NOVA IMS Information Management School

MDSAA

Master's Degree Program in

Data Science and Advanced Analytics

SHIPPING VOLUME FORECASTING IN AN INTERNATIONAL LIFESTYLE COMPANY

Comparing Time Series Forecasting techniques

João Maria Cardoso Nogueira da Silva

Master's Thesis

presented as partial requirement for obtaining the Master Degree Program in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

SHIPPING VOLUME FORECASTING IN AN INTERNATIONAL LIFESTYLE COMPANY

by

João Nogueira da Silva

Master's Thesis presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics, with a Specialization in Business Analytics

Supervised by

Roberto Pereira Henriques

May, 2024

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Rules of Conduct and Code of Honor from the NOVA Information Management School.

Lisbon, 27-05-2024

ACKNOWLEDGMENTS

Firstly, I would like to thank my supervisor Roberto Henriques, for all the support during the development of this thesis.

Secondly, I would like to thank both my parents for providing me the opportunities to complete my studies and all the love and care during these challenging years of my student life.

I would also like to thank all my friends, especially Carla, Beatriz, David and Tomás, for all the help and beautiful moments spent together in the last years of the master's degree.

Finally, I would like to thank my girlfriend Madalena, for all she has done and endured during this time, and all the emotional support and comprehension.

ABSTRACT

In today's world, it is necessary to utilize all available tools to improve our productivity and make accurate decisions. Forecasting product sales has been a common theme within the world of supply chain management, particularly in the apparel and lifestyle industry. Although many organizations have already transitioned to more advanced forecasting systems and techniques, some still rely on internal, intuition-based approaches to estimate the amount of products being shipped and sold each year. This project focuses on offering a time series forecasting approach as a solution to such a company. Based on previous work done in Sales forecasting, this work aims to apply similar methodologies to forecasting a similar variable, Units Shipped. This analysis was done on both Footwear and Apparel products, where the aim was not only to provide a reliable forecasting system but also to test the assumption that deep learning techniques outperform other types of time series forecasting techniques such as ARIMA or Prophet. Besides the previously mentioned models, an LSTM, a stacked two-layer LSTM and a CNN network were implemented. For Footwear products, the CNN model performed best with an RMSE of 18060.33, and for Apparel, the best performing models were the 1-layer LSTM and the CNN. Although satisfactory results were achieved for Footwear products, shipped units predicted for Apparel were significantly off, with MAPE scores of around 80%. As expected, deep learning algorithms performed best in this predictive analysis. Despite achieving poor results, it is believed that should the company opt for a time series forecasting approach, both LSTM and CNN models will perform best, providing a more complex model architecture and a bigger observation pool.

KEYWORDS

Shipping Volume; Time Series Forecasting; Deep Learning; Lifestyle Apparel Industry; Machine Learning

Sustainable Development Goals (SDG)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

TSF	Time Series Forecasting
FT	Footwear products
AP	Apparel Products
ACF	Auto-Correlation Function
PACF	Partially Auto-Correlation Function
ADF	Augmented Dickey-Fueller
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
ARIMA	Auto-Regressive Integrated Moving Average
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
EDA	Exploratory Data Analysis

1. INTRODUCTION

In today's fast-paced business landscape, it's essential to tackle the challenges that arise and try to use the best and most appropriate technological tools to save time, plan resources, and make better decisions. In the past, forecasting was often frowned upon until researchers started being recognized for its practice, with some of them earning relevant awards in the area of Economics. There are numerous examples throughout history that demonstrate this concept, specifically the reduction of inaccuracies in election surveys or the enhancement of long-term flight scheduling and climate predictions (Armstrong, 2001).

This pattern regarding forecasting and its evolution is no exception in the realm of lifestyle, apparel, or fashion business. Forecasting is indeed one of the critical topics in the vast world of information systems and takes a lead role in the fashion apparel industry (Choi et al., 2013).

According to the OEC Ranking[1], in 2022, the non-knitted clothing and footwear industries have a total trade value of about 430 billion dollars and represent about 1.8% of the world's trade, reinforcing the industry's importance and dimension.

The clothing sales market has a lot of specific factors that render this type of predictions an extremely delicate and complex issue, since it not only takes into account the production processes but also the seasonality of sales data, fashion trends, and the different singularities of a clothing item itself like the colour, size or design (Thomassey, 2010).

Despite this complexity however, for some years, regression, time series and Machine learning models are and were successfully used to predict sales in this particular industry, (Thomassey, 2010)with the most common methods being ARIMA, SARIMA, exponential smoothing, regression, Box & Jenkins and Holt Winters (Loureiro et al., 2018).

In the retail business the decision-making process regarding the application of forecasting methods is largely driven by companies, however in the technological era in which we live it is expected that most companies are actively engaged in automating and innovating their current planning processes. Meanwhile, there are still organizations that have yet to integrate forecasting methods into critical operational processes (Yaremko et al., 2022).

[1] Ranking can be consulted in https://oec.world/en/profile/hs/non-knitted-clothing-accesories?redirect=true for non-knitted clothing and in https://oec.world/en/profile/hs/footwear-1264?redirect=true#product-complexity for footwear

This study focuses on such a company, which based their decision of the number of units shipped each month in intuition, market knowledge and client preferences depending on which territory the goods would be shipped to.

In this process there was no use of forecasting tools in the number of products that were allocated to each customer, resulting in inconsistencies in the orders and leftover stock. Thus, by gathering product historical data from the company under study, this thesis aims to predict gross shipped units (GSU).

To the best of our knowledge, it was concluded that the shipping volume is a relatively unexplored variable, and so it was decided that the applied techniques will draw inspiration from other works, mainly ones that forecast Sales variables, due to the similarity to our target variable and to the abundance of relevant literature on this type of forecasting.

As the data only provides two types of continuous variables, the target and date variables, it is possible to hypothesise that the target variable exhibits temporal dependence. And so, since the objective is to make a forecasting tool, the project will be conducted by comparing several time series forecasting techniques, starting with a traditional statistic model such as the Auto Regressive Integrated Moving Average (ARIMA), moving on to more advanced models like Prophet and furthermore some commonly used Deep Learning techniques such as the Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN).

This study aims to investigate the effectiveness of deep learning models in time series forecasting of net shipped units, comparing them to less complex approaches like ARIMA and Prophet. Knowing this, two research questions can be drawn: Is a deep learning model more effective than less complex approaches, such as ARIMA and Prophet, in time series forecasting for net shipped units? And secondly: Which model performs best among the considered approaches for time series forecasting of net shipped units? By addressing these questions, this work aims to accomplish the following objectives:

- Evaluate the performance of deep learning models in comparison to less complex approaches
- Identify the most optimal model among the compared approaches, providing valuable insights for the company when deciding to adopt a time series forecasting approach
- Draw a conclusion regarding the best-performing model, provided that the results are robust an reliable
- Address the primary and secondary research questions based on the findings of this work, providing recommendations for the company regarding the adoption of a time series forecasting methodology for net shipped units

The following structure highlights the structure of this work after the current section :

Literature Review (section 2) offers a critical review of earlier research in the area of sales forecasting techniques, covering both traditional statistic methods and modern machine learning methodologies,

comparing them in a wide range of scenarios. Methodology (Section 3), defines the methodology used to analyse and draw conclusions from the data collected. Furthermore, Results and Discussion (section 4) summarizes the results obtained along with an in depth discussion of the findings, highlighting the implications and limitations of the results, and Conclusion (section 5) discusses conclusions and limitations of the conducted analysis, along with recommendations and future work, offering suggestions on how to improve the study and investigate new opportunities.

2. LITERATURE REVIEW

2.1. TIME SERIES FORECASTING

A time series is a set of observations generated sequentially over time. Examples of time series can happen in various fields, from economics to engineering, and its analysis plays a key role in statistics (Chatfield, 2001). These observations can be recorded continuously or at specific discrete time intervals. Essentially, the latter can be classified as continuous or discrete. Within the scope of this report, and following the common trend of time series problems, this time series can be considered discrete, therefore, no further investigation will be performed on continuous problems (Chatfield, 2001). Discrete time series can appear in three different scenarios (Chatfield, 2003):

- **Sampled** from a continuous time series, where the sampling interval between recordings has to be carefully chosen due to the risk of information lost.
- **Aggregated** over a period of time, examples can be monthly exports or daily rainfalls.
- **Inherently discrete,** with an example being the dividend paid by a company to shareholders in successive years.

The particularity of time series analysis is that, usually, successive observations are dependent. This correlation between them is studied with the use of time series analysis, which provides several techniques for doing so (Box et al., 2008). To better describe these relationships, it is necessary to develop stochastic and dynamic models and use them in different areas of application, namely the forecasting of future values which will explored in this work (Box et al., 2008). These models are especially effective when there is limited information on how the data was generated, or no apparent alternative modelling solution that correctly associates the target variable to other explanatory variables (Zhang, 2003) . Models are widely applied when referring to a time series problem, being the ARIMA model one of the most important ones. Firstly introduced in the 70s (Box & Jenkins, 1970), its popularity is mostly due to its statistical properties and model-building methodology (Zhang, 2003). More recently, a powerful time series forecasting model was introduced by the Facebook company (Taylor & Letham, 2018), a decomposable time series model based on an additive regression approach and capable of modeling trends, seasonal patterns, and even outliers related to weekends or holidays (Ensafi et al., 2022; Papacharalampous et al., 2018). Other powerful tools for accurate forecasting are deep learning and machine learning techniques, more particularly LSTMs (Ensafi et al., 2022 ; Santos et al., 2022), and CNNs (Wibawa et al., 2022).

2.2. TIME SERIES FORECASTING MODELS

This section will provide an explanation of the above-mentioned time series models, which will subsequently be applied in this study.

2.2.1. ARMA and ARIMA

Auto Regressive Moving Average (ARMA), consists of an integration of the Moving average (MA) and autoregressive (AR) models and can be represented as ARMA (p, q).

A derivation from this model, the Autoregressive Integrated Moving Average (ARIMA) model is one of the most popular approaches to forecasting. It is usually applied on non-stationary time series due to its capability of making such a time series stationary and because it follows the (Box & Jenkins, 1970) methodology.

In this model, the future value of a variable is conceived as a linear combination of historical values and preceding errors, depicted by the following expression (Pai & Lin, 2005):

$$y_t = \theta_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \quad (1)$$

where y_t is the value and ε_t the random error at a time t. ϕ_i and θ_j are the coefficients and p and q are integers that are often referred to as autoregressive (AR) and moving average (MA) polynomials, respectively. The I element of the model or the d in ARIMA (p, d, q) represents the order of differencing, which stands for the amount of logarithmic reductions that have to be made to make the time series stationary (Khandelwal et al., 2015). Although a popular and effective model, ARIMA can present limitations when handling larger forecasts (Ji et al., 2016) or nonlinear patterns (Pai & Lin, 2005; Khandelwal et al., 2015) due to its linearity restriction.

2.2.2. Prophet

One of the most accomplished companies in developing time series forecasting models is Facebook (Ensafi et al., 2022). Lately, the company released a new model called Prophet, a model implemented as open-source software available in Python and R programming languages. (Taylor & Letham, 2018) have proposed a modular regression model with interpretable parameters, which can be managed with ease by analysts with time series knowledge. It is a decomposable time series model based on additive regression and it is capable of modeling trends, seasonal patterns, and even outliers related to weekends or holidays (Santos et al., 2022). The model can be represented with the following equation:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad (2)$$

Where y(t) is the additive regression model, and g(t), s(t), and h(t) represent the trend, seasonality, and holidays components respectively. The error ε_t stands for any factors that are not taken into account by the model.

2.2.3. Deep Learning Models Overview

2.2.3.1 LSTM

A Long Short-Term Memory (LSTM) is a deep learning model and a type of Recurrent Neural Network (RNN) that can solve the "short-term memory" problem by using a mechanism of gates that regulate

the flow of information (Santos et al., 2022). Before explaining LSTMs, it is important to understand briefly two important concepts:

1. **Artificial Neural Network (ANN) –** A neural network is composed of at least three layers, an input layer, hidden layers, and output layers. The input layer is composed by one or more nodes, which number is decided by the number of observations in an input dataset (Siami-Namini et al., 2018).
   As its architecture aims to mimic human brain activity, these nodes are called "neurons" and the links between them are known as "synapses". Weights are another important concept within ANNs and play a key role in them by deciding which signal or input passes through. In the hidden layers, the outcome of the input layer is processed and then passed on to the output layer which returns the actual prediction. In this type of model, the output result of a layer is always the input of the succeeding layer. A disadvantage of ANNs is that it does not have a memory to store information from the past, and so it won't perform as well when dealing with nonlinear problems.

2. **Recurrent Neural Network (RNN)** - An alternative to ANNs that store past information is RNNs. In this approach, this information is kept in the hidden layers level where a recurrent loop to the back exists. This means that the output is a function of the previous input, concatenated on the activation value of the past hidden layers (Ensafi et al., 2022). A representation of an RNN is displayed in Figure 2.1.

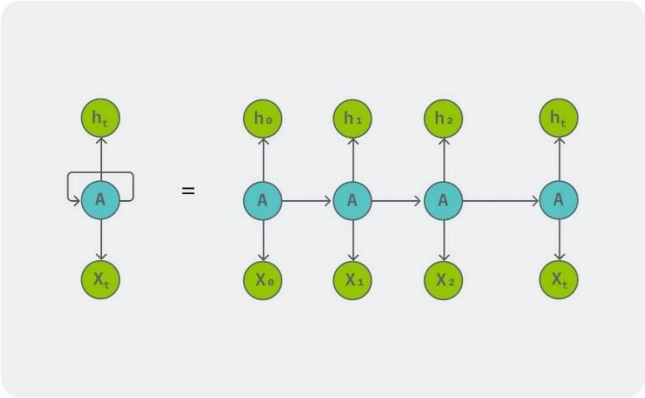

Figure 2.1 – RNN (Baheti, 2022).

A traditional RNN however, has some disadvantages in the way it "records" past steps of the network, especially when trying to remember longer sequences of data as these networks remember only a few earlier steps in the sequence. In other words, this can be called a vanishing gradient problem.

Having established the fundamentals, an in-depth explanation of the LSTM model is presented below. Firstly introduced by (Hochreiter & Urgen Schmidhuber, 1997), this type of RNN architecture aims to solve the aforementioned vanishing gradient problem by using a special type of structure called memory cells and gate units.

A memory cell is represented by (Yu et al., 2018) :

- Input gate - protects the memory state stored in each memory cell from perturbation by unrelated inputs. Has a sigmoid function and ***tanh*** function, with the purpose of adding new information.
- Output gate - protects other units from perturbations by irrelevant memory contents, stored in the memory cells. Is another sigmoid function that decides the amount of information that will be included in h.
- Forget gate - allows memory to forget irrelevant memory cell content. In practice, it is represented by a sigmoid function which returns a value between 0 and 1 that determines whether or not information passes through.
- Activation function - an element-wise application which supports the calculation of the internal memory cell states (Cortez et al., 2018).

A traditional LSTM architecture consists of three gate activation functions (sigmoid functions), and two output activation functions (tanh functions) (Kumar et al., 2018). Figure 2.2 depicts a traditional LSTM architecture where X_t represents input X at time-step t. H_t is the output for one time step and C_t represents the cell state.

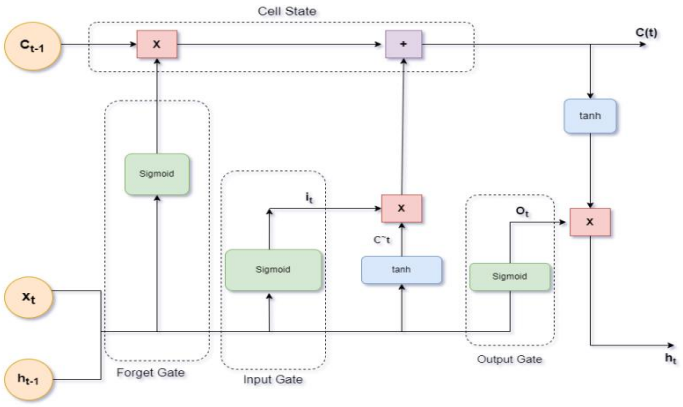

Figure 2.2 - LSTM cell architecture (Thomas, 2023).

By stacking these memory cells, LSTMs can be used for time series forecasting, an example of the relationship between a time series and an LSTM is shown in Figure 2.3.

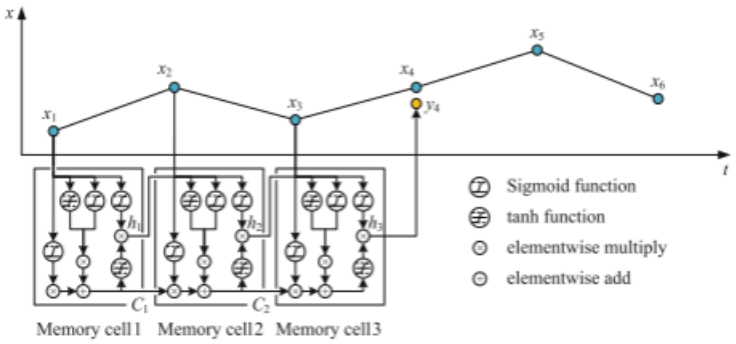

Figure 2.3 - LSTM network for time series forecasting (Yu et al., 2018).

2.2.3.2 CNN

Traditionally used for image processing and text recognition, a Convolutional Neural Network (CNN) has been also recently applied to time series forecasting problems(Wibawa et al., 2022) mainly due to its ability to recognize patterns. These convolutional layers interchange with max-pooling layers, and by doing this mimic complex and simple cells of a mammalian visual cortex. In a time series approach, however, results from previous CNNs are fed into the next layer and the max pooling layer acts as a way to prevent overfitting. (Ensafi et al., 2022) . In these cases, and also the approach chosen in this study, the most common type of CNN is a 1D CNN model, represented in Figure 2.4.

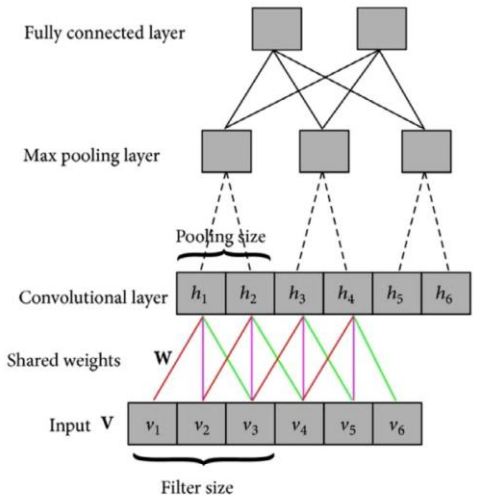

Figure 2.4 - 1D CNN architecture (Yu et al., 2018).

2.3. RELATED WORK

This section is dedicated to previous research works that have been done in the field of time-series forecasting, more specifically studies that tackle comparisons between the previously mentioned models, and preferably but not exclusively in the field of Sales forecasting since it closely resembles this study's problem.

The authors in the study (Ramos et al., 2015) compare state space models with ARIMA models. The study applies these methods to women's footwear products, namely Boots, Booties, Flats, Sandals, and shoes with monthly data ranging from 2007 to 2012 with a total of 64 observations. To evaluate the models the Akaike Information Criteria (AIC) was chosen to select the best model. Both single-step and multi-step forecasts were produced. It was shown that when using Mean Error (MA) Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) as performance metrics, the obtained values are quite similar for both types of forecasts. The conclusion was that state space and ARIMA produce similar coverage probabilities to the nominal rates of one-step and multi-step forecasting.

The researchers on (Papacharalampous et al., 2018), aimed to predict temperature and precipitation by using automatic univariate time series forecasting methods. The monthly data was composed of 985 data points during 40 years for temperature and 1552 data points for precipitation in the same time frame. Models tested were ARMA, Trend and Seasonal Components (BATS), AutoRegressive Fractionally Integrated Moving Average (ARFIMA), Theta, and Prophet. These were tested in multiple-step-ahead forecasts for the last 48 months of data. The authors concluded that for this type of data, all the methods were suitable for long-term applications except for the naïve and random walk ones.(Papacharalampous et al., 2018) It was also found that this type of data can barely be improved when using other methods and that the classic seasonal decomposition models perform better than the automatic ones like BATS and Prophet. Finally, it was concluded that Prophet is a competitive model, especially when combined with external classic seasonal decomposition.

In (Santos et al., 2022), the researchers test a variant of the traditional LSTM network called Seq2Seq LSTM. With a dataset containing daily footwear sales for a 3-year period, seven-time series corresponding to seven distinct types of products are analyzed. In this work, this model is compared with the popular variant of the ARIMA model, Seasonal ARIMA (SARIMA), and the Prophet model. With a prediction horizon of 7 days, the evaluation is made assuming a rolling window scheme with 28 training and test iterations, with the main metric of model performance being Normalized Mean Absolute Error (NMAE). The most competitive results were achieved through the proposed LSTM approach with an NMAE ranging from 5% to 11%, with the Prophet and SARIMA models showing equivalent results. Researchers aim to expand the proposed model to be applied to monthly data, multivariate time series, and admit that it would be interesting to apply a CNN approach(Santos et al., 2022) .

In the article (Ensafi et al., 2022), authors compare different time series techniques to predict seasonal sales items with a focus on testing deep learning approaches. Given a dataset with data from 2014-2017, researchers pre-processed their data and performed an extensive Exploratory Data Analysis to find seasonality in three different product categories, Furniture, Tech, and Office Supplies. After only selecting the first and only seasonal category, Furniture, the authors decided to resample their data to a monthly frequency due to the high fluctuation of daily sales(Ensafi et al., 2022) .The next step was applying the common train test split method to the data. Since the aim was to predict the last year of sales data, with 4 years' worth of records, the decision was to do a 75% training and 25% test split. Consequently, 10 different models were tested, including ARMA, ARIMA, and SARIMA, two variants of exponential smoothing, two different Prophet configurations, four variants of LSTM, and finally a CNN network. To evaluate results, the metrics chosen were Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). Results showed that the best-performing model was a stacked LSTM with two layers, with CNN and Prophet models also showing good results.

In (Wibawa et al., 2022), the main focus is to propose a CNN method and compare it with other deep learning approaches such as Multilayer Perception (MLP) and LSTM, using MSE, MAPE, and Training Time as performance metrics. To be precise, the model proposed was a hybrid model between exponential smoothing and CNN which is designated as S-CNN.Click or tap here to enter text. For this study, 4 datasets or 4 different time series with different data split techniques were analysed. The methodology used for the experiment of comparing the S-CNN was composed of an initial phase of data normalization and exponential smoothing with an optimal α, combined with a CNN model with Lucas Hidden Layers. The results were then compared with the above-mentioned methods and evaluated with the above-mentioned metrics. Final results showed that the S-CNN model performed better than the others at an 80%:20% data composition, using 76 hidden layers and with the best MSE of 0.012147693.

2.4. CONCLUSION

By analyzing the research done for this study, it can be inferred that, whenever applied as a method, with realistic parameter values, LSTM networks usually outperform other methods. Prophet is highly popular in comparative studies and usually outperforms more traditional forecasting techniques such as ARIMA or Exponential Smoothing. CNNs although recent, are increasing in popularity and some cases even outperformed LSTMs. Moreover, ARIMA stood out as a popular forecasting method and a strong baseline in a significant amount of the reviewed literature. The research done builds a solid foundation for the application of the above-described models to the problem under analysis in this project.

3. METHODOLOGY

This section will outline the methodology used in this project work. An in-depth description of the methodology followed will be presented. This includes data collection, Exploratory Data Analysis (EDA), Data preparation, modeling approach, implementation and evaluation techniques.

3.1. APPLIED METHODOLOGY

After an initial look at the collected data, and looking at the problem in hands, it was inferred that between the two most common retail forecasting approaches, regression and time series(Chu & Zhang, 2003), the latter would be adopted since our target variable is the only numerical feature available in the raw data. Having this into account, and following the scope and objective of this project, the methodology applied will be performed assuming an univariate time-series problem, where the data in analysis is converted from the raw data into a one-column (target variable) dataset with a datetime index corresponding to the period between 01 January 2021 and 31 December 2022. An overview of the methodology is shown in Figure 3.1.

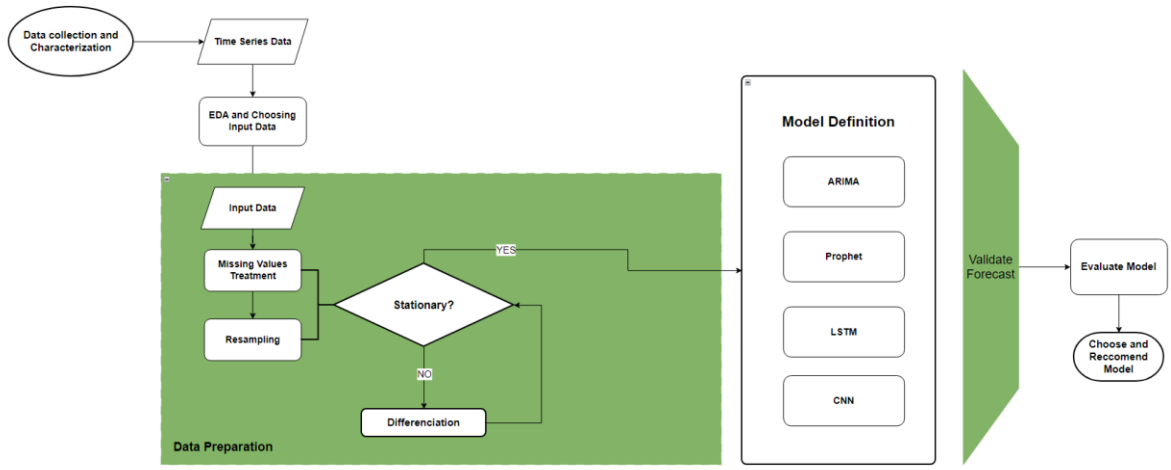

Figure 3.1 – Methodology followed

Figure 3.1. shows that the methodological approach starts with data collection and characterization. After collecting and characterizing the data, the output of this process is the view of the product types as time series, analysed in detail in the EDA and Input Data section. Moreover, the missing values and resampling subsection deals with the missing value imputation technique used and whether or not resampling is required for the time series data. In the stationarity check subsection, the two main techniques used for stationarity evaluation are presented, and a decision will be made on whether or not to differentiate the time series. After checking for stationarity, the model definition step of the methodology defines which models are under study how to find the best parameters for their application. In the evaluation metrics subsection, an overview of the chosen performance metrics is explained, and finally a recommendation of the best model can be chosen,

explained in detail in the Results section. Additionally, a Tools Used subsection is present to list all the technologies used in the process.

3.1.1. Data Collection and Characterization

To choose between the most appropriate data to collect, one has to take into account which variables had the most influence on the Net Shipped Units planning process within the company. After discussing with specialists within the supply chain and operations department, there were some significant limitations on the confidentiality of the data in question, indicators like price or SKUs had to be omitted, as well as some other important quantitative variables. As a result, a dataset containing 2.420.283 rows was extracted, with each row corresponding to an order of units made for a specific product from a retailer between 2021 and 2022. The data is composed of 12 variables, 9 product-related categorical variables, 2 date variables, the requested delivery date and the actual delivery date of the order, and finally the target variable per order, Net Shipped Units. The extracted metadata description can be found in more detail in Table 3.1. It is important to note that the original column names were renamed to maintain data confidentiality.

Table 3.1 – Raw data description

Column Name	**Description**
GC	Gender the product was made for
SE	The segment of the product, a famous retail company example could be within any big retail company, there is the Running segment and the Football segment
DI	The main division of the product, in our case products can be Footwear, Apparel, or Accessories
C	Product category, a more detailed categorization within the product segment, similar to SE but more specified
SI	The silhouette of the product, in this particular company there is an important differentiation to make at this level, especially in the Footwear division
M_D	A more detailed categorization than in SE and C
F_N	Lowest level of product categorization
S_Y	Differentiation the season where the product was released, there are four main seasons in the fashion industry, Summer, Holiday, Fall and Spring
N_S_U(thousands)	The target prediction variable, the number of units shipped per order, expressed in thousands
GD	The actual order delivery date
DDATE	The requested order delivery date of the order
TER	The location of the order defined by a specific company segmentation of European retailers.

3.1.2. Exploratory Data Analysis (EDA) and Choosing Input Data

As outlined earlier, this section is dedicated to visualizing and analyzing the time series data. By segmenting the product data into its three primary clothing categories - Footwear, Apparel, and Accessories - we can refine our predictions and develop a more nuanced understanding of the time series. Before plotting the three-time series, some outlier data corresponding to periods outside the two-year scope was spotted, so small processing was done to remove data that didn't belong in the years 2021 and 2022, giving the final time series plots shown in Figures 3.2, 3.3, and 3.4. Through visual analysis of the aforementioned plots, which display net shipped units against time, it can be assumed that both Footwear (Figure 3.2) and Apparel (Figure 3.3) time series exhibit seasonality, indicating a yearly pattern. When looking at the Accessories time series (Figure 3.4) there is a very reduced number of recorded data. It only has 161 data points which is considered a very low sample especially taking into account that these full two years of data should have registered 730 observations. For this reason, the third time series is discarded, and only Footwear and Apparel products will be analysed in this work.

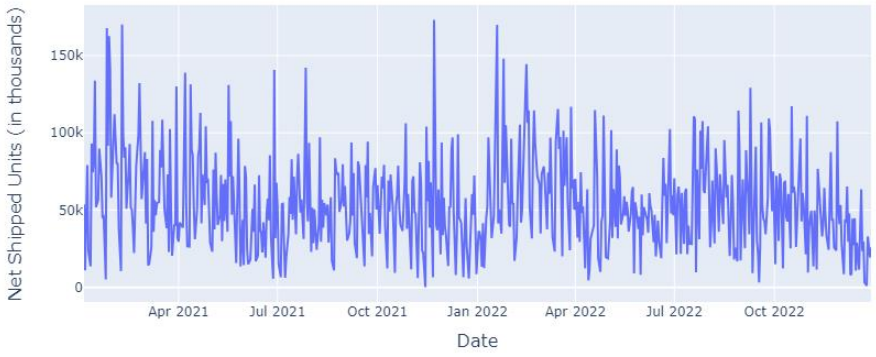

Figure 3.2 – Footwear products net shipped units from 2021 to 2022

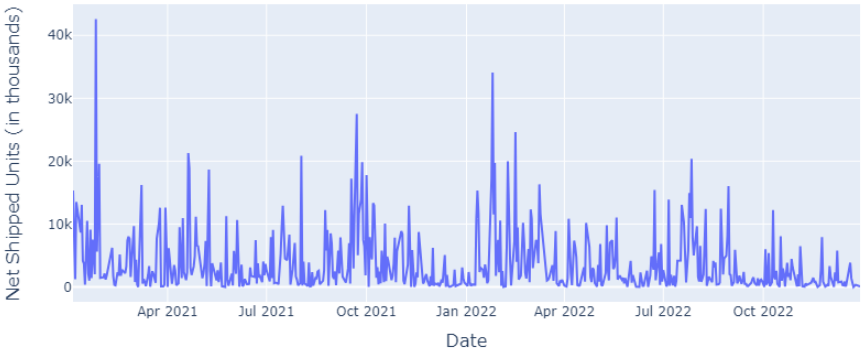

Figure 3.3 - Apparel net shipped units from 2021 to 2022

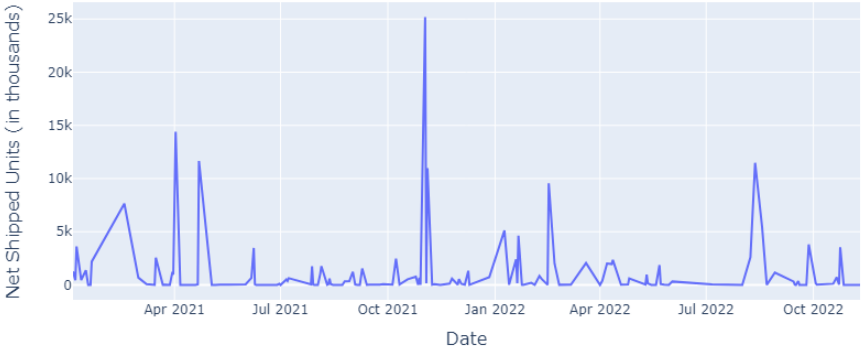

Figure 3.4 - Accessories net shipped units from 2021 to 2022

As for the amount of data for Footwear (FT) and Apparel (AP), there were 609 and 563 datapoints respectively, which translates into 121 missing datapoints for FT and 167 for AP. In other to further check seasonality and prepare our data to feed the chosen models, a small analysis of the missing values and how they were imputed was conducted in the Data Preparation section.

3.1.3. Data Preparation

3.1.3.1. Missing Values Treatment and Resampling

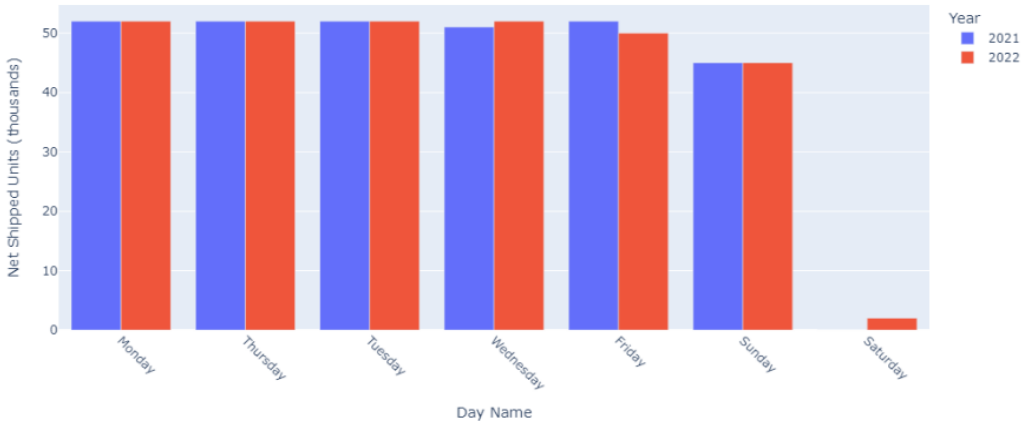

Figure 3.5 - Footwear observation count per weekday

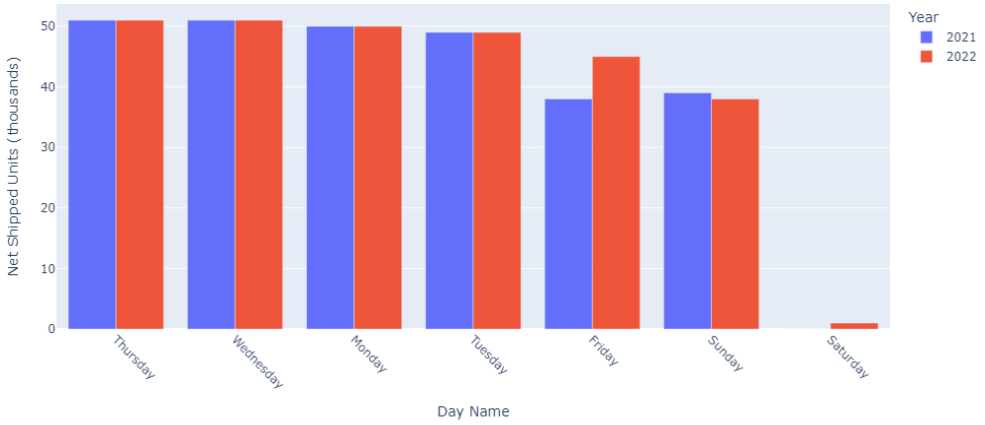

Figure 3.6 – Apparel observation count per weekday

Dealing with missing values in a time series problem is always a crucial task for better and more accurate model performance. There are several methods used for imputing missing data, most commonly constant replacing, either with a zero replacement or mean replacement, or forward filling, which takes a value and fills the next seen values with it or backward filling, which applies the opposite logic.(Ahn et al., 2021). The success of these methods relies on the characteristics of the data being examined. Since the data is collected daily, it can be safely assumed that missing values represent days with no order deliveries.

Due to this reason, the decision was to replace all missing values with a constant replacing technique, in this case the value 0. Moreover, an analysis of the missing observations was made in respect to which day of the week they occurred in, and the results for AP and FT can be seen in Figures 3.5 and 3.6.

Looking at the results, it is easily verifiable that most missing values fall within the weekend categorization. More specifically, for Footwear products, 116 corresponded to weekends, and in Apparel, 130 were weekends and 21 were Fridays. It is important to understand that in the context of our problem, the data obtained depends on whether or not an order was placed, making the constant replacement method mentioned above the more adequate in this situation, which rules out resampling of any kind. The final format our data assumed for the remainder of the analysis is, for both time series, 730 datapoints, with a constant replacement method for imputing missing values. For the Footwear time series, values range from 0 to around 170.000.000 shipped units and for Apparel from 0 to around 40.000.000 units. Our data was then ready to perform a stationarity check, the last phase of our data preparation process.

3.1.3.2. Stationarity Check

One of the most important characteristics of a time series is its stationarity. A time series is called stationary if its statistical properties remain constant over time. (Pinho et al., 2019). A stationary time series is important because if a property shows similar behaviour over time, it means it will replicate that in the future with more certainty. Moreover, this will make the time series more model friendly as compared to non-stationary ones. (Aarshay Jain, 2023).

To determine whether or not a time series is stationary there are two main methods: plot the rolling statistics, which are the statistical properties over time, and the Augmented Dicky-Fueller (ADF) statistical test. ADF measures the presence of a unit root in the time series. The level of significance for all statistical tests was set at a p-value <0.05, with a 95% confidence interval for each test. This indicates the range of values within which the true population value is likely to fall, meaning that the series is stationary. If its greater than the aforementioned value, the null hypothesis is not rejected, and the series is not stationary. In this case, a differentiation of the data has to be done in order to make it stationary. When performing ADF, the p-values were 0.016141 and 0.001127 for the FT and AP time series respectively. Both p-values fell below 0.05 meaning the time series are stationary and no further transformations have to be performed to the data. A plot of the rolling statistics is displayed in Figures 3.7 and 3.8.

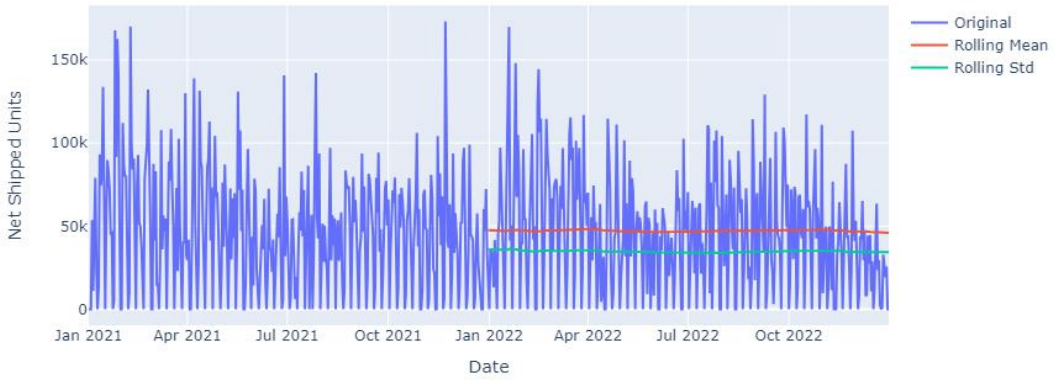

Figure 3.7 – Rolling statistics for Footwear products

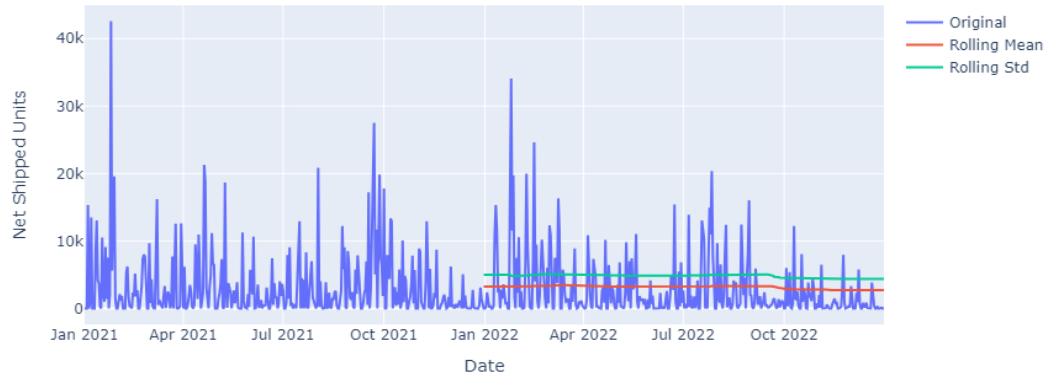

Figure 3.8 – Rolling statistics for Apparel products

Figures 3.7 and 3.8 show that the rolling statistics for both Apparel and Footwear products remain relatively constant over time, another indicator that our time series under analysis are stationary.

3.1.4. Model Definition

3.1.4.1. Data Split

A crucial step when performing time series forecasting, just like when adopting machine learning approaches is to split the data into train and test set. Since an observation in a time series is dependent

on the precedent observations, advanced techniques such as K-fold cross validation are not useful for TSF (Ensafi et al., 2022). Taking this into account a common train test split was the chosen criteria.

With a length of 730 observations, our time series can be classified as having a reduced number of datapoints, which served as a decisive factor to split the data into a bigger training set than usual. Following the apparent pattern shapes, and in an attempt to capture as much data as possible, it was decided that the training set would correspond to 90% of the data, and the test set to the remaining 10%. Unlike most machine learning models, time series require the order of the split to be preserved, and so the technique used was simply to allocate the first 80% observations to train set the next 10% to validation, and the remaining 10% to test. This split was used in the same shape consistently in all the model implementation. Data was further scaled due to the high fluctuations in the observations.

The following sections will provide a short but concise review on the chosen parameters for each model. For this work, shipped units won't be forecasted into the future but tested on how well the past observations in the obtained data predict future values.

3.1.4.2. ARIMA

In the process of implementing ARIMA (p, d, q), the optimal values for these three parameters have to be found in order to ensure that the model has the most accurate prediction possible. There are essentially two paths to find the optimal values. Firstly, one has to start by looking at the Auto Correlation Function (ACF) and Partial Auto Correlation Function (PACF) plots for each time series. To find the lag order (AR or *p* parameter), one can look at the ACF plot. For the moving average (MA or *q* parameter), the PACF plot. The d parameter, however, represents the level of differencing. Since our series is stationary, no differencing was done, so the assumption can be made that 0 will be the best value, and possibly 1. This method, however, may not present the best results. Thus, a hyperparameter tunning algorithm was applied in order to experiment different combinations of models in the training set, and later choose the best model to predict unseen data in the test set. Tables 2.1. and 2.2. show the range of values for each parameter fed onto the parameter tunning algorithm and the chosen value based on the Root Mean Squared Error (RMSE) metric for Footwear and Apparel Respectively. For ARIMA models it is also common to use the Akaike Information Criterion (AIC) as a grid search model evaluation metric, however, due to the objective of having the best prediction possible of unseen values in this work, RMSE was chosen.

Table 3.2 – Footwear ARIMA hyperparameters

	Chosen parameter	**Range of values**
p	5	[1,2,4,5,8]
d	1	[0,1,2]
q	2	[0,1,2]

Table 3.3 – Apparel ARIMA hyperparameters

	Chosen parameter	**Range of values**
p	8	[1,2,4,5,8]
d	0	[0,1,2]
q	2	[0,1,2]

Table 3.2 and 3.3 show that for both products, despite applying a grid search, the *q* values selected were significantly close to the ones inferred from the autocorrelation plots. On the other hand, there was a difference on the outcome with a higher *p* parameter value than expected. Parameter tunning also confirmed a *d* value of 0 and 1, as expected.

3.1.4.3. Prophet

When using a Prophet model, the simplicity of its implementation it's one of the major advantages of using it. All it was done was a grid search with the most commonly tested parameters, in addition to the *yearly_seasonality* parameter which was included due to the apparent yearly pattern of the data. This allows us to change the Fourier order by passing a range of values on to the aforementioned parameter, which can greatly improve the model. Moreover, holiday data was added correspondent to the countries where the company that provided the data distributes its products to. Similarly to the ARIMA hyperparameter tunning, the RMSE metric was the chosen one to determine the best model to apply. Tables 3.1 and 3.2 depict the parameters tested and chosen for this model.

Table 3.4– Hyperparameters for Prophet (Footwear)

	Chosen parameter	**Range of values**
changepoint_prior_scale	0.5	[0.01,0.1,0.5]
seasonality_prior_scale	1.0	[0.01,0.1,1.0,10.0]
holidays_prior_scale	1.0	[0.01,0.1,1.0,10.0]
seasonality_mode	Additive	["additive", "multiplicative"]
yearly_seasonality	15	[15, 20,25,30]

Table 3.5 – Hyperparameters or Prophet (Apparel)

	Chosen parameter	**Range of values**
changepoint_prior_scale	0.1	[0.01,0.1,0.5]
seasonality_prior_scale	0.01	[0.01,0.1,1.0,10.0]
holidays_prior_scale	0.1	[0.01,0.1,1.0,10.0]
seasonality_mode	Multiplicative	["additive", "multiplicative"]
yearly_seasonality	20	[15, 20,25,30]

3.1.4.4. Deep Learning Algorithms

When it comes to implementing a Deep Learning model, the complexity and number of parameters to take into account is significantly higher than in the previously mentioned models. Hyperparameter tuning becomes an essential task and ensuring a satisfactory performance relies heavily on how well our parameters are chosen. Table 3.6 shows the parameter grid for all the models used in this work.

Table 3.6 – Hyperparameter tunning for deep learning models

LSTM	**Range of values**
LSTM layer n_inputs	[10, 20, 30, 40, 50, 60, 80, 90, 100]
Dense layer n_inputs	[8,16,32]
Dense_layer activation	[*relu, elu, tanh*]
Dropout rate	[0.2, 0.3, 0.4, 0.5]
Learning_rate	[0.01, 0.001,0.0001]
Stacked LSTM	**Range of values**
1st LSTM layer n_inputs	[10, 20, 30, 40, 50, 60, 80, 90, 100]
2nd LSTM layer n_inputs	[10, 20, 30, 40, 50]
Dense layer n_inputs	[8,16,32]
Dense_layer activation	[*relu, elu, tanh*]
Dropout rate	[0.2, 0.3, 0.4, 0.5]
Learning_rate	[0.01, 0.001,0.0001]
CNN	**Range of values**
CNN layer n_inputs	[16, 32, 64, 128, 256]
Dense layer n_inputs	[8,16,32]
Dense_layer activation	[*relu, elu, tanh*]
Dropout rate	[0.2, 0.3, 0.4, 0.5]
Learning_rate	[0.01, 0.001,0.0001]

As this project only aims to implement a simple forecasting alternative, both vanilla versions of LSTM and CNN where applied, meaning each only had one hidden layer. Moreover, a stacked version of LSTM with two hidden layers was tested in order to test the effect of an increase in model complexity. Moreover, a dropout layer was introduced in the LSTM models in order to prevent overfitting. Finally, a second Dense layer was added to all models to increase complexity.

When compiling the model, the Adam optimizer was chosen with the loss being evaluated with the mean squared error, and RMSE as a model metric.

A crucial parameter when tunning our models is time steps. These represent the dimensionality of our input layer, or how many values will be in an input sequence to help predict the next value. In order to experiment with different time steps, the Footwear time series was tested with 8, 10, 20 and 30 steps, in this case days, and Apparel 20, 30, 40 and 50 days. For the parameters fine-tunning process, the *Keras-tuner* package was used. The next section will briefly present and justify the choice of the model evaluation metrics used in this exercise.

3.1.5. Evaluation Metrics

To evaluate the model's performance, the two most common metrics were used. These are the Root Mean Squared Error (RMSE), which measures the standard deviation of the prediction errors, and Mean Absolute Percentage Error (MAPE), which measures the percentage of absolute difference between the actual and predicted values. Both represented by equations (3) and (4) respectively. Where Y_t is the actual value of the observations and F_t the forecasted value. The value of n represents the total number of observations.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(Y_t - F_t)^2} \quad (3)$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{(Y_t - F_t)}{Y_t}\right| * 100 \quad (4)$$

For a more in context explanation, the RMSE will compare our predictions of unseen data with the actual values, and return a numerical value that represents, on average, the deviation of our prediction when compared to the real values. The MAPE metric allows us to evaluate the how off our predictions are in terms of percentage.

RMSE will also have the leading role when comparing model performance since it is the metric that is most sensitive to significant differences between values. In a context of where a company has to accurately predict its shipping volume, it is important to stay truthful to the reality of the observations. MAPE will serve mostly as a second metric which allows us to look at the performance in a percentage format.

These metrics were the ones chosen since they are used in a variety of related works that tackle similar time series problems. Examples are (Ramos et al., 2015; Wibawa et al., 2022; Chu & Zhang, 2003; Ensafi et al., 2022).

3.2. TOOLS USED

This subsection will briefly describe the crucial tools used in this project to conduct a time series forecasting project adequately. The chosen programming language for initial data collection was Structured Query Language (SQL), integrated within the Snowflake Data Warehousing platform used in the company. This was due to the way the internal data structure of the company was designed, making them the only possible tools to use for this purpose.

To write the project itself, the chosen language was Python programming language, due to its flexibility, the author's previous familiarity with the language, and the wide range of libraries that allow the user to perform all the needed tasks required in a work of this nature. A list of the chosen libraries is displayed below:

- NumPy - Fundamental package for scientific computing in Python[2]. Used to perform some operations in arrays, mainly when dealing with missing values or rearranging array shapes for the deep learning models applied.
- Pandas - Aims to be the fundamental building block for doing practical, real-world data analysis and manipulation in Python[3]. One of the most popular packages used in data science, it serves as the main library of this project, its data structures, pandas Series, and Data Frames are used in this project as the main data analysis, manipulation, and visualization tool.
- Matplotlib - comprehensive library for creating static, animated, and interactive visualizations[4]. Used as the main visualization tool in the project.
- Statsmodels – provides a complement to SciPy for statistical computations[5]. Used for every statistic-related operation done. More specifically to test stationarity and build ARIMA models.
- Scikit-Learn - machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection, model evaluation, and many other utilities[6]. Used in our particular case to scale data and to compute evaluation metrics.
- TensorFlow – end-to-end open-source machine learning and deep learning framework[7]. Started as a backend for Keras, which is the high-level API of the TensorFlow platform, and it is what allows us to apply deep learning models such as LSTM and CNN

[2] https://numpy.org/doc/1.26/
[3] https://pandas.pydata.org/pandas-docs/stable/
[4] https://matplotlib.org/stable/
[5] https://pypi.org/project/statsmodels/
[6] https://scikit-learn.org/stable/
[7] https://www.tensorflow.org/api_docs/python/tf/keras/

4. RESULTS AND DISCUSSION

In this section, a detailed analysis of the results obtained from the test set will help us to make a comparative analysis regarding the performance of each model. This analysis aims to deliver concrete insights on how time series forecasting techniques predict shipping volume forecasting in the context of the company under study. The results for each model will be presented, and interpreted, and finally, the significance of the results and any problems with the analysis will also be addressed.

4.1. MODEL RESULTS

The primary goal of this work is to evaluate different time series forecasting techniques, by comparing traditional statistical methods such as ARIMA, with more recent approaches like Prophet, combining with further applications of deep learning models. This allows us to recommend a modeling approach based on the results obtained to the company that provided the data necessary for this work's development.

As mentioned before, to assess model performance, RMSE and MAPE will be our key metrics. These provide different angles on the accuracy and robustness of the models, already explained in section 3.4. Tables 4.1 and 4.2 display the results obtained in terms of error, for Footwear and Apparel products respectively.

Table 4.1 – Model results for Footwear products

Model	RMSE	MAPE
ARIMA	32512.26	54.15 %
Prophet	23550.55	65.34 %
LSTM	20115.42	48.44 %
Stacked LSTM	18663.29	48.77 %
CNN	18060.33	46.92 %

Table 4.2- Model results for Apparel products

Model	RMSE	MAPE
ARIMA	2868.26	86.54 %
Prophet	2116.17	83.13 %
LSTM	1693.29	85.54 %
Stacked LSTM	2028.75	80.71 %
CNN	2188.16	79.86 %

At first glance, it can be determined that advanced forecasting techniques do outperform traditional statistical models, even though the results are not quantitively the best, this trend is present in both time series, with more emphasis on the Apparel one. For the context of the analysis, it is important to reference that values in the footwear time series range from 0 to 173000 thousand units, and its average value is around 46700 thousand units, whilst for apparel the range goes from 0 to around 42000 thousand units and its average value is 2970 units.

To interpret the results, it facilitates to address each type of products separately:

Footwear products:

- The Prophet model, although not expected to be the worst performer in this case, since it manages seasonality well and adds the holiday dates into its equation, shows the worse MAPE value with 65.34 %, and the second lowest RMSE only behind prophet, which suggests difficulty in handling high fluctuations in data.
- A relatively satisfactory performance in comparison to other models was obtained with the stacked LSTM model, which presents an RMSE of 18663.29 and a MAPE of 48.77%
- The best model for the footwear time series, however, was the CNN model, with a MAPE value of 46.92 % and the lowest RMSE of 18060.33.

Apparel products:

- For apparel products, the CNN model shows the lowest MAPE of 79.86 %, being the model that best handles percentage errors.
- The vanilla LSTM model stands out with a much lower RMSE of 1693.29, which indicates good absolute error handling. When compared to others, however, has the second highest MAPE of 85.54 %.
- The ARIMA model shows the worst performance of all as expected with the highest MAPE and RMSE values.

After presenting the results, it is interesting to visually see how well our models predicted the number of units shipped. Figures 4.1 and 4.2 illustrate the above results for the last 40 days of our test set for Footwear products and Apparel products respectively

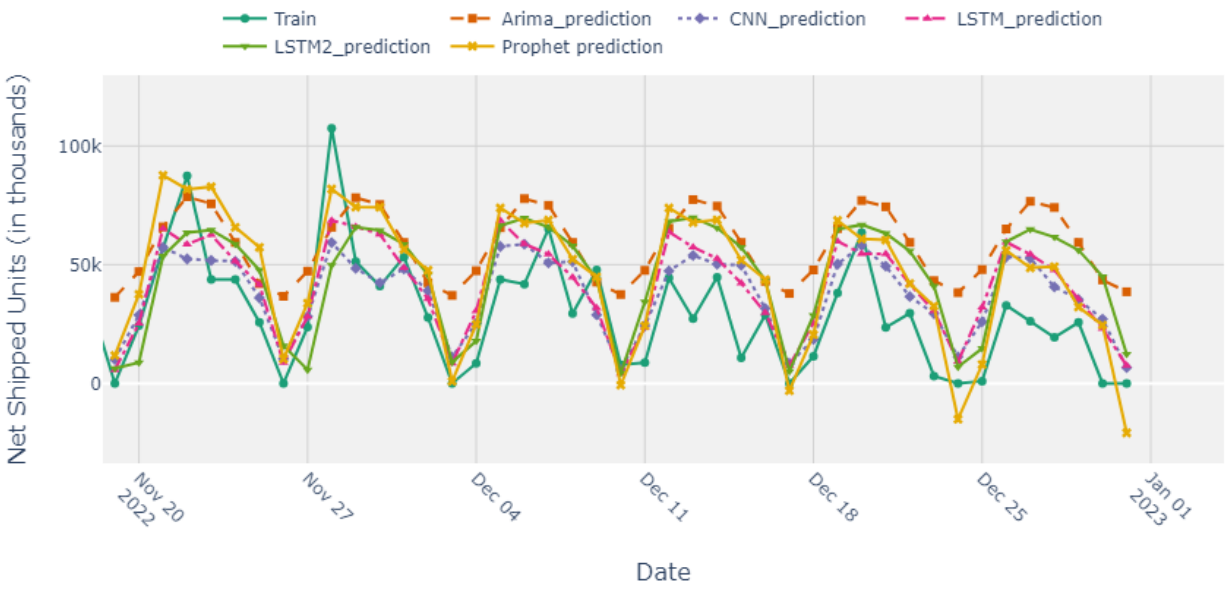

Figure 4.1 – Net Shipped Units test set prediction for Footwear products

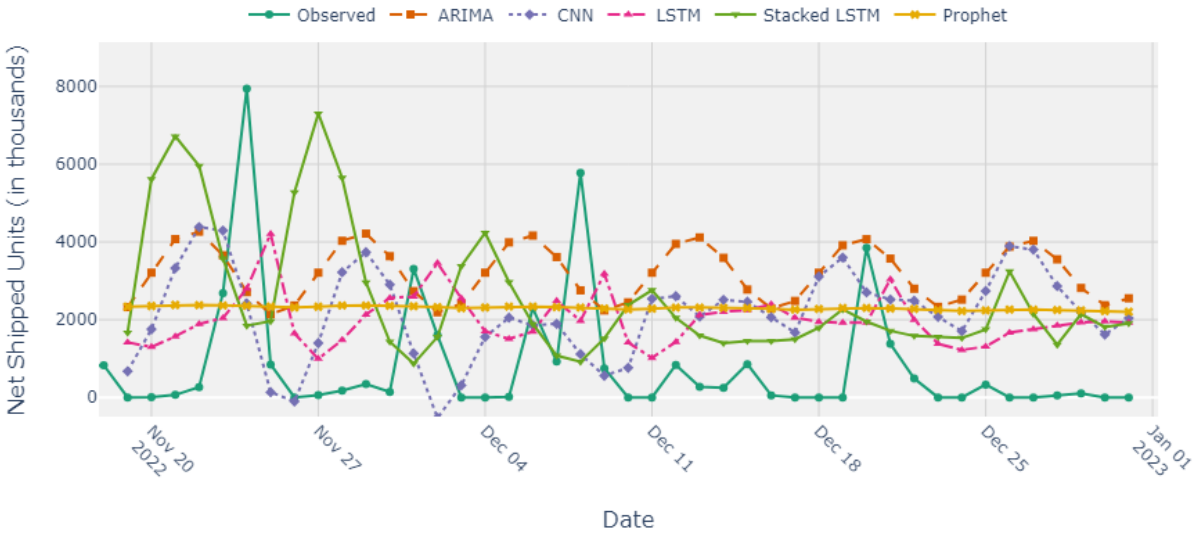

Figure 4.2 – Net Shipped Units test set prediction for Apparel products

The results obtained with this project work showed, as the literature suggests, that statistical models such as ARIMA are outperformed by more advanced techniques, namely LSTM networks or CNN networks. It is important to note that for the Apparel products, the models performed extremely poorly, especially when compared to Footwear products.

The Prophet model showed overall better results than ARIMA, which was also an expected outcome. In terms of the results however, both RMSE values and MAPE values are relatively high for the problem under study, showing that, in terms of utility to the business, the models and architectures tested out in this work might not bring the value expected. It is believed, however, that results were highly influenced by some crucial data limitations namely:

- **Limited number of observations** – as observed before in this work, the seasonality present in the data was more evident in early data. In this context, more than two years should have been provided to obtain better results with time series forecasting techniques.
- **High discrepancies in data** – with the data obtained, high volatility was present in the observations. This may have significantly affected the accuracy of the proposed models.

5. CONCLUSION

This project work aimed to investigate the effectiveness of deep learning model in time series forecasting for net shipped units, comparing them to less complex approaches such as ARIMA and Prophet. The research questions guiding this study were, Is a deep learning model more effective than less complex approaches, such as ARIMA and Prophet, in time series forecasting for net shipped units? and Which model performs best among the considered approaches for time series forecasting of net shipped units?

Through a comprehensive evaluation of the performance of deep learning models, Prophet and ARIMA, this work successfully addressed the research questions and its objectives. The main objective of model performance measurement for this type of variable was achieved by conducting a thorough comparison of the models performance based on suitable evaluation metrics, for both Footwear and Apparel products. The secondary objective was accomplished by determining that CNN and LSTM-based predictive models presented the best and most consistent results.

When looking at our findings, it can be inferred that overall, results were not as satisfiable as expected. Although the absolute error handling of the models, measured with the RMSE variable, was somewhat satisfactory for both types of products, the percentual error was too high for what is expected with the best-performing model having almost a 50% chance of predicting wrong shipped product amounts for Footwear and around 80% for Apparel.

Despite this, it is strongly believed that a successful comparison between time series forecasting methods was done, which corroborated the initial hypothesis that advanced techniques prevail over classical statistical approaches for this particular company, answering the first research question. Furthermore, and answering the second research question, the work done allows to recommend, in case the company ever decides on a time series forecasting approach, either a CNN-based or LSTM-based predictive model, as these presented the best and most consistent results.

Ultimately, this project provides valuable insights for the company when deciding to adopt a time series forecasting approach for predicting net shipped units, and its findings can inform supply chain management decisions. Forecasting problems are aways difficult to analyze and solve, and it is believed that through incentives like the one in this project, that the company will greatly benefit from the aid of advanced technology and forecasting tools, in addition to the current and somewhat traditional decision-making process.

5.1. LIMITATIONS

As already mentioned briefly in the previous section, some limitations, not only to the nature of the data but also to other factors, were faced during this project:

- **Data shape** - As already mentioned in this work, the data under scrutiny had significant peaks at certain observed values of both time series. This is a challenging factor in many predictive models and can explain the low performance of some models. However, these are not treated as outliers, since they occur consistently in the same periods of time, indicating evident yearly seasonality patterns.
- **Data size** – The previous observation brings us to point out problems with the size of the collected data. As a yearly pattern can be recognized, for a better time series analysis it was considered that a two-year range is too low to obtain accurate results, therefore more samples would have allowed more accurate predictions.
- **Lack of numerical variables** – variables such as price or demand would have allowed a multivariate time series analysis or even different machine learning approaches, which could be tested alongside time series forecasting. This however was not possible to obtain due to data collection restrictions.
- **Low computational power** – all these approaches were conducted locally with the objective of implementing a simple forecasting tool. However, better models require more memory and resources, which can be perfectly achieved in a company environment.

5.2. FUTURE IMPROVEMENTS AND RECOMMENDATIONS

To improve the final Net Shipped Units predictions and provide a better understanding of how footwear and apparel products behave in this company, it is advised to keep testing deep learning algorithms for this purpose, providing a bigger volume and better quality data. The algorithms applied in this model were fairly simple in complexity due to some hardware limitations, so a future improvement would be to test more complex models, adding more layers and testing bigger time steps for neural network-based models. As a final recommendation for the company under analysis, since LSTM and CNN performed better than the other compared models, a hybrid of both could also present substantially better results.

Additionally, with more quantitative data, other methods could be tested in the field of Machine Learning, such as regression-based approaches which take into account decisive variables that influence this type of forecasting.

Finally, it would be interesting to develop a quality visualization tool such as a dashboard to display the models under consideration, results obtained and predicted values in a more business-friendly shape, for people with more limited technical knowledge to understand and evaluate the work under development.

6. REFERENCES

Aarshay Jain. (2023, May 2). *A Comprehensive Beginner's Guide to Creating a Time Series Forecast (With Codes in Python and R)*. Https://Www.Analyticsvidhya.Com/Blog/2016/02/Time-Series-Forecasting-Codes-Python/.

Ahn, H., Sun, K., & Kim, K. P. (2021). Comparison of missing data imputation methods in time series forecasting. *Computers, Materials and Continua*, *70*(1), 767–779. https://doi.org/10.32604/cmc.2022.019369

Armstrong, J. S. (2001). Introduction. In J. S. Armstrong (Ed.), *Principles of Forecasting: A Handbook for Researchers and Practitioners* (pp. 1–12). doi:10.1007/978-0-306-47630-3_1

Baheti, P. (2022, July 29). *The Complete Guide to Recurrent Neural Networks*. https://www.v7labs.com/blog/recurrent-neural-networks-guide

Box, G. E. P., & Jenkins, G. M. (1970). *Time Series Analysis: Forecasting and Control*. Holden-Day.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2008). *Time Series Analysis*. Wiley. https://doi.org/10.1002/9781118619193

Chatfield, C. (2003). *The Analysis of Time Series*. Chapman and Hall/CRC. https://doi.org/10.4324/9780203491683

Chatfield, Christopher. (2001). *Time-series forecasting*. Chapman & Hall/CRC. https://doi.org/10.1201/9781420036206

Choi, T.-M., Hui, C.-L., & Yu, Y. (2013). *Intelligent Fashion Forecasting Systems: Models and Applications*. https://doi.org/10.1007/978-3-642-39869-8

Chu, C. W., & Zhang, G. P. (2003). A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of Production Economics*, *86*(3), 217–231. https://doi.org/10.1016/S0925-5273(03)00068-9

Cortez, B., Carrera, B., Kim, Y. J., & Jung, J. Y. (2018). An architecture for emergency event prediction using LSTM recurrent neural networks. *Expert Systems with Applications*, *97*, 315–324. https://doi.org/10.1016/j.eswa.2017.12.037

Ensafi, Y., Amin, S. H., Zhang, G., & Shah, B. (2022). Time-series forecasting of seasonal items sales using machine learning – A comparative analysis. *International Journal of Information Management Data Insights*, *2*(1). https://doi.org/10.1016/j.jjimei.2022.100058

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735

Ji, S., Yu, H., Guo, Y., & Zhang, Z. (2016, December 23). Research on sales forecasting based on ARIMA and BP neural network combined model. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3028842.3028883

Khandelwal, I., Adhikari, R., & Verma, G. (2015). Time series forecasting using hybrid arima and ann models based on DWT Decomposition. *Procedia Computer Science*, *48*(C), 173–179. https://doi.org/10.1016/j.procs.2015.04.167

Kumar, J., Goomer, R., & Singh, A. K. (2018). Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model for Cloud Datacenters. *Procedia Computer Science*, *125*, 676–682. https://doi.org/10.1016/j.procs.2017.12.087

Loureiro, A. L. D., Miguéis, V. L., & da Silva, L. F. M. (2018). Exploring the use of deep neural networks for sales forecasting in fashion retail. *Decision Support Systems*, *114*, 81–93. https://doi.org/10.1016/j.dss.2018.08.010

Pai, P. F., & Lin, C. S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*, *33*(6), 497–505. https://doi.org/10.1016/j.omega.2004.07.024

Papacharalampous, G., Tyralis, H., & Koutsoyiannis, D. (2018). Predictability of monthly temperature and precipitation using automatic time series forecasting methods. *Acta Geophysica*, *66*(4), 807–831. https://doi.org/10.1007/s11600-018-0120-7

Pinho, A., Costa, R., Silva, H., & Furtado, P. (2019). *Comparing Time Series Prediction Approaches for Telecom Analysis* (pp. 331–345). https://doi.org/10.1007/978-3-030-26036-1_23

Ramos, P., Santos, N., & Rebelo, R. (2015). Performance of state space and ARIMA models for consumer retail sales forecasting. *Robotics and Computer-Integrated Manufacturing*, *34*, 151–163. https://doi.org/10.1016/j.rcim.2014.12.015

Santos, L., Matos, L. M., Ferreira, L., Alves, P., Viana, M., Pilastri, A., & Cortez, P. (2022). A Sequence to Sequence Long Short-Term Memory Network for Footwear Sales Forecasting. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *13756 LNCS*, 465–473. https://doi.org/10.1007/978-3-031-21753-1_45

Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2018). A Comparison of ARIMA and LSTM in Forecasting Time Series. *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, 1394–1401. https://doi.org/10.1109/ICMLA.2018.00227

Taylor, S. J., & Letham, B. (n.d.). *Forecasting at Scale*. https://doi.org/10.7287/peerj.preprints.3190v2

Taylor, S. J., & Letham, B. (2013). Forecasting at Scale. *PeerJ*. https://doi.org/https://doi.org/10.7287/peerj.preprints.3190v2

Thomas, E. B. (2023, July 29). Understanding LSTM: An in-depth look at its architecture, functioning, and pros & cons. *Medium*. https://medium.com/@ebinbabuthomas_21082/understanding-lstm-an-in-depth-look-at-its-architecture-functioning-and-pros-cons-5424dd7215e6

Thomassey, S. (2010). Sales forecasts in clothing industry: The key success factor of the supply chain management. *International Journal of Production Economics*, *128*(2), 470–483. https://doi.org/10.1016/j.ijpe.2010.07.018

Wibawa, A. P., Utama, A. B. P., Elmunsyah, H., Pujianto, U., Dwiyanto, F. A., & Hernandez, L. (2022). Time-series analysis with smoothed Convolutional Neural Network. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00599-y

Yaremko, S. A., Kuzmina, E. M., Savina, N. B., Yepifanova, I. Yu., Gordiichuk, H. B., & Mussayeva, D. (2022). FORECASTING BUSINESS PROCESSES IN THE MANAGEMENT SYSTEM OF THE CORPORATION. *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, *12*(4), 53–59. https://doi.org/10.35784/iapgos.3249

Yu, Q., Wang, K., Strandhagen, J. O., & Wang, Y. (2018). Application of Long Short-Term Memory Neural Network to Sales Forecasting in Retail—A Case Study. *Lecture Notes in Electrical Engineering*, *451*, 11–17. https://doi.org/10.1007/978-981-10-5768-7_2

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 159–175. doi:10.1016/S0925-2312(01)00702-0

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa