Facultad de Ciencias Económicas y Empresariales

Product Demand Prediction using Neural Networks with Attention

Autor: Roberto Gozalo Brizuela
Director: Jenny Alexandra Cifuentes Quintero

MADRID| Diciembre 2024

Resumen

El Deep Learning ha experimentado un enorme crecimiento en el campo de la investigación de predicción de ventas en los últimos años, ya que modelos más complejos impulsados por redes neuronales parecen capaces de superar a los modelos tradicionales en la previsión. A través de este proyecto, se examina un conjunto de datos sobre ventas en tiendas para realizar un análisis de previsión de ventas utilizando modelos LSTM de última generación. El objetivo de la investigación es dilucidar si los mecanismos de atención mejoran la previsión de ventas. Se implementará el modelo LSTM con Mecanismos de Atención y se comparará con los modelos LSTM tradicionales (Vanilla LSTM, Stacked LSTM, Bidirectional LSTM y Convolutional LSTM).

A través de nuestros experimentos, el modelo más preciso al comparar a través de RMSE fue el Convolutional LSTM, seguido por el LSTM con Mecanismos de Atención, el Bidirectional LSTM y el Vanilla LSTM en último lugar. Esto nos mostró que la técnica recientemente propuesta de incluir mecanismos de atención a un modelo LSTM tradicional es muy comparable a los modelos anteriores de última generación e incluso supera a la mayoría de las otras técnicas en nuestro experimento con la segunda volatilidad de error más baja también.

Palabras clave: Predicción de series temporales, Predicción de ventas, Redes neuronales, Aprendizaje profundo, LSTM

Abstract

Deep Learning has seen a huge growth in the field of sales prediction research over the last years, as more complex models driven by neural networks seem able to outperform traditional models in forecasting. Through this project, a dataset regarding store sales is examined in order to perform a sales forecasting analysis regarding state-of-the-art LSTM models. The aim of the investigation is to elucidate whether attention mechanisms improve sales forecasting. The LSTM model with Attention Mechanisms will be implemented and compared to traditional LSTM models (Vanilla LSTM, Stacked LSTM, Bidirectional LSTM and Convolutional LSTM).

Through our experiments, the most accurate model when comparing through RMSE was the Convolutional LSTM, followed by the LSTM with Attention Mechanisms, the Bidirectional LSTM and the Vanilla LSTM coming at last. This showed us that the newly proposed technique of including attention mechanisms to a vanilla LSTM model is very comparable to state-of-the-art previous models and even outperforms most other techniques through our experiment with the second lowest error volatility.

Keywords: Time-series Forecasting, Sales Forecasting, Neural Networks, Deep Learning, LSTM

Acknowledgments

Thanks to my family and friends. Also, thanks to my tutor who has helped me a lot throughout the completion of my work.

Contents

List of Figures

List of Tables

Acronyms

ANN	Artificial Neural Networks
ANFIS	Adaptive Network-based Fuzzy Inference System
ARFIMA	Autoregressive fractionally integrated moving average
ARIMA	Autoregressive integrated moving average
DNN	Deep Neural Networks
DT	Decision Trees
ES	Exponential Smoothing
ETS	State-Space model
GLM	Generalized Linear Model
GRU	Gate Recurrent Unit
HLM	Holt's Linear Method
HW	Holt-Winters model
KNN	K-nearest neighbours algorithm
LR	Linear Regression
LSE	Least Squares Estimate
LSTM	Long-Short Term Memory
LightGBM	Light Gradient Boosting Model
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLP	Multilayer Perceptron
MSE	Mean Squared Error
RF	Random Forest
RMPSE	Root Mean Percentage Squared Error
RNN	Recurrent Neural Network
RMSE	Root Mean Squared Error
SARIMA	Seasonal Autoregressive integrated moving average
SVR	Support Vector Regression
WMA	Weighted Moving Average
WM	Winter's Method
XGBoost	Extreme Gradient Boosting

Chapter 1

Introduction

1.1 Motivation

In today's fast-paced and increasingly competitive business environment, the ability to accurately forecast product demand is more critical than ever. Effective demand prediction not only ensures optimal inventory management, reducing the risk of overstocking or stockouts, but also facilitates more informed strategic planning and resource allocation (Blum, 2020; Tadayonrad & Ndiaye, 2023). Accurate demand forecasts enable businesses to predict their production schedules, manage supply chain logistics efficiently, and optimize their financial planning (de Carvalho Lima, Firmino, & Rocha, 2023; Makridakis, Hyndman, & Petropoulos, 2020). This, in turn, enhances customer satisfaction by ensuring product availability while simultaneously controlling costs. In industries where trends and consumer preferences evolve rapidly, such as fashion, technology, and consumer goods, the agility provided by precise demand forecasting becomes a significant competitive advantage.

Despite these advantages, current industry practices reveal significant gaps in the effective use of sales forecasting models. Surveys indicate that only 40% of sales forecasting opportunities actually result in closure, suggesting ample room for improvement in this domain (Rotenberg & Lindquist, 2013). In addition, the study carried out by (G. Research, 2020) further reveals that a staggering 90% of sales companies may prefer intuition over data analytics in their forecasting processes, indicating a reluctance to fully embrace advanced technological solutions for navigating uncertain futures. Furthermore, even among those employing sales forecasting models, (Blum, 2020) reports that about 50% of sales leaders lack confidence in these forecasts. This lack of confidence often leads to a reversion to intuition-based methods, which, while familiar, do not provide the reliability needed for accurate forecasting (Xu, Tang, & Rangan, 2017). The inherent biases in expert judgments, such as over-confidence, anchoring, illusory patterns, and group thinking, further compound these challenges.

Consequently, the development of robust and reliable demand prediction models has become a pivotal focus for businesses seeking to thrive in an ever-changing market landscape. However, achieving high precision in demand forecasting remains challenging due to the unique demand characteristics of each product. The variability in consumer preferences, market trends, and external factors like economic conditions or seasonal changes means that demand patterns can be highly idiosyncratic (Padilla,

García, & Molina, 2021). Furthermore, the complexity increases when considering the diverse range of products a company might offer, each with its own demand cycle and influencing factors. This complexity requires the development of advanced models that can capture and analyze these nuances, thereby enabling more accurate predictions.

In this research, a dataset covering retail store sales from 2014 to the end of 2017 has been analyzed. This dataset is composed of three distinct product segments: office supplies, technology, and furniture. Among these, the furniture category has been specifically selected for a comprehensive analysis focused on sales forecasting. This selection is primarily due to the unique seasonal sales patterns evident in the furniture category, a feature not commonly found in publicly accessible datasets. With the global furniture market evaluated at USD 630.55 billion in 2022 and expected to reach approximately USD 1,051.77 billion by 2032, growing at a CAGR of 5.3%, the sector's analysis is particularly pertinent (see Figure 1.1). The historical data's seasonality provides a solid foundation for future demand predictions, offering crucial insights into consumer behavior in both residential and commercial realms (Bednárik & Pakainé Kováts, 2010). The analysis of this category, therefore, holds significant potential for enhancing the understanding of market trends and customer preferences. The findings from this study are anticipated to be instrumental in shaping effective strategies for inventory management, marketing, and comprehensive business planning. This is particularly relevant for products that demonstrate similar seasonal trends and market behaviors, thereby allowing businesses to make more informed decisions in these key operational areas.

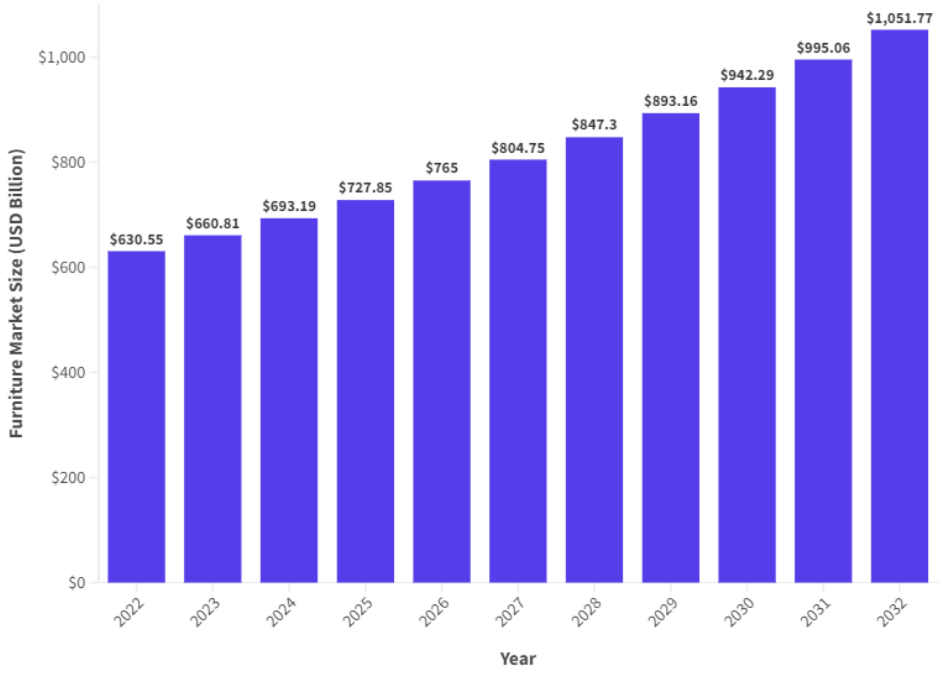

Figure 1.1: Furniture Market Size, 2022 to 2032. From: Precedence Research (P. Research, 2023). Own Elaboration

Considering that seasonal time-series forecasting is pivotal for strategic decision-making and planning future activities, different forecasting strategies have been pro-

posed. As such, traditional models like Seasonal Autoregressive Integrated Moving Average (SARIMA), Autoregressive Integrated Moving Average (ARIMA), and the Holt-Winters Triple Exponential Smoothing model have been extensively employed in this area. In contrast, modern methods such as Light Gradient Boosting Model (LightGBM), Facebook Prophet, Random Forest, and Long-Short Term Memory models (LSTM) models (Ali & Nakti, 2023; Fierro Torres, Castillo Pérez, & Torres Saucedo, 2022; Hasan, Kabir, Shuvro, & Das, 2022) are acknowledged for their superior handling of non-linear data (da Fonseca Marques, 2020). In this context, the focus of this study is on using past sales data for future furniture sales forecasting, rather than analyzing the performance of the retail store itself. This research aims to investigate the capabilities of neural network models with attention mechanisms in seasonal sales forecasting, where LSTM models with attention mechanisms have been identified as particularly effective in other time series forecasting applications (Hollis, Viscardi, & Yi, 2018; Wen & Li, 2023). A comparison of this proposed approach with various deep learning (DL)-based forecasting methods is undertaken, aiming to conclude about the most effective approach. To our knowledge, this specific methodology has not been previously applied to sales forecasting, offering potential benefits for business operations.

1.2 General and Specific Objectives

1.2.1 General Objective

The main goal of this thesis is to propose and evaluate a product demand prediction model, based on neural networks with attention mechanisms, and compare its performance with other *Deep Learning* (DL) time series prediction algorithms. The study seeks to provide a deeper understanding of the effectiveness of attention models in demand prediction and offer practical recommendations for its application in real business scenarios.

1.2.2 Specific Objectives

- Contextualize the importance of product demand forecasting in business planning, highlighting its role in improving efficiency and profitability.
- Analyze current methodologies for product demand forecasting, with a focus on DL techniques, identifying and comparing the most common techniques for time series forecasting in this field of application.
- Understand neural network strategies with attention mechanisms, addressing their performance and applicability in time series forecasting.
- Implement a recurrent neural network model with attention mechanisms in the context of cabinet demand prediction.
- Propose a product demand prediction model based on neural networks with attention mechanisms and validate its performance through comparative experiments with other DL time series prediction techniques.

1.3 Document Structure and Organization

This document is organized into seven chapters. Chapter 2, the state-of-the-art review, presents a comparison of significant sales forecasting methodologies, contrasting results from traditional and DL techniques. Chapter 3 is dedicated to elucidating modern DL models, including various LSTM Recurrent Neural Network architectures. The methodology for applying these models to the dataset is detailed in Chapter 4. Chapter 5 discusses the performance of these models within the specific context of this research. Finally, conclusions drawn from the study are summarized in Chapter 5.

Chapter 2

Literature Review

Sales forecasting is a key component of a company's operational strategy, providing essential insights that significantly improve the accuracy of future outcome predictions. This ability to foresee upcoming trends is critical in guiding investment decisions, optimizing inventory management, and impacting various other core aspects of business operations. However, despite its significance, many companies still predominantly rely on the intuitive judgments of their sales staff for forecasting. To address this gap, various strategies have been proposed, encompassing traditional models such as the Box and Jenkins model, Autoregressive Integrated Moving Average (ARIMA), linear regression (LR), and Holt-Winters (HW), each offering a structured approach to predict sales more accurately and systematically.

In the specific domain of sales forecasting, two predominant methodologies emerge as focal points: Monte Carlo (MC) simulations and time series regression analysis. A detailed study by (Paixão & da Silva, 2019) provides an in-depth comparison of these methodologies, analyzing a historical dataset of mechanical components sales from 2011 to 2018. Monte Carlo simulations, efffective in predicting highly variable and unpredictable factors, offer a unique approach in the unpredictable realm of sales forecasting. This method employs stochastic simulations generated through random variables, a technique beneficial in scenarios where the independent variable is inherently uncertain.

Conversely, time series regression analysis adopts a deterministic method, leveraging historical data to forecast future trends. The research reported by (Paixão & da Silva, 2019) examines various time series analysis techniques, including moving average (MA), weighted moving average (WMA), least squares estimate (LSE), and HW, and compares them with Monte Carlo simulations. The results demonstrate a consistent superiority of time series regression analysis techniques over Monte Carlo simulations, with MA and WMA emerging as especially effective. This outcome shows the effectiveness of deterministic approaches, which are based on historical data, over the random and stochastic processes of Monte Carlo simulations in the sales forecasting field.

In addition, a notable study proposed by (Demir & Akkaş, 2018) researches the use of models for a dataset containing the daily sales of a feed company, using real-world data. This research conducts an extensive comparison between traditional sales forecasting methods, such as MA, Exponential Smoothing (ES), Holt's Linear Method (HLM), and Winter's Method (WM), and modern approaches like Artificial Neural

Networks (ANN) and Support Vector Regression (SVR). Surprisingly, these research works, including that proposed by (Green & Armstrong, 2015), suggest the efficacy of simpler models, raising the question of whether complex models truly surpass the accuracy of simpler ones. Analyzing sales data from May 2014 to March 2015 across five different products, the last study evaluates the distinct benefits of each forecasting method, acknowledging that no single methodology is universally applicable to all scenarios. However, the study reveals that while there is no single best-performing traditional model, as their effectiveness varies across different products, they generally underperform compared to non-traditional methods employing machine learning-based algorithms like ANN and SVR.

Building on ANNs, (Ansuj, Camargo, Radharamanan, & Petry, 1996) focuses on predicting the sales of a medium-sized enterprise in Brazil from 1979 to 1989. This study was groundbreaking as it highlighted the superior performance of ANNs, particularly in outperforming the ARIMA model in a dataset characterized by noise, seasonality, non-stationarity, and randomness. The capacity of ANNs to capture non-linearities, as emphasized in further studies (Adya & Collopy, 1998) and (Zhao et al., 2009), proves fundamental for capturing data patterns. This ability not only enhances predictability (Tkáč & Verner, 2016) but also showcases their remarkable adaptability and learning capabilities when confronted with real-world datasets marked by noise (Livieris, Kiriakidou, Kanavos, Vonitsanos, & Tampakas, 2019).

Expanding on previous research on ANNs, the field of sales forecasting really improved predictions with Deep Neural Networks (DNNs). Unlike traditional ANNs, DNNs are distinguished by their multiple hidden layers and a specialized memory block that retains prior input data, facilitating a deeper extraction data across various levels of complexity (Rafi & Karim, 2020). DL models, especially DNNs, have shown their effectiveness in handling complex networks and executing sophisticated prediction calculations more efficiently. Their core strength lies in adeptly identifying and learning from complex patterns within extensive datasets, a key advantage highlighted in (LeCun, Bengio, & Hinton, 2015). Moreover, a critical aspect of DNNs is their exceptional ability to perform feature extraction. They excel at transforming unstructured data into structured, coherent vectors of information, a process that is particularly challenging for traditional ARIMA models, as noted in (Murray, Du Bois, Hollywood, & Coyle, 2023).

In this research line, a pioneering study by (Das & Chaudhury, 2007) introduces a DNN architecture for predicting weekly footwear sales. Specifically, it employs a Recurrent Neural Network (RNN),as detailed in (Lipton, Berkowitz, & Elkan, 2015). Reported results have demonstrated RNNs ability to accurately predict footwear sales for the upcoming week or month and a half, achieving an impressive Mean Absolute Percentage Error (MAPE) of 9% for short-term forecasts within this dataset. Specifically, RNNs are particularly adept at processing information sequentially, managing data one element at a time while adeptly capturing dependencies and patterns across time sequences. This characteristic makes them exceptionally suitable for modeling sequences where inputs are interdependent and vary over time. An added benefit of RNNs, as emphasized in (Eddy & Allman, 2000), is their parallelization capability, which significantly reduces the training time. Extensive studies across diverse fields that utilize RNNs consistently confirm their effectiveness in not just short, but also

medium and long-term forecasts, highlighting the critical role of hidden layers that retain memory of past inputs (Yasdi, 1999). This factor is instrumental in RNNs' ability to outperform traditional ANNs in time series analysis, reinforcing their significance in this domain (Rafi & Karim, 2020).

Considering the advancements in models based on RNNs, further research efforts have shifted their focus towards a specialized variant known as Long Short-Term Memory (LSTM) networks. LSTMs are distinguished by their capacity to retain and process large amounts of data over prolonged periods, a feature extensively discussed in (Staudemeyer & Morris, 2019). Their unique ability to manage long-term dependencies, which includes recalling data from earlier stages in the time series, plays a crucial role in significantly enhancing the accuracy of predictions, as highlighted in (Q. Yu, Wang, Strandhagen, & Wang, 2018).

A notable example of this is the study by (Q. Yu et al., 2018), where LSTMs were used to predict sales figures for 66 products across a 45-week dataset. The model uses data from the latter four weeks to forecast sales in the fifth week. Despite achieving low forecasting errors for only 25% of the products, this was attributed to data scarcity rather than any inherent limitations of the LSTM architecture. Their superior performance comes from the fact that, unlike traditional models, LSTM networks effectively model long-term dependencies and patterns in sequential data, making them exceptionally well-suited for time-series forecasting (Beheshti-Kashi, Karimi, Thoben, Lütjen, & Teucke, 2015). This capability allows companies to enhance the robustness of their forecasting models, making for easier adaptation to seasonal variations and evolving data trends. Additionally, LSTM models avoid the vanishing gradient problem, a big issue in other machine learning approaches (Almqvist, 2019).

However, it is important to recognize that the implementation of LSTM RNNs is associated with challenges, notably their complexity and computational requirements. In response, recent research efforts have been directed towards evaluating the comparative effectiveness of modern and traditional forecasting models. These studies highlight the necessity of balancing accuracy with practicality in forecasting models (Haselbeck, Killinger, Menrad, Hannus, & Grimm, 2022; Quevedo, 2020).

A study by (Elmasdotter & Nyströmer, 2018) conducted a comparative analysis of the LSTM model with the ARIMA model, using a sales dataset from an Ecuadorian grocery chain. It was found that LSTMs outperformed the ARIMA model in capturing the complexity of sales forecasting problems, demonstrating superior predictive capabilities by addressing the nonlinearities inherent in sales data. The iterative optimization algorithm used in the LSTM approach was a contributing factor to its efficacy compared to other models (Siami-Namini, Tavakoli, & Namin, 2018). However, it was noted that the successful implementation of LSTM networks requires hyperparameter optimization, as a universally applicable set of hyperparametersparameters may not exist for all forecasting problems (Elmasdotter & Nyströmer, 2018).

Regarding studies researching LSTM models further than the Vanilla LSTM model, a study conducted by (Murugesan, Mishra, & Krishnan, 2021) is highlighted, in which time series analysis is performed on various LSTM architectures. These include the basic LSTM, Bidirectional LSTM, Stacked LSTM, and Convolutional Neural Network (CNN) LSTM. A historical dataset including agricultural commodities prices is used in this research, forecasting prices for commodities such as wheat, gram, banana, rice,

and groundnut. The research achieves high forecasting accuracy, with a 95% confidence interval demonstrated across all LSTM-based models. Notably, the CNN-LSTM variant exhibits slightly superior forecasting results compared to other LSTM techniques, highlighting its efficacy in this context.CNN-LSTMs combine the strengths of CNNs and LSTMs, making them highly effective for tasks involving spatial-temporal data. The CNN layers are great at extracting spatial features within the data, while the LSTM layers capture temporal dependencies, resulting in a robust model that can understand complex, multi-dimensional datasets (Wang, Cao, & Philip, 2020).

Additional studies have conducted comparative analyses between traditional methods and modern DL-based technologies. A significant study by (Ensafi, Amin, Zhang, & Shah, 2022) compares classical models like ARIMA and SARIMA against advanced forecasting models, including LSTM and CNN, using a retail sales dataset focused on furniture. The findings reveal the Stacked LSTM's superior performance, with the CNN model also showing notable success. The main advantage of Stacked LSTM, which underpins these findings, lies in its ability to effectively layer multiple LSTM units This layered structure and enhanced memory capability are key to the model's predictability (Pavlyshenko, 2019) This is shown in the 156.15% reduction in Root Mean Square Error (RMSE) when comparing the best traditional model (SARIMA 1) to the top machine learning model (Stacked LSTM).

Initially renowned for its remarkable achievements in image synthesis, in which it demonstrates a high-fidelity replication of inputs, the Autoencoder has undergone successful adaptations for the domain of sales forecasting. Within this context, the model distinguishes itself by because of being great at representing abstract data representations (great for feature extraction) (W. Yu, Kim, & Mechefske, 2021). This can be great for models in other fields such as anomaly detection (Meng, Catchpoole, Skillicom, & Kennedy, 2017). The Autoencoder's capacity for abstraction comes from its training, where reward mechanisms are used. This methodological approach has yielded promising outcomes, as evidenced by its application to a dataset originating from an Ecuadorian pharmacy, thereby substantiating its practical utility and efficacy in real-world scenarios (Chang et al., 2017).

Furthermore, the application of attention mechanisms in time series prediction has gained notable attention in recent researches. Attention mechanisms have proven to be crucial in enhancing the forecasting accuracy of various models. Notably, a study conducted by (Li, Yang, Zhu, & Zhang, 2021) proposes a sales forecasting model using a historical dataset of clothing sales. The study demonstrated that the integration of attention mechanisms led to improved predictions compared to traditional individual models such as Prophet, Vanilla Gated Recurrent Unit (GRU), Vanilla LSTM, Vanilla RNN, and ARIMA. The research specifically highlighted the superior performance of both the composite GRU-Prophet model with attention mechanisms and the GRU model with attention mechanisms. These models outperformed invidual models, showcasing the efficacy of attention mechanisms in capturing complex patterns within sales data. However, despite these promising results, there remains a gap in comprehensive comparative analysis to evaluate other LSTM variants, some of which have reported even better performances. This indicates a need for further exploration and validation across a broader range of LSTM-based models.

Moreover, an alternative approach within the domain of sales forecasting involves

the deployment of ensemble methods, in which multiple models operate independently, and their predictions are subsequently combined during the forecasting phase. Ensemble methods have demonstrated high effectiveness by harnessing the unique strengths of underlying individual models. A case in point is the study conducted by (Fleurke, 2017) which explores the application of ensemble techniques in predicting automobile sales data. This comprehensive study encompasses datasets spanning total sales in the USA from 1992 to 2017 and new car registrations in the Netherlands from 2012 to 2017. By combining the predictive capabilities of five uncorrelated models (HW, ANN, Theta, Random Forest (RF), and a Generalized Linear Model (GLM)) through a straightforward averaging approach, significant results were achieved. Impressively, the ensemble consistently outperforms the individual forecasting models, highlighting its potential despite its relative simplicity.

Another remarkable research related to ensemble methods combines ARIMA and ANNs to enhance forecasting accuracy for retail clothing sales prediction (S. Yu, Dong, Chen, He, & Shi, 2019). This model strategically leverages the linear information learned by ARIMA and the non-linear insights acquired by ANNs. In contrast to the straightforward averaging approach used by (Fleurke, 2017), this study incorporates the expected error to determine model weights. The combined model, when compared to standalone models, consistently exhibits superior performance.

In addition to this approach, stacking strategies have also been explored as ensemble methods for generating forecasts. Unlike ensemble strategies that simply average the outputs of different models, stacking strategies integrate diverse learners as sequential layers within the model. A notable application of this approach is in the study by (Jiang, Fan, Sun, & Liu, 2021), where ARIMA, LSTM, and XGBoost are used as base models and LightGBM as the final prediction model. This study concludes that stacking effectively consolidates the strengths of each individual model, resulting in enhanced forecasting performance beyond what ARIMA and LSTM could achieve alone. However, it is important to acknowledge that stacking models can lead to longer training times and increased computational costs due to the integration of multiple layers.

Considering the literature previously reviewed, this project is dedicated to a comparative analysis of various DL architectures, placing special focus on different LSTM variations and the integration of attention models. The inclusion of attention mechanisms, as informed by the literature review, is expected to exhibit enhanced performance compared to many existing DL frameworks. Moreover, the comparison will consider various state-of-the-art LSTM network variations. This methodology seeks to achieve an ideal balance between computational burden and performance efficiency. The objective of this analysis is to identify the most efficient and effective approach for complex data, with the potential to set new benchmarks in the product demand forecasting field.

Citation	Goal	Dataset	Models	Results
(Paixão & da Silva, 2019)	Comparing sale forecasting performance between a time series analysis approach (deterministic) and a Monte-Carlo simulation approach (stochastic).	Specific mechanic component sales from September 2011 to August 2018.	MA, WMA, HW, LSE and Monte-Carlo Estimation.	The most effective models were the MA and the WMA, based on time-series analysis. MAPE for the last tested month was 14% for MA, 14% for WMA, 19% for HW, 18% for LSE and 18% for MC, showing the effectiveness for MA and WMA.
(Demir & Akkaş, 2018)	Comparing sales forecasting performance of traditional models against modern machine learning-based models	Feed company's sales from May 2014 to March 2015	MA, ES, HLM, and SVR.	The results show no clear traditional best-performing model. When compared to non-traditional models (SVR and ANN), non-traditional models clearly outperform traditional models. As an example, for product 5, MAPE was 25% for MA, 35% for ES, 32% for HLM, 32% for WM, 13% for ANN and 6% for SVR, showing outperformance from modern models (SVR and ANN).
(Loureiro, Miguéis, & da Silva, 2018)	Evaluating the performance of Deep Neural Networks in sales forecasting.	Product sales of 684 types of women bags.	DNN, DT, RF, SVR, ANN, and LR.	Performance metrics showed modern models outperforming, with the DF being the best-performing model, as we can see through MAPE (0.38 for DNN, 0.38 for DT, 0.35 for DF, 0.39 for SVR, 0.39 for ANN, and 0.45 for LR).
(Lakshmanan, Vivek Raja, & Kalathiappan, 2020)	Comparing a basic LSTM model forecasting to a baseline of other more traditional models. This is done for four different prediction ranges: one week, two weeks, three weeks and four weeks.	Store and sales information from 2013 to 2015.	LSTM, KR, ANNs.	For all the prediction ranges, the LSTM model outperformed all of the other models. MSE for the LSTM model was 2.89 while it was 4.52 for the ANN, 6.72 for the MLP and 12.1 for the LR.
(Murugesan et al., 2021)	Comparing time series forecasting performance of different types of LSTM models.	Five agricultural commodity prices from January 2000 to July 2020.	Basic LSTM, bidirectional LSTM, Stacked LSTM, CNN LSTM	All models obtained good results at a 95% interval. The different commodities showed a different best-performing model. For Rice prices, MAE was 0.19 for Basic LSTM, 0.26 for Bi LSTM, 0.27 for Stacked LSTM, 0.29 for CNN LSTM.
(Fleurke, 2017)	To evaluate whether employing an ensemble approach can enhance the accuracy of forecasting	Car registration in the Netherlands from 2012 to 2017 and total vehicle sales in the USA from 1992 to 2007	ES, ARIMA, ANN, VAR, Theta, RF, GLM, and Ensembles.	Performance metrics show that the combination of models with a MAPE of 10.3% and the GLM with a MAPE of 10% are the most effective models for this set of data.
(S. Yu et al., 2019)	The study investigates the efficacy of integrating ARIMA models with BP ANNs in enhancing the precision of sales forecasting.	Sales of a clothing seller from March 2014 to December 2018	ARIMA, BP Neural Network and ARIMA-BP	Metrics showed outperformance by the ensemble model, with a 7.3% MAPE. Individual models ARIMA and BP have MAPEs of 27.7% and 15.5% respectively.
(Jiang et al., 2021)	This research delves into the potential of enhancing sales forecasting by employing a composite model that synergizes ARIMA, LSTM, and XGBoost as foundational models, with LightGBM serving as the mechanism for final predictions	Pharmaceutical sales of a certain company in 2020	ARIMA-LightGBM, LSTM-LightGBM and XGBoost-LightGBM	MAPE for 5.7% for ARIMA, 5.2% for LSTM, 5.4% for XGBoost and 4.9% for Stacking. Stacking techniques such as this one improve accuracy but have a higher computational cost.

Citation	Goal	Dataset	Models	Results
(Li et al., 2021)	Implementing attention mechanisms in a composite GRU- Prophet model and comparing the performance to the baseline of other models.	Daily sales of a product from September 2016 to February 2019	GRU-Prophet with Attention Mechanisms, GRU with Attention Mechanisms, Prophet, GRU, LSTM, RNN, and ARIMA	The best-performing models are those that include attention mechanisms, with the GRU-Prophet with Attention Mechanisms and the GRU with attention mechanisms. Attention mechanisms appear to drop MAE from 0.091 to 0.09. This paper provides insights into how models including attention mechanisms can improve the forecast's accuracy.

Table 2.1: Comparison of research papers related to sales forecasting techniques.

Chapter 3

Theoretical background

DL, a subset of machine learning, is a rapidly evolving field that has significantly transformed the landscape of artificial intelligence. Characterized by its ability to learn from and interpret complex data structures through layered computational models, DL has found applications across diverse domains ranging from natural language processing to image recognition. At the core of DL are ANNs, which are great at interpreting complex data. These networks, which we can see in Figure 3.1, are structured in layers comprising interconnected nodes or 'neurons'. The input layer receives the initial data, which is then processed through one or more hidden layers (LeCun et al., 2015). Each neuron in these layers transforms the input, extracting features and identifying patterns before passing the information to the subsequent layer. The process culminates in the output layer, where the final analysis or prediction is generated. This architecture allows ANNs to learn from high-dimensional data, enabling sophisticated applications such as image and speech recognition, and complex time series forecasting.

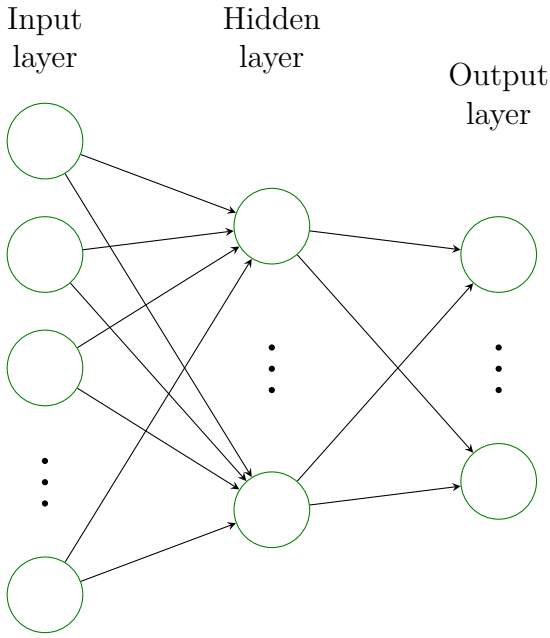

Figure 3.1: Neural network architecture from own elaboration

The widespread adoption DL can be attributed to several overarching advantages

(Sarker, 2021). DL models have proved unparalleled proficiency in capturing nonlinear relationships within datasets. Their ability to self-learn feature representations (bypassing the need for manual feature engineering) streamlines model development and enhances scalability. Additionally, the continuous advancements in computational power, alongside the increasing availability of large datasets, have fueled their ascendancy. These factors, combined with DL's versatility in handling a variety of complex tasks, from vision systems to natural language understanding, render it a powerful tool in the modern AI toolkit.

3.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are distinguished by the presence of backward connections between neurons, which enables them to effectively process sequential data (Oliver-Muncharaz, 2020). As it can be seen in Figure 3.5, each recurrent unit within an RNN, composed of multiple neurons, calculates an output at every time step. This output, designated as y_t, is dependent on the next temporal process. As the network goes onto the next time step, each neuron receives a new input vector x_t and also considers the output from the previous time step, y_{t-1}, which serves as the recurrent input. This streamlined architecture of RNNs facilitates the network's ability to draw upon previously acquired information, which is essential for managing long short-term dependencies. The neuron then employs an activation function, denoted as θ, to compute the output vector based on both the current input vector (x_t) and the recurrent input (y_{t-1}). Common choices for θ include linear, sigmoidal, or hyperbolic tangent (tanh) functions, with the tanh function often preferred in time series applications due to its efficacy in handling temporal data (Oliver-Muncharaz, 2020).

The operational principle of an RNN can be summarized by the following equation:

$$y_t = \theta(h_x \cdot x_t + h_y \cdot y_{t-1} + b) \tag{3.1}$$

where h_x and h_y represent the weight matrices for the input and the recurrent input, respectively, and b is the bias term. The function θ acts upon the linear combination of the inputs and their corresponding weights, plus the bias, to generate the output y_t for each time step

RNNs present significant advantages for time series forecasting due to their unique ability to process sequential data. Unlike traditional ANNs, RNNs can retain information from previous inputs, enabling them to understand temporal dynamics and contextual dependencies. This memory component allows for more accurate predictions in scenarios where historical data is the focus. RNNs also accommodate variable-length input sequences, a notable advantage over ANNs which require fixed-size inputs (Hewamalage, Bergmeir, & Bandara, 2021). However, while RNNs are adept at processing sequential data for time series forecasting, they face challenges like the vanishing gradient problem, where training becomes ineffective over long sequences. This limitation slow their ability to capture long-term dependencies. Additionally, RNNs can struggle with overfitting, especially in complex models with large datasets. To overcome these issues, advanced variants such as LSTM networks GRU have been developed. These new approaches collectively make RNNs a more effective choice for complex time series forecasting tasks

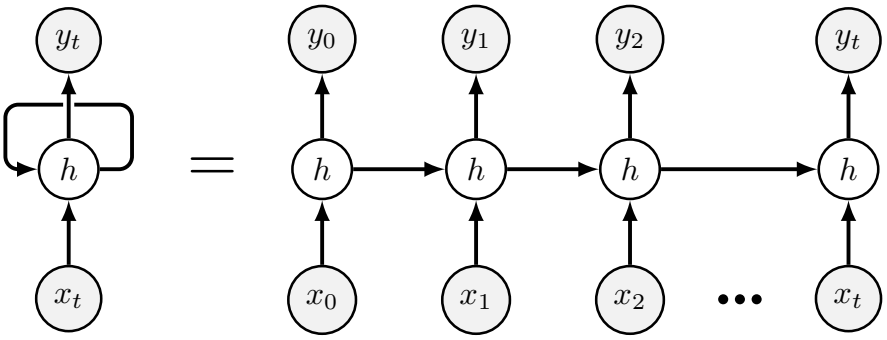

Figure 3.2: RNN Arquitecture from own Elaboration.

3.2 Vanilla LSTM Recurrent Neural Networks

LSTM networks, a specialized type of RNNs, are adept at handling long-term dependencies in sequential data due to their distinctive architectural design. The structure of an LSTM unit, as illustrated in Figure 3.3, includes various gates (the input gate, forget gate, output gate, and a cell state), each playing important roles in the information processing. The forget gate, governed by the equation:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{3.2}$$

determines which information should be discarded from the cell state. Here, f_t represents the forget gate's output, σ denotes the sigmoid function, W_f is the weight matrix associated with the forget gate, h_{t-1} is the previous hidden state, x_t is the current input, and b_f is the bias term for the forget gate. Similarly, the input gate, described by

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{3.3}$$

and the equation

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \tag{3.4}$$

decides what new information is added to the cell state. The variables i_t and $\tilde{C}_t$ represent the input gate's output and the candidate cell state, respectively, with W_i and W_C as their corresponding weight matrices, and b_i and b_C as biases. In addition, the cell state update is captured by

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t, \tag{3.5}$$

where C_t is the current cell state, and $*$ denotes element-wise multiplication. Finally, the output gate, another critical component of the LSTM unit, is governed by:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{3.6}$$

where o_t is the output gate's activation, W_o is the corresponding weight matrix, and b_o is the bias. This gate controls the extent to which the cell state influences the final output. It works together with the previously described gates to refine the LSTM's ability to make precise predictions. The LSTM's proficiency in managing sequential

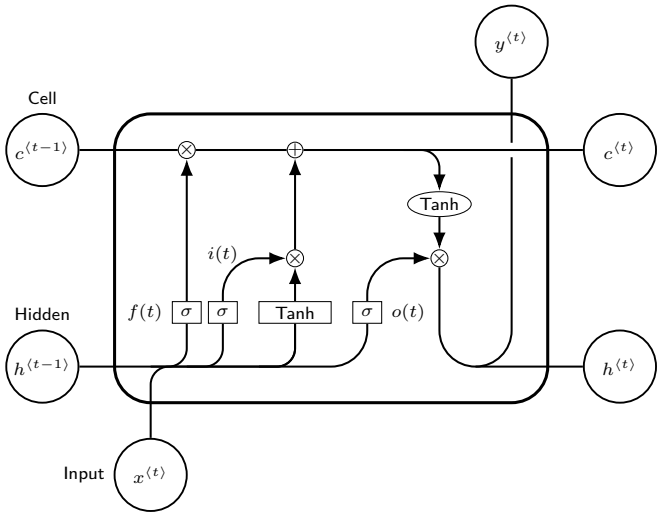

Figure 3.3: LSTM cell architecture from own elaboration

data, bolstered by these mechanisms, makes it suitable for tasks where understanding past context is key to predicting future events.

LSTM networks stand out due to their specialized architecture, which offers several advantages in processing sequential data (Staudemeyer & Morris, 2019). Firstly, their ability to effectively handle long-term dependencies allows them to retain crucial historical information over lengthy sequences, a task where traditional RNNs often fall short. This feature is particularly beneficial in applications like language modeling and time series forecasting, where past context significantly influences future predictions. Secondly, LSTMs mitigate the vanishing gradient problem, common in standard RNNs, through their complex gating mechanism. This ensures more stable and effective training over extended periods (Kelleher, 2019). Furthermore, LSTMs' unique gating system, including forget, input, and output gates, allows for more control over the flow of information. This results in a further ability to model complex patterns and relationships in data, leading to more accurate predictions.

While LSTM networks offer substantial benefits in handling sequential data, they are not without limitations, prompting the development of more advanced architectures such as Stacked LSTM and Bidirectional LSTM. One primary drawback of standard LSTMs is their computational complexity and time-consuming training process, which can be prohibitive, especially with large datasets. This complexity also translates into challenges in tuning hyperparameters and the need for substantial computational resources (Granata & Di Nunno, 2023). Additionally, while LSTMs are proficient at capturing long-term dependencies, they might not efficiently extract and use all available information in the input sequence, particularly in cases with very complex patterns or long sequences (Zhang et al., 2021). Stacked LSTMs address this by layering multiple LSTM layers to improve feature extraction capabilities, while Bidirectional LSTMs process data in both forward and backward directions, offering a more comprehensive understanding of context. These evolved architectures aim to refine the LSTM's predictive power and efficiency, tackling constraints in processing sequential data.

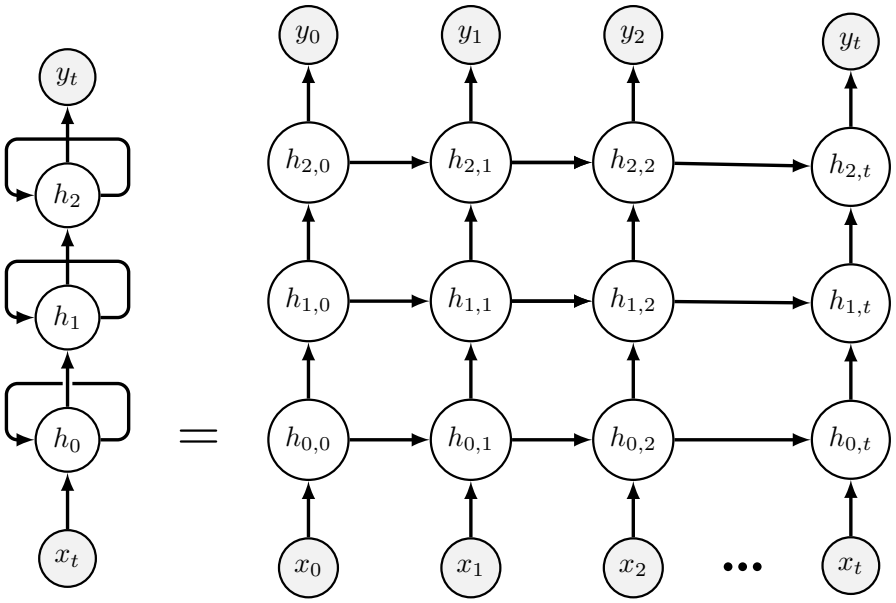

Figure 3.4: Stacked RNN architecture. Own Elaboration from (*How do I draw a simple recurrent neural network with Goodfellow's style? — tex.stackexchange.com*, n.d.)

3.3 Stacked LSTM Recurrent Neural Networks

Stacked LSTM networks, an advanced variant of the basic LSTM architecture, were developed to improve the model's ability to capture more complex patterns in sequential data. Unlike a simple LSTM that consists of a single layer of LSTM units, a Stacked LSTM incorporates multiple layers of LSTM units stacked one after the other (Y. Yu, Si, Hu, & Zhang, 2019). This structure allows the network to learn at various levels of abstraction, as information is processed through multiple layers, adding depth to the model's learning capability.

Figure 3.4 illustrates the difference between a simple LSTM architecture and a Stacked LSTM architecture. In the simple LSTM, each LSTM unit passes its output directly to the next time step in the sequence. On the other hand, the Stacked LSTM has several layers of LSTM units, where the output of one layer of LSTMs serves as the input to the next layer. This creates a hierarchy of layers where higher levels can learn to recognize more abstract features in the data sequence, potentially improving the network's predictive performance on complex time series forecasting tasks. However, these benefits come at the cost of increased computational complexity and the risk of overfitting due to the added layers, which may also complicate model tuning. Consequently, alternative architectures like Bidirectional LSTMs have been proposed to address these challenges by processing data in both forward and backward directions, aiming to enhance performance without excessively complicating the model structure.

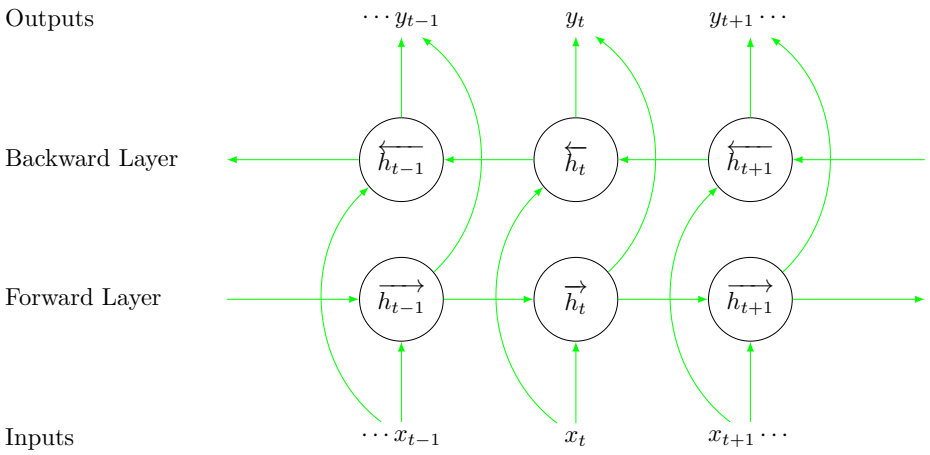

Figure 3.5: Bi-directional LSTM Arquitecture. Own Elaboration from (*How to draw BiLSTM neural network in latex? — tex.stackexchange.com*, n.d.)

3.4 Bidirectional LSTM Recurrent Neural Networks

Bidirectional Long Short-Term Memory (BiLSTM) networks are an advanced iteration of LSTMs designed to improve upon the ability to capture dependencies in sequence data (Schuster & Paliwal, 1997). BiLSTMs process data in both forward and backward directions, employing two separate hidden states to integrate both past (backward) and future (forward) information at each point in the sequence. Based on this consideration, Equations 3.7, 3.8 and 3.9 define forward and backward information flows, which directly impact the model's predictive performance. For the forward sequence, the hidden state at each time step t, denoted by $\overrightarrow{h_t}$ is updated based on the current input X_t and the previous hidden state $\overrightarrow{h_{t-1}}$:

$$\overrightarrow{h_t} = \sigma(W_{\overrightarrow{xh}} X_t + W_{\overrightarrow{h}\,\overrightarrow{h}} \overrightarrow{h_{t-1}} + b_{\overrightarrow{h}}). \tag{3.7}$$

In the backward sequence, the hidden state $\overleftarrow{h_t}$ is similarly updated, but uses the future state $\overleftarrow{h_{t+1}}$ instead:

$$H_t = (W_{\overrightarrow{xh}} \overrightarrow{h} + W_{\overleftarrow{h}\,yh} \overrightarrow{h} + b_y) \tag{3.8}$$

The final output H_t at each time step is a combination of both forward and backward states, ensuring the model captures information from the entire sequence:

$$H_t = (W_{\overrightarrow{xh}} \overrightarrow{h} + W_{\overleftarrow{h}\,yh} \overrightarrow{h} + b_y) \tag{3.9}$$

This dual processing structure allows BiLSTMs to effectively model complex dependencies in time series data, and preserve information over a longer period than unidirectional LSTMs (Baldi, Brunak, Frasconi, Soda, & Pollastri, 1999; Siami-Namini, Tavakoli, & Namin, 2019). In summary, they can provide additional context to the model, which can be particularly beneficial for tasks where understanding the entire sequence is the focus, such as in language translation or time series forecasting. However, despite their advantages, the complexity of BiLSTMs can make them prone to overfitting, especially with smaller datasets. Therefore, while they can offer superior

performance in certain applications, it's essential to consider the trade-offs between performance gains and computational efficiency.

3.5 Convolutional LSTM Neural Networks

Convolutional Neural Networks (CNNs) represent an important architecture in DL, particularly good at processing data with a grid-like topology, such as images or multi-dimensional time series. Through the use of convolutional layers, CNNs apply various filters to input data, efficiently identifying spatial hierarchies and patterns such as edges, textures, and shapes. These filters, or kernels, slide across the input data to produce feature maps, highlighting important features without the need for manual feature extraction. The architecture of CNNs, characterized by alternating convolutional and pooling layers, progressively reduces the dimensionality of the data while retaining critical information. This layered approach allows CNNs to capture complex, high-level features in the data by building upon simpler, low-level features identified in earlier layers (Adler, 2017). The final stages of a CNN typically include one or more fully connected layers, which interpret the high-level features extracted by the convolutional layers to perform classification or regression tasks.

Combining CNNs with LSTM networks takes CNNs' spatial feature extraction for grid-like data and LSTMs' temporal dependency modeling in sequences. This approach is particularly effective for analyzing sequential data with inherent spatial properties, offering a comprehensive understanding of both spatial and temporal dimensions (Adler, n.d.). The process involves initial data preprocessing, followed by spatial feature extraction through convolutional and pooling layers as it can be seen in Figure 3.6. LSTM layers then process these features to capture temporal dynamics, with a softmax classifier employed for final classification tasks, demonstrating a robust method for complex sequence data analysis (Zhou, Sun, Liu, & Lau, 2015).

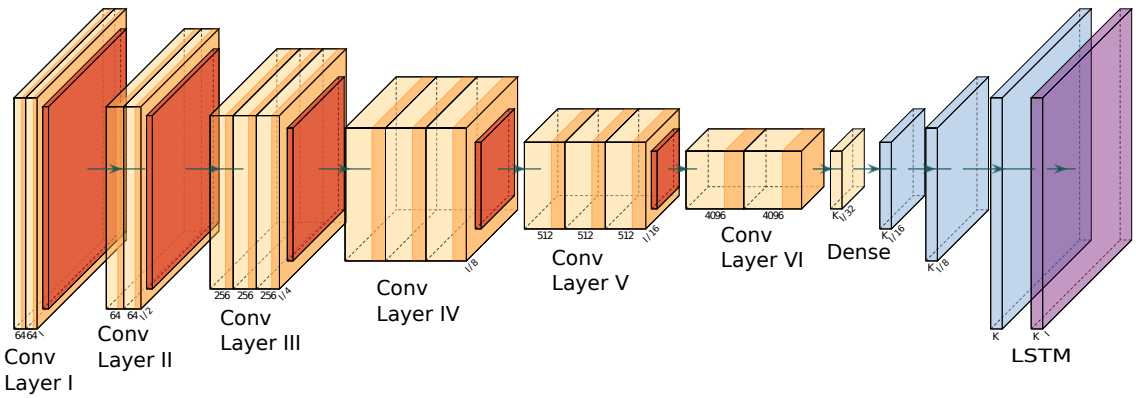

Figure 3.6: CNN LSTM Neural Network Architecture. Own Elaboration.

In extending the capabilities of CNNs for temporal data, convolutional LSTM (Conv LSTM) networks are introduced, which modify the traditional LSTM structure to handle high- dimensional spatial input. ConvLSTMs replace the matrix multiplications in the LSTM's state-to-state transitions with convolution operations, enabling the model to make use of spatial correlations effectively. As outlined by (Shi et al.,

2015), the ConvLSTM's forward update formula is adapted to include these convolutional operations, improving the network's ability to learn from both spatial and temporal dimensions. This is reflected in the modified update equations, such as the state update: $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$ described in Equation 3.5, where the standard multiplications are now convolutions, allowing for spatial feature maps to be integrated over time. The ConvLSTM architecture then becomes particularly adept for tasks that require an understanding of both spatial features and temporal sequences, such as video frame prediction or weather pattern forecasting.

In this context, Conv LSTM networks offer precision advantages by effectively capturing spatial-temporal patterns, a critical aspect for tasks such as video frame prediction and weather forecasting. They are particularly good at handling data where both spatial features and temporal progression are important. However, these networks are computationally intensive due to the combined complexity of convolutional and recurrent operations. This can lead to longer training times and the need for more computational resources (Wang et al., 2020).

3.6 Attention Mechanisms

The attention mechanism was originally introduced in the context of neural machine translation, with the primary objective of automating the translation of text across languages. In the conventional sequence-to-sequence model, input sentences are encoded into fixed-length vectors, which are subsequently decoded to generate output sentences. However, this fixed-length encoding can be problematic, particularly when dealing with longer inputs, potentially resulting in the loss of pertinent information and leading to inaccuracies in translations. To address this issue, the attention mechanism enables the decoder to selectively focus on different segments of the input sentence. During the decoding process, this mechanism computes a weighted sum of the input sentence encoding vectors, with the model autonomously learning the weights (Marulanda, Cifuentes, Bello, & Reneses, 2023).

In the context of time series forecasting, attention mechanisms have demonstrated efficacy in capturing intricate dependencies and patterns within data (Abbasimehr & Paki, 2022; Fu, Zhang, Yang, & Wang, 2022). Similar to machine translation, time series forecasting often involves managing lengthy input sequences. Here, attention mechanisms offer the potential to better model temporal dependencies by prioritizing the most relevant segments of the series throughout the prediction process. When integrating attention with LSTM, the model undergoes training on a time series of a certain length to capture the hidden state at each time step. The attention mechanism then uses these hidden states from previous steps $H = h_1, h_2, ..., h_{t-1}$ to calculate a context vector v_t that encapsulates relevant information for the current prediction (Bahdanau, Cho, & Bengio, 2014). This context vector is a weighted sum where the weights are learned during training, allowing the model to focus on the most influential factors at each time step. It is calculated through functions 3.10 3.11 by using a scoring function $f : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$, which evaluates the significance of the input vectors:

$$\alpha_i = \frac{\exp(f(h_i, h_t))}{\sum_{j=1}^{t-1} \exp(f(h_j, h_t))} \tag{3.10}$$

$$v_t = \sum_{i=1}^{t-1} \alpha_i h_i. \tag{3.11}$$

in which α_i is computed from the exponential nonlinear transformation. Increasing the parameter α will result in the model allocating a greater degree of attention. As the value of α increases, the model's focus intensifies on that part of the data.

Attention mechanisms within LSTM networks offer significant benefits for time series forecasting. These mechanisms enhance the network's ability to discern and prioritize the most impactful features within a time series, thus refining the prediction accuracy. By focusing on the relevant time steps and discarding extraneous information, attention-augmented LSTMs provide a nuanced understanding of demand patterns (Qiu, Wang, & Zhou, 2020). Such capabilities are instrumental for accurately forecasting product demand, which is often subject to complex temporal dynamics. Given these advantages, this final project will focus on leveraging LSTM networks with attention mechanisms to forecast product demand, aiming to harness their superior pattern recognition and predictive performance.

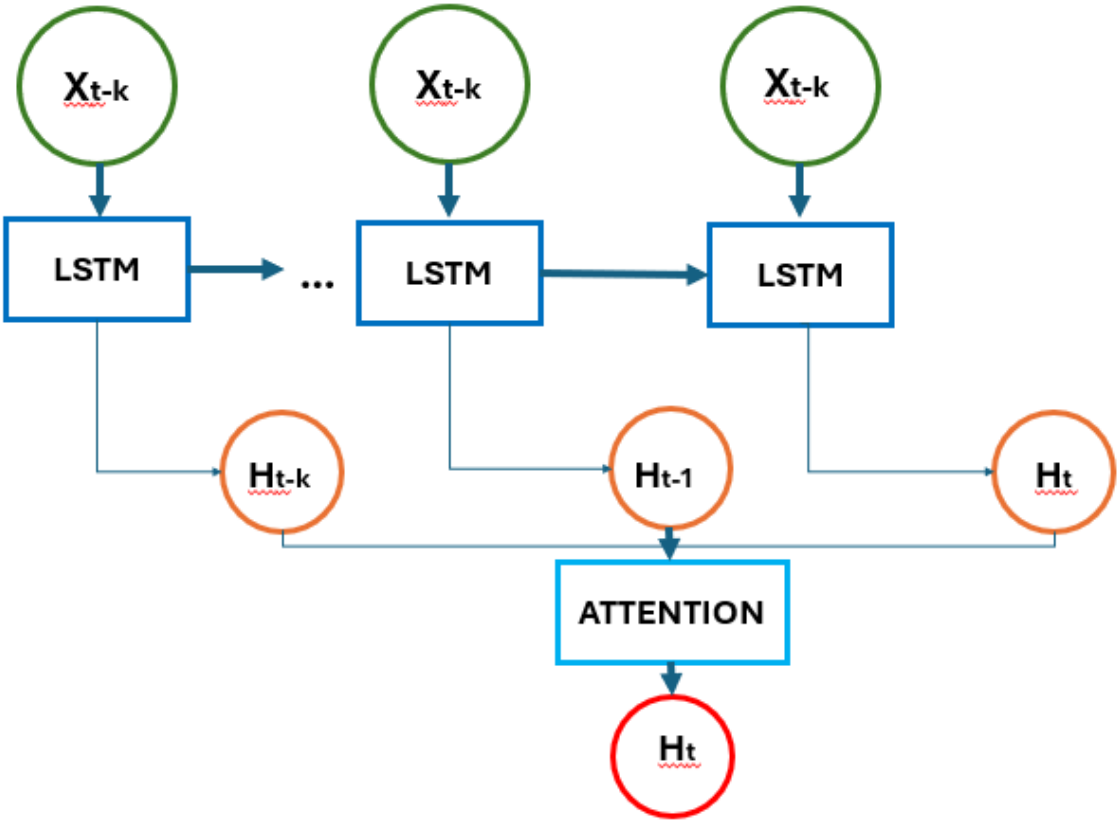

Figure 3.7: LSTM with Attention Mechanisms Architecture, own elaboration from Linkedin (2023)

3.7 Evaluation Metrics

In the field of predictive modeling, the selection and application of evaluation metrics are fundamental for objectively assessing the performance of algorithms. This section

is dedicated to discussing key measures that will gauge the accuracy and effectiveness of the predictive models under consideration. They provide essential insights into the strengths and limitations of the models, ensuring the optimal choice for specific forecasting scenarios. In time series forecasting, errors are calculated as the difference between the prediction and the real value (Botchkarev, 2018).

$$A_j - P_j, \tag{3.12}$$

where A_j represents the actual value observed at time j, while P_j denotes the prediction made by the proposed model for the same time point.

3.7.1 Root Mean Squared Error (RMSE)

The RMSE is a widely used metric in predictive modeling, particularly valued for its ability to quantify the magnitude of prediction errors. It calculates the square root of the average squared differences between the predicted values and the actual values (Botchkarev, 2018), as it can be seen in the following equation:

$$RMSE = \sqrt{\frac{\sum_{j=1}^{n} e_j^2}{n}} \tag{3.13}$$

The main concern with RMSE as a metric is its tendency to be influenced by outliers. This sensitivity is explained by the normal distribution, which underpins RMSE's application. In cases of pronounced model biases, it's often necessary to correct for systematic errors prior to computing RMSE(Chai & Draxler, 2014).

3.7.2 Mean Absolute Error (MAE)

The MAE is calculated as the sum of the absolute value of the errors divided by the number of observations:

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|e_j| \tag{3.14}$$

MAE offers computational simplicity and uniformity in evaluating model accuracy. It is particularly robust against outliers, minimizing their impact on the overall error metric. However, by assigning equal weight to all errors, MAE's simplicity might not always capture the nuanced performance of predictive models. Additionally, its linear nature could complicate optimization efforts, particularly in gradient descent scenarios, potentially leading to challenges in achieving the algorithm's minimum error rate (Jadon, Patil, & Jadon, 2022; Qi, Du, Siniscalchi, Ma, & Lee, 2020).

3.7.3 Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error (MAPE) is considered a relevant metric for evaluating the accuracy of prediction models by comparing experimental results to benchmarks in current state-of-the-art projects. By calculating the MAE divided by the number of observations (see Equation 3.15), MAPE offers a normalized measure,

enabling a direct comparison of model performance across different datasets. This passive approach ensures an objective assessment of a model's accuracy,facilitating the identification of improvements or advancements over existing models.

$$MAPE = \sqrt{\frac{\sum_{j=1}^{n} e_j^2}{n}} \tag{3.15}$$

This metric has several advantages for its error normalization across varying scales and for providing percentage-based error estimates. Nonetheless, it faces challenges because of the possibility of undefined values when predictions hit zero and its tendency to unevenly penalize discrepancies, potentially skewing precision assessment across prediction algorithms (de Myttenaere, Golden, Le Grand, & Rossi, 2016; Jadon et al., 2022)

Chapter 4

Time Series Forecasting of Product demand using Deep Learning

Our study will aim to use an LSTM with attention mechanisms to produce sales forecasts, something that has not been yet proposed to our knowledge. Our study will answer the predictability of this model as well as how our approach compares to the methodologies of other LSTMs: Vanilla LSTM, Stacked LSTM, Bidirectional LSTM, and Convolutional LSTM.

4.1 Methodology

Through this thesis, we will use a dataset (Martin, 2022) with product sales data from 2014 to 2017 divided into three categories: Furniture, Office Supplies and Technology. The thesis aims to produce a model that will better allow companies to produce sales forecasts. Coding used for the different models' implementation can be found under (Brizuela, 2024), which is mainly based on Ensafi (2020) , and *GitHub - AinhoaGallego/TGF: TFG predicción de acciones — github.com* (n.d.). Several steps were followed for the experimental part of this research. For the context of representation purposes, 4.1 sums up this part of the research.

1. **Acquisition and data pre-processing:** Sales data was compiled from a database of superstore (large supermarket) sample data. In this case, a database with provinces changed to Canadian provinces was obtained via a Tableau repository (Martin, 2022). They were then utilized in the Python file for further data preprocessing. This included separating data by category, setting index for original datasets, excluding irrelevant columns such as row id and order id, and separating data into months instead of daily data.

2. **Descriptive data analysis:** Data analysis techniques were conducted in order to understand the structure of our dataset. Distribution, seasonality, and evolution was analyzed to make a preliminary understanding of the shape of our data.

3. **Model implementation** Having performed a state-of-the-art analysis of the data, a series of models were selected to be executed in our environment. Our

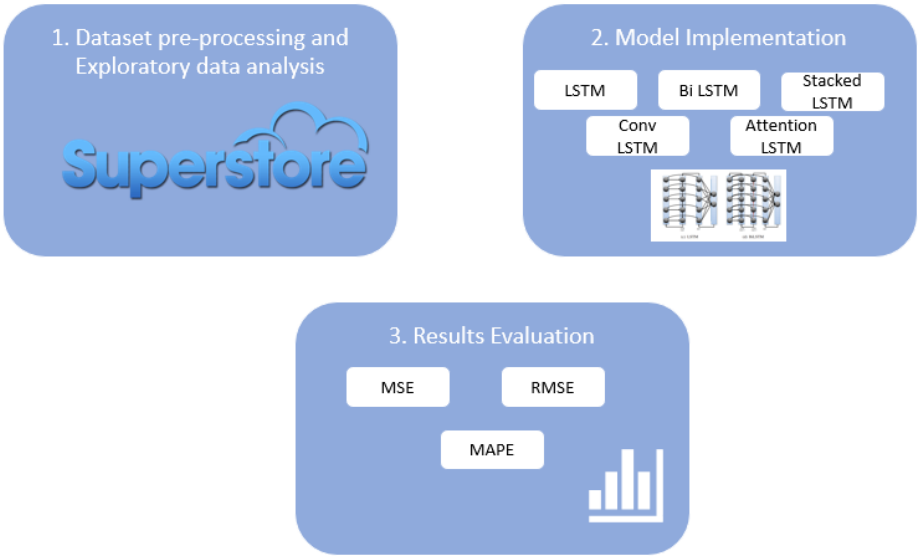

Figure 4.1: Methodological Approach used in this project

study will implement and compare five models: CNN , Vanilla LSTM, Stacked LSTM, Bidirectional LSTM, and LSTM with Attention Mechanisms. The code may be found under (Brizuela, 2024), the author's GitHub page, which is in turn based on (*GitHub - AinhoaGallego/TGF: TFG predicción de acciones — github.com*, n.d.) and (Ensafi, 2020) We will change the lookback and the number of neurons hyperparameters to find the optimal combination of hyperparameters.

4. **Evaluation and comparison of models** After the implementation of the models, we will compare and observe whether the LSTM with Attention Mechanisms ends up performing better, which is this research's initial thesis. Final model election will be chosen based upon the selected evaluation metrics.

4.2 Acquisition and data pre-processing

The dataset utilized in this study comprises Superstore Sales data spanning from 2014 to the conclusion of 2017, containing nearly 10,000 observations and featuring 21 variables. It covers sales information across three primary categories: furniture, technology, and office supplies. This study specifically focuses on examining furniture sales due to the presence of seasonal trends. This dataset, publicly available, offers essential elements for univariate forecasting, including sales figures and order dates for each data point. Additionally, it incorporates other variables such as Order ID, Order Date, Ship Date, Ship Mode, Customer ID, Customer Name, Segment, Country, City, State, Postal Code, Region, Category, Sub-Category, Product Name, Quantity, Discount, and Profit. A concise overview of the dataset pertinent to this research can be found in Figure 4.1.Ensafi (2020)

Through this table 4.2 we can observe a description of each of the columns in the processed dataset that has 9994 rows, each representing the sale of a product. Data

Postal Code	Sales	Quantity	Discount	Profit	Furniture	Technology
count	9994.0	9994.0	9994.0	9994.0	9994.0	9994.0
mean	55190.4	229.9	3.8	0.2	28.7	74.2
std	32063.7	623.2	2.2	0.2	234.3	272.4
min	1040.0	0.4	1.0	0.0	-6600.0	0.0
25%	23223.0	17.3	2.0	0.0	1.7	0.0
50%	56430.5	54.5	3.0	0.2	8.7	0.0
75%	90008.0	209.9	5.0	0.2	29.4	0.0
max	99301.0	22638.5	14.0	0.8	8400.0	4416.2

Table 4.1: Main Statistics of the Dataset

was later grouped into months so the predictions could be monthly, something more stable than daily predictions.

Value	**Description**	**Type of Value**
Ship mode	Defines the Mode of shipment.	Categorical
Segment	Defines the type of customer ordering.	Categorical
Country	Defines the country in which the sale is produced.	Categorical
Region	Defines where in the US the sale was produced.	Categorical
Category	Defines what category the product sold is in.	Categorical
Sub-category	Defines what sub-category the product sold is in.	Categorical
Quantity	Defines how many products were sold in that sale.	Discrete
Discount	Defines what discount percentage was applied to the sale.	Continuous
Average_volatility	Defines the main table statistics.	Continuous
Furniture	Defines furniture sales.	Continuous
Office supplies	Defines office supply sales.	Continuous
Technology	Defines technology product sales.	Continuous

Table 4.2: Description of Data Columns

4.3 Exploratory Data Analysis

Through this section, a series of data analysis techniques will be performed in order to achieve a better understanding of the dataset.

It's imperative to choose the variable with the highest seasonality to be able to capture said seasonality in our model. Capturing these repeated patterns throughout the years will surely be very valuable for businesses that need to adapt their inventories to said patterns. Within Figure 4.2, discernible fluctuations in sales are apparent across all three product categories. Notably, the Furniture category exhibits the most pronounced seasonality. It is due to this seasonality that Furniture has been identified as the subject of investigation for this thesis.

To extract the trend and seasonal components, the time series is decomposed using the STL (Seasonal-Trend decomposition using LOESS) decomposition approach. In Figure 4.3, we can observe a decomposition of the Furniture Sales data points in which

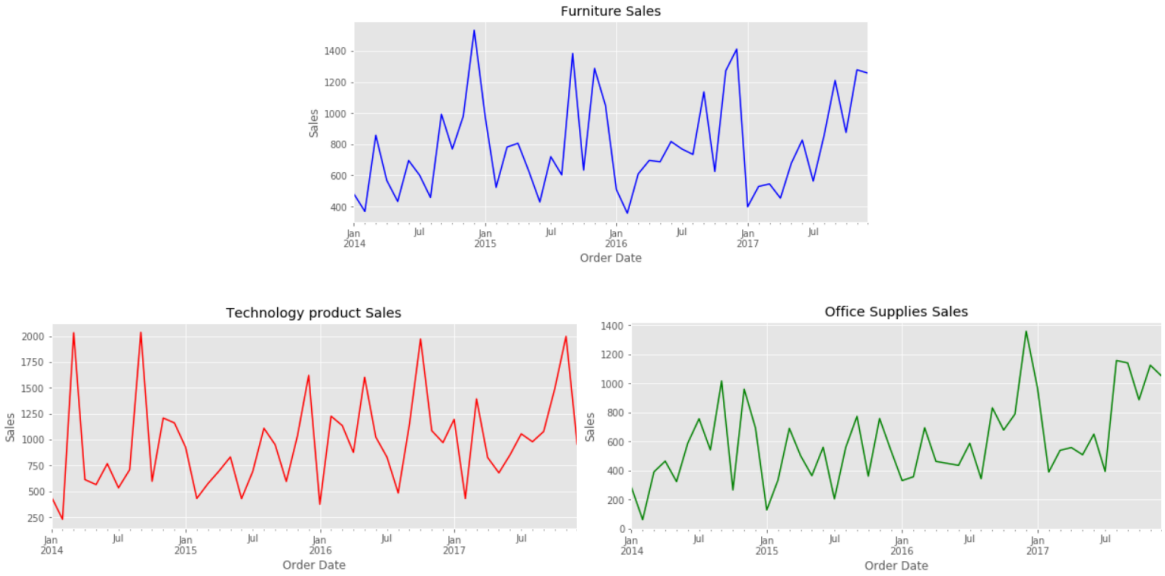

Figure 4.2: Sales Evolution by Category

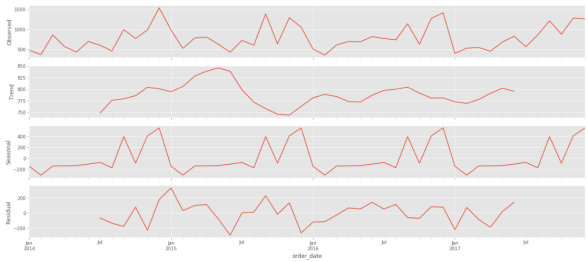

Figure 4.3: Decomposition of Time Series Using an Additive Model

seasonality is demonstrated. We can clearly observe a pattern over the analyzed years, with a spike in the end of the year, where more furniture is being sold.

We now proceed to make a deep analysis of the furniture sales distribution. In the table 4.3, we can observe the main statistics of Furniture. We can observe that 2,121 sales of the furniture category were completed between 2014 and 2017. We can also confirm the distribution findings made with the 75th percentile, as 75% of the sales totaled less than 436 dollars. We can as well see a very significant standard deviation, in which results close to the maximum sales value of 4416 dollars.

Statistic	**Value**
Count	2121
Minimum	1.9
25th Percentile	47.04
Average	349.67
75th Percentile	435.17
Maximum	4416.2
Standard Deviation	503.06

Table 4.3: Main Furniture Sales Statistics

4.4 Model Implementation

4.4.1 Train-Test Split

The dataset was divided into a train-test split of 60/40.

4.4.2 LSTM Model Hyperparameters

Details on the model's hyperparameters are now provided. These hyperparameters will be repeated for all the LSTM networks throughout this study.

The model starts compilation with the Adam optimizer employed as the stochastic gradient descent method. This optimizer offers various advantages, including ease of implementation, computational efficiency, minimal memory usage, invariance to diagonal gradient rescaling, and suitability for large-scale data and parameter problems Kingma and Ba (2014). Subsequently, the model employs Mean Squared Error (MSE) as the loss function, a widely favored choice due to its properties of making all biases positive and amplifying the influence of outliers. This characteristic renders it particularly appropriate for scenarios where observation noise follows a normal distribution (Ciampiconi, Elwood, Leonardi, Mohamed, & Rozza, n.d.). Both Adam and MSE are extensively applied in time-series forecasting using Deep Learning (Helmini, Jihan, Jayasinghe, & Perera, 2019; Terven, Cordova-Esparza, Ramirez-Pedraza, & Chavez-Urbiola, 2023). Additional key hyperparameters are informed by previous studies (*GitHub - AinhoaGallego/TGF: TFG predicción de acciones — github.com*, n.d.). Batch size, defined as the number of training samples used to update network parameters in a single iteration (Radiuk, 2017), was set to 16. Epochs, which denote the number of times the training set is iterated over during training (Afaq & Rao, 2020), were set to 500. Other hyperparameters specific to each model include using the tanh activation function for LSTM with attention mechanisms and using a pool size equal to 1 and a kernel size equal to 1 for the Convolutional LSTM model.

During the training process, neural network algorithms initialize random weights, resulting in different outcomes even when the same network is trained on identical data. To achieve stable results, models were run five times, and the performance metrics were averaged for the final model results (Ensafi et al., 2022).

4.5 Evaluation of results

In the previous section, a comprehensive overview of various model architectures was provided. Based on these architectures, the ensuing section will analyze the results to identify the most accurate model.

Multiple metrics will be used to evaluate what the optimal model is: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). Each metric provides insights into the type of errors incurred by the methods. To sum up the latter parts of this paper, both MSE, and RMSE are beneficial when prioritizing the spread of forecast values and penalizing larger discrepancies. However, interpreting MSE can sometimes pose challenges due to its squared error

values. Conversely, MAPE is valuable when comparing different forecast models or datasets, as it yields a percentage value.

In determining the best-performing model, the ideal candidate will exhibit low values for MSE, RMSE, and MAPE. MAPE is particularly useful for benchmarking this study against others employing diverse datasets. We have obtained all volatilities, we will be comparing through the MAPE volatility as it is a relative measure of volatility.

We will be changing the lookback and the number of neurons per layer. The lookback is the number of months at which the model is looking for predicting each moment(Lee, Lin, & Gran, 2021). The number of neurons is changed to obtain the best combination

4.5.1 Vanilla LSTM

As outlined in the theoretical background, the Vanilla LSTM comprises a solitary hidden layer of LSTM and one output layer. For this model, we can see a comparison between prediction and reality in performance metrics in table 4.4. Every single iteration was run 5 times to obtain standard deviation and the average of each pair of lookback and number of neurons. Highlighted in green is the combination of lookback and neurons that obtains the lowest RMSE performance metric. The best-performing hyperparameters for the Vanilla LSTM were a lookback of 1 month and 64 neurons per layer when ranking through the test RMSE metric. This combination of hyperparameters has also the lowest MAPE volatility, which is a indication of stability.

4.5.2 Stacked LSTM

Unlike the simple LSTM configuration, which features a solitary layer of LSTM units, a Stacked LSTM incorporates multiple layers of LSTM units stacked sequentially (Y. Yu et al., 2019). We can see performance metrics for this model in table 4.5. Every single iteration was run 5 times in order to obtain standard deviation and the average of each pair of lookback and number of neurons. Highlighted in green is the combination of lookback and neurons that obtains the lowest RMSE performance metric. It should be noted that when changing the Lookback, the Lookforward parameter was too changed in the same manner. The best performing hyperparameters for the Stacked LSTM were a lookback of 1 month and 32 neurons per layer when ranking through the test RMSE metric.

4.5.3 Bidirectional LSTM

Bidirectional Long Short-Term Memory (BiLSTM) networks represent an enhanced version of LSTMs crafted to enhance the capability of capturing dependencies in sequential data (Schuster & Paliwal, 1997). BiLSTMs handle data in both forward and backward directions, utilizing two distinct hidden states to amalgamate past (backward) and future (forward) information at every stage in the sequence. For this model, we can see performance metrics in table 4.6. Every single iteration was run 5 times in order to obtain the standard deviation and the average of each pair of lookback and number of neurons. Highlighted in green is the combination of lookback and neurons that obtains the lowest RMSE performance metric. It should be noted that when

changing the Lookback, the Lookforward parameter was too changed in the same manner. The best performing hyperparameters for the Bidirectional LSTM were a lookback of 1 month and 32 neurons per layer when ranking through the test RMSE metric.

4.5.4 Convolutional LSTM

The ConvLSTM neural network, stemming from LSTM (Hochreiter & Schmidhuber, 1997), is distinguished by its incorporation of a convolution operation within the LSTM cell. To capture spatial similarity, the ConvLSTM employs a fully-connected (FC-LSTM) extension, integrating convolutional structures in transitions both from input to state and from state to state (Chao, Pu, Yin, Han, & Chen, 2018; Nazir, Ab Aziz, Hosen, Aziz, & Murthy, 2021). For this model, we can see performance metrics in table 4.7. Every single iteration was run 5 times in order to obtain standard deviation and the average of each pair of lookback and number of neurons. Highlighted in green is the combination of lookback and neurons that obtains the lowest RMSE performance metric.It should be noted that when changing the lookback, the lookforward parameter was too changed in the same manner.The best performing hyperparameters for the Convolutional LSTM were a lookback of 2 month and 32 neurons per layer when ranking through the test RMSE metric.

4.5.5 LSTM with Attention Mechanisms

The LSTM with attention mechanisms essentially upgrades the Vanilla LSTM model by integrating an attention mechanism layer, as discussed in the theoretical background section. This attention mechanism allows each input element to selectively focus on other elements, establishing its own distinct context vector. Unlike RNNs, which must consider the entire context, the attention mechanism only needs to focus on the local context, leading to a more accurate context vector (Mehta, 2023). Significantly, attention mechanisms enable the distinction between past times, as applying uniform attention to all information can result in sub-optimal predictions (Pantiskas, Verstoep, & Bal, 2020). Performance metrics are presented in Table 4.8. The best performing hyperparameters for the LSTM with Attention mechanisms were a lookback of 1 month and 64 neurons per layer when ranking through the test RMSE metric. The prediction seems not so volatile when compared to other hyperparameter combinations.

4.5.6 Comparative of the Best Performing Models models

In figure 4.9 below, we can observe a performance comparison between the different deployed best-performing (looking at the Test Sample RMSE Test performance metric) models. In the table, we can see the best performing model is the Convolutional LSTM with Lookback=2 months and Neurons=32, as it obtained the lowest RMSE with the lowest Standard Deviation as well. The order of performance would be Convolutional LSTM, LSTM with Attention Mechanisms, Stacked LSTM, Bidirectional LSTM, and Vanilla LSTM. In order of RMSE volatility, we have (from least volatile to most volatile): Convolutional LSTM, LSTM with Attention Mechanisms, Vanilla LSTM, Bidirectional LSTM, and Stacked LSTM. The Stacked LSTM being the most volatile is probably due to the fact of implementing more than one LSTM in the model.

From this research, we can also get some insights into the most useful hyperparameters. We can tell that neither 3 months as a lookback nor 16 neurons ever became part of the most accurate models. 1 month as a lookback and either 32 or 64 neurons as the layer were the most common hyperparameter combinations in the most predictive models in this research.

Lookback	Neurons	Dataset Split	Measure	RMSE	MAPE	MAE
1	16	Train	Average	318.8	31.1%	239.0
1	16	Train	Std. Deviation	15.1	2.2%	3.1
1	16	Test	Average	343.9	30.4%	265.2
1	16	Test	Std. Deviation	27.4	0.3%	18.7
1	32	Train	Average	300.8	33.3%	235.0
1	32	Train	Std. Deviation	0.8	0.7%	1.0
1	32	Test	Average	313.4	30.1%	243.7
1	32	Test	Std. Deviation	4.0	0.2%	4.4
1	64	Train	Average	299.7	34.4%	236.3
1	64	Train	Std. Deviation	0.4	0.3%	0.4
1	64	Test	Average	307.7	30.3%	237.9
1	64	Test	Std. Deviation	1.3	0.1%	1.1
2	16	Train	Average	306.7	32.0%	236.1
2	16	Train	Std. Deviation	0.7	0.5%	0.7
2	16	Test	Average	328.1	32.0%	259.7
2	16	Test	Std. Deviation	2.7	0.2%	2.5
2	32	Train	Average	307.8	33.2%	239.0
2	32	Train	Std. Deviation	9.4	0.6%	9.5
2	32	Test	Average	312.4	32.9%	244.3
2	32	Test	Std. Deviation	7.1	0.4%	8.5
2	64	Train	Average	302.7	33.8%	233.4
2	64	Train	Std. Deviation	0.9	0.5%	0.4
2	64	Test	Average	316.8	32.9%	250.7
2	64	Test	Std. Deviation	2.1	0.3%	1.8
3	16	Train	Average	311.3	30.5%	180.2
3	16	Train	Std. Deviation	3.8	1.6%	90.0
3	16	Test	Average	338.9	33.4%	277.7
3	16	Test	Std. Deviation	9.0	0.7%	5.1
3	32	Train	Average	307.1	32.5%	229.7
3	32	Train	Std. Deviation	1.4	0.7%	1.3
3	32	Test	Average	326.3	34.5%	271.6
3	32	Test	Std. Deviation	2.9	0.4%	2.8
3	64	Train	Average	306.7	32.4%	228.4
3	64	Train	Std. Deviation	2.0	0.7%	2.1
3	64	Test	Average	323.0	34.7%	270.9
3	64	Test	Std. Deviation	4.5	0.2%	2.9

Table 4.4: Vanilla LSTM Performance Metrics

Lookback	Neurons	Dataset Split	Measure	RMSE	MAPE	MAE
1	16	Train	Average	300.0	35.6%	238.4
1	16	Train	St. Deviation	0.5	0.1%	0.3
1	16	Test	Average	306.2	30.5%	242.0
1	16	Test	St. Deviation	0.9	0.1%	0.7
1	32	Train	Average	298.8	35.5%	237.5
1	32	Train	St. Deviation	0.0	0.4%	1.0
1	32	Test	Average	304.3	30.5%	239.8
1	32	Test	St. Deviation	1.7	0.2%	0.8
1	64	Train	Average	298.4	35.0%	236.1
1	64	Train	St. Deviation	0.1	0.4%	0.8
1	64	Test	Average	305.2	30.3%	239.0
1	64	Test	St. Deviation	1.7	0.2%	1.0
2	16	Train	Average	304.3	34.7%	240.4
2	16	Train	St. Deviation	0.1	0.2%	0.6
2	16	Test	Average	312.8	33.3%	255.4
2	16	Test	St. Deviation	0.5	0.1%	0.4
2	32	Train	Average	304.5	34.9%	240.7
2	32	Train	St. Deviation	0.1	0.5%	1.3
2	32	Test	Average	312.7	33.4%	255.6
2	32	Test	St. Deviation	1.3	0.4%	0.2
2	64	Train	Average	304.7	35.0%	241.2
2	64	Train	St. Deviation	0.3	0.6%	1.7
2	64	Test	Average	312.3	33.4%	255.4
2	64	Test	St. Deviation	1.4	0.4%	0.2
3	16	Train	Average	308.8	34.2%	245.0
3	16	Train	St. Deviation	1.2	1.1%	3.9
3	16	Test	Average	325.3	37.5%	277.2
3	16	Test	St. Deviation	1.5	1.0%	2.7
3	32	Train	Average	308.8	34.1%	244.3
3	32	Train	St. Deviation	1.4	1.1%	3.6
3	32	Test	Average	325.6	37.4%	277.3
3	32	Test	St. Deviation	1.7	0.9%	2.7
3	64	Train	Average	302.4	34.1%	243.2
3	64	Train	St. Deviation	2.6	1.3%	4.2
3	64	Test	Average	319.0	35.9%	267.4
3	64	Test	St. Deviation	2.8	1.2%	3.5

Table 4.5: Performance Metrics for Convolutional LSTM

Lookback	Neurons	Dataset Split	Measure	RMSE	MAPE	MAE
1	16	Train	Average	301.9	33.0%	235.1
1	16	Train	St. Deviation	1.8	0.8%	0.9
1	16	Test	Average	315.7	30.1%	245.5
1	16	Test	St. Deviation	6.0	0.1%	6.0
1	32	Train	Average	299.5	34.5%	236.7
1	32	Train	St. Deviation	0.6	0.3%	1.0
1	32	Test	Average	307.0	30.3%	237.8
1	32	Test	St. Deviation	1.3	0.2%	1.0
1	64	Train	Average	300.0	34.4%	236.8
1	64	Train	St. Deviation	0.7	0.8%	2.0
1	64	Test	Average	308.2	30.4%	239.0
1	64	Test	St. Deviation	3.1	0.3%	2.8
2	16	Train	Average	308.2	35.1%	241.8
2	16	Train	St. Deviation	3.0	0.5%	2.5
2	16	Test	Average	308.2	32.9%	240.1
2	16	Test	St. Deviation	3.8	0.5%	2.3
2	32	Train	Average	309.3	35.3%	242.2
2	32	Train	St. Deviation	3.1	0.4%	2.1
2	32	Test	Average	309.5	33.2%	240.9
2	32	Test	St. Deviation	4.5	0.4%	1.8
2	64	Train	Average	308.9	35.8%	241.8
2	64	Train	St. Deviation	1.9	0.8%	1.6
2	64	Test	Average	307.8	33.8%	241.1
2	64	Test	St. Deviation	2.0	0.9%	1.8
3	16	Train	Average	300.2	31.4%	229.3
3	16	Train	St. Deviation	0.7	1.2%	3.4
3	16	Test	Average	319.1	38.5%	272.2
3	16	Test	St. Deviation	1.0	0.9%	1.3
3	32	Train	Average	301.3	31.2%	228.0
3	32	Train	St. Deviation	0.7	1.1%	2.1
3	32	Test	Average	321.0	38.7%	272.9
3	32	Test	St. Deviation	0.4	1.2%	2.1
3	64	Train	Average	305.1	33.2%	233.8
3	64	Train	St. Deviation	2.7	1.2%	4.6
3	64	Test	Average	324.1	40.8%	276.6
3	64	Test	St. Deviation	2.2	1.1%	2.0

Table 4.6: Performance Metric for Bidirectional LSTM

Lookback	Neurons	Dataset Split	Measure	RMSE	MAPE	MAE
1	16	Train	Average	298.9	35.3%	237.0
1	16	Train	Std. Deviation	0.3	0.1%	0.7
1	16	Test	Average	305.4	30.5%	240.4
1	16	Test	Std. Deviation	1.4	0.4%	1.5
1	32	Train	Average	298.4	35.2%	237.4
1	32	Train	Std. Deviation	0.0	0.2%	0.2
1	32	Test	Average	303.1	30.2%	236.8
1	32	Test	Std. Deviation	0.4	0.2%	0.6
1	64	Train	Average	298.5	35.0%	236.9
1	64	Train	Std. Deviation	0.1	0.2%	0.6
1	64	Test	Average	304.1	30.1%	236.6
1	64	Test	Std. Deviation	0.9	0.3%	0.9
2	16	Train	Average	301.1	34.0%	238.8
2	16	Train	Std. Deviation	1.9	1.2%	2.0
2	16	Test	Average	299.5	32.5%	242.9
2	16	Test	Std. Deviation	7.3	2.2%	12.5
2	32	Train	Average	301.2	34.4%	238.6
2	32	Train	Std. Deviation	0.2	0.1%	0.3
2	32	Test	Average	294.5	31.1%	233.8
2	32	Test	Std. Deviation	0.7	0.2%	0.6
2	64	Train	Average	297.4	34.2%	234.9
2	64	Train	Std. Deviation	0.1	0.1%	0.2
2	64	Test	Average	294.3	32.1%	235.8
2	64	Test	Std. Deviation	0.1	0.2%	0.2
3	16	Train	Average	296.3	31.5%	233.0
3	16	Train	Std. Deviation	1.9	0.6%	3.4
3	16	Test	Average	311.3	36.8%	265.5
3	16	Test	Std. Deviation	3.2	0.8%	3.6
3	32	Train	Average	294.4	31.2%	230.6
3	32	Train	Std. Deviation	0.4	0.6%	2.3
3	32	Test	Average	311.3	37.4%	266.0
3	32	Test	Std. Deviation	1.6	0.7%	1.6
3	64	Train	Average	292.2	30.9%	226.1
3	64	Train	Std. Deviation	0.3	0.3%	1.3
3	64	Test	Average	312.7	37.8%	265.9
3	64	Test	Std. Deviation	1.3	0.4%	1.5

Table 4.7: Performance Metric for Convolutional LSTM

Lookback	Neurons	Dataset Split	Measure	RMSE	MAPE	MAE
1	16	Train	Average	300.1	36.2%	240.1
1	16	Train	Std. Deviation	0.8	0.3%	1.1
1	16	Test	Average	304.4	30.6%	241.0
1	16	Test	Std. Deviation	1.1	0.2%	1.4
1	32	Train	Average	298.4	35.4%	236.6
1	32	Train	Std. Deviation	1.3	1.1%	3.8
1	32	Test	Average	305.6	31.3%	244.2
1	32	Test	Std. Deviation	3.8	1.2%	8.4
1	64	Train	Average	299.1	35.9%	238.6
1	64	Train	Std. Deviation	0.3	0.3%	0.8
1	64	Test	Average	303.5	30.6%	239.7
1	64	Test	Std. Deviation	0.9	0.1%	0.5
2	16	Train	Average	296.8	34.2%	235.5
2	16	Train	Std. Deviation	1.1	0.3%	1.4
2	16	Test	Average	310.1	32.2%	251.4
2	16	Test	Std. Deviation	2.4	0.2%	2.2
2	32	Train	Average	296.1	34.0%	234.4
2	32	Train	Std. Deviation	0.5	0.3%	1.4
2	32	Test	Average	309.9	32.3%	251.0
2	32	Test	Std. Deviation	1.0	0.2%	1.2
2	64	Train	Average	295.9	33.8%	233.7
2	64	Train	Std. Deviation	0.3	0.2%	1.0
2	64	Test	Average	310.0	32.3%	250.6
2	64	Test	Std. Deviation	0.8	0.1%	0.9
3	16	Train	Average	297.4	33.8%	233.8
3	16	Train	Std. Deviation	1.6	0.5%	3.3
3	16	Test	Average	315.9	33.6%	263.1
3	16	Test	Std. Deviation	2.3	0.1%	1.0
3	32	Train	Average	296.0	33.3%	229.7
3	32	Train	Std. Deviation	0.9	0.4%	1.7
3	32	Test	Average	313.8	33.5%	261.7
3	32	Test	Std. Deviation	1.6	0.2%	1.5
3	64	Train	Average	296.0	33.2%	229.5
3	64	Train	Std. Deviation	0.6	0.2%	0.9
3	64	Test	Average	313.4	33.5%	261.5
3	64	Test	Std. Deviation	1.4	0.2%	1.1

Table 4.8: Performance Metrics for Attention LSTM

Model	Lookback	Neurons	Dataset Split	Measure	RMSE	MAPE	MAE
Vanilla LSTM	1	64	Train	Average	299.7	34.4%	236.3
Vanilla LSTM	1	64	Train	Std. Deviation	0.4	0.3%	0.4
Vanilla LSTM	1	64	Test	Average	307.7	30.3%	237.9
Vanilla LSTM	1	64	Test	Std. Deviation	1.3	0.1%	1.1
Stacked LSTM	1	32	Train	Average	298.8	35.5%	237.5
Stacked LSTM	1	32	Train	Std. Deviation	0.0	0.4%	1.0
Stacked LSTM	1	32	Test	Average	304.3	30.5%	239.8
Stacked LSTM	1	32	Test	Std. Deviation	1.7	0.2%	0.8
Bidirectional LSTM	1	32	Train	Average	299.5	34.5%	236.7
Bidirectional LSTM	1	32	Train	Std. Deviation	0.6	0.3%	1.0
Bidirectional LSTM	1	32	Test	Average	307.0	30.3%	237.8
Bidirectional LSTM	1	32	Test	Std. Deviation	1.3	0.2%	1.0
Convolutional LSTM	2	32	Train	Average	301.2	34.4%	238.6
Convolutional LSTM	2	32	Train	Std. Deviation	0.2	0.1%	0.3
Convolutional LSTM	2	32	Test	Average	294.5	31.1%	233.8
Convolutional LSTM	2	32	Test	Std. Deviation	0.7	0.2%	0.6
LSTM with Attention	1	64	Train	Average	299.1	35.9%	238.6
LSTM with Attention	1	64	Train	Std. Deviation	0.3	0.3%	0.8
LSTM with Attention	1	64	Test	Average	303.5	30.6%	239.7
LSTM with Attention	1	64	Test	Std. Deviation	0.9	0.1%	0.5

Table 4.9: Best Performing Models

Chapter 5

Conclusions

This study evaluated the effectiveness of LSTM models in predicting future sales, recognizing the critical role of accurate demand forecasting for business operations. As highlighted in the introduction, leveraging historical trends enables businesses to drive their operations and planning effectively. Demand forecasting facilitates inventory management, mitigates the risk of overstocking, and supports informed strategic planning and resource allocation (Blum, 2020; Tadayonrad & Ndiaye, 2023). Particularly in dynamic industries like fashion, technology, and consumer goods, where competitive advantage hinges on precise demand forecasting, maintaining product availability and managing costs enhance customer satisfaction. Despite the evident benefits, prevailing industry practices often rely on intuition rather than data-driven models for sales forecasting, with only 40% of businesses currently implementing such initiatives (Rotenberg & Lindquist, 2013). However, in a rapidly evolving landscape, accurate prediction models present significant growth opportunities. Achieving high accuracy remains challenging due to inherent demand patterns within each product category, necessitating advanced models capable of capturing complex trends in data. The remarkable progress that deep learning has experienced in capturing intricate data trends suggests that advancements in technology can greatly benefit sales prediction and business operations.

Findings during the state-of-the-art section of this research project indicate that LSTM architectures with attention mechanisms demonstrate superior capability in adjusting to these trends. Recent studies have suggested that LSTM models with attention mechanisms are more effective for time-series forecasting compared to other LSTM variations (Hollis et al., 2018; Wen & Li, 2023). Based on these insights, it was hypothesized that integrating attention mechanisms into LSTM architectures could enhance sales forecasting accuracy. The proposed model was compared against commonly used LSTM variations, including Vanilla LSTM, Stacked LSTM, Bidirectional LSTM, and Convolutional LSTM to validate this hypothesis. These LSTM architectures were selected for their diverse settings and robustness, as demonstrated in previous studies (Ensafi et al., 2022; Padilla et al., 2021). The lookback and the number of neurons per layer hyperparameter were changed to find the optimal combination of these vital hyperparameters.

For this study, a sales dataset from the Tableau superstore was utilized Martin (2022), containing orders spanning the years 2014 to 2017 and categorized into Fur-

niture, Office Supplies, and Technology. Subsequently, preprocessing was conducted to ensure data quality, including verifying the absence of missing values and applying various transformations such as column renaming and grouping data into months. Furthermore, exploratory data analysis was carried out to gain a comprehensive understanding of the dataset structure. An examination of the dataset's distribution, through histograms and statistical analyses, revealed a left-skewed distribution. Additionally, a yearly sales trend analysis was performed for each category, highlighting furniture as exhibiting the highest degree of seasonality. A decomposition analysis of furniture sales time series was conducted to validate this observation, revealing distinct seasonal patterns within the data.

Our final results found interesting insights for this dataset, finding the Convolutional LSTM as the most effective of the implemented models, with the LSTM with Attention Mechanisms coming in second place when ranking the models by the RMSE metric. Our proposed method was found to have very comparable performance to other more traditional LSTM models used in sales forecasting such as the Vanilla LSTM. It was even found to obtain better performance than the Stacked LSTM. For these reasons, our proposed method yields good performance, not having been used for sales forecasting as for our knowledge.

As well, our final results yield insights as to what the most optimal hyperparameters were for using LSTMs. In 4 out of the 5 models, the most accurate combination of hyperparameters had a lookback of 1 month. In 3 out of the 5 models, the most accurate combination of hyperparameters used 32 neurons. Not one of the most effective combination of results used a lookback of 3 months or 16 neurons. This does not mean that these hyperparameters could be used, but this does give insights into the most useful hyperparameters for sales forecasting. Other hyperparameters such as the batch size can be explored, but our results yield insights into how these models should be used in the context of sales forecasting.

Regarding how the performance of our model compares to other past research projects, we should look at MAPE from page 10 of this research. When compared to traditional models such as the ones in (Demir & Akkaş, 2018), our model compares very favorably. When comparing to other studies that use deep learning such as (Murugesan et al., 2021), our performance is comparable to that research project. Perhaps the most comparable project to ours, (Ensafi et al., 2022) has models performing at a similar rate, with our project developing a more precise BiLSTM and them having a more accurate Vanilla LSTM. Having said that, we need to take into consideration that other research like (Ensafi et al., 2022) distinguishes holidays which is very meaningful for these models. They also perform the model 20 times per iteration and they do grid search hyperparameter optimization. This is very likely driving the differences between our performance and the performance of (Ensafi et al., 2022)

While the models employed can be adapted to different seasonal time series, it remains essential to fine-tune parameters and identify the most effective forecasting model for each different forecasting problem. Every dataset has its particularities so not one model fits all problems. Assessing these forecasting techniques across various seasonal datasets and comparing their performance could be the focus of future investigations. Moreover, experimenting with more intricate LSTM and CNN models could enhance results, exploring multivariate time-series forecasting is another poten-

tial direction, and developing hybrid models that blend classical and contemporary forecasting methods could offer valuable predictions.(Ensafi et al., 2022)

Declaración de Uso de Herramientas de Inteligencia Artificial Generativa en Trabajos Fin de Grado

ADVERTENCIA: Desde la Universidad consideramos que *ChatGPT* u otras herramientas similares son herramientas muy útiles en la vida académica, aunque su uso queda siempre bajo la responsabilidad del alumno, puesto que las respuestas que proporciona pueden no ser veraces. En este sentido, NO está permitido su uso en la elaboración del Trabajo fin de Grado para generar código porque estas herramientas no son fiables en esa tarea. Aunque el código funcione, no hay garantías de que metodológicamente sea correcto, y es altamente probable que no lo sea.

Por la presente, yo, Roberto Gozalo Brizuela, estudiante de E-2 Analytics de la Universidad Pontificia Comillas al presentar mi Trabajo Fin de Grado titulado "Product Demand Prediction using Neural Networks with Attention Mechanisms', declaro que he utilizado la herramienta de Inteligencia Artificial Generativa *ChatGPT* u otras similares de IAG de código sólo en el contexto de las actividades descritas a continuación:

1. Brainstorming de ideas de investigación: Utilizado para idear y esbozar posibles áreas de investigación.

2. Crítico: Para encontrar contra-argumentos a una tesis específica que pretendo defender.

3. Referencias: Usado conjuntamente con otras herramientas, como Science, para identificar referencias preliminares que luego he contrastado y validado.

4. Metodólogo: Para descubrir métodos aplicables a problemas específicos de investigación.

5. Interpretador de código: Para realizar análisis de datos preliminares.

6. Estudios multidisciplinares: Para comprender perspectivas de otras comunidades sobre temas de naturaleza multidisciplinar.

7. Constructor de plantillas: Para diseñar formatos específicos para secciones del trabajo.

8. Corrector de estilo literario y de lenguaje: Para mejorar la calidad lingüística y estilística del texto.

9. Generador previo de diagramas de flujo y contenido: Para esbozar diagramas iniciales.

10. Sintetizador y divulgador de libros complicados: Para resumir y comprender literatura compleja.

11. Generador de datos sintéticos de prueba: Para la creación de conjuntos de datos ficticios.

12. Generador de problemas de ejemplo: Para ilustrar conceptos y técnicas.

13. Revisor: Para recibir sugerencias sobre cómo mejorar y perfeccionar el trabajo con diferentes niveles de exigencia.

14. Generador de encuestas: Para diseñar cuestionarios preliminares.

15. Traductor: Para traducir textos de un lenguaje a otro.

Afirmo que toda la información y contenido presentados en este trabajo son producto de mi investigación y esfuerzo individual, excepto donde se ha indicado lo contrario y se han dado los créditos correspondientes (he incluido las referencias adecuadas en el TFG y he explicitado para que se ha usado *ChatGPT* u otras herramientas similares). Soy consciente de las implicaciones académicas y éticas de presentar un trabajo no original y acepto las consecuencias de cualquier violación a esta declaración.

Fecha: Junio 2024

Firma: Roberto Gozalo Brizuela

References

Abbasimehr, H., & Paki, R. (2022). Improving time series forecasting using lstm and attention models. *Journal of Ambient Intelligence and Humanized Computing*, 1–19.

Adler, T. (n.d.). *Convolutional LSTM for Next Frame Prediction.* `https://epub.jku.at/obvulihs/download/pdf/1825342?originalFilename=true`. ([Accessed 25-02-2024])

Adler, T. (2017). *Convolutional lstm for next frame prediction* (Unpublished master's thesis). Johannes Kepler University Linz, Linz, Austria.

Adya, M., & Collopy, F. (1998). How effective are neural networks at forecasting and prediction? a review and evaluation. *Journal of forecasting*, *17*(5-6), 481–495.

Afaq, S., & Rao, S. (2020). Significance of epochs on training a neural network. *Int. J. Sci. Technol. Res*, *9*(06), 485–488.

Ali, Y., & Nakti, S. (2023). Sales forecasting: A comparison of traditional and modern times-series forecasting models on sales data with seasonality. In *2023 10th international conference on computing for sustainable global development (indiacom)* (p. 159-163).

Almqvist, O. (2019). *A comparative study between algorithms for time series forecasting on customer prediction: An investigation into the performance of arima, rnn, lstm, tcn and hmm.*

Ansuj, A. P., Camargo, M., Radharamanan, R., & Petry, D. (1996). Sales forecasting using time series and neural networks. *Computers & Industrial Engineering*, *31*(1-2), 421–424.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baldi, P., Brunak, S., Frasconi, P., Soda, G., & Pollastri, G. (1999). Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, *15*(11), 937–946.

Bednárik, É., & Pakainé Kováts, J. (2010). Consumer behaviour model on the furniture market= vásárlói magatartásmodell a bútorpiacon. *Acta Silvatica Et Lignaria Hungarica*, *6*, 75–88.

Beheshti-Kashi, S., Karimi, H. R., Thoben, K.-D., Lütjen, M., & Teucke, M. (2015). A survey on retail sales forecasting and prediction in fashion markets. *Systems Science & Control Engineering*, *3*(1), 154–161.

Blum, K. (2020). *Gartner says less than 50% of sales leaders and sellers have high confidence in forecasting accuracy.* `https://www.gartner.com/en/newsroom/press-releases/2020-02-12-gartner-says-less-than-50--of-sales-leaders-and-selle`. ([Accessed 07-November-2023])

Botchkarev, A. (2018). Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. *arXiv preprint arXiv:1809.03006*.

Brizuela, R. G. (2024). *Github roberto gozalo brizuela.* `https://github.com/robertogozalo/Final-TFG-ADE`. ([Accessed 03-06-2024])

Chai, T., & Draxler, R. R. (2014). Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, *7*(3), 1247–1250.

Chang, O., Naranjo, I., Guerron, C., Criollo, D., Guerron, J., & Mosquera, G. (2017). A deep learning algorithm to forecast sales of pharmaceutical products. *no. August*.

Chao, Z., Pu, F., Yin, Y., Han, B., & Chen, X. (2018). Research on real-time local rainfall prediction based on mems sensors. *Journal of Sensors*, *2018*.

Ciampiconi, L., Elwood, A., Leonardi, M., Mohamed, A., & Rozza, A. (n.d.). A survey and taxonomy of loss functions in machine learning. arxiv 2023. *arXiv preprint arXiv:2301.05579*.

da Fonseca Marques, R. A. (2020). A comparison on statistical methods and long short term memory network forecasting the demand of fresh fish products.

Das, P., & Chaudhury, S. (2007). Prediction of retail sales of footwear using feedforward and recurrent neural networks. *Neural Computing and Applications*, *16*, 491–502.

de Carvalho Lima, J. E., Firmino, P. R. A., & Rocha, L. A. O. (2023). Demand forecasting, production planning, and control: A systematic literature review. *Engineering Design Applications V*, 377–399.

Demir, L., & Akkaş, S. (2018). A comparison of sales forecasting methods for a feed company: A case study. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, *24*(4), 705–712.

de Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F. (2016, June). Mean absolute percentage error for regression models. *Neurocomputing*, *192*, 3848. Retrieved from `http://dx.doi.org/10.1016/j.neucom.2015.12.114` doi: 10.1016/j.neucom.2015.12.114

Eddy, W. M., & Allman, M. (2000). *Advantages of parallel processing and the effects of communications time* (Tech. Rep.).

Elmasdotter, A., & Nyströmer, C. (2018). *A comparative study between lstm and arima for sales forecasting in retail.*

Ensafi, Y. (2020). *Neural network approach for seasonal items forecasting of a retail store.* `https://github.com/yasamanensafi/retail_store_sales_forecasting`. ([Accessed 29-02-2024])

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), 100058. Retrieved from `https://www.sciencedirect.com/science/article/pii/S2667096822000027` doi: https://doi.org/10.1016/j.jjimei.2022.100058

Fierro Torres, C. ., Castillo Pérez, V. H., & Torres Saucedo, C. I. (2022). Análisis comparativo de modelos tradicionales y modernos para pronóstico de la demanda: enfoques y características. *RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, *12*(24), e048. Retrieved from `https://doi.org/10`

.23913/ride.v12i24.1203 (Epub 25 de julio de 2022) doi: 10.23913/ride.v12i24.1203

Fleurke, S. (2017). Forecasting automobile sales using an ensemble of methods. *WSEAS Transactions on Systems, WSEAS*, *16*, 337–345.

Fu, E., Zhang, Y., Yang, F., & Wang, S. (2022). Temporal self-attention-based conv-lstm network for multivariate time series prediction. *Neurocomputing*, *501*, 162–173.

GitHub - AinhoaGallego/TGF: TFG predicción de acciones — github.com. (n.d.). https://github.com/AinhoaGallego/TGF/tree/main. ([Accessed 26-05-2024])

Granata, F., & Di Nunno, F. (2023). Neuroforecasting of daily streamflows in the uk for short-and medium-term horizons: A novel insight. *Journal of Hydrology*, *624*, 129888.

Green, K. C., & Armstrong, J. S. (2015). Simple versus complex forecasting: The evidence. *Journal of Business Research*, *68*(8), 1678-1685. Retrieved from https://www.sciencedirect.com/science/article/pii/S014829631500140X (Special Issue on Simple Versus Complex Forecasting) doi: https://doi.org/10.1016/j.jbusres.2015.03.026

Hasan, M. R., Kabir, M. A., Shuvro, R. A., & Das, P. (2022). *A comparative study on forecasting of retail sales.*

Haselbeck, F., Killinger, J., Menrad, K., Hannus, T., & Grimm, D. G. (2022). Machine learning outperforms classical forecasting on horticultural sales predictions. *Machine Learning with Applications*, *7*, 100239.

Helmini, S., Jihan, N., Jayasinghe, M., & Perera, S. (2019). Sales forecasting using multivariate long short term memory network models. *PeerJ PrePrints*, *7*, e27712v1.

Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, *37*(1), 388–427.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hollis, T., Viscardi, A., & Yi, S. E. (2018). A comparison of lstms and attention mechanisms for forecasting financial time series. *arXiv preprint arXiv:1812.07699*.

How do I draw a simple recurrent neural network with Goodfellow's style? — tex.stackexchange.com. (n.d.). https://tex.stackexchange.com/questions/494139/how-do-i-draw-a-simple-recurrent-neural-network-with-goodfellows-style. ([Accessed 26-05-2024])

How to draw BiLSTM neural network in latex? — tex.stackexchange.com. (n.d.). https://tex.stackexchange.com/questions/564305/how-to-draw-bilstm-neural-network-in-latex. ([Accessed 26-05-2024])

Jadon, A., Patil, A., & Jadon, S. (2022). A comprehensive survey of regression based loss functions for time series forecasting. *arXiv preprint arXiv:2211.02989*.

Jiang, H., Fan, Y., Sun, H., & Liu, S. (2021). Multi-algorithm fusion pharmaceutical sales forecasting mode. In *Icmlca 2021; 2nd international conference on machine learning and computer application* (pp. 1–5).

Kelleher, J. D. (2019). *Deep learning.* MIT press.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lakshmanan, B., Vivek Raja, P. S. N., & Kalathiappan, V. (2020). Sales demand forecasting using lstm network. In *Artificial intelligence and evolutionary computations in engineering systems* (pp. 125–132).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

Lee, M.-C., Lin, J.-C., & Gran, E. G. (2021). How far should we look back to achieve effective real-time time-series anomaly detection? In *Lecture notes in networks and systems* (p. 136148). Springer International Publishing. Retrieved from http://dx.doi.org/10.1007/978-3-030-75100-5_13 doi: 10.1007/978-3-030-75100-5_13

Li, Y., Yang, Y., Zhu, K., & Zhang, J. (2021). Clothing sale forecasting by a composite gru–prophet model with an attention mechanism. *IEEE Transactions on Industrial Informatics*, *17*(12), 8335–8344.

Linkedin. (2023). *Sales forecasting.* https://business.linkedin.com/sales-solutions/resources/sales-terms/sales-forcasting. ([Accessed 07-November-2023])

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

Livieris, I. E., Kiriakidou, N., Kanavos, A., Vonitsanos, G., & Tampakas, V. (2019). Employing constrained neural networks for forecasting new products sales increase. In *Artificial intelligence applications and innovations: Aiai 2019 ifip wg 12.5 international workshops: Mhdw and 5g-pine 2019, hersonissos, crete, greece, may 24–26, 2019, proceedings 15* (pp. 161–172).

Loureiro, A., Miguéis, V., & da Silva, L. F. (2018). Exploring the use of deep neural networks for sales forecasting in fashion retail. *Decision Support Systems*, *114* , 81-93. Retrieved from https://www.sciencedirect.com/science/article/pii/S0167923618301398 doi: https://doi.org/10.1016/j.dss.2018.08.010

Makridakis, S., Hyndman, R. J., & Petropoulos, F. (2020). Forecasting in social settings: The state of the art. *International Journal of Forecasting*, *36*(1), 15-28. Retrieved from https://www.sciencedirect.com/science/article/pii/S0169207019301876 (M4 Competition) doi: https://doi.org/10.1016/j.ijforecast.2019.05.011

Martin, S. (2022). *Tableau Community Forums — community.tableau.com.* https://community.tableau.com/s/question/0D54T00000CWeX8SAL/sample-superstore-sales-excelxls. ([Accessed 26-02-2024])

Marulanda, G., Cifuentes, J., Bello, A., & Reneses, J. (2023). A hybrid model based on lstm neural networks with attention mechanism for short-term wind power forecasting. *Wind Engineering*, 0309524X231191163.

Mehta, R. (2023). *momath.org.* https://momath.org/wp-content/uploads/2023/06/Rohan_Mehta_compressed.pdf. ([Accessed 04-03-2024])

Meng, Q., Catchpoole, D., Skillicom, D., & Kennedy, P. J. (2017). Relational autoencoder for feature extraction. In *2017 international joint conference on neural networks (ijcnn)* (pp. 364–371).

Murray, C., Du Bois, N., Hollywood, L., & Coyle, D. (2023). State-of-the-art deep

learning models are superior for time series forecasting and are applied optimally with iterative prediction methods. *SSRN Electronic Journal*. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4361707

Murugesan, R., Mishra, E., & Krishnan, A. H. (2021). Deep learning based models: Basic lstm, bi lstm, stacked lstm, cnn lstm and conv lstm to forecast agricultural commodities prices.

Nazir, S., Ab Aziz, A., Hosen, J., Aziz, N. A., & Murthy, G. R. (2021). Forecast energy consumption time-series dataset using multistep lstm models. In *Journal of physics: Conference series* (Vol. 1933, p. 012054).

Oliver-Muncharaz, J. (2020). Comparing classic time series models and the lstm recurrent neural network: An application to s&p 500 stocks. *Finance, Markets and Valuation*, *6*(2), 137–148.

Padilla, W. R., García, J., & Molina, J. M. (2021). Improving time series forecasting using information fusion in local agricultural markets. *Neurocomputing*, *452*, 355–373.

Paixão, K. W. M., & da Silva, A. M. (2019). Sales forecasting in a mechanical component manufacturer: comparison between monte carlo simulation and time series analysis. *Independent Journal of Management & Production*, *10*(4), 1324–1340.

Pantiskas, L., Verstoep, K., & Bal, H. (2020). Interpretable multivariate time series forecasting with temporal attention convolutional neural networks. In *2020 ieee symposium series on computational intelligence (ssci)* (pp. 1687–1694).

Pavlyshenko, B. M. (2019). Machine-learning models for sales time series forecasting. *Data*, *4*(1), 15.

Qi, J., Du, J., Siniscalchi, S. M., Ma, X., & Lee, C.-H. (2020). On mean absolute error for deep neural network based vector-to-vector regression. *IEEE Signal Processing Letters*, *27*, 1485–1489.

Qiu, J., Wang, B., & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PloS one*, *15*(1), e0227222.

Quevedo, F. C. S. (2020). *A comparison of machine learning and traditional demand forecasting methods* (Unpublished doctoral dissertation). Clemson University.

Radiuk, P. M. (2017). Impact of training set batch size on the performance of convolutional neural networks for diverse datasets.

Rafi, T., & Karim, R. (2020, 09). Time series analysis -a comparative analysis between ann and rnn. , *15*, 20-24.

Research, G. (2020). *Improve revenue forecast accuracy with emerging forms of sales forecasting technology.* https://www.gartner.com/en/documents/3983193. ([Accessed 07-November-2023])

Research, P. (2023). *Furniture market (by product: Beds, tables desks, sofa couch, chairs stools, cabinets shelves, others; by material: Metal, wood, plastic, glass, others; by application: Residential, commercial) - global industry analysis, size, share, growth, trends, regional outlook, and forecast 2023-2032.* https://www.precedenceresearch.com/furniture-market. ([Accessed 27-November-2023])

Rotenberg, & Lindquist. (2013). *Insight squared.* https://www.insightsquared.com/

wp-content/uploads/downloads/2013/05/Sales_Forecasting_Methods_eBook_v6_.pdf. ([Accessed 26-May-2024])

Sarker, I. H. (2021). Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, *2*(6), 420.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, *45*(11), 2673–2681.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, *28*.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A comparison of arima and lstm in forecasting time series. In *2018 17th ieee international conference on machine learning and applications (icmla)* (pp. 1394–1401).

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). The performance of lstm and bilstm in forecasting time series. In *2019 ieee international conference on big data (big data)* (pp. 3285–3292).

Staudemeyer, R. C., & Morris, E. R. (2019). *Understanding lstm – a tutorial into long short-term memory recurrent neural networks.*

Tadayonrad, Y., & Ndiaye, A. B. (2023). A new key performance indicator model for demand forecasting in inventory management considering supply chain reliability and seasonality. *Supply Chain Analytics*, *3*, 100026.

Terven, J., Cordova-Esparza, D. M., Ramirez-Pedraza, A., & Chavez-Urbiola, E. A. (2023). Loss functions and metrics in deep learning. a review. *arXiv preprint arXiv:2307.02694*.

Tkáč, M., & Verner, R. (2016). Artificial neural networks in business: Two decades of research. *Applied Soft Computing*, *38*, 788–804.

Wang, S., Cao, J., & Philip, S. Y. (2020). Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering*, *34*(8), 3681–3700.

Wen, X., & Li, W. (2023). Time series prediction based on lstm-attention-lstm model. *IEEE Access*.

Xu, X., Tang, L., & Rangan, V. (2017). Hitting your number or not? a robust intelligent sales forecast system. In *2017 ieee international conference on big data (big data)* (p. 3613-3622). doi: 10.1109/BigData.2017.8258355

Yasdi, R. (1999). Prediction of road traffic using a neural network approach. *Neural computing & applications*, *8*(2), 135–142.

Yu, Q., Wang, K., Strandhagen, J. O., & Wang, Y. (2018). Application of long short-term memory neural network to sales forecasting in retaila case study. In *Advanced manufacturing and automation vii 7* (pp. 11–17).

Yu, S., Dong, H., Chen, Y., He, Z., & Shi, X. (2019). Clothing sales forecast based on arima-bp neural network combination model. In *2019 ieee international conference on power, intelligent computing and systems (icpics)* (pp. 367–372).

Yu, W., Kim, I. Y., & Mechefske, C. (2021). Analysis of different rnn autoencoder variants for time series classification and machine prognostics. *Mechanical Systems and Signal Processing*, *149*, 107322.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, *31*(7), 1235–1270.

Zhang, B., Zou, G., Qin, D., Lu, Y., Jin, Y., & Wang, H. (2021). A novel encoder-decoder model based on read-first lstm for air pollutant prediction. *Science of the Total Environment*, *765*, 144507.

Zhao, L., et al. (2009). Neural networks in business time series forecasting: benefits and problems. *Review of Business Information Systems (RBIS)*, *13*(3).

Zhou, C., Sun, C., Liu, Z., & Lau, F. (2015). A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*.