



ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΤΟΥΡΙΣΜΟΥ



ΤΜΗΜΑ ΤΟΥΡΙΣΜΟΥ
ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

**Ανάλυση και πρόβλεψη χρονοσειρών στην τουριστική
βιομηχανία**

Time Series Prediction in the Tourism Industry

ΟΝΟΜΑ: Τσουκιάς Παύλος

ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ: T2020076

ΕΠΙΒΛΕΠΩΝ/ΟΥΣΑ ΚΑΘΗΓΗΤΗΣ/ΤΡΙΑ: Αρτίκης Κωνσταντίνος

ΚΕΡΚΥΡΑ 2024

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΜΗ ΛΟΓΟΚΛΟΠΗΣ

Αυτή η διπλωματική εργασία υποβάλλεται από το προπτυχιακό φοιτητή Τσουκιά Πάυλο ως μερική εκπλήρωση των απαιτήσεων του Προπτυχιακού Προγράμματος Σπουδών του τμήματος Τουρισμού του Ιονίου Πανεπιστημίου. Υπεύθυνα δηλώνεται ότι, η συγκεκριμένη διπλωματική εργασία έχει συγγραφεί από τον Υπογράφοντα και δεν έχει υποβληθεί ούτε έχει αξιολογηθεί στο πλαίσιο άλλου μεταπτυχιακού ή προπτυχιακού τίτλου σπουδών, στην Ελλάδα ή στο εξωτερικό.

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: Τσουκιάς Πάυλος

ΥΠΟΓΡΑΦΗ:



Ευχαριστίες

Σε αυτό το σημείο, θα ήθελα να ευχαριστήσω ιδιαιτέρως τον επιβλέποντα καθηγητή μου, Δρ. Αρτίκη Κωνσταντίνο του Ιονίου Πανεπιστημίου, για την πολύτιμη καθοδήγησή του κατά τη διάρκεια της συγγραφής της παρούσας πτυχιακής. Επίσης, θα ήθελα να ευχαριστήσω την οικογένειά μου, αλλά και τη σύντροφό μου, Βαρβάρα Μίαρη, για την υποστήριξή τους καθ' όλη τη διάρκεια της φοιτητικής μου πορείας.

Abstract

The dissertation critically examines the pivotal role of forecasting in the tourism and aviation industries. Over recent years, major software companies have dedicated efforts to providing forecasts, assisting businesses and organizations in making informed short-term and long-term decisions aimed at optimizing operational day-to-day activities and crucial long-term strategies. Beyond the realm of software companies, numerous organizations and academic researchers have developed their own tools, harnessing open-source resources like Python and other open-source libraries, as demonstrated in this dissertation.

The primary objectives of the dissertation revolve around deepening our understanding of time series forecasting applications. Specifically, the study addresses the challenges of forecasting during non-normal periods, exemplified by the impact of the COVID-19 pandemic on passenger arrivals at Athens International Airport. Additionally, the research seeks to compare traditional time series models like SARIMA (Seasonal AutoRegressive Integrated Moving Average) with modern deep learning models like LSTM (Long Short-Term Memory). This comparative analysis involves experimenting with optimization techniques, including grid search, autoARIMA, and rolling forecast origin.

The dissertation is structured into three main sections. The initial part encompasses the introduction and a comprehensive literature review. The second part delves into the experimental phase, which is divided into two segments. The first segment introduces models to a dataset without anomalies and attempts forecasting, while the second phase introduces anomalies into the training data. Finally, the dissertation concludes with findings derived from the experiments. Through this research, the author aims to provide valuable insights into the effectiveness of various forecasting models, particularly in the context of disruptive events like the COVID-19 pandemic.

Περίληψη

Στη σύγχρονη εποχή, ο τομέας των προβλέψεων βιώνει ταχεία εξέλιξη, ειδικά στους τομείς του τουρισμού και των αερομεταφορών. Τα τελευταία χρόνια, παρατηρείται αυξημένο ενδιαφέρον από εταιρίες λογισμικού που προσπαθούν να παρέχουν λύσεις σε οργανισμούς, επιτρέποντάς τους να λαμβάνουν πιο ακριβείς μακροπρόθεσμες στρατηγικές αποφάσεις και να βελτιστοποιούν τις καθημερινές τους λειτουργίες. Πέρα από τις εταιρίες λογισμικού, πολλές εταιρίες και ερευνητές έχουν αναπτύξει τις δικές τους λύσεις, χρησιμοποιώντας εργαλεία που έχουν αναπτυχθεί από την open-source κοινότητα όπως η γλώσσα προγραμματισμού Python και σχετικές βιβλιοθήκες, όπως και στην περίπτωση αυτής της διατριβής.

Οι κύριοι στόχοι της παρούσας εργασίας επικεντρώνονται στην καλύτερη κατανόηση των εφαρμογών πρόβλεψης χρονολογικών σειρών. Συγκεκριμένα, η μελέτη ασχολείται με τις προκλήσεις των προβλέψεων κατά τη διάρκεια μη κανονικών περιόδων, με παράδειγμα τις επιπτώσεις της πανδημίας COVID-19 στις αφίξεις επιβατών στο Διεθνή Αερολιμένα Αθηνών. Επιπλέον, η έρευνα επιδιώκει να συγκρίνει παραδοσιακά μοντέλα χρονοσειρών όπως το SARIMA (Seasonal AutoRegressive Integrated Moving Average) με σύγχρονα μοντέλα βαθιάς μάθησης όπως το LSTM (Long Short-Term Memory). Αυτή η συγκριτική ανάλυση περιλαμβάνει τον πειραματισμό με τεχνικές βελτιστοποίησης, συμπεριλαμβανομένης της αναζήτησης πλέγματος, του αλγορίθμου autoARIMA και της τεχνικής rolling forecast origin.

Η διπλωματική εργασία διαιρείται σε τρεις κύριες ενότητες. Το πρώτο μέρος περιλαμβάνει την εισαγωγή και μια ολοκληρωμένη ανασκόπηση της βιβλιογραφίας. Το δεύτερο μέρος εμβαθύνει στην πειραματική φάση, η οποία διαιρείται σε δύο τμήματα. Το πρώτο τμήμα εκπαιδεύει τα μοντέλα σε ένα σύνολο δεδομένων χωρίς να εισάγει ανωμαλίες και επιχειρεί την πρόβλεψη, ενώ η δεύτερη φάση εισάγει ανωμαλίες στα δεδομένα εκπαίδευσης και προβαίνει σε προβλέψεις. Τέλος, η διπλωματική εργασία ολοκληρώνεται με τα συμπεράσματα που προκύπτουν από τα πειράματα. Μέσω αυτής της έρευνας, ο συγγραφέας στοχεύει να παράσχει πολύτιμες πληροφορίες σχετικά με την αποτελεσματικότητα των διαφόρων μοντέλων πρόβλεψης, ιδίως στο πλαίσιο γεγονότων τα οποία δημιουργούν αστάθειες όπως η πανδημία COVID-19.

Table Of Contents

Ευχαριστίες	4
Abstract	5
Περίληψη	6
Chapter 1 Introduction	10
Chapter 2 Literature Review	11
2.1 Predictive Methods	11
2.2 Evolution of Airline Passenger Forecasting Models: A Historical Overview (1995-2020).....	11
2.3 What Do We call Time Series And What Makes It Distinct From The Conventional Forecasting Methods?.....	13
2.4 The Basic Components Of A Time Series	13
2.5 Seasonality	13
2.6 Cyclic Patterns	13
2.7 Irregular/Random Fluctuations	13
2.8 Other Very Important Concepts For Time Series.	14
2.9 Traditional Time Series Models.....	14
2.9.1 The AR, MA, ARMA and ARIMA models.....	14
2.9.2 MA (Moving Average) model:	15
2.9.3 ARMA (Auto Regressive Moving Average) model:	15
2.9.4 ARIMA (Auto Regressive Integrated Moving Average) model:	16
2.9.5 SARIMA (Seasonal Auto Regressive Integrated Moving Average) model: ..	16
2.10 Metrics	16
Forecast Accuracy Metrics	16
2.10.1 MAE (Mean Absolute Error):.....	16
2.10.2 MAPE (Mean Absolute Percentage Error):	17
2.10.3 RMSE (Root Mean Squared Error):	17
2.11 Model selection metrics	18
2.11.1 Akaike Information Criterion (AIC):.....	18
2.11.2 Bayesian Information Criterion (BIC):.....	18
2.11.3 The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF).....	19
2.12 Deep Learning Models.....	19
2.12.1 Basics Of Deep Learning	19

2.13 LSTM.....	22
Chapter 3 Methodology	25
3.1 The Data.....	25
3.2 The Experimental Design	25
• AutoARIMA	26
• GridSearch	26
3.3 Data Analysis	27
3.4 Statistical Testing.....	28
.....	29
Chapter 4 The Experiment	30
4.1 Experiment Part One	30
4.2 Experiment Part Two	34
Chapter 5 Final Results	38
5.1 The Results From Phase One.	38
5.2 The Results From Phase Two.	39
5.3 Comparing And Summarizing The Results From Both Phases.....	39
Chapter 6 Conclusions	39
References.....	41

Table Of Figures

Figure1: A single neuron with multiple inputs and one output.	20
Figure 2: Image demonstrating the flow of forward and back propagation.....	21
Figure 3: Image emphasizing the cell state of an LSTM network.	22
Figure 4: Image Emphasizing the forget gate of an Lstm network.	23
Figure 5: Line plot illustrating the annual passenger arrivals at Athens International Airport from the years 2002 to 2022. The graph also indicates the two dates chosen as separators for the creation of distinct training sets.....	25
Figure 6: Figure illustrating the rolling forecast origin technique with a fixed window.	27
Figure 7: Line plot depicting the timeline of major crises with corresponding dates.	27
Figure 8: Line plot representing the first dataset earmarked for analysis and further work.	28
Figure 9: ADF and KPSS results 1	29
Figure 10: ADF and KPSS results 2	29
Figure 11: Seasonal trend decomposition plots.....	29
Figure 12: Grid Search and AutoArima results 1	30
Figure 13: Phase one Grid_Search_1 results vs predicted.....	32
Figure 15: Auto_Arima_1 after the use of the rolling forecast origin technique.	33
Figure 16: Grid_Serach_1 after the use of the rolling forecast origin technique.	34
Figure 17: Grid Search and AutoArima results 2	35
Figure 18: Phase two Grid_Search_2 results vs predicted	36
Figure 19: Phase two Auto_Arima_2 results vs predicted	36
Figure 20: Phase two LSTM_2 results vs predicted	37
Figure 21: Auto_Arima_2 after the use of the rolling forecast origin technique.	37
Figure 22: Grid_Serach_2 after the use of the rolling forecast origin technique	38
Table 1: Phase one of the experiment: Initial results	31
Table 2: Phase one of the experiment: Final results	33
Table 3: Phase two of the experiment: Initial results	35
Table 4 : Phase two of the experiment: Final results	37
Table 5: Table with all the final results.....	38

Chapter 1

1 Introduction

In contemporary times, tourism has evolved into one of the most significant and rapidly expanding industries globally (Buhalis κ.ά., 2023). Many countries regard tourism as a primary source of revenue, with Greece being a notable example. According to a report by (INSETE intelligence, 2020), tourism's direct and indirect contributions to Greece's economy in 2021 accounted for 19.8% of its total GDP. With the majority of incoming tourists utilizing Greece's airports as their primary entry points.

At the start of 2020, the COVID-19 pandemic hit globally, severely impacting the aviation industry. For nearly two years, a significant number of planes were grounded, and many seasonal employees working at airports lost their jobs due to the downturn (Kazda κ.ά., 2022). Following the pandemic, a major issue emerged at airports worldwide. They struggled to cope with the massive influx of passengers either leaving or entering their countries. This led to a global phenomenon of people stranded for days at airports, and operational challenges arose due to the inability to accurately forecast the sudden surge in passenger volumes (Kazda κ.ά., 2022).

Forecasting plays a crucial role in the tourism industry, utilized by airlines, hoteliers, and airports to predict demand and optimize their operations. Its primary goal is to aid these businesses in selling as many products and services as possible. A key challenge for all tourism-related businesses is the inability to stockpile unsold products, making accurate forecasting essential for maximizing revenue and minimizing losses (Frechtling, 2001). Besides maximizing profits forecasting is also used for the creation of strategic, tactical and operational decision making (Lim, 1997).

Athens International Airport, recognized as Greece's largest international airport in 2023, reported a substantial 24% increase in passenger traffic, reaching an impressive total of 28.174.150 passengers. This surge represents a 10.2% rise from 2019, a year acclaimed by tourism specialists as a record-breaker for Greek tourism (AIA/Management Information System, 2023). However, this upward trend calls for strategic operational management to sustain growth.

Similar challenges have been observed at major airports such as Amsterdam's Schiphol, where rapid growth led to operational issues, including temporary airport closures and strikes by overburdened staff (Elton, 2022). Such incidents highlight how unchecked growth can potentially disrupt steady progress. To maintain this momentum, it is crucial for Athens International Airport to engage in comprehensive strategic planning. This includes long-term forecasting for overarching strategic development and short-term forecasting to optimize daily airport operations (Law κ.ά., 2019). Implementing these proactive measures is essential to ensure that the airport's increasing popularity does not lead to operational inefficiencies or a decline in service quality, thereby safeguarding its ongoing success.

Traditional time series models such as ARIMA and its derivatives have been used in the aviation industry by airlines and airports to forecast passenger arrivals (Guo κ.ά., 2022). Nowadays, with the development of technology, newer and more sophisticated models have been introduced, including deep learning models based on neural networks. The purpose of this dissertation is to benchmark the classic SARIMA models against the newer and more sophisticated LSTM models (Long Short-Term Memory) to see which ones can more accurately predict passenger arrivals on a monthly basis at the Athens International Airport during the COVID-19 crisis, and determine which of the two models managed to perform better.

The objectives of the study are to provide a better understanding for both academia and managerial purposes. The dissertation builds upon previous work done in this field by Do et al. (2020) and is adapted to accommodate the limitations of the dataset derived from the statistical reports of Athens International Airport.

Chapter 2

2 Literature Review

2.1 Predictive Methods

Over the years, a wide array of predictive forecasting methods have been developed. These methods are commonly categorized in the literature as either quantitative or qualitative. Among the qualitative methods, the Delphi and Consensus approaches are particularly notable. Generally, these methods are considered more ‘abstract’ and tend to have limited generalization capabilities (Chan κ.ά., 2010; Law κ.ά., 2019). "In the quantitative approach, models are divided into two types: time series models and causal models. Time series models aim to predict the future by analyzing data from the past (Asteriou & Hall, 2011). Some of the most commonly used time series models include AR (Autoregressive), MA (Moving Average), ARIMA (Autoregressive Integrated Moving Average), and various extensions of these models. In contrast, causal models are designed to quantify and understand the relationships and effects between variables (Hitchcock, 2023). Causal approaches most commonly utilize regression analysis to analyze the causality between independent and dependent variables. The primary aim is to identify the independent variable that most logically expresses the dependent variable, create a model representing this relationship, and ultimately use this model for forecasting. In recent years, with the significant growth of computer science, more sophisticated forecasting models based on Artificial Neural Networks have emerged such as LSTM models. This dissertation examines two types of time series models: the first being SARIMA, and the second, LSTM models.

2.2 Evolution of Airline Passenger Forecasting Models: A Historical Overview (1995-2020)

Building on the foundational research conducted by Do et al. (2020), which spanned from 1995 to 2019, numerous researchers have developed various forecasting models and techniques aimed at predicting airline passenger arrivals and related metrics.

Specifically, in 1995, Nam and Schaefer proposed an Artificial Neural Network (ANN) as a predictive model for airline passenger traffic. Later, in 2001, a regression model was devised to forecast airline passenger demand in Saudi Arabia, utilizing 22 years of data for training and evaluating the model.

In 2007, two significant studies were published in the field of air passenger traffic forecasting. The first study, as detailed by Grosche et al. (2007), utilized a gravity model to predict the volume of air passengers between Berlin, the capital of Germany, and 28 European countries. This approach was particularly tailored to estimate the flow of travelers to and from Berlin's airports. The second study, conducted by Bermúdez et al. (2007), applied the Holt-Winters method for forecasting the number of passengers in the UK, projecting trends from the year 2005 to 2030. This method is renowned for its effectiveness in time series forecasting, especially in the context of transportation and travel demand.

In 2010, three additional papers were published in the field of airline passenger forecasting. The first of these, authored by Samagaio and Wolters (2010), applied exponential smoothing and autoregressive models to predict passenger numbers at Lisbon Airport for the period from 2008 to 2020. This model was trained using data from 1995 to 2007. The study concluded that government forecasts for passenger traffic tended to be overly optimistic.

In the year 2011 David Duval proposed a Regression model in order to predict passenger arrivals visiting New Zealand. The following year 2012 some Post-mortem techniques were utilized to assess air transportation predictions, concentrating on time trend analysis and econometrics. These methods demonstrated their efficiency using Rhodes Airport as a case study (Profillidis, 2012). At the year 2013 A combined local experts model was used in order to predict air passenger numbers at São Paulo International Airport (Scarpel, 2013).

In 2014, Tsui et al. utilized SARIMA and ARIMAX models to predict rising passenger traffic at Hong Kong International Airport until 2015, incorporating different growth scenarios (Tsui et al., 2014).

In 2016, a novel method for short-term airport passenger forecasting was introduced, utilizing internet search keywords as independent variables for prediction (Kim & Shin, 2016). In the subsequent year, a deep learning strategy was proposed for forecasting passenger demand in Egypt, featuring a neural network trained and optimized using back propagation (M. κ.ά., 2017).

Aircraft ground operations impact flight punctuality more than airborne uncertainties. At the year 2019 a recurrent neural network, specifically Long Short-Term Memory, was used to predict boarding progress, considering passenger interactions and a complexity metric for better accuracy (Schultz & Reitmann, 2019). In 2020, a comparative study was conducted between LSTM and SARIMA models to determine which performed better in forecasting short-term and mid-term passenger demand at

Incheon International Airport. As outlined in the introduction, this dissertation builds upon the benchmarking concept by Do et al. (2020).

2.3 What Do We call Time Series And What Makes It Distinct From The Conventional Forecasting Methods?

A time series refers to a sequence of observations recorded at regular time intervals for a specific variable or phenomenon. Mathematically, a time series is typically represented as $Y = F(t)$ where Y represents the variable at a given time t . Therefore, Y symbolizes a sequence of values $Y_1, Y_2, \dots, Y_n$ over respective time points $t_1, t_2 \dots t_n$. This sequential nature is a defining aspect of time series data.

Moreover, time series analysis differs from conventional regression-based forecasting. In typical regression models, forecasts often predict within a known range of existing data (interpolation). Conversely, in time series analysis, the goal often involves predicting beyond the established range (extrapolation), which introduces increased uncertainty due to forecasting outside the known dataset range.

2.4 The Basic Components Of A Time Series

Trend: The long-term movement or direction of the data. It can be increasing, decreasing, or stable over time. If the trend is moving upwards, it implies a consistent increase in the values over time. Conversely, a downward trend indicates a consistent decrease in the values.

2.5 Seasonality: Patterns that occur at regular intervals within the data, often corresponding to calendar seasons, months, days of the week, etc. For example, take the flight data at Corfu airport. Every year, there's a clear pattern: more flights during summer and fewer as summer ends. This cycle repeats, showing the predictable increase and decrease in flights based on the seasons

2.6 Cyclic Patterns: Repeating patterns or fluctuations that are not fixed to a specific time period. Unlike seasonality, cycles might not have a fixed duration. For example, in the world economy, we see cycles of growth and downturns. Sometimes, the economy grows, then slows down or even goes into a recession. These cycles don't always happen at the same time or for the same length, showing a repeating pattern in the global economy.

2.7 Irregular/Random Fluctuations: Unpredictable variations that don't follow any specific trend, seasonality, or cycle. They represent the randomness or noise within the data. For example, think about the daily sales at a neighborhood store. Some days, the sales might be higher or lower without a clear reason or pattern. These unpredictable changes don't relate to specific days or seasons, showing random ups and downs in the sales data.

2.8 Other Very Important Concepts For Time Series.

2.8.1 Stationarity: In order to call a time series stationary, it needs to have three specific characteristics.

1. The mean of the time series needs to be constant
2. The Standard Deviation σ needs to be constant
3. There must not be at all seasonality

The most common ways to check for stationarity are:

1. Visually
2. With local and Global tests (By sampling from the dataset and checking out the criteria previously mentioned)
3. With the Augmented Dickey Fuller Test commonly known as ADF

2.8.2 White noise: In order for a time series to be considered white noise it need to have four specific criteria.

1. The mean of the series is zero, implying no overall trend.
2. The Standard Deviation σ needs to be constant over time
3. The correlation between lags needs to be 0 showing no predictable pattern or relationship
4. It must not be predictable

2.8.3 Unit root: A time series exhibits a unit root if its statistical properties don't meet the criteria for stationarity. In other words, a unit root implies the time series data has a root that equals 1, indicating non-stationarity.

2.9 Traditional Time Series Models

2.9.1 The AR, MA, ARMA and ARIMA models.

AR (Auto Regressive) model: The Auto Regressive (AR) model is employed to forecast future values by assuming a linear relationship between past and present values of a variable or phenomenon. It predicts future outcomes based on how previous values of the same variable are related to each other. Mathematically an AR model Is usually represented as.

$$Y_t = \Phi_1 Y_{t-1} + \varepsilon_t$$

Were

Y_t The predicted value for time t

Φ_1 : The coefficient

Y_{t-1} : The value Y at a previous time stamp

ε_t : The error term

2.9.2 MA (Moving Average) model: The Moving Average (MA) model is like using an average of past errors to predict future outcomes. Let's say you're predicting tomorrow's weather based on the errors in your past weather predictions. If you've often been a bit off in the same way, this model helps adjust future predictions based on those past mistakes. Mathematically an MA model is usually represented as:

$$Y_t = \mu + \varepsilon_t + \theta_1 \times \varepsilon_{t-1}$$

Where

Y_t : The predicted value for time t

μ : Represents the mean of the series

θ_1 : represents the parameter or coefficient that signifies the relationship between the current observation and the previous error term. (weight)

ε_t : Error term at time t

ε_{t-1} : The error term at the previous time point

2.9.3 ARMA (Auto Regressive Moving Average) model: The ARMA model blends the concepts of both AR and MA models to forecast future values in a time series. The Auto Regressive (AR) part of it captures linear relationships among past observations, and the Moving Average (MA) part, which considers past errors. This dual inclusion allows the ARMA model to handle time series data exhibiting both linear patterns and past error influences, making it a robust tool for forecasting and analyzing diverse types of time-dependent data. Before moving on with the mathematical representation it is worth mentioning that the ARMA model is a foundation for more complicated models such as the ARIMA, SARIMA, ARIMAX etc. Mathematically, an ARMA model is often represented as:

$$Y_t = c + \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}$$

Where:

Y_t : signifies the predicted value at time ' t '.

c denotes the constant term or mean of the series.

ϕ_1 : represents the Auto Regressive parameters, depicting the influence of previous observations on the current value.

ε_t : stands for the error term at time ' t '.

θ_1, θ_q : represent the Moving Average parameters, characterizing the relationship between the error terms at different time points and the current error term

2.9.4 ARIMA (Auto Regressive Integrated Moving Average) model: The ARIMA model is used in time series data that follow trends. With the I component symbolizes the differencing that is crucial in order to achieve stationarity and detrend the data before applying the AR and MA components.

2.9.5 SARIMA (Seasonal Auto Regressive Integrated Moving Average) model:

The SARIMA (Seasonal Auto Regressive Integrated Moving Average) model is an advanced version of the ARIMA model, specifically developed to include seasonal variations frequently seen in time series data. The AR (Autoregressive), I (Integrated), and MA (Moving Average) components are retained as in ARIMA, but SARIMA extends their application to seasonal data. It is expressed as SARIMA (p, d, q) (P, D, Q) s, where p, d, q are the non-seasonal components and P, D, Q are their seasonal counterparts, with s indicating the seasonal cycle's length. This model is particularly effective in analyzing and forecasting time series data that exhibits both non-seasonal and seasonal patterns, offering a refined approach for scenarios where seasonality is a key factor.

2.10 Metrics

Forecast Accuracy Metrics

This analysis employs specific metrics to evaluate the accuracy of forecasts generated by the chosen models. The metrics used include:

2.10.1 MAE (Mean Absolute Error): Measuring the average magnitude of errors between predicted and actual values, the MAE provides a straightforward assessment of the forecast accuracy.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Where:

y_i : Represents the actual i th value.

$\hat{y}_i$: Represents the i th forecasted value.

n : Represents the total number of observations in the data

Characteristics:

- It treats all errors equally, giving equal weight to both small and large errors.
- Easier to interpret than RMSE as it's in the same unit as the original data.

Use Cases:

- Suitable for regression problems where all errors are equally important.

- Often used in financial contexts where overestimates and underestimates are equally bad.

2.10.2 MAPE (Mean Absolute Percentage Error): Offering an average percentage difference between predicted and actual values, the MAPE serves as a relative measure of the accuracy of the forecasts.

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100$$

Where:

y_i : Represents the actual i th value.

$\hat{y}_i$: Represents the i th forecasted value.

n : Represents the total number of observations in the data

Characteristics:

- It expresses accuracy as a percentage, which is very intuitive.
- Can be skewed or misleading if dealing with values close to zero.

Use Cases:

- Useful in scenarios where you want to express the error as a percentage, which can be more interpretable for business stakeholders.
- Common in economic forecasting and sales forecasting.

2.10.3 RMSE (Root Mean Squared Error): Derives from MSE by taking the square root of the average of squared errors. This metric serves to provide an interpretable metric in the original units of the data, offering a measure of the typical size of the errors.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Where:

y_i : Represents the actual i th value.

$\hat{y}_i$: Represents the i th forecasted value.

n : Represents the total number of observations in the data

Characteristics:

- It gives a relatively high weight to large errors. This means the RMSE should be more useful when large errors are particularly undesirable.
- The units of RMSE are the same as the original data, which can make interpretation easier.

Use Cases:

- Good for regression problems where you want to penalize large errors more.
- Useful in spatial analysis and forecasting where precision is crucial.

These metrics play a crucial role in the comparative analysis of models, enabling a comprehensive evaluation of their accuracy in forecasting time series data (Do κ.ά., 2020).

2.11 Model selection metrics

2.11.1 Akaike Information Criterion (AIC):

AIC is a method used to compare statistical models. It provides a relative estimate of the quality of statistical models for a given set of data. AIC aims to select a model that best explains the data with the fewest parameters. The lower the AIC value, the better the model, indicating a better trade-off between model fit and complexity (Akaike, 1974).

$$AIC = 2k - 2 \ln(L)$$

Where:

k is the number of estimated parameters in the model.

$\ln(L)$ is the log-likelihood of the model, i.e., the logarithm of the likelihood function for the model.

2.11.2 Bayesian Information Criterion (BIC):

BIC is a criterion for model selection among a finite set of models. It is derived from a Bayesian framework. Like AIC, BIC balances model fit and complexity (Neath & Cavanaugh, 2012). However, BIC imposes a stronger penalty for including additional parameters in the model. BIC typically results in the selection of simpler models compared to AIC. Both AIC and BIC are used in various statistical analyses to select the most appropriate model among a set of candidates, considering both the model's goodness of fit to the data and its complexity. Models with lower AIC or BIC values are generally preferred as they represent a better balance between accuracy and simplicity.

$$"BIC" = \ln(n)k - 2 \ln(L)$$

k : is the number of estimated parameters in the model.

n : is the number of observations.

$\ln(L)$: is the natural logarithm of the likelihood of the model.

2.11.3 The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF)

Primarily, the ACF and PACF serve as diagnostic tools. We include them in the model selection process because they aid in identifying potential orders for the AR and MA models.

2.11.4 ACF (Autocorrelation Function): Measures the correlation between an observation and its lagged values at previous time steps, considering both direct and indirect effects in the time series data.

2.11.5 PACF (Partial Autocorrelation Function): Determines the correlation between an observation and its lagged values at prior time steps, focusing solely on the direct effects while excluding the indirect effects in the time series.

2.12 Deep Learning Models

2.12.1 Basics Of Deep Learning

In order to understand the Deep Learning models used in this dissertation we will first need to lay down some foundational knowledge for how Neural Networks and Deep Neural Networks actually work.

We will start with the most basic of Neural Network a single-neuron and the components it is consisted.

Neuron: In artificial intelligence and deep learning, a neuron is a fundamental unit that receives input data, processes it through mathematical operations and produces an output. This computational unit is inspired by the way biological neurons function in the human brain, where individual neurons receive, process, and transmit signals to other connected neurons.

Input: The neuron receives input data or signals $(x_1, x_2, \dots, x_n)$.

Linear Transformation: Each input is associated with a weight $(w_1, w_2, \dots, w_n)$

representing its importance or influence on the neuron.

The neuron performs a weighted sum of the inputs and weights:

$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n$$

Activation Function:

The weighted sum (z) is passed through an activation function, which introduces non-linearity and helps the network learn complex patterns.

Common activation functions include ReLU, Sigmoid, Tanh, etc.

Output: The output (y) is the result of the activation function applied to the weighted sum:

$y=f(z)$, where f is the chosen activation function.

The following figure 1.1 can help you visualize the concept

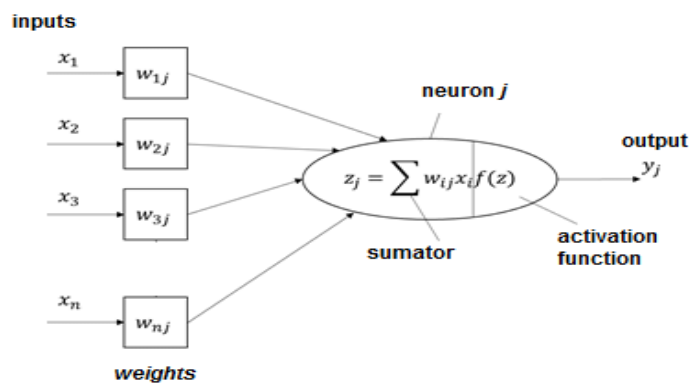


Figure1: A single neuron with multiple inputs and one output.

Source: <https://oarjpublication.com/journals/oarjet/content/review-machine-learning-applications-power-systems-studies>

Now that we clearly addressed what a neuron is we need to explain the components a deep neural network.

Input Layer:

The first layer of the neural network that receives the raw input data.

Each neuron in this layer represents a feature or variable in the dataset.

Hidden Layers: Intermediate layers situated between the input and output layers.

Each hidden layer contains multiple neurons that process and transform the input data.

The depth and width of hidden layers contribute to the network's learning capacity.

Output Layer: The final layer that produces the network's predictions or classifications.

The number of neurons in this layer is determined by the nature of the problem (e.g., one neuron for regression, multiple neurons for classification).

Deep neural networks leverage these layers to learn complex representations and patterns from the data. The input layer receives information, which is processed through the hidden layers, allowing the network to abstract increasingly complex features and, finally, the output layer produces the desired results based on these learned representations.

The configuration and arrangement of these layers significantly impact the network's ability to learn intricate relationships within the data and make accurate predictions or classifications. This layered architecture enables deep neural networks to handle complex tasks in various domains like image recognition, natural language processing, and more.

And finally, we need to talk about two very important concepts forward and back propagation.

Forward Propagation: Forward propagation involves the process of inputting data, processing it through a neural network, and obtaining an output.

Back Propagation: This process fine-tunes the weights in a neural network's hidden layers to enhance predictions by learning from the discrepancies between predicted and actual outputs. It occurs over several iterations, termed epochs, with the primary objective of minimizing the loss function. Through continuous parameter adjustments, it strives to approach a local minimum for this function, steadily refining the model's accuracy with each iteration.

The following figure helps to visualize the concepts.

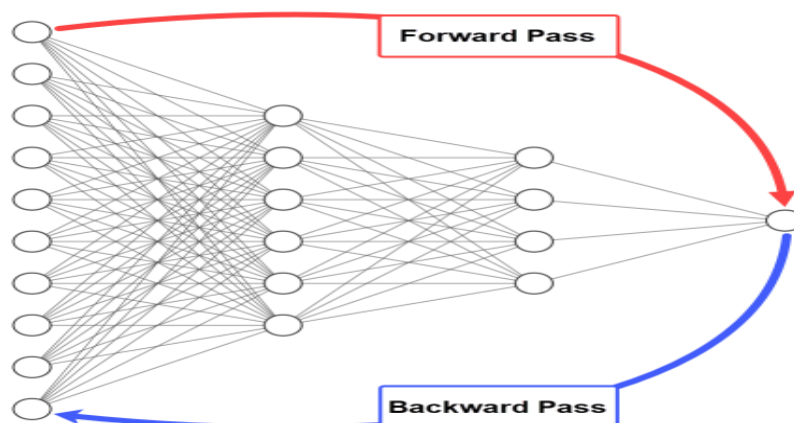


Figure 2: Image demonstrating the flow of forward and back propagation.

Source: <https://neptune.ai/blog/backpropagation-algorithm-in-neural-networks-guide>

2.13 LSTM

Long Short-Term Memory networks (LSTMs) are a specialized form of recurrent neural networks RNNs designed to tackle the vanishing gradient issue, thus enhancing the retention and retrieval of information over lengthy sequences. They were introduced by (Hochreiter & Schmidhuber, 1997) and been used widely in order to tackle a variety of problems. The core components of an LSTM unit include a cell state, hidden state, input gate, forget gate, and output gate.

The LSTM networks architecture

- **Hidden State:** The hidden state denoted as h , represents the short-term memory in the LSTM architecture. It captures information relevant to the current time step, serving as a form of short-term memory for the network. In the following figure, h_t also represents the output of the LSTM. Additionally, h_{t-1} signifies the hidden state from the previous time step.
- **The Input x_t :** The variable x_t typically represents the input at the current time step, providing information to the LSTM. It does not necessarily denote the last input but rather the input at the specific point in the sequence.
- **Cell State:** The cell state serves as the long term memory of the LSTM unit, allowing it to retain information over long sequences. It runs straight down the entire chain of LSTM units, facilitating the flow of information (Calzone, 2022). The cell state is basically the core of the concept of the LSTM network (Understanding LSTM Networks -- colah's blog, $\chi\chi$). In the following figure, it is represented as a straight line and gives the network the ability to store or forget information, controlled by structures called gates.

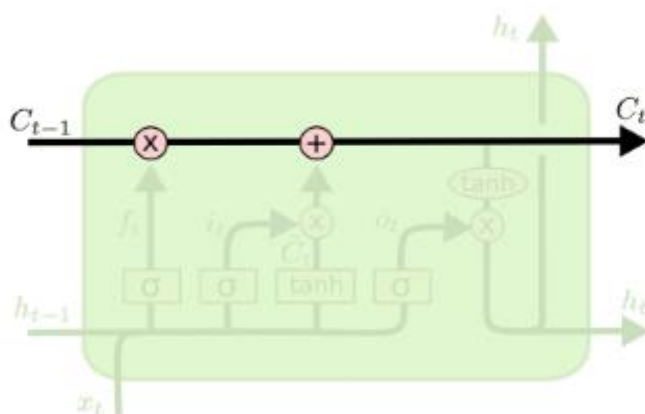


Figure 3: Image emphasizing the cell state of an LSTM network.

Source: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

- **Forget Gate Layer:** The forget gate manages what information is retained in the cell state and what information is discarded. This decision is made by the forget gate, which is essentially a sigmoid layer. The forget gate outputs a value between 0 and 1, where 1 signifies keeping the information, and 0 indicates complete removal of the information from the cell state.

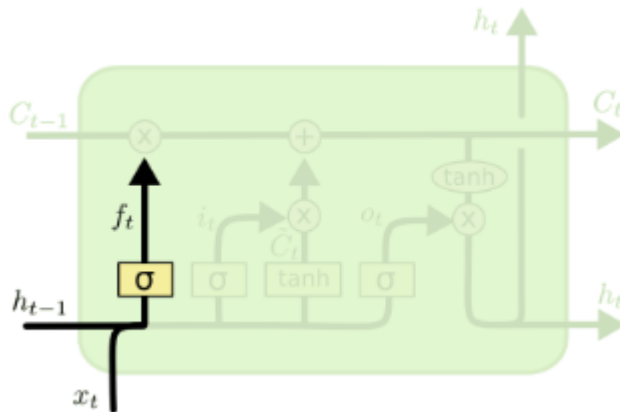


Figure 4: Image Emphasizing the forget gate of an Lstm network.

Source: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

- **Input Gate Layer:** The input gate consists of two parts. Firstly, a sigmoid function determines whether to input the data, producing a value of 0 or 1. This decision dictates whether the information is incorporated into the cell state. Secondly, a tanh function generates a vector containing the updates that will be added to the cell state.

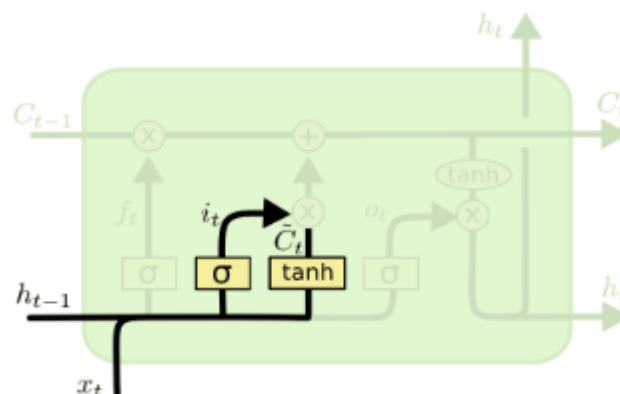


Figure 5: Image emphasizing the input gate of an LSTM network.

Source: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

- **Output Gate Layer:** The output gate is composed of two essential functions. Firstly, a sigmoid activation function determines the relevance of each element in the internal state to the final output, producing values between 0 and 1. This process signifies the significance of individual components within the internal

state for the output. Secondly, concurrently with the sigmoid activation, the current internal state undergoes a tanh activation function. This operation transforms the internal state values, constraining them to a range between -1 and 1. The resulting values act as candidate contributions to the final output. The combination of these two functions in the output gate facilitates the selective integration of relevant information from the internal state into the output.

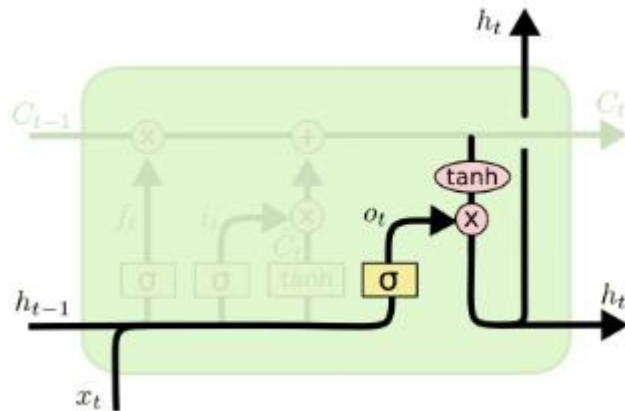


Figure 6: Image Emphasizing the output gate of an LSTM network

Source: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

This architecture enables LSTM units to effectively capture long-term dependencies in sequential data by selectively retaining and updating information over time.

Applications of the LSTM networks:

LSTM models have gained popularity across various domains due to their ability to model sequential data effectively. Some common applications include:

- **Natural Language Processing (NLP):** LSTMs are widely used in tasks such as sentiment analysis, machine translation, and text generation. They can effectively process sequences of words or characters to understand and generate human-like text.
- **Time Series Prediction:** LSTMs are well-suited for forecasting time series data, including applications such as predicting stock prices, weather forecasts, and patterns in passenger arrivals, as demonstrated in this dissertation. Leveraging historical observations, LSTMs excel at predicting future values, rendering them invaluable tools for various forecasting tasks.
- **Speech Recognition:** LSTMs are applied in speech recognition systems to transcribe spoken language into text. They analyze sequences of audio samples to recognize speech patterns and convert them into text.
- **Gesture Recognition:** LSTMs are utilized in gesture recognition systems, where they analyze sequences of hand movements to recognize gestures and commands. They play a crucial role in human-computer interaction and virtual reality applications.

Overall, LSTM models offer a powerful framework for capturing temporal dependencies in sequential data, making them versatile tools with applications in NLP, time series forecasting, speech recognition, gesture recognition, and beyond.

Chapter 3

3. Methodology

3.1 The Data

The data utilized in this dissertation were scraped from the Athens International Airport website, specifically from the statistics page. These data were subsequently transformed to fit the requirements of the experiment. This time series dataset is structured with monthly intervals, beginning in January 2002 and concluding in December 2022.

3.2 The Experimental Design

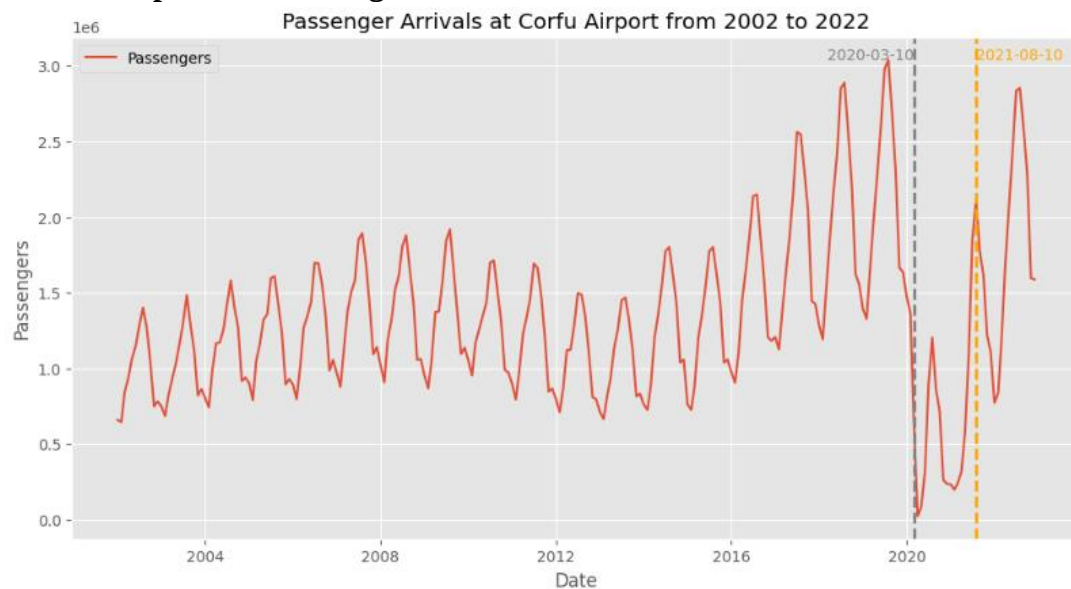


Figure 5: Line plot illustrating the annual passenger arrivals at Athens International Airport from the years 2002 to 2022. The graph also indicates the two dates chosen as separators for the creation of distinct training sets.

In this dataset, a noticeable anomaly occurs during the COVID-19 period when global flight activity came to a halt, primarily around March 2020. Our goal is to investigate how forecasting models respond when they encounter this anomaly. To do this, we will conduct two experiments: one where the anomaly is introduced during the forecasting phase, extending beyond March 1, 2020, and another where the anomaly is factored into the training phase, starting August 1, 2021. As can be seen in the following plot.

We monitored and assessed our models using three metrics: RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error). After each experiment, we recorded these metrics in a Pandas data frame to track and analyze their performance.

To enhance the performance of our SARIMA models, we explored two optimization algorithms:

- **AutoARIMA:**

AutoARIMA is an automated model selection tool designed for time series forecasting (Garg & Jindal, 2021). It systematically tests various combinations of ARIMA parameters and seasonality settings to identify the model with the best fit. Its automated approach streamlines the process of finding an optimal ARIMA model, particularly beneficial for users without extensive time series expertise.

- **GridSearch:**

Grid search is a hyperparameter tuning technique widely used in machine learning. It involves exhaustively searching through a predefined grid of hyperparameter values to find the combination that maximizes model performance. In the context of SARIMA models, grid search allows us to systematically explore different sets of parameters, ensuring we identify the most effective configuration for our specific forecasting task. During GridSearch the parameters that have been tested at the LSTM models are:

1. Number of LSTM layers
2. Units
3. Activation function
4. Batch Size
5. Epochs
6. Dropout
7. Learning rate

Autoarima was utilized exclusively with SARIMA models, while Grid Search was applied to both SARIMA and LSTM models to identify the best performers in each phase of the experiment.

In addition to the hyperparameter optimization algorithms, another technique, 'Rolling Forecast Origin,' was employed to enhance the efficiency of all ARIMA-based models. In this experiment, 'Rolling Forecast Origin' was implemented with an expanding window, which, while computationally intensive, was deemed manageable due to the small size of the dataset. This approach was chosen to enhance the model's forecasting efficiency. In order to better understand the concept In a Rolling Forecast Origin with an Expanding Window, a model's performance is tested over time by progressively increasing the training data's size. Initially, the model is trained on a set period, then makes a forecast for the following period. With each new forecast iteration, the training data includes one additional past observation, expanding the window, while the forecast horizon remains fixed, enhancing the model's accuracy by learning from more historical data (Shojaei κ.ά., 2018).

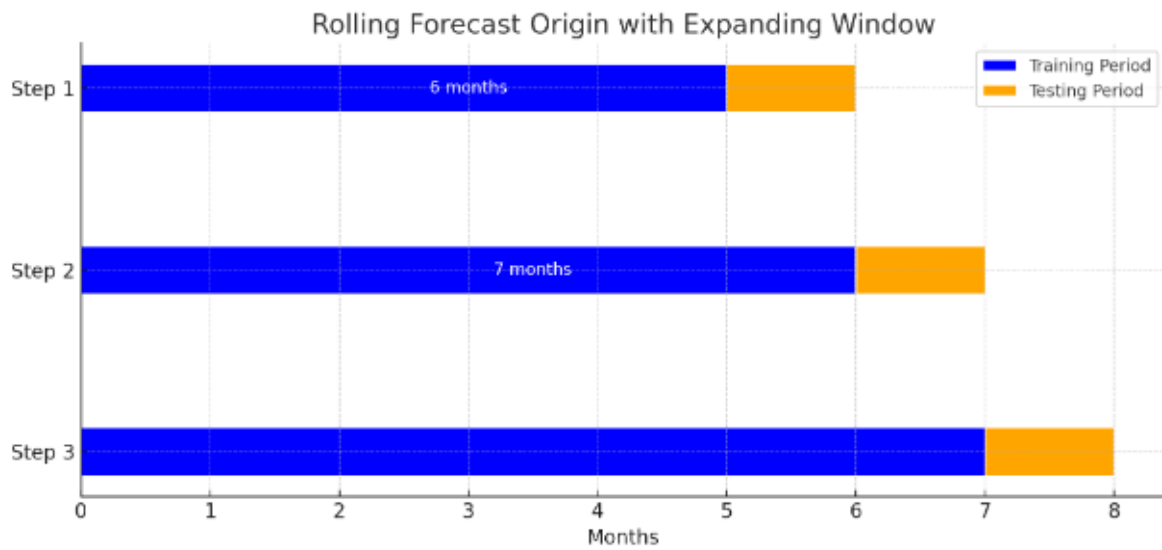


Figure 6: Figure illustrating the rolling forecast origin technique with a fixed window.

3.3 Data Analysis

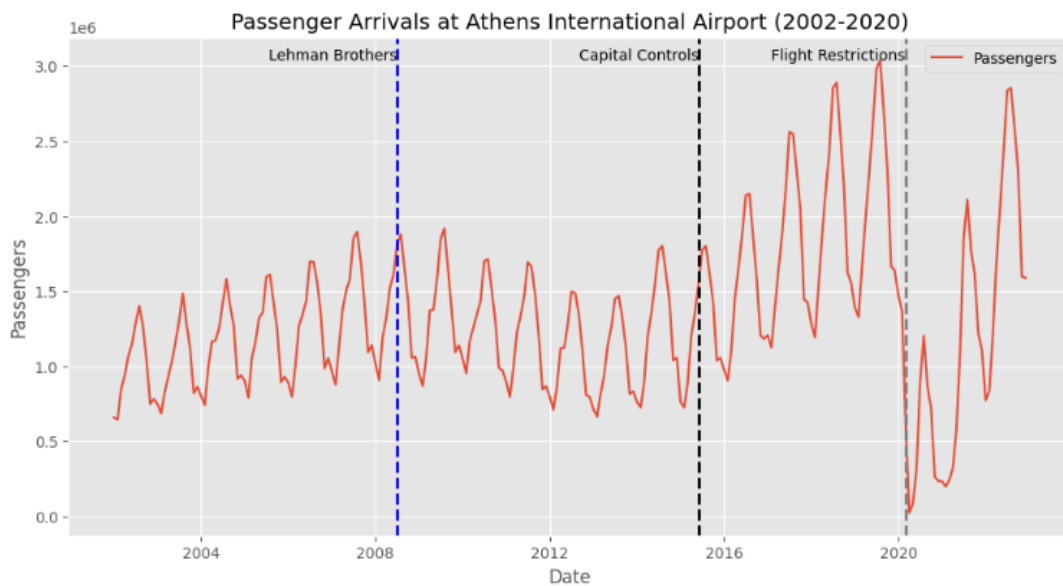


Figure 7: Line plot depicting the timeline of major crises with corresponding dates.

This dataset encompasses the period from January 2002 to December 2022 and describes monthly passenger arrivals at Athens International Airport. The dataset was compiled from reports available on their website. From 2002 to 2008, there is a noticeable, albeit modest, upward trend in passenger arrivals. However, in September 2008, the global financial crisis, triggered by the collapse of Lehman Brothers, significantly impacted this trend. The repercussions of this crisis were evident until the end of 2014 in many European countries and the USA. This impact is clearly reflected in the dataset, where a decline in passenger arrivals can be observed until the beginning of 2015.

According to the World Economic Forum's 2015 report, the 2008 global crisis led to a decrease in tourism arrivals in Greece. However, a gradual recovery commenced towards the end of 2013 (WEF_Global_Travel&Tourism_Report_2015.pdf, p. 160). Despite Greece undergoing another economic challenge, which began with the imposition of capital controls in June 2015, the data shows a continuing upward trend in passenger arrivals, persisting until the onset of the COVID-19 pandemic. The pandemic, which began in March 2020, resulted in widespread travel restrictions and a significant reduction in global air travel, with few exceptions. After the conclusion of the pandemic, when flights gradually resumed in 2021, there has been a steady increase in passenger arrivals. By 2022, these numbers were approaching the record highs seen in 2019. The year 2019 is esteemed by tourism experts as a standout year for the industry, often serving as a foundational benchmark for hospitality specialists to assess performance.

3.4 Statistical Testing

Before launching the experimental and forecasting stages, we conducted essential statistical tests and here are the results.

First, we plotted out the time series we are working on.

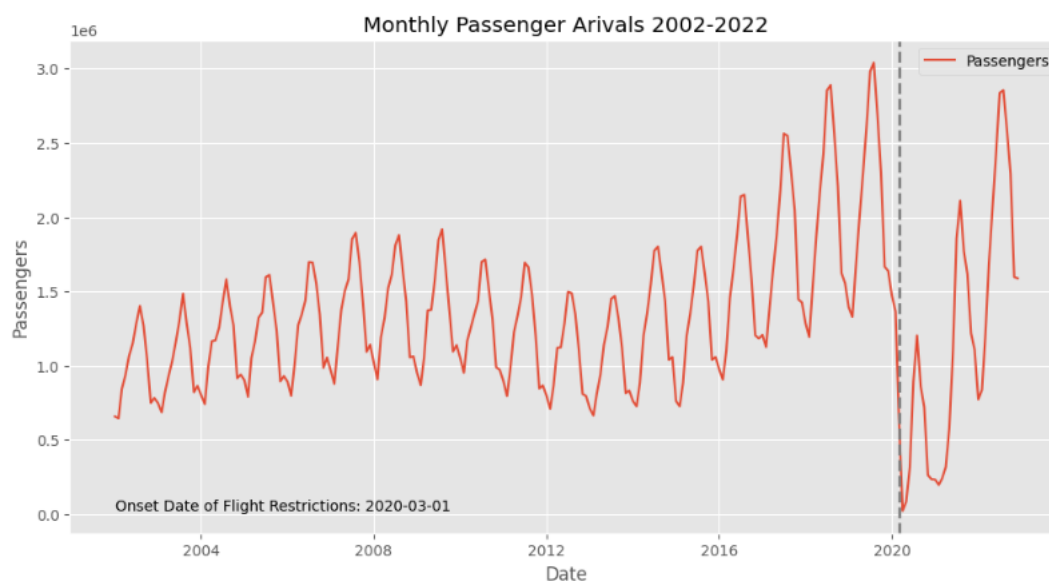


Figure 8: Line plot representing the first dataset earmarked for analysis and further work.

The anomaly in the dataset is due to the COVID-19 lockdown, which lasted almost two years and greatly reduced flight activity. This impact is visible in the graph, starting from the flight limitations in Greece on 2020-03-01.

Both the **Augmented Dickey Fuller (ADF)** test and the **Kwiatkowski-Phillips-Schmidt-Shin (KPSS)** test indicate that the data are not stationary. This can be attributed to the abnormal patterns observed during the pandemic period.

```

1. ADF : -3.5163983140208193
2. P-Value : 0.0075758036833884195
3. Num Of Lags : 13
4. Num Of Observations Used For ADF Regression: 238
5. Critical Values :
   1% : -3.458128284586202
   5% : -2.873761835239286
  10% : -2.5732834559706235

```

```

KPSS TEST
Test Statistic: 0.49482880236666826
P-value: 0.04283135081831796

```

Figure 9: ADF and KPSS results 1

To confirm that the non-stationarity of the dataset was due to the pandemic's impact on passenger arrivals, we excluded this period from the data and successfully achieved stationarity. In the KPSS test, the approach remained unchanged, while in the ADF test, we applied differencing with two lags.

```

1. ADF : -2.6614320986976674
2. P-Value : 0.08095325764960964
3. Num Of Lags : 11
4. Num Of Observations Used For ADF Regression: 204
5. Critical Values :
   1% : -3.462818103072159
   5% : -2.8758145932936805
  10% : -2.574378674548251

```

```

KPSS TEST
Test Statistic: 0.019493905267932727
P-value: 0.1

```

Figure 10: ADF and KPSS results 2

To enhance comprehension, Seasonal Trend Loess (STL) decomposition has been employed.

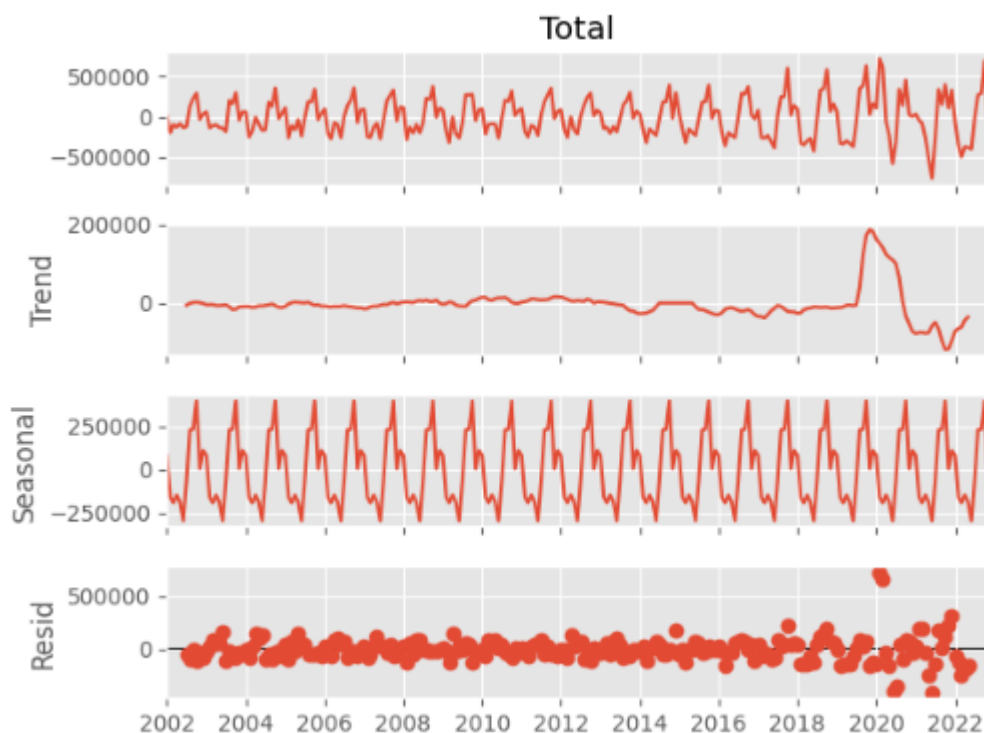


Figure 11: Seasonal trend decomposition plots

This STL decomposition, using first differencing, reveals that prior to the anomaly's introduction, the data is stationary with a distinct seasonal pattern recurring every 12

months. Additionally, after removing the trend and seasonal component, a substantial amount of residuals is evident during the pandemic period.

To summarize, since achieving stationarity in the dataset is hindered by the pandemic's impact, we will use the data as is. To identify the best models, we will employ Auto-Arima and grid search methods.

Chapter 4

4 The Experiment

4.1 Experiment Part One

As previously outlined, the experimental phase is divided into two segments. In this segment, none of the models has been exposed to the disruption brought about by the worldwide cessation of flights due to COVID-19 prior to commencing their forecasts.

To determine the best SARIMA models, we utilized two parameter optimization techniques: Autoarima and Gridsearch. Additionally, for fine-tuning the LSTM models, we employed Grid Search.

The models that have been Selected from the two algorithms can be found in the next figure.

Auto_Arima_1

Dep. Variable:	Total		No. Observations:				
Model:	ARIMA(2, 1, 4)x(1, 1, [1], 12)		Log Likelihood		-2544.		
Date:	Sun, 07 Jan 2024		AIC		5106.		
Time:	18:50:26		BIC		5136.		
Sample:	01-01-2002		HQIC		5118.		
	- 02-01-2020						
Covariance Type:	opg						
	coef	std err	z	P> z	[0.025	0.975]	
ar.L1	1.7242	0.013	135.706	0.000	1.699	1.749	
ar.L2	-0.9920	0.010	-95.177	0.000	-1.012	-0.972	
ma.L1	-1.8557	0.064	-28.854	0.000	-1.982	-1.730	
ma.L2	1.3507	0.153	8.812	0.000	1.050	1.651	
ma.L3	-0.3501	0.175	-2.001	0.045	-0.693	-0.007	
ma.L4	0.0822	0.097	0.850	0.395	-0.107	0.272	
ar.S.L12	0.3938	0.114	3.442	0.001	0.170	0.618	
ma.S.L12	-0.8693	0.096	-9.057	0.000	-1.057	-0.681	
sigma2	4.088e+09	2.05e-11	2e+20	0.000	4.09e+09	4.09e+09	
Ljung-Box (L1) (Q):	1.02	Jarque-Bera (JB):	21.41				
Prob(Q):	0.31	Prob(JB):	0.00				
Heteroskedasticity (H):	1.19	Skews:	-0.31				
Prob(H) (two-sided):	0.47	Kurtosis:	4.46				

Grid_Search_1

Dep. Variable:	Total		No. Observations:		218		
Model:	ARIMA(0, 1, 0)x(1, 0, 0, 12)		Log Likelihood		-2780.347		
Date:	Sun, 07 Jan 2024		AIC		5564.695		
Time:	18:55:31		BIC		5571.455		
Sample:	01-01-2002		HQIC		5567.425		
	- 02-01-2020						
Covariance Type:	opg						
	coef	std err	z	P> z	[0.025	0.975]	
ar.S.L12	0.9364	0.009	100.304	0.000	0.918	0.955	
sigma2	4.533e+09	7.6e-13	5.97e+21	0.000	4.53e+09	4.53e+09	
Ljung-Box (L1) (Q):	1.86	Jarque-Bera (JB):	3247.95				
Prob(Q):	0.17	Prob(JB):	0.00				
Heteroskedasticity (H):	0.66	Skews:	-2.37				
Prob(H) (two-sided):	0.07	Kurtosis:	21.35				

Figure 12: Grid Search and AutoArima results 1

In both datasets, the models that performed the best, as identified through GridSearch, are consistently **LSTM_1** and **LSTM_2**.

Folowing are the parammeters of the best performing model:

- num_lstm_layers = 3
- units = 16
- activation = "relu"
- batch_size= 64
- epochs= 50
- dropout= 0
- learning_rate = 0.0001

In this phase, it's important to acknowledge that the extensive training conducted would not have been feasible on my laptop. Therefore, I would like to express my gratitude to Google, and specifically to their Kaggle service, as well as the entire community. Their contributions have been a tremendous inspiration throughout this process.

Results:

	Metric	Auto_Arima_1	Grid_Search_1	LSTM_1
0	MAE	1173648.65	971686.84	380355.09
1	RMSE	1333561.22	1197309.31	445109.04
2	MAPE	532.49	500.25	104.31

Table 1: Phase one of the experiment: Initial results

The results clearly indicate that among the two SARIMA methods, the Grid Search tuned approach consistently excels in all metrics. However, the deep learning model LSTM notably outperforms both ARIMA methods, highlighting its superior ability in generalizing, which is key for effective predictive modeling.

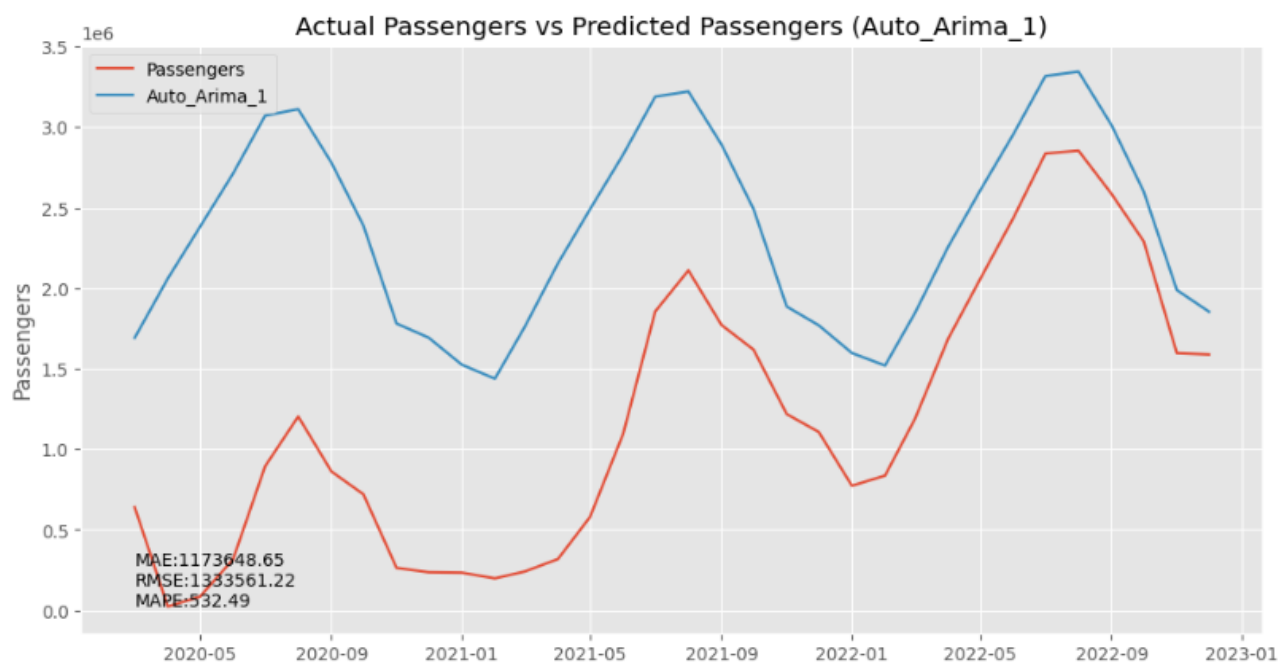


Figure 14: Phase one LSTM_1 results vs predicted

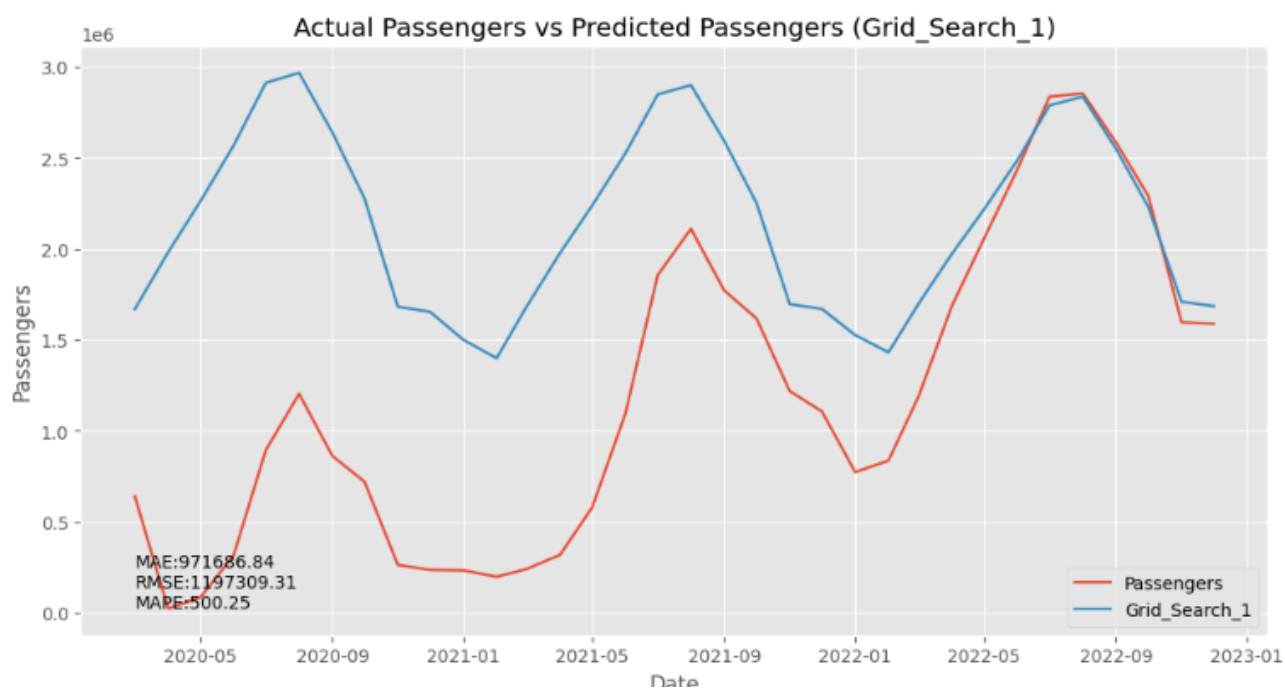


Figure 13: Phase one Grid_Search_1 results vs predicted

To enhance efficiency, the technique of Rolling Forecast Origin was applied to the SARIMA models.

Results:

	Metric	Auto_Arima_1	Auto_Arima_1_R	Grid_Search_1	Grid_Search_1_R	LSTM_1
0	MAE	1173648.65	200863.91	971686.84	163600.62	380355.09
1	RMSE	1333561.22	306018.46	1197309.31	233388.17	445109.04
2	MAPE	532.49	162.11	500.25	89.5	104.31

Table 2: Phase one of the experiment: Final results

Following the application of the Rolling Forecast Origin technique, we observed a significant improvement in the metrics of both the Autoarima-tuned and Grid Search models. Notably, the Grid Search model even surpassed the performance of the Deep Learning LSTM model. An important observation is that although LSTM_1 performed better than Auto_Arima_1_R in terms of results, when we consider the MAPE metric, Auto_Arima_1_R outperforms LSTM_1 in both MAE and RMSE. This discrepancy can be attributed to how these metrics penalize errors. MAE and RMSE give more weight to larger errors, influencing the overall performance, while MAPE emphasizes the relative percentage errors. Auto_Arima_1_R's success in MAE and RMSE indicates its proficiency in minimizing absolute errors, even though LSTM_1 had an edge in other metrics.

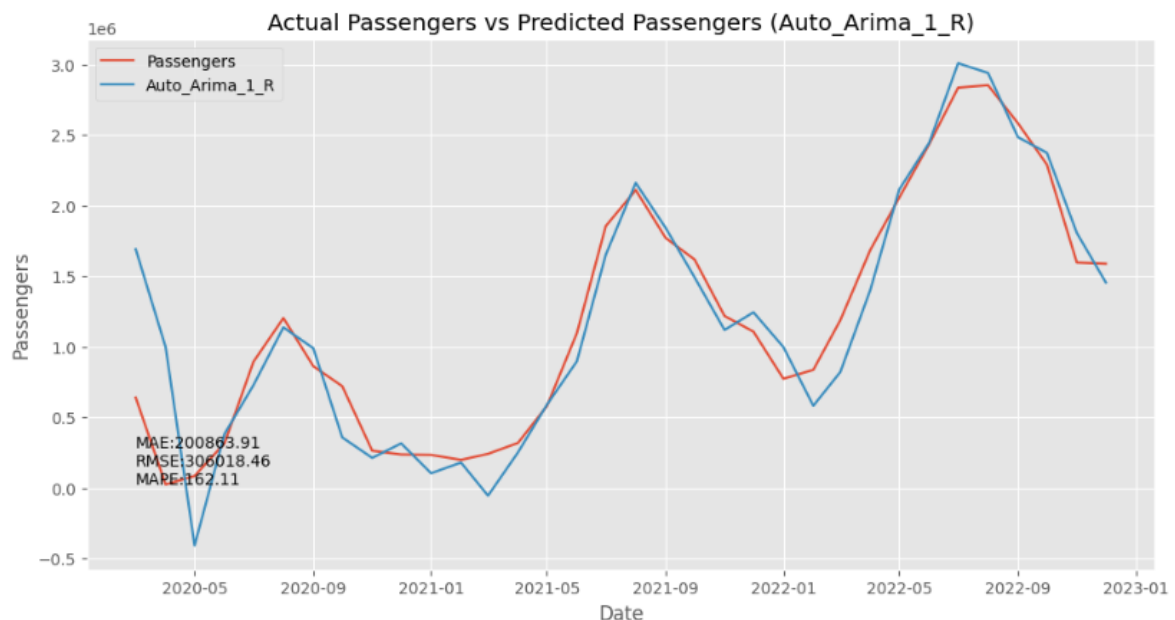


Figure 14: Auto_Arima_1 after the use of the rolling forecast origin technique.

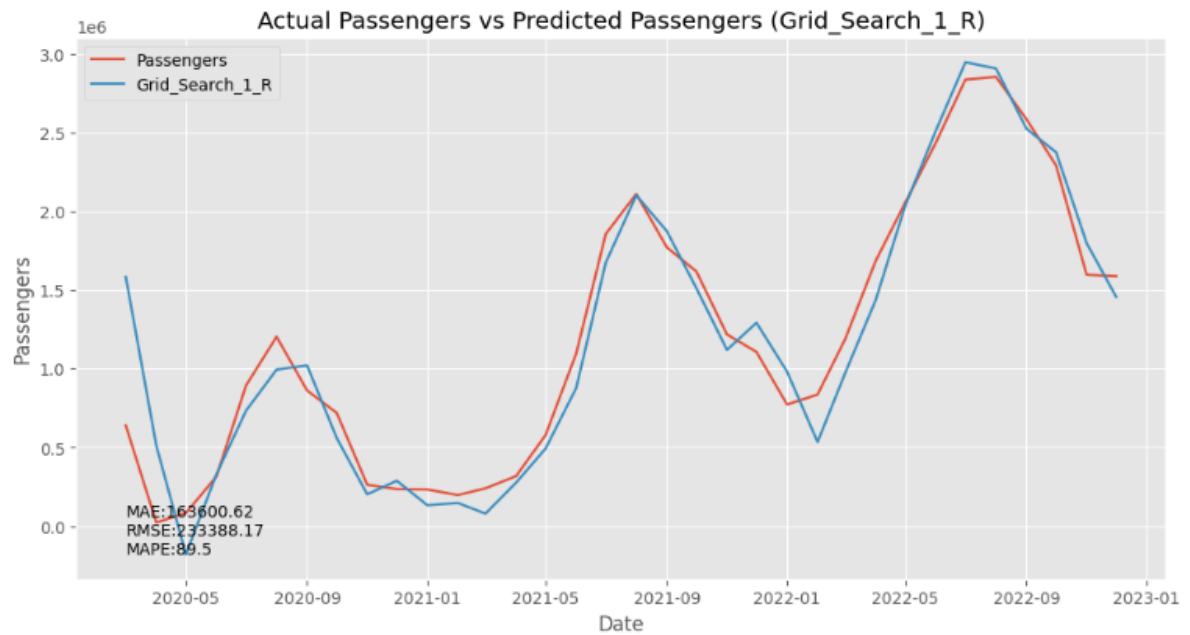


Figure 15: Grid_Serach_1 after the use of the rolling forecast origin technique.

4.2 Experiment Part Two

In the second phase of the experiment, the models were exposed to a portion of the anomaly during their training phase. The algorithms and techniques employed remained identical to those used in the first phase, with the sole distinction being the inclusion of the anomalous event within the training data.

The models selected by the two algorithms are presented in the following figure.

Auto_Arima_2						
Dep. Variable:	Total		No. Observations:		235	
Model:	ARIMA(4, 0, 5)x(1, 0, [1], 12)		Log Likelihood		-3079.632	
Date:	Sun, 07 Jan 2024		AIC		6185.265	
Time:	19:01:45		BIC		6230.239	
Sample:	01-01-2002		HQIC		6203.396	
	- 07-01-2021					
Covariance Type:	opg					
	coef	std err	z	P> z	[0.025	0.975]
const	1.294e+06	5.4e-08	2.4e+13	0.000	1.29e+06	1.29e+06
ar.L1	0.7211	0.157	4.590	0.000	0.413	1.029
ar.L2	0.2923	0.102	2.862	0.004	0.092	0.492
ar.L3	0.8192	0.113	7.239	0.000	0.597	1.041
ar.L4	-0.8782	0.112	-7.830	0.000	-1.098	-0.658
ma.L1	0.6567	0.210	3.169	0.002	0.254	1.079
ma.L2	0.2924	0.257	1.138	0.255	-0.211	0.796
ma.L3	-0.8885	0.296	-2.999	0.003	-1.469	-0.308
ma.L4	-0.4546	0.256	-1.778	0.075	-0.956	0.047
ma.L5	-0.2439	0.198	-1.234	0.217	-0.631	0.144
ar.S.L12	0.9477	0.070	13.552	0.000	0.811	1.085
ma.S.L12	-0.5385	0.266	-2.028	0.043	-1.059	-0.018
sigma2	2.249e+10	2.86e-11	7.88e+20	0.000	2.25e+10	2.25e+10
Ljung-Box (L1) (Q): 0.05 Jarque-Bera (JB): 6930.05						
Prob(Q): 0.82 Prob(JB): 0.00						
Heteroskedasticity (H): 4.15 Skew: -3.26						
Prob(H) (two-sided): 0.00 Kurtosis: 28.79						

Grid_Search_2						
SARIMAX Results						
Dep. Variable:		Total		No. Observations:		235
Model:		ARIMA(2, 1, 2)		Log Likelihood		-3139.044
Date:		Sun, 07 Jan 2024		AIC		6288.089
Time:		19:04:44		BIC		6305.366
Sample:		01-01-2002		HQIC		6295.055
		- 07-01-2021				
Covariance Type:		opg				
	coef	std err	z	P> z	[0.025	0.975]
ar.L1	1.7329	0.007	265.968	0.000	1.720	1.746
ar.L2	-0.9961	0.007	-141.196	0.000	-1.010	-0.982
ma.L1	-1.6803	0.039	-43.467	0.000	-1.756	-1.605
ma.L2	0.9408	0.040	23.692	0.000	0.863	1.019
sigma2	3.312e+10	2.56e-13	1.29e+23	0.000	3.31e+10	3.31e+10
Ljung-Box (L1) (Q):		5.26	Jarque-Bera (JB):		400.89	
Prob(Q):		0.02	Prob(JB):		0.00	
Heteroskedasticity (H):		2.84	Skew:		-1.06	
Prob(H) (two-sided):		0.00	Kurtosis:		9.05	

Figure 16: Grid Search and AutoArima results 2

Results:

	Metric	Auto_Arima_2	Grid_Search_2	LSTM_2
0	MAE	334499.76	421346.17	282314.26
1	RMSE	388652.19	518093.44	342056.46
2	MAPE	22.15	22.32	16.97

Table 3: Phase two of the experiment: Initial results

The results from this phase suggest that there is no distinct superior performer between the Autoarima algorithm and Gridsearch, as they both yielded almost identical outcomes, with the Autoarima version being marginally better. However, the LSTM model, optimized with Grid Search, outperforms the other two models.

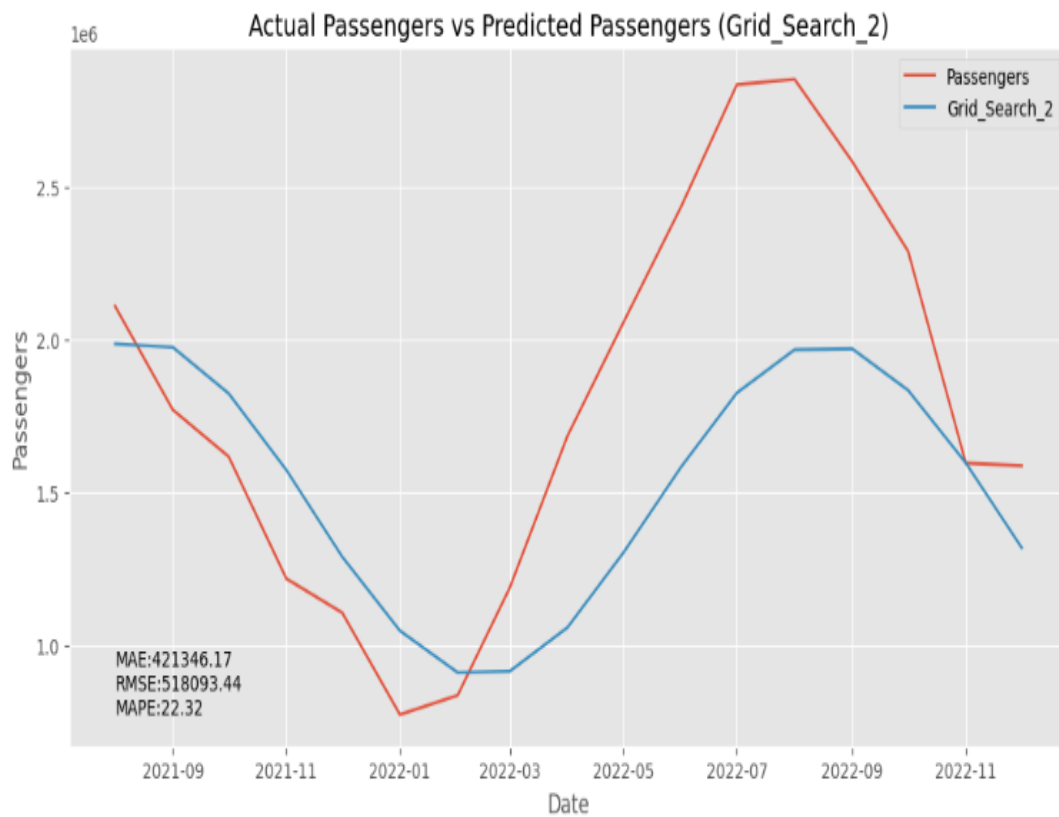


Figure 17: Phase two Grid_Search_2 results vs predicted

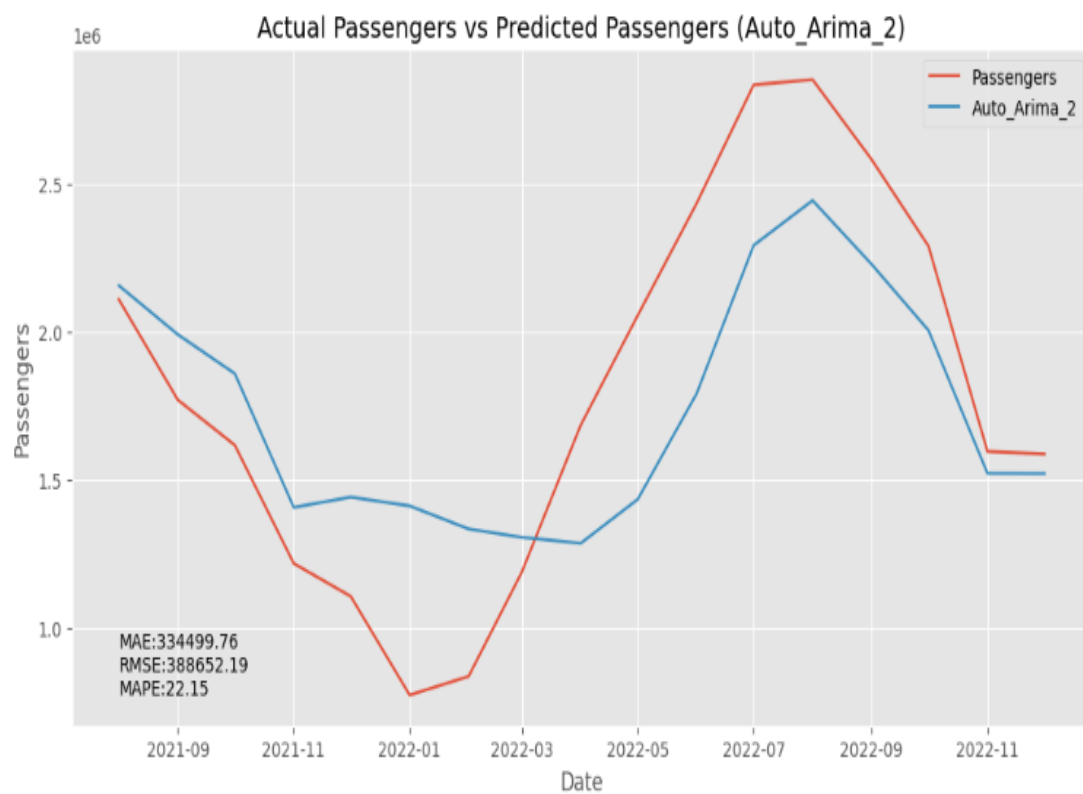


Figure 18: Phase two Auto_Arima_2 results vs predicted

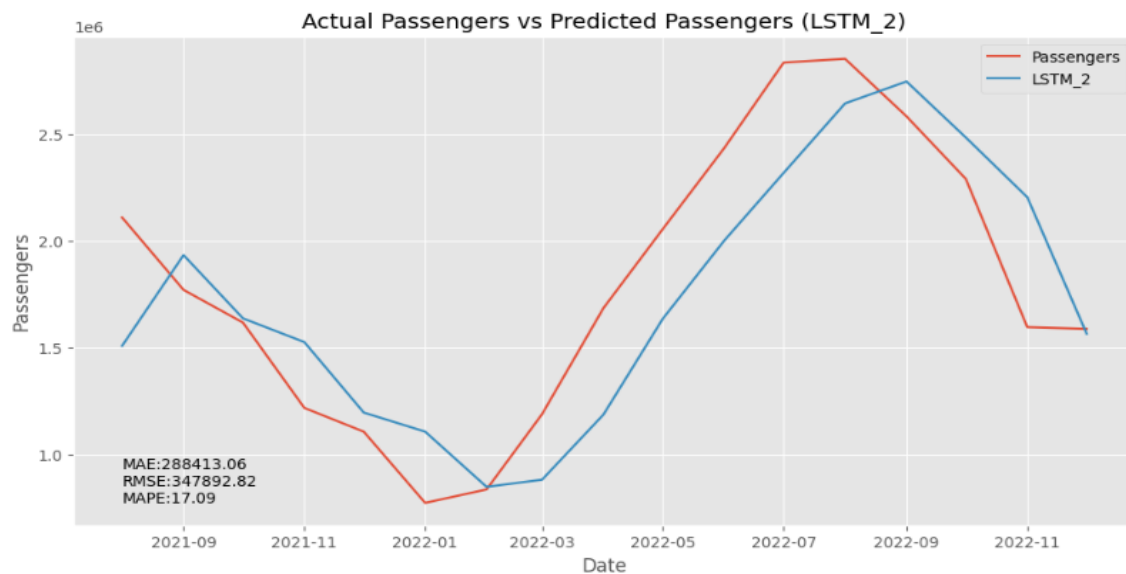


Figure 19: Phase two LSTM_2 results vs predicted

To improve efficiency, the Rolling Forecast Origin technique was implemented for the SARIMA models.

	Metric	Auto_Arima_2	Auto_Arima_2_R	Grid_Search_2	Grid_Search_2_R	LSTM_2
0	MAE	334499.76	142581.37	421346.17	123660.2	282314.26
1	RMSE	388652.19	178172.31	518093.44	152134.1	342056.46
2	MAPE	22.15	11.13	22.32	9.77	16.97

Table 4 : Phase two of the experiment: Final results

After applying the Rolling Forecast Origin technique, there was a marked enhancement in the performance metrics of both the Autoarima-tuned and Grid Search SARIMA models. Remarkably, the Grid Search model not only showed improvement but also significantly outperformed the Deep Learning LSTM model across all metrics.

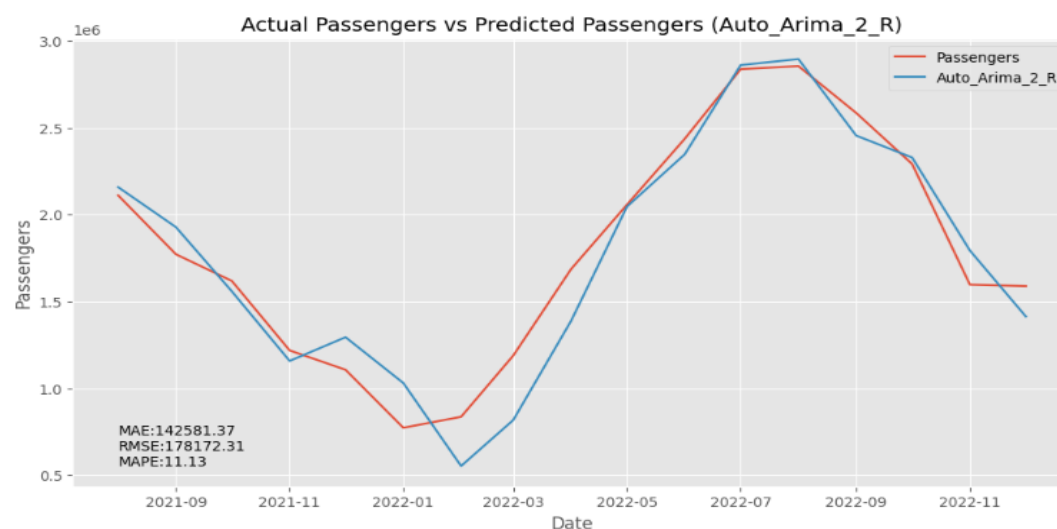


Figure 20: Auto_Arima_2 after the use of the rolling forecast origin technique.

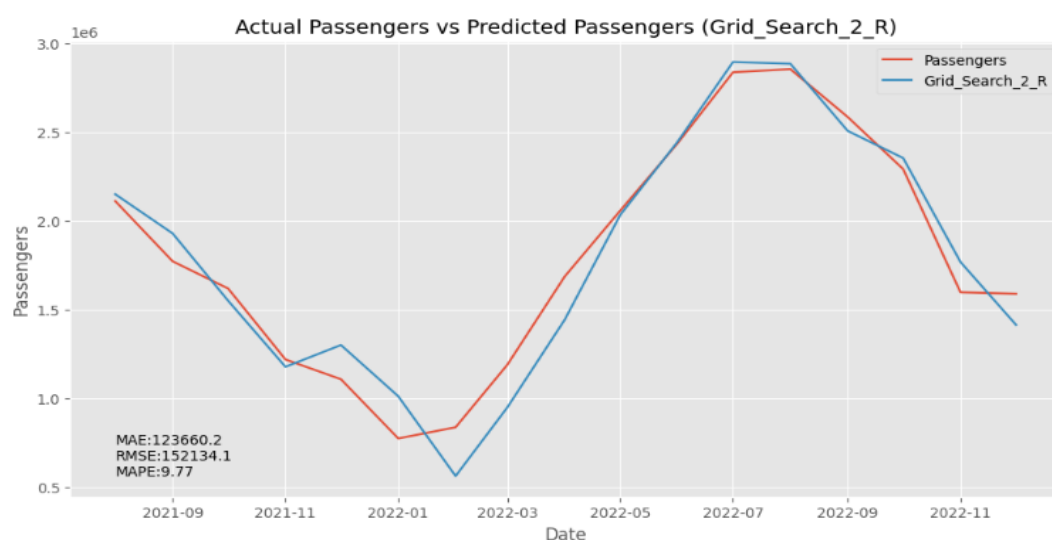


Figure 21: Grid_Serach_2 after the use of the rolling forecast origin technique

	Metric	Auto_Arima_1	Auto_Arima_1_R	Grid_Search_1	Grid_Search_1_R	Auto_Arima_2	Auto_Arima_2_R	Grid_Search_2	Grid_Search_2_R	LSTM_1	LSTM_2
0	MAE	1173648.65	200863.91	971686.84	163600.62	334499.76	142581.37	421346.17	123660.2	399461.29	282314.26
1	RMSE	1333561.22	306018.46	1197309.31	233388.17	388652.19	178172.31	518093.44	152134.1	470987.38	342056.46
2	MAPE	532.49	162.11	500.25	89.5	22.15	11.13	22.32	9.77	113.82	16.97

Table 5: Table with all the final results

Chapter 5

5 Final Results

5.1 The Results From Phase One.

- 1) The LSTM approach demonstrated significantly better performance compared to other methods before the application of the rolling forecast origin.
- 2) Among the two optimization algorithms used for SARIMA models, Grid Search yielded the best-performing model.
- 3) The application of the Rolling Forecast Origin technique to both the Grid Search tuned and AutoArima tuned models resulted in a tremendous increase in efficiency.
- 4) After implementing the Rolling Forecast Origin technique, the Grid Search tuned model outperformed all other models evaluated in part one of the experiment.
- 5) The LSTM_1 model has demonstrated really good performance in handling non-linear data and exhibited better generalization capabilities, despite the dataset being relatively small.

5.2 The Results From Phase Two.

- 1) The LSTM approach demonstrated significantly better performance compared to other methods before the application of the Rolling Forecast Origin technique.
- 2) Among the two optimization algorithms used for SARIMA models, both yielded fairly similar results prior to the application of the Rolling Forecast Origin technique.
- 3) The use of the Rolling Forecast Origin technique with the Grid Search and AutoArima tuned models significantly improved their efficiency.
- 4) After the application of the Rolling Forecast Origin technique, the model tuned using Grid Search outperformed all other models assessed in the second part of the experiment.
- 5) The LSTM approach is in harmony with existing literature, as evidenced by the model's commendable performance in scenarios where data exhibits non-linear patterns, especially when introduced to crisis data. This alignment with established findings underscores the robustness and adaptability of the LSTM model in effectively handling complex and dynamic datasets.

5.3 Comparing And Summarizing The Results From Both Phases.

- 1) It can be concluded, as suggested in the literature, that traditional time series models like the SARIMA models used in this study can yield quite accurate predictions in small datasets, often outperforming more sophisticated deep learning models.
- 2) The LSTM models have shown accuracy in predicting anomalies due to their proficiency with nonlinear data. However, as observed in the second phase of the experiment, their performance improves with larger datasets.
- 3) In both phases of the experiment, the application of the Rolling Forecast Origin technique significantly enhanced the performance of the SARIMA models.
- 4) The fixed window approach within the rolling forecast technique is computationally intensive and is generally more suitable for smaller datasets.

Chapter 6

6 Conclusions

In conclusion, this dissertation confirms the alignment of its findings with established literature. The comparison between ARIMA-based and LSTM models underscores the superior performance of ARIMA-based models in smaller datasets and short-term prediction tasks. A significant observation emphasizes the pivotal role of metric selection in influencing benchmarking outcomes, underscoring the need for a meticulous approach in metric choice during the design of experimental procedures.

The implications of these findings extend to fields where passenger arrival volume plays a critical role, such as the day-to-day operations of airports. In such contexts, the identified models offer a practical and efficient means to enhance overall performance. Leveraging the strengths of ARIMA-based models, especially in scenarios with limited

data and computational resources, and recognizing the crucial role of chosen metrics empower stakeholders in various sectors to make informed decisions and optimize predictive modeling.

In summary, this study not only contributes to the existing body of knowledge but also provides actionable insights for practitioners. The adoption of ARIMA-based models and heightened awareness of metric significance offer a clear pathway to more effective decision-making in domains where accurate prediction of outcomes, particularly related to passenger arrival volume, is indispensable for operational success.

References

- AIA/Management Information System, A. I. S. (2023). *ATHENS INTERNATIONAL AIRPORT PASSENGER TRAFFIC DEVELOPMENT 2023*. AIA/Management Information System. https://www.aia.gr/userfiles/675393df-ab1a-4b77-826c-f3096a3d7f12/Passenger-Traffic_december-2023.pdf
- Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716–723.
<https://doi.org/10.1109/TAC.1974.1100705>
- Asteriou, D., & Hall, S. G. (2011). *Applied econometrics* (2. ed). Palgrave Macmillan.
- Buhalis, D., Leung, X. Y., Fan, D., Darcy, S., Chen, G., Xu, F., Wei-Han Tan, G., Nunkoo, R., & Farmaki, A. (2023). Editorial: Tourism 2030 and the contribution to the sustainable development goals: the tourism review viewpoint. *Tourism Review*, 78(2), 293–313. <https://doi.org/10.1108/TR-04-2023-620>
- Calzone, O. (2022, Απρίλιος 10). An Intuitive Explanation of LSTM. *Medium*.
<https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c>
- Chan, C. K., Witt, S. F., Lee, Y. C. E., & Song, H. (2010). Tourism forecast combination using the CUSUM technique. *Tourism Management*, 31(6), 891–897. Scopus. <https://doi.org/10.1016/j.tourman.2009.10.004>
- Do, Q. H., Lo, S.-K., Chen, J.-F., Le, C.-L., & Anh, L. H. (2020). Forecasting Air Passenger Demand: A Comparison of LSTM and SARIMA. *Journal of Computer Science*, 16(7), 1063–1084.
<https://doi.org/10.3844/jcssp.2020.1063.1084>
- Elton, C. (2022, Μάιος 25). *Schiphol airport chaos continues but end in sight as workers cancel strike action*.

- <https://www.euronews.com/travel/2022/05/25/long-queues-remain-at-schiphol-airport-after-hellish-weekend-for-travellers>
- Frechtling, D. C. (2001). *Forecasting tourism demand: Methods and strategies*. Butterworth-Heinemann.
- Garg, S., & Jindal, H. (2021). Evaluation of Time Series Forecasting Models for Estimation of PM2.5 Levels in Air. *2021 6th International Conference for Convergence in Technology (I2CT)*, 1–8.
<https://doi.org/10.1109/I2CT51068.2021.9418215>
- Guo, X., Grushka-Cockayne, Y., & De Reyck, B. (2022). Forecasting Airport Transfer Passenger Flow Using Real-Time Data and Machine Learning. *Manufacturing & Service Operations Management*, 24(6), 3193–3214.
<https://doi.org/10.1287/msom.2021.0975>
- Hitchcock, C. (2023). Causal Models. Στο E. N. Zalta & U. Nodelman (Επιμ.), *The Stanford Encyclopedia of Philosophy* (Spring 2023). Metaphysics Research Lab, Stanford University.
<https://plato.stanford.edu/archives/spr2023/entries/causal-models/>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural computation*, 9, 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- INSETE intelligence. (2020). *Tourism's contribution to the Greek Economy 2020-2021*. INSETE intelligence. https://insete.gr/wp-content/uploads/2022/09/2020-2021_EN.pdf
- Kazda, A., Badanik, B., & Serrano, F. (2022). Pandemic vs. Post-Pandemic Airport Operations: Hard Impact, Slow Recovery. *Aerospace*, 9(12), Article 12.
<https://doi.org/10.3390/aerospace9120810>

- Kim, S., & Shin, D. (2016). Forecasting short-term air passenger demand using big data from search engine queries. *Automation in Construction*, 70.
<https://doi.org/10.1016/j.autcon.2016.06.009>
- Law, R., Li, G., Fong, D. K. C., & Han, X. (2019). Tourism demand forecasting: A deep learning approach. *Annals of Tourism Research*, 75, 410–423.
<https://doi.org/10.1016/j.annals.2019.01.014>
- Lim, C. (1997). Review of international tourism demand models. *Annals of Tourism Research*, 24(4), 835–849. [https://doi.org/10.1016/S0160-7383\(97\)00049-2](https://doi.org/10.1016/S0160-7383(97)00049-2)
- M., M., S., M., & A., A. (2017). Airline Passenger Forecasting in EGYPT (Domestic and International). *International Journal of Computer Applications*, 165(6), 1–5. <https://doi.org/10.5120/ijca2017913113>
- National Bank of Greece, N. B. of G. (2024). *Business Trends: Tourism (Q4:2023)*. NBG. <https://www.nbg.gr/en/group/studies-and-economic-analysis/reports/tourism-2023q4>
- Neath, A., & Cavanaugh, J. (2012). The Bayesian information criterion: Background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4. <https://doi.org/10.1002/wics.199>
- Profillidis, V. (2012). An ex-post assessment of a passenger demand forecast of an airport. *Journal of Air Transport Management*, 25, 47–49.
<https://doi.org/10.1016/j.jairtraman.2012.08.002>
- Scarpel, R. (2013). Forecasting air passengers at São Paulo International Airport using a mixture of local experts model. *Journal of Air Transport Management*, 26, 35–39. <https://doi.org/10.1016/j.jairtraman.2012.10.001>

Schultz, M., & Reitmann, S. (2019). Machine learning approach to predict aircraft boarding. *Transportation Research Part C Emerging Technologies*, 98.

<https://doi.org/10.1016/j.trc.2018.09.007>

Shojaei, A., Izadi Moud, H., & Flood, I. (2018). *Forecasting Transportation Project Frequency using Multivariate Regression with Elastic Net Regularization*.