own elaboration

Figure 4 represents a line graph showing the total number of room nights from 2005 to 2022. The graph shows a cyclical pattern, indicating seasonality in the data. Peaks in the graph likely correspond to high tourist seasons, while troughs could indicate off-season periods. This kind of data can be crucial for understanding market trends and planning in the tourism industry.

Via the time series decomposition, the data seasonality, trend and residuals were obtained. The trend represents the trend isolation where a general increasing trend over time can be observed; the seasonal decomposition can provide removing the trend from the original data by subtracting and a clear seasonal trend is depicted; the residual provides an insight of the data noise (Figure 5). The seasonal decomposition can provide removing the trend from the original data by subtracting, and a clear seasonal trend is depicted. Seasonality is the repeating short-term cycle in the series. In the "Seasonal" graph, we can see a consistent pattern of peaks and troughs over the years, which could be related to factors such as tourist seasons or specific events that occur at regular intervals. Residuals are the difference between the actual observations and the values predicted by the model. In the "Resid" scatter plot, the points seem to be randomly scattered around zero, indicating that the model has done a good job capturing the trend and seasonality, leaving only the random noise in the residuals. This allows for an easier understanding of how each component contributes to overall patterns in data over time.

Figure 5: Time series decomposition



Source: author's own elaboration

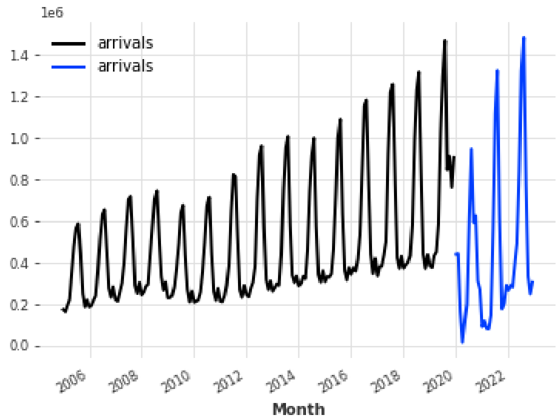
Following which, for the purpose of the forecasting task, the time series were split (Figure 6). This is a common practice in time series analysis where the data is divided into a "training" set that includes the historical data, and a "test" or "validation" set that is used for predictions. This method helps validate the performance of the forecasting model on unseen data. As the last 36 months of data were used for the prediction, this suggests that the model was "trained" on the data up to that point, and then used to forecast the subsequent 36 months. This approach allows the model to learn from the most recent trends and patterns in the data.

With accordance to the data observation, the models for the forecast were created and performed, in order to evaluate the best performing model, namely

- ExponentialSmoothing: Holt-Winters Exponential Smoothing, used for time series forecasting when the data has linear trends and seasonal patterns;
- TBATS: Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend and Seasonal components;
- AutoARIMA: Automatically discover the optimal order for an ARIMA model;
- Theta is a simple method for forecasting the involves fitting two θ-lines, forecasting the lines using a Simple Exponential Smoother, and then combining the forecasts from the two lines to produce the final forecast;

 ISSN 1337-0839 (print) / 2585-7258 (online)

Figure 6: Total number room nights



Source: author's own elaboration

- LSTSM: deep ANN using vanilla RNN or GRU instead, replace LSTM by RNN or GRU, respectively.

4. Discussion

For models estimation the Mean Absolute Percentage Error (MAPE) was used as it is quite convenient and scale independent for empirical experiment purposes due to the fact that measures the average magnitude of error produced by a model, or how far off predictions are on average. In Darts library it is a simple function call instead an equational estimation by a software interface. MAPE models' estimations are represented in Table 1.

Table 1: MAPE Comparison of Forecasting Models

Model	MAPE
Exponential Smoothing	49.96%
TBATS	56.45%
AutoARIMA	77.88%
Theta	51.13%
LSTM	22.16%

Source: author's own elaboration

The results presented in Table 1 highlight the superior performance of the LSTM model in forecasting overnight stays, particularly in the context of the volatile and unprecedented fluctuations caused by the COVID-19 pandemic. The MAPE of 22.16% for LSTM indicates a significantly lower average error compared to other models tested, including traditional time-series methods like Exponential Smoothing, TBATS, and AutoARIMA. This finding aligns with a growing body of international research that underscores the effectiveness of LSTM networks in tourism demand forecasting. Thus, we can assume that the ANN model, including LSTM, was able to adapt to the sudden changes in tourism patterns during the pandemic, outperforming traditional models that struggled to capture the non-linearity and volatility of the data (Silva and Alonso, 2020). What is more, the results considerably demonstrate the ability of LSTM to effectively model the complex relationships between multiple variables influencing tourism demand, such as economic indicators, seasonality, and events (Dowlut and Gobin-Rahimbux, 2023). Numerous other studies have confirmed the superiority of LSTM in various tourism forecasting scenarios, highlighting its ability to handle long-term dependencies and capture underlying patterns in time-series data (Dwyer et al., 2020; Zhang et al., 2021; Provenzano and Volo, 2022; Bufalo and Orlando, 2024).

The findings of this study reaffirm the potential of LSTM networks as a powerful tool for forecasting overnight stays in the tourism industry. By capturing the complex dynamics of demand, even in the face of unexpected disruptions, LSTM offers a valuable asset for researchers, practitioners, and policymakers seeking to navigate the ever-changing landscape of tourism.

5. Conclusions

Various forecasting methods using real data on monthly overnight stays in Bulgaria were deployed via Jupyter Notebook and the computer language python in order to estimate which of them could generalise well thus providing valuable tool for tourism experts. This included seasonal models, exponential smoothing, and neural networks. The results demonstrate that a specific neural network type, RNN-LSTM, performed best according to two different accuracy measures. Recent dramatic drops in tourism due to the pandemic make traditional forecasting methods less reliable. Interestingly, these classical models did worse than expected, while the LSTM neural network performed rather well.

This research is valuable for both academic study and practical applications of artificial intelligence (AI) in tourism as LSTM can be perceived as a powerful and reliable technique for forecasting tourism data as it outperformed one of the most applicable methods like AutoARIMA and exponential smoothing, even those based on Fourier transformations methodology. The future research elaborations can focus on using other deep learning models to forecast tourism data, including single or multivariate models for domestic and international flows. We'll also explore combining tourism data with economic and marketing factors using different neural network architectures.

Finally, the model can be continuously improved by incorporating new data, features optimisation, and parameters adjusting based on the foreseeable results. In such manner, forecasts can stay relevant and reflect the changing dynamics of one of the most dynamic economic sectors, namely tourism. Such improved models could then be applied by various tourism stakeholders to benefit their businesses and operational processes.

Author contributions: All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

Funding: This research was supported by the project RP-A1/24 “Upgrading the capacity of the Scientific research centre „Observatory Economics“ for students and doctoral students training and their participation in scientific research – phase 2”.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to [insert reason here].

Conflicts of Interest: The authors declare no conflict of interest.

References

Akin, M. (2015). A novel approach to model selection in tourism demand modeling. *Tourism Management*, *48*, 64-72.

An, H., & Moon, N. (2019). Design of recommendation system for tourist spot using sentiment analysis based on CNN-LSTM. *Journal of Ambient Intelligence and Humanized Computing*.

Athanasopoulos, G., Hyndman, R. J., Kourentzes, N., & Panagiotelis, A. (2024). Forecast reconciliation: A review. *International Journal of Forecasting*, *40*(2), 430-456.

Bi, J.-W., Li, H., & Fan, Z.-P. (2021). Tourism demand forecasting with time series imaging: A deep learning model. *Annals of Tourism Research*, *90*.

Brownlee, J. (2018). *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery.

Bufalo, M., & Orlando, G. (2024). Improved tourism demand forecasting with CIR# model: a case study of disrupted data patterns in Italy. *Tourism Review*, *79*(2), 445-464.

Cheng, L., Chen, X., Vos, J. D., Lai, X., & Witlox, F. (2019). Applying a random forest method approach to model travel mode choice behavior. *Travel Behaviour and Society*, *14*, 1-10.

Chenguang, D., Song, W. H., & Shen, S. (2017). New developments in tourism and hotel demand modeling and forecasting. *International Journal of Contemporary Hospitality Management*, *29*(1), 507 - 529.

Claveria, O., Monte, E., & Torra, S. (2016). Combination forecasts of tourism demand with machine learning models. *Applied Economics Letters*, *23*(6), 427-431.

Crone, S. F., & Graffeille, P. C. (2004). An Evaluation Framework for Publications on Artificial Neural networks in Sales Forecasting. *Proceedings of the International Conference on Artificial Intelligence (IC-AI '04)*, 221-227.

Jakhar, D., & Kaur, I. (2019). Artificial intelligence, machine learning and deep learning: definitions and differences. *Clinical and Experimental Dermatology*, *45*(1), 131-132.

Dowlut, N., & Gobin-Rahimbux, B. (2023). Forecasting resort hotel tourism demand using deep learning techniques – A systematic literature review. *Heliyon*, *9*(7), e18385.

Dwyer, L., Forsyth, P., & Dwyer, W. (2020). *Forecasting tourism demand*. Bristol: Blue Ridge Summit: Channel View Publications.

Egger, R. (2022). Applied Data Science in Tourism: Interdisciplinary Approaches, Methodologies, and Applications. *Zeitschrift Fur Tourismuswissenschaft*, *13*(2), 303-304.

Eurostat. (2023). Arrivals at tourist accommodation establishments - monthly data. Eurostat. https://ec.europa.eu/eurostat/databrowser/view/TOUR_OCC_ARM__custom_6400112/default/table

Frechtling, D. (2001). *Forecasting Tourism Demand: Methods and Strategies*. Oxford: Butterworth-Heinemann.

Geron, A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow. O'Reilly Media. https://www.oreilly.com/catalog/errata.csp?isbn=9781491962299

Golshani, N., Shabanpour, R., Mahmoudifard, S. M., Derrible, S., & Mohammadian, A. (2018). Modeling travel mode and timing decisions: Comparison of artificial neural networks and copula-based joint model. *Travel Behaviour and Society*, *10*, 21-32.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning. MIT Press*. http://www.deeplearningbook.org

Han, X., Li, J., Hu, Y., Yuan, J., & Ye, S. (2018). Tourism Information Push System Based on Convolutional Neural Network. *3rd International Conference on Advances in Energy and Environment Research (ICAEER 2018)*.

Herzen, J., Lassig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., & Grosh, G. (2022). Darts: User-Friendly Modern Machine Learning for Time Series. *Journal of Machine Learning Research*, *23*(1), 5442-5447.

Hirotsu, T., Hirota, M., Araki, T., Endo, M., & Ishikawa, H. (2020). Tourism application with CNN-Based Classification specialized for cultural information. *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services (iiWAS2019)*, 8-14.

Lazzeri, F. (2021). *Machine Learning for Time Series Forecasting with Python*. Indianapolis, Indiana: John Wiley & Sons, Inc.

Li, Y., & Cao, H. (2018). Prediction for Tourism Flow based on LSTM Neural Network. *Procedia Computer Science*, *129*, 277-283.

Lu, W., Jin, J., Wang, B., Li, K., Liang, C., Dong, J., & Zhao, S. (2020b). Intelligence in Tourist Destinations Management: Improved Attention-based Gated Recurrent Unit Model for Accurate Tourist Flow Forecasting. *Sustainability*, *12*(4), 1390.

Lu, W., Rui, H., Liang, C., Jiang, L., Zhao, S., & Li, K. (2020a). A Method Based on GA-CNN-LSTM for Daily Tourist Flow Prediction at Scenic Spots. *Entropy*, *22*(3), 261.

Martin, C. A., Torres, J. M., Aguilar, R. M., & Diaz, S. (2018). Overcoming "Big Data" Barriers in Machine Learning Techniques for the Real-Life Applications. *Complexity*, 1234390.

National Statistical Institute. (2024). *National Statistical Institute*. https://www.nsi.bg/

Nguyen, G., Dlugolinsky, S., Bobak, M., Tran, V., Garcia, A. L., Heredia, I., Malik, P., & Hluchy, L. (2019). Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review*, *52*(1), 77–124.

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-Beats: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting. *ICLR*. https://iclr.cc/virtual_2020/papers.html?filter=titles&search=

Provenzano, D., & Volo, S. (2022). Tourism recovery amid COVID-19: The case of Lombardy, Italy. *Tourism Economics*, *28*(1), 110–130.

Silva, I., & Alonso, H. (2020). New developments in the forecasting of monthly overnight stays in the North Region of Portugal. *Journal of Applied Statistics*, *47*(13-15), 2927–2940.

Stylos, N., Zwiegelaar, J., & Buhalis, D. (2021). Big data empowered agility for dynamic, volatile, and time-sensitive service industries: the case of tourism sector. *International Journal of Contemporary Hospitality Management*, *33*(3), 1015-1036.

Zhang, H., Song, H., Wen, L., & Liu, C. (2021). Forecasting tourism recovery amid COVID-19. *Annals of Tourism research*, *87*, 103149.