INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Tourism Flow Forecasting for Inbound European Travel

André Guilherme Ramalho Garcia

Master in Computer Engineering

Supervisor:
Adriano Martins Lopes, PhD, Invited Assistant Professor,
Iscte

October, 2023

TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology

Tourism Flow Forecasting for Inbound European Travel

André Guilherme Ramalho Garcia

Master in Computer Engineering

Supervisor:
Adriano Martins Lopes, PhD, Invited Assistant Professor,
Iscte

October, 2023

Tourism Flow Forecasting for Inbound European Travel

Acknowledgements

I would first like to show my deepest gratitude to my adviser, Professor Adriano Lopes, ISTAR researcher at Iscte-IUL, whose availability and expertise allowed me to confidently take every step during this journey and feel supported in every needed way. Additionally, his focus on defending my interests was very important to me.

I would also like to thank Professor Fernando Brito e Abreu, ISTAR researcher at Iscte-IUL, who, together with Professor Adriano Lopes, introduced me to the theme of this work and to the opportunity to participate in the very noble project RESETTING.

I would like to thank everyone involved in the RESETTING project, specially on WP2, for sharing with me the responsibility of developing technology to help SMEs in need. A special thanks to my fellow MSc student Nuno Dias, also inserted in WP2, with whom I shared a lot of hours of work for this project, and is as hardworking as he was fun to work with. I would also like to extend my sincere gratitude to BRU, at Iscte, for their financial support through the award of a research fellowship in regards to my participation in RESETTING, which not only eased the financial aspects of my academic journey, but also allowed me to contribute effectively to the project's goals while pursuing my educational aspirations. I am deeply appreciative of this opportunity and the invaluable experience it provided. Additionally, I would like to thank my classmates during my Master's degree, whom I shared thoughts and ideas with, and supported me when I needed it most.

Finally, I wish to express my very profound thanks to my parents, who provided me with unfaltering support and love, not only for the duration of this work, but during my entire life, I'm forever grateful. To my sister, who, no matter how many petty fights or silly arguments, I have nothing but love and admiration for. To my grandparents, who during my childhood were there for me every step of the way, and specially to my grandmother, who taught me how to be successful in school and in life. To my girlfriend, who, no matter how far or how close, has loved and supported me and immensely helped me grow personally every single day. And to my family in general, for caring and asking about the progress of my work at any chance they got. I love you all.

Resumo

O turismo desempenha um papel fundamental na economia europeia. Entre outros aspectos, a gestão da procura de produtos e serviços é fundamental para o desenvolvimento do turismo, pelo que modelos de previsão contribuem para esse esforço. Isto é ainda mais relevante hoje em dia à medida que o setor está a recuperar da perturbação causada pela pandemia global da COVID-19.

Neste estudo desenvolvemos modelos de previsão avançados para a procura turística. Em particular, modelos baseados em algoritmos de Deep Learning, como Long Short-Term Memory (LSTM) e Gated Recurrent Unit (GRU), e considerando a frequência de dados diária e mensal. A contribuição vai para além da mera seleção de algoritmos – também analisa as nuances de *feature engineering*, incorporando dados de variáveis exógenas, como volume de pesquisas em motores de busca, inflação, PIB e taxa de câmbio.

Através de uma avaliação rigorosa, apoiada por métricas de avaliação de previsões, como RMSE, R-quadrado e MAPE, constata-se que os modelos GRU superaram consistentemente os modelos LSTM. Além disso, a nossa investigação revelou que a inclusão de factores externos teve um impacto limitado no aumento da precisão das previsões.

Este trabalho serve como um recurso valioso para o setor do turismo para além do domínio académico. Os resultados deste estudo são disponibilizados numa aplicação web desenvolvida no contexto do projecto RESETTING, financiado pela União Europeia. Refira-se que este projeto visa ajudar as PME do setor do turismo na recuperação dos anos difíceis e restritivos durante a pandemia.

Palavras-Chave:

Turismo Europeu, Previsão de Turismo, Modelos de Previsão

Abstract

Tourism plays a pivotal role in the European economy. Among other aspects, the management of demand of products and services is critical to tourism development, so forecasting models contribute to such endeavour. This is even more relevant nowadays as the sector is recovering from the disruption caused by the global COVID-19 pandemic.

In this study we build state-of-art forecasting models for tourism demand. In particular, models based on Deep Learning algorithms, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), and considering both daily and monthly data frequency. But the contribution goes beyond mere algorithm selection – it also delves into the nuances of feature engineering, incorporating data from exogenous variables such as search engine volume of searches, inflation, GDP and currency exchange rate.

Through rigorous evaluation, supported by forecasting evaluation metrics, such as RMSE, R-squared and MAPE, it was discovered that GRU models consistently outperformed LSTM models. Additionally, our exploration revealed that the inclusion of external factors had limited impact on enhancing forecast accuracy.

This work serves as a valuable resource for industry stakeholders beyond the academic realm. Its findings are used to deploy forecasts into a web-application developed in the context of the European Union funded project RESETTING, which aims to help SMEs of the tourism sector as they make a comeback from the tough restrictive years during the pandemic.

Keywords:

European Tourism, Tourism Forecasting, Forecasting Models

Contents

List of Figures

List of Tables

Glossary

Darts A PyTorch based timeseries forecasting and anomaly detection library. 17

PyTorch PyTorch is a machine learning framework based on the Torch library. It is used for applications such as computer vision, natural language processing or timeseries forecasting. 35

pytrends Unofficial API for Google Trends. Allows simple interface for automating downloading of reports from Google Trends. xv, 22, 23, 24

RESETTING “This project aims at facilitating a transition towards more resilient, circular, and sustainable operating models of European tourism companies through the testing and integration of innovative digitally-driven solutions that reduce unnecessary burdens, improve the quality of the tourism experience, contribute to the decarbonization of the tourism sector and a more inclusive economic growth – not only for small and medium-sized companies but also for the residents of the destinations.” (https://www.resetting.eu/) 2, 55, 56, 60

statsmodels 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. 17

Acronyms

NN Neural Network 7

R2 R-Squared 15, 34, 48
RBF Radial Basis Function 7, 10
REST API RESTful API 55
RMSE Root Mean Square Error 15, 34, 36, 48, 49
RNN Recurrent Neural Network 10, 17, 18

SARIMA Seasonal Auto-Regressive Integrated Moving Average 2, 17, 33, 34, 35, 48, 51, 61
SME Small or Medium-sized Enterprise 2, 55, 56
SVM Support Vector Machines xi, 7, 9, 10, 11
SVR Support Vector Regression 7, 10, 11

tanh Hyperbolic Tangent Function 18, 19
TDF Tourism Demand Forecasting 2, 5, 6, 10
TVP Time-Varying Parameter 6, 7, 8

VAR Vector Auto-Regressive 6, 7, 8

WTTC World Travel & Tourism Council 1

1

Introduction

In this chapter, some introductory aspects will be addressed, such as the motivation for this work, its goals and research questions, as well as the methodology to be followed.

1.1 Background and Motivation

Tourism is an important sector of the European economy, accounting for a significant portion of the business in the continent. According to the World Travel & Tourism Council (WTTC), tourism's total Gross Domestic Product (GDP) contribution in Europe was equivalent to 9.2% in 2019 [1]. However, this number was significantly lower in the following year, 2020, having declined to 5.2%. While this steep decline seems to imply that tourism is losing its power and appeal to the general population, it is explained by the Covid-19 global pandemic that was officially announced in March of 2020 by the World Health Organization (WHO) [2]. Most European countries imposed new rules regarding public health safety due to the new corona virus, which made traveling much harder and more expensive than in the previous year. Crossing most European borders required a negative Covid-19 test and/or a proof of vaccination, and in some cases, nation borders were completely closed between certain countries, which hindered international travel. In truth, the desire to travel did not seem to decrease at all, as the European Travel Commission showed in September of 2020, with a study on Monitoring Sentiment for Intra-European Travel [3]. It states that 73% of surveyed Europeans were planning to travel in the following six months, and that, despite the advances of the pandemic, the key sources of distress for travelers were price inflation and personal finances. In fact, the total European GDP contribution of tourism raised to 6.2% in 2021, an increase of 28%. Although tourism seems to have had a setback in terms of its expression in the European economy, it looks to be slowly gaining back its original spot.

Given the importance of tourism for the global and, in this case, European economy, it is of interest to explore and create tools that drive the evolution and development of tourism businesses, such as hotels, travel agencies or tour companies. One important aspect of improving touristic products and services is demand management, in which tourism related businesses attempt to predict demand in order to adjust the conditions for the future. This is a sensitive matter, as under or overestimating demand can easily become costly. Underestimating demand can cause loss of business opportunity and dissatisfaction from clients, whereas overestimating it can lead to excessive spending. Tourism demand forecasting models are tools that can help tourism business owners to adjust their demand related investments to approach optimal performance.

In this context, this work will create state-of-art solutions for Tourism Demand Forecasting (TDF), with the support of the European-funded project RESETTING. It has the objective of serving the needs of Small or Medium-sized Enterprises (SMEs) in the European tourism industry with their recovery efforts after the COVID-19 pandemic. This work's contribution to this project and more information about it is mentioned in more detail on chapter 5, where the development process of a web-app, which has the purpose of deploying the results of the developed forecasting models, is shown.

1.2 Goals and Research Questions

The primary objective of this research is to advance the understanding of Inbound European tourism forecasting models, shedding light on the intricacies of this dynamic field. It seeks to provide tourism industry stakeholders with more precise predictive models and to find insights into the best processes, data and implementations to achieve them.

The main focus regarding models to be used in this research is on deep learning models, specifically, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Recently, these models have been gaining traction in the field of time-series forecasting when compared to the simpler Auto-Regressive Integrated Moving Average (ARIMA). Their performance can be better than the traditional techniques, depending on the data and implementation [4].

Therefore, the first research question for this work is: "How do deep learning models, specifically GRU and LSTM, compare to traditional forecasting methods like Exponential Smoothing and Seasonal Auto-Regressive Integrated Moving Average (SARIMA), and to each other, in predicting inbound European tourism arrivals?"

In addition to evaluating these algorithms, the external variables used as features are also the focus of this work. Specifically, Google Trends data, inflation, GDP and currency exchange rate could be incorporated to improve model accuracy.

Hence, the following research question arises: "Can the integration of external variables, such as Google Trends data, inflation, GDP, and currency exchange rates, enhance model accuracy in the context of inbound European tourism forecasting?"

By addressing these research questions, this research aims to contribute to the field of Inbound European tourism forecasting, paving the way for further research and development in this ever evolving domain, while also offering practical solutions to support post-pandemic recovery efforts.

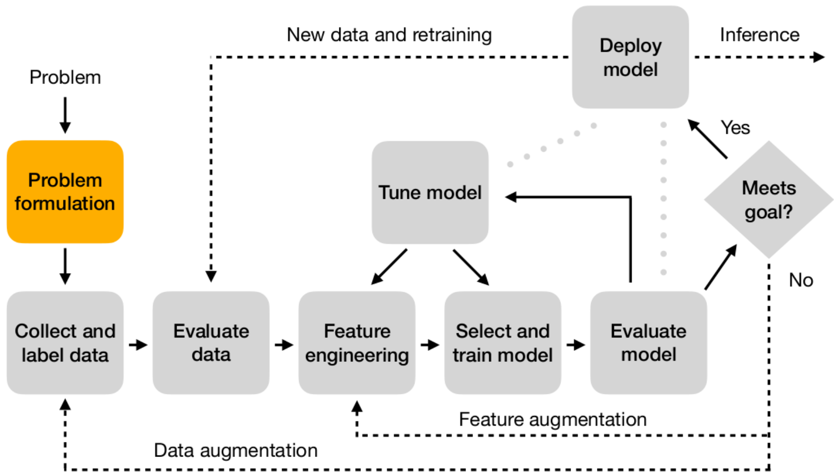


Figure 1.1: This work's methodology, based on CRISP-DM and Machine Learning and Deep Learning pipelines.

Source: Handouts from course unit entitled *Big Data Algorithms*, Iscte-IUL, 2022.

1.3 Methodology

In order to fulfill this work's goals, a carefully tailored methodology was followed. It was mostly based on the CRISP-DM framework. Figure 1.1 has a diagram representing the stages undertaken in this work. Here is a brief explanation of each step.

Problem formulation. Sometimes called business understanding, this is the step when the main problem that is trying to be solved is defined, and the objectives for the project are set.

Collect and label data, evaluate data. These steps correspond to the data collection and understanding phase, where the data is collected and its volume, quality and frequency are evaluated, to ensure that it is suitable to use in the application at hand.

Feature engineering. This step is the data treatment phase, where the collected data is transformed to conform to modeling norms.

Select and train model, evaluate model, tune model. This is the modeling stage, where the forecasting algorithms are chosen and the models are trained using the prepared data. The trained models are then evaluated, supported by evaluation metrics. In this phase, the model is subject to change through various iterations of the process. Changes might include the algorithm, its hyperparameters, the features used and the implementation itself.

Deployment. Once they have been evaluated positively and meet the required goals, the models can be deployed and used to make new predictions.

1.4 Document Structure

This document is segmented into six chapters.

In the first and current chapter, named *Introduction*, the motivations, goals and methodology followed are presented.

Chapter 2 is called *Related Work*. In this chapter, the literature review of the domain of tourism forecasting is presented, laying the foundation for the research undertaken in this work.

Chapter 3 is called *Methodology*. It presents the theoretical basis and reasoning behind the modeling done in the following chapter.

Chapter 4, named *Modeling*, is the main chapter of this document. The work done on model building is shown. It begins with data collection, followed by data cleaning. Then the baseline models are developed and evaluated, as well as the deep learning models. Finally, the results are discussed and a practical application is shown.

A web-application was developed in order to deploy results from the developed models. Its implementation is shown in chapter 5, called *Web-app Deployment*.

Finally, chapter 6 shows the conclusions from this research, as well as future work suggestions that arose from this work.

2

Related Work

Tourism Demand Forecasting (TDF) has been accurately predicting tourist flows and helping tourism companies and businesses alike for decades [5]. Papers as soon as 1973 were published with the first travel demand models and a review of the existing methods and mathematical tools [6]. However, it was in 1995 that one of the most influential papers in the field was published, by Witt and Witt [7], in which they emphasize the perishability of tourism demand forecasting and, therefore, the importance of the accuracy of the predictions. In their studies, they found out that no single method outperformed others consistently across different scenarios, but there were a few select that often worked best.

In this chapter, the literature review on the base of this research is shown. The scientific articles mentioned in this chapter were collected through keyword search in sources such as Scopus, ScienceDirect and Google Scholar. The keywords included expressions like “tourism”, “forecasting models”, “time-series”, “econometric” and “artificial intelligence”. These keywords were progressively adapted as knowledge was acquired from the reviewed work. In order to review the most recent literature, the focus was on articles published no earlier than 2019, with a few exceptions depending on relevance. Additionally, relevant citations in reviewed articles were also taken into account.

2.1 Types of Forecasting Models

Throughout the years, countless articles on TDF have been released. They have evolved the existing strategies and built new methods that could outperform previous ones in different and more complex environments. Today, they can be separated into three main buckets [8], although a significant portion of their use is in a hybrid approach:

- Econometric-based models;
- Time-Series-based models;
- Artificial Intelligence-based models.

Although this division between models is common, econometric and time-series based models refer to the data used, while artificial intelligence based models refer to the algorithms implemented. Because econometric data is usually in time-series form, both econometric and time-series models can use the same algorithms. Therefore, in this work, this type of models are called statistical models.

Table 2.1 displays a summary of the main scientific articles on TDF in this chapter.

The following sections will discuss the different types of forecasting models and their use in recent times.

2.2 Statistical models

A time-series is a series of points laid in historical order, usually spaced by equal intervals of time. The study of time-series is called time-series analysis, and it encompasses methods that analyse the time-series and extract information, in the form of statistics or other characteristics of interest.

Econometric models are usually comprised of time-series data, and attempt to represent the relationship between a particular econometric variable of interest (Dependent Variable) and other explanatory variables. Normally, this relationship is represented mathematically, and can, for example, be linear, polynomial or logarithmic.

The following sub-sections shed some light on the different approaches and models that use time-series data of econometric nature, followed by displaying examples of their use. The first four models, shown in section 2.2.1, are more traditionally called econometric models, while the last two are more often associated with time-series models.

2.2.1 Econometric Models

Time-Varying Parameter (TVP) has been used with great success in tourism demand forecasting. It is a model that differs from traditional econometrics techniques, as it overcomes the unrealistic assumption of constant coefficients [9] by taking into account the possibility of parameter changes throughout time. It is particularly useful for short-run forecasting [10].

The Vector Auto-Regressive (VAR) approach, unlike most traditional tourism demand models, is most adequate for situations in which the explanatory variables of the model are not exogenous. To deal with this predicament, VAR is represented as a system of equations, in which all variables are considered endogenous [9].

Error Correction Model (ECM) is an econometric forecasting model that estimates both the long and short term effects of the interaction between two variables. It is called error correction for its ability to track short term variations compared to the long term trend, and correcting the error.

Autoregressive Distributed Lag Model (ADLM) is a multivariate model that allows for past realizations of the dependent variable and current and past realizations of the

Index	Authors	Title of Article	Methods
1	Wong (2007)	Tourism forecasting: to combine or not to combine?	Main: VAR, ECM, ADLM, ARIMA.
2	Gunter (2015)	Forecasting international city tourism demand for Paris: Accuracy of uni-and multivariate models employing monthly data	Main: ADLM, VAR, TVP, ARMA, ETS, Naïve-1
3	Ismail (2020)	Forecasting the number of Arab and foreign tourists in Egypt using ARIMA models	ARIMA with BIC criterion
4	Ma (2021)	Tourism Demand Forecasting Based on Grey Model and BP Neural Network	Main: BPNN with Grey prediction model. Comparison: RBF NN, SVM NN and BPNN
5	Hu et al. (2019)	Forecasting tourism demand by incorporating neural networks into Grey–Markov models	Main: Grey Model Neural Network. Comparison: Other Grey prediction models
6	Huang et al. (2022)	Tourist hot spots prediction model based on optimized neural network algorithm	Main: RBF with Particle Swarm Optimization. Comparison: BPNN, Clustering analysis
7	Shi (2020)	Tourism culture and demand forecasting based on BP neural network mining algorithms	Main: BPNN. Comparison: SVM and ARIMA
8	Wang (2022)	Tourism Demand Forecast Based on Adaptive Neural Network Technology in Business Intelligence	Main: Modified Neural Network
9	Kulshrestha et al. (2020)	Bayesian BILSTM approach for tourism demand forecasting	Main: BiLSTM with Bayesian optimization. Comparison: LSTM, SVR, RBF NN, ADLM
10	Hsieh (2021)	Tourism demand forecasting based on an LSTM network and its variants	Main: LSTM, BiLSTM, GRU
11	Nawaz et al. (2021)	Machine Learning based Forecasting Systems for Worldwide International Tourists Arrival	Main: SVR. Comparison: RFR
12	Kirkos (2022)	Airbnb listings' performance: determinants and predictive models	Main: Random Forest. Comparison: C4.5 Decision Tree, Logistic Regression, Multilayer Perceptron Neural Network, SVM

Table 2.1: List of articles on forecasting methods and their use.

explanatory variables. This model can always be rewritten as an ECM model, which makes the two models two ways of formulating the same model [11].

Wong (2007) [12] compared the results of VAR, ECM, ADLM and a time-series model, Auto-Regressive Integrated Moving Average (ARIMA) (talked about in section 2.2.2), in predicting inbound tourism to Hong Kong from the top 10 tourism generating countries/regions. The results varied with the origin of the tourists, but they concluded that combining models was the safest and often best accuracy option.

Gunter (2015) [11] pitted ADLM, VAR, TVP, Auto-Regressive Moving Average (ARMA), ETS and Naïve-1 to compare their predictive accuracy, using RMSE and MAE as the error metrics. The Naïve-1 method was significantly outperformed across nearly all source markets and forecast horizons.

2.2.2 ARIMA

An Auto-Regressive Integrated Moving Average (ARIMA) model is a forecasting model popularized by Box and Jenkins [13] that is used in time-series analysis and forecasting. It comes from the Auto-Regressive Moving Average (ARMA) model, which lacks the ability to make regressions on non-stationary time-series, that is, time-series which don't have constant mean throughout the time axis. ARIMA can achieve this by making transformations to the time-series in order to turn it stationary.

It is a model that has had success throughout the years. It is a combination of three models: Auto-Regressive, Integrated and Moving Average. Therefore, the ARIMA model is often represented as arima(p,d,q), in which p, d and q are the hyperparameters of the model that control each of the different components of ARIMA (p is the number of auto-regressive terms, d is the number of non-seasonal differences to achieve stationarity and q is the order of the moving-average model). Although useful in its base implementation, the ARIMA model has evolved to tackle more time-series problems, having evolved into SARIMA and SARIMAX. The "S" stands for seasonal, as the regular ARIMA model has trouble having good results with seasonal data, and the "X" for eXogenous variables, meaning that this ARIMA model can make use of other variables to improve forecasting accuracy.

Ismail (2020) [14] uses the ARIMA model to predict annual number of tourists in Egypt in the years 2018-2022. Historical data of annual tourists in Egypt from 1981 to 2017 was used. Two models were trained, one for Arab arrivals and one for foreign arrivals. They use the BIC criterion to ascertain which of the tested models would be the most suited. BIC is a heuristic method that takes a model as input and outputs a real number. The lower the number, the better pick the model is. The results of BIC showed ARIMA was the most fitting model for both the Arab arrivals and foreign arrivals. After training and testing, the r-square result for Arab and foreign arrivals was 84.7% and 83.1% respectively.

2.2.3 Grey Model

Grey Prediction Model comes from the research field of Grey Systems. Grey System theory started in 1982 and is characterised by systems that lack information [15]. Examples of Grey systems are the human body, agriculture or the economy. Grey can, in this sense, mean poor, incomplete or uncertain, the latter synonym being the most befitting for the case of tourism.

Grey forecasting is, among others, one of the areas that Grey systems encompass. It is the area that aims to "unify the field and bridge the gap between grey process theory and practice". The commonly used model in grey forecasting is the grey differential equation represented by GM(1, 1).

Much of nowadays' use of Grey models is in cooperation with other forecasting models, namely, and most commonly, neural networks. Ma (2021) [16] used a Back Propagation Neural Network (BPNN) with a Grey Prediction model to predict travel time and number of tourists in domestic tourism in China. Their objective was to explore a new prediction method for tourism's complex environment, and to achieve better results for decision making. Their results for the proposed algorithm achieved a very high r-squared of 0.998 and a MSE of 0.0039. The second best r-squared was 0.979, by a Support Vector Machines (SVM) Neural Network, which shows the proposed algorithm's superiority in this case.

Hu (2019) [17] compared ten different Grey models' prediction capabilities by using them to predict foreign tourists in Taiwan and in China. The main algorithm that was being tested against all others was the NNGM(1, 1), a hybrid between a Grey Model and a Neural Network. By using the Friedman and the Bonferroni-Dunn tests, the paper concludes that the results obtained from NNGM(1, 1) were either similar – with little difference of p factor in the tests – or superior to all the other pitted Grey Models.

2.3 Artificial Intelligence-based Models

This branch is the most recent among the ones previously referred to, but perhaps the most exciting. It's no surprise that machine learning models have gained so much traction in modern times, since the computing power of our machines has been increasing exponentially for as long as modern computers have been around.

The first attempt to mathematically explain the human thought process and decision-making was in 1943, a time in which was known that a Turing Machine was theoretically capable of supporting artificial neural networks, but there were no machines that were powerful enough to do so efficiently [18]. But as machine learning evolved, so did our computers' ability to withstand the computations necessary to make it a viable tool.

2.3.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) were one of the first machine learning methods to be developed [18], and the most commonly used for forecasting, normally in collaboration with other forecasting models, either econometric, time-series or even other machine learning models.

ANNs have various real-world problem-solving applications, namely in business, education and economics. This is, in part, because of their great ability to identify trends in data and patterns, which makes them a very viable choice for forecasting [19].

Some of the most used ANNs for forecasting include BPNN, Radial Basis Function (RBF) networks and Deep Neural Network (DNN). Their results can be better than econometric or time-series models on certain circumstances.

Huang (2022) [20] proposed an RBF network to predict tourist volume on popular touristic attractions, alongside a Particle Swarm Optimization algorithm to optimize the ANN's parameters. Their purpose was to prove this method's superiority against the traditional forecasting methods, such as Econometric models, Time-series models or Grey prediction models. They compared their method to a BPNN and a Clustering analysis algorithm and found great success in terms of not only the accuracy of the predictions, but also time to train and to predict. The improvement compared to the BPNN was consistent if slight in terms of accuracy, but trained in almost half the time. On the other hand, the difference in performance was very significant when compared to the clustering algorithm, with an average accuracy improvement of 8.5% and nearly 1.5 times faster.

Shi (2020) [21] demonstrated another use of ANN that produced better results than more traditional methods, by using a BPNN to predict tourism demand in Yangjiang, a rural area. It was compared to another machine learning model, SVM and to ARIMA. The models used economic and SVM demographic variables, such as per capita GDP and population to predict the number of inbound tourists for Yangjiang. The results indicated that the BPNN produced the best predictions for all but one of the 20 countries in evaluation. The article provides an explanation to these results, stating that data with strong seasonal patterns and high-level fluctuations requires some pre-processing, which BPNN perform well, making them suitable for dealing with non-linear data.

Wang (2022) [22] aimed to improve the effect of TDF for China's tourism industry and the tourists' experience. To achieve it, they developed a new tourism demand forecasting system, integrated with tools to "search for, roam, navigate, collect, share and pay for scenic spots", which uses Adaptive Neural Networks to make its predictions. Furthermore, they modified and improved the Adaptive Neural Network algorithm in order to handle multiple sources of spacio-temporal data.

One particular case of interest in ANNs are deep learning models like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).

Kulshrestha (2020) [23] used Bayesian BiLSTM to predict tourism demand in Singapore. The prediction model is a type of Recurrent Neural Network (RNN) and is further improved by the use of Bayesian optimization to find the optimal values for the Bidirectional Long shot-term memory network's (BiLSTM) hyperparameters. The model found great accuracy when compared to other models like LSTM, Support Vector Regression (SVR) and RBF network.

Hsieh (2021) [24] used LSTM, bilateral LSTM and GRU in order to forecast Taiwan's tourism demand. It concluded that these models were adequate to predict arrivals after shock events, like the Covid-19 pandemic.

2.3.2 Support Vector Machines (SVM)

SVM are machine learning models that make predictions based on the Vapnik-Chervonenkis theory. They were first named Support-Vector Networks and were said to solve two-group classification problems [25].

This machine learning model achieves linear classification on its default intended use, but it can also perform non-linear classification using a "kernel trick", mapping the non-linear trends in input space to linear trends in a higher-dimensional feature space [26].

Nawaz (2021) [27] used SVR, an adaptation of SVM for the regression type data, and Random Forest Regression to make tourism forecasting models that predicted international tourist arrivals. The data used was the tourists arrival on the years prior, which produced better results for the SVR with a linear kernel, with an R-square of 0.994, against an R-square of 0.847 obtained by Random Forest Regression. The article also shows how the method did fail predicting the 2020 tourists arrival due to the Covid-19 pandemic, although a new prediction from 2020 onward is presented.

2.3.3 Random Forest

Random Forest Regression is a machine learning method developed in 1995 [28], and it was characterized as an intuitively appealing idea, training in a straight-forward and extremely fast manner. It works by training multiple decision trees at random and choosing and combining the most recurring outputs. Decision trees often overfit the training data-sets, hence why Random Forest combines the results of many, as the decision trees cancel out each other's faults. It uses the idea of "wisdom of the crowds".

Kirkos (2022) [29] created predictive models for Airbnb performance in Greece. This allows for predicting how many tourists would be staying in the country. To achieve this, 5 models were trained, including C4.5 Decision Tree, Logistic Regression, Multilayer Perceptron Neural Network, SVM and Random Forest. The independent variables used came from Airbnb statistics, and the target variables for prediction were Occupancy rate, number of bookings and revenue. Random Forest was consistently the model with the best results for all used metrics and highest accuracy of prediction with an average of 82.28%.

3

Methodology

After surveying the recent literature on tourism forecasting, shown in the previous chapter, the present chapter shines light on the architecture of this project, clarifying the theoretical foundations and the rationale behind the choices to be made. Furthermore, it serves as a guide to the next chapter 4, entitled *Modeling*.

3.1 Data Collection

The tourism industry is inherently complex, influenced by a myriad of factors that contribute to its dynamics. As such, the selection and curation of variables for analysis plays a crucial role in building effective forecasting models. When collecting data, two main factors have to be considered in order to ensure their fit for modeling use: The types of data and the sources. The two following subsections dive into these topics.

3.1.1 Data Types

There is a plethora of potential variables for use in forecasting. Econometric, environmental, financial, touristic, demographic, digital, to name a few. The choice of variables in each case is of paramount importance in determining the accuracy and reliability of forecasting models. Variables that are relevant in the tourism domain hold the potential to enhance predictions. In tourism forecasting, touristic and econometric variables are most fitting, as the economic state of the world dictates the volume and flow of tourism. In this work, the econometric variables used include GDP, inflation, and exchange rate, while the touristic variables are arrivals by airplane and at accommodations.

There is a clear distinction exists between target and feature variables in forecasting. Target variables represent the data that the model tries to regress and predict, while feature variables are used to find correlations with the target and improve on the predictions.

Touristic variables, such as arrivals and hotel stays, have the potential to be used as both target and feature variables. As feature variables, they can be used to further enhance the predictive capabilities of the models. Econometric variables, on the other hand, in the context of tourism forecasting, will solely serve as exogenous features to improve the models – so they will not be target variables.

3.1.2 Data Sources

After studying what data should be used to forecast tourism demand, the next task is finding adequate and reliable data-sets. All of this work's data sources are exclusively publicly available online. After some exploratory analysis, the main choices to retrieve touristic and econometric European data for this work are Eurocontrol[1], Eurostat[2] and *Yahoo!* Finance[3]. Platforms like Eurostat and World Bank Open Data expose API's for public use, which allows for automatic data collection. This is an advantage in the case of an existing need to easily collect new and more updated data.

3.2 Data Understanding and Preparation

Collected data very rarely comes in the right formatting and conditions to directly use in modeling. It may come with gaps, wrong or broken values, irrelevant columns or with custom formats, like the date. Hence the need to apply pre-processing to the data, in order to allow the models to use it. However, before employing pre-processing techniques, understanding the data and knowing its shortcomings is advisable. This is achieved through some exploratory data analysis. Different visualizations of the data are needed to highlight the needed treatment. Another common type of processing applied is outlier handling, in which values that seem to stand out from the norm are interpreted as mistakes and removed or normalized. However, in this work, outliers may not be removed or normalized, as tourism is a volatile market, and can be explained by sudden peaks or falls in certain exogenous variables. For example, in the case of (internet) online data, peaks in interest of certain areas can generate profound changes for tourism. Nonetheless, analysis is done on a case-by-case basis.

3.3 Models

As for the models to be developed, first baseline models will be constructed to evaluate the forecastability of the data, and also to have a point of comparison to the latter, more advanced models. Then comes choosing, fitting and evaluating the Deep Learning models.

[1] https://www.eurocontrol.int/our-data
[2] https://ec.europa.eu/eurostat
[3] https://finance.yahoo.com/

3.3.1 Evaluation Metrics

The evaluation metrics are the techniques used to evaluate and compare the different trained forecasting models. Their use is crucial for forecasting-models training, as they show how well they fit the training data and how accurate the predicted values are expected to be. For the following expressions, let X_i be the predicted i^{th} value, and Y_i the actual i^{th} value. The mean of the actual values will be defined as

$$\overline{Y} = \frac{1}{m}\sum_{i=1}^{m} Y_i \ .$$

There is a large amount of commonly used evaluation metrics. The ones used in this work are: Root Mean Square Error (RMSE), R-Squared (R2) and Mean Absolute Percentage Error (MAPE). More details about each one are presented in the following sub-sections.

3.3.1.1 RMSE

The RMSE is one of the most common evaluation metrics when it comes to assessing a forecasting model's performance. It is obtained by applying the square root to the Mean Square Error (MSE) ($\sqrt{MSE}$). Its values have the same unit of measure as the target variable, although normalization makes the values comparable between models, and they range from 0 to $+\infty$, the former being a perfect fit. Due to the square root, this metric punishes single large errors in predictions, which makes it a good forecasting model evaluator. Its expression is as follows:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2}$$

3.3.1.2 R-Squared

R2 is another very commonly used metric. Described as the coefficient of determination, it quantifies how much the dependent variable is determined by the independent variables, in terms of proportion of variance. It is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(X_i - Y_i)^2}{\sum_{i=1}^{m}(\overline{Y} - Y_i)^2}$$

and, therefore, has values between $-\infty$ and 1, the latter being the best fit possible for the evaluated model. It is often said that this metric's range is between 0 and 1, as values below zero mean the predictions are worse than using the average value of the series as the prediction.

3.3.1.3 MAPE

The MAPE is, in nearly all aspects, identical to Mean Absolute Error (MAE), which measures the average difference between the predicted and actual values. However, its results are given in percentage form, allowing the analysis of relative variations. This metric only works with positive data by definition, which will not pose a limitation to its use in this work. It can take values from 0 to $+\infty$, 0 being perfect fit, and its expression is

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}|\frac{X_i - Y_i}{Y_i}| \ .$$

3.3.2 Bayesian Optimization

Most algorithms used in forecasting models have parameters that need to be set in order to dictate the way the algorithm functions. They are called hyperparameters. When training a model, hyperparameter tuning will significantly improve the model's ability to adapt correctly to the input data, outputting better results. The tuning can be carried out through trial and error, by iteratively changing the hyperparameter values and testing the model until it reaches the desired outcome. However, there are hyperparameter optimization algorithms that can be used to automatically and more accurately ascertain the best hyperparameter values. One popular algorithm is Bayesian optimization.

Let's assume a model training algorithm can be represented as a function of x, $f(x)$, in which x represents the hyperparameter(s) value(s) and $f(x)$ is the value of the evaluation metric for the trained model. Bayesian optimization uses a Gaussian process that determines the values of x with highest uncertainty and potential for a better $f(x)$. It begins by calculating $f(x)$ of an arbitrary number of initial x values, and then uses the Gaussian process to find potential improvements. This is, therefore, a faster way of converging to the best hyperparameters than random search [30].

One use of this algorithm is mentioned in chapter 2, in the Artificial Neural Network (ANN) section 2.3.1, where Kulshrestha (2020) [23] used Bayesian optimization to optimize the hyperparameters of Bidirectional Long shot-term memory network's (BiLSTM).

In this work, the bayesian-optimization python library was the used implementation. It implements a *BayesianOptimization* instance that takes as parameters the number of initial points, the number of iterations and a "black box function". The "black box function" must take as keyword parameters the hyperparameters of the model in float numbers, train a model and output an evaluation metric. The algorithm will then attempt to maximize the output of the "black box function" with every iteration. If minimizing the output is the desired functionality, using the negative of the metric is a viable option.

3.3.3 Baseline Models

As forementioned, forecasting can use numerous variables. However, not all data is suitable for use in forecasting models, as there are factors that dictate the usability of a variable for predictions. One way to quickly ascertain whether a desired variable is useful or not for modeling is through baseline models. They are models that can be quickly trained to get rough yet fast results. If the predictions are acceptable to a certain degree, with not much effort to begin with, there is higher confidence that the data is usable for good predictions with some more effort and complex algorithms. Once these models are in place, they can be used as a comparison (baseline) to the more advanced, higher effort models.

Usually, the best models for the effect are Naïve methods, statistical models (like moving average, exponential smoothing), or models that have stood the test of time, such as Auto-Regressive Integrated Moving Average (ARIMA). There are models that have simple implementations, normally through pre-programmed libraries. The chosen models for this work are: Naïve Drift, Naïve Moving Average, Naïve Seasonal, Exponential Smoothing and Seasonal Auto-Regressive Integrated Moving Average (SARIMA). In this context, the python library Darts was used to develop the first four baseline models, while SARIMA was implemented with statsmodels.

3.3.4 Deep Learning Models

The focus of this work is on the deep learning models. They're machine learning models that use artificial neural networks to make their predictions. Their choice for this work is explained by their ability to use large sets of data and features. Moreover, as was mentioned in chapter 2, they're relatively recent forecasting techniques, with great success in forecasting time-series. However, not all deep learning models are relevant for this type of application. In this work, two deep learning models were used: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).

3.3.4.1 LSTM

LSTM is a deep learning artificial intelligence model. It evolved from the Recurrent Neural Network (RNN) to overcome the exploding/vanishing gradient problem [31]. This problem appears when stacking a large number of RNNs, in which case, during back-propagation, the gradient is multiplied continuously by a number either smaller or bigger than one. On the former case, the gradient will exponentially decrease to a near zero value, virtually preventing the weights of the network to change. Whereas on the latter, the gradient will exponentially increase to a number too big, resulting in a much too rapid change of the weight values, not allowing them to converge to an optimal state.

The LSTM implementation is a partial fix to this problem, as it learns patterns on the data and can discern when to remember and when to forget previous information, thus not allowing the gradient to explode or vanish as the model gets deeper. It achieves this with the implementation of the cell state. Figure 3.1 shows a diagram of an LSTM cell. Each activation function has its own set of weights and biases. On a high level, the cell is composed by the cell state, represented by C_t, and three gates: The forget gate, the input gate and the output gate.

The cell state gets first updated by the forget gate, which takes in an input and the hidden state of the previous cell and runs it through a sigmoid activation function. The

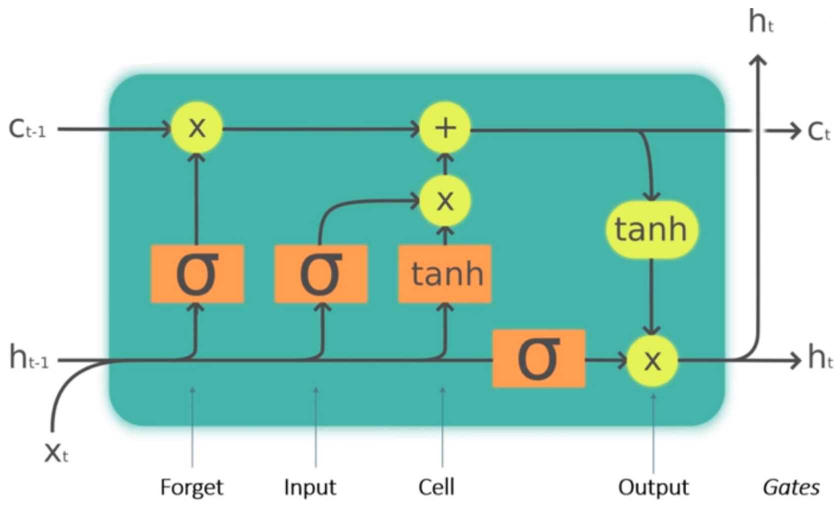


Figure 3.1: Diagram of LSTM cell.

Source: https://en.wikipedia.org/wiki/Long_short-term_memory#/media/File:LSTM_Cell.svg

activation function will output a number between 0 and 1, which will be multiplied with the cell state, effectively rendering how much influence it will carry to the next cell.

The input is parallelly fed to the input gate. Inside this gate, it goes through another sigmoid activation function and a Hyperbolic Tangent Function (tanh). The former will scale how much of the output of the latter will affect the state. Once the output of the tanh has been scaled, it is then added to the state.

Finally, the output gate calculates the output of the cell. The input is fed to this final gate, which is run through yet another sigmoid function. It will serve as a scaler. In turn, the state scale is run through a tanh function. The product of the two will be the output of the cell.

3.3.4.2 GRU

GRU, much like LSTM, is an evolution of the RNN with a gated architecture. It is quite a recent algorithm, as it was created in 2014 by Kyunghyun Cho et al [32]. According to Chung (2014) [33], neither of these algorithm categorically outperforms the other. GRU is, however, less complex and can run faster than LSTM [34].

This algorithm only contains two gates, as opposed to the three of LSTM. The two gates are the reset gate and update gate. Instead of a cell state, GRU makes use of a hidden state to transfer information, which means that, unlike LSTM, the cell does not output the state. In figure 3.2, a GRU cell is represented.

The reset gate decides how much past information to forget. It does this by running the input and last cell output through a sigmoid activation function. The outcome of the activation function will scale the last cell's output values that will later go into the tanh function.

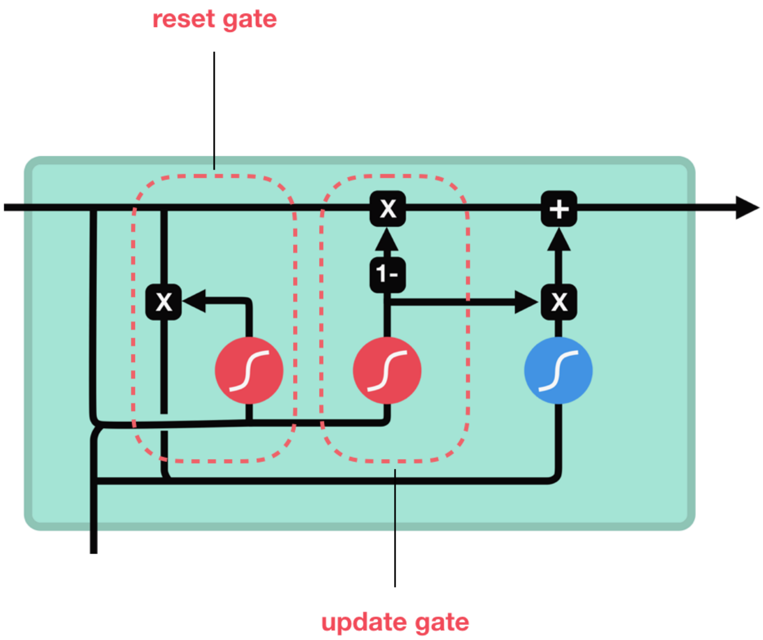


Figure 3.2: Diagram of GRU cell.

Source: https://miro.medium.com/v2/resize:fit:1400/1*S0rXIeO_VoUVOyrYHckUWg.gif

The update gate works much like the forget and input gates of LSTM combined, deciding which information to forget and which to add. It takes the input and output of the last cell and runs it through a sigmoid activation function. These values are then parallelly used in two ways: 1- The sign of these values is inverted and multiplied with the last cell output; 2- The values are multiplied with the aforementioned tanh's output. Finally, the two flows of values are summed to make the output of the cell.

4

Modeling

4.1 Introduction

The previous chapter has established the theoretical framework upon which this work relies on, explaining each of the steps undertaken and their reasoning. This chapter shifts the focus to the practical approach, where the whole modeling process takes place.

It starts with data collection, where the process of collecting data is presented. Following, the Data Understanding and Preparation section delves into the treatment of the collected data and, in the end shows, as outcome, the modeling-ready features. Next is the Modeling section, where all the models are trained and the results are shown. It starts with the baseline models, training and showing their results. They will serve as a point of comparison for the deep learning models, which are talked about in the Deep Learning Based Models subsection. This section explains which models will be developed, shows the model development and Bayesian optimization implementation and which features were used for each model. Then a discussion of the results obtained will be presented. Finally, there's a summary of the work done in this chapter.

4.2 Data Collection

This section delves into the sources chosen for data collection. The sources of the collected data are shown, as well as visualizations to highlight the characteristics of the data.

4.2.1 Eurostat

Eurostat is one of the largest data-sets available for European data. It contains monthly, quarterly, and yearly data of numerous econometric variables. The data of interest to be extracted and the respective Eurostat code and frequency are as follows:

- Arrivals at tourist accommodation establishments (tour_occ_arm) - monthly;
- GDP (NAMQ_10_GDP) - quarterly;
- Inflation (prc_hicp_manr) - monthly.

Column name	Source	Pivot Label	Description	Example
YEAR	*Network Manager*	*YEAR*	*Reference year*	2014
MONTH_NUM	*Network Manager*	*MONTH*	*Month (numeric)*	1
MONTH_MON	*Network Manager*	*MONTH_MON*	*Month (3-letter code)*	JAN
FLT_DATE	*Network Manager*	*DATE_FLT*	*Date of flight*	###
APT_ICAO	*Network Manager*	*APT_ICAO*	*ICAO 4-letter airport designator*	EDDM
APT_NAME	*PRU*	*APT_NAME*	*Airport name*	Munich
STATE_NAME	*PRU*	*STATE_NAME*	*Name of the country in which the airport is located*	Germany
FLT_DEP_1	*Network Manager*	*Departures - (NM)*	*Number of IFR departures*	278
FLT_ARR_1	*Network Manager*	*IFR arrivals - (NM)*	*Number of IFR arrivals*	241
FLT_TOT_1	*Network Manager*	*IFR flights (arr + dep) - (NM)*	*Number total IFR movements*	519
FLT_DEP_IFR_2	*Airport Operator*	*IFR departures - (APT)*	*Number of IFR departures*	278
FLT_ARR_IFR_2	*Airport Operator*	*IFR arrivals - (APT)*	*Number of IFR arrivals*	241
FLT_TOT_IFR_2	*Airport Operator*	*IFR flights (arr + dep) - (APT)*	*Number total IFR movements*	519

Figure 4.1: Metadata of the Eurocontrol arrivals data.

4.2.2 Eurocontrol

Eurocontrol possesses daily arrivals data that will be a major source for the daily forecasting models. There were 754972 entries collected, equating to a time-frame of 2016 to 2022 and containing most European countries and respective airports. Figure 4.1 shows the metadata relative to the Eurocontrol data-set.

4.2.3 Yahoo! Finance

Yahoo! Finance is a platform owned by *Yahoo!* that contains financial information, as well as data-sets. In this case, it is used for its daily currency exchange rate data-set. The downloaded data has a time range between '1999-01-05' and '2023-01-02'. It is composed of comparisons between the euro and multiple other currencies. These are: South Korean Won (EUR-KRW), Swiss Franc (EUR-CHF), Russian Ruble (EUR-RUB), Indian Rupee (EUR-INR), Brazilian Real (EUR-BRL), British Pound (EUR-GBP), Japanese Yen (EUR-JPY), Hong-Kong Dollar (EUR-HKD), Chinese Yuan (EUR-CNY), Canadian Dollar (EUR-CAD), Australian Dollar (EUR-AUD) and United States Dollar (EUR-USD). Each currency has daily data on the opening value (Open), the highest value (High), the lowest value (Low), the close value (Close) and the adjusted close value (AdjClose). This data contained a lot of missing values, which will be imputed in section 4.3.9.

4.2.4 Google Trends

The modern world has allowed for data creation through the internet at a rate beyond what was previously imaginable. Companies compete to have the most internet generated data to model into usable information for competitive advantage. This abundance of data has created a new source for modeling use, internet data. One example of internet data is search engine data. It is characterized by time-series of volume of searches per keyword. In this work, the search engine data is extracted through a program made with the python library pytrends. This library uses web scraping techniques to extract data from Google Trends. The data is pre-normalized to values in a range of 0 to 100 and its frequency depends on the window of time chosen. These are the different frequencies:

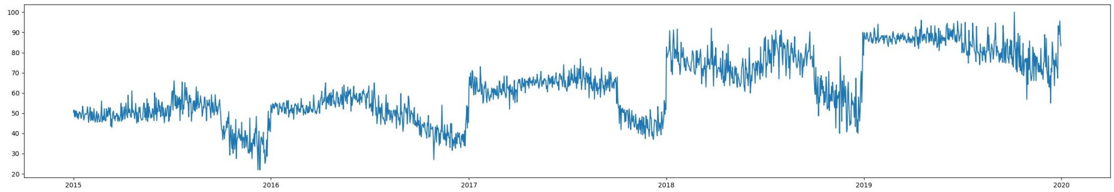


Figure 4.2: The output of the pytrends aided algorithm to obtain daily data from Google Trends.

- Last hour and Last 4 hours: every minute;
- Last 24 hours: every 8 minutes;
- Last seven days: hourly;
- Past 30 days and Past 90 days: daily;
- Past 12 months and Last five years: weekly;
- 2004 - present: monthly.

Having previously collected daily data from other sources, it would be useful to extract daily data from Google Trends to use in improving accuracy of the daily models. Unfortunately, daily data can only be extracted at 90-day intervals. Given that the data comes pre-normalized, concatenating different downloads is not viable, as they will have differing scales. The solution for this problem is as follows:

1. Download weekly data from a desired 5-year window;
2. Download daily data for every 90 days inside the 5-year window;
3. Scale every 90-day period of daily data to conform to the adequate time-frame in the five yearlong weekly data, effectively transforming all 90-day periods to have the same scale for the 5-year period;
4. Concatenate the now equally scaled 90-day daily data.

After these transformations, the result will be a daily time-series for the 5-year period. As an example of the results, the time-series generated by the algorithm for the keyword 'Portugal flights' is shown on figure 4.2. For comparison, the weekly results directly from the Google Trends website [1] are displayed in figure 4.3, for the same time window and keyword.

As is evident when visually comparing both graphs, the scaled daily data is faithful to the original weekly data.

[1] https://trends.google.com/trends/

Figure 4.3: The Google Trends weekly data for the same time frame as figure 4.2.

The keywords chosen were based on the work of Wen (2019) [35] and adapted to suit the needs for the case in hand. Google Trends only saves search volume for a specific keyword if the number of searches reaches a certain threshold. Therefore, some of the data is unreliable if the searches often do not reach the threshold, as there will be clear gaps in the time-series. For this reason, some adaptation of which keywords to use is needed.

4.2.5 Collected Data Overview

In total, the collected data can be boiled down to the following:

- Eurocontrol:
  - Daily arrivals in European countries between 2016 and 2022 from Eurocontrol.
- Eurostat:
  - Monthly arrivals at accommodation establishments in European countries between 1995 and 2022.
  - Quarterly GDP of European countries data between 1995 and 2022.
  - Monthly inflation of European countries data between 1997 and 2022.
- *Yahoo!* Finance:
  - Daily currency exchange rates data between 1999 and 2023. The exchange rates are: EUR-KRW, EUR-CHF, EUR-RUB, EUR-INR, EUR-BRL, EUR-GBP, EUR-JPY, EUR-HKD, EUR-CNY, EUR-CAD, EUR-AUD and EUR-USD.
- Daily and monthly Google Trends data, collected with the pytrends python library:

- The keywords were composed of the five countries Portugal, Spain, Italy, France and Greece concatenated with: accommodation, airport, attractions, flights, food, hotel booking, hotels, map, shopping, shopping map, snack, specialty, subway, tickets, tourist attractions, travel, travel guide, travel map, weather.
- The daily data ranges from the beginning of 2016 to the end of 2019.
- The monthly data ranges from the beginning of 2004 to the end of 2019.

4.3 Data Understanding and Preparation

Once data has been collected, then it will be transformed to conform to the standards of forecasting model development. In the following sub-sections, shaping and reformatting columns of columns will be performed, as well as imputing data. The last sub-section shows the clean data, ready to be used in the modeling stage.

4.3.1 Common Characteristics

In the following sub-sections, each variable will be subject to unique treatment. However, the outcome will have some common characteristics for all variables.

Data for daily models will have a 'date' column, which will adhere to the 'YYYY-MM-DD' pattern and have a range of '2016-01-01' to '2019-12-31'.

Similarly, data for monthly models will have a 'date' column, in turn with the pattern 'YYYY-MM', and will have a range of '2004-01' to '2019-12'.

Despite this work being carried out in 2022/2023, the date limit chosen for the data is the end of 2019. This is due to the Covid-19 pandemic that began at the end of 2019, which rendered the use of the following years' data unusable, due to perturbations to the arrivals data that are inexplicable by readily available external variables.

4.3.2 Eurocontrol Daily Arrivals

The Eurocontrol daily arrivals data came with columns: ['YEAR', 'MONTH_NUM', 'MONTH_MON', 'FLT_DATE', 'APT_ICAO', 'APT_NAME', 'STATE_NAME', 'FLT_DEP_1', 'FLT_ARR_1', 'FLT_TOT_1', 'FLT_DEP_IFR_2', 'FLT_ARR_IFR_2', 'FLT_TOT_IFR_2', 'Pivot Label']. An example row can be seen in figure 4.4. It's evident that some of these columns have redundant information. Some others have information on departures, which are irrelevant for this project. Therefore, these columns were dropped, leaving only ['FLT_DATE', 'STATE_NAME', 'FLT_ARR_1', 'Pivot Label']. 'FLT_ARR_1' was chosen as the arrivals column instead of 'FLT_ARR_IFR2' for they both served the wanted purpose and it was the least sparse.

Next, to get arrivals by country, the arrivals at each airport for the desired countries was grouped by the same date and summed.

YEAR	MONTH_NUM	MONTH_MON	FLT_DATE	APT_ICAO	APT_NAME	STATE_NAME	FLT_DEP_1	FLT_ARR_1	FLT_TOT_1	FLT_DEP_IFR_2	FLT_ARR_IFR_2	FLT_TOT_IFR_2	Pivot Label
2016	1	JAN	03-01-2016	GCTS	Tenerife Sur - Re...	Spain	129	128	257	129	129	258	Tenerife Sur - Re...

Figure 4.4: Example row of raw Eurocontrol data.

After cleaning, five dataframes were created with columns 'date' and 'arrivals' for the countries Portugal, Spain, Italy, France, Greece. As an example, the chart generated with the Portuguese dataframe is visible in figure 4.5.

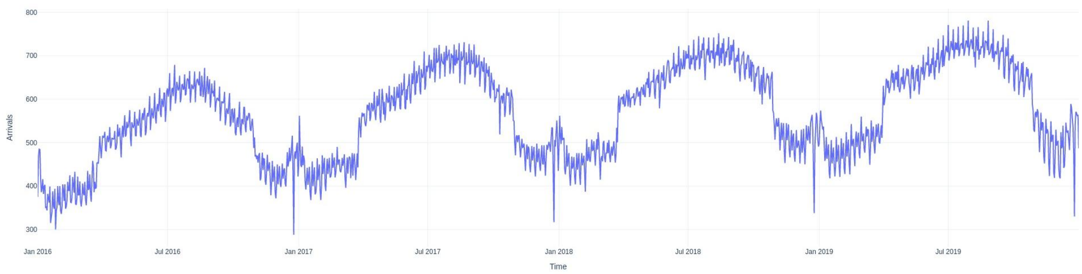


Figure 4.5: Daily arrivals chart for Portugal, from '2016-01-01' to '2019-12-31'.

4.3.3 Eurostat Date and Frequency Handling

In all the downloaded Eurostat data, the date is spread across multiple columns, evidenced in figure 4.6. However, the best way to deal with date in modeling is in a single column, . Therefore, after asserting all the date columns had values of type float, as some inexplicably and inconsistently were not, all the Eurostat data is reformatted accordingly using the pyspark library. This transformation generates some repeated column values, as shown in figure 4.7. However, this does not pose an issue, as only one variation of the values – for example, the country or the unit of the values – is normally used at any given time. The result after this reformatting is a dataframe with one single 'date' column and a 'values' column.

At times, data from Eurostat comes with multiple frequencies in the same data-set, as can be seen in figure 4.7. When this occurs, the lower frequencies are deleted, and only the highest is kept (e.g. if a data-set comes with yearly, quarterly and monthly data, only monthly data will remain).

4.3.4 Eurostat Monthly Arrivals

After the processing mentioned in section 4.3.3, the rows are filtered along the descriptive columns ['c_resid', 'unit', 'nace_r2']. 'c_resid' is filtered to 'FOR', which means "only foreign arrivals", 'unit' is filtered to 'NR', meaning "number of arrivals", and 'nace_r2' is filtered to "I551-I553", representing "arrivals at hotels, holiday and other short-stay

	A	B	C	D	E	F	G	H	I
1		c_resid	unit	nace_r2	geo	2022M11	2022M10	2022M09	2022M08
2	0	DOM	NR	I551	AL		47467.0	62928.0	103134.0
3	1	DOM	NR	I551	AT	732743.0	976058.0	1078662.0	1194552.0
4	2	DOM	NR	I551	BE	384576.0	455496.0	454534.0	512063.0
5	3	DOM	NR	I551	BG	269748.0	311338.0	378584.0	572338.0
6	4	DOM	NR	I551	CH	640674.0	971417.0	1095376.0	1130648.0
7	5	DOM	NR	I551	CY		48008.0	49095.0	132076.0
8	6	DOM	NR	I551	CZ		821705.0	919380.0	1071302.0
9	7	DOM	NR	I551	DE	8148547.0	10071190.0	10740307.0	10604703.0
10	8	DOM	NR	I551	DK	333181.0	371405.0	409710.0	471805.0
11	9	DOM	NR	I551	EA	25133419.0	31459886.0	33845619.0	39269548.0
12	10	DOM	NR	I551	EA19	25133419.0	31459886.0	33845619.0	39269548.0
13	11	DOM	NR	I551	EA20	25231568.0	31600962.0	33969181.0	39397918.0

Figure 4.6: Crop of a spreadsheet with Eurostat data to highlight the date columns. The four rightmost columns are a sample of the numerous date specific columns with the corresponding value.

```
+---+------+--------+---------------+-------+------+
|_c0|  unit|tra_meas|        airp_pr|   date| value|
+---+------+--------+---------------+-------+------+
|  0|FLIGHT| CAF_PAS|PT_LPAZ_DO_MDPP|   1998|  61.0|
|  0|FLIGHT| CAF_PAS|PT_LPAZ_DO_MDPP|   1997| 112.0|
|  0|FLIGHT| CAF_PAS|PT_LPAZ_DO_MDPP|   1996| 103.0|
|  1|FLIGHT| CAF_PAS|PT_LPAZ_ES_LEMD|   1998|  94.0|
|  1|FLIGHT| CAF_PAS|PT_LPAZ_ES_LEMD|   1997| 176.0|
|  1|FLIGHT| CAF_PAS|PT_LPAZ_ES_LEMD|   1996| 116.0|
|  2|FLIGHT| CAF_PAS|PT_LPAZ_PT_LPPD|   1999|1038.0|
|  2|FLIGHT| CAF_PAS|PT_LPAZ_PT_LPPD|   1998| 989.0|
|  2|FLIGHT| CAF_PAS|PT_LPAZ_PT_LPPD|   1997| 951.0|
|  2|FLIGHT| CAF_PAS|PT_LPAZ_PT_LPPD|   1996|   8.0|
|  3|FLIGHT| CAF_PAS|PT_LPFR_AT_LOWW| 2021Q4|  27.0|
|  3|FLIGHT| CAF_PAS|PT_LPFR_AT_LOWW| 2021Q3| 127.0|
|  3|FLIGHT| CAF_PAS|PT_LPFR_AT_LOWW| 2021Q2|  33.0|
|  3|FLIGHT| CAF_PAS|PT_LPFR_AT_LOWW| 2021Q1|   4.0|
|  3|FLIGHT| CAF_PAS|PT_LPFR_AT_LOWW|2021M12|   1.0|
|  3|FLIGHT| CAF_PAS|PT_LPFR_AT_LOWW|2021M10|  26.0|
|  3|FLIGHT| CAF_PAS|PT_LPFR_AT_LOWW|2021M09|  33.0|
|  3|FLIGHT| CAF_PAS|PT_LPFR_AT_LOWW|2021M08|  46.0|
|  3|FLIGHT| CAF_PAS|PT_LPFR_AT_LOWW|2021M07|  48.0|
|  3|FLIGHT| CAF_PAS|PT_LPFR_AT_LOWW|2021M06|  33.0|
```

Figure 4.7: Crop of a dataframe with Eurostat data to highlight the new date column.

accommodation, camping grounds, recreational vehicle parks and trailer parks", which is the widest option.

Following this, all columns are dropped except for 'date' and 'value', the latter being renamed to 'arrivals'.

Five dataframes are created for each of the five countries Portugal, Spain, Italy, France and Greece, with time ranges from '2004-01' to '2019-12' for all countries except France, which gets a range of '2011-01' to '2019-12', due to large gaps in the earlier dates. When later discussing results, this fact will be noted.

As an example, figure 4.8 shows the chart of the Portuguese data.

4.3.5 Eurostat Monthly Inflation

After the processing talked about in section 4.3.3, all columns except 'date' and 'values' are dropped, the latter being renamed to 'inflation'. Finally The data is separated into five dataframes for each of the countries Portugal, Spain, Italy, France and Greece.

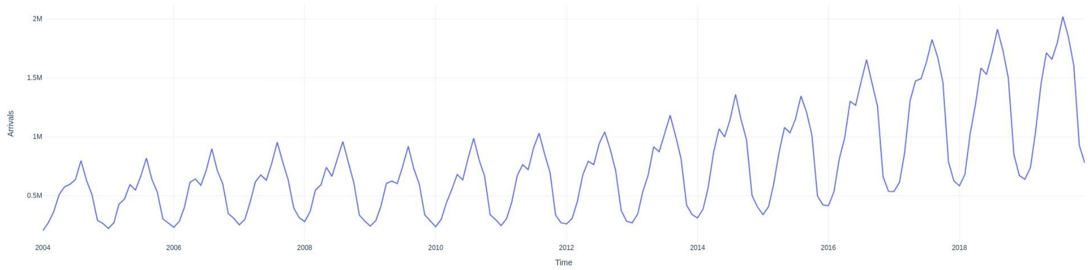


Figure 4.8: Monthly arrivals chart for Portugal, from '2004-01' to '2009-12'.

Figure 4.9 has a chart generated from the Portuguese dataframe.

To use the inflation data for the daily models, imputation methods were used to transform the data from monthly to daily. Section 4.3.9 will discuss the issue of imputing in more detail.

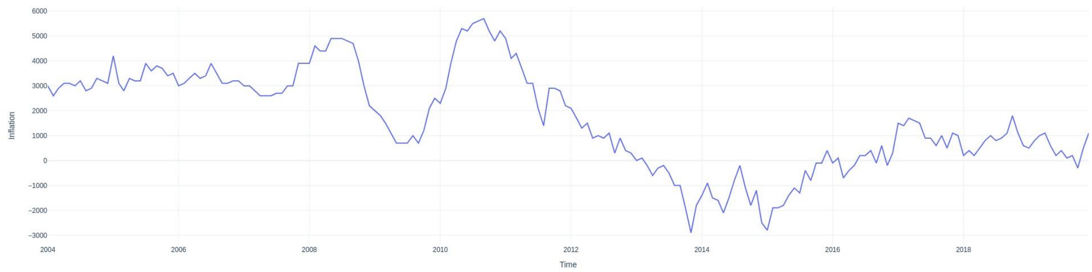


Figure 4.9: Monthly inflation chart for Portugal, from '2004-01' to '2019-12'.

4.3.6 Eurostat Quarterly GDP

After the processing talked about in section 4.3.3, the GDP data is filtered along the columns ['na_item', 'unit', 's_adj']. 'na_item' is set to 'B1GQ', which stands for "Gross domestic product at market prices", 'unit' is set to 'CP_MEUR', which stands for "Current prices in million euro" and 's_adj' is set to 'NSA', which stands for "Unadjusted data", as the real values are wanted. Finally, the 'value' column is renamed to 'gdp'.

The data is separated into five dataframes for each of the countries Portugal, Spain, Italy, France and Greece. Figure 4.10 has a chart generated with the Portuguese dataframe.

This data is quarterly and, thus, needs imputing as to be converted into monthly data. More about this process in section 4.3.9.

4.3.7 Currency Exchange Rate

As previously mentioned, the currency exchange data has five values, which are 'Open', 'High', 'Low', 'Close' and 'AdjClose'. For consistency's sake, the value for 'Open' is used.

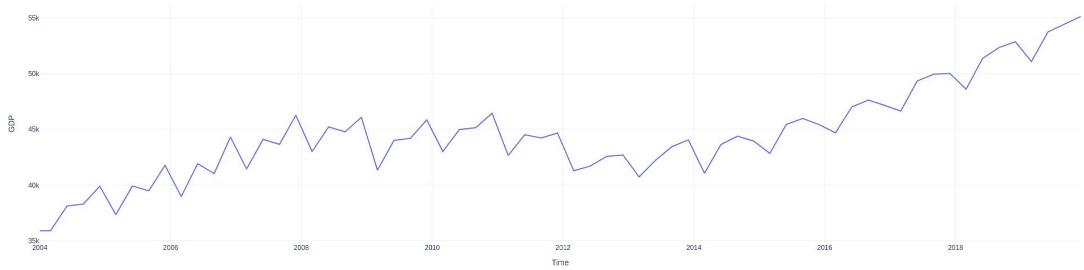


Figure 4.10: Quarterly GDP chart for Portugal, from '2004-01' to '2019-12'.

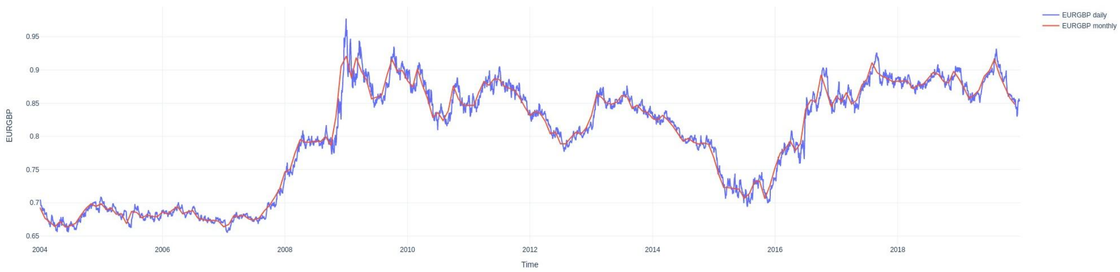


Figure 4.11: Daily and monthly currency exchange rate chart for 'EURGBP_Open', from '2004-01-01' to '2019-12-31'.

While the real values are used directly for the daily models, as the data-set itself has daily frequency, in order to use this data for monthly models, the average 'Open' value per month is calculated.

This data contained a lot of gaps, which will be treated in section 4.3.9.

Figure 4.11 shows the chart of the daily and monthly exchange rate data for 'EU-RGBP_Open'.

4.3.8 Google Trends

In section 4.2.4, the process of Google Trends data collection is shown. Given the very customized solution created, the data comes with the desired formatting. However, as forementioned in section 4.2.4, some keywords do not yield usable results, as Google Trends only saves data for a keyword that reaches an arbitrary threshold of total searches. This can produce gaps that render the data unusable. Figure 4.12 shows an example of such unusable data. The kept columns are shown in sub-section 4.3.11.

4.3.9 Imputing

When using publicly available data, as is the case in this work, it is common to encounter data with gaps or missing values. In such cases, there are three options: (i) Keep the data as is, if it does not pose a problem; (ii) Delete features that are too sparse; (iii) Impute the data.

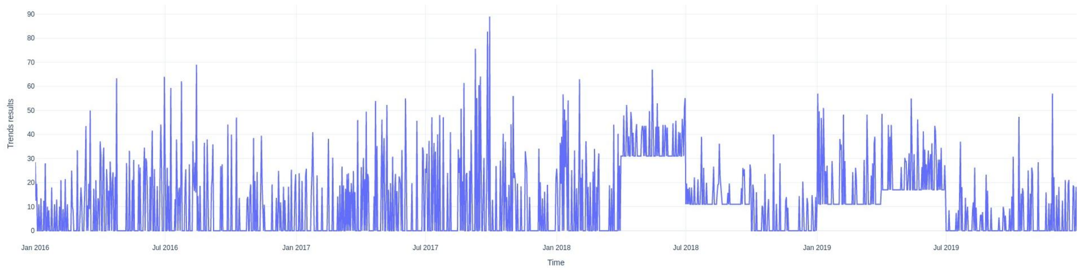


Figure 4.12: Google Trends results for the keyword ‘Greece tourist attractions’.

Imputation is the process of replacing missing values with artificial ones in order to fix the integrity of the data as a whole. A wide range of imputation techniques exist, and their choice depends on the characteristics of the data to impute. In this case, since time-series data is used, gaps of multiple steps are the main target of imputation. One common method of imputation is interpolation. It involves using mathematical methods, such as regression techniques, to estimate values within a gap, based on the two known data points on its edges. Its simplest form is linear interpolation, which draws a straight line between the ends of the gap. For this work, quadratic interpolation is used, which creates a smooth curve in place of the missing values.

In addition to filling in gaps in data, imputing can also serve to artificially enhance the data’s frequency. This is accomplished by simulating gaps between each value of the time-series, generating additional data points between existing ones. This technique is particularly useful for integrating lower-frequency data into a higher-frequency context. This strategy is used in this work to fit the Gross Domestic Product (GDP) data-set, which has quarterly frequency, into the monthly models data. Moreover, it is also used to add the monthly inflation data in the daily models’ features.

4.3.10 Correlations

Correlation is a statistical measure that quantifies the degree to which two variables are related. It indicates the direction and strength of the relationship between the variables. A positive correlation suggests that as one variable increases, the other follows its trend, while a negative correlation implies that as one variable increases, the other tends to decrease. A correlation coefficient quantifies this relationship, with values ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation), and 0 indicating no correlation.

To show an example of the correlation between features and target variable, figure 4.13 was produced. It contains the correlations of the monthly features for Portugal with the its monthly arrivals. Although some features have considerable correlation with the target, like ‘gdp’ and ‘Portugal_weather’, a large amount of them don’t have significant values. Even so, they will be kept, as the deep learning models used in this work can fine

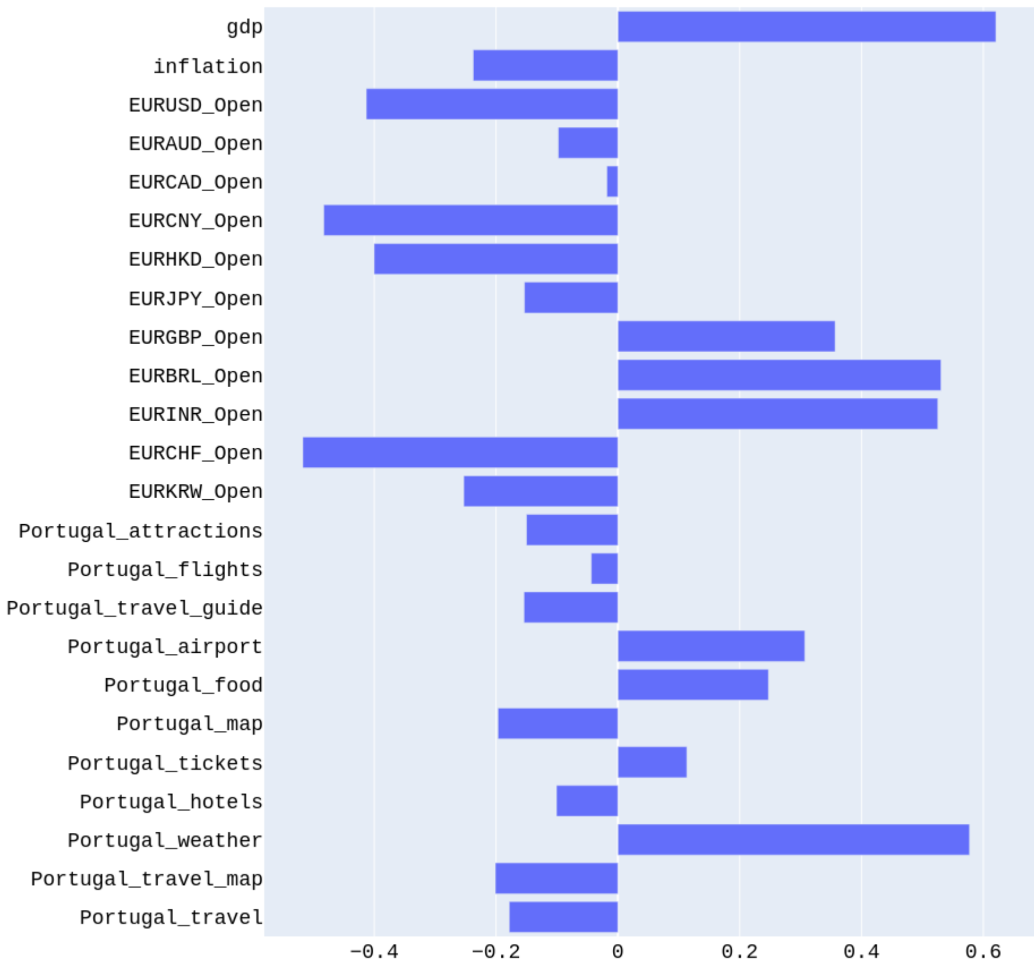


Figure 4.13: Correlations between Portuguese monthly data features and the monthly arrivals for Portugal.

tune the use of each of them, and discover ways to improve results.

4.3.11 Clean Data Overview

After all the data cleaning, the remaining data boils down to 10 dataframes, two with daily and monthly data for each of the five countries Portugal, Spain, Italy, France and Greece.

Each daily dataframe contains 1461 rows of data from '2016-01-01' to '2019-12-31' with the following country specific columns: Daily arrivals, Google Trends volume of search per custom keywords, currency exchange rates and inflation.

When it comes to monthly dataframes, the 192 rows of data range from '2004-01' to

'2019-12' - except for France, which ranges from '2011-01' to '2019-12', with 65 rows - and has the country specific columns: Monthly arrivals, Google Trends volume of search per custom keywords, currency exchange rates, inflation and GDP.

The currency exchance rate features are the following: ['EURKRW_Open', 'EURCHF_Open', 'EURINR_Open', 'EURBRL_Open', 'EURGBP_Open', 'EURJPY_Open', 'EURHKD_Open', 'EURCNY_Open', 'EURCAD_Open', 'EURAUD_Open', 'EURUSD_Open'].

For each country and frequency, these are the clean Google Trends features:

- Portugal:
  - Monthly: ['Portugal_travel', 'Portugal_travel_map', 'Portugal_weather', 'Portugal_hotels', 'Portugal_tickets', 'Portugal_map', 'Portugal_food', 'Portugal_airport', 'Portugal_travel_guide', 'Portugal_flights', 'Portugal_attractions']
  - Daily: ['Portugal_shopping_norm', 'Portugal_airport_norm', 'Portugal_travel_guide_norm', 'Portugal_travel_map_norm', 'Portugal_travel_norm', 'Portugal_tickets_norm', 'Portugal_map_norm', 'Portugal_flights_norm', 'Portugal_hotels_norm', 'Portugal_weather_norm', 'Portugal_attractions_norm', 'Portugal_food_norm']
- Spain:
  - Monthly: ['Spain_attractions', 'Spain_travel_guide', 'Spain_food', 'Spain_map', 'Spain_tickets', 'Spain_travel_map', 'Spain_weather', 'Spain_travel', 'Spain_airport', 'Spain_hotels', 'Spain_flights']
  - Daily: ['Spain_food_norm', 'Spain_weather_norm', 'Spain_accommodation_norm', 'Spain_travel_map_norm', 'Spain_attractions_norm', 'Spain_airport_norm', 'Spain_travel_norm', 'Spain_flights_norm']
- Italy:
  - Monthly: ['Italy_map', 'Italy_flights', 'Italy_travel_guide', 'Italy_weather', 'Italy_travel_map', 'Italy_food', 'Italy_travel', 'Italy_attractions', 'Italy_hotels', 'Italy_tickets', 'Italy_airport']
  - Daily: ['Italy_map_norm', 'Italy_flights_norm', 'Italy_accommodation_norm', 'Italy_travel_norm', 'Italy_weather_norm', 'Italy_food_norm', 'Italy_hotels_norm']
- France:
  - Monthly: ['France_travel', 'France_weather', 'France_airport', 'France_travel_map', 'France_map', 'France_tickets', 'France_hotels', 'France_travel_guide', 'France_food', 'France_attractions', 'France_flights']
  - Daily: ['France_weather_norm', 'France_flights_norm', 'France_accommodation_norm', 'France_map_norm', 'France_attractions_norm', 'France_hotels_norm', 'France_travel_guide_nor 'France_travel_map_norm', 'France_travel_norm', 'France_food_norm', 'France_tickets_norm']
- Greece:

- Monthly: ['Greece_travel_guide', 'Greece_airport', 'Greece_travel_map', 'Greece_food', 'Greece_weather', 'Greece_attractions', 'Greece_hotels', 'Greece_tickets', 'Greece_flights', 'Greece_map', 'Greece_travel']
- Daily: ['Greece_flights_norm', 'Greece_food_norm', 'Greece_map_norm']

4.4 Modeling

In this section, the process of training and evaluating the models is shown. Firstly, baseline models will be constructed as to evaluate the forecastability of the data and have a point of comparison to the latter, more advanced models. Afterwards, comes choosing, fitting and evaluating the main Deep Learning models. Finally, the models are evaluated and compared amongst themselves. All the experiments hereafter are executed on a personal computer running Ubuntu 22.04. The computer's specifications are the following: CPU- Ryzen 5 5600X, GPU- Nvidia GeForce RTX 3050, Memory- 16GB 3200MHz.

4.4.1 Baseline Models

In this work, there are five baseline models to be considered: Naïve Drift, Naïve Moving Average, Naïve Seasonal, Exponential Smoothing, and Seasonal Auto-Regressive Integrated Moving Average (SARIMA). For all the models, the train/test split was 0.8/0.2, and all models are univariate, using only arrivals as the feature and target.

The Naïve models simply take previous information and use it as the new predictions. Naïve Drift draws a line between the first and last values of the training data and extends it to the test data.

The Naïve Moving Average simply uses the moving average of the previous seasonal cycle. Its implementation in darts requires one parameter, named input_chunk_length. It was set as 365 and 72 for the daily and monthly model, respectively.

The Naïve Seasonal uses the same exact values of the previous seasonal cycle as the new prediction. Its implementation takes the number of steps that constitute the seasonal period as a parameter, which was set to 365 for the daily and 12 for the monthly models.

Exponential smoothing in forecasting involves using a weighted average of past observations to predict future points. The weights decrease exponentially as the data points get older. This gives more importance to recent observations. Moreover, the technique is generally computationally efficient, therefore a great addition to the baseline models.

The SARIMA model was chosen for its seasonal component added to the classic and highly successful Auto-Regressive Integrated Moving Average (ARIMA). Chapter 2 already introduced ARIMA, highlighting its hyperparameters, p, d and q. Similarly, SARIMA requires these hyperparameters and, additionally, their seasonal counterpart and the seasonal interval. In total, it requires the regular p, d and q, the seasonal P, D and Q, and the number of steps that constitute one season. After analysis, the hyperparameters (p, d, q)X(P, D, Q) were set as = (2, 1, 2)X(2, 1, 2), and the season step number set to

Country	Model	Daily RMSE	Daily R2	Daily MAPE	Monthly RMSE	Monthly R2	Monthly MAPE
Portugal	Naïve Drift	0.32120	-2.29383	0.38714	0.32594	-0.59764	0.84056
	Naïve MA	0.20808	-0.38235	0.27482	0.34481	-0.78801	0.45007
	Naïve Seasonal	0.08095	0.79081	0.10969	0.16297	0.60059	0.30430
	Exp Smoothing	0.33124	-2.50292	0.39570	0.08911	0.88059	0.20858
	SARIMA	0.36869	-40.92178	0.89813	0.05639	0.93157	0.08402
Spain	Naïve Drift	0.24686	-1.03076	0.31037	0.44743	-1.28185	1.35229
	Naïve MA	0.19622	-0.28298	0.24657	0.32434	-0.19906	0.62836
	Naïve Seasonal	0.05914	0.88345	0.07097	0.07655	0.93321	0.16522
	Exp Smoothing	0.23233	-0.79875	0.33228	0.04180	0.98009	0.11744
	SARIMA	0.32339	-50.47069	0.60164	0.03008	0.98839	0.05323
Italy	Naïve Drift	0.17393	-0.69102	0.21218	0.45046	-1.09323	2.13327
	Naïve MA	0.14937	-0.24713	0.18056	0.32191	-0.06900	1.02575
	Naïve Seasonal	0.09757	0.46784	0.11002	0.08574	0.92416	0.17870
	Exp Smoothing	0.17781	-0.76723	0.23415	0.05828	0.96496	0.08440
	SARIMA	0.25184	-17.02387	0.41018	0.06333	0.94945	0.09197
France	Naïve Drift	0.20123	-0.25322	0.38515	0.45294	-1.18432	0.60887
	Naïve MA	0.19022	-0.11983	0.31198	0.33336	-0.18319	0.81944
	Naïve Seasonal	0.15767	0.23062	0.22731	0.07298	0.94329	0.15162
	Exp Smoothing	0.21334	-0.40857	0.34663	0.06089	0.96053	0.13047
	SARIMA	0.19154	-2.99483	0.29439	0.07048	0.94406	0.16267
Greece	Naïve Drift	0.40315	-1.57801	0.47534	0.41639	-0.50621	7.78177
	Naïve MA	0.26767	-0.13643	0.46212	0.35695	-0.10687	3.14033
	Naïve Seasonal	0.05394	0.95385	0.09199	0.14171	0.82553	0.31316
	Exp Smoothing	0.35742	-1.02624	0.51584	0.12042	0.87402	0.29303
	SARIMA	0.34960	-31.27249	0.90804	0.10763	0.85150	0.23812

Table 4.1: Metrics results of the daily and monthly baselines models.

seven for the daily models – SARIMA is not able to use 365 step seasons, seven steps (one week) was the second most adequate seasonal window – and 12 for the monthly models.

Table 4.1 contain the results of the daily and monthly models using three metrics: Root Mean Square Error (RMSE), R-Squared (R2) and Mean Absolute Percentage Error (MAPE).

When it comes to the daily models, it's clear to see that most algorithms did not

```
X_train shape:  torch.Size([84, 12, 25])
y_train shape:  torch.Size([84, 1])
X_val shape:    torch.Size([36, 12, 25])
y_val shape:    torch.Size([36, 1])
X_test shape:   torch.Size([36, 12, 25])
y_test shape:   torch.Size([36, 1])
```

Figure 4.14: Tensor shapes example.

perform well. This is due to the fact that most statistical models don't work well with seasonality periods as large as this daily data with strong yearly seasonality. Only the Naïve Seasonal model has acceptable results, since, as was just mentioned, the time-series has strong seasonal patterns, and this model simply uses the last seasonal period as its prediction.

Out of all the monthly baseline models, SARIMA and exponential smoothing seem to achieve the best results. The fact that exponential smoothing was able to at times outperform SARIMA is somewhat surprising, considering the fact that SARIMA is widely regarded as one of the best forecasting models for time-series, while exponential smoothing is a simple numerical analysis technique.

4.4.2 Deep Learning Based Models

Having created some baseline models, it is now adequate to develop some more complex models. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) will be used to create such models. For an introduction about the inner workings of these models, please refer to the previous chapter's sections 3.3.4.1 and 3.3.4.2.

This work includes models for five different countries: Portugal, Spain, France, Italy and Greece. Each of them has its own set of feature variables, depending on what data is available and which features correlated better with the target. Section 4.3.11 refers to what data was used for each country.

The developed function to train the models uses PyTorch's standard procedure in model development. The feature data is loaded into tensors of shape [S, lookback, F] and the target into tensors of shape [S, T], where S represents the number of steps/rows in the original time-series, lookback represents the lookback window for the algorithm, that is, the amount of steps the algorithm can look back to in order to make its prediction, F represents the number of features and T represents the number of targets. As an example, figure 4.14 shows the shapes of tensors X_train, y_train, X_val, y_val, X_test and y_test. The variables beginning with X represent the features, while the ones beginning with y are the target. Train and test represent the train/test split. As can be seen, in this example there are 931 rows of train data and 348 rows of test data, the lookback window on this case was 12, and there were 25 features and 1 target.

Next, the PyTorch library's model class was extended to create the LSTM and GRU models, which are set to use Adam gradient optimization. The function created to train the models checks every 10 epochs if the RMSE of the model's predictions is still improving. If the model worsens for a predetermined amount of iterations – which was five for most use –, it saves the best trained model, calculating the evaluation metrics, and stops. The metric chosen to evaluate and rank the models is the RMSE, since it is the most commonly used to evaluate LSTM and GRU.

To tune the the models' hyperparameters, Bayesian optimization is used. For an introduction to Bayesian optimization, refer to section 3.3.2. LSTM and GRU have five main hyperparameters that change both the results' accuracy and the run time of the algorithm: hidden size (int), number of layers (int), lookback (int), learn rate (float) and batch size (int). Because Bayesian optimization inherently operates with floating point numbers, any whole number hyperparameters produced by the algorithm are converted to integers within the "black box function".

The limits of the intervals fed to the optimization algorithm were chosen so that the corresponding whole numbers would have the same chance of being picked. To achieve a [1, 10] interval, [0.51, 10.49] is inserted. After trial and error testing, there was a decision to manually insert the values for the learn rate and batch size, since these parameters seemed to only change the speed of execution, as long as they stayed within reasonable limits.

In each Bayesian optimization run, the parameters 'Number of initial points' and 'Number of iterations' are configured to 75 and 150, respectively. This configuration generates 225 unique models for every combination of country, model, and frequency. The model with the lowest RMSE is selected for further analysis in the following sections. RMSE serves as the chosen evaluation metric given its role in assessing model convergence during training.

In this work, there are three sets of features used to model: (i) all the features available; (ii) only Google Trends data and arrivals; (iii) only arrivals. This will help identify which data is best to train these models.

Table 4.2 shows the time in seconds that Bayesian optimization took for each of the frequency/country/algorithm/feature-set combination. F1 represents the feature-set of all features, F2 the feature-set of only Google Trends data and arrivals and F3 the feature-set of only arrivals.

4.4.2.1 Daily Models Results

As forementioned, Bayesian optimization is used in this project to determine the hyperparameters of the models in an empirical manner. In order to efficiently explore optimal parameter configurations, the algorithm requires a predefined search space, with minimum and maximum values for each hyperparameter. During manual testing, the adequate intervals for hidden size, number of layers and lookback were defined as [0.51, 100.49], [0.51, 10.49] and [0.51, 90.49], respectively. The maximum value for lookback is set to 90, which corresponds to 3 months of previous information. This is the most that the scarce daily data available allows without sacrificing too much of the usable data.

The features used for all daily models are the country specific Google Trends data and inflation, and the currency exchange rate data.

The metrics' results of the daily models are displayed in table 4.3. Table 4.4 displays models' hyperparameters that generated the best metrics results. Table 4.5 shows the means and standard deviations for each type of model characteristic.

4.4.2.2 Monthly Models Results

For the Bayesian optimization of the monthly models, the parameters are the same as for the daily ones, except for the lookback interval. The intervals are [0.51, 100.49], [0.51, 10.49] and [0.51, 24.49] for hidden size, number of layers and lookback, respectively. A maximum of 24 lookback corresponds to 24 months, or two years.

After running Bayesian optimization, two sets of models were picked for evaluation. The first set contains the models for each country, feature set and algorithm that achieved the best validation set metrics. These models will be called Best Validation-set Metrics (BVM) monthly models. The second set contains the models that had the best metrics for the test set, and will be called Best Test-set Metrics (BTM) monthly models. Note that on table 4.2, containing the execution times of the Bayesian optimization, there are only three monthly models, as both BVM and BTM models were extracted from the same Bayesian optimization run. This also means that, for any combination of frequency, country, algorithm and feature-set, the BVM model and the BTM model could be the same, if the model that produced the best validation-set metrics also produced the best test-set metrics.

The features used for all monthly models are the country specific Google Trends data, gdp and inflation, and the currency exchange rate data.

The metrics' results of the monthly models are displayed in tables 4.6 and 4.9. Tables 4.7 and 4.10 display models' hyperparameters that generated the best metrics results. On tables 4.8 and 4.11, the means and standard deviations are presented for each type of model characteristic.

Country	Alg	Daily F1	Daily F2	Daily F3	Monthly F1	Monthly F2	Monthly F3
Portugal	LSTM	2949	2671	6641	471	627	448
	GRU	3568	4333	8188	507	940	514
Spain	LSTM	2626	4438	5678	677	1032	613
	GRU	4228	7661	10090	621	1201	799
Italy	LSTM	2074	5781	4808	702	1000	692
	GRU	3137	4927	5238	758	1034	817
France	LSTM	2505	3264	7750	269	490	873
	GRU	2803	4720	11725	277	536	605
Greece	LSTM	4038	6808	7211	743	939	720
	GRU	5445	8593	4786	873	1116	819

Table 4.2: Execution times of all Bayesian optimization runs in seconds.

Country	Features	Model	RMSE	R2	MAPE
Portugal	All features	LSTM	0.05979	0.91037	0.09719
		GRU	0.05452	0.92582	0.08869
	Trends and arrivals	LSTM	0.06076	0.90912	0.10076
		GRU	0.04595	0.94706	0.06749
	Arrivals	LSTM	0.04486	0.95008	0.06522
		GRU	0.03916	0.96192	0.05496
Spain	All features	LSTM	0.05891	0.90646	0.08979
		GRU	0.05815	0.90490	0.08550
	Trends and arrivals	LSTM	0.05354	0.92387	0.08798
		GRU	0.04305	0.94883	0.05949
	Arrivals	LSTM	0.04358	0.94822	0.05957
		GRU	0.03642	0.96288	0.04699
Italy	All features	LSTM	0.07513	0.74559	0.12620
		GRU	0.06733	0.78649	0.08643
	Trends and arrivals	LSTM	0.06059	0.83456	0.09664
		GRU	0.05561	0.85696	0.07046
	Arrivals	LSTM	0.05821	0.84918	0.08978
		GRU	0.04937	0.88569	0.05784
France	All features	LSTM	0.10066	0.72222	0.17749
		GRU	0.07732	0.82944	0.14501
	Trends and arrivals	LSTM	0.07852	0.82386	0.12594
		GRU	0.06889	0.86476	0.12515
	Arrivals	LSTM	0.06562	0.87714	0.11216
		GRU	0.06319	0.88593	0.10426
Greece	All features	LSTM	0.03427	0.98209	0.07440
		GRU	0.02823	0.98841	0.06385
	Trends and arrivals	LSTM	0.02682	0.98954	0.06220
		GRU	0.02249	0.99265	0.05249
	Arrivals	LSTM	0.02570	0.99022	0.06382
		GRU	0.02066	0.99378	0.04027

Table 4.3: Metric results for the daily models.

Country	Features	Model	Hidden size	Layers (#)	Lookback
Portugal	All features	LSTM	84	1	7
		GRU	63	1	10
	Trends and arrivals	LSTM	100	1	18
		GRU	83	2	89
	Arrivals	LSTM	54	1	75
		GRU	51	4	86
Spain	All features	LSTM	95	10	7
		GRU	66	8	90
	Trends and arrivals	LSTM	56	1	16
		GRU	90	3	84
	Arrivals	LSTM	98	1	74
		GRU	94	3	89
Italy	All features	LSTM	83	1	7
		GRU	90	1	84
	Trends and arrivals	LSTM	77	4	7
		GRU	35	1	73
	Arrivals	LSTM	67	1	17
		GRU	91	3	78
France	All features	LSTM	96	5	7
		GRU	72	2	81
	Trends and arrivals	LSTM	62	1	80
		GRU	55	1	75
	Arrivals	LSTM	59	1	81
		GRU	36	4	79
Greece	All features	LSTM	67	10	7
		GRU	61	7	83
	Trends and arrivals	LSTM	95	2	85
		GRU	100	1	71
	Arrivals	LSTM	67	1	29
		GRU	33	3	75

Table 4.4: Hyperparameters for the daily models.

Models	Stats	H Size	Layers (#)	Lookback	RMSE	R2	MAPE
LSTM	Mean	77.33333	2.73333	34.46667	0.05646	0.89083	0.09528
	Std	16.08588	3.08689	32.11618	0.01924	0.07996	0.03032
GRU	Mean	68.0000	2.93333	76.46667	0.04869	0.91570	0.07659
	Std	22.07865	2.08060	18.68642	0.01659	0.06107	0.02863
All features	Mean	77.7000	4.60000	38.3000	0.06143	0.87018	0.10346
	Std	12.77537	3.66606	37.79167	0.01980	0.08934	0.03346
Trends and arrivals	Mean	75.3000	1.70000	59.8000	0.05162	0.90912	0.08486
	Std	21.14734	1.00499	30.76622	0.01666	0.05848	0.02535
Arrivals	Mean	65.0000	2.20000	68.3000	0.04468	0.93050	0.06949
	Std	21.98181	1.24900	23.25962	0.01422	0.04874	0.02302
Portugal	Mean	72.5000	1.66667	47.5000	0.05084	0.93406	0.07905
	Std	17.76467	1.10554	36.23419	0.00804	0.02022	0.01731
Spain	Mean	83.16667	4.33333	60.0000	0.04894	0.93253	0.07155
	Std	16.10814	3.44803	34.78026	0.00842	0.02218	0.01678
Italy	Mean	73.83333	1.83333	44.33333	0.06104	0.82641	0.08789
	Std	19.16956	1.21335	34.31067	0.00829	0.04683	0.02143
France	Mean	63.33333	2.33333	67.16667	0.07570	0.83389	0.13167
	Std	18.16284	1.59861	26.98405	0.01250	0.05496	0.02410
Greece	Mean	70.5000	4.00000	58.33333	0.02636	0.98945	0.05950
	Std	22.32898	3.36650	29.58979	0.00436	0.00376	0.01069

Table 4.5: Means and standard deviations of the daily models' results.

Country	Features	Model	Validation RMSE	Validation R2	Validation MAPE	Test RMSE	Test R2	Test MAPE
Portugal	All features	LSTM	0.05338	0.90899	0.16094	0.36842	-1.2715	0.41078
		GRU	0.05000	0.92785	0.23080	0.31229	-0.44753	0.47336
	Trends and arrivals	LSTM	0.04614	0.93661	0.16749	0.11455	0.79789	0.16321
		GRU	0.06941	0.85656	0.26291	0.26003	-0.04151	0.29170
	Arrivals	LSTM	0.05988	0.88872	0.16589	0.20600	0.31604	0.24977
		GRU	0.04185	0.94903	0.10314	0.16379	0.58787	0.29377
Spain	All features	LSTM	0.06821	0.92728	0.21010	0.13113	0.79884	0.19861
		GRU	0.05507	0.95370	0.17197	0.28054	0.10296	0.60668
	Trends and arrivals	LSTM	0.04354	0.97036	0.12155	0.14232	0.76304	0.23824
		GRU	0.03935	0.97648	0.13528	0.16685	0.68207	0.26292
	Arrivals	LSTM	0.03934	0.97649	0.09719	0.08667	0.91422	0.10272
		GRU	0.03815	0.97805	0.09806	0.10368	0.87878	0.20269
Italy	All features	LSTM	0.04990	0.96528	0.16836	0.14675	0.77585	0.36443
		GRU	0.05736	0.95449	0.27871	0.22937	0.45728	0.64764
	Trends and arrivals	LSTM	0.04528	0.97179	0.12314	0.16839	0.70676	0.40160
		GRU	0.03271	0.98528	0.08466	0.07179	0.94670	0.20722
	Arrivals	LSTM	0.03405	0.98396	0.13773	0.06964	0.94870	0.16412
		GRU	0.02838	0.98937	0.10548	0.05015	0.97421	0.11736
France	All features	LSTM	0.05414	0.96677	0.15826	0.11188	0.86896	0.36737
		GRU	0.04130	0.98066	0.23267	0.09869	0.89803	0.22447
	Trends and arrivals	LSTM	0.06060	0.95479	0.29218	0.12888	0.80854	0.20355
		GRU	0.05670	0.96355	0.20826	0.12445	0.83785	0.29437
	Arrivals	LSTM	0.05409	0.96683	0.18447	0.08172	0.93009	0.16227
		GRU	0.04325	0.97879	0.16798	0.06684	0.95322	0.15866
Greece	All features	LSTM	0.05284	0.94851	0.66934	0.17494	0.73960	1.16547
		GRU	0.04177	0.96774	0.45546	0.19715	0.66763	0.40523
	Trends and arrivals	LSTM	0.03210	0.98100	0.19819	0.14725	0.81551	0.34370
		GRU	0.02594	0.98760	0.31200	0.14582	0.81909	0.35722
	Arrivals	LSTM	0.02041	0.99273	0.25437	0.08966	0.93575	0.32677
		GRU	0.02482	0.98866	0.54640	0.07379	0.95383	0.45164

Table 4.6: Metric results for the monthly BVM models.

Country	Features	Model	Hidden size	Layers (#)	Lookback
Portugal	All features	LSTM	77	7	14
		GRU	86	4	11
	Trends and arrivals	LSTM	98	9	12
		GRU	16	9	12
	Arrivals	LSTM	66	10	13
		GRU	94	2	18
Spain	All features	LSTM	81	9	13
		GRU	91	4	9
	Trends and arrivals	LSTM	31	6	13
		GRU	58	10	12
	Arrivals	LSTM	78	9	12
		GRU	98	4	17
Italy	All features	LSTM	96	6	7
		GRU	72	4	9
	Trends and arrivals	LSTM	79	6	12
		GRU	83	10	12
	Arrivals	LSTM	94	4	15
		GRU	97	2	24
France	All features	LSTM	83	10	12
		GRU	33	3	12
	Trends and arrivals	LSTM	61	1	9
		GRU	19	10	12
	Arrivals	LSTM	95	9	12
		GRU	82	10	12
Greece	All features	LSTM	76	7	13
		GRU	94	8	14
	Trends and arrivals	LSTM	50	7	13
		GRU	88	9	13
	Arrivals	LSTM	85	1	19
		GRU	98	1	12

Table 4.7: Hyperparameters for the BVM monthly models.

Models	Stats	H Size	Layers (#)	Lookback	Validation val RMSE	val R2	val MAPE	Test test RMSE	test R2	test MAPE
LSTM	Mean	76.66667	6.73333	12.6000	0.04759	0.95601	0.20728	0.14455	0.65655	0.32417
	Std	17.80886	2.79205	2.52455	0.01197	0.02822	0.13274	0.07010	0.53582	0.24429
GRU	Mean	73.93333	6.00000	13.26667	0.04307	0.96252	0.22625	0.15635	0.61803	0.33300
	Std	27.79081	3.34664	3.69624	0.01230	0.03303	0.12851	0.08089	0.41345	0.15157
All features	Mean	78.9000	6.20000	11.4000	0.05240	0.95013	0.27366	0.20512	0.35901	0.48640
	Std	17.04377	2.27156	2.24499	0.00731	0.02120	0.15641	0.08620	0.67396	0.26354
Trends and arrivals	Mean	58.3000	7.70000	12.0000	0.04518	0.95840	0.19057	0.14703	0.71359	0.27637
	Std	27.66243	2.68514	1.09545	0.01301	0.03691	0.07402	0.04604	0.26098	0.07200
Arrivals	Mean	88.7000	5.20000	15.4000	0.03842	0.96926	0.18607	0.09919	0.83927	0.22298
	Std	10.12966	3.65513	3.85227	0.01171	0.02944	0.12907	0.04596	0.20445	0.10301
Portugal	Mean	72.83333	6.83333	13.33333	0.05344	0.91129	0.18186	0.23751	-0.00979	0.31377
	Std	27.53432	2.91071	2.28522	0.00909	0.03122	0.05174	0.08639	0.69525	0.10209
Spain	Mean	72.83333	7.00000	12.66667	0.04728	0.96373	0.13903	0.15186	0.68998	0.26864
	Std	22.46046	2.44949	2.35702	0.01097	0.01827	0.04059	0.06308	0.27318	0.15917
Italy	Mean	86.83333	5.33333	13.16667	0.04128	0.97503	0.14968	0.12268	0.80158	0.31706
	Std	9.44134	2.49444	5.45945	0.01034	0.01237	0.06328	0.06418	0.18289	0.17982
France	Mean	62.16667	7.16667	11.5000	0.05168	0.96857	0.20730	0.10208	0.88278	0.23511
	Std	27.75138	3.71558	1.11803	0.00702	0.00887	0.04532	0.02230	0.05026	0.07446
Greece	Mean	81.83333	5.50000	14.0000	0.03298	0.97771	0.40596	0.13810	0.82190	0.50834
	Std	15.83684	3.25320	2.30940	0.01116	0.01531	0.16659	0.04372	0.10082	0.29681

Table 4.8: Means and standard deviations of the BVM monthly models' results.

Country	Features	Model	Validation RMSE	Validation R2	Validation MAPE	Test RMSE	Test R2	Test MAPE
Portugal	All features	LSTM	0.08207	0.78487	0.33197	0.23684	0.06126	0.26346
		GRU	0.15034	0.29857	0.82089	0.18453	0.45114	0.38072
	Trends and arrivals	LSTM	0.04614	0.93661	0.16749	0.11455	0.79789	0.16321
		GRU	0.10032	0.67611	0.29895	0.19380	0.39115	0.28744
	Arrivals	LSTM	0.10514	0.64421	0.44671	0.15339	0.61859	0.25943
		GRU	0.05553	0.90430	0.16816	0.11814	0.77504	0.16324
Spain	All features	LSTM	0.07320	0.91623	0.26553	0.09719	0.88949	0.19985
		GRU	0.07162	0.91798	0.17050	0.08225	0.91928	0.16826
	Trends and arrivals	LSTM	0.06165	0.94058	0.21751	0.07535	0.93358	0.15743
		GRU	0.07107	0.92105	0.28785	0.08007	0.92500	0.15692
	Arrivals	LSTM	0.04786	0.96419	0.13307	0.06987	0.94289	0.09281
		GRU	0.04556	0.96846	0.12406	0.07405	0.93737	0.10541
Italy	All features	LSTM	0.05661	0.95494	0.17518	0.07787	0.93565	0.18056
		GRU	0.06503	0.94054	0.30139	0.07203	0.94494	0.19222
	Trends and arrivals	LSTM	0.06110	0.94751	0.25569	0.08009	0.93193	0.16215
		GRU	0.03962	0.97840	0.12338	0.07170	0.94684	0.20382
	Arrivals	LSTM	0.03711	0.98077	0.15938	0.05044	0.97323	0.11246
		GRU	0.02838	0.98937	0.10548	0.05015	0.97421	0.11736
France	All features	LSTM	0.06715	0.94153	0.26311	0.05816	0.95952	0.17358
		GRU	0.07514	0.93435	0.35310	0.07317	0.94205	0.34852
	Trends and arrivals	LSTM	0.11296	0.84294	0.51750	0.06049	0.95783	0.16713
		GRU	0.14327	0.76730	0.73896	0.06394	0.95721	0.24374
	Arrivals	LSTM	0.05409	0.96683	0.18447	0.08172	0.93009	0.16227
		GRU	0.05588	0.96460	0.17128	0.06222	0.95948	0.12481
Greece	All features	LSTM	0.05656	0.94102	0.77730	0.15144	0.80488	1.60745
		GRU	0.05377	0.94670	0.82615	0.09913	0.91639	0.47349
	Trends and arrivals	LSTM	0.03608	0.97600	0.55001	0.12781	0.86102	0.38191
		GRU	0.08087	0.86813	0.75815	0.14349	0.80660	0.58282
	Arrivals	LSTM	0.02617	0.98842	0.20408	0.08369	0.94715	0.33309
		GRU	0.02482	0.98866	0.54640	0.07379	0.95383	0.45164

Table 4.9: Metric results for the BTM monthly models.

Country	Features	Model	Hidden size	Layers (#)	Lookback
Portugal	All features	LSTM	78	7	14
		GRU	84	8	13
	Trends and arrivals	LSTM	98	9	12
		GRU	47	1	1
	Arrivals	LSTM	46	5	1
		GRU	76	1	13
Spain	All features	LSTM	81	9	13
		GRU	94	8	14
	Trends and arrivals	LSTM	31	6	13
		GRU	90	8	13
	Arrivals	LSTM	84	1	13
		GRU	86	1	12
Italy	All features	LSTM	87	3	14
		GRU	59	8	14
	Trends and arrivals	LSTM	35	5	14
		GRU	82	10	12
	Arrivals	LSTM	61	1	13
		GRU	97	2	24
France	All features	LSTM	89	2	8
		GRU	47	2	10
	Trends and arrivals	LSTM	23	3	9
		GRU	25	4	12
	Arrivals	LSTM	95	9	12
		GRU	82	10	12
Greece	All features	LSTM	76	8	13
		GRU	88	8	13
	Trends and arrivals	LSTM	50	7	13
		GRU	45	8	2
	Arrivals	LSTM	87	1	22
		GRU	98	1	12

Table 4.10: Hyperparameters for the BTM monthly models.

Models	Stats	H Size	Layers (#)	Lookback	Validation val RMSE	val R2	val MAPE	Test test RMSE	test R2	test MAPE
LSTM	Mean	68.06667	5.06667	12.26667	0.06159	0.91511	0.30993	0.10126	0.83633	0.29445
	Std	24.11215	2.95447	4.18675	0.02332	0.08913	0.17764	0.04758	0.22563	0.35896
GRU	Mean	73.33333	5.33333	11.8000	0.07075	0.87097	0.38631	0.09616	0.85337	0.26669
	Std	21.98686	3.51505	5.03587	0.03547	0.17406	0.26500	0.04284	0.17814	0.14328
All features	Mean	78.3000	6.30000	12.6000	0.07515	0.85767	0.42851	0.11326	0.78246	0.39881
	Std	13.8856	2.64764	1.90788	0.02652	0.19211	0.25487	0.05552	0.28021	0.41489
Trends and arrivals	Mean	52.6000	6.10000	10.1000	0.07531	0.88546	0.39155	0.10113	0.85091	0.25066
	Std	26.14269	2.70000	4.48219	0.03281	0.09354	0.22051	0.04090	0.16343	0.13080
Arrivals	Mean	81.2000	3.20000	13.4000	0.04805	0.93598	0.22431	0.08175	0.90119	0.19225
	Std	15.68949	3.37046	5.90254	0.02224	0.10007	0.14067	0.03014	0.10884	0.11284
Portugal	Mean	71.5000	5.16667	9.00000	0.08992	0.70744	0.37236	0.16687	0.51584	0.25292
	Std	19.02411	3.18416	5.68624	0.03452	0.21191	0.22263	0.04326	0.25287	0.07502
Spain	Mean	77.66667	5.50000	13.0000	0.06183	0.93808	0.19975	0.07980	0.92460	0.14678
	Std	21.28119	3.30404	0.57735	0.01133	0.02153	0.06250	0.00876	0.01751	0.03678
Italy	Mean	70.16667	4.83333	15.16667	0.04797	0.96525	0.18675	0.06705	0.95113	0.16143
	Std	20.77191	3.23608	4.01732	0.01360	0.01838	0.07000	0.01221	0.01676	0.03524
France	Mean	60.16667	5.00000	10.5000	0.08475	0.90293	0.37140	0.06662	0.95103	0.20334
	Std	29.75689	3.26599	1.60728	0.03265	0.07344	0.20145	0.00823	0.01116	0.07389
Greece	Mean	74.0000	5.50000	12.5000	0.04638	0.95149	0.61035	0.11323	0.88164	0.63840
	Std	19.84103	3.20156	5.79511	0.01969	0.04169	0.21177	0.02948	0.06145	0.44031

Table 4.11: Means and standard deviations of the BTM monthly models' results.

4.5 Results Discussion

In order to better understand the results, the initial focus for discussion will be the daily models and the BVM monthly models. This is because these two sets of models present similar results and can be used to draw similar conclusions. Then, the results of the BTM monthly models will be discussed as well.

4.5.1 Daily and BVM Monthly Models

In general, all the deep learning based models achieved quite accurate results. The evaluation metrics prove that both GRU and LSTM modeled the data more accurately than the baseline models. The daily baseline models were clearly not able to model complex daily data accurately, with the exception of Naïve Seasonal, which had good results using the previous year's values, due to the highly seasonal nature of the data. The monthly baseline models had more satisfying results, with SARIMA and exponential smoothing presenting some good metric scores. However, the deep learning models achieved similar or better scores for most countries.

On the other hand, the results of the BVM models' test-set are not as impressive, with metric values much worse than the validation set. This subject will be talked about in more detail at the end of this sub-section.

When comparing the two main models to each other, GRU has categorically outperformed LSTM on the validation-set, but not on the test-set. In the validation-set, it consistently achieved better evaluation metrics scores across all countries and feature sets with average RMSE values of 0.04869 for the daily and 0.04307 for the BVM models, against LSTM's 0.05647 and 0.04759, respectively. All other evaluation metrics reveal the same information, as, generally, the performance rankings stay unchanged when using either RMSE, R2 or MAPE. The situation is reversed for test-set, but it is not reason enough to believe that LSTM outperforms GRU in this strand of the results. This will be explained in section 4.5.2.

When it comes to hyperparameters, while hidden size and number of layers seem to stay within similar ranges for both LSTM and GRU, the lookback seems to divide the two. Moreover, GRU used values significantly higher than its counterpart, almost reaching the imposed limits with an average value of 76.47 against the 34.47 of the LSTM models.

Focusing now on the different country models, the French models seem to have the worst evaluations. On the other hand, the Greek models appear to produce the best results. These facts could be explained by the more complex arrivals time-series from France, comparing to the simpler Greek data. For a visual representation, refer to figures 4.15, 4.16, 4.17 and 4.18. In these images, it's possible to see that the daily chart of the Greek arrivals is much simpler than the French. When seeing the monthly graphs, it's possible to see that the French results are not so bad after all, as the general shape and upper and lower limits of the test data were captured quite accurately. Perhaps the fact

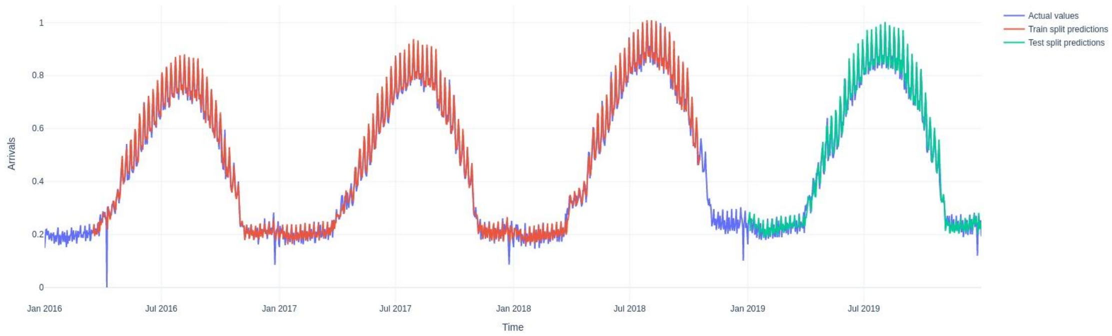


Figure 4.15: Daily GRU predictions for Greece.

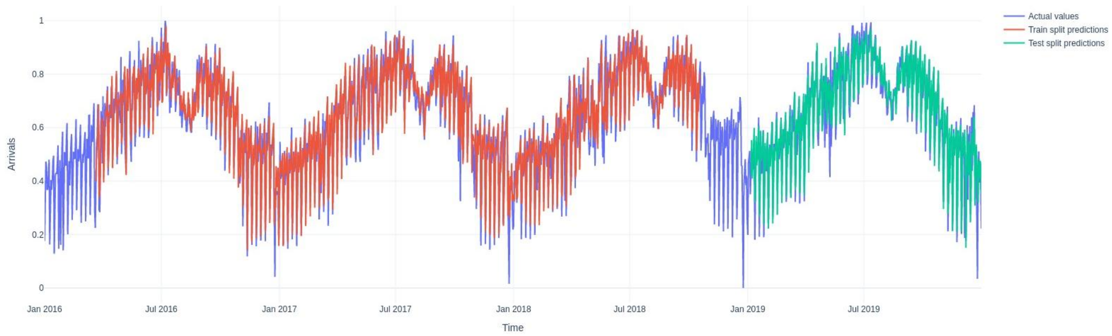


Figure 4.16: Daily GRU predictions for France.

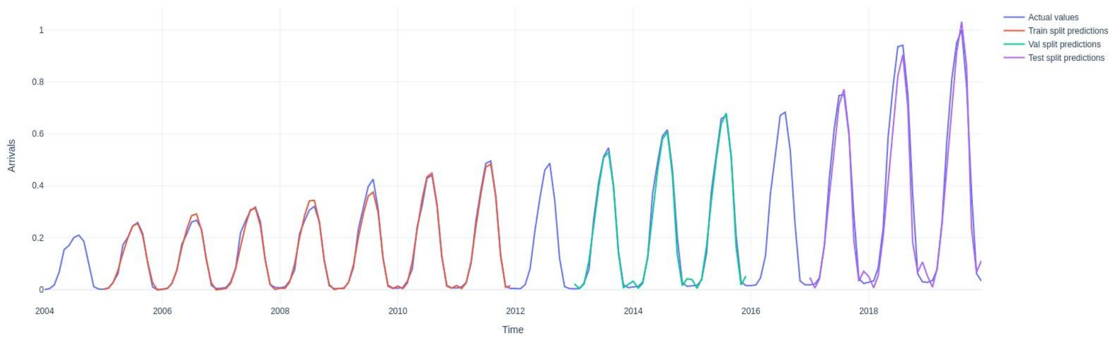


Figure 4.17: BVM and BTM Monthly GRU predictions with only arrivals for Greece.

that the French data-set has less data negatively affected the value of RMSE.

In terms of features used, surprisingly, the less features included in the model training, the better the metrics evaluate the models. It seems that LSTM and GRU could not use the exogenous variables in order to make better predictions. This could be caused by a myriad of factors. One possible cause is low correlation between the exogenous variables chosen and the target variable. This however, does not seem to be the case. One other

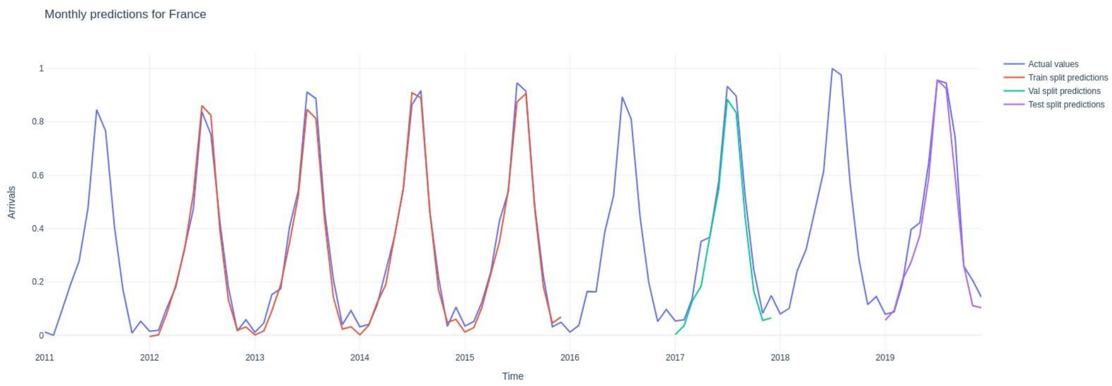


Figure 4.18: BVM Monthly GRU predictions with only arrivals for France.

possible reason is the small lag between the features' time frame and the target's time frame. In the case of the daily models, there is a lag of one day between the features and the target. For the monthly models, the lag is one month. Perhaps the additional feature variables only influence the arrivals within a longer time period. The inflation, GDP, or even Google searches might have influence on the arrivals much later than the lag selected for this project.

As for the execution times, it is firstly important to note that the computer used to train the models was not solely used for this purpose, which could greatly influence the run-time. However, it is clear that the daily models had much lengthier training than the monthly models. This is attributed to the much larger data-set used in modeling. The daily models used data with 1461 rows, while the monthly models had 192 rows of data. Furthermore, LSTM run times were widely shorter than GRU, which seems counter intuitive if GRU's less complex structure is taken into account. However, the convergence of the model during training is highly dependent on the learn rate, which is hard to calibrate from model to model. Perhaps if this hyperparameter was tuned manually for each model, the run times would be different.

Finally, directing the attention to the test set of the BVM models, the results are not as satisfactory. One could argue that these models are unfit to use with new, unseen data. However, the BVM models were picked to have the best validation set possible. This could mean that the chosen "best models" are the models with the highest amount of over-fitting to the validation set. To attempt to prove this hypothesis, sub-section 4.5.2 will bring to attention the BTM models' results and discuss the findings.

4.5.2 BTM Monthly Models

Upon studying the BTM model results, some conclusions differ from the observations made in the previous sub-section 4.5.1.

The first observation is that, even if subpar when comparing with the validation set of the BVM models, the BTM models' metrics show that the outcome for the test set can be much more accurate when picking models that, in theory, did not overfit to the validation data. It is important to note that the process of choosing the best model produced by the Bayesian optimization is inevitably somewhat biased towards the used data. If the BVM models can be overfit to the validation set, the BTM models can also overfit, or coincidentally be specially adept in predicting, the test set.

Unlike what was observed for the BVM models, BVM GRU models outperformed LSTM in the test-set, although the gap between the results of the two is considerably less significant. Moreover, GRU's validation set results are slightly worse than LSTM's and test set results are slightly better, which suggests higher over-fitting on the latter algorithm. This confirms that GRU is indisputably the best performing model of this work, as it consistently achieved the best metric scores for each focused analysis.

The hyperparameters appear uniform between the two algorithms. The averages dictate that the best values for hidden size, number of layers and lookback are 70, 5 and 12.

Although the feature sets present the same behavior that was previously observed, the country models have very different results. The rankings seem to be reversed when comparing the test set metrics. French models have the highest metrics scores and Greek models the lowest. This indicates, once again, that over-fitting plagues the models with good validation set results.

4.5.3 Forecasting/Nowcasting

Forecasting can be used in different contexts, depending on what information is paramount for the application in hand. Each case has a set desired forecasting window, which can go from the next hour up to ten years. The nature and unpredictability of the data, as well as the importance of the forecasts and the goals of the research will define the aim in each use case.

Nowcasting is forecasting of the very near future. For example, in economics, it is used to predict variables like inflation or the GDP of a country for the following month, or week. These forecasts, if sufficiently accurate, can immensely aid in good decision making for the stakeholders.

The models developed in this work are adequate to be used in the context of nowcasting, as they can generate predictions for the following day, in the case of the daily models, and the following month, for the monthly models. These predictions can help stakeholders like tourism company managers to direct their investments for the near future, by using this demand forecast to their advantage.

When it comes to larger forecasting windows, algorithms like ARIMA are very adept and easily implemented to achieve this end. As an example, figure 4.19 shows the next year – 12 step/12 month – predictions for the monthly Portuguese data using a SARIMA model. It contains cones of uncertainty, which aid visualizing the uncertainty level associated to the results. It's noteworthy that, in truth, the model predicted four years, as it only

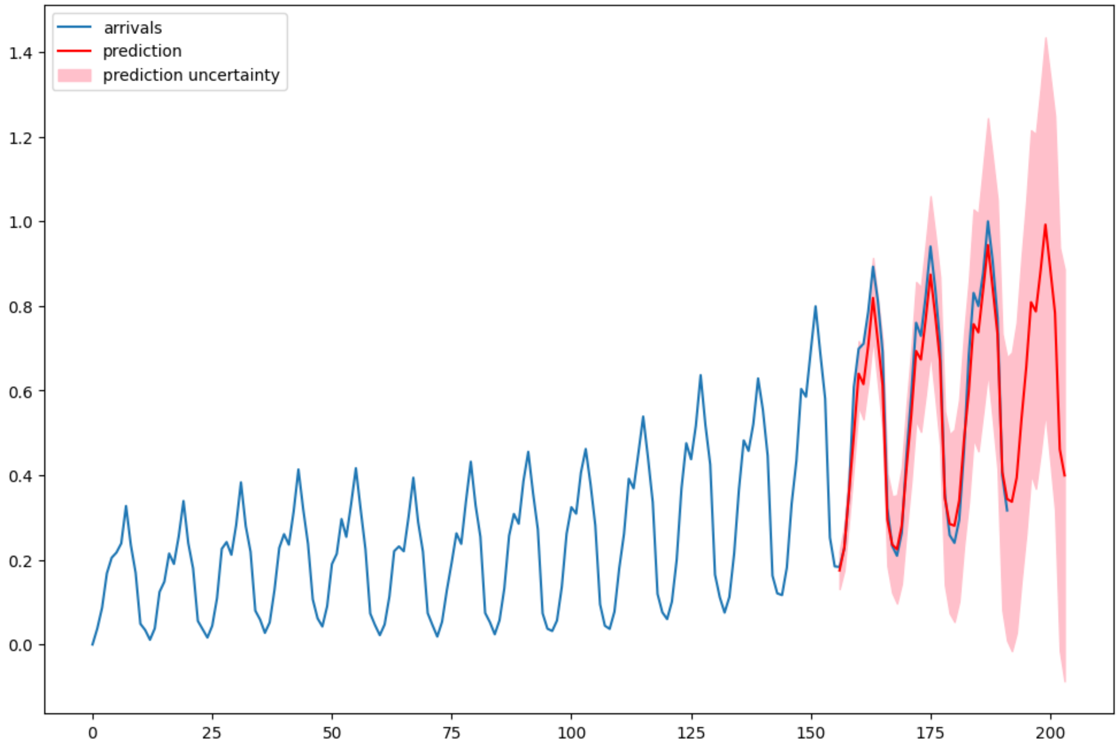


Figure 4.19: Predictions for the following year using SARIMA, with cone of uncertainty.

predicts beyond the training data. A new model could be trained with all the available data, but there would be no way of evaluating it.

In order to generate equivalent predictions using deep learning models, the focus would have to change to multi-step prediction models. Instead of predicting a single step in the future, this type of model can predict multiple consecutive steps simultaneously. This is out of the scope of this work, but it is definitely an interesting avenue to be studied in further research.

4.6 Modeling Summary

In this chapter, the process of modeling for this work was shown.

First, the data collection section displayed the sources and process of collection of data, listing the data that would be available for modeling.

Then, the raw data was treated until it possessed no missing values, was uniformly formatted and had all important features to be used for model training.

With the clean data, both baseline and deep learning models were carefully trained and evaluated, supported by evaluation metrics.

Finally, the results were thoroughly discussed. Conclusions were drawn from the results and hypotheses for the reasoning behind the findings were formulated .

In general, the results produced three important conclusions.

The first is that it is important to use validation and test sets to ascertain the models' over-fitting. However, the models without validation set show that great accuracy can be achieved if they are trained to directly predict the desired data-set.

The second is that, for this context, GRU was the superior model when comparing to every other model in this work, including its deep learning counterpart, LSTM. The great majority of daily baseline models generated results that were not at all satisfactory. However, on such seasonally relevant data as tourism, the deep learning models clearly beating Naïve Seasonal's results proves their superiority to previous season analysis. On the other hand, even with very acceptable results, the monthly baseline models were beat by the deep learning models.

The third is that further research is needed to ascertain whether the picked exogenous variables can be used to enhance the predictions of deep learning models, as, in the context of this work, they were not successful in doing so.

5 Web-App Deployment

5.1 Introduction

As mentioned before, tourism forecasting models are valuable tools for tourism companies, offering the ability to predict demand and guide investments effectively. However, to make these models useful in practice, they need to be deployed on a suitable platform.

In the context of this work, a web application was developed to facilitate the deployment of the tourism demand predictions. The reason for this is that it will serve the needs of Small or Medium-sized Enterprises (SMEs) as part of the European-funded project RESETTING, which aims to assist SMEs in the European tourism industry with their recovery efforts in the wake of the COVID-19 pandemic. Further details about the RESETTING project can be found on their website [1].

5.2 Architecture

The web-app is comprised of three main components: The back-end server, the front-end server and the database. Figure 5.1 shows a deployment diagram that represents the system components. This system is implemented with a decoupled structure, where the back-end directly accesses the database and connects with the front-end via an exposed RESTful API (REST API). The user has access to a web page presented by the front-end server, which gets its data through the REST API endpoints exposed by the back-end. Additionally, the database is fed with results from independent python notebooks that will train and run the models.

Django is used as the back-end server, with Django REST Framework (DRF) [2] as its support. DRF is a toolkit made for Django that handles REST API's, using data structures that have predefined, but configurable, behaviour. Using this framework, it is possible to quickly expose endpoints with easily managed security and data access. These endpoints implement features such as authentication through JSON web tokens with HttpOnly cookies, as well as data fetching and updating.

[1] https://www.resetting.eu/

[2] https://www.django-rest-framework.org/

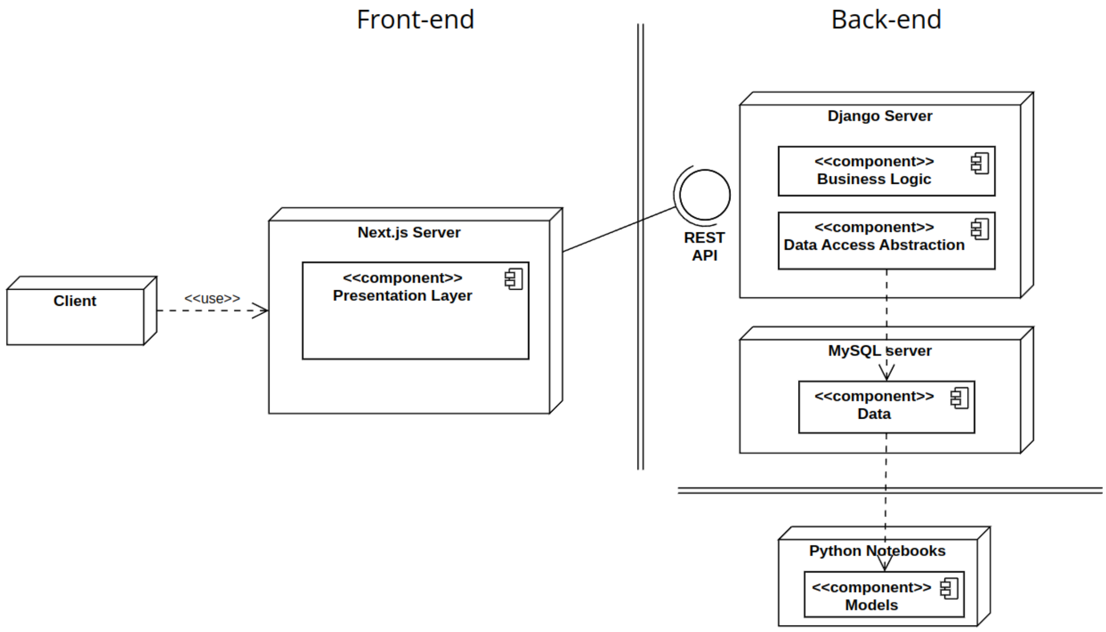


Figure 5.1: Deployment diagram of the web-application.

MySQL is the database linked with the Django server. In it, all data held by the web-application is stored, including time-series data. This can be a bad practice for large time-series, however the number of rows in this work's time-series are in the order of magnitude of hundreds, or low thousands, which will pose no issue to this database. Figure 5.2 shows the created tables. All data in this database is managed entirely and solely by the Django server, including adding, removing and updating records and tables.

The front-end is made with Next.js, a powerful React based framework that is characterized by its responsiveness, fast development process and security. This component was mainly developed by fellow MSc student Nuno Dias, who participates in the RESETTING project. Figure 5.3 shows the index page of the web-app. The forecasting module of the web-app is one of the tools built to be used by the SMEs. Along with the engineering team, a marketing team developed analytical tools that aid the companies with self evaluation. These tools were deployed on the same web-app by the engineering team.

These three components run in separate Docker containers, for ease of deployment.

5.3 Models Deployment

The models' results are shown on the web-app using graphical visualizations. The historical data is plotted on a line graph, along with the predictions and uncertainty cone. Figure 5.4 highlights such approach. The user is able to switch between the different models, that represent each country available, and check the predictions of arrivals for each of them.

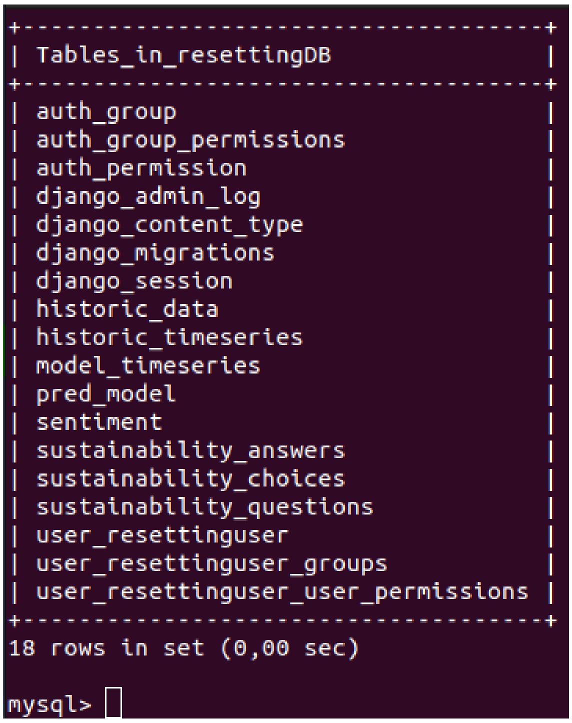


Figure 5.2: MySQL tables for this work.

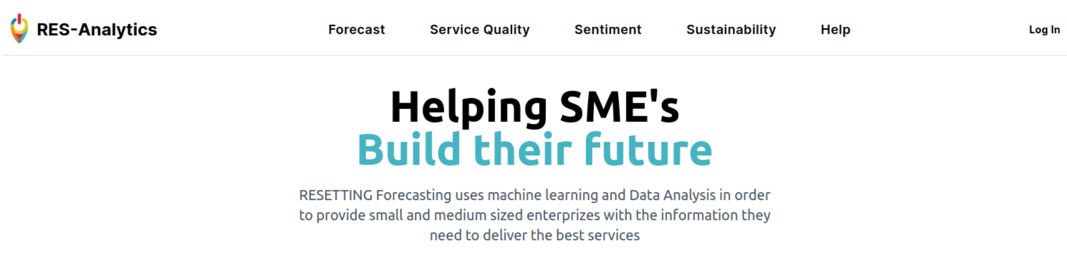


Figure 5.3: Index page of the web-app.

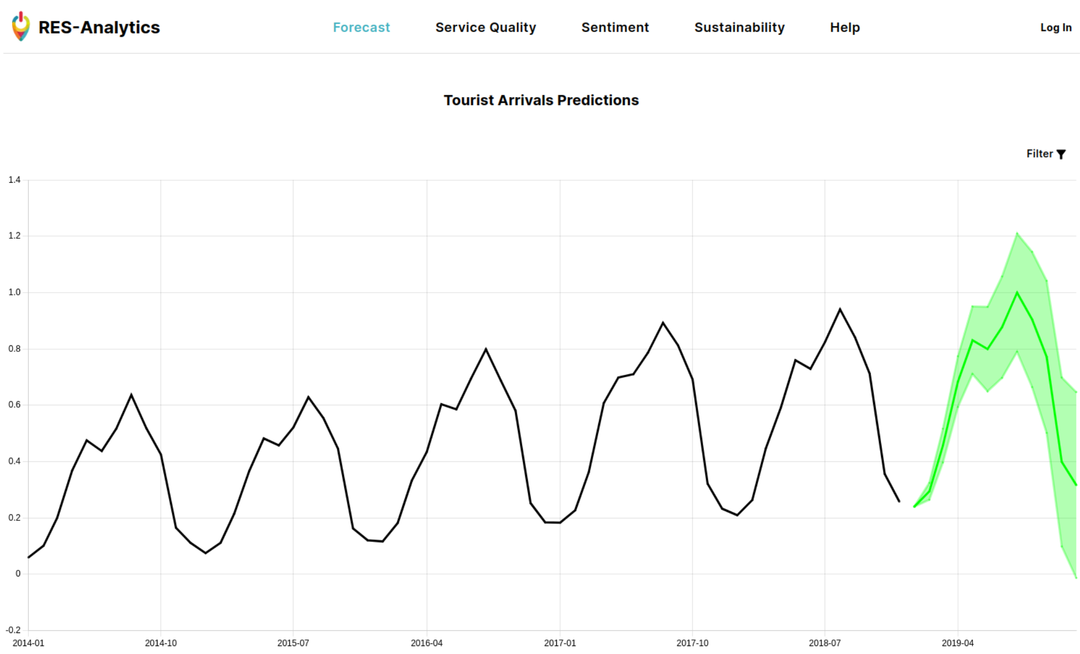


Figure 5.4: Predictions with uncertainty cone.

6 Conclusions and Future Work

6.1 Introduction

During this study, we embarked on a comprehensive journey to forecasting inbound European tourism by employing a diverse range of models and data sources.

This chapter summarizes the results of this research. It will reflect on the path we've traveled, focusing on the central findings, exploring their impact on tourism forecasting, and suggesting a course for future research endeavors.

6.2 Main Findings

Our study employed an array of forecasting models, from baseline, traditional models, like the Naïve models or Exponential Smoothing, to state-of-art, deep learning models, like Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM). The extensive analysis yielded two significant insights.

First, it was observed that, surprisingly, more variables did not equal better results. The models using only arrivals as their feature outperformed others with an array of features with exogenous variables, raising the question of refining them with different methods. Perhaps studying the use of different lags and their correlation with the target would allow enough time to accommodate the effect window of the external variables on the target variable.

Second, GRU outperformed LSTM in a dominant fashion. In both validation and test sets alike, it produced much better (on one case only slightly better) metric scores. A noteworthy observation is the common occurrence of lower validation set metric scores producing higher test set scores, and vice-versa.

6.3 Practical Outreach

The implications of this work extend beyond the academic realm. The output of the forecasting models that have been developed will be used primarily in the context of the

RESETTING project. In that regard, the web application that has been developed, and presented in chapter 5, plays an important role in making those results easily available to the stakeholders of the project. Furthermore, it is expected that the models themselves will be updated overtime, as more data is being collected.

6.4 Limitations and Future Work

The research undertaken opened various possibilities for future exploration, not only to attempt to overcome limitations, but also posing additional questions that emerged from the work carried out. The following sub-sections delve into the different types of limitations and/or possibilities of future research.

6.4.1 Data

The data used in this project was entirely acquired through public sources, with publicly available data. Perhaps using premium, privately owned data would yield different results when it comes to the influence of external variables on the predictions. Furthermore, there would be the possibility of more a choices of data variables, which could relate better to the target. Nonetheless, the proof-of-concept is already implemented. Of course, the more volume and high quality of available data, the better.

6.4.2 Features

When it comes to the features used, all of them were used with the values as is, save for the normalization. This means that all features contained absolute values, except the inflation feature. It is a possibility worth exploring that, by using seasonal adjustment, moving averages or derivatives of these features, the models could produce better results. Furthermore, as some external variables can take time to exert their influence on the target variable, using higher lags for the features could help in finding their time window of effect. This could lead to better aiding in their use for forecasting.

6.4.3 Implementation

The implementation may change the results of the models, as different implementations can have different functions or inner workings. Even if similar in architecture, they can marginally affect the process of modeling. Changing the implementation could mean changing the library used, the optimization and loss functions or the parameters of the models. In this work, the optimization function was the Adam optimizer and the loss function the Mean Square Error (MSE). Changing these could yield different but better results. Another thing that can change the training of the models is adaptive learn rate, as it was static in this work.

6.4.4 Computational Environment

The computational power available was somewhat limited, as the computer used to train all the models was a personal computer, that was at times simultaneously used for model training and other purposes. This could misrepresent the run-time of model training. One possible workaround would be to use cloud solutions like Amazon's AWS or Microsoft's Azure. This would allow the use of larger thresholds for the hyperparameters or more iterations of Bayesian optimization, which would improve the models' performance. Indeed, deep learning models are known for their heavy use of computational resources, and allowing the models to converge to their own limitations would be optimal.

6.4.5 Forecasting with Deep Learning Models

In chapter 4, section 4.5.3 mentions nowcasting, and how the models developed in this work are viable for this type of forecasting. It also shows the use of Seasonal Auto-Regressive Integrated Moving Average (SARIMA) for larger forecasting windows, as this algorithm can be used recursively using its output as the following input. Deep learning models can achieve similar forecasting windows if trained for multi-step predictions. It is of great interest to use the findings of this research to create multi-step deep learning models that can predict further into the future.

6.5 Conclusion

In conclusion, this research contributes to the ongoing efforts to understand and forecast inbound European tourism. As the tourism industry continues to evolve, adapt, and recover, the quest for more accurate, insightful, and practical forecasting models remains a valuable pursuit. Our work paves the way for further studies and innovations in this dynamic field, as we strive to improve humanity's knowledge of the future unknown.

Bibliography

[1] W. T. T. C. (WTTC). *Travel Tourism Economic Impact*. URL: https://wttc.org/research/economic-impact. (accessed: 26.10.2022) (cit. on p. 1).

[2] W. H. Organization. *Coronavirus disease (COVID-19) pandemic*. URL: https://www.who.int/europe/emergencies/situations/covid-19. (accessed: 26.10.2022) (cit. on p. 1).

[3] E. T. Commission. "MONITORING SENTIMENT FOR DOMESTIC AND INTRA-EUROPEAN TRAVEL". In: (2022) (cit. on p. 1).

[4] V. I. Kontopoulou et al. "A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks". In: *Future Internet* 15.8 (2023), p. 255 (cit. on p. 2).

[5] I. Ghalehkhondabi et al. "A review of demand forecasting models and methodological developments within tourism and passenger transportation industry". In: *Journal of Tourism Futures* (2019) (cit. on p. 5).

[6] D. Brand. "Travel demand forecasting: some foundations and a review". In: *Highway Research Board Special Report* 143 (1973) (cit. on p. 5).

[7] S. F. Witt and C. A. Witt. "Forecasting tourism demand: A review of empirical research". In: *International Journal of forecasting* 11.3 (1995), pp. 447–475 (cit. on p. 5).

[8] H. Song and G. Li. "Tourism demand modelling and forecasting—A review of recent research". In: *Tourism management* 29.2 (2008), pp. 203–220 (cit. on p. 5).

[9] H. Song, S. F. Witt, and T. C. Jensen. "Tourism forecasting: accuracy of alternative econometric models". In: *International Journal of Forecasting* 19.1 (2003), pp. 123–141. ISSN: 0169-2070. DOI: https://doi.org/10.1016/S0169-2070(01)00134-0. URL: https://www.sciencedirect.com/science/article/pii/S0169207001001340 (cit. on p. 6).

[10] G. Li et al. “Tourism demand forecasting: A time varying parameter error correction model”. In: *Journal of Travel Research* 45.2 (2006), pp. 175–185 (cit. on p. 6).

[11] U. Gunter and I. Önder. “Forecasting international city tourism demand for Paris: Accuracy of uni-and multivariate models employing monthly data”. In: *Tourism management* 46 (2015), pp. 123–135 (cit. on p. 8).

[12] K. K. Wong et al. “Tourism forecasting: to combine or not to combine?” In: *Tourism management* 28.4 (2007), pp. 1068–1078 (cit. on p. 8).

[13] G. J. George Box. *Time Series Analysis Forecasting and Control.* JSTOR, 1971 (cit. on p. 8).

[14] E. A. A. Ismail. “Forecasting the number of Arab and foreign tourists in Egypt using ARIMA models”. In: *International Journal of System Assurance Engineering and Management* 11.2 (2020), pp. 450–454 (cit. on p. 8).

[15] D. Julong et al. “Introduction to grey system theory”. In: *The Journal of grey system* 1.1 (1989), pp. 1–24 (cit. on p. 9).

[16] X. Ma. “Tourism Demand Forecasting Based on Grey Model and BP Neural Network”. In: *Complexity* 2021 (2021) (cit. on p. 9).

[17] Y.-C. Hu, P. Jiang, and P.-C. Lee. “Forecasting tourism demand by incorporating neural networks into Grey–Markov models”. In: *Journal of the Operational Research Society* 70.1 (2019), pp. 12–20 (cit. on p. 9).

[18] W. S. McCulloch and W. Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133 (cit. on p. 9).

[19] O. I. Abiodun et al. “State-of-the-art in artificial neural network applications: A survey”. In: *Heliyon* 4.11 (2018), e00938 (cit. on p. 10).

[20] X. Huang et al. “Tourist hot spots prediction model based on optimized neural network algorithm”. In: *International Journal of System Assurance Engineering and Management* 13.1 (2022), pp. 63–71 (cit. on p. 10).

[21] X. Shi. “Tourism culture and demand forecasting based on BP neural network mining algorithms”. In: *Personal and Ubiquitous Computing* 24.2 (2020), pp. 299–308 (cit. on p. 10).

[22] L. Wang. “Tourism Demand Forecast Based on Adaptive Neural Network Technology in Business Intelligence”. In: *Computational Intelligence and Neuroscience* 2022 (2022) (cit. on p. 10).

[23] A. Kulshrestha, V. Krishnaswamy, and M. Sharma. “Bayesian BILSTM approach for tourism demand forecasting”. In: *Annals of tourism research* 83 (2020), p. 102925 (cit. on pp. 10, 16).

[24] S.-C. Hsieh. "Tourism demand forecasting based on an LSTM network and its variants". In: *Algorithms* 14.8 (2021), p. 243 (cit. on p. 11).

[25] C. Cortes and V. Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297 (cit. on p. 11).

[26] B. E. Boser, I. M. Guyon, and V. N. Vapnik. "A training algorithm for optimal margin classifiers". In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 144–152 (cit. on p. 11).

[27] N. Nawaz. "Machine Learning based Forecasting Systems for Worldwide International Tourists Arrival". In: *Available at SSRN* (2021) (cit. on p. 11).

[28] T. K. Ho. "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282 (cit. on p. 11).

[29] E. Kirkos. "Airbnb listings' performance: Determinants and predictive models". In: *European Journal of Tourism Research* 30 (2022), pp. 3012–3012 (cit. on p. 11).

[30] P. I. Frazier. "A tutorial on Bayesian optimization". In: *arXiv preprint arXiv:1807.02811* (2018) (cit. on p. 16).

[31] S. Hochreiter and J. Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780 (cit. on p. 17).

[32] K. Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014) (cit. on p. 18).

[33] J. Chung et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014) (cit. on p. 18).

[34] K. Zarzycki and M. Ławryńczuk. "Advanced predictive control for GRU and LSTM networks". In: *Information Sciences* 616 (2022), pp. 229–254 (cit. on p. 18).

[35] L. Wen, C. Liu, and H. Song. "Forecasting tourism demand using search query data: A hybrid modelling approach". In: *Tourism Economics* 25.3 (2019), pp. 309–329 (cit. on p. 24).

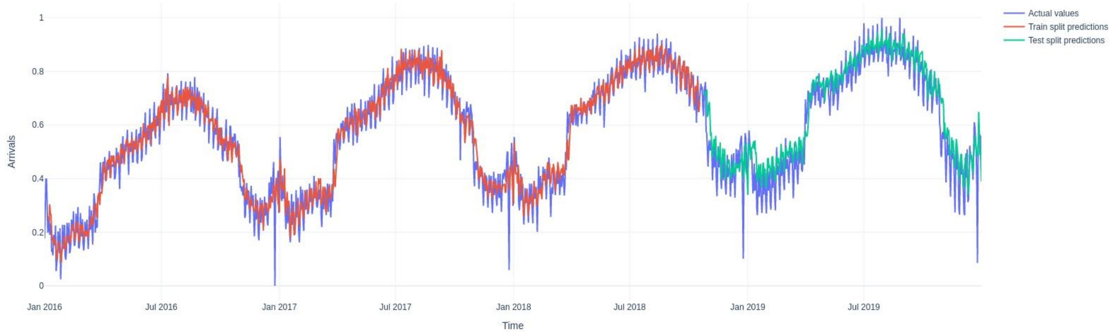
A

Appendix

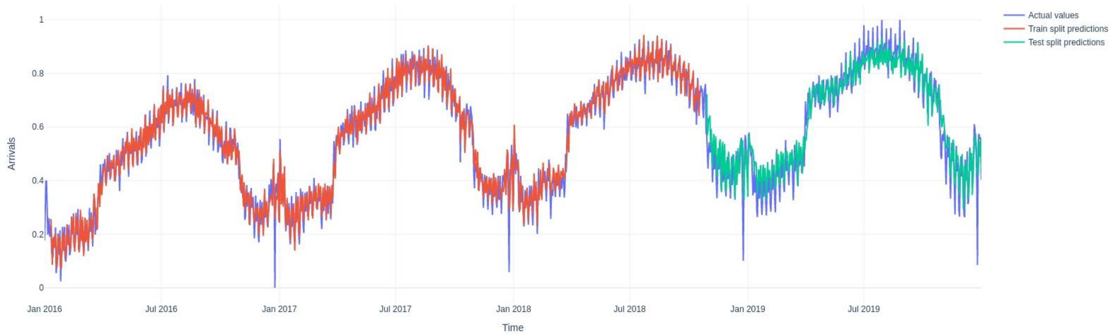
In this appendix chapter, all charts generated by each of the models will be presented.

A.1 Daily Models Results

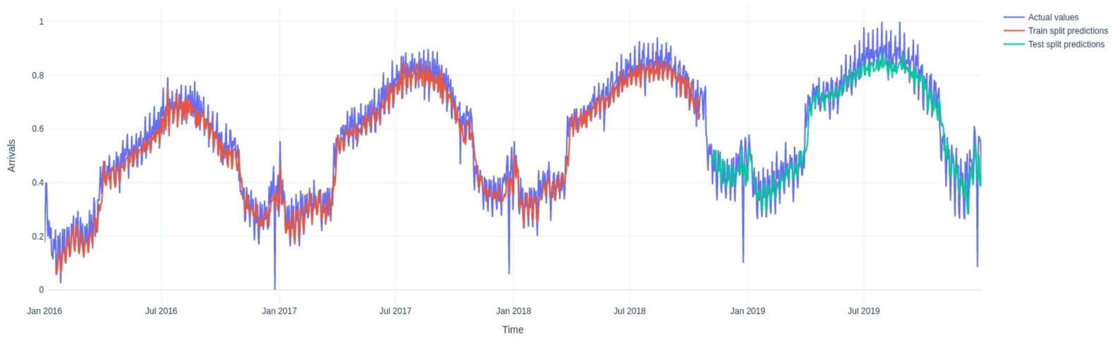
A.1.1 Portugal



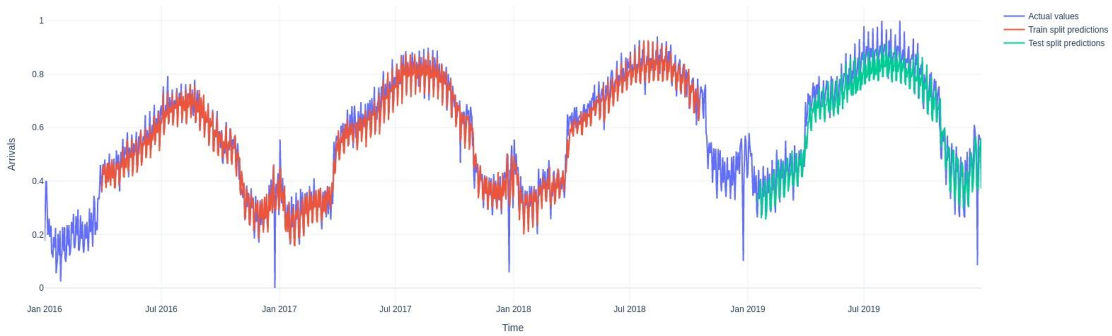
Results: Chart generated with the results of the Daily LSTM model with All_features as features for Portugal.



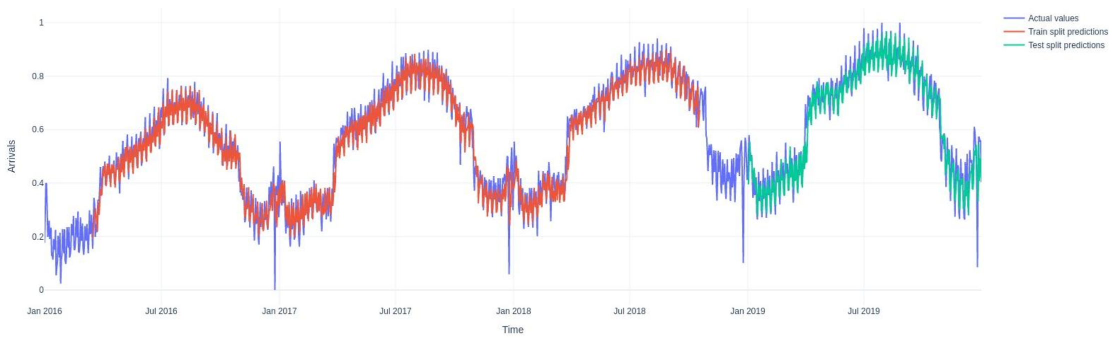
Results: Chart generated with the results of the Daily GRU model with All_features as features for Portugal.



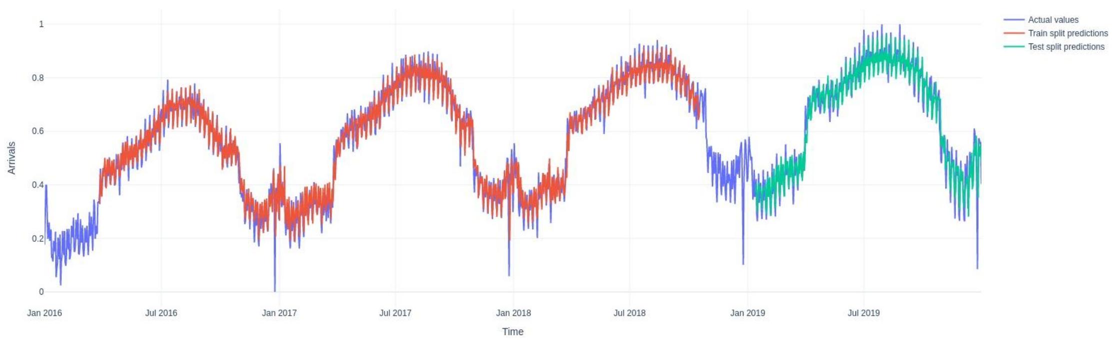
Results: Chart generated with the results of the Daily LSTM model with Trends_arrivals as features for Portugal.



Results: Chart generated with the results of the Daily GRU model with Trends_arrivals as features for Portugal.

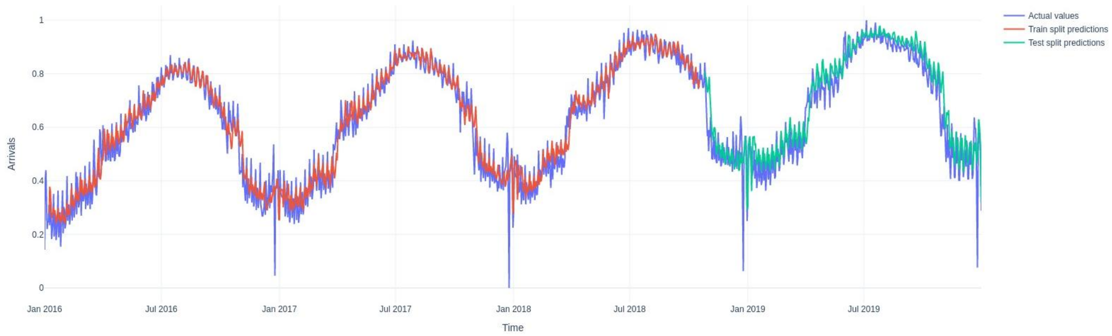


Results: Chart generated with the results of the Daily LSTM model with Arrivals as features for Portugal.

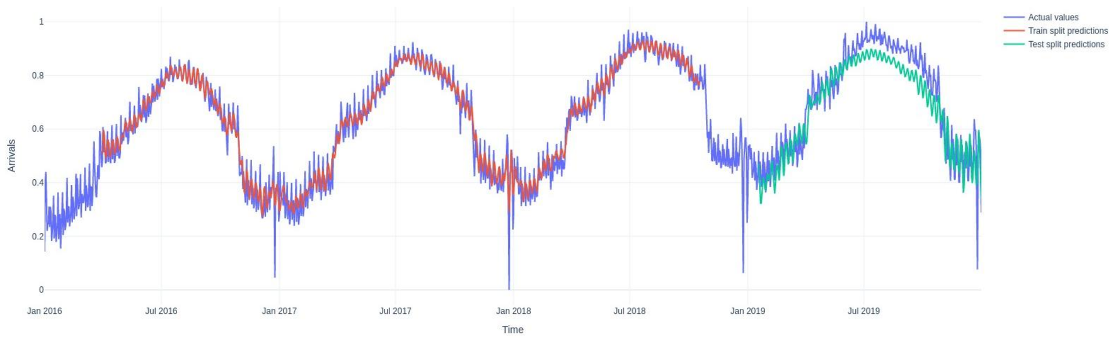


Results: Chart generated with the results of the Daily GRU model with Arrivals as features for Portugal.

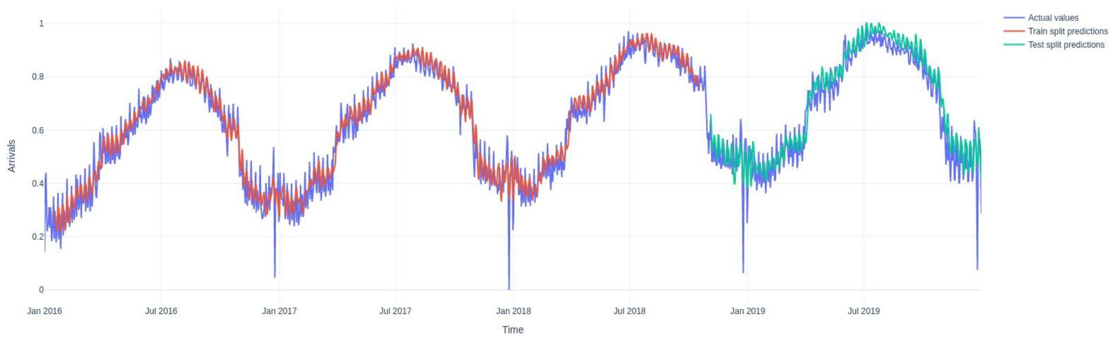
A.1.2 Spain



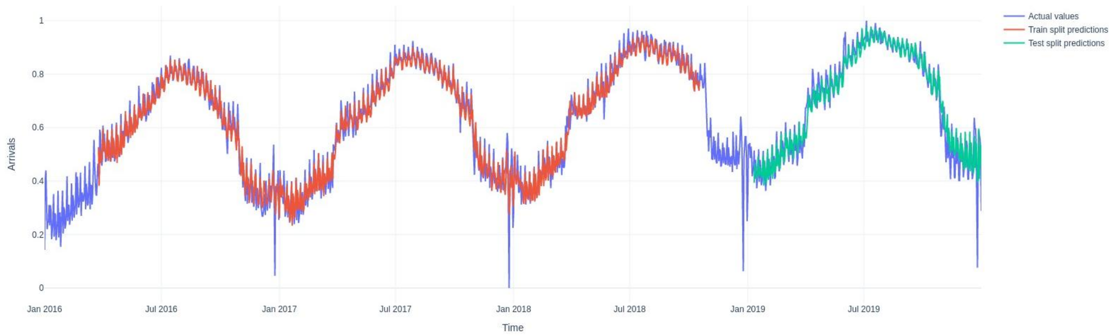
Results: Chart generated with the results of the Daily LSTM model with All_features as features for Spain.



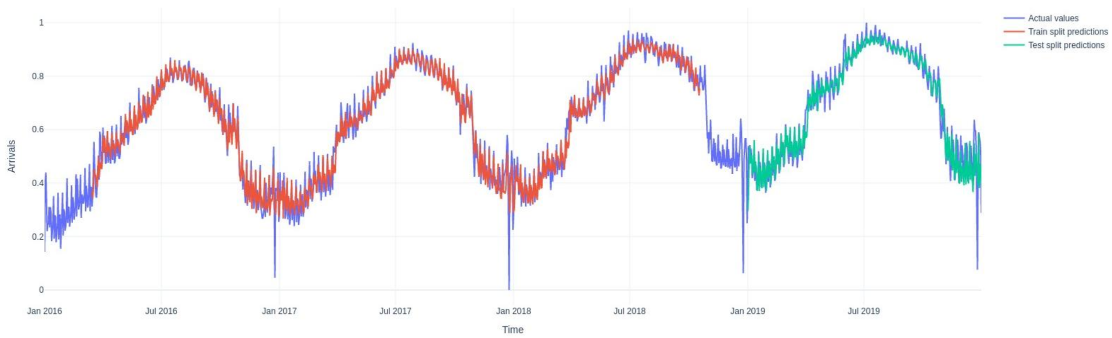
Results: Chart generated with the results of the Daily GRU model with All_features as features for Spain.



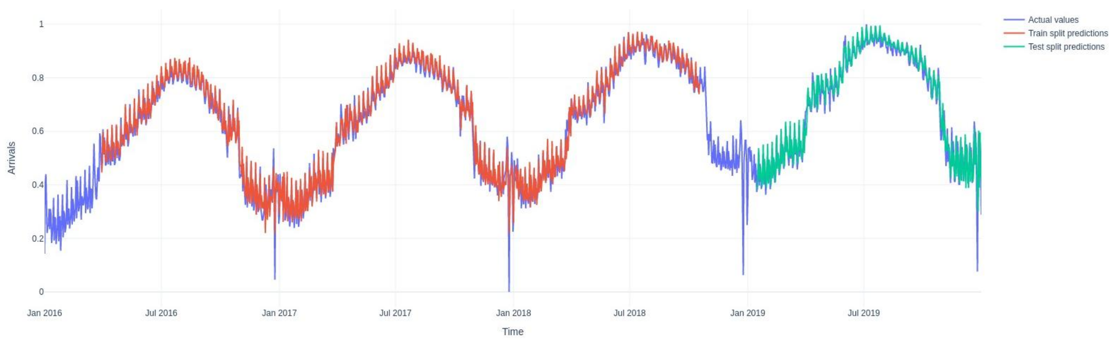
Results: Chart generated with the results of the Daily LSTM model with Trends_arrivals as features for Spain.



Results: Chart generated with the results of the Daily GRU model with Trends_arrivals as features for Spain.

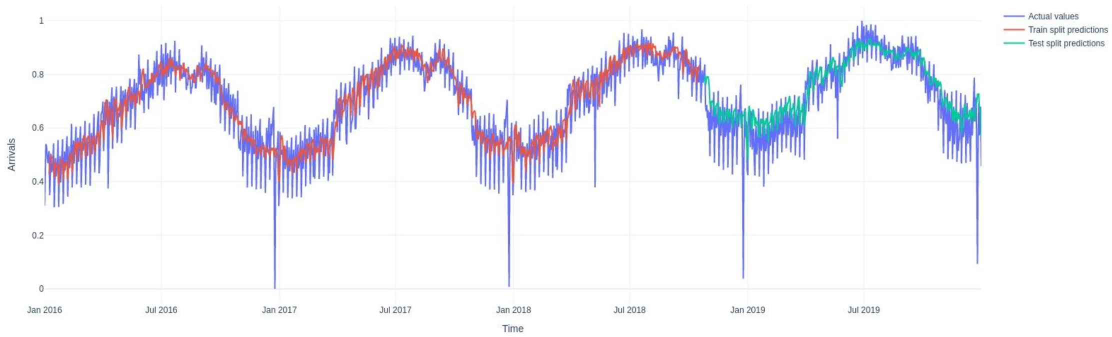


Results: Chart generated with the results of the Daily LSTM model with Arrivals as features for Spain.

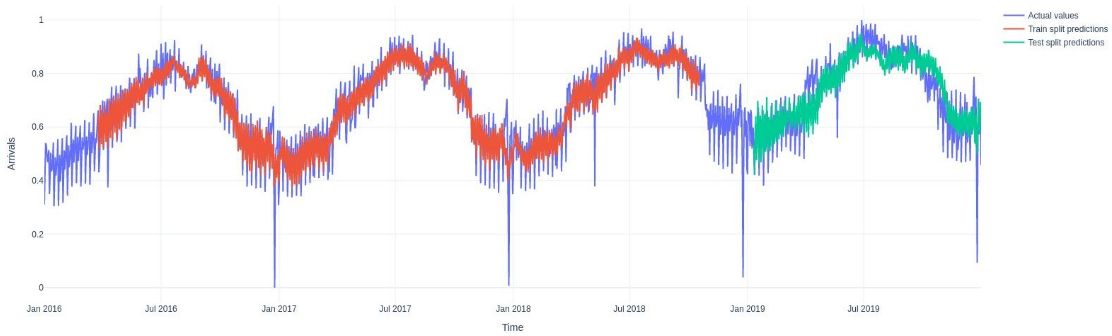


Results: Chart generated with the results of the Daily GRU model with Arrivals as features for Spain.

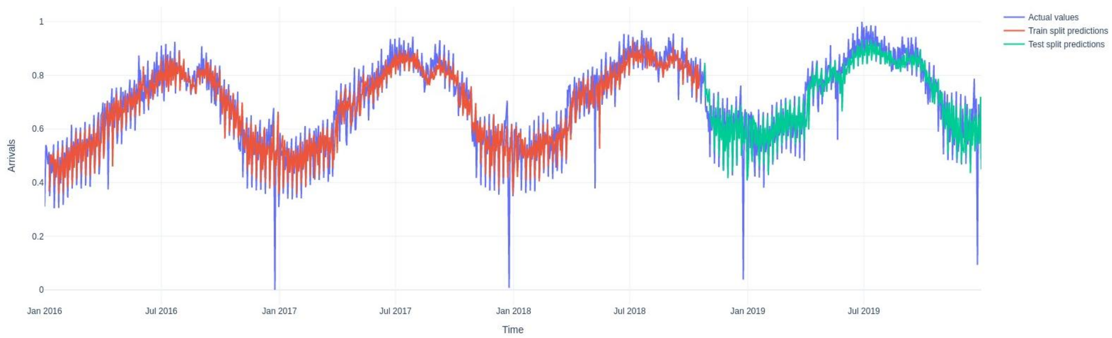
A.1.3 Italy



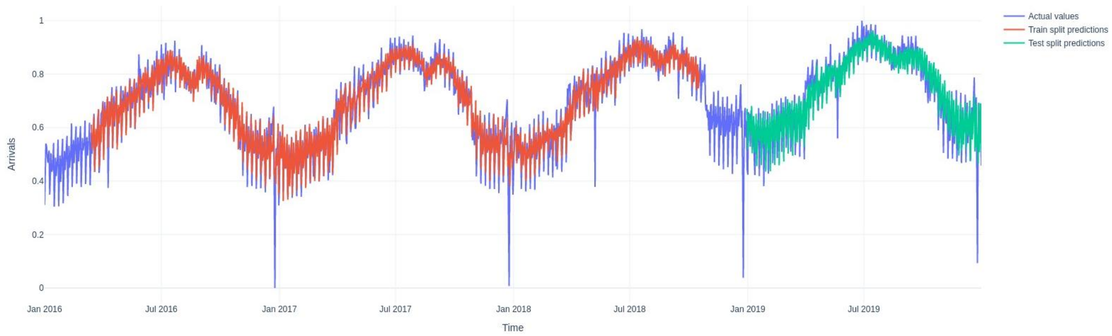
Results: Chart generated with the results of the Daily LSTM model with All_features as features for Italy.



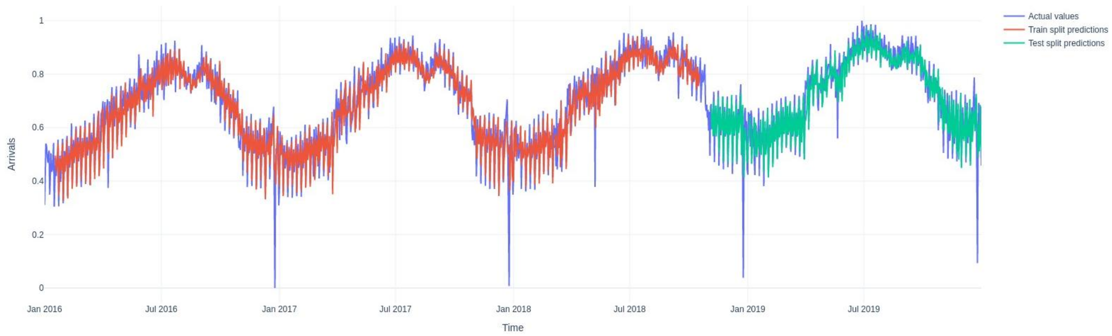
Results: Chart generated with the results of the Daily GRU model with All_features as features for Italy.



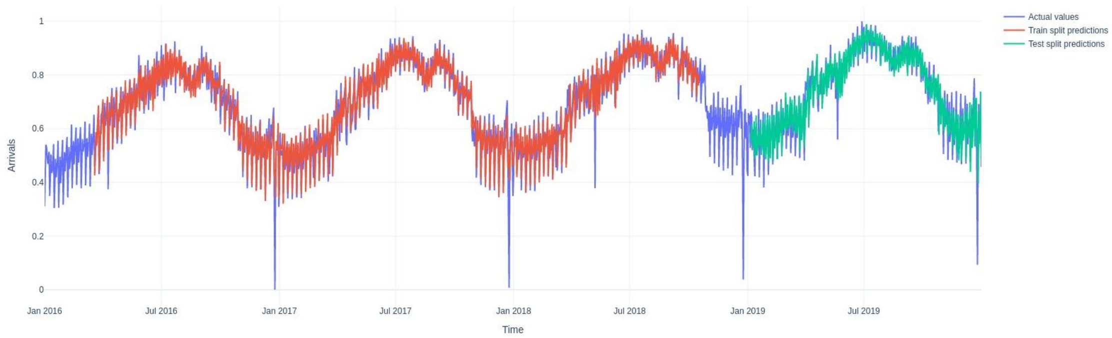
Results: Chart generated with the results of the Daily LSTM model with Trends_arrivals as features for Italy.



Results: Chart generated with the results of the Daily GRU model with Trends_arrivals as features for Italy.

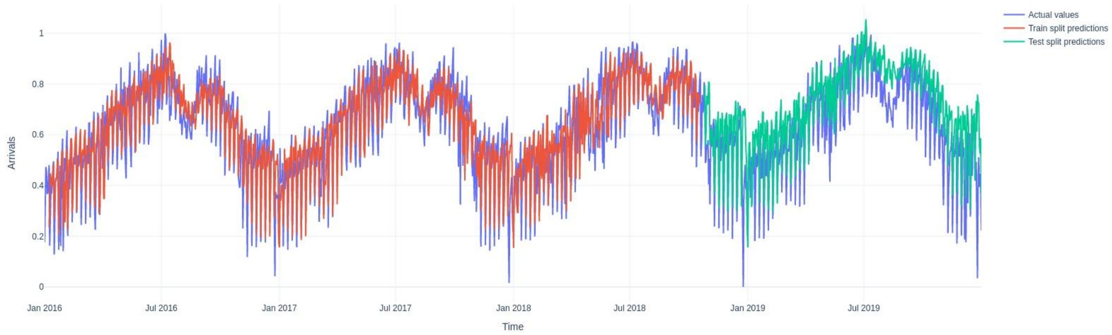


Results: Chart generated with the results of the Daily LSTM model with Arrivals as features for Italy.

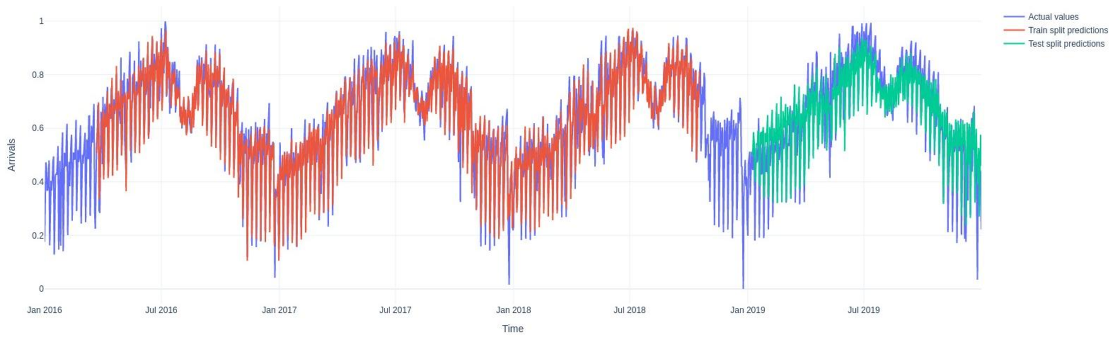


Results: Chart generated with the results of the Daily GRU model with Arrivals as features for Italy.

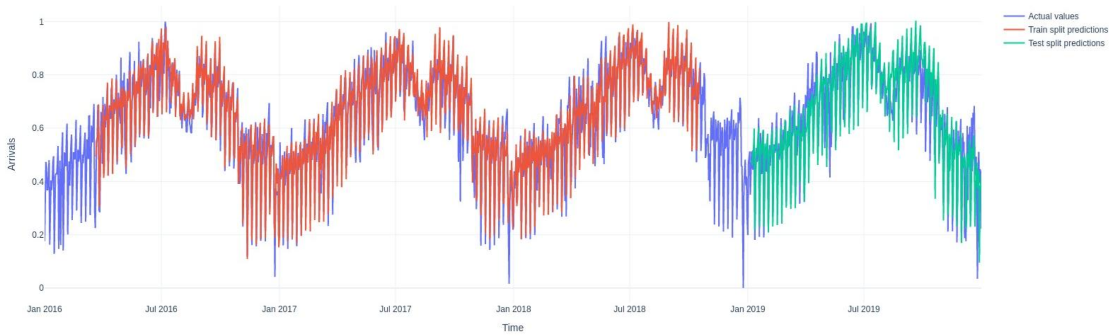
A.1.4 France



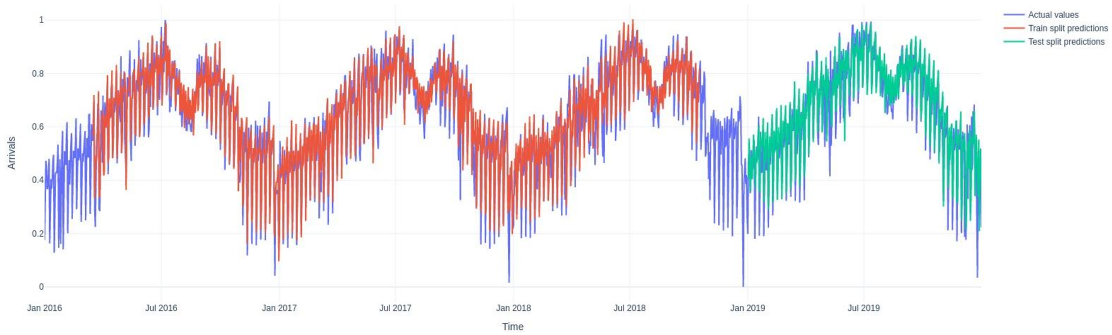
Results: Chart generated with the results of the Daily LSTM model with All_features as features for France.



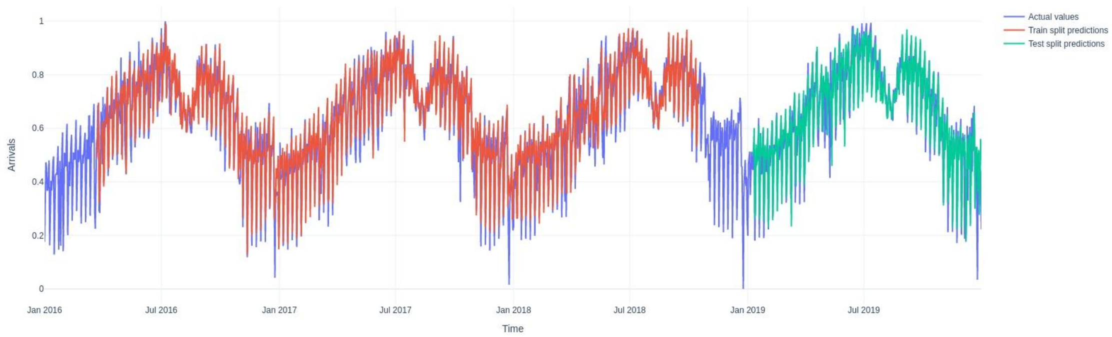
Results: Chart generated with the results of the Daily GRU model with All_features as features for France.



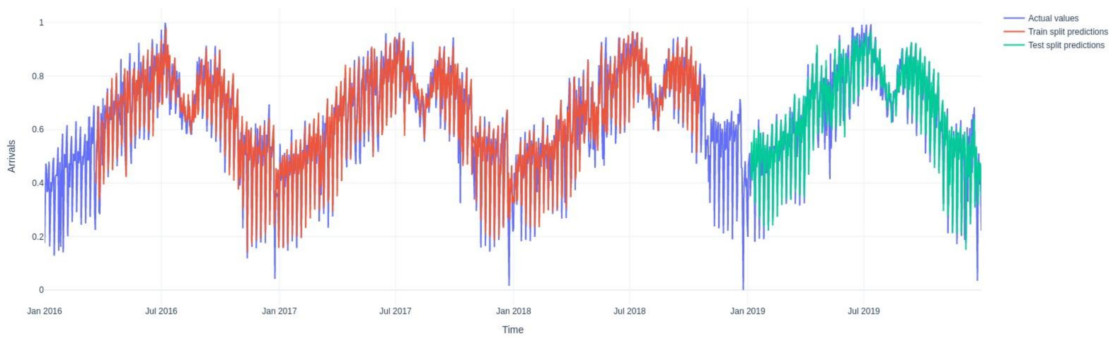
Results: Chart generated with the results of the Daily LSTM model with Trends_arrivals as features for France.



Results: Chart generated with the results of the Daily GRU model with Trends_arrivals as features for France.

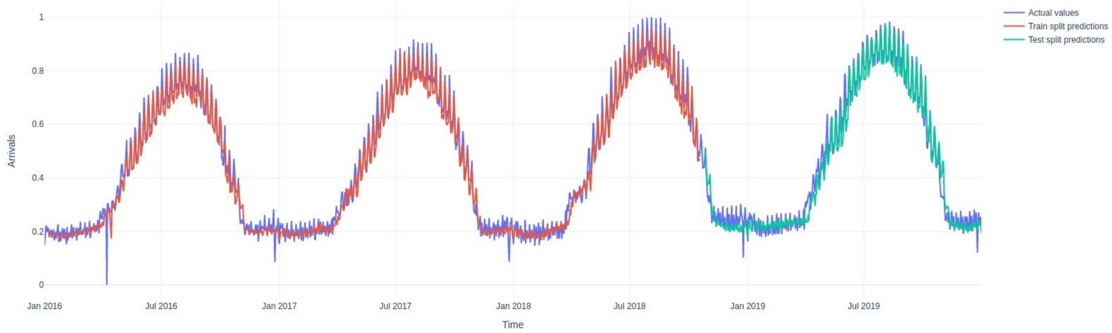


Results: Chart generated with the results of the Daily LSTM model with Arrivals as features for France.

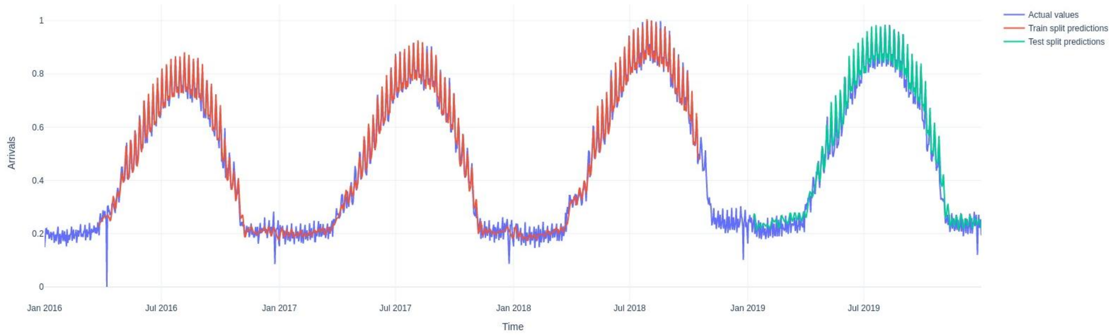


Results: Chart generated with the results of the Daily GRU model with Arrivals as features for France.

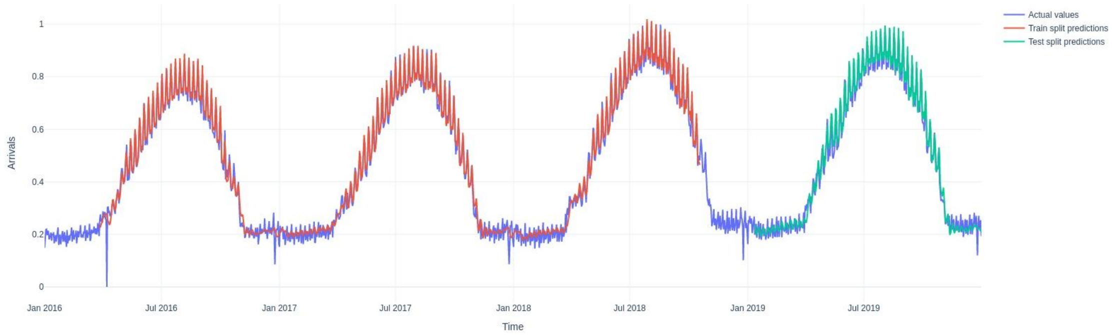
A.1.5 Greece



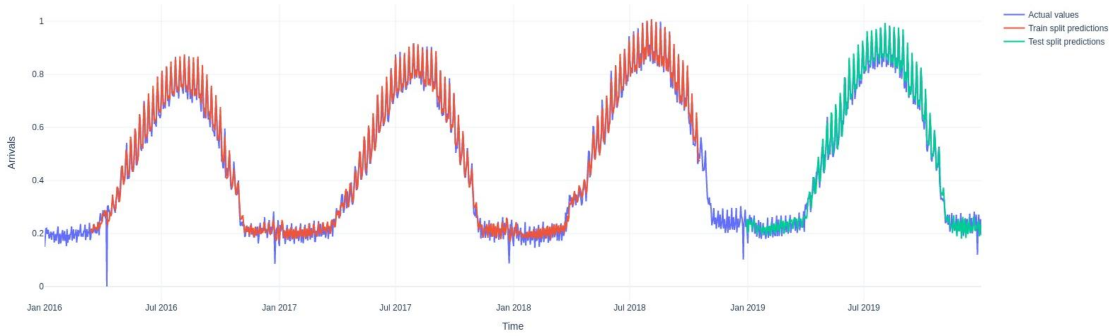
Results: Chart generated with the results of the Daily LSTM model with All_features as features for Greece.



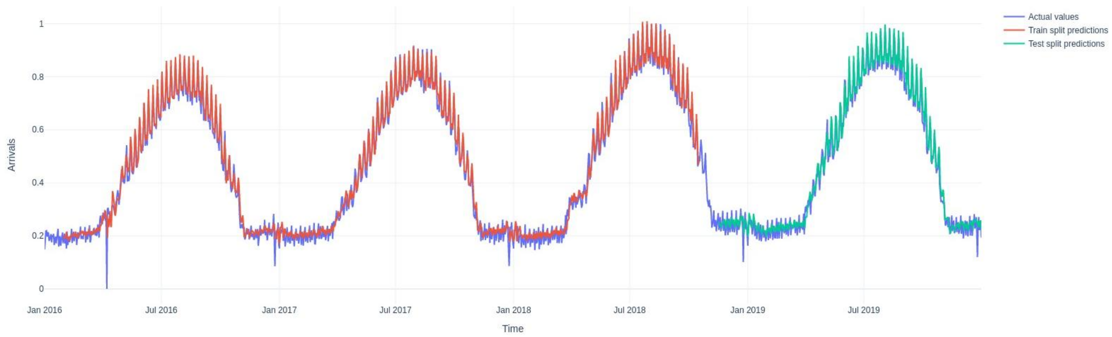
Results: Chart generated with the results of the Daily GRU model with All_features as features for Greece.



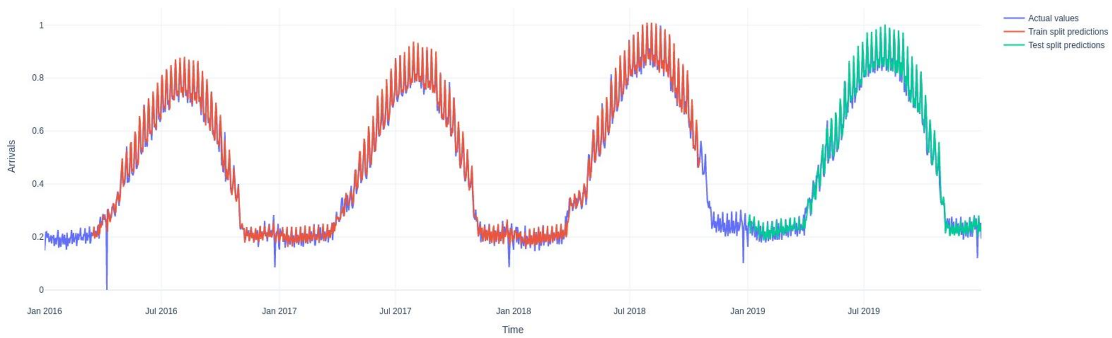
Results: Chart generated with the results of the Daily LSTM model with Trends_arrivals as features for Greece.



Results: Chart generated with the results of the Daily GRU model with Trends_arrivals as features for Greece.



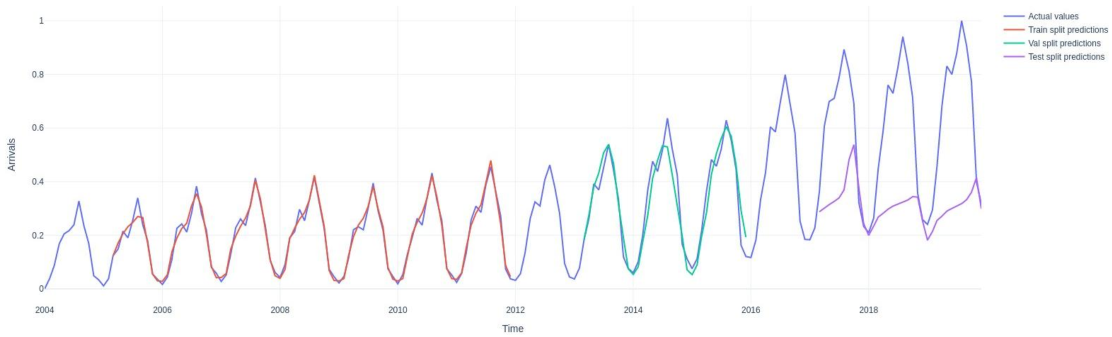
Results: Chart generated with the results of the Daily LSTM model with Arrivals as features for Greece.



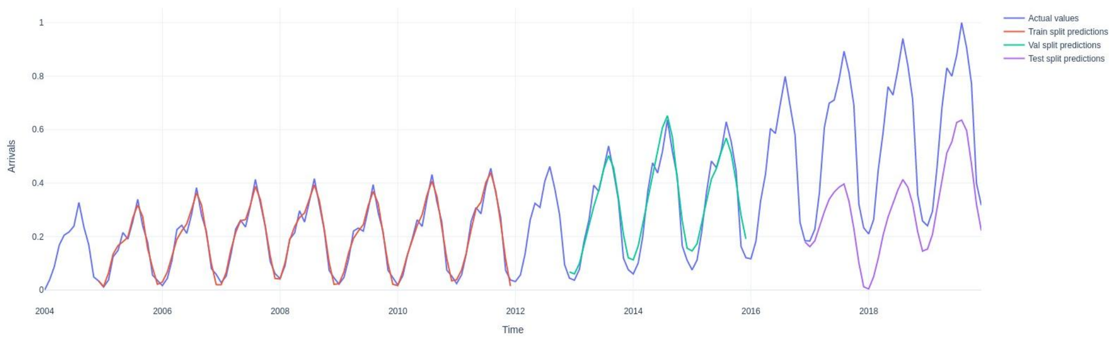
Results: Chart generated with the results of the Daily GRU model with Arrivals as features for Greece.

A.2 BVM Monthly Models Results

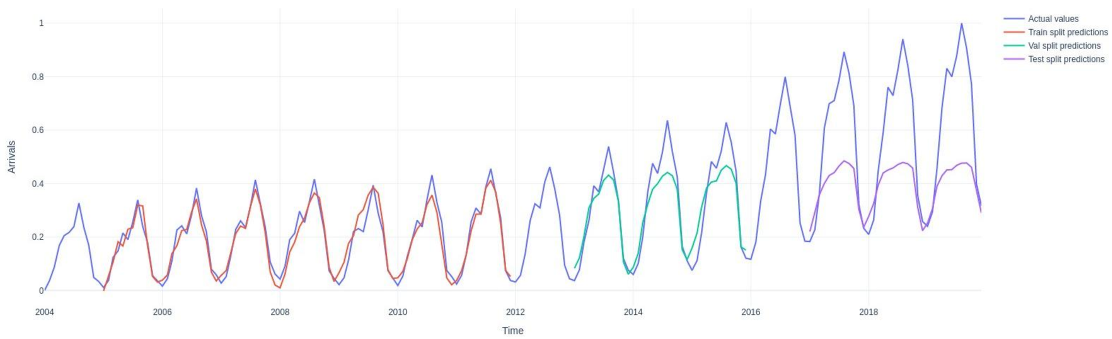
A.2.1 Portugal



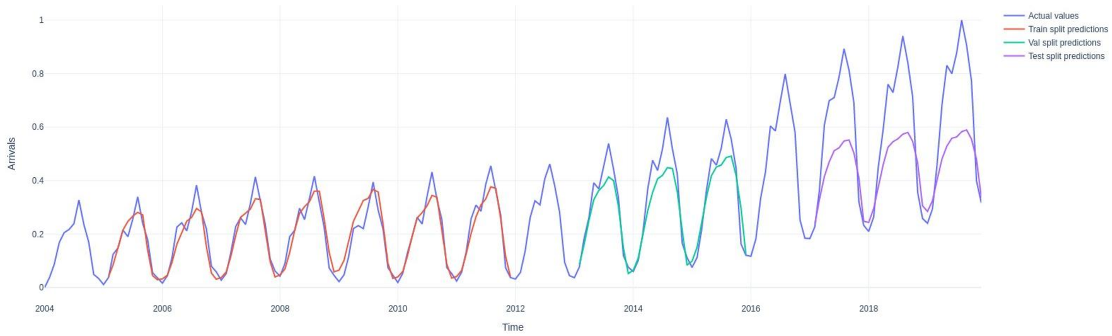
Results: Chart generated with the results of the Best Validation-set Metrics (BVM)_Monthly LSTM model with All_features as features for Portugal.



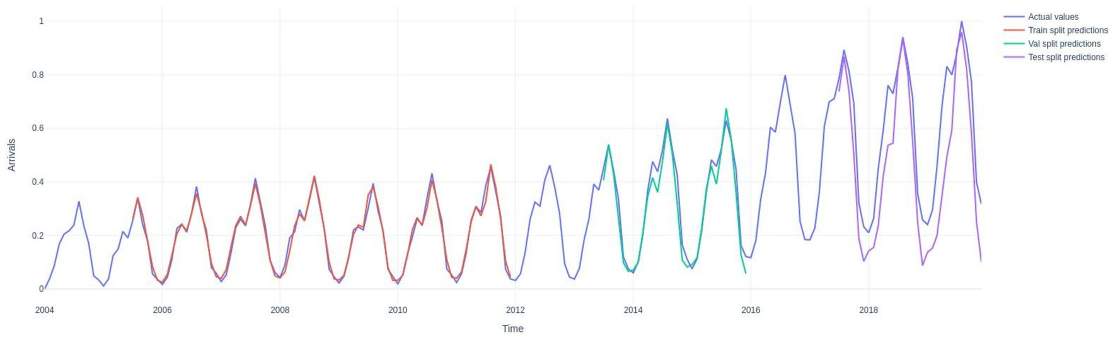
Results: Chart generated with the results of the BVM_Monthly GRU model with All_features as features for Portugal.



Results: Chart generated with the results of the BVM_Monthly GRU model with Trends_arrivals as features for Portugal.

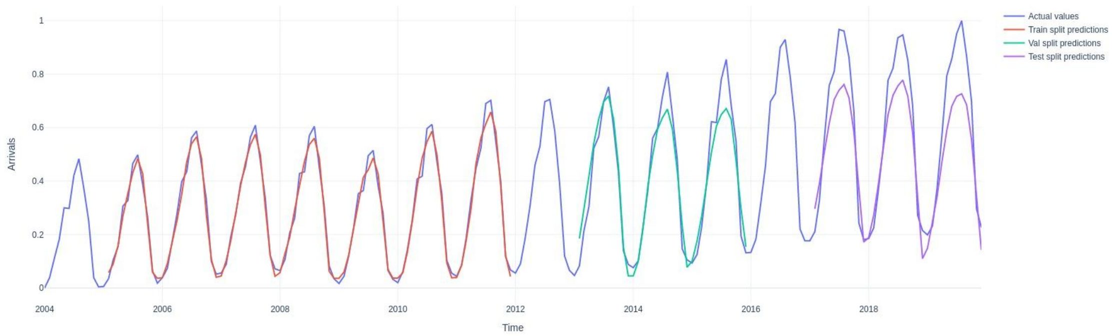


Results: Chart generated with the results of the BVM_Monthly LSTM model with Arrivals as features for Portugal.

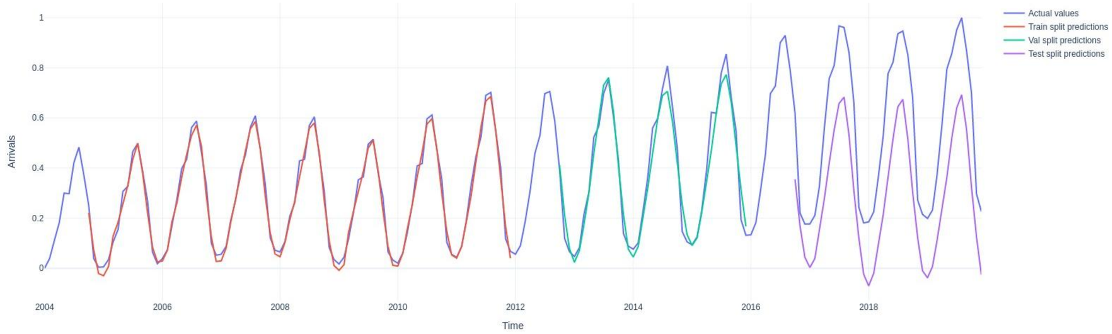


Results: Chart generated with the results of the BVM_Monthly GRU model with Arrivals as features for Portugal.

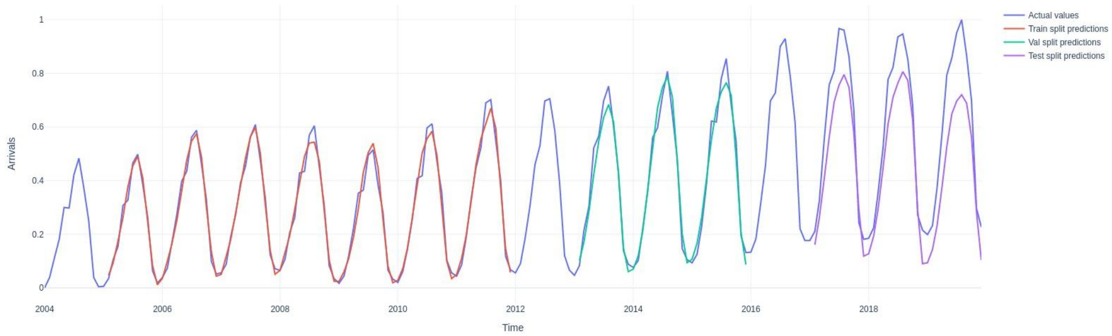
A.2.2 Spain



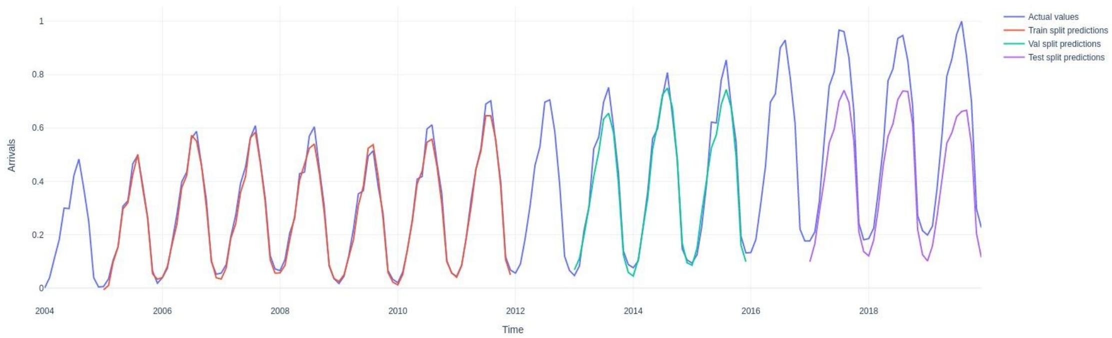
Results: Chart generated with the results of the BVM_Monthly LSTM model with All_features as features for Spain.



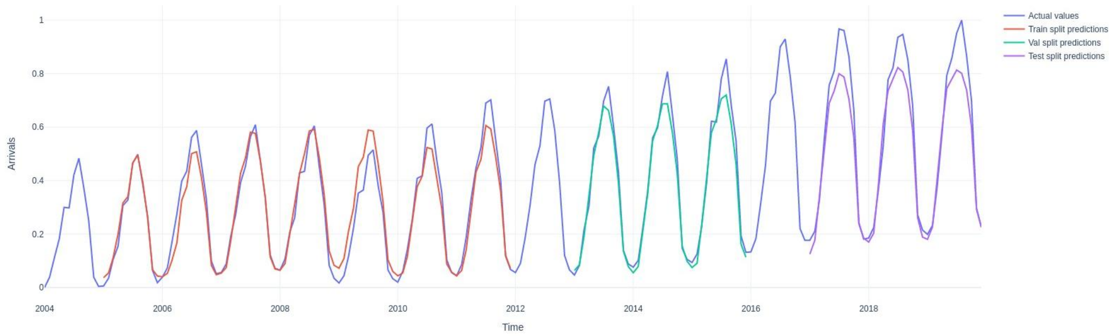
Results: Chart generated with the results of the BVM_Monthly GRU model with All_features as features for Spain.



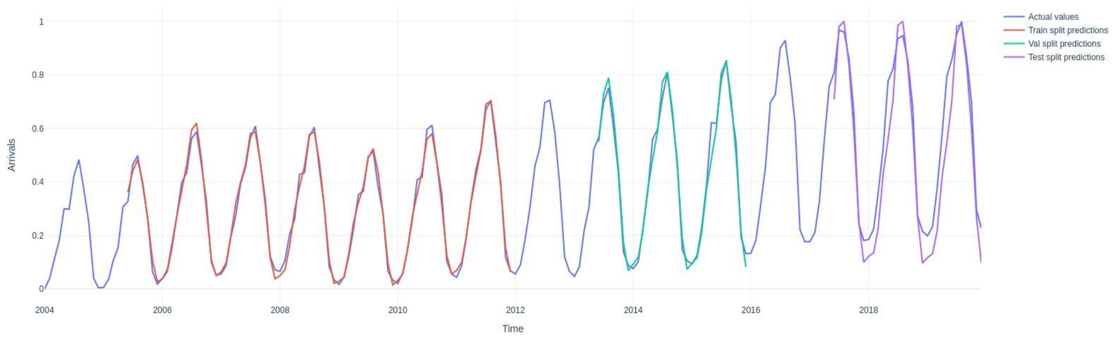
Results: Chart generated with the results of the BVM_Monthly LSTM model with Trends_arrivals as features for Spain.



Results: Chart generated with the results of the BVM_Monthly GRU model with Trends_arrivals as features for Spain.

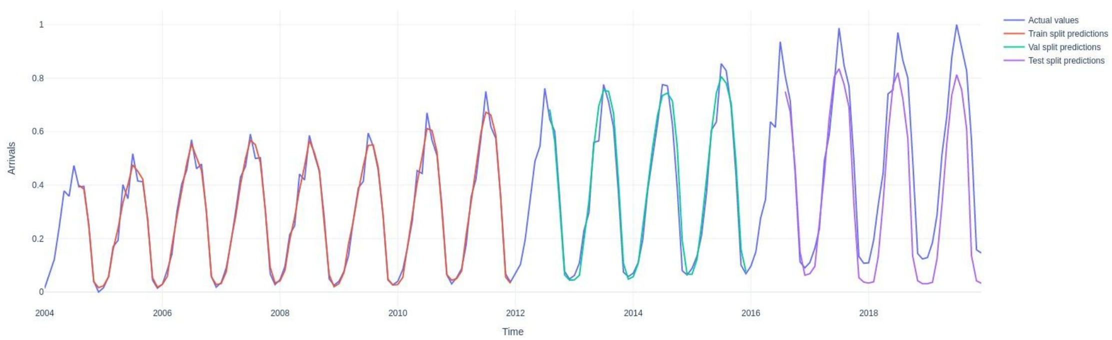


Results: Chart generated with the results of the BVM_Monthly LSTM model with Arrivals as features for Spain.

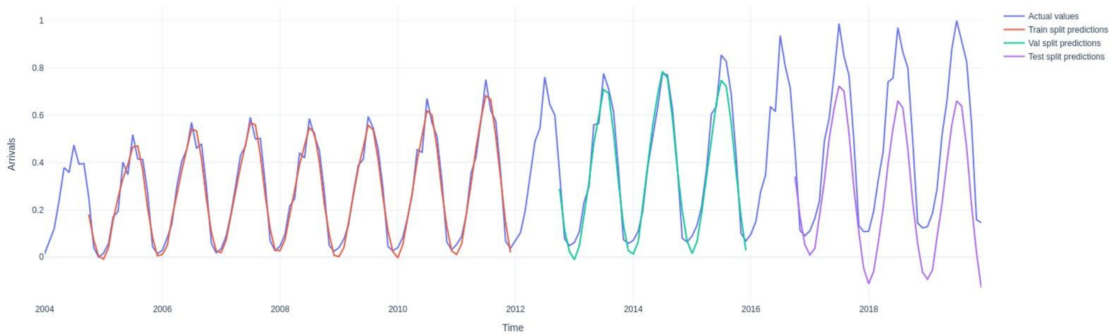


Results: Chart generated with the results of the BVM_Monthly GRU model with Arrivals as features for Spain.

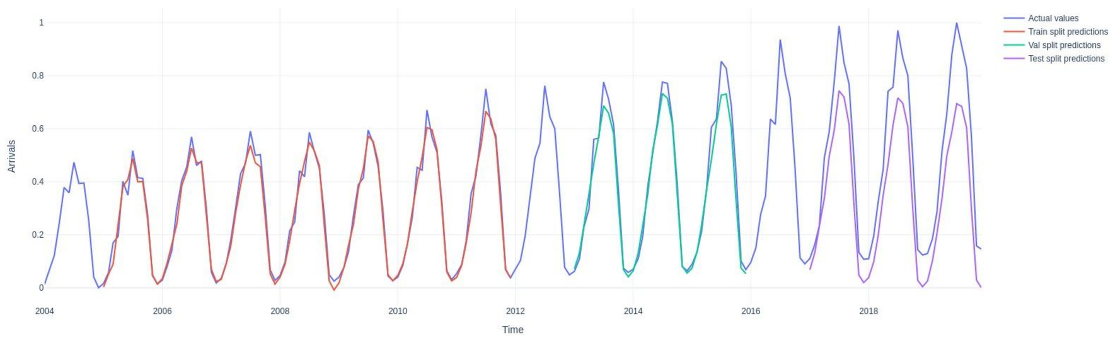
A.2.3 Italy



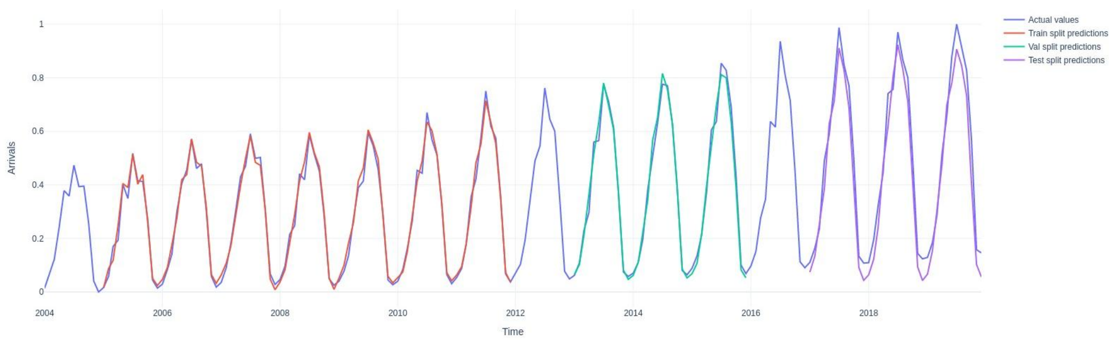
Results: Chart generated with the results of the BVM_Monthly LSTM model with All_features as features for Italy.



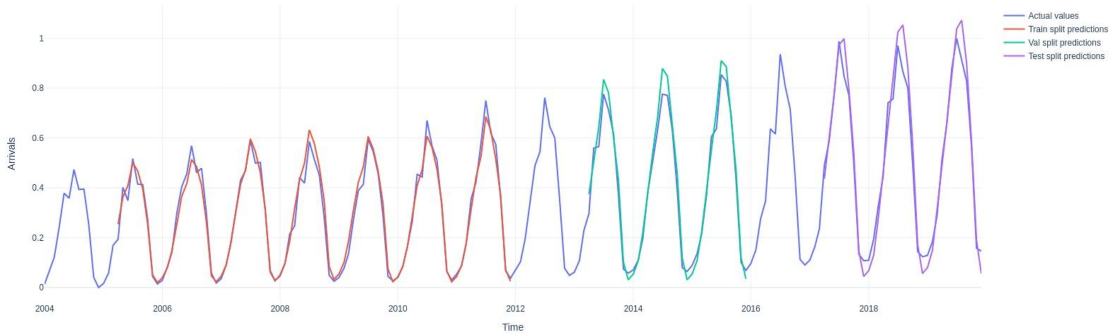
Results: Chart generated with the results of the BVM_Monthly GRU model with All_features as features for Italy.



Results: Chart generated with the results of the BVM_Monthly LSTM model with Trends_arrivals as features for Italy.

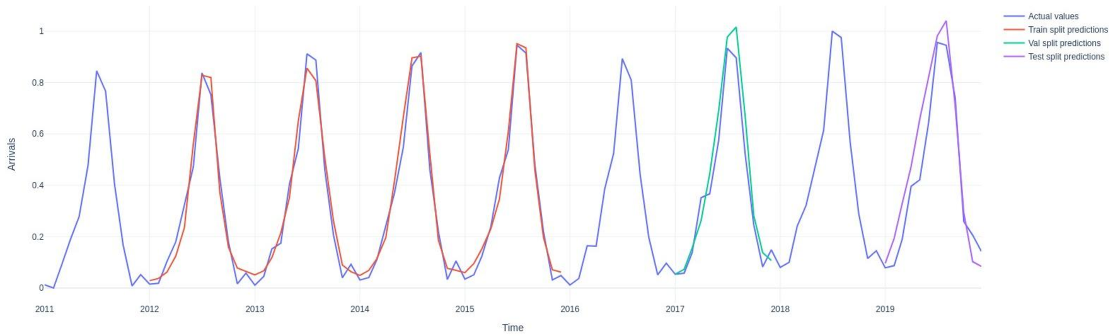


Results: Chart generated with the results of the BVM_Monthly GRU model with Trends_arrivals as features for Italy.

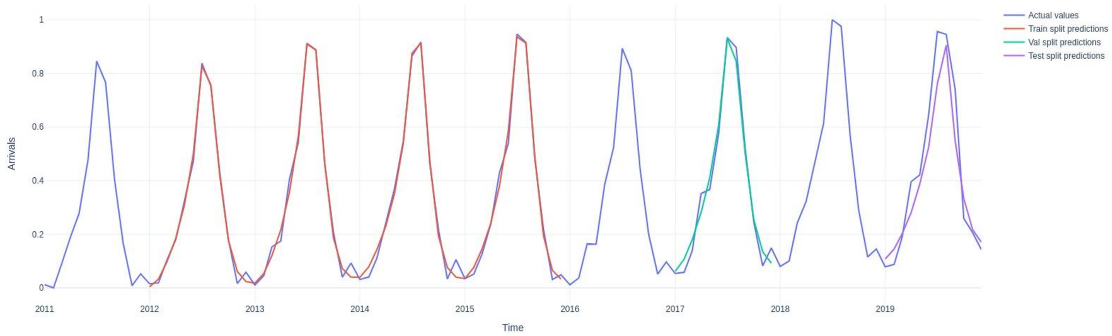


Results: Chart generated with the results of the BVM_Monthly LSTM model with Arrivals as features for Italy.

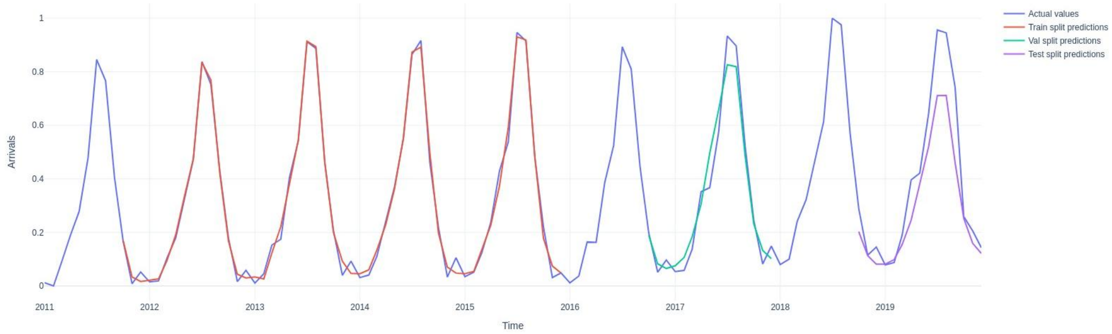
A.2.4 France



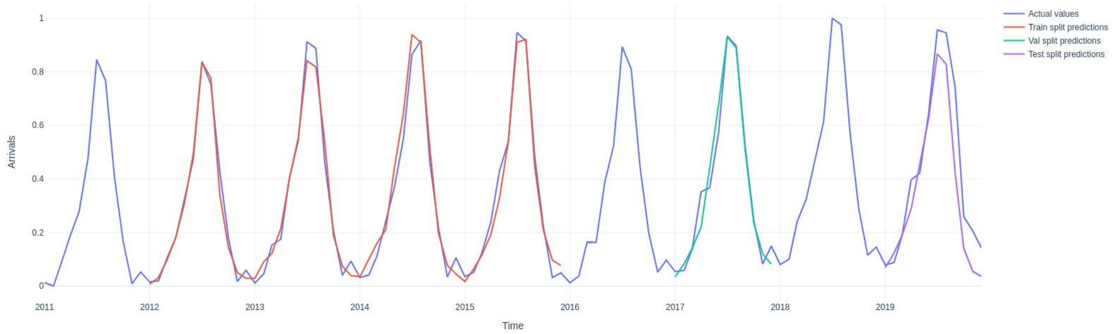
Results: Chart generated with the results of the BVM_Monthly LSTM model with All_features as features for France.



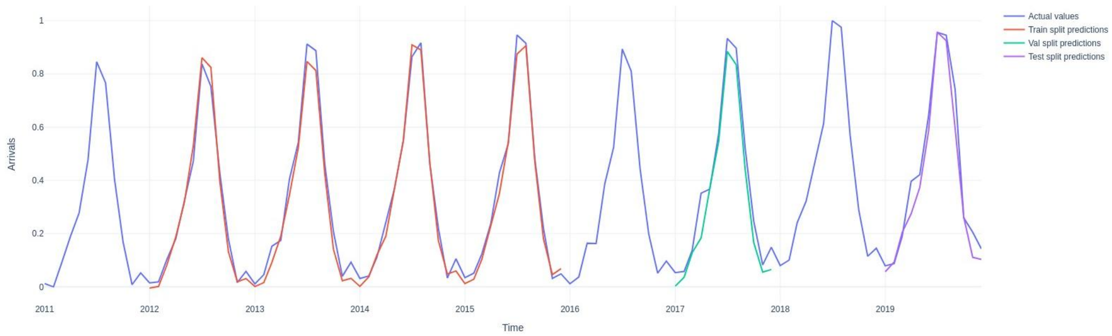
Results: Chart generated with the results of the BVM_Monthly GRU model with All_features as features for France.



Results: Chart generated with the results of the BVM_Monthly LSTM model with Trends_arrivals as features for France.

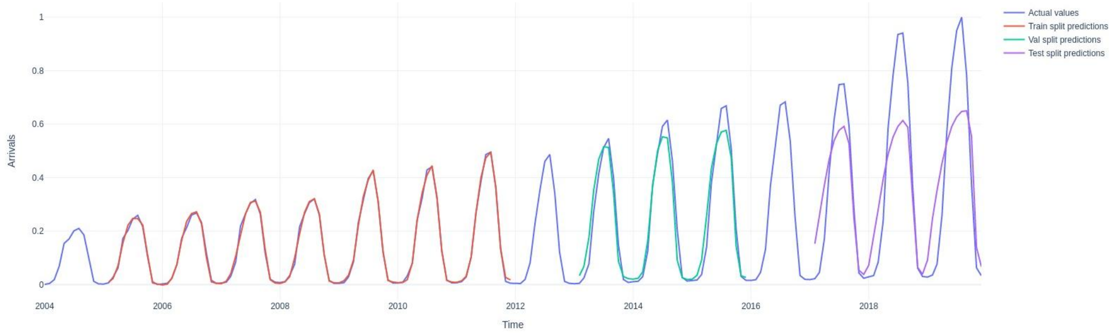


Results: Chart generated with the results of the BVM_Monthly GRU model with Trends_arrivals as features for France.

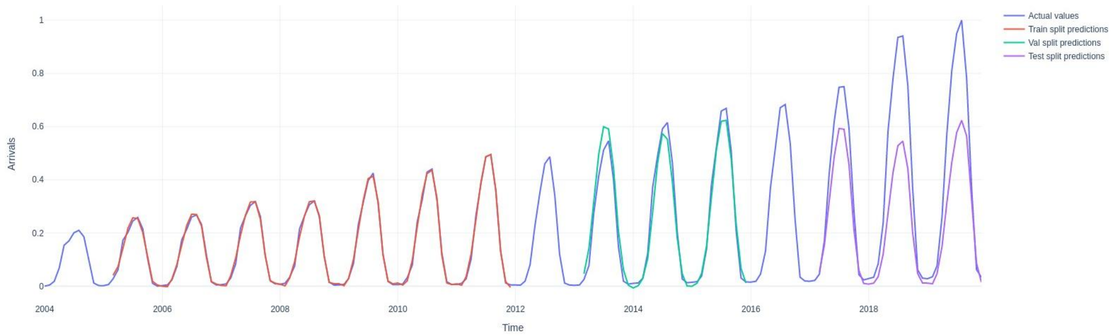


Results: Chart generated with the results of the BVM_Monthly GRU model with Arrivals as features for France.

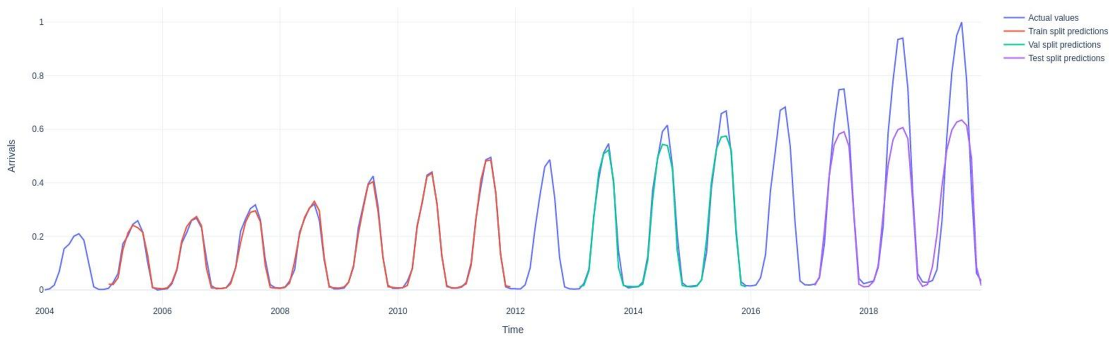
A.2.5 Greece



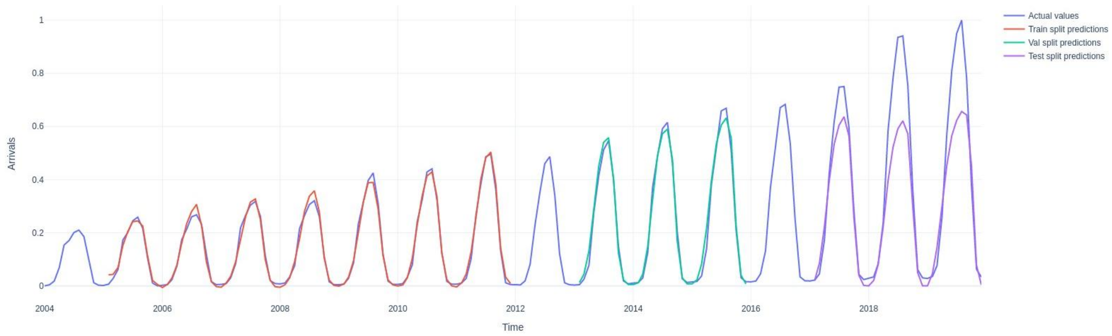
Results: Chart generated with the results of the BVM_Monthly LSTM model with All_features as features for Greece.



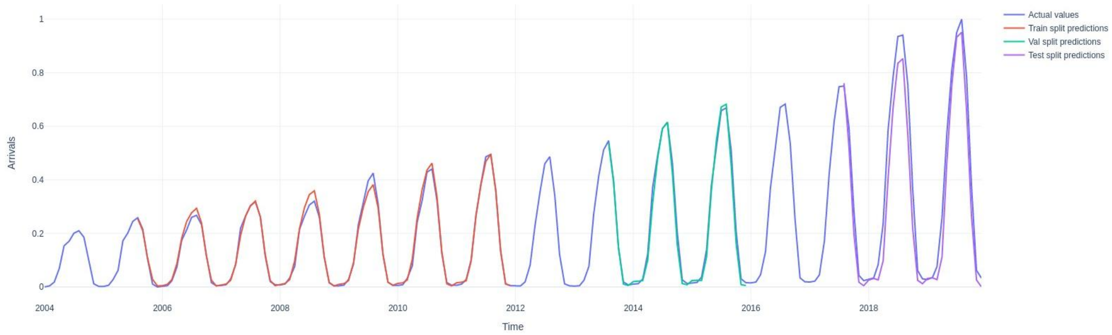
Results: Chart generated with the results of the BVM_Monthly GRU model with All_features as features for Greece.



Results: Chart generated with the results of the BVM_Monthly LSTM model with Trends_arrivals as features for Greece.



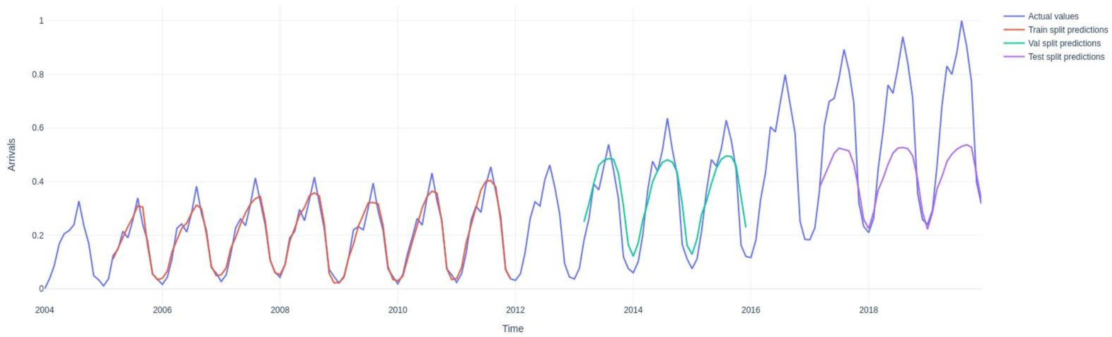
Results: Chart generated with the results of the BVM_Monthly GRU model with Trends_arrivals as features for Greece.



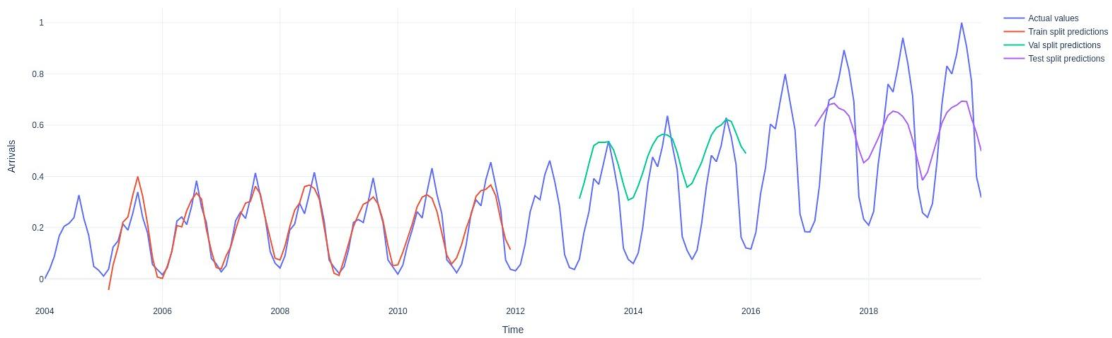
Results: Chart generated with the results of the BVM_Monthly LSTM model with Arrivals as features for Greece.

A.3 BTM Monthly Models Results

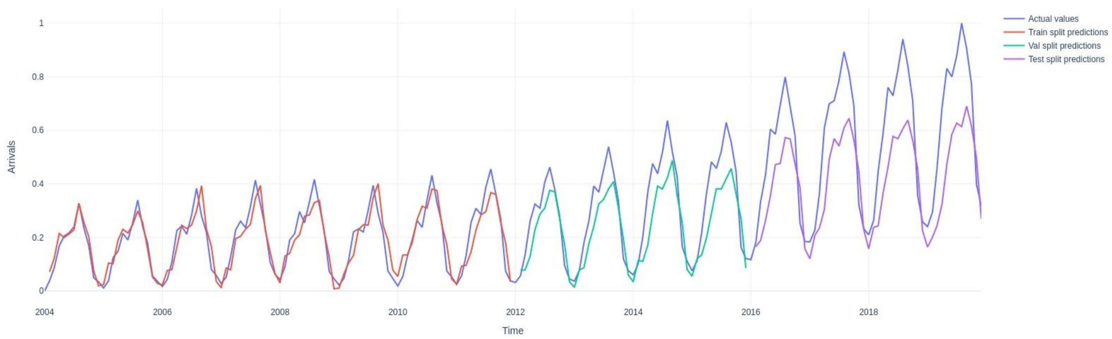
A.3.1 Portugal



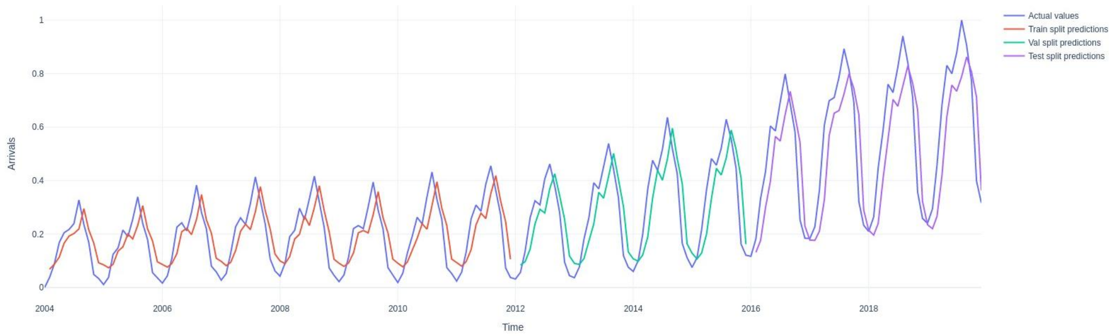
Results: Chart generated with the results of the Best Test-set Metrics (BTM)_Monthly LSTM model with All_features as features for Portugal.



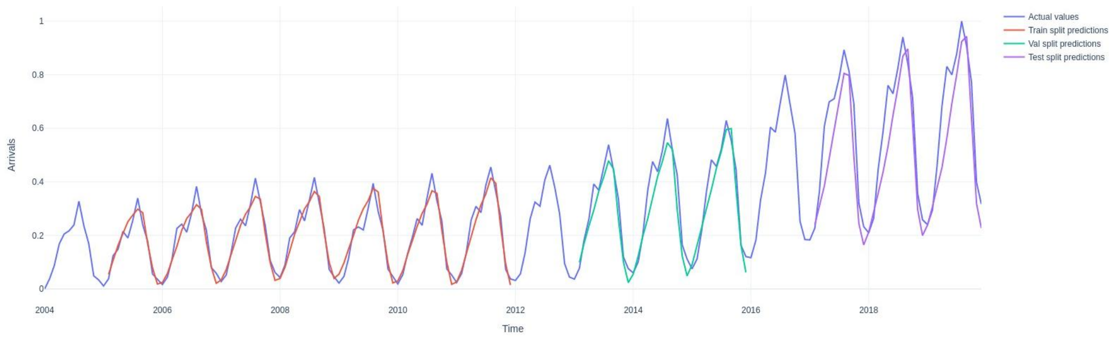
Results: Chart generated with the results of the BTM_Monthly GRU model with All_features as features for Portugal.



Results: Chart generated with the results of the BTM_Monthly GRU model with Trends_arrivals as features for Portugal.

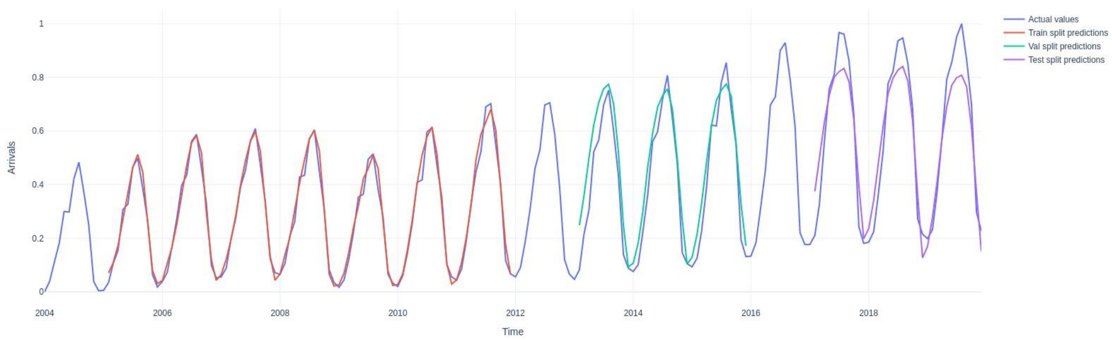


Results: Chart generated with the results of the BTM_Monthly LSTM model with Arrivals as features for Portugal.

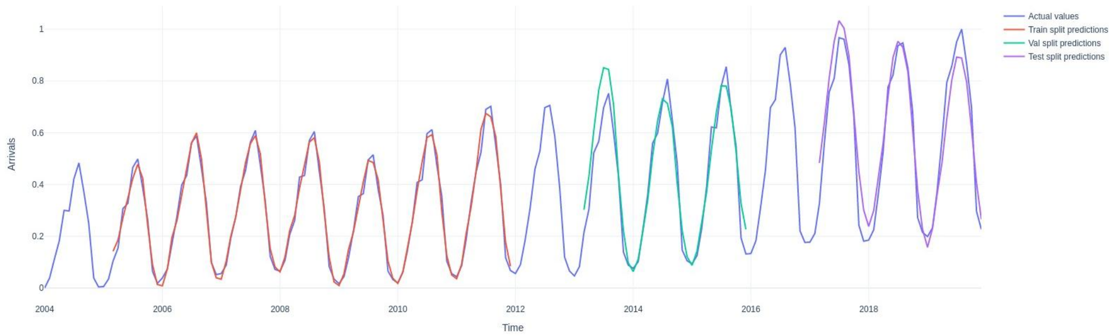


Results: Chart generated with the results of the BTM_Monthly GRU model with Arrivals as features for Portugal.

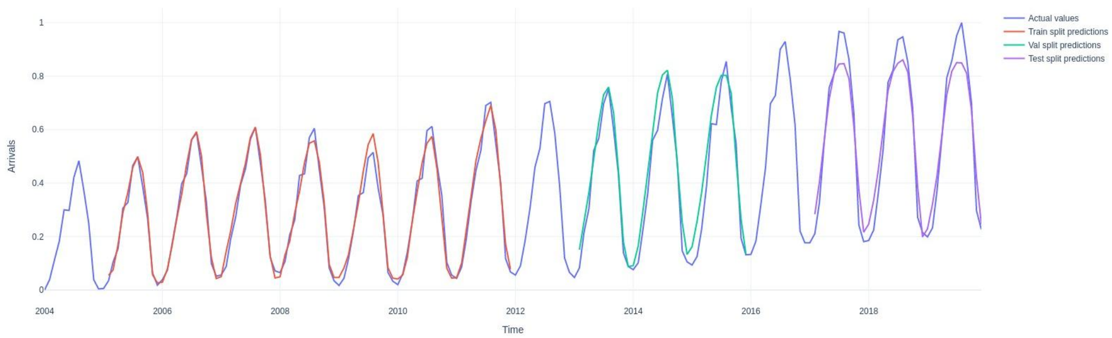
A.3.2 Spain



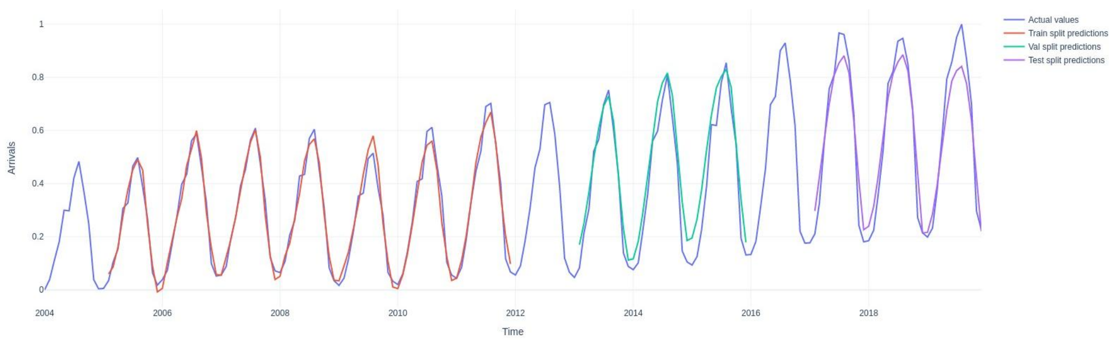
Results: Chart generated with the results of the BTM_Monthly LSTM model with All_features as features for Spain.



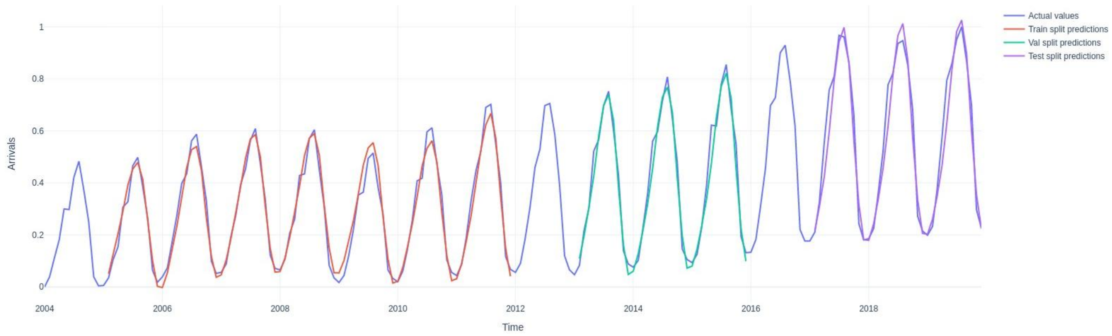
Results: Chart generated with the results of the BTM_Monthly GRU model with All_features as features for Spain.



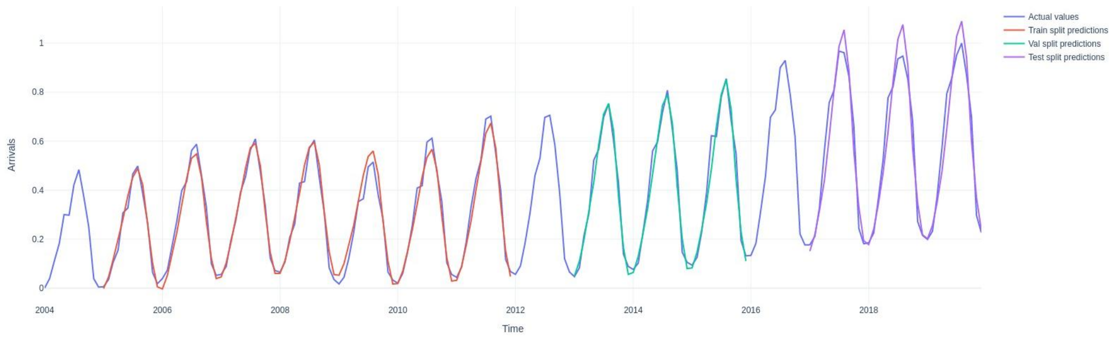
Results: Chart generated with the results of the BTM_Monthly LSTM model with Trends_arrivals as features for Spain.



Results: Chart generated with the results of the BTM_Monthly GRU model with Trends_arrivals as features for Spain.

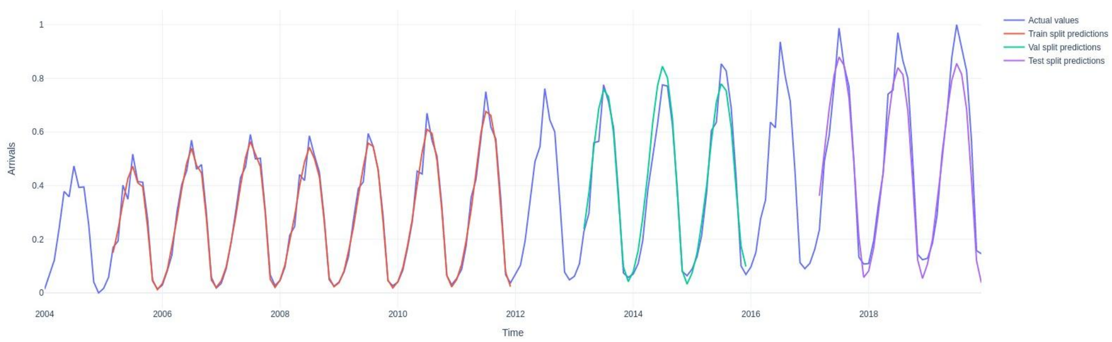


Results: Chart generated with the results of the BTM_Monthly LSTM model with Arrivals as features for Spain.

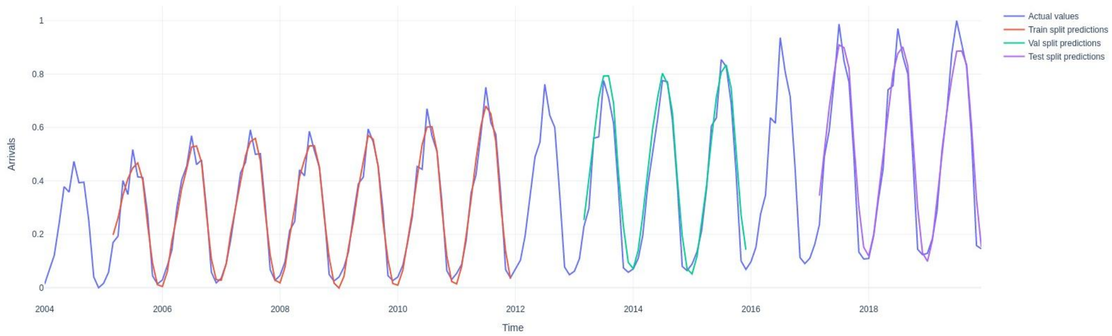


Results: Chart generated with the results of the BTM_Monthly GRU model with Arrivals as features for Spain.

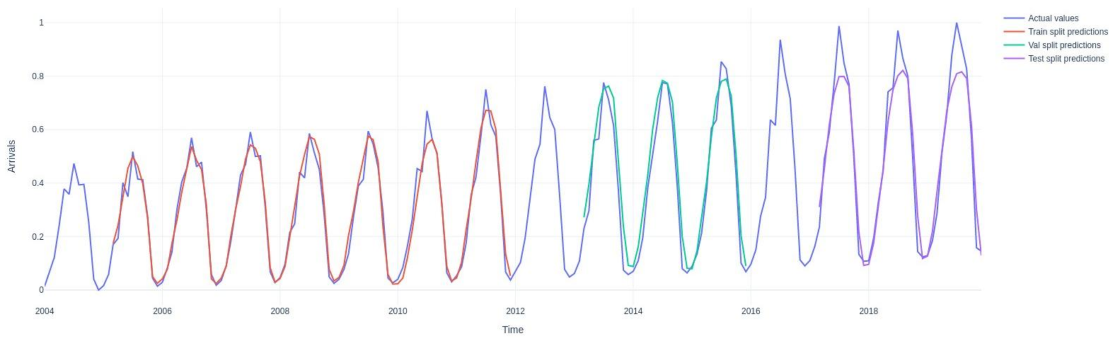
A.3.3 Italy



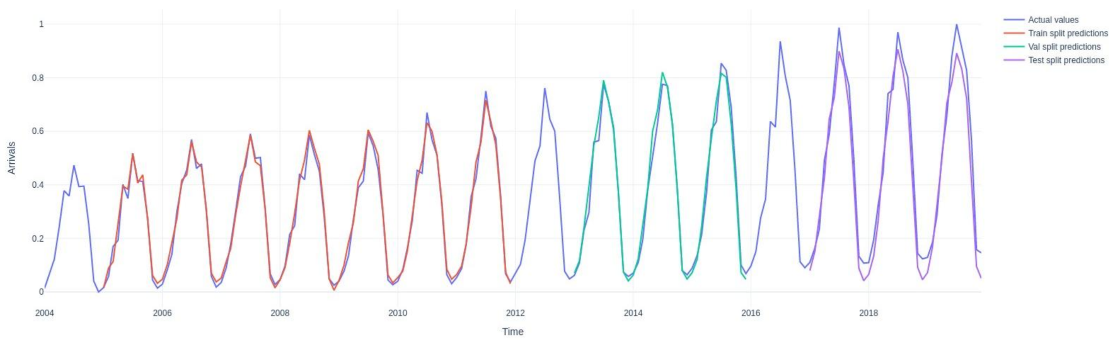
Results: Chart generated with the results of the BTM_Monthly LSTM model with All_features as features for Italy.



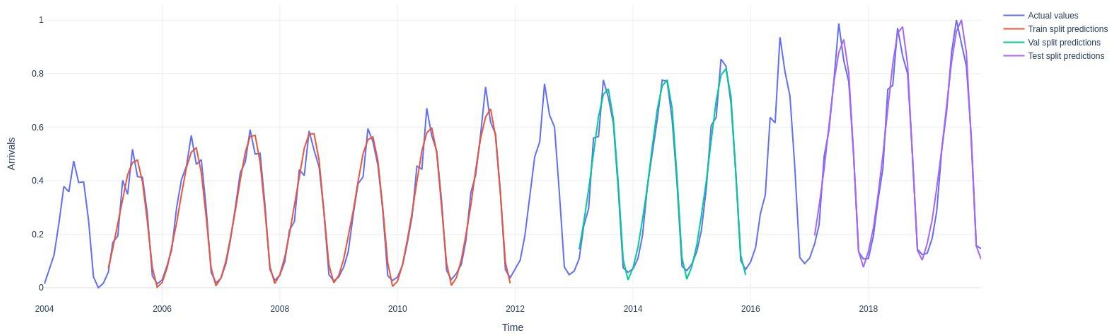
Results: Chart generated with the results of the BTM_Monthly GRU model with All_features as features for Italy.



Results: Chart generated with the results of the BTM_Monthly LSTM model with Trends_arrivals as features for Italy.

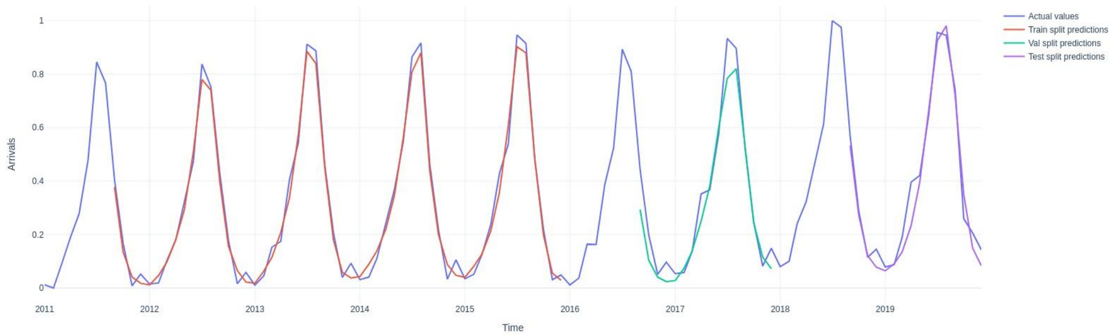


Results: Chart generated with the results of the BTM_Monthly GRU model with Trends_arrivals as features for Italy.

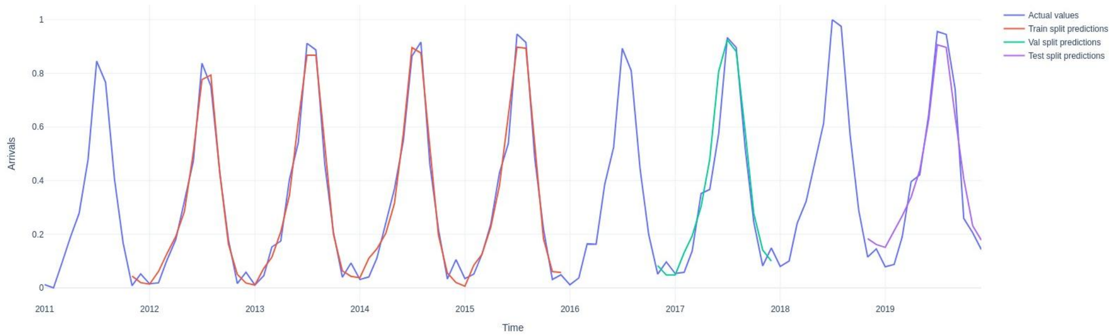


Results: Chart generated with the results of the BTM_Monthly LSTM model with Arrivals as features for Italy.

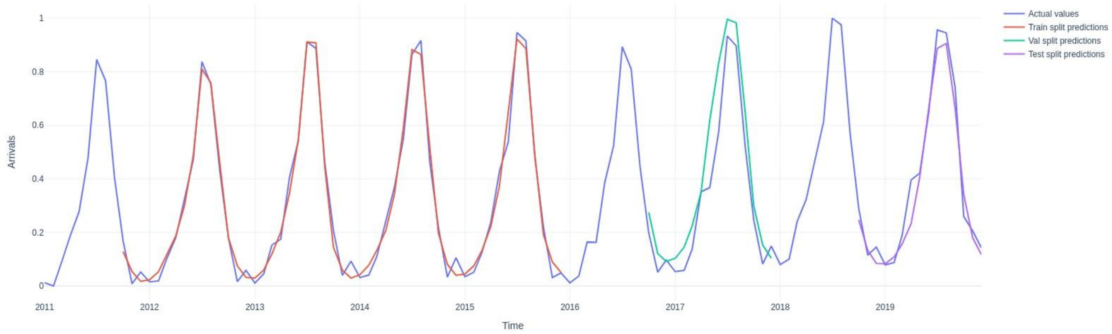
A.3.4 France



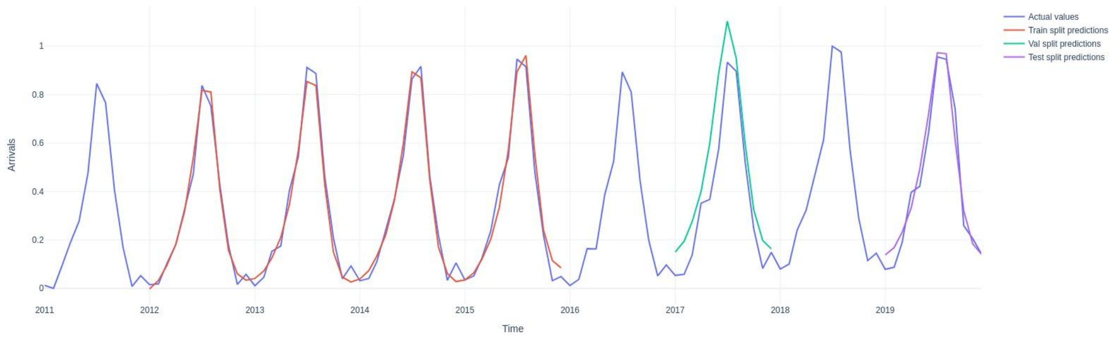
Results: Chart generated with the results of the BTM_Monthly LSTM model with All_features as features for France.



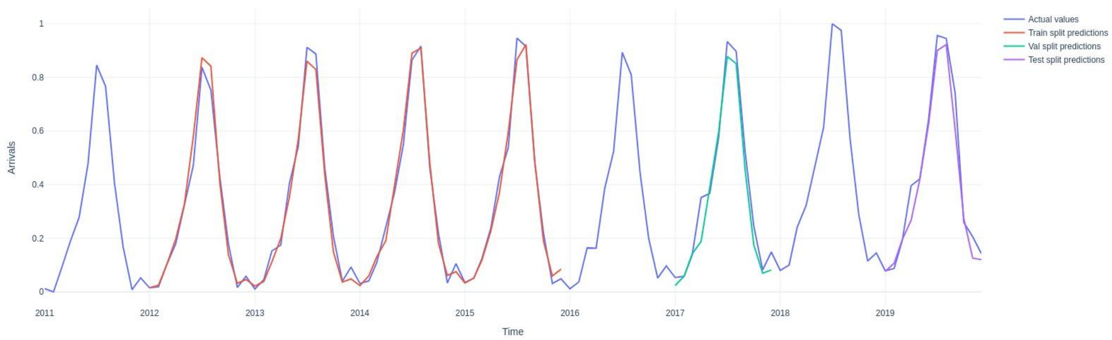
Results: Chart generated with the results of the BTM_Monthly GRU model with All_features as features for France.



Results: Chart generated with the results of the BTM_Monthly LSTM model with Trends_arrivals as features for France.

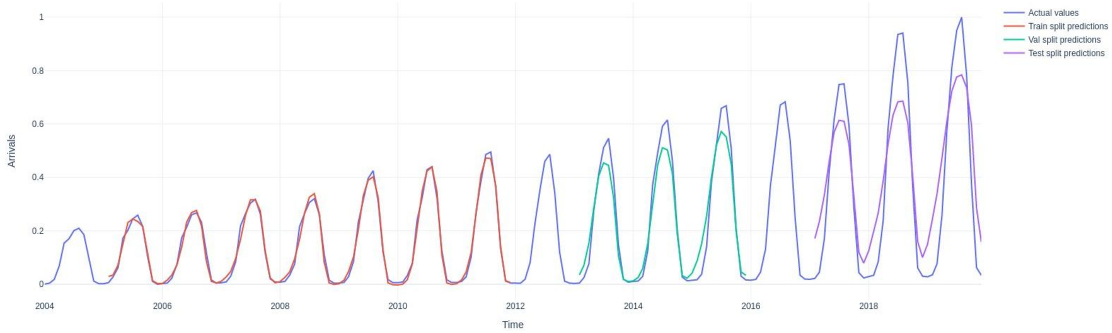


Results: Chart generated with the results of the BTM_Monthly GRU model with Trends_arrivals as features for France.

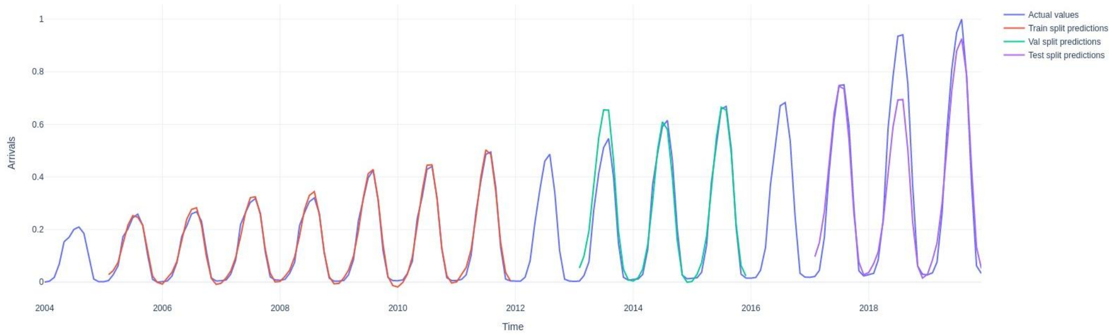


Results: Chart generated with the results of the BTM_Monthly GRU model with Arrivals as features for France.

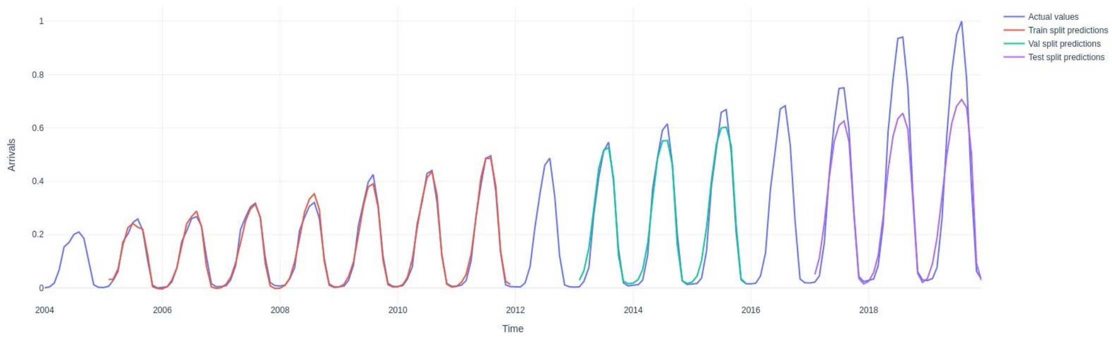
A.3.5 Greece



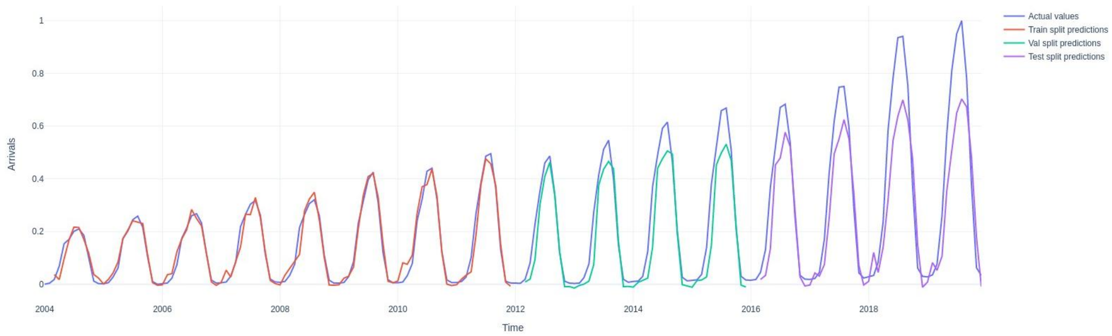
Results: Chart generated with the results of the BTM_Monthly LSTM model with All_features as features for Greece.



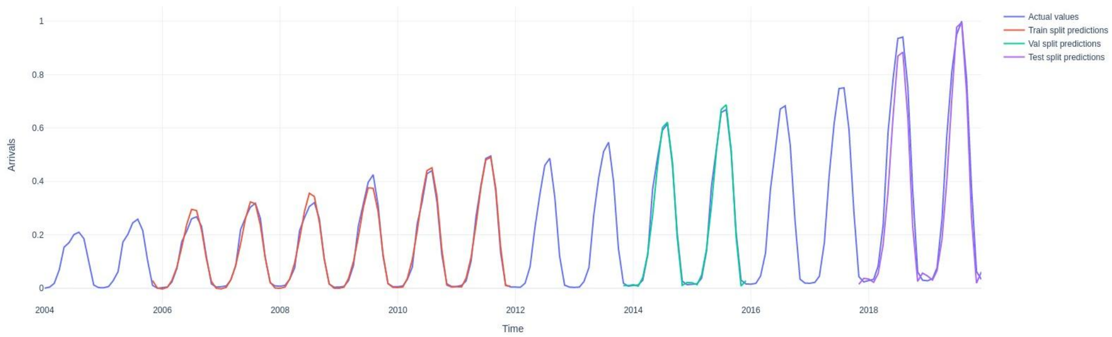
Results: Chart generated with the results of the BTM_Monthly GRU model with All_features as features for Greece.



Results: Chart generated with the results of the BTM_Monthly LSTM model with Trends_arrivals as features for Greece.



Results: Chart generated with the results of the BTM_Monthly GRU model with Trends_arrivals as features for Greece.



Results: Chart generated with the results of the BTM_Monthly LSTM model with Arrivals as features for Greece.