2023

Application of Shallow Neural Networks to Retail Intermittent Demand Time Series

Urko Allende
Technological University Dublin, Ireland

Application of Shallow Neural Networks to Retail Intermittent Demand Time Series

Urko Allende

A dissertation submitted in partial fulfilment of the requirements of
Dublin Institute of Technology for the degree of
M.Sc. in Computing (Data Science)

02/03/2023

Declaration

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Science), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: Urko Allende

Date: 02/03/2023

Abstract

Accurate sales predictions are essential for businesses in the fast-moving consumer goods (FMCG) industry. However, their demand forecasts are often unreliable, leading to imprecisions that affect downstream decisions. This dissertation proposes using an artificial neural network to improve intermittent demand forecasting in the retail sector.

The research investigates the validity of using unprocessed historical information, eluding hand-crafted features, to learn patterns in intermittent demand data. The experiment tests a selection of shallow neural network architectures that can expedite the time-to-market in comparison to conventional demand forecasting methods. The results demonstrate that organisations that still rely on manual and direct forecasting methods could improve their predicting accuracy and establish a high-performing baseline for future development. The solution also offers an end-to-end systematic forecasting landscape enabling a lift-and-shift and easy transition from design to deployment. A practical implementation should bring about stable and reliable forecasts, resulting in cost savings, improved customer service, and increased profitability. Lastly, the research findings contribute to the broader academic field of forecasting and ML with a seminal proposal that provides insights and opportunities for future research.

Keywords: Demand Forecasting, Neural Networks, Intermittent Demand, Fast Moving Consumer Goods (FMCG), Recurrent Neural Networks (RNN), Convolutional Neural Network (CNN)

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Luca, for his unwavering support and guidance throughout my research. His insightful comments and valuable suggestions have greatly contributed to the success of this project. I would like to extend my gratitude to the TU Dublin staff who supported me throughout my master's program.

I dedicate this work to my beloved family, Alexandra, Alba, Sara and Mirela, whose endless love and support have been my inspiration and motivation. Without their encouragement and sacrifices, I would have not crossed the finish line.

Contents

List of Figures

List of Tables

List of Acronyms

SNN	Shallow Neural Network
ANN	Artificial Neural Network
DNN	Deep Neural Network
LSTM	Long-short Term Memory
RNN	Recurrent Neural Network
GRU	Gated Recurrent Unit
HPO	Hyperparameter Optimisation
SKU	Stock Keeping Unit
FMCG	Fast Moving Consumer Goods
ML	Machine Learning
GBDT	Gradient Boosted Decision Trees
LGBM	Light Gradient Boosting Machine

Chapter 1

Introduction

1.1 Background

1.1.1 Retail demand forecasting overview

According to R. J. Hyndman and Athanasopoulos (2018) demand forecasting is the process of estimating the quantity of a product or service that consumers will purchase over a specific time in the future. The estimates are based on historical data, market trends, and other relevant information and are typically used to help businesses plan production, manage inventory, set prices, and other significant decisions.

Demand forecasting can be divided into two broad categories: qualitative and quantitative methods. Qualitative methods rely on subjective information such as expert opinions, market research, and consumer surveys (Gilliland et al., 2016). Quantitative methods, on the other hand, use historical data and statistical models to make predictions about future demand. One of the most commonly used quantitative methods for demand forecasting is time series analysis. Time series analysis involves using historical data to identify patterns and trends in demand. These patterns and trends are then used to predict future demand. (Box et al., 2008)

Another widely accpted quantitative method for demand forecasting is causal analysis. Causal analysis attempts to identify the factors that drive demand and uses this information to predict future demand. This method can be further divided into two

categories: econometric and judgmental. Econometric methods use statistical models to identify the relationships between demand and relevant factors such as economic indicators, demographic data, and competitor information. Judgmental methods rely on expert opinions and intuition. This dissertation's focal point will be intermittent demand forecasting within the Fast Moving Consumer Goods (FMCG) industry (Makridakis et al., 2008). It will delve into the relationship between accurate forecasting and efficient business management.

1.1.2 The importance of demand forecasting

Demand forecasting plays a vital role in managing supply chains and is considered a fundamental aspect of broader business planning across the FMCG sphere. According to J. Armstrong (2001), demand forecasting allows organisations to anticipate future customer needs and make informed decisions concerning production, inventory management and product pricing. The ability to accurately identify and determine the necessary production of goods, in turn, allows for the efficient allocation of resources, including but not limited to capital and human investment, as well as the estimation of production costs. The primary objective of demand forecasting is to equip organisations with the information required to make sensible decisions in various areas of business management.

"Çakanyildirim and Roundy (2002) posit that in the semiconductor manufacturing industry, companies must engage in demand forecasting to inform strategic decision-making, such as new facilities, new technologies, the adjustment of capacity, the procurement of equipment, and the outsourcing of production. By analogy, this view can be broadly extrapolated to most manufacturing organisations. However, these organisations often face a forecasting accuracy constraint, resulting in reactive, abrupt and disruptive business decisions. This constraint is attributed to the unpredictability of demand and existing forecasting inaccuracies, which forces the upholding of safety stock. Corporations are impelled to develop flexible forecasting systems that respond quickly to demand changes to mitigate the *Bullwhip Effect's* adverse effects (Lee et al., 1997). Still, demand fluctuations caused by shortened product life cycles and ex-

panding product diversity make forecasting gradually more problematic. In practice, most companies are stuck with ancient predictive methods and not evolving with the times. A common practice that illustrates outdated practices is estimating demand by combining regional sales inputs from various customers and adjusting them via their market knowledge and insights.

In light of the research, it becomes clear how reliable demand forecasting is fundamental for companies to appraise strategic decision-making and avoid small fluctuations from turning into large inefficiencies and expenses. The task, however, grows increasingly difficult as new complex dynamics emerge, introducing more substantial uncertainty, especially with the shortening of the product life cycle and escalating diversification. Thus, it is essential for these companies to develop flexible forecasting systems that allow them to respond quickly to changes in demand and mitigate the negative impacts of the *Bullwhip Effect*. Diverse forecasting methods have been applied in various areas, with most existing demand forecasting practitioners using statistical approaches (Hamilton, 2020). Be that as it may, these methods cannot effectively deal with adopting new products or inter-generational substitutions (Chien et al., 2010)

1.2 Research statement

The investigation undertaken in this dissertation is motivated by a real business challenge presented in the workplace. The organisation sits within the FMCG industry and operates two manufacturing plants that provide a substantial portion of the world's supply – an estimated 70% of the total. The demand forecast, which informs capacity and materials planning, is based on estimates from various other businesses - treated as customers - that consume the semi-finished goods produced.

1.2.1 Current challenges

The inaccuracies in historical demand forecasts induce a series of practical challenges that are arduous and costly to deal with. For example, over-forecasting causes un-

wanted stockpiling of raw materials, while under-forecasting forces procurement teams to expedite the shipment of materials. One of the added complexities is the significantly high number of unique SKUs that enter the production schedule, some of which contribute only a tiny percentage of the orders but have a substantial financial impact. Incorrect forecasts are primarily instigated by many SKUs exhibiting acute intermittent demand. Intermittent demand can occur due to two main factors: First, some goods are seasonal and only sold during specific periods. Second, an innovation pipeline of new products introduced to capture the market size and drive top-line growth can be of low demand or have a short life. An additional intricacy is inherent in the hierarchical structure of the data, whereby each SKU is categorised as a member of a product family, a category, and a region. Simultaneously, distinct SKUs can have overlapping raw material requirements, leading to competition for shared raw materials in the event of under-forecasting for one product.

Once demand is locked, weekly materials and capacity planning occur to determine the manufacturing schedule and the raw material requirements. Given the lead time, which varies by region, planning must materialise several weeks to months ahead of order placement, which takes place just a few weeks before an order is shipped to a customer. Currently, there is a significant discrepancy between the forecasted and actual demand, leading to ineffective and inefficient planning that propagates errors across production orders and creates backlogs. Backlogs are typically dealt with over time, negatively affecting production costs. One existing buffer mechanism is to hold more extensive stocks of materials; however, incurring large inventories is not desired due to the high overhead expenses and associated warehouse capacity issues, as well as the increased risk to the entire manufacturing execution process.

The existing IT solutions to the problem rely on straightforward rules-based processes operated on spreadsheets. Despite the vast sea of accurate historical data, the information is not leveraged according to modern standards and available forecasting know-how. Therefore, a lot of potential benefits are yet to be realised. It is worth clarifying that this is not an IT infrastructure constraint and that the organisation possesses the means and capabilities to explore a more sophisticated course of action.

Considering the state of currently used solutions, it is advisable to implement one that is easy and quick to deploy while also capable of addressing the data's hierarchical intricacies and intermittent nature.

1.2.2 Advancing a research proposal

A possible initial approach is to operationalise a shallow neural network (SNN) that capitalises on an architecture that interprets temporal and sequential data points. An SNN is an artificial neural network with a limited number of layers, typically only one or two hidden layers. SNNs are employed for simple tasks where the complexity of a deep neural network is not necessary. They are easy to train, fast to run, and suitable for solving problems where the relationship between inputs and outputs is relatively uncomplicated. They are often used for basic tasks such as linear regression or binary classification. Without domain-specific knowledge, an SNN can extract patterns and non-linear relationships from the data by simply processing a sequence of historical data points.

IT teams within FMCG organisations are typically narrow and resource-scarce, which enables them to support and advance technological strategies at a manageable and predictable cost. At the same time, these organisations rely on big projects outsourced to IT vendors when it is imperative to explore new frontiers and look for a competitive edge. The model proposed though, does away with the possibility of implementing more refined and sophisticated approaches that require larger teams with the time and budget to experiment and trial. Under this scenario, the implementation of an SNN is advantageous in several ways:

1. It is simple to deploy in practical terms, as the data already exists and requires minimal processing.

2. The abundance of historical data available enhances the robustness of the proposed solution, making it well-suited for the type of problems that neural networks can address.

3. Data extraction is also straightforward, with minimal effort and preprocessing

required, thus avoiding the resource-intensive and intimidating aspects of data engineering that, in some cases, account for up to 90% of the effort in a data science project (Osama, 2021).

4. The data modelling stage is also uncomplicated and not computationally intensive, as it involves extracting sequences of historical demand as they occurred, thereby eliminating the need for feature engineering and selection during both the machine learning (ML) modelling phase and model monitoring phase.

In summary, this problem is significant as the FMCG industry faces unique challenges in demand forecasting due to its products' irregular and sparse demand patterns while constrained by its IT operating model. This research aims to address these challenges by developing a new approach to demand forecasting that can effectively handle the complexities of the FMCG industry.

1.3 Research objectives

The following section details the general and specific objectives that, in the aggregate, define the scope and direction and provide a framework for evaluating the success of the research. The objectives will specify the necessary steps to find the key architectural choices for an effective neural network capable of handling intermittent demand forecasting.

1. **To conduct a literature review on the research statement.**

    1.1 To review key aspects and concepts on time series and retail intermittent demand forecasting.

    1.2 To survey commonly used forecasting methods for intermittent demand problems.

    1.3 To assess the differences and trade-offs between theoretical research and practical implementations of forecasting models.

1.4 To identify in the literature newer methods that offer improvements over commonly used ones.

1.5 To narrow down a workable sample of learning techniques.

1.6 To analyse the gaps in the literature review and identify potential areas for further research.

1.7 To state the research problem formulate the research question.

(a) To develop and articulate a motivation for a research experiment

(b) To state the alternate and null hypotheses for the research question.

2. **To conduct empirical primary research to address the research problem.**

2.1 To preprocess and reformat the original data to suit the requirements of a supervised ML task.

(a) To join the relevant data sets to create a primary data frame that contains the information and structure for training neural networks.

(b) To clean the data by preprocessing the columns in the data set and removing any unwanted characters from strings.

(c) To convert all numeric columns to the appropriate data type.

(d) To filter a subset of the most recent data and remove any leading zeros in the selected data to improve training efficiency.

(e) To input missing values using the *last observation carried forward* method.

(f) To convert categorical variables in the data set to a numerical format using the label encoding technique to facilitate training.

(g) To create input sequences (features) based on the dependent variable by shifting and transposing the dependent variable.

2.2 **To identify optimal hyperparameters for training shallow neural networks in RNN and CNN architectures.**

(a) To decide hyperparameters fixed at the outset or experimented with based on their potential impact on the error metric.

(c) To evaluate and assign values to hyperparameters that will remain fixed throughout the experiment.

(e) To perform a grid search for training each model with the hyperparameters that remain variable throughout the experiment.

(d) To conduct an empirical analysis to determine the most suitable values for the hyperparameters used in the experiment.

2.3 **To develop and train multiple RNN and CNN models to address the research statement.**

(a) To perform a grid search to train each model on various architectures with different numbers of training samples and lengths of input sequences.

(b) To use the trained models to make predictions on the test set and record the resulting values.

(b) To aggregate the results for each architecture and calculate the average error score.

3. **To evaluate the effectiveness of the research experiment in addressing the research statement.**

3.1 To statistically evaluate the stability of the models trained (i.e., lower error spread).

3.2 To statistically evaluate the impact of the modified hyperparameters on forecasting accuracy (i.e., lower error scores).

3.3 To statistically compare the effect of the architectures tested on forecasting accuracy.

3.4 To statistically evaluate the effect of the various architectural configurations on forecasting accuracy.

3.5 To answer the research question and determine whether to accept or reject the null hypothesis.

1.4 Evaluation methodologies

The proposed approach involves using quantitative methods to perform an empirical evaluation of the research. The primary method of analysis is a robust regression analysis to gauge the contribution of the independent variables to the error loss recorded. Additionally, the research will employ a combination of statistical techniques, mathematical models, and neural network modelling. Moreover, descriptive statistics will be employed to gather and analyse the experiment results, ensuring a systematic and objective review.

1.4.1 Robust regression analysis:

The results will be collected at the most granular level, which involves predicted and actual values of the target variable for all observations of the hold-out set. The scope of the analysis examines the relationship between the input sequence length and the error metric evaluations. Accordingly, the results will be averaged by architecture before being fit into a regression model, so the focus is centred around the architectures.

The Huber loss will be used as the loss function in the robust regression analysis. This loss function calculates the sum of the squared deviations for small residuals and the absolute deviations for larger residuals. The Huber loss balances the sensitivity to outliers with the precision of the regression estimates. A user-defined parameter determines the transition point between the squared deviations and the absolute deviations, often referred to as the "scale" or "threshold" parameter, denoted as "T" in the Huber T-norm. The Huber T-norm minimises the effect of outliers, making it a suitable loss function for robust regression (Huber, 1973).

Robust regression is more resistant to outliers than traditional least squares regression, making it suitable for data sets containing outliers, precisely the case for intermittent demand. It can also be used to estimate regression parameters when the underlying distribution of the errors is not Gaussian. However, robust regression can be more computationally demanding than traditional least squares regression and can produce less precise estimates, especially when there are many outliers in the

data. These concerns do not apply as the amount of data will be minimal, and the hyperparameter optimisation process will stabilise the spread of the error (Lawrence, 1989).

In conclusion, robust regression is a suitable approach for the proposed regression analysis. The techniques employed to gather and analyse experimental data will shed light on the research problem by providing a systematic and objective description and explanation of the experiment's outcome. Finally, the methods employed will help achieve the research project objectives and resolve the research hypothesis.

1.5 Document outline

The subsequent section presents an outline of the chapters in this dissertation and illustrates their purpose.

The second chapter offers a comprehensive overview of academic subjects relevant to the research. The areas under examination include time series modelling, intermittent demand, statistical and ML techniques, and analysing the most successful methods in forecasting competitions. The chapter concludes with an assessment of current limitations and gaps in the literature.

The third chapter delineates the design of the experiments conducted in this study. The data engineering process is comprehensively described, as well as the choices and assumptions made regarding the modeling. The chapter concludes with a broad examination of the technical and functional limitations of the experimental design and motivates further exploration of unanswered questions.

The fourth chapter assesses the experimental results and explores their significance. A side-by-side comparison of all architectures is performed, and the research hypothesis is evaluated in light of the experimental outcomes.

The fifth chapter summarizes the research project undertaken, draws conclusions from the experimental results, and evaluates the contribution of the research to the advancement of the field. Recommendations for further academic work are also provided.

Chapter 2

Review of existing literature

The first part of the literature review is divided into two sections. Firstly, a survey of the most common forecasting methods in the specific domain of intermittent and retail demand. Then, an appraisal of diverse data preparation, modelling and validation techniques typically employed to obtain better performance and more generalisable models. The second part will summarise and highlight the gaps and motivation for the research problem and question.

2.1 Introduction to time series

A time series is a set of data points recorded at equally spaced time points arranged in chronological order. The time points can be regularly spaced, such as hourly, daily, or monthly, representing the evolution of a variable over time. Time series data is often analysed to understand the underlying patterns and trends in the data and make forecasts about the variable's future values. Time series analysis uses statistical and mathematical methods to model the time series behaviour and extract meaningful insights from the data. It is commonly applied in finance, economics, and engineering (Tripathi et al., 2021).

Several properties of time series data can affect the accuracy of time series models and predictions (Shumway & Stoffer, 2017). These include:

- Stationarity: The assumption that the mean and variance of the time series are

constant over time.

- Seasonality: A repeating pattern within the data that occurs at regular intervals, such as annually or monthly.
- Trend: A systematic pattern in the data over time, such as a general increase or decrease.
- Autocorrelation: The dependence of a time series value on previous values.

Retail demand time series typically exhibit specific characteristics that make them unique from other types of time series data (Ord & Fildes, 2013). These include:

1. A seasonal pattern with fluctuations in demand at different times of the year, such as increased demand during holidays.
2. An underlying trend in retail demand, such as a general increase or decrease over time.
3. A cyclical pattern with regular fluctuations that occur over more extended periods than the seasonal pattern.
4. An unforeseen event or external factors that cause irregular fluctuations in demand, such as weather conditions and extreme events.
5. Intrinsic patterns such as sales promotions, discounts, and other marketing activities, as well as others related to a product lifecycle.
6. An influence by competition from other retailers and the availability of similar products.

When it comes to predictive modelling, time series present added complexity due to the sequence dependencies that exist among the dependent variable. Consequently, the challenge lies in developing an appropriate predictive model that effectively leverages sequence dependencies.

2.1.1 Intermittent demand forecasting

Intermittent demand is a type of demand pattern in which the demand for a product or service is irregular, sporadic, and unpredictable. This type of demand poses significant challenges for inventory management and forecasting, as it is difficult to predict when demand will occur and in what quantities. Intermittent demand is defined by prolonged intervals of minimal or no demand followed by sporadic bursts of high demand. This form of demand is prevalent in industries such as fashion, electronics, and consumer goods. The investigation in this study focuses on intermittent demand forecasting and the methods to achieve the best results. When inaccurately executed, intermittent demand forecasting can significantly hinder businesses, as it poses unique challenges that traditional forecasting methods are not equipped to handle. Due to the sporadic and lumpy nature of the demand patterns, traditional forecasting methods, such as moving averages and exponential smoothing, are not suitable for forecasting intermittent demand (Syntetos & Boylan, 2006). These methods assume a stable and consistent demand pattern, which does not hold for intermittent demand. As a result, researchers have developed specialised methods such as Croston's and the Syntetos-Boylan (Croston, 1972).

Common challenges

One major challenge businesses face when forecasting intermittent demand is the lack of historical data. Traditional forecasting methods rely heavily on historical data to predict future demand. However, with intermittent demand, there may be long periods of no demand, making it challenging to gather enough data to make accurate predictions (Syntetos & Boylan, 2005). The absence of sufficient data leads to businesses overstocking or understocking their inventory, resulting in lost sales or excess inventory costs. Another challenge that businesses face when forecasting intermittent demand is the unpredictability of demand. According to C. Chen et al. (2017), with traditional forecasting methods, businesses can predict future demand based on past trends and patterns. However, intermittent demand makes it difficult to predict when the next burst of high demand will occur. This unpredictability can lead or-

ganisations to miss out on sales opportunities or be off guard when demand suddenly increases. Accurate demand forecasting allows businesses to make informed decisions about production, inventory management, and pricing. Businesses that can anticipate future demand and respond accordingly will be better armed to meet the needs of their customers and stay ahead of the competition. Furthermore, businesses unable to accurately forecast intermittent demand may fall behind their competitors who can do so (Willemain et al., 2004)

Accurately forecasting intermittent demand requires specialised methods and a deep understanding of the industry and products. Businesses that effectively forecast intermittent demand will be ready to meet customer needs and stay ahead of the competition.

2.1.2 Statistical methods

Croston method

Croston's method is a forecasting technique designed for intermittent time series data, which displays periods of zero demand interspersed with periods of non-zero demand. This type of data emerges in inventory and supply chain management, where demand may be sporadic and unpredictable. *Croston's method* leverages decomposing the time series into two processes: one for the non-zero demand and another for the inter-demand times (i.e., the time between non-zero demand events). The method then uses exponential smoothing techniques to forecast these two processes separately. The forecasting of the non-zero uses a two-parameter exponential smoothing technique, where one parameter predicts the mean demand and the other forecasts the frequency of demand. The forecasting of the inter-demand times obeys a Poisson process, where the forecasted inter-demand time is the reciprocal of the forecasted frequency of demand.

Finally, *Croston's method* uses a simple algorithm to calculate the final forecast by combining the forecasted non-zero demand and the forecasted inter-demand times, using the last observed demand and the last observed inter-demand time as inputs (Croston, 1972).

Peaks-over-threshold (POT)

Peaks-over-threshold (POT) is a statistical method that analyses extreme events in time series data. The method involves identifying peaks above a certain threshold and then analysing their characteristics, such as their frequency and distribution. This approach is commonly used in fields such as hydrology, meteorology, and finance to study extreme events such as floods, storms, and financial crises. POT methods estimate the probability of extreme events and model the data's underlying probability distributions (Coles, 2013).

In the context of intermittent demand forecasting, POT can be used to identify and analyse extreme demand events, typically characterised by significant spikes that occur infrequently. By identifying these extreme events, POT can provide insights into the underlying patterns and structures, which might improve forecasting models and estimate the likelihood of future extreme events (Embrechts, Klüppelberg, & Mikosch, 2013).

One approach is to use POT to model the underlying probability distributions of the extreme demand events to improve the performance of traditional models such as *Croston*. For example, this could involve incorporating POT-based estimates of the probability of extreme demand events into the forecasting process or using POT-based models to estimate the parameters of traditional intermittent demand models. It is important to note that POT is not a forecasting method by itself but rather a technique to analyse extreme events and gain insights into the underlying structures. This information can improve the performance of existing forecasting methods or develop new models better suited to the characteristics of the data (Syntetos & Boylan, 2005).

ADIDA

ADIDA is an acronym for "Auto-regressive Integrated Distributed lag Asymmetric". ADIDA model is a time series forecasting model used to analyse and forecast data with multiple seasonal and non-seasonal components and asymmetric effects (Spithourakis et al., 2014).

The ADIDA model combines several components, each of which captures a differ-

ent aspect of the time series. The "Auto-regressive" component captures the linear dependence between the current value of the time series and its past values. The "Integrated" captures the presence of non-stationarity in the data, such as trend or seasonality. The "Distributed lag" captures the impact of exogenous variables on the time series, such as economic indicators or external events. Finally, the "Asymmetric" captures the presence of asymmetry in the data, such as differing responses to positive and negative shocks. The ADIDA model is advantageous for modelling time series data that exhibit multiple seasonal and non-seasonal components and asymmetric effects. It is well suited to a wide range of applications, including economic, financial, and energy forecasting. The ADIDA model is complex, requiring a large amount of data and computational resources to estimate its parameters accurately and, therefore, only suitable for some data types, such as high-frequency data. ADIDA is not explicitly designed for intermittent demand, but more so a general model that applies to any time series data. It is possible to use the ADIDA for intermittent demand time series, however, it may not capture the specific characteristics of the data or provide as accurate forecasts as specialised methods. Additionally, it may require a large amount of data and computational resources to estimate its parameters accurately (K. Nikolopoulos et al., 2011).

Syntetos-Boylan Approximation (SBA)

Syntetos-Boylan Approximation (SBA) is a forecasting method explicitly designed for intermittent demand time series, grounded on decomposing the time series into two processes: othe non-zero demand and the inter-demand arrival times (i.e., the time between non-zero demand events). The SBA method uses a simple heuristic algorithm to calculate the final forecast; it uses the last observed demand and the last observed inter-demand time as inputs and then estimates the average and the average inter-demand time. Then the forecasted demand is calculated as the average demand multiplied by the probability of a non-zero demand event, and the forecasted inter-demand time becomes the average inter-demand time multiplied by the probability of a zero demand event.

The SBA method is simple to implement, does not require prior knowledge of the underlying probability distributions of the data, and is suitable for data with small sample sizes. However, it does not account for the uncertainty of the data, and it assumes that the demand is independent of the inter-demand times, which might only be true in some situations (Syntetos & Boylan, 2005).

Bootstrap

Bootstrap methods are a family of statistical methods that operate on the idea of resampling the original data with replacement. These methods estimate the distribution of statistics, such as means and variances, and generate confidence intervals for forecasts.

In the context of intermittent demand forecasting, bootstrap methods estimate the distribution and generate prediction intervals for the forecasts. Obtaining prediction intervals can be done by resampling the original data with replacement and applying the forecasting method of choice to each resampled data set. The resulting forecasts can then determine the forecast distribution and inform the prediction intervals. Bootstrap methods are convenient when the sample size is small and traditional methods for estimating the distribution of statistics may not be applicable. However, they can be computationally intensive (Kourentzes & Petropoulos, 2016).

Temporal aggregation

In the context of time series forecasting, temporal aggregation refers to combining observations over a predefined interval, such as daily, weekly, or monthly data. This methodology is applied to the dependent and independent variables in order to enhance prediction accuracy. The primary objective of temporal aggregation is to reduce the noise and volatility in the data, making it more stable and easier to forecast. The utilisation of temporal aggregation is particularly beneficial when the original data has a high frequency, such as hourly or minute-by-minute. By aggregating the data to a lower frequency, patterns become more discernible, thereby facilitating the forecasting process. In the context of time series modelling, temporal aggregation generates new

variables that capture diverse aspects of the data, such as the weekly average temperature or the monthly demand. These new variables become inputs in a forecasting model in conjunction with other exogenous variables. Temporal aggregation may introduce bias into the data. Therefore, it is crucial to select an appropriate temporal interval based on the specific characteristics of the data and the forecasting problem at hand (K. I. Nikolopoulos & Thomakos, 2019).

In the context of the research problem examined in this dissertation, using temporal aggregation to perform accumulated forecasts can enhance the accuracy of the performance, as it eliminates the challenge of determining the arrival of demand while making the signal constant over time. Furthermore, temporal aggregation can also result in losing some information in the original data, which may be necessary for the forecasting model. As a result, it is essential to weigh the benefits and drawbacks before implementing it. It is also important to leverage in conjunction with the original problem, which means that temporal aggregation can be the input of a separate method.

Limitations of statistical methods

Traditionally, forecasting techniques such as moving averages, the Croston approach Croston (1972) or related methods are utilised along with domain knowledge to predict demand in the industry. However, with the rising volatility in the supply chains and the decreasing lifespan of components, semi-finished goods and raw materials, conventional methods and human judgments must be challenged. These constraints emphasise that decision-making processes and systems must be equipped for Industry 4.0 (Dassisti et al., 2019). The limitations of traditional statistical methods are even more pronounced when compared to newer approaches such as ML and artificial neural networks. In recent years, ML approaches have gained dominance due to their impressive development and increased access to computational power. The following section offers a practical evaluation of statistical methods presenting a better appraisal of their strengths and limitations.

2.1.3 Trade-offs between theoretical and practical forecasting

Lack of research

Very few forecasting methods have been created specifically for intermittent data over the last 50 years, including the first and defining method by Croston in 1972 and later ones by Syntetos and Boylan in 2001, Willemain, Smart, and Schwarz in 2004, Teunter, Syntetos, and Babai in 2011 (K. Nikolopoulos et al., 2011). These methods were primarily designed and tested for forecasting spare part demand and managing inventory but have yet to be considered for outside operations research and inventory management applications. Nikolopoulos argues that there is a need for more practitioners and academics who use and recognise the potential of intermittent demand forecasting methods beyond the narrow application of spare parts and inventory management. He describes how despite the focus on modelling fast-moving time series and using causal models when information is available in the forecasting literature, there needs to be more focus on intermittent demand series and associated forecasting methods. In Nikolopoulos' opinion, the lack of research is mainly due to the widespread belief that these methods are only relevant for spare parts demand. However, this is a significant oversight, as 60% of the inventory consists of spare parts with pertinent intermittent demand. According to Nikolopoulos, the challenge lies in the uncertainty of the volume and timing of the demand. Additionally, these can be costly parts, as evidenced in various papers. The lack of academic research on this topic is also partly due to the perception that it is twice as challenging compared to other forecasting problems, causing academics to avoid it early in their careers.

Bojer and Meldgaard acknowledges a significant interest in developing accurate and reliable forecasting methods, with many new proposed each year (2021). Forecasting competitions are held to evaluate these methods and are considered the standard by the forecasting community. The methods evaluated include both statistical (previously discussed) and newer, cutting-edge ones based on machine learning, and more particularly, artificial neural networks.

M competitions

The M competitions have been significant in the forecasting community because they focus on the empirical accuracy rather than theoretical models. They allow for open participation and facilitate fair comparisons of methods. In 2020, Hyndman reviewed the M4 competition which required participants to predict higher frequency data with prediction intervals, prioritise reproducibility, use established methods as benchmarks, and have a larger sample size of 100,000-time series. The findings of the competition showed that complex ML methods could outperform simple models and confirmed the benefits of cross-learning and ensembling. The competition raised the question of the generalisability of the findings and how to use them to improve forecasting practices. The conclusion was that no single method is best for all forecasting tasks and that selecting an appropriate method should be based on the specific use case.

Pardalos studied the history of forecasting competitions and found that they have significantly contributed to understanding forecasting methods and their performance in different circumstances (2020). In particular, the M competitions have evolved from being criticised to attracting many participants using extensive, high-frequency data and testing innovative approaches to enhance forecasting accuracy (K. Nikolopoulos, 2021). The results of the M competitions have revolutionised the forecasting field by introducing novel ideas and inspiring further research. Despite its benefits, it is essential to acknowledge that no competition is perfect, and they should always be considered ongoing developments. Makridakis et al. (2022) evaluated the design aspects of previous forecasting competitions, considering their limitations and practical concerns, and proposed principles to guide the design of future competitions. They emphasised the importance of continuous learning from implementing these competitions and suggested a multi-contest approach that involves varying forecasting challenges of different characteristics and practical significance.

Kaggle competitions

A similar set of forecasting competitions are those hosted in Kaggle. Bojer and Meldgaard also discuss the practical applications of the insights obtained and the limitations. Their review supports that ensembles of models and cross-learning techniques performed better than single and local models. External information such as hierarchy and predictive variables (holidays, events and promotions) provided benefits to the performance of models. The competitions were won by innovative applications of time series and statistical methods early on and by non-traditional forecasting methods like GBDT and neural networks as of late. The success of GBDT and neural networks leans on their ability to handle intermittency and access relevant external information. The complex ML methods used in the later competitions performed better than statistical ones but with added complexity and computational requirements. The authors also highlight some limitations of using Kaggle competitions. They mention that the lack of access to the test set after the competition has ended limits the ability to test different solutions and evaluate performance using alternative error measures. The competition also needed to address prediction uncertainty, which is crucial for decision-making based on forecasts and often forgotten (2021).

Additionally, Kaggle does not require contestants to share their solutions or code publicly, making it harder to learn from the competitions and reproduce results. The authors suggest that Kaggle should require contestants to submit their code or complete a small survey to facilitate learning. Despite these limitations, the authors believe much can still be learned by focusing on patterns that worked across the competitions and the relationship between their findings and the data set characteristics. In conclusion, based on the analysis of six recent Kaggle forecasting competitions, the authors consider that the forecasting community has much to learn from the Kaggle community. They found that global ensemble models outperformed single local models and that ML methods outperformed conventional time series and statistical methods in the four latest Kaggle competitions due to the utilization of external information. The authors recommend that the forecasting community learn from the ML strategies for time series forecasting and participate in their further development.

Bojer and Meldgaard (2021) share, in a section called *"Gradient boosted decision trees vs. neural networks"*, the top-performing methods in forecasting competitions. They appreciate differences in performance and advance that GBDT is particularly effective in modelling external information, while neural networks exploit the lack of useful exogenous variables and large data sets. This circumstance is the perfect pretext to leverage the properties of ANNs in future research, especially if substantial amounts of data are available. The following section will explore the application of artificial neural networks in greater depth.

2.1.4 Artificial Neural Networks for time series forecasting

Since their inception in 1943, Artificial Neural Networks (ANN) have been utilised to address a diverse range of problems, such as automated processing, object recognition, speech and handwriting recognition, and even real-time sign-language translation. Despite the assumption that deeper network architectures would produce better results than shallow ones, empirical tests with deep networks yielded comparable or even inferior results (Bianchini & Scarselli, 2014). ANNs that perform time series forecasting modelling can identify the interaction between the inter-demand interval and the demand size. Kourentzes indicates that neural networks possess a few intrinsic features that set them apart from other statistical methods utilised in time series forecasting—their non-parametric and assumption-free data (2013). Specifically, multilayer perceptrons have proved to be universal approximators and, in theory, capable of capturing underlying time series patterns and structures. G.-Q. Zhang et al. (1998) argue that these features offer a much more flexible tool to address time series problems and eliminate, for the most part, the need for human experts that prescribe to rigid model structures.

Recurrent Neural Networks

Concerning the specific structural design, recurrent neural networks (RNNs) are a learning architecture well suited for time series forecasting. Their appropriateness lies in their capability to handle sequential data, where a previous time step influences

the current one, a common characteristic of time series data. RNNs perform well in a wide range of time series forecasting tasks, including stock price prediction, energy consumption forecasting, and climate forecasting (J. Zhang & Man, 1998).

The Long Short-Term Memory (LSTM) architecture, a particular type of RNN, is effective and robust for modelling long-term dependencies in various studies (Hochreiter & Schmidhuber, 1997). LSTM networks are designed to handle the problem of vanishing gradients, which can occur in traditional RNNs when the data has a long temporal arrangement. LSTMs introduce a memory cell, which retains information for an extended time, allowing the network to capture long-term dependencies in the data (Pascanu et al., 2013). As a result, LSTMs are particularly well suited for time series forecasting tasks where long-term dependencies are necessary. LSTM bidirectional networks are an extension of LSTMs that process the data in both forward and backward directions. The bi-directional nature allows the network to capture dependencies in past and future time steps, which are helpful in speech and language tasks, as well as time series (Said et al., 2021). For example, in financial time series forecasting, information from past and future time steps may be crucial for accurately forecasting future trends (Siami-Namini et al., 2019). Gated Recurrent Units (GRUs) are another type of RNN designed to address the vanishing gradient problem in traditional RNNs. GRUs are computationally more efficient than LSTMs and perform well in time-series forecasting tasks, particularly in applications with limited computational resources (Rehmer & Kroll, 2020).

Convolutional Neural Networks

Convolutional neural networks (CNNs) have traditionally been used for image and video processing tasks, but they have also shown promising results in time series forecasting. CNNs can be used to learn the spatial correlations between different features in a time series, which is practical in identifying patterns in the data. CNNs are typically applied to the raw or transformed data in the time domain. One popular approach is to use one-dimensional (1D) CNNs for time series forecasting, where the input data is a 1D signal with time as the only dimension. The CNN can then learn

the signal's features and identify relevant patterns (Jin et al., 2020). The filters in the CNN can capture patterns at different scales and resolutions, which makes them suitable for abstracting different types of patterns in the time series (Xue et al., 2019). Another popular architecture is to use a combination of CNNs and RNNs for time series forecasting. In this approach, a CNN is used to extract features, and an RNN is used to model the temporal dependencies in the data. This approach has been shown to be effective in capturing both local and global patterns (Han et al., 2021).

In practical applications, a recent study describes a meta-heuristic approach to automatically evolve CNN-LSTM architectures for time series forecasting using real-world data from a local food shop. The evaluation of three architectures show that the proposed evolutionary approach outperformed the baseline solution. This approach is effective detecting fitting architectures of deep neural networks. Further work is needed to improve the method by including external data sets and using high-performance computing (Xue et al., 2019). The same authors suggest a prediction method based on CNN and Bi-LSTM networks with multidimensional variables. The CNN learns the horizontal relationships between the variables of multivariate raw data, while the Bi-LSTM extracts temporal relationships. The proposed model was tested on Beijing meteorological data, delivering high data accuracy for wind speed and temperature. It indicates that the model can effectively explore the features of multivariable non-stationary time series data (Jin et al., 2020).

Applications of ANN to intermittent demand forecasting

In all the literature, only one publication applies a combination of an RNN and LSTM model to forecast intermittent demand data using only the demand values as input (Kourentzes, 2013). The paper evaluates the RNN model on three real-world data sets against several other methods: Croston, exponential smoothing and a deep neural network (DNN). The author finds that the RNN outperforms the DNN on most data sets. The exciting characteristic of the model is that it only utilises sequences with past observations of the dependent variable as an input for the network, showcasing promising results with a relatively straightforward implementation.

Summary of Artificial Neural Networks

In conclusion, the above architectures, RNNs, LSTMs, LSTM bidirectional networks, GRUs and CNNs, can be utilised for time series forecasting. The choice of architecture has to be tested and evaluated on each specific problem; the distribution and nature of the data might require approaches that cannot be known in advance. In general and concerning architecture choices, recurrent neural networks (RNN) are prevalent for time series prediction. However, as the duration of the time series increases, it can become harder to train a model using conventional techniques, leading to less precise forecasts. Accuracy loss is often due to gradient disappearance during training (J. Zhang & Man, 1998). To address this, some researchers propose using Long Short-term Memory (LSTM) and Gated Recurrent Unit (GRU), which handle long-term dependencies better. Many sequence learning problems, such as machine translation, audio encoding and video editing, have successfully leveraged LSTM and GRU network architectures. To further improve performance, researchers have added attention mechanisms to the coding-decoding framework, which improves the selection of input sequences and encoding of semantics in long-term memory applications. Attention mechanisms have proven effective in many deep-learning tasks, including the ones mentioned above.

2.1.5 Data preparation, modelling and evaluation strategies

Described below are techniques considered best practices when dealing with time series forecasting. Applying these strategies will prevent overfitting and produce models capable of better generalisation accomplishing superior results.

Data preparation

Data scaling: Data normalisation plays a crucial role in the performance of artificial neural networks. Scaling the data can enable the model to learn more effectively by ensuring the input values are within a similar range. Neural networks, including LSTMs, often use gradient-based optimisation algorithms to adjust the model's parameters and

minimise the error (Jentzen, Kuckuck, Neufeld, & Von Wurstemberger, 2021). These algorithms use the change in error (gradients) to adjust the model's parameters and make predictions more accurate. However, the gradients can be significant for some input features and small for others if the input data is not scaled. The disparity of scales can make the optimisation process unstable and slow because the parameters update at different rates for each one of the features. By scaling the data, the input values and gradients remain on a similar scale, making the optimisation process more stable and faster. Also the optimisation process can converge to a better optimum solution. In general, the neural network model's objective function (error function) is a non-convex function, which means it can have multiple local minima. If the input data is scaled, the optimisation algorithm can avoid getting stuck in a poor local minimum, which leads to poor performance. Scaling the data can help the optimisation algorithm escape poor local minima and converge to a better global minimum.

Another viewpoint offered by Sugiyama and Kawanabe is that scaling the data can improve the model's generalisation performance by reducing the internal covariate shift, which means that the distribution of the inputs to a layer of a neural network changes during training. Covariate shifting can happen when the input features have different scales, which can cause the model to adjust to ranging scales during training and make it more difficult to generalise to new data. Scaling the data can reduce the internal covariate shift and improve the model's generalisation performance (2012).

Although Singh and Singh (2020) concluded that data normalisation helps improve performance over un-normalised data, no single normalisation method emerged as superior. It is necessary to examine these findings within the context of stock market returns and be cognizant of the small sample size that they present. More research is needed to determine whether this method applies to other domains or data sets. Through the lens of its mathematical properties, it is noteworthy that normalisation assumes a normal distribution of data, which does not match the properties of the data set employed in this experiment. In a separate paper by Panigrahi et al. (2013), the authors show how vector normalisation provided the best accuracy compared to other normalisation techniques such as median, decimal scaling, z-score or min-max.

Vector normalisation is a procedure that scales the data by dividing each value by the original series' root sum squared (RSS) value. This technique, also known as "root-mean-square normalisation" scales the data so that the final values have an average of zero and a standard deviation of one. It enables the comparison of time series with different units of measurement or scales, making them equivalent by adjusting their relative magnitudes. Accordingly, this technique is commonly used in time series forecasting produce more accurate forecasts by making the data stationary. The main drawback, however, is that each time series is independently scaled and becomes a more computationally intensive process. This method has yet to be replicated in other experiments with a more robust design and might owe its performance to a small sample size. Conversely, as demonstrated in research by T. Zhang et al., a more tested and widespread technique is the MinMaxScaler normalisation (2019). In an experiment the authors enhanced their model predictive accuracy, achieving results comparable to existing operational ones used in Chinese coal mines. Therefore, it is essential to carefully consider the normalisation technique when applying deep neural networks for time series forecasting, considering the specific characteristics of the data set and the problem at hand. In light of the evidence, data scaling should be considered an additional hyperparameter. The scaling method and the scaling range are decided before the training process begins, and are set by the data scientist based on prior knowledge and experimental results. Because the choice of method has an impact on the model's performance, these are considered hyperparameters rather than parameters learned by the model during training.

Missing values: Generating large amounts of data in various domains has highlighted the importance of extensive data analysis. Ensuring the collected data is trustworthy and valuable is crucial, as poor-quality data can lead to unreliable models. Unfortunately, missing values in the data set are a common and unavoidable issue that can result in ambiguity during analysis. This issue can occur in domains such as gene expression, traffic control, industrial informatics, image processing, and software project. Neglecting to address missing values can result in misleading outcomes, mak-

ing it necessary to improve data quality by effectively handling these missing values (Khan et al., 2022).

Modelling

Cross learning: Cross-learning is a time-series modelling technique that involves discovering and applying shared patterns across multiple time series. It amplifies the signal-to-noise ratio by searching in a cross-sectional fashion across multiple time series (global models). Interestingly, cross-learning is only possible using ML (including general regression techniques) and neural networks, unlike Croston - or other statistical methods, which are always uni-variate (local models). Research suggests that combining the strengths of a top local and global approach could have a comparative advantage (Semenoglou et al., 2021).

Makridakis et al. (2022) shows how historically, winning submissions utilised "cross-learning" from multiple series concurrently instead of sequentially. To that effect, the newer competition data sets are composed of highly-correlated, hierarchically-organised series that enable cross-learning techniques. Historically, global models that leverage cross-learning have resulted in better performance than methods trained on a series-by-series basis. It is fundamental to remember that cross-learning improves forecasting accuracy and uses just one model instead of multiple models. This approach reduces the computational cost and eliminates issues related to limited historical data (Semenoglou et al., 2021). In summary, to-date research, utilising "cross-learning" and exploiting all the information in the data set is desirable.

Cross-validation: Cross-validation is a statistical technique used to assess the performance of an ML model. It involves dividing a data set into multiple subsets and training the model on one subset while evaluating its performance on a different subset. This process is repeated multiple times with different subsets, and the average performance across all iterations is used to determine the model's overall accuracy. The goal of cross-validation is to prevent overfitting, which is when a model fits the training data too well but needs to generalise better to new, unseen data. Using ef-

fective cross-validation (CV) strategies is crucial for complex forecasting tasks. These strategies help to objectively measure the accuracy of predictions, prevent overfitting and reduce uncertainty (Tashman, 2000). Different CV strategies are at disposal; however, it is essential to consider several factors, such as the validation period, the size of the validation window, how validation windows are updated and the criteria for evaluating forecasting performance. The most common strategy among winners in intermittent demand competitions, is to choose the last four horizon windows of data available to assess forecasting performance. Empirical evidence suggests measuring and factoring the models' mean and standard deviation should be used to assess the model's stability. It is of the utmost importance to consider the entire distribution of forecasting errors and their tails when evaluating forecasting methods to ensure robustness and high accuracy. This exercise will inform whether or not ensembling models is a viable or recommended approach (Makridakis, Spiliotis, & Assimakopoulos, 2021a). However, all in all, research is in demand of a systematic approximation to researching cross-validation techniques, which at the moment are a crafty undertaking without much research on the topic.

Evaluation

Error metrics: There exists a myriad of error metrics employed for measuring the performance of time-series models. However, different metrics may possibly yield incongruent results, even when conducting the same research experiment (K. Nikolopoulos et al., 2011). In light of this fact, researchers have to be cautious when deciding how the error is to be evaluated, as performance ranking of the various methods can vary depending on the error score utilised (W. Chen & Shi, 2021). For this reason that as evidence amounts, researchers have narrowed the number of valid metrics down to a selective group (J. S. Armstrong & Collopy, 1992). There are several properties that error metrics should possess in order to be deemed as valuable. One, it must exhibit stable statistical properties, that is, contained spread statistics (e.g. standard deviation), coherent error evaluation (e.g. non-infinity values), resistance to outliers, normality and heteroscedasticity (Chai & Draxler, 2014). Two, they must display

transitive error measuring properties across different types of time-series, in other words, an error metric must possess real business realisation regardless of the domain application and the time-series type. Three, they should be easily interpretable so that models can be contrasted and insights be inferred into the realworld.

- **MSE benefits**

  Mean Squared Error (MSE) is a popular metric for evaluating the performance of time series models in the retail domain. The benefits of using MSE include its simplicity, versatility, and ability to penalise significant errors. In terms of simplicity, MSE is a straightforward metric that is easy to understand and calculate, making it a convenient option for evaluating time series errors. Its lack of sophistication is fundamental in retail, where quick and accurate evaluations are crucial for effective decision-making. In addition, MSE's design penalises substantial errors, which can be crucial in the retail domain, where large forecasting errors can have significant consequences. This property of MSE helps to ensure that models that rely on gradually minimising the error loss can effectively learn the most relevant patterns and ensure that significant errors are not overlooked (Hyndman, 2006).

- **MSE disadvantages**

  While Mean Squared Error (MSE) is widely used in evaluating the performance of time series models in the retail domain, it also has several limitations that should be considered. These include its inability to account for trends or seasonality, its failure to distinguish between over- and under-predictions, and its sensitivity to outliers. Failing to consider any underlying trends or seasonality in the time series can lead to misleading results in some cases. In the retail domain, where trends and seasonality can play a significant role in sales patterns, this can result in inaccurate evaluations. Additionally, MSE treats over and under-predictions equally, which may not be appropriate in some retail applications where one type of error is more damaging than the other. For example, in a retail setting where stock management is critical, an over-prediction may

be more damaging than an under-prediction, as it can result in excess inventory and higher costs. Finally, MSE can be sensitive to outliers, which can disproportionately impact the results in the retail domain, where sales can fluctuate greatly. Boosted variances can result in an overly optimistic evaluation of the performance of a time series model. They should be taken into account when choosing a metric to evaluate forecasting performance (Wallström & Segerstedt, 2010).

Hyndman believes that Mean Squared Error (MSE) is a preferred metric for evaluating forecast performance in certain situations. Specifically, if all series are on the same scale and the primary objective is to evaluate forecast performance, MSE can be preferred due to its simplicity, effectiveness, and ease of calculation. Hyndman highlights that MSE penalizes large deviations, which is particularly important in the case of erratic demand, where the inter-arrival and size of demand can be highly unpredictable. Also, MSE is non-computationally intensive, which is essential when time and resources are limited. This feature of MSE helps to ensure that significant errors are appropriately accounted for, making it an effective tool for evaluating forecast performance in such a scenario (2006).

In conclusion, Hyndman believes that MSE is a suitable metric for evaluating forecast performance, mainly when all series are on the same scale, and the main objective is to evaluate forecast performance. Given its simplicity, effectiveness, and ease of calculation, MSE is a strong contender for evaluating forecast performance in the case of erratic demand.

2.2 Summary of gaps and motivation

2.2.1 Gaps in the literature review

1. As highlighted by K. Nikolopoulos (2021), there is a need for more research on intermittent demand forecasting for other applications, as most of the research has focused on inventory management and spare parts demand. More ML and forecasting practitioners must appraise the transformational value of adding specific research to the existing body of work. The academic community must recognise the applicability of newer learning techniques, especially in today's constantly evolving and more dynamic markets.

2. While the unique characteristics of retail demand time series, such as seasonality and cyclical patterns, are a core part of time series modelling in statistical methods, ML ones rarely incorporate them. ML techniques could harness the upside - and well-established - of statistical modelling by absorbing and integrating their most advantageous and relevant contribution.

3. Modelling external factors on intermittent demand appears to significantly augment the data quality, leading to a forecasting improvement. These variables comprise elements such as weather conditions, market prices, sales and promotions and calendar and holiday events.

4. More research is required to validate the performance of forecasting methods on real-world data, especially for heterogeneous industries and applications. It is the case that most industries are now beginning a digital transformation journey and can avail of state-of-the-art implementations as of recently. As a result, the academic community can avail of a new platform to test their hypotheses and advance the research.

5. Intermittent demands with long lead times, in particular, deserves more academic attention. The rising volatility and decreasing lifespan of components, semi-finished goods, and raw materials have profound effects on the supply chains,

which can suffer to detriment of organisations' bottom line. The aforementioned becomes an even more crucial feature given the nature of industry 4.0 where supply chain infrastructures are powered by complex digital systems, with data travelling the end-to-end network of systems.

6. The generalisability of findings from forecasting competitions such as the M4 and Kaggle. While these competitions have shown that complex ML methods can outperform simple models, there is a need to understand how to apply these findings to real-world forecasting scenarios and evaluate alternative error measures beyond comparative performance rankings.

7. Attention mechanisms have shown promise in improving the selection of input sequences and encoding of semantics in long-term memory, but further research is needed to explore their potential in time series forecasting. Although recurrent neural networks, such as LSTMs and GRUs, are designed for sequential and temporal data and have time series forecasting capabilities, transformers have not yet proved reliable and accurate.

2.2.2 Motivation

A purview of the literature reveals that recurrent and convolutional (1D) neural networks have, in practice, demonstrated their versatility in modelling complex intermittent demand. One of their key features is the ability to abstract, without exogenous variables, patterns that lead to accurate forecasting. The trade-off, or rather mechanism, is the need for a large volume of data on which to train. There is a clear opportunity to utilise real retail data to train an ANN that is simple to set up and thus has the potential to generalise across other data sets. It is well understood that, in most cases, ANNs fare better with larger training samples. The remaining inquiry, however, is how condensed or expanded should the number of features of the data set be. Or, in other words, how does the length of demand data impact the performance of an ANN. There is evidence, in recent competition winners, that more than a couple of months might be detrimental. However, it is an open question that merits more

study (In, 2020). Additionally, the literature analysis has shed some light on the best practices for taking full advantage of ANNs and, as such, will be incorporated into the research experiment.

2.2.3 Research hypothesis

If a shallow RNNs or CNN (1 hidden layer) is trained using the M5 Walmart data set with a varying length of the input sequence - based only on past observations - then it is expected that the longer input sequence will exhibit a significantly lower mean squared error (MSE) than a shorter input sequence.

Null hypothesis (H0): Increasing the number of past observations and utilising an RNN and CNN in a shallow neural network implementation will not result in a significant improvement in the accuracy of models trained with unprocessed historical demand values for time series forecasting, as measured by the mean squared error (MSE) evaluation metric.

Alternate hypothesis (H1): Increasing the number of past observations and utilising an RNN and CNN in a shallow neural network implementation will result in a significant improvement in the accuracy of models trained with unprocessed historical demand values for time series forecasting, as measured by the mean squared error (MSE) evaluation metric.

Chapter 3

Experiment design and methodology

3.1 Research question

What is the impact of increasing the number of past observations as and input vector on the accuracy of shallow neural network (1 hidden layer) models using GRU, LSTM, RNN, and CNN architectures and trained with unprocessed historical demand values for intermittent time series forecasting?

3.2 Experiment design overview

The experiment will follow the CRISP-DM framework. CRISP-DM stands for Cross-Industry Standard Process for Data Mining and is a widely used methodology for data science projects. The design divides the experiment into six main stages: *1. Business Understanding, 2. Data Understanding, 3. Data Preparation, 4 . Modelling, 5. Evaluation, and 6 Deployment.* The diagram below represents a concise overview of the experiment process concerning stages 3,4, and 5 (Tripathi et al., 2021).

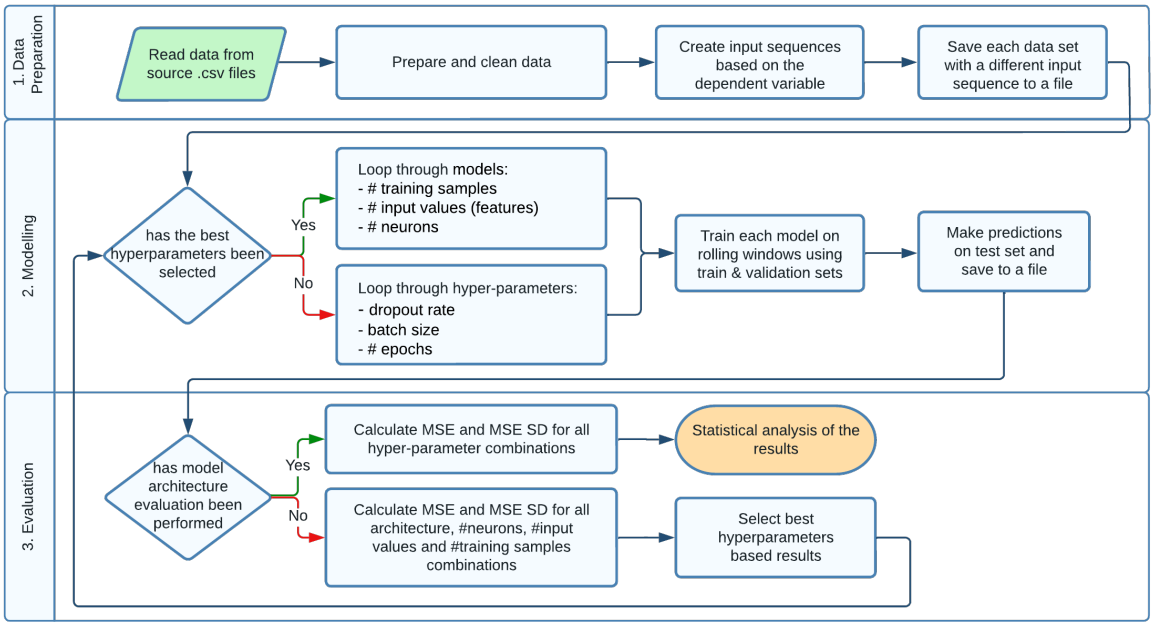

Figure 3.1: Experiment design flow diagram

The research problem section in this document exhaustively covers stage *1. Business Understanding*, therefore, will not be detailed here. At the same time, stage *6. Deployment* is not treated as it is not part of the dissertation scope. Some general ideas will be discussed nonetheless. At a high level, the experiment will answer the research question by moving through three phases. First transforming the provided time series into a supervised ML problem. Whereby the data will be transformed into an X data set consisting of a set of features, and Y is the target variable we aim to model. Then, conducting two grid searches and collecting the results associated. Finally, evaluating the results obtained with the use of statistical methods and techniques.

Steps:

1. **Read data from source .csv files** involves reading data stored in separate files, and the goal is to combine them into a single data set.

2. **Prepare and clean data** is related to preparing and cleaning the data. The original data frame is in a unique structure unsuitable for neural network training and must be transformed into a long format to be compatible with the network architecture.

3. **Create input sequences based on the dependent variable** requires the extraction of historical values that must be appended to the data frame as features of the dependent variable 'sales'.

4. **Save each data set with a different input sequence as a file.** The length of input sequences utilised is [56, 112, 365]. A file is saved for each input sequence for later retrieval during training and prediction.

5. **Hyperparameters (HPO) grid search.** The first grid search is comprised of three elements: the model batch size, the rate of dropout, and the number of epochs. The patience value is constant at 10 for early stopping.

6. **Architecture grid search.** The second grid search is comprised of four parameters. First, the model architecture. Second, the number of training days for each product-store combination, namely the training sample size. Third, the length of the input sequence of historical sales data, namely the number of features. Four, the number of neurons in the first layer.

7. **Train each model on a sliding window.** The number of training samples - a grid search parameter- determines the size of the rolling windows. Finally, each model loops through four equally-sized folds.

8. **Make predictions on the test set and save them to a results file.** After each training of each model – on each day of every fold – the model parameters are used to forecast using the test set. The predictions are appended to a data structure. The data frame captures the values of the prediction, the actual result, the identifier ('id'), the specific day, and any parameters associated with the grid search process in every model. Upon conclusion of any particular combination of the grid search, whether during the hyperparameter optimisation sequence or the selection of the model architecture, the data frame is saved in a .csv file format for preservation.

9. **Calculate MSE and MSE standard deviation for all hyperparameter combinations.** A data frame will capture the predicted and actual values of

each of the seven days in the forecasting horizon across the four folds used for cross-validation. The data frame will include the product id, the date, the dropout rate, the number of epochs and the batch size. The results for every combination are stored in a file.

10. **Select the best hyperparameters based on the results.** A data frame will capture the predicted and actual values of each of the seven days in the forecasting horizon across the four folds used for cross-validation. The data frame will include the product id, the date, the dropout rate, the number of epochs and the batch size. The results for every combination are stored in a file.

11. **Calculate MSE and MSE standard deviation for all model architecture combinations.** A data frame will capture the predicted and actual values of each of the seven days in the forecasting horizon across the four folds used for cross-validation. The data frame will include the architecture, number of neurons in the hidden layer, length of the input sequence and number of training samples. The results for every combination are stored in a file.

12. **Select the best model based on the results.** A descriptive analytical approach will be used to examine the best-performing models and discuss the experiment's outcome. A Kruskal-Wallis test will be conducted to observe if the architecture samples share the same distribution. The Kruskal-Wallis test is in non-parametric statistics tool for analysing data with more than two groups and is especially useful when assumptions of normality and homoscedasticity are not met.

13. **Analyse final results by performing a robust regression analysis.** The outcome of this step will assess the architecture comparatively as well as the number of neurons in the hidden layer. Finally and more importantly, it will answer whether or not the number of training samples and the length of the input sequence are statistically significant in impacting the accuracy of the models. As a result, the research hypothesis will be answered.

3.3 Data understanding

The data set contains a total of 42,840 series one for each Store-Product combination. The time series within the data set are highly correlated, which allows for cross-learning methods, and is organised in grouped series classified along 3 product categories (Hobbies, Foods and Households), 7 product departments across ten stores located in three States in the US (Makridakis, Spiliotis, Assimakopoulos, Chen, et al., 2021).

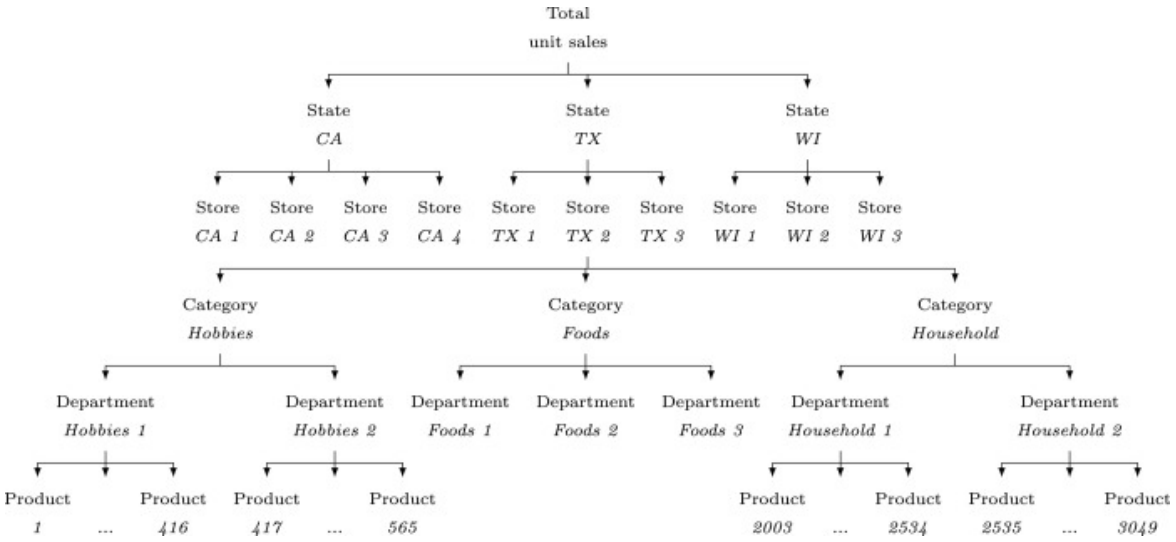

Figure 3.2: M5 products hierarchy. Source: (Makridakis et al., 2022)

3.3.1 Target variable 'Sales'

'Sales' is the dependent variable of the experiment. A broad view of most items reveals that most of the products in the data set display erratic demand patterns, characterised by long periods of zero demand followed by sporadic bursts. Therefore, the data in the experiment is categorised as intermittent according to the model proposed by (Syntetos & Boylan, 2005), which classifies periodic demand according to the Absolut Deviation from Idependently estimated mean (ADI) and Squared Coefficient of Variation (CV2) (2005). This model distinguishes four types of demand arise: smooth, erratic, intermittent, and lumpy. Smooth and erratic demand exhibit regular patterns; the former has reduced fluctuations in demand size, while the latter experiences significant variation. Intermittent and lumpy demand display irregular demand

intervals over time, with the former having limited variation in demand size, unlike the latter, which shows more significant variation (Rožanec et al., 2022). Although this approach has met some criticism, it helps to generally illustrate the nature of demand seen across the broad spectrum of products selected in the data set selected for this dissertation (Kostenko & Hyndman, 2006)

Table 3.1: Description of M5 (Walmart) data set

data set	Observations	Timeseries	Erratic	Lumpy	Smooth
M5 (Walmart)	1,507	30,490	2.88%	17.01%	6.76%

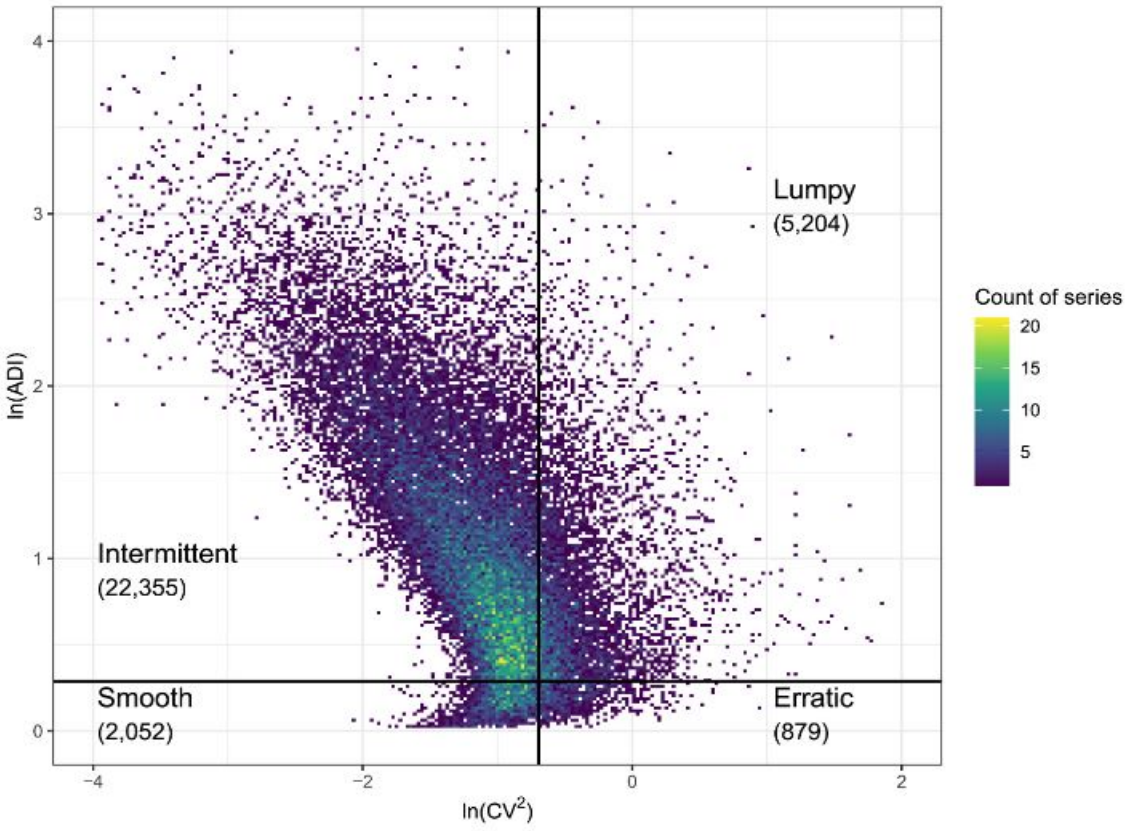

Figure 3.3: Source: (Makridakis, Spiliotis, & Assimakopoulos, 2021a)

The sales demand data adheres to what is known as a Tweedie distribution. The Tweedie distribution is a valuable tool for analysing intermittent demand because it can account for the unusual spread of patterns and inform on how to shape neural

network models. The Tweedie distribution can handle more considerable variations in the data than other theoretical distributions, including extreme values. Its more exciting feature, however, is that it can model zero-inflated data, which is common in the Walmart data set that the experiment uses, where there are many zero observations in the data.

3.3.2 Exogenous variables

The data set comprises complementary information in the form of two extra files. "Calendar" and "Selling prices". "Calendar" consists of the date and date-related features: weekday, month, year and day, and it spans from 29-01-2011 to 19-06-2016. Special days and holidays are grouped into four categories: Sporting, Cultural, National, and Religious, encompassing about 8% of the days in the data set. Sporting events account for 11% of these days, Cultural events for 23%, National events for 32%, and Religious events for 34%. SNAP activities, serving as promotions, are indicated by a binary variable (0 or 1). If the CA, TX, or WI stores allow SNAP purchases on a specific date, represented as 1. All three states have ten days each month with SNAP purchases allowed, meaning approximately 33% of the days are affected by these activities (Makridakis, Spiliotis, & Assimakopoulos, 2021a).

The selling prices are provided at a week-store level, with an average of seven days, and may change over time. If prices are unavailable, the product did not sell during that week.

3.4 Data preparation

This section describes the initial steps to be taken before conducting neural network training and evaluation. The main emphasis is on two crucial actions: data cleaning and feature engineering. Data cleaning involves preparing the data set by eliminating errors or inconsistencies. In contrast, feature engineering involves creating past sales sequences that will provide the model information during the fitting process.

3.4.1 Data cleaning

The main file where the sales information exists is in a relatively unique structure unsuitable for neural network training. In its original framing, the data frame presents each unique product ('id') in a row, while sales for each day are displayed in columns.

1. Trim the text of the day column to remove the string 'd_' and convert it to an integer.

2. Clip the data set maintaining the last two years of data to make the data set smaller for faster training and to remove any leading zeros in items for which there is no information during the first years of data.

3. Join the calendar and sell prices data sets to the primary data frame.

4. Apply label encoding to categorical variables.

3.4.2 Feature creation

Now that the data frame is in the sought-after structure - long as opposed to wide format, it is possible to shift the values of the 'sales' column from 1 to n times to obtain as many lagged values as desired. It is important to note that the values must first be grouped by 'id' to avoid shifting them onto a different product. The process continues by transposing the resulting array so that each row has the 'sales' value (y) and the associated past historical values (x).

Once the process is completed, the data is transformed using a MinMaxScaler scaler while filling all NaN values with a -1, which is common practice for handling missing values. In this case, there are two types of missing values. Some prices are missing from the data set, meaning those items did not sell during that week. Separately, when sales values are lagged, the first n observations of the lagged-n value are also NaN. Both cases are treated equally for simplicity, according to the guiding principles of the design experiment decided at the outset.

3.5 Modelling

This section is concerned with preparing the data set and training multiple models to assess their effectiveness in predicting intermittent demand. The optimisation of hyperparameters will be conducted through the use of a grid search, which will inform a subsequent search for comparing architectures side-by-side. In order to enhance the reliability of the results, the modelling process will be subjected to a rolling window cross-validation process.

3.5.1 Hyperparameters (HPO) grid search

Parameters:

- Batch size: [64, 512, 1024, 2048]
- Dropout rate: [0.1, 0.25, 0.5]
- Epochs: [25, 100, 250]

There are a total of 36 (4 x 3 x 3) potential hyperparameter configurations.

3.5.2 Architecture grid search

Parameters:

- Model architecture: [RNN, LSTM, LSTM bi, Conv1D, GRU]
- Size of training sample per product: [56, 112, 365]
- Length of input sequence: [56, 112, 365]
- Number of neurons in the first layer: [64, 128, 192]

There are a total of 135 (5*3*3*3) potential architecture configurations.

Algorithm 1 Hyperparameter search algorithm

```
Define hyperparameter search grid:
Batch size: [64, 512, 1024, 2048]
Dropout rate: [0.1, 0.25, 0.5]
Epochs: [25,100,250]
Initialize empty list for results
for each Batch size, Dropout, Epoch do
    for each CV Fold, day do
        Train model with current hyperparameters
        Evaluate model on validation set
        Predict on test set
        Save prediction and yhat
    end for
    Save file with results
end for
```

Algorithm 2 Architecture search algorithm

```
Define architecture search grid:
Model architecture: [RNN, LSTM, LSTM bi, Conv1D, GRU]
Training sample size: [56, 112, 365]
Input sequence length: [56, 112, 365]
Number of neurons in the first layer: [64, 128, 192]
Initialize empty list for results
for each Model architecture, Training sample size, Input sequence length, Number of neurons do
    for each CV Fold, day do
        Train model with current architecture and hyperparameters
        Evaluate model on validation set
        Predict on test set
        Save prediction and yhat
    end for
    Save file with results
end for
```

3.5.3 Sliding Windows Training

Sliding Windows

The sliding window method is a commonly used technique in time series forecasting. This method divides the time series data into a series of fixed-sized windows or subsequences, usually overlapping. Each window represents a specific time range and contains a set of observations. The algorithm defining the size and slide of the windows will take the number of folds as an input parameter. The folds created will overlap the least possible among themselves. Once the boundaries are specified, the model trains and predicts once on each window and, thus, simulate the same problem several times while keeping training & validation data different. After analysing the outputs, the results from all the windows determine the best and most general model overall. The sliding window method is simple and effective. However, it can also be computationally intensive if the window size is significant, as it requires modelling the time series multiple times. The total number of folds in the experiment is set to four, which strikes a balance between various distributions in the data and a performing experiment (R. J. Hyndman & Athanasopoulos, 2018).

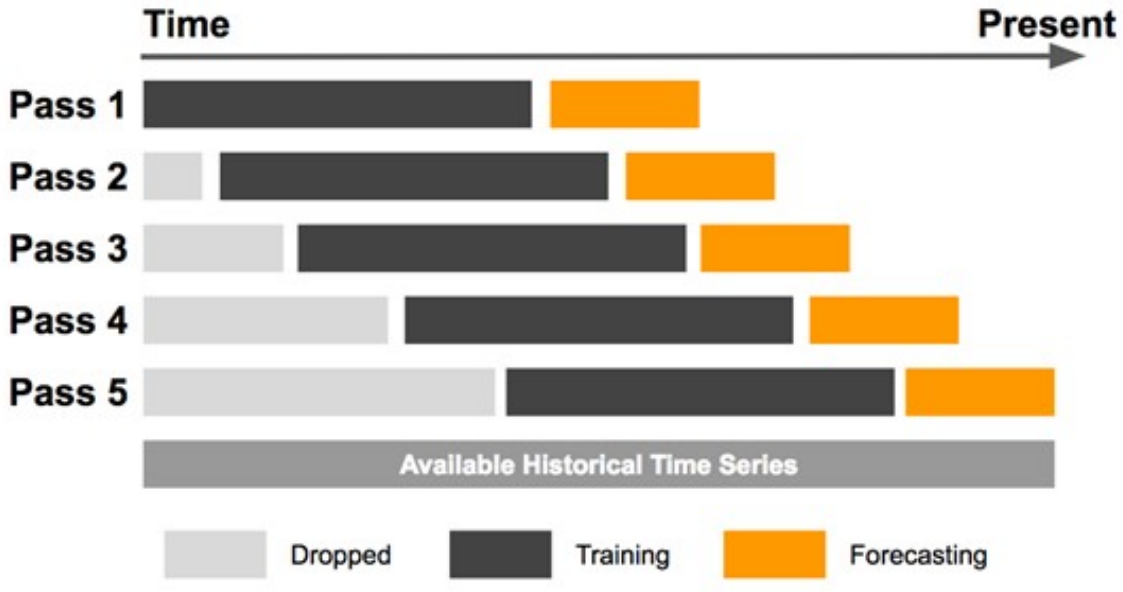

Figure 3.4: Sliding Windows. Source: (Brannan, 2022)

Multi-step predictions

The problem is modelled using as current historical data as possible by performing multi-step training and prediction without utilising predicted for training – multi-step recursion. The mathematical formulation of the multi-step approach can be formally represented as follows:

Let $X(t)$, $X(t-1)$, and $X(t-2)$ be the input features at time t, $t-1$, and $t-2$, respectively. Let $Y(1)$, $Y(2)$, and $Y(3)$ be the target variables to be predicted at time $t+1$, $t+2$, and $t+3$, respectively.

The model takes in $X(t)$, $X(t-1)$, and $X(t-2)$ and predicts $Y(1)$, $Y(2)$, and $Y(3)$, respectively:

- $Y(1) = f(X(t))$
- $Y(2) = f(X(t-1))$
- $Y(3) = f(X(t-2))$
- $Y(n) = f(X(t-(n-1)))$

where f is the prediction function learned by the model.

This approach assumes that the target variable at each future time step depends only on the input features at a specific historical time step; it does not introduce predicted values in the training process. In practical terms, each predicted day has to be trained and predicted separately. In the proposed experiment, a 7-day-ahead forecast is aimed for; thus, seven different models will need to be trained, one for each predicted day, regardless of the type of grid search or architecture used.

3.6 Evaluation

The experiment results incorporate several dimensions to facilitate thorough analysis, which provides a rich evaluation and enhances the context. Dimensions include

store-product combinations, the date and the cross-validation fold. All the detail supplied will unambiguously expose the performance of the different architectures tested, revealing the strengths and weaknesses of the top-performing models and parameters.

A broad-brush view of the outcome presents two branches of inferences. Firstly, an overview of the results - MSE - in the context of the dimensions. I.e. store, department, product, day & week aggregations. Secondly, it is possible to calculate the error spread precisely and thus assess the stability of the models.

The evaluation design will be essential for the resolution of some of the objectives in the Research Objectives section. More specifically:

- Determine optimised hyperparameters for training an SNN
- Assess the performance of the architectures used
- Determine if the number of training samples and the length of the input sequence impact accuracy
- Answer the research hypothesis

3.7 Limitations and delimitations

3.7.1 Data set

The current design only explores the Walmart data set provided for the M5 competition. It is noteworthy that the data, as presented, includes ten stores belonging to the Walmart chain in the United States. Consumer behaviour may differ in countries or regions that were not part of the original data set. Similarly, although this problem presents several parallels with other intermittent demand data sets in the retail sector, the outcomes may not be not directly applicable to problems within the same domain or industry. Though some generalisations can be substantiated based on the project's findings, these will require replication and additional validation. Several similar competitions share overlapping levels of demand intermittency, which should direct ensuing research, as a first step, to the replication and validation of results.

data set	Observations	Time series	Erratic	Lumpy	Smooth
M5 (Walmart)	1,507	30,490	2.88%	17.01%	73.35%
Greek retail firm	748	7,248	18.10%	41.75%	29.57%
Corp. Favorita	1114	174,654	20.65%	30.91%	25.37%

Table 3.2: Summary of data sets (Theodorou et al., 2021).

Another constraint is the ability to model promotions as an exogenous variable, a feature not included in the data set that possesses high predictability power in time-series forecasting problems (Doszyń, 2019). Additionally, the inability to distinguish between zero sales due to lack of demand and zero sales due to unavailability of a product severely limits the ability to model the data appropriately. In fact both scenarios represent opposite extremes. In the former, there is no demand. In the latter there would have been demand had it not been unavailable. The corollary is that unavailability should have been counted as actual demand but is not possible to model it.

3.7.2 Shallow Architectures

The project focuses on SNNs for demand forecasting and only considers historical sales as the input features for model learning. The experimentation with these elements could be expanded, providing a richer and more comprehensive modelling setting. For example, other practitioners might use hand-crafted features or a combination of hand-crafted and raw historical sales. Alternative models that blend heterogeneous feature extraction techniques merit further investigation as they fuse the flexibility of a straightforward implementation and the sophistication of methods that win forecasting competition prices. It should be emphasised that the application of shallow networks was motivated by the aim of avoiding the complexities and time constraints of hyperparameter optimisation and other model tunning activities. Accordingly, it should not be assumed that using only historical sequences as input features necessarily results in superior accuracy. Their simplicity makes them easier to incorporate into

an ML pipeline and to deploy in a real-world setting. It would be short-sighted to assume that feature engineering with either exogenous variables or dependent variables would not lead to improved results (LeCun & Bengio, 1998).

3.7.3 Architecture selection

The choice of architecture dramatically influences the overall behaviour of the model, having a significant impact on the outcome of the experiment. As a result, selecting the appropriate neural network architecture for the experiment was crucial and posed an evident challenge. Based on the literature review, two major families of layer types were identified, namely recurrent networks and convolutional networks. However, a deeper examination revealed a vast array of options within each family. For example, among the common types of recurrent neural networks are Simple RNN, Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and bidirectional RNN. In the convolutional space, some of the commonly used networks are Simple Convolutional Neural Networks (ConvNet or CNN), Temporal Convolutional Networks (TCN), and Attention-based Convolutional Networks (ACNN). Additionally, hybrid networks such as Convolutional LSTM (ConvLSTM), Encoder-Decoder LSTM, and Convolutional GRU (ConvGRU) also exist (Srinivas et al., 2016).

The experiment uses well-tested neural network architectures (Yamak et al., 2019) to ensure the results obtained are not domain-specific and have the potential for generalisability (Siami-Namini et al., 2018). A summary of the literature shows that the most suited NN architectures for intermittent demand forecasting and time series forecasting, in general, can capture the underlying temporal patterns within the data. LSTM and CNN networks have shown promising results in this domain, and hybrid NN architectures have also shown advantages. Thus, after reviewing the literature, a smaller selection of architectures was chosen. The fallout of that decision is twofold.

On the one hand, the experiment only tests a small sample of options within a vast sea of choices, some of which are very suited challengers to the task at hand. On the other hand, as shown in research, it is common to opt for deep architectures for this and similar tasks that could not be tested and are left as a potential path for future

investigation. One of the most popular time series forecasting tools in recent years is transformers, which have been excluded from the experiment altogether. They use attention mechanisms to weigh the importance of different parts of the input sequence in generating the output. The attention mechanism lets the decoder focus on the most relevant parts of the encoder's output for each time step, handling long sequences and model dependencies between time steps. Similar to neural network designs discussed here, the architecture adheres to a supervised learning problem to minimise a loss function that measures the difference between the network's predictions and the true output (Zerveas et al., 2021).

3.7.4 Limited grid search

The first significant decision in the experiment design was establishing a course of action to determine the best model for the task. Yu and Zhu discuss the effectiveness of neural networks in applications highlighting how inefficient it is to obtain a working production-ready model. They suggest that achieving an accurate model is considered a brute force method that requires a large data set and dedication to model design, algorithm design, and hyper-parameter selection, leading to a high cost for its application. They argue that the most commonly used method for determining hyper-parameters is one's experience, but this approach lacks logical reasoning and weakens the credibility of empirical research (2020). The data suggested that training all possible combinations (i.e. 36 HPO models * 135 architectures = 4,860 models) was undesirable. A grid search of the entire parameter space may generate many low-performing models and, therefore, not be worth exploring. Thus, the high time cost was avoided by breaking out the grid search into two stages at the expense of not cross-referencing all possible combinations of hyperparameters.

Robinson et al. (2006) argue that direct evaluation of the target data set is the most common and straightforward method in their review of algorithms for hyperparameter optimisation. This method involves analogous training of models with different hyper-parameter sets until they converge when their error loss or accuracy can be quantified. Although this method is the most accurate, it is also slow and inefficient, taking a

long time between samplings. Due to its high cost in terms of time, it is not feasible for users with limited resources. Hence, they advocate for faster methods that favour evaluation speed over accuracy. It is also commonly accepted that a direct approach is less empirical as it relies mainly on an individual's expertise. So to that effect, it was decided to use an algorithm that walks all permutations of the grid search and provides the test performed with a formal experimental backing.

3.7.5 Hyperparameter optimisation

Similarly, other hyperparameters were not a part of the experiment hyperparameter optimisation process—namely, loss optimisation and activation functions. In this case, the loss and activation function designations are based on an extensive literature examination. Activation functions are applied element-wise to the output of each neuron, determining the activation or output signal. Choosing the activation function helps find complex non-linear relationships between inputs and outputs. The decision to utilise ReLU is due to its simplicity and effectiveness. ReLU is computationally efficient and does not require exponential or trigonometric operations. ReLU solves the vanishing gradient problem by returning 0 for negative inputs, resulting in faster convergence and improved performance. It also eliminates the issue of dead neurons, as the gradient for positive inputs is always 1, keeping the neurons active. All in all, ReLU is the most widespread and robust activation function across domains and architectures (Jentzen et al., 2021).

Loss optimising functions measure the difference between the predicted and actual outputs, and its value updates the weights during the backpropagation phase of training. According to research, the reason for choosing Adam as it signifies an improvement over the Stochastic Gradient Descend (SGD) and Root Mean Square Propagation (RMSProp) algorithms, combining their advantages. Adam uses moving averages of the gradient and squared gradient to adapt the learning rate for each parameter dynamically. This results in faster convergence and improved performance, making it a popular choice for deep learning tasks. Additionally, Adam requires fewer hyperparameters to be tuned compared to other algorithms, making it more user-friendly

(Jais et al., 2019). An optimal combination of the activation and loss optimisation functions can lead to faster convergence, better accuracy, and a more stable training process for the neural network.

3.7.6 Data scaling

Data scaling, as seen in the literature review section, resolves that data scaling can be considered a separate hyperparameter of its own. Various sources support that results fluctuate significantly based on the selected scaling method. In this dissertation, Min-Max-Scaling is the method of choice for scaling the data for two primary reasons: its ease of implementation and its ability to effectively handle sparse data, a characteristic of the demand data. Additionally, Sinsomboonthong Sinsomboonthong in a comparative study of several scaling methods across a large variety of widely used data sets demonstrates that Min-Max-Scaling is the best-performing one (2022).

3.7.7 Categorical encoding

Categorical encoding presents a vast assortment of options. Among the most popular algorithms used within the neural network domain are one-hot encoding, label encoding, target encoding, and embeddings. Once again, the choice of encoding method can significantly impact the outcome of an experiment (Hancock & Khoshgoftaar, 2020). In this instance, categorical variables have been converted to numeric using label encoding, a simple approach that has been widely used and has a proven track record of success.

Embeddings merit additional discussion, as they require using a unique architecture. Embeddings represent categorical variables as dense vectors in a low-dimensional space, making it easier for the neural network to learn the relationships between the categories and the target variable. By transforming categories into dense vectors, embeddings capture the relationships between categories in a continuous and meaningful manner, enabling the neural network to learn more sophisticated representations of the data (Russac et al., 2018). There are several reasons why the use of embeddings

was discarded:

1. If a categorical variable lacks significant predictive power, representing it as a dense vector using an embedding layer can result in overfitting and decreased performance when combined with other features.

2. The size of the output vector must be adjusted based on the number of unique categories and the desired level of complexity. A larger embedding dimension can capture more complex relationships between categories, but it also increases the number of parameters in the model, leading to overfitting.

3. Furthermore, related to the previous point, this would increase the overall complexity of the proposed architecture, which is contrary to one of the goals set out in this dissertation, which is to achieve maximum generalisability.

3.7.8 Size of training data

Neural networks generally require large amounts of data to identify patterns effectively (Hastie et al., 2013). In this sense, the size of the data imposes an inescapable constraint on the experiment scope in the search for the optimal model. Despite efforts to optimise memory management, the compiler crashed several times when handling arrays of sizable dimensions. This restricted both the number of training samples and the number of features in the input limited used for prediction and experimentation. Preliminary experiment results suggest that larger input sequences and sample sizes yield better results when applied to the particular task at hand. However, the limitations in data size precluded determining the point at which increased size begins to impact results negatively, resulting in inconclusive findings.

3.7.9 Comparative analysis of other algorithms

Another limitation is that the experiment was bounded to SNNs, thereby limiting the scope of the research. While the findings might prove a case for the effectiveness of SNNs for demand forecasting, the bias in favour of SNNs precludes a fair appraisal of

various algorithms' strengths and weaknesses. As a result, the available options for addressing a problem of a similar nature become scarce. Another decision was that the sole input of the network would be historical 'sales' sequences complemented by exogenous variables given to us without incorporating hand-crafted features. For example, standard time series forecasting techniques utilise moving averages, lags and differencing to capture the trend and level of the series. These are very easy to manufacture by a person and possess significant predictive power, as demonstrated countless times (Bojer & Meldgaard, 2021). Similarly, the price data included offered the opportunity to model it in ways that immediately increased the information gain of the model, such as price elasticity or price during special days. This decision may have limited the model's ability to capture complex relationships that a human with domain expertise could have otherwise articulated. However, this approach raises several concerns, including the difficulty in determining the impact on the performance of algorithm selection versus modelling choices. For instance, the winner of the m5 competition utilised an ensemble of 6 a gradient-boosted trees algorithm (LGBM) with direct and auto-recursive forecasting techniques (Makridakis, Spiliotis, & Assimakopoulos, 2021b). Only a sophisticated statistical approach would be able to establish what factors actually contributed to a winning execution. All these reasons, though a limitation, motivated the decision to confine the design to only exploring analogous models while leaving the door open for further testing.

3.7.10 Custom evaluation metric

An alternative method to evaluation loss is to incorporate a tailored metric that is not just the epoch where the validation score was the lowest but one that acts more nuanced. One possibility is to establish a metric that calculates the mean of both the training loss and validation loss and select the model that retains the minimum value of this custom metric. This approach ensures that model selection is not solely based on the validation set but a combination of the training and validation results.

3.7.11 Proof of concept (POC) to production gap

The following limitation of the experiment is commonly referred to as the Proof of Concept (POC) to production gap. The authors of a publication Paleyes et al. (2022) discuss how deploying ML models in production systems poses a multitude of software engineering challenges, one of which is model and data drift. Model and data drifting refer to the gradual changes in the underlying data distribution or relationships between the input and output variables in a ML model over time. An example will help illustrate how concept drift has the potential to impact the experiment negatively. During the COVID-19 pandemic, consumer demand was the subject of a major shift and exhibited unpredictable behaviour. One of the most apparent outcomes was that online shopping became more prevalent than retail while quarantines were compulsory. The results shown in this dissertation could have been disastrous had these models been deployed during the COVID period. From a computational perspective, the drift can be addressed by continuously updating the model to account for new data or by using transfer learning techniques that enable the model to generalise to new data distributions (Karmarkar et al., 2020). This approach, however, can only be slightly tamed by applying cross-validation techniques without an assurance that the model will continuously perform in time. For these reasons, it should be expected that models become less accurate and lead to decreased performance or even complete failure in real-world applications, a limitation not fully addressed in this experiment.

3.7.12 Experiment design summary

This section condenses the information presented in the earlier section and organizes it into the following categories:

Strenghts:

- The data is cleaned and feature engineered with a minimal number of steps, facilitating an easy implementation and avoiding the excessive use of computational resources.
- The experiment conducts a hyperparameter grid search to fine-tune the model before evaluating the architectures, number of training samples, and input sequence length.. As a result, this study contributes to the body of literature with a better benchmark for future research.
- Sliding windows training is a cross-validation technique that allows for as much historical data as possible to be used in training and prediction. At the same time, employing four folds guarantees that the outcomes are not excessively adjusted to one dataset, thereby providing more robust results.
- Multi-step predictions ensures that the most recent observations are used for training and predicting, demonstrating a better efficacy, and more robustness and reliability (Livieris & Pintelas, 2022). Additionally, the method employed does away with auto-recursive techniques, which incorporates the prediction on the next step forecast. Brownlee (2019) states that a recursive strategy can suffer from error accumulation and error propagation issues, as each prediction depends on the previous one and any error made by the model will affect all subsequent predictions.
- The experiment results are recorded and evaluated on several dimensions, including store-product combinations, the date, and the cross-validation fold. This

provides a rich evaluation and enhances the context, making it possible to assess the stability of the models and reveal the strengths and weaknesses of the top-performing models and parameters.

Assumptions:

- The Walmart data set provided for the M5 competition is a representative sample of Walmart stores in the United States.
- The Walmart data set is comparable to other intermittent demand data sets in the retail sector.
- Historical sales data can be used to forecast future sales.
- Min-Max-Scaling is the best scaling method for the data set.
- Label encoding is an appropriate categorical encoding method for the data set.
- The findings are generalisable to other similar data sets in the retail sector.
- The fixed hyperparameters selected for the shallow neural network models are appropriate for the study's research objectives.

Limitations:

- The study only explores the Walmart data set provided for the M5 competition and may not apply to other regions or countries.
- The outcomes may not be directly applicable to other problems within the same domain or industry.
- The ability to model promotions as an exogenous variable, which was not included in the Walmart data set.

- The inability to distinguish between zero sales due to lack of demand and zero sales due to the unavailability of a product limits the ability to properly model the data.

Delimitations:

- The experiment uses well-tested neural network architectures to ensure the results obtained are not domain-specific and have the potential for generalisability.
- Only focuses on shallow neural network architectures, and no comprehensive comparison with other ML algorithms is performed.
- The findings may be biased in favour of shallow neural network models, and may not provide a fair appraisal of various algorithms' strengths and weaknesses.
- The study only explores the Walmart data set provided for the M5 competition and does not investigate other data sets in the retail sector or other industries.
- The study assumes that historical sales data is the only relevant input feature for demand forecasting with shallow neural network models.

Chapter 4

Results, evaluation and discussion

4.1 Model stability

An essential feature of the model is how stable so that we can assess whether the models can perform in the presence of noise, extreme values or uncertainty. The robust regression analysis shows that the GRU architecture has a significant impact on the predicted MSE SD, with a statistically significant coefficient of -8.01 (p-value <0.001). The RNN architecture also affects the predicted MSE SD, though with a lower coefficient of -6.72 (p-value <0.007). Both LSTM architectures do not achieve statistical significance with p-values over 0.05.

In addition, increasing the number of neurons is associated with a decrease in MSE SD, with a negative and statistically significant coefficient of -0.0309 (p-value <0.001), while increasing the number of days used for training the model is associated with a decrease in MSE SD, with a negative and statistically significant coefficient of -0.0216 (p-value <0.001). Lastly, the number of lag values used in the model does not significantly affect MSE SD, with a coefficient of 0.0011 and a p-value of 0.717.To run the robust regression test on the data to generate results to accept or reject the research hypothesis.

Table 4.1: Model stability (MSE SD): Robust regression analysis

	coef	std err	z	P> |z|	[0.025-0.975]
Intercept	52.25	2.29	22.78	0.000	47.75-56.74
GRU	-8.02	2.44	-3.28	0.001	-12.81–3.23
LSTM	-4.37	2.43	-1.80	0.072	-9.14-0.40
LSTM bidirectional	-4.51	2.44	-1.85	0.065	-9.30-0.27
RNN	-6.72	2.44	-2.75	0.006	-11.51–1.94
Number of lag values	0.00	0.00	0.36	0.717	-0.01-0.01
Neurons	-0.03	0.01	-3.96	0.000	-0.05–0.02
Training days	-0.02	0.00	-7.31	0.000	-0.03–0.02

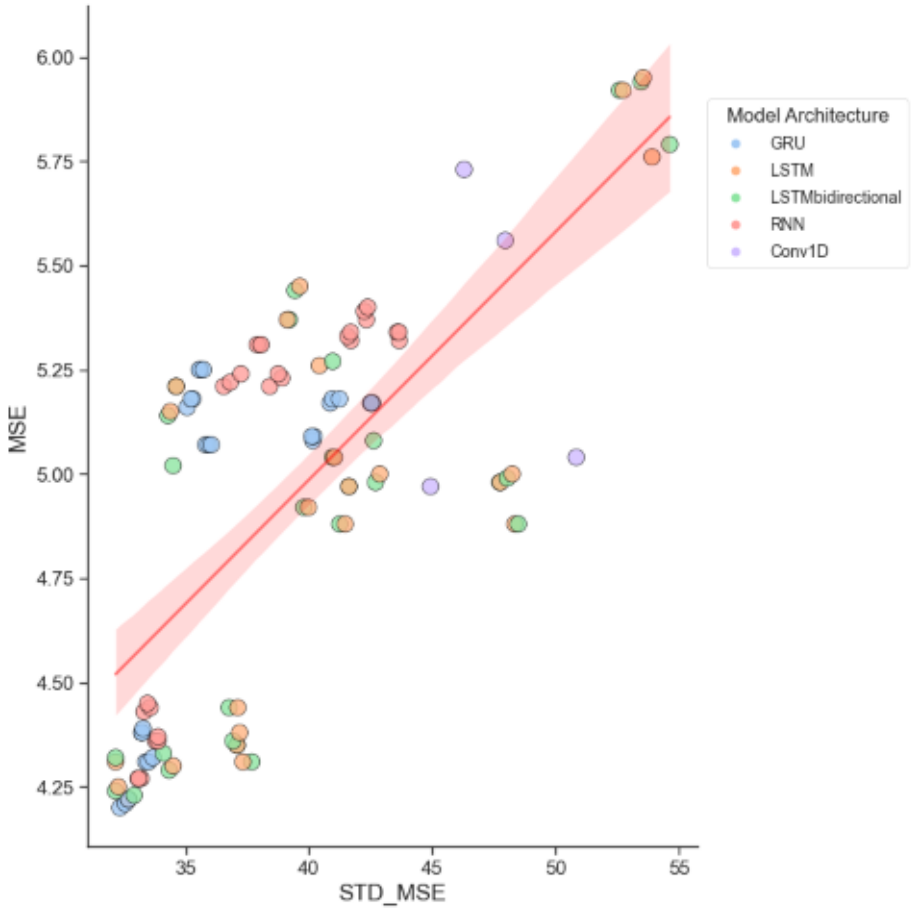

Figure 4.1: Model stability MSE vs. MSE SD

4.2 Hyperparameter optimisation evaluation

HPO was performed to determine the optimal batch size, number of epochs, and dropout rate for an LSTM model. The LSTM architecture was used as an anchor while the other parameters were permutated. The initial grid search explored batch sizes of 64, 512, 1024, and 2048; epochs of 25, 100, and 250 with ten patience for early stopping and dropout rates of 0.1, 0.25, and 0.5. Given the potentially large number of iterations and combinations involved in selecting the optimal model architecture, the initial results from HPO guided the ulterior architecture search.

Table 4.2: Top 5 models performance HPO

Batch size	Epochs	Dropout	MSE	STD
2048	25	0.50	4.23	32.91
2048	25	0.25	4.28	34.53
1024	25	0.50	4.29	34.69
1024	25	0.25	4.30	35.37
2048	25	0.10	4.32	35.73
1024	25	0.10	4.35	36.90

The combination of a batch size of 2048, a dropout rate of 0.5, and 25 epochs with early stopping was selected as the optimal option. This combination obtained the lowest mean squared error (MSE) values among the tested hyperparameter configurations.

Table 4.2 is a regression output obtained using a Robust Linear Model. The model is fit to predict the Mean Squared Error (MSE) based on three predictor variables: batch size, epochs, and dropout rate. The output provides information on the coefficients (weights) of the predictor variables, their standard errors, the z-scores, and the associated p-values.

The intercept, batch size, epochs, and dropout coefficient estimates are 4.6451, -8.066e-05, 0.0013, and -0.2040, respectively. The intercept is the predicted MSE when

Table 4.3: Model performance (MSE): HPO robust regression analysis

	coef	std err	z	P> |z|
Intercept	4.64	0.090	51.404	0.000
batch_size	-8.06e-05	3.43e-05	-2.350	0.019
epochs	0.001	0.000	3.359	0.001
dropout	-0.204	0.205	-0.993	0.321

all predictor variables are zero. The negative coefficient for the batch size variable suggests that a decrease in batch size is associated with an increase in MSE. The positive coefficient for the epochs variable suggests that an increase in the number of epochs is associated with an increase in MSE. Lastly, the negative coefficient for the dropout variable suggests that an increase in the dropout rate is associated with a decrease in MSE. The standard errors for the coefficient estimates are provided, which can be used to calculate the t-statistics (z-scores) for testing the null hypothesis that the coefficients are equal to zero. The p-values suggest that batch size and epochs are significantly associated with MSE at a significance level of 0.05, while dropout is not significantly associated with MSE. Overall, the regression analysis reveals that batch size and epochs statistically impact MSE, whereas dropout does not. Nonetheless, it is worth noting that the robust linear model was applied to account for the presence of outliers and influential observations in the data. While the dropout variable is not associated with statistical significance, the best overall hyperparameter combination based on MSE was selected to enable the grid search for architecture selection to proceed.

An interpretation of the results suggests that intermittent demand modelling vastly benefits from learning general patterns, which facilitate the model to generalise better to new and unseen data and ignore the relatively high volume of noisy information. The relatively high dropout rate encourages the network to learn more general features to prevent overfitting. It can be thought of as a form of ensemble learning where multiple networks are trained on different subsets of the input features. These results

are consistent with the data distribution properties that contain many intervals with zero demand (Srivastava et al., 2014).

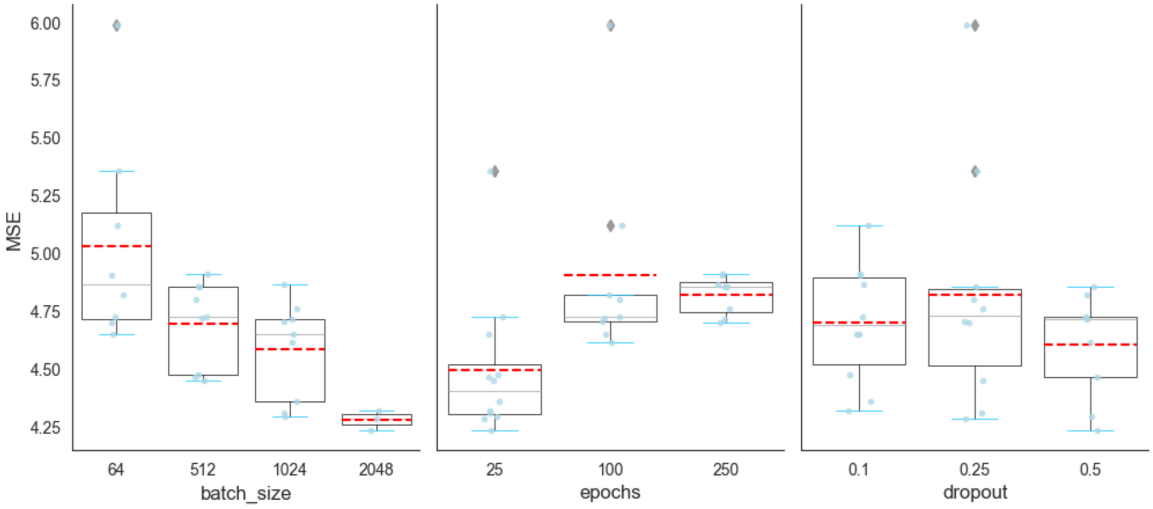

Figure 4.2: MSE comparison across different model parameters

A similar reading can be made of the low number of epochs (25) that yield the best accuracy. Fewer training rounds can prevent overfitting, which occurs when the model learns the noise and specific features of the training data instead of generalising to new observations. By using a low number of epochs, the model is forced to learn the most important and general patterns in the data; this is particularly useful when dealing with intermittent demand modelling, where the model needs to be able to recognise and predict patterns even when there are extended periods of no demand or when the demand is sporadic (R. J. Hyndman & Athanasopoulos, 2018).

4.3 Architecture evaluation

The architecture selection was performed to identify the optimal model configuration for LSTM, LSTM-Bi, RNN, GRU, and CNN models, considering the number of training samples, the input sequence length, and the number of neurons for the hidden layer as parameters. To perform this task, an iterative approach was employed, where each architecture was evaluated with different combinations of these parameters. The initial search explored the following architectures: 64, 128, and 192 neurons for LSTM,

LSTM-Bi, RNN, GRU, and CNN. The search space for the number of training samples and the input sequence length was set to 56, 112, and 365.

Given the potential combinatorial explosion of possible architectures, the initial results from the hyperparameter optimization (HPO) guided the architecture search. The HPO determined the optimal batch size, number of epochs, and dropout rate for the LSTM model. The combination of a batch size of 2048, a dropout rate of 0.5, and 25 epochs with early stopping was selected as the optimal choice based on the lowest mean squared error (MSE) values among the tested hyperparameters. These initial results were used as a benchmark to determine the optimal architecture among the evaluated ones, the number of training samples and the length of the input sequence.

Table 4.3 is a regression output obtained using a Robust Linear Model. The intercept value of 6.1217 indicates that when all independent variables are held constant at zero, the predicted value of the dependent variable is 6.1217. The negative coefficients for the different model architectures - GRU, LSTM, LSTM bidirectional, and RNN - suggest that changing the model architecture from the baseline to any of these architectures is associated with an increase in the predicted value of the dependent variable. However, the magnitudes of these coefficients range from -0.5044 to -0.6229, indicating that the choice of model architecture does not substantially affect the dependent variable's predicted value. The coefficient for the length of the input sequence is also negative, suggesting that an increase in the lag values is associated with a decrease in the error value of the dependent variable. Increasing the input sequence length may significantly improve the model's predictive performance.

The coefficient for neurons is negative but small in magnitude, and the associated p-value of 0.142 suggests that the effect of neurons on the predicted value of the dependent variable is not statistically significant. The number of neurons, in the range between 64 and 256, implies that the number of neurons used in the model's hidden layer is unlikely to impact the model's performance significantly.

In contrast, the coefficient for train days is negative and relatively significant in magnitude, indicating that an increase in the number of train days is associated with a decrease in the error value of the dependent variable. This result is statistically

Table 4.4: Model performance (MSE): Architectures robust regression analysis

	coef	std err	z	P> |z|	[0.025-0.975]
Intercept	6.1217	0.068	90.400	< 0.001	5.989-6.254
GRU	-0.6148	0.072	-8.525	< 0.001	-0.756-0.473
LSTM	-0.6149	0.072	-8.558	< 0.001	-0.756-0.474
Bi-LSTM	-0.6229	0.072	-8.638	< 0.001	-0.764-0.482
RNN	-0.5044	0.072	-6.994	< 0.001	-0.646-0.363
Sequence Length	-0.0003	8.75e-05	-3.588	< 0.001	-0.000-0.000
Neurons	-0.0003	0.000	-1.468	0.142	-0.001-0.000
Training samples	-0.0030	8.73e-05	-34.661	< 0.001	-0.003-0.003

Note: The statistical results of the Sequence Length variable, denoted red, highlight its significance in answering the research question. Refer to Figure 4.3 for the Z-scores of all independent variables.

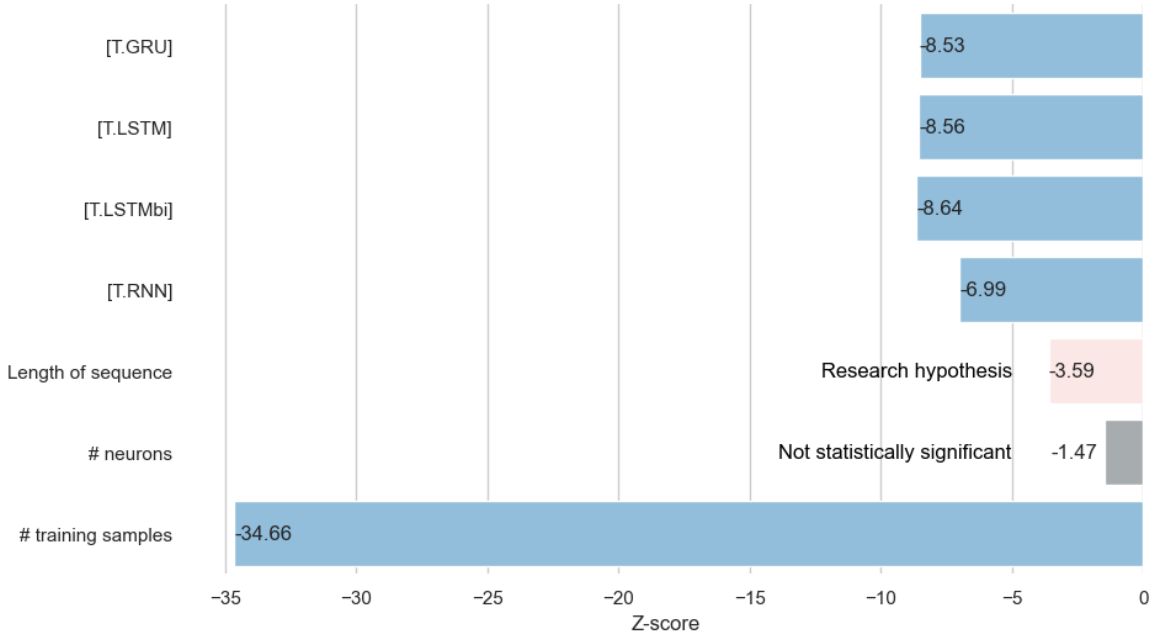

Figure 4.3: Regression z-scores for all independent variables

significant with a very small p-value, suggesting that a more extended training period will likely lead to improved model performance. Finally, the standard errors associated with the coefficient estimates are all relatively small (0.072 or less), indicating that the estimates of the coefficients are relatively precise. Overall, the results of this study suggest that the choice of model architecture has a relatively modest impact on model performance and that other factors, such as the length of the input sequence and training period, may be more critical in determining the model's predictive accuracy.

In summary, the experiment finds that the choice of model architecture has a relatively modest impact on model performance, and the length of the input sequence and the training period was more critical in determining the model's predictive accuracy - more so the latter than the former.

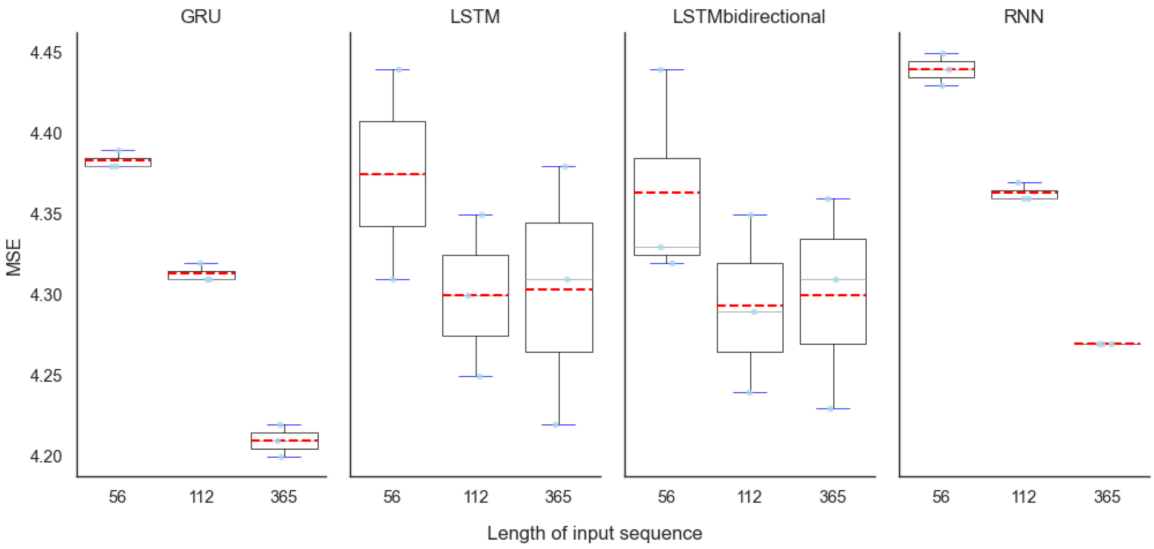

Figure 4.4: Model performance across different architectures

4.4 Hypothesis evaluation

This section evaluates the hypothesis used to address the research question. The hypothesis is evaluated at a significance level (α) of 0.05.

Based on the regression results, it is apparent that the length of the input sequence and the SNN architectures, including GRU, LSTM, LSTM-bidirectional, and RNN, are significant factors that affect the model's performance. The p-value of the length

of the input sequence coefficient is less than 0.05, indicating that there is a relationship between the two variables. Additionally, the model architectures' negative and statistically significant coefficients imply that they are associated with improved performance. Therefore, the null hypothesis is rejected, and it is concluded that the input sequence length significantly affects the models' performance.

4.5 Strengths and limitations of the results

4.5.1 Limitations

- The results are grouped at the architecture level. A more granular account of the outcome could have been provided had other aggregations been studied. For example, it would be interesting to understand the relationship of the architecture, number of training samples and length of the input sequence in specific slices of the same data: products divided by intermittency, stores & departments, categories.

- The lack of interpretability of the results might misguide future research: Neural networks are often considered a black box and it is difficult to unravel how the model arrives at its predictions. In particular, the HPO results only allow for conjectures about intermittent data favouring more underfitting strategies. A remmediation, to some extent, would be the application of Explainable Artificial Intelligence (XAI) techniques, that aim at making the decision-making process (Vilone & Longo, 2021).

4.5.2 Strengths

- The results offer a detailed analysis of model stability, HPO evaluation, and architecture evaluation, which can help researchers understand the models' performance in various scenarios and according to different criteria.

- The evaluation presents statistical results and regression outputs, which provide

rigorous insights into the relationship between different variables and the dependent variable. The findings go beyond descriptive statistics and pin down causal relationships between architectural choices and model accuracy.

- The use of robust regression analysis for both the HPO and architecture evaluation help account for outliers and noise in the data, resulting in more reliable and resilient findings.

- The architecture evaluation contributes, on top of answering the research question, with a quantification of the effect of the number of training samples in model accuracy. It can be validated that an increased number of training samples awards improved performance.

Chapter 5

Conclusion

5.1 Research Overview

The main purpose of the research is to investigate the validity of using historical information – unprocessed – as a means to learn patterns about intermittent demand data. More specifically, to evaluate the impact of training sequence length on the accuracy of neural network models and investigate the relationship between sequence length and the precision of the resulting predictions. The models examined are a selection of SNN architectures that facilitate a straightforward business implementation, aiming at yielding results early on. The architectures' selection process ensures that their design properties are fit for dealing with sequential and temporal data.

The research three objectives that have been achieved are as follows:

1. To conduct a literature review on the research statement.
2. To conduct empirical primary research to address the research problem.
3. To evaluate the effectiveness of the research experiment in addressing the research statement

5.2 Problem definition

Organisations in the FMCG industry face capacity and materials planning challenges derived from the inability to make accurate sales predictions. Their demand forecasts are substandard compared to advanced techniques, leading to imprecisions down the stream of processes that ingest and consume this information. Inaccurate predictions result in practical problems that take time and effort to address appropriately. The growing variety of products offered by companies in the market has introduced new challenges to the production schedule. Even products with relatively low sales volume can have a significant financial impact, adding to the complexity of the situation. The experiment proposes using an SNN capable of handling temporal and sequential data points and, therefore, capable of making accurate predictions. Using an SNN is advantageous because it is simple to deploy and utilises the abundance of historical data available to enhance the robustness of the solution. In this approach, the data extraction and modelling stages are uncomplicated and not computationally intensive.

5.3 Design/Experimentation, evaluation & results

The skeleton of the experiment rests on the CRISP-DM framework that informs the three main stages of the experiment: data preparation, modelling and evaluation. Data preparation involves processing the data to convert the ask into a supervised ML problem so that the neural networks can ingest the data in the format required. Modelling is concerned with training models and recording predictions from the architectures brought to the test while examining a matrix of parameter configurations. Finally, the evaluation involves applying statistical techniques to extract meaningful information and evaluate the set objectives.

The evaluation and interpretation of the results are approached from the lens of real-world applications. Businesses are interested in trustworthy information when implementing and using predictive models; it allows them to operate within bounded variance and deal better with uncertainty. Therefore, prediction stability is a sought-after feature that will be a beneficiary of the model selection process. For shallow

architectures, the results find that the GRU and RNN architectures lower the spread of the error significantly in dealing with intermittent demand forecasts and thus position themselves as a better choice for achieving variance-stable estimates. Other elements, such as the number of neurons in the model's hidden layer, are unlikely to impact the model's variance significantly. The results find a statistical significance in that increasing the number of days used for training the model leads to a smaller error spread, indicating that the model becomes steadier with more extended training. Conversely, the input sequence length has no significant effect and therefore does not make the model more stable.

Through HPO the study determined the optimal batch size, number of epochs, and dropout rate. The combination of a batch size of 2048, a dropout rate of 0.5, and 25 epochs with early stopping were selected as the optimal choice based on the lowest MSE values and statistical significance. These parameters suggest that the success of intermittent demand forecast sides with conservative models that are shaped to avoid overfitting. Regarding architectures and their accuracy, GRUs have demonstrated, although with a limited effect, the best results with statistical significance. Also, with significant statistical weight, the input sequence's length and the training period's duration affect the accuracy more positively. Specifically, the number of training samples appears to have a greater impact on performance overall.

5.4 Contributions and impact

The contributions of this dissertation have the potential to advance the development of best practices for intermittent demand forecasting in the retail sector. The findings can lead to organisations incorporating similar ML solutions into existing business operations providing they have the required technical infrastructure. By avoiding the need for manual feature engineering, this approach can expedite the time-to-market in comparison to conventional demand forecasting methods, which should, in turn, yield benefits early on. The results demonstrate that organisations could improve their forecasting accuracy if still reliant on manual and direct forecasting methods

or establish a high-performing baseline for future development. The solution offers a systematic forecasting methodology including most of the stages of an ML project: scoping, data and modelling, enabling a lift-and-shift and easy transition. A practical implementation should bring about stability and reliability, resulting in cost savings, improved customer service, and increased profitability.

The research findings also contribute to the broader academic field of forecasting and ML, providing new insights and opportunities for future research. In particular, and as previously mentioned, there are still some research gaps concerning ML and intermittent demand forecasting. Additionally, using only historical data as input features is innovative in a field where hand-crafted features are the norm. There are two specific outcomes of the experiment that have not widely discussed in the literature. It seems, judging from the results obtained that intermittent demand is better modelled with strong overfitting guard rails, so that the model is able to learn the right level of the demand. In other words, intermittent demand forecasting requires strong overfitting prevention mechanisms. This point is illustrated by the very high dropout rate (0.5) and the small number of epochs (25) which resulted in statistically significant improvements in performance.

5.5 Future work & recommendations

A re-examination of the assumptions and limitations should be the starting point for future research. There exist two meaningful and distinct points for subsequent development. On the one hand, the choice of a shallow architecture profoundly inhibits the ability of the models to effectively capture features and patterns from high-dimensional data due to their limited depth and representational capabilities. Deep learning models have proven their ability to conceive complex and hierarchical representations of the information provided. Intermittent demand appears to be such an ML challenge, where only large amounts of data can truly inform of real demand. It would be intriguing to explore deep learning architectures that incorporate an SNN as a feature vector generator continuing the work presented. A shallow network combined with

hand-crafted features can take advantage of the strengths of automatic and human feature creation. This approach is comparable to using an embedding layer for feature creation and has demonstrated its effectiveness in some use cases(Yao et al., 2017)

On the other hand, focusing on the applicability of the proposal, it would be appropriate to directly compare and evaluate uncomplicated SNN architectures with top-performing algorithms such as GBDT. A comparison between the two approaches though should be based on training speed while achieving a minimum performance baseline, which is one of their top-selling features. Other related assumptions could be deemed the subject of future work as well. Likewise, the exploration of transfer learning is a staple in the ML field and is heavily relied upon for a multitude of tasks. Transfer learning offers the advantage of reducing the time spent on modelling and thus presents itself as a suitable challenger.

Lastly, this dissertation has not directly addressed the issue of new products for which it does not exist any previous historical data. Although this problem may not be apparent due to error metrics having a biased representation of the overall data set, it can become problematic if not handled appropriately. One possible solution is to use a "proxy" product to forecast new items. Algorithms like k-nearest neighbours (KNN) can help pinpoint the closest related item.

References

Armstrong, J. (2001). *Principles of Forecasting*. Springer Science Business Media.

Armstrong, J. S., & Collopy, F. (1992, 6). Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting*, *8*(1), 69–80. doi: 10.1016/0169-2070(92)90008-w

Bianchini, M., & Scarselli, F. (2014, 1). On the Complexity of Neural Network Classifiers: A Comparison Between Shallow and Deep Architectures. *IEEE transactions on neural networks and learning systems*, *25*(8), 1553–1565. doi: 10.1109/tnnls.2013.2293637

Bojer, C. S., & Meldgaard, J. P. (2021, 4). Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, *37*(2), 587–603. doi: 10.1016/j.ijforecast.2020.07.007

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2008). *Time Series Analysis*. Wiley.

Brannan, L. (2022, 8). Omphalos, Uber's Parallel and Language-Extensible Time Series Backtesting Tool. Retrieved from https://eng.uber.com/omphalos

Brownlee, J. (2019, 8). 4 Strategies for Multi-Step Time Series Forecasting. Retrieved from https://machinelearningmastery.com/multi-step-time-series-forecasting/

Chai, T. C., & Draxler, R. R. (2014, 6). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*(3), 1247–1250. Retrieved from https://www

.geosci-model-dev.net/7/1247/2014/gmd-7-1247-2014.pdf doi: 10.5194/gmd-7-1247-2014

Chen, C., Twycross, J., & Garibaldi, J. M. (2017, 3). A new accuracy measure based on bounded relative error for time series forecasting. *PLOS ONE*, *12*(3), e0174202. Retrieved from https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0174202&type=printable doi: 10.1371/journal.pone.0174202

Chen, W., & Shi, K. (2021, 1). Multi-scale Attention Convolutional Neural Network for time series classification. *Neural Networks*, *136*, 126–140. doi: 10.1016/j.neunet.2021.01.001

Chien, C.-F., Chen, Y., & Peng, J.-T. (2010, 12). Manufacturing intelligence for semiconductor demand forecast based on technology diffusion and product life cycle. *International Journal of Production Economics*, *128*(2), 496–509. doi: 10.1016/j.ijpe.2010.07.022

Coles, S. (2013). *An Introduction to Statistical Modeling of Extreme Values*. Springer Science Business Media.

Croston, J. D. (1972, 9). Forecasting and Stock Control for Intermittent Demands. *Operational research quarterly*, *23*(3), 289. doi: 10.2307/3007885

Dassisti, M., Giovannini, A., Merla, P., Chimienti, M., & Panetto, H. (2019, 1). An approach to support Industry 4.0 adoption in SMEs using a core-metamodel. *Annual Reviews in Control*, *47*, 266–274. doi: 10.1016/j.arcontrol.2018.11.001

Doszyń, M. (2019, 8). Intermittent demand forecasting in the enterprise: Empirical verification. *Journal of Forecasting*. doi: 10.1002/for.2575

Embrechts, P., Klüppelberg, C., & Mikosch, T. (2013). *Modelling Extremal Events*. Springer Science Business Media.

Gilliland, M., Tashman, L., & Sglavo, U. (2016). *Business Forecasting*. John Wiley Sons.

Hamilton, J. D. (2020). *Time Series Analysis*. Princeton University Press.

Han, Z., Zhao, J., Leung, H., Ma, F., & Wang, W. (2021, 3). A Review of Deep Learning Models for Time Series Prediction. *IEEE Sensors Journal*, *21*(6), 7833–7848. doi: 10.1109/jsen.2019.2923982

Hancock, J. F., & Khoshgoftaar, T. M. (2020, 12). Survey on categorical data for neural networks. *Journal of Big Data*, *7*(1). Retrieved from https://journalofbigdata.springeropen.com/track/pdf/10.1186/s40537-020-00305-w doi: 10.1186/s40537-020-00305-w

Hastie, T., Tibshirani, R., & Friedman, J. (2013). *The Elements of Statistical Learning*. Springer Science Business Media.

Hochreiter, S., & Schmidhuber, J. (1997, 11). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Huber, P. (1973, 9). Robust Regression: Asymptotics, Conjectures and Monte Carlo. *Annals of Statistics*, *1*(5). Retrieved from https://doi.org/10.1214/aos/1176342503 doi: 10.1214/aos/1176342503

Hyndman. (2006, 10). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688. doi: 10.1016/j.ijforecast.2006.03.001

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

In, Y. (2020). M5 Forecasting - Accuracy — Kaggle. Retrieved from https://www.kaggle.com/competitions/m5-forecasting-accuracy/discussion/163684

Jais, I. K. M., Ismail, A. R., & Nisa, S. Q. (2019, 6). Adam Optimization Algorithm for Wide and Deep Neural Network. *Knowledge engineering and data science*, *2*(1), 41. Retrieved from http://journal2.um.ac.id/index.php/keds/article/download/6775/3971 doi: 10.17977/um018v2i12019p41-46

Jentzen, A., Kuckuck, B., Neufeld, A., & Von Wurstemberger, P. (2021, 1). Strong error analysis for stochastic gradient descent optimization algorithms. *Ima Journal of Numerical Analysis*, *41*(1), 455–492. doi: 10.1093/imanum/drz055

Jin, X.-B., Yu, X.-H., Wang, X., Bai, Y., Su, T., & Kong, J.-L. (2020, 1). Prediction for Time Series with CNN and LSTM. *Springer Singapore eBooks*, 631–641. doi: 10.1007/978-981-15-0474-7_59

Karmarkar, A., Altay, A., Zaks, A., Ramesh, A., Mathes, B., Vasudevan, G., ... Li, Z. (2020, 09). Towards ml engineering: A brief history of tensorflow extended (tfx).

Khan, H., Wang, X., & Liu, H. (2022, 3). Handling missing data through deep convolutional neural network. *Information Sciences*, *595*, 278–293. doi: 10.1016/j.ins.2022.02.051

Kostenko, A. V., & Hyndman, R. J. (2006, 9). A note on the categorization of demand patterns. *Journal of the Operational Research Society*, *57*(10), 1256–1257. doi: 10.1057/palgrave.jors.2602211

Kourentzes, N. (2013, 5). Intermittent demand forecasts with neural networks. *International Journal of Production Economics*, *143*(1), 198–206. doi: 10.1016/j.ijpe.2013.01.009

Kourentzes, N., & Petropoulos, F. (2016, 11). Forecasting with multivariate temporal aggregation: The case of promotional modelling. *International Journal of Production Economics*, *181*, 145–153. Retrieved from https://doi.org/10.1016/j.ijpe.2015.09.011 doi: 10.1016/j.ijpe.2015.09.011

Lawrence. (1989). *Robust Regression*. CRC Press.

LeCun, Y., & Bengio, Y. (1998, 10). Convolutional networks for images, speech, and time series. *MIT Press eBooks*, 255–258. Retrieved from https://dl.acm.org/citation.cfm?id=303568.303704

Lee, H. L., Padmanabhan, V., & Whang, S. (1997, 4). Information Distortion in a Supply Chain: The Bullwhip Effect. *Management Science*, *43*(4), 546–558. doi: 10.1287/mnsc.43.4.546

Livieris, I. E., & Pintelas, P. E. (2022, 3). A novel multi-step forecasting strategy for enhancing deep learning models' performance. *Neural Computing and Applications*, *34*(22), 19453–19470. doi: 10.1007/s00521-022-07158-9

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2021a, 9). The M5 competition: Background, organization, and implementation. *International Journal of Forecasting*, *38*(4), 1325–1336. Retrieved from https://doi.org/10.1016/j.ijforecast.2021.07.007 doi: 10.1016/j.ijforecast.2021.07.007

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2021b, 11). Predicting/hypothesizing the findings of the M5 competition. *International Journal of Forecasting*, *38*(4), 1337–1345. doi: 10.1016/j.ijforecast.2021.09.014

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022, 1). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, *38*(4), 1346–1364. Retrieved from https://doi.org/10.1016/j.ijforecast.2021.11.013 doi: 10.1016/j.ijforecast.2021.11.013

Makridakis, S., Spiliotis, E., Assimakopoulos, V., Chen, Z., Gaba, A., Liu, J., & Winkler, R. (2021, 12). The M5 uncertainty competition: Results, findings and conclusions. *International Journal of Forecasting*, *38*(4), 1365–1385. Retrieved from https://doi.org/10.1016/j.ijforecast.2021.10.009 doi: 10.1016/j.ijforecast.2021.10.009

Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (2008). *FORECASTING METHODS AND APPLICATIONS, 3RD ED.* John Wiley Sons.

Nikolopoulos, K. (2021, 6). We need to talk about intermittent demand forecasting. *European Journal of Operational Research*, *291*(2), 549–559. Re-

trieved from https://www.bangor.ac.uk/business/research/documents/BBSWP-15-03.pdf doi: 10.1016/j.ejor.2019.12.046

Nikolopoulos, K., Syntetos, A., Boylan, J. E., Petropoulos, F., & Assimakopoulos, V. (2011, 3). An aggregate–disaggregate intermittent demand approach (ADIDA) to forecasting: an empirical proposition and analysis. *Journal of the Operational Research Society*, *62*(3), 544–554. doi: 10.1057/jors.2010.32

Nikolopoulos, K. I., & Thomakos, D. D. (2019). *Forecasting With The Theta Method*. John Wiley Sons.

Ord, K., & Fildes, R. (2013). *Principles of Business Forecasting*. Cengage Learning.

Osama, A. (2021). *Azure Data Engineering Cookbook*. Packt Publishing Ltd.

Paleyes, A., Urma, R.-G., & Lawrence, N. D. (2022, 4). Challenges in Deploying Machine Learning: A Survey of Case Studies. *ACM Computing Surveys*, *55*(6), 1–29. Retrieved from https://dl.acm.org/doi/pdf/10.1145/3533378 doi: 10.1145/3533378

Panigrahi, S., Karali, Y., & Behera, H. S. (2013, 9). Normalize Time Series and Forecast using Evolutionary Neural Network. *International journal of engineering research and technology*, *2*(9). Retrieved from https://www.ijert.org/research/normalize-time-series-and-forecast-using-evolutionary-neural-network-IJERTV2IS90892.pdf

Pardalos, P. M. (2020, 1). A brief history of forecasting competitions. *International Journal of Forecasting*, *36*(1), 7–14. doi: 10.1016/j.ijforecast.2019.03.015

Pascanu, R., Mikolov, T., & Bengio, Y. (2013, 6). On the difficulty of training recurrent neural networks. *International Conference on Machine Learning*, 1310–1318. Retrieved from http://proceedings.mlr.press/v28/pascanu13.pdf

Rehmer, A., & Kroll, A. (2020, 1). On the vanishing and exploding gradient problem in Gated Recurrent Units. *IFAC-PapersOnLine*, *53*(2), 1243–1248. Retrieved from

https://doi.org/10.1016/j.ifacol.2020.12.1342 doi: 10.1016/j.ifacol.2020.12.1342

Robinson, T., Eldred, M. S., Willcox, K., & Haimes, R. (2006, 5). Strategies for Multifidelity Optimization with Variable Dimensional Hierarchical Models. *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conferencelt;BRgt; 14th AIAA/ASME/AHS Adaptive Structures Conferencelt;BRgt; 7th*. doi: 10.2514/6.2006-1819

Rožanec, J. M., Fortuna, B., & Mladenic, D. (2022, 7). Reframing Demand Forecasting: A Two-Fold Approach for Lumpy and Intermittent Demand. *Sustainability*, *14*(15), 9295. Retrieved from https://www.mdpi.com/2071-1050/14/15/9295/pdf?version=1659082931 doi: 10.3390/su14159295

Russac, Y., Caelen, O., & He-Guelton, L. (2018, 2). Embeddings of Categorical Variables for Sequential Data in Fraud Context. *Springer International Publishing eBooks*, 542–552. doi: 10.1007/978-3-319-74690-6_53

Said, A. S., Erradi, A., Aly, H. H., & Mohamed, A.-M. S. (2021, 5). Predicting COVID-19 cases using bidirectional LSTM on multivariate time series. *Environmental Science and Pollution Research*, *28*(40), 56043–56052. Retrieved from https://link.springer.com/content/pdf/10.1007/s11356-021-14286-7.pdf doi: 10.1007/s11356-021-14286-7

Semenoglou, A.-A., Spiliotis, E., Makridakis, S., & Assimakopoulos, V. (2021, 7). Investigating the accuracy of cross-learning time series forecasting methods. *International Journal of Forecasting*, *37*(3), 1072–1084. doi: 10.1016/j.ijforecast.2020.11.009

Shumway, R. H., & Stoffer, D. S. (2017). *Time Series Analysis and Its Applications*. Springer.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018, 12). A Comparison of ARIMA and LSTM in Forecasting Time Series. *International Conference on Machine Learning and Applications*. doi: 10.1109/icmla.2018.00227

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019, 12). The Performance of LSTM and BiLSTM in Forecasting Time Series. *International Conference on Big Data*. doi: 10.1109/bigdata47090.2019.9005997

Singh, D., & Singh, B. (2020, 12). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524. doi: 10.1016/j.asoc.2019.105524

Sinsomboonthong, S. (2022, 4). Performance Comparison of New Adjusted Min-Max with Decimal Scaling and Statistical Column Normalization Methods for Artificial Neural Network Classification. *International Journal of Mathematics and Mathematical Sciences*, *2022*, 1–9. Retrieved from https://downloads.hindawi.com/journals/ijmms/2022/3584406.pdf doi: 10.1155/2022/3584406

Spithourakis, G. P., Petropoulos, F., Nikolopoulos, K., & Assimakopoulos, V. (2014, 4). A systemic view of the ADIDA framework. *Ima Journal of Management Mathematics*, *25*(2), 125–137. doi: 10.1093/imaman/dps031

Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S. S. S., & Babu, R. V. (2016, 1). A Taxonomy of Deep Convolutional Neural Nets for Computer Vision. *Frontiers in Robotics and AI*, *2*. Retrieved from https://www.frontiersin.org/articles/10.3389/frobt.2015.00036/pdf doi: 10.3389/frobt.2015.00036

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014, 1). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958. Retrieved from https://jmlr.csail.mit.edu/papers/volume15/srivastava14a/srivastava14a.pdf

Sugiyama, M., & Kawanabe, M. (2012). *Machine Learning in Non-stationary Environments*. MIT Press.

Syntetos, A., & Boylan, J. E. (2005, 4). The accuracy of intermittent demand estimates. *International Journal of Forecasting*, *21*(2), 303–314. doi: 10.1016/j.ijforecast.2004.10.001

Syntetos, A., & Boylan, J. E. (2006, 9). On the stock control performance of intermittent demand estimators. *International Journal of Production Economics*, *103*(1), 36–47. doi: 10.1016/j.ijpe.2005.04.004

Tashman, L. J. (2000, 10). Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting*, *16*(4), 437–450. doi: 10.1016/s0169-2070(00)00065-0

Theodorou, E. A., Wang, S., Kang, Y., Spiliotis, E., Makridakis, S., & Assimakopoulos, V. (2021, 9). Exploring the representativeness of the M5 competition data. *International Journal of Forecasting*, *38*(4), 1500–1506. doi: 10.1016/j.ijforecast.2021.07.006

Tripathi, S., Muhr, D., Brunner, M., Jodlbauer, H., Dehmer, M., & Emmert-Streib, F. (2021, 6). Ensuring the Robustness and Reliability of Data-Driven Knowledge Discovery Models in Production and Manufacturing. *Frontiers in artificial intelligence*, *4*. Retrieved from https://www.frontiersin.org/articles/10.3389/frai.2021.576892/pdf doi: 10.3389/frai.2021.576892

Vilone, G., & Longo, L. (2021, 8). Classification of Explainable Artificial Intelligence Methods through Their Output Formats. *Machine learning and knowledge extraction*, *3*(3), 615–661. Retrieved from https://www.mdpi.com/2504-4990/3/3/32/pdf?version=1628601193 doi: 10.3390/make3030032

Wallström, P., & Segerstedt, A. (2010, 12). Evaluation of forecasting error measurements and techniques for intermittent demand. *International Journal of Production Economics*, *128*(2), 625–636. doi: 10.1016/j.ijpe.2010.07.013

Willemain, T. R., Smart, C. K., & Schwarz, H. G. (2004, 7). A new approach to forecasting intermittent demand for service parts inventories. *International Journal of Forecasting*, *20*(3), 375–387. doi: 10.1016/s0169-2070(03)00013-x

Xue, N., Triguero, I., Figueredo, G. P., & Landa-Silva, D. (2019, 6). Evolving Deep CNN-LSTMs for Inventory Time Series Prediction. *Congress on Evolutionary Computation*. doi: 10.1109/cec.2019.8789957

Yamak, P. T., Yujian, L., & Gadosey, P. K. (2019, 12). A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting. *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*. Retrieved from https://doi.org/10.1145/3377713.3377722 doi: 10.1145/3377713.3377722

Yao, L., C, P. E., Dagunts, D., Covington, B., Bernard, D., & Lyman, K. (2017, 10). Learning to diagnose from scratch by exploiting dependencies among labels. *arXiv: Computer Vision and Pattern Recognition*. Retrieved from https://arxiv.org/pdf/1710.10501.pdf

Yu, T., & Zhu, H. (2020, 3). Hyper-Parameter Optimization: A Review of Algorithms and Applications. *arXiv: Learning*. Retrieved from https://arxiv.org/pdf/2003.05689

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., & Eickhoff, C. (2021, 8). A Transformer-based Framework for Multivariate Time Series Representation Learning. *Knowledge Discovery and Data Mining*. doi: 10.1145/3447548.3467401

Zhang, G.-Q., Patuwo, B. E., & Hu, M. S. (1998, 3). Forecasting with artificial neural networks:. *International Journal of Forecasting*, *14*(1), 35–62. doi: 10.1016/s0169-2070(97)00044-7

Zhang, J., & Man, K. (1998, 10). Time series prediction using RNN in multi-dimension embedding phase space. *Systems, Man and Cybernetics*. doi: 10.1109/icsmc.1998.728168

Zhang, T., Song, S., Li, S., Ma, L., Pan, S., & Han, L. (2019, 1). Research on Gas Concentration Prediction Models Based on LSTM Multidimensional Time Series. *Energies*, *12*(1), 161. Retrieved from https://www.mdpi.com/1996-1073/12/1/161/pdf?version=1546571570 doi: 10.3390/en12010161

Çakanyildirim, M., & Roundy, R. O. (2002, 5). SeDFAM: semiconductor demand forecast accuracy model. *Iie Transactions*, *34*(5), 449–465. doi: 10.1080/07408170208928882