Investigating Accuracy and Complexity Trade-offs for Multiple Time Series Forecasting
Team 01

Jesher Appels (12865362), Palash Kanwar (13837494), Francisco Pereira (13911376), Leyla Rahimi (13648063)
Applied Forecasting in Complex Systems 2021
University of Amsterdam

ABSTRACT

Forecasting product demand is a challenging concept due to the change and irregularities in demand. This paper describes the methods used to forecast Walmart's number of sold products. The data comes from the M5 Kaggle competition and includes various time series. The aim is to investigate the accuracy and complexity trade-offs between simple and advanced models in forecasting multiple times series. We investigated three types of models: Baseline (Mean, Seasonal Naïve), Advanced (ETS, ARIMA, Prophet), and Deep Learning (DeepAR). The results show that the deep learning models performed best with an RMSE of 3.359 with less than one hour of processing time. This was 8.8 percent better than the second-best performing model, which was the mean with a score of 3.685. While the mean was computationally inexpensive, DeepAR captured more hidden relationships, and structural patterns across multiple time series.

1 INTRODUCTION

Walmart is an American multinational retail corporation that operates a chain of hypermarkets, discount department stores, and grocery stores from the United States. The company was founded in 1962, and according to the Fortune Global 500 list of 2020, it is the world's largest company by revenue. Walmart currently operates more than 11,000 stores with approximately 240 million customer visits per week [1]. Walmart is the front-runner in terms of product variety, providing customers with 75 million products, out of which 60 million items are currently listed for sale [2]. For a retail company of this magnitude, having the right amount of products in stock is a core challenge. A good forecast makes sure there are enough customers' favorite products in stock, at all times.

The demand for a product keeps changing from time to time. Sales forecasting refers to the process of estimating demand for or sales of a particular product over a specific period of time. Accurate sales forecasts enable companies to make informed business decisions and predict short-term and long-term performance. The sales forecast has a tremendous impact on many business decisions. It allows companies to predict achievable sales revenue, efficiently allocate resources and plan for future growth [3]. For this Kaggle competition, we are required to investigate the hierarchical sales data from a Walmart store in the State of Texas in the United States. The data includes explanatory variables such as price, promotions, day of the week, and special events that occur on the given day of sale for a particular product. In order to forecast the sales of 823 products for the next 28 days, we are given the sales of each product for the past 19,412 days (5.4 years) with a date range from 29-01-2011 to 19-06-2016.

There are three different strategies to validate the sales forecast – qualitative techniques, time series analysis and projections, and causal models. The first uses qualitative data, for example, expert opinion and information about special events of the kind already mentioned, and may or may not take the past into consideration. The second, on the other hand, focuses entirely on patterns and pattern changes, and thus relies entirely on historical data. The third uses highly refined and specific information about relationships between system elements and is powerful enough to take special events formally into account. As with time series analysis and projection techniques, the past is important to causal models [4].

In order to address this challenge of forecasting sales of products at Walmart's Texas store, we intend to build a hybrid model that combines time series analysis and projections and causal models. We build and test 3 types of models to investigate the research question – "To what extent are advanced/deep learning models better in accuracy and complexity while forecasting multiple times series (with varying characteristics) compared to simple models?". In particular, the three types of models we investigate are – Baseline (Mean, Seasonal Naïve), Advanced (ETS, ARIMA, Prophet), and Deep Learning (Deep AR). Our hypothesis is that deep learning models are able to capture similarities between time series and hidden relations with related yet independent data. While the complexity increases, an accurate forecast of the product sales, with a holistic understanding of trends and the special events such as holidays, sports games, marketing promotions, etc. occurring at the time of sale [5] is attainable. This reports presents an Exploratory Data Analysis, Methodology including characteristics, and limitations of different models when applied to the Walmart product sales time series, with Results addressing the hypothesis followed by the Discussion and Conclusion.

2 EXPLORATORY DATA ANALYSIS

The first step in any time series analysis is to perform an exploratory data analysis. The data for this project comes from the 2020 M5 Kaggle competition and contains a subset of an extensive data set. The data used for this project consists of the number of food commodities sold per day in Walmart stores in the State of Texas [6]. The data set consists of three parts, namely:

- A train set with a total of 823-time series, each having 1913 data points. The series range between 2011-01-29 and 24-04-2016.
- A test set containing 823-time series but this time, the series has only 28 days of data points which range between 2016-04-25 and 2021-05-22.
- A calendar data set containing information about the events on a specific date.

It is crucial to understand the data characteristics and patterns before creating all sorts of different models. Plotting the different individual series is not realistic due to a large number of data points. However, figure 1 shows a subset of the data, and it immediately becomes clear that there is diversity between the different time series. For example, item Foods-3-348 is being sold in high numbers throughout the time span compared to the rest. On the other hand, Item Foods-3-438 starts selling later in the time span and in smaller quantities (the number of items sold is zero for the first few months). Item Foods-3-191 shows another pattern with high selling numbers early in the time span and later stabilizing at a lower level.

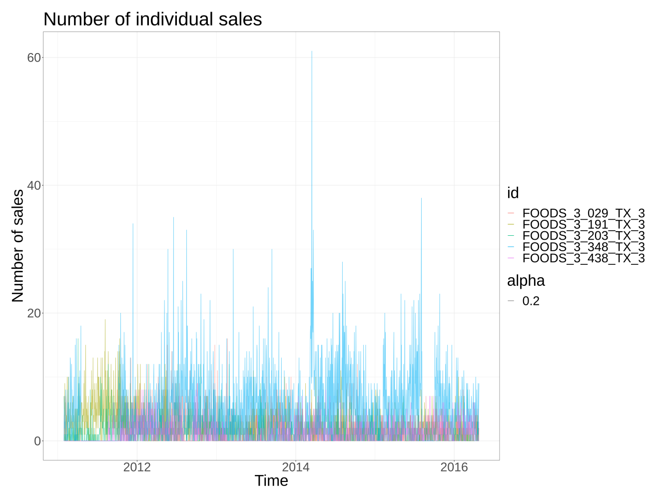

Figure 1: A subset of ten random time series plotted over each other.

The chaotic picture of the individual time series changes when the data is grouped. Figure 2 shows the total number of products sold in a day, together with a moving average. The graph shows a clear seasonal pattern, with yearly peaks in the third quarter. There is no trend since the highs and lows remain more or less the same over the years. There is, however, a jump from the first half of 2011 to the second half of 2011 and beyond. While the precise reason for this is difficult to ascertain with the current data, one reason could be that the number of stores has increased, but this cannot be said with certainty.

Furthermore, the graph shows a peak on 2015-06-15. This is one day before the NBA finals of that year and could be a possible cause for the increase in sales. However, the finals were between a team from Ohio and California, so there does not seem to be a direct correlation between product sales in Texas and that year's finals. The number of products sold in previous years does not show the same pattern either.

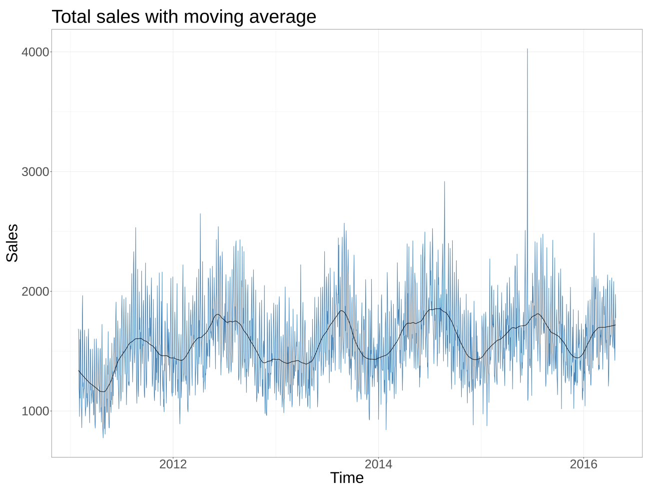

Figure 2: The total number of sales.

One can further explore the data by creating a heatmap of the average sales with the months on the x-axis and the day on the y-axis as in Figure 3 [7]. The heatmap has two primary apparent characteristics. The first one relates to the average number of sales being higher during the weekend. This pattern results in the data not just having a yearly seasonality, but also a weekly pattern, with the number of sales being the highest during the weekend. Second, the number of products sold on Tuesdays in June is very low, and this then seems to increase the day after. The calendar data shows some NBA Finals events taking place around this period in time, but only two of them (one in 2015 and one in 2012) happen on this exact day, which means that one can assume that these events cause the pattern.

The more advanced models (Prophet and DeepAR) can include other parameters for their time series predictions, and one of these parameters might be the event data. Figure 4 shows the mean number of sales per event type. The graph shows that the mean number of sales is the highest for sporting events, which include the NBA final and Superbowl. National events have the lowest mean number of sales, which might be due to stores not being open on these days.

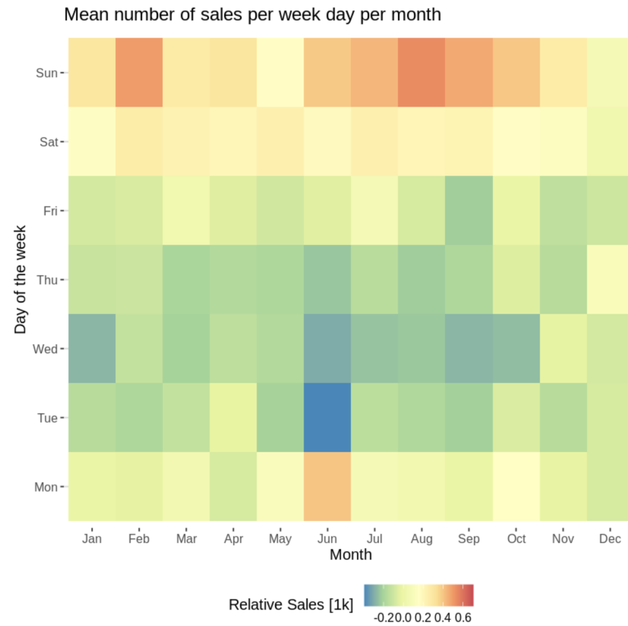

Figure 3: Mean number of sales per week day per month. The graph shows a clear weekly pattern

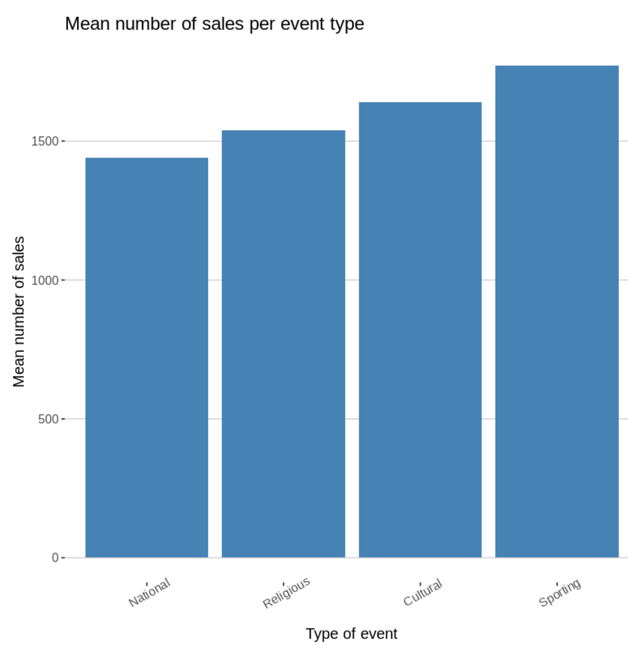

Figure 4: Mean number of sales per event type

3 METHODOLOGY

The next phase of the analysis involves the pre-processing of the data before sending through the model. The first step was to merge the train data with the calendar data set. It was necessary to merge these two data sets since it created a time series with actual dates. The same procedure was followed for the test set. In order to address the inconsistency of product sales across different products, a column for the cumulative sum of product sales is added to the training set.

The cumulative data takes into account the late start, early stop and high variance of product sales for each product. This would potentially allow the advanced models to capture changes in the sales performance of a product and particularly detect changes in the sales mean for each product [8]. Additionally, as elaborated in the EDA, there might be some hidden relations between special events and product sales. The special events can be regarded as categorical variables which were one hot encoded into the training data. The one hot encoding indicates the occurrence of a specific special event such as holidays, sports matches, etc. on the day of the product sale.

The aim of this report is to compare the three different types of models - Baseline (Mean, Seasonal Naive), Advanced (ETS, ARIMA, Prophet) and Deep Learning (DeepAR) based on their accuracy measures and complexity trade-offs. The models generate a 28 day sales forecast for each product. The accuracy of models are compared over their respective Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Scaled Error (MASE). The models ensure that the time series is back transformed where applicable before accuracy comparison. The evaluation of complexity is done by comparing processing times and computational costs for each model, and labelled as - 'High', 'Medium' or 'Low'.

Baseline Models

After the data exploration and wrangling of the time series, we decided to start forecasting using simple methods. These baseline methods are the Mean and the Seasonal Naïve. In the following sections, these baseline methods are compared to the more advanced models to determine whether advanced/complex models result in higher accuracy for forecasting the time series.

Mean. The mean is considered a simple yet effective baseline model [9]. The forecast of all future values is set equal to the average of that particular time series. The forecast formula can be written as follows:

$$\hat{y}_{T+h|T} = \bar{y} = (y_1 + \cdots + y_T)/T. \tag{1}$$

The implementation of the mean method is computationally inexpensive since all future values are set to the average value. However, there is a significant limitation to this method because, during the EDA phase, we have seen that some products were taken off the shelves or were introduced in stores much later. This leads certain forecasts to deviate more than some other time series.

Seasonal Naïve. The seasonal naïve is another baseline method that was used for forecasting. If the data shows seasonality related to the day of the week, the seasonal naïve method is considered useful for forecasting this type of data [9]. Since the time series illustrates varying seasonality, this method is an appropriate method to use as a baseline model. The forecasts are estimated with the use of the following formula:

$$\hat{y}_{T+h|T} = y_T + h - m(k+1). \tag{2}$$

The seasonal naïve sets future values equal to the last observed value from the same season. Therefore, the estimated forecasts are taken over from the previous period without any adjustments, which however can be a limiting factor because not all time series are indicative of a seasonal pattern.

Both baseline methods do not establish causal factors and are not flexible since they give a point forecast and therefore do not deal with uncertainty. However, the forecasts obtained by baseline models can be used to compare the estimates generated by more complex techniques, which are known to be more refined.

Advanced Models

Previously, we have seen the baseline models considered. These are simple models that serve as benchmarks throughout this work. This section refers to more advanced models used to forecast the data. ETS and ARIMA are commonly used forecast methods. While ETS attempts forecast based mostly on trend and seasonality, ARIMA explores existing auto-correlation in the data. The Prophet model is also considered in this section. Prophet is an additive regression model and will be explained further in detail.

ETS. ETS is one of the widely used methods [10]. The ETS framework uses weighted averages of past observations to create forecasts. Recent observations will have higher weights, with the weights exponentially decaying the further into the past the observations get. The ETS method looks at each time series's different components and determines which ETS model would be more suitable. The various components in an ETS model are: - Level which represents the average value of the time series - Trend is the increasing or decreasing value in the time series - Seasonality is the repeating seasons in the time series. Each component can be additive, multiplicative, or non-existing. Each of the possibilities refers to the linkages between the different elements. Furthermore, the ETS framework has various assumptions that should hold to construct proper models. Assumptions that need to hold are that the residuals should have homoscedasticity, be independent and follow a normal distribution with a mean zero.

The implementation of the ETS model is done automatically. The ETS model estimates the parameters and chooses the best model based on the corrected Akaike's Information Criterion (AICc) [9]. Since each time series has varying levels, trends, and seasonality, it is best to let the ETS model determine which model is appropriate. One limitation of this method is that we need to inspect the assumptions to determine whether the model captures all information. Since the data set consists of many time series, it is impossible to check whether the assumptions are held for every time series.

ARIMA. The Auto Regressive Integrated Moving Average (ARIMA) model is a class of models that explains a given times series based on the past values of the series, its lags, and its previous lagged forecast errors. ARIMA models are essentially regarded as agnostic, as they do not assume knowledge of any underlying economic model, hidden relations in the data set, or structural relationships [11]. However, it is highly advantageous when forecasting a large number of time series. Additionally, if the data set consists of multiple time series existing over different time periods with varied lags, then forecasts using this model are conditional forecasts based on forecasts of the unavailable observations, adding an additional source of forecast uncertainty.

An ARIMA model is characterized by three terms – 'p', 'd', and 'q'. The Auto Regressive term is represented by 'p' and refers to the number of lags of the time series to be used as predictors. The 'q' is the order of the Moving Average term which refers to the number of lagged forecast errors that should be incorporated into the ARIMA model [12]. Lastly, the value 'd' denotes the minimum number of differencing needed to make the series stationary. Given that we are trying to forecast 823 time series with varying characteristics, it is appropriate to fit the ARIMA model using the Auto Arima function. Instead of manually fitting different models and deciding which one is the best for each time series. The Auto Arima function automates the process by fitting ARIMA models with different order terms for each time series and outputs the most appropriate fitted forecasts for the particular time series [13].

While the best fit for each time series is obtained using the lower AICc score, increasing the complexity of the model substantially increases processing times [14]. Given the nature of the data set, it is also important to realize that ARIMA models are poor at predicting turning points unless the turning point represents a return to long-run equilibrium. The lack of embedded structural relationships or underlying similarities between certain products at sale are not taken into account in ARIMA models. This urges us to investigate other advanced models.

Prophet. Prophet is a time series forecasting model proposed by Facebook in 2018 [15]. It is an additive model combining trend, seasonality, holiday effects and an error term.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{3}$$

$g(t)$ corresponds to the trend component; it can be modeled as a piece-wise linear trend or a logistic trend model. In this case the piece-wise linear trend was used.

The seasonality $s(t)$ is modeled through Fourier series:

$$s(t) = X(t)\beta \tag{4}$$

With $X(t)$ and β:

$$X(t) = [cos(\frac{2\pi 1t}{P}), sin(\frac{2\pi 1t}{P}), \ldots, cos(\frac{2\pi Nt}{P}), sin(\frac{2\pi Nt}{P})] \tag{5}$$

$$\beta = [a_1, b_1, \ldots, a_N, b_N] \tag{6}$$

Where N is the order of the Fourier series and P is the period. In this case, weekly seasonality was considered with $P = 7$ and $N = 3$.

Holidays effects are modeled by $h(t)$.

$$h(t) = Z(t)k \tag{7}$$

$Z(t)$ is a one hot vector encoding the holiday occurrences and k is the corresponding change in the forecast.

The data from demand forecasting depicts complex patterns. Some weekly seasonality is observed as well as yearly patterns. Prophet performs better when the data presents historical trend changes or non-linear growth trends that have a ceiling point; this is because the model was primarily developed to deal with business challenges that Facebook faces [16]. However, in demand forecasting, data is intermittent, with bursts of demand on one day and none on the other. Prophet does not handle this data type very well. To tackle this problem, a cumulative sum was used (transformation seen in figure 5).

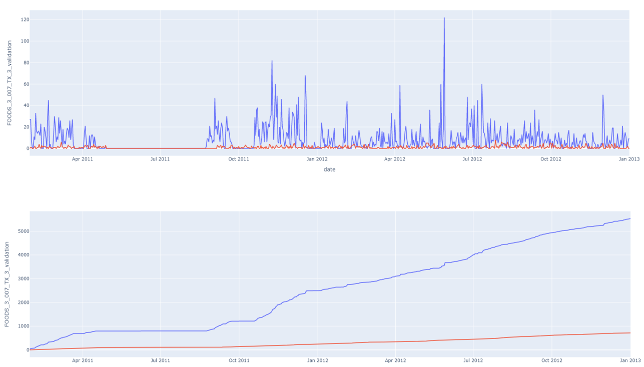

Figure 5: Transformation of the data using a cumulative sum - used for the Prophet model.

Prophet also allows to account for holidays and special events. This feature is also explored, as the *calendar.csv* file contains information about events that occurred but also the ones occurring on the 28 days being forecasted. This information is introduced in the model and its impact is explored in the results section.

Deep Learning in Forecasting

The typical forecasting methods consist of fitting a model into a single time series, from which future values can then be inferred. However, in a lot of applications (like the one presented throughout this work) there is the need to forecast a large amount of time series (for each product). Models such as ETS, ARIMA and Prophet turn out to be computationally expensive as they can only be used sequentially. Furthermore, it was seen that the data exhibits patterns that violate core assumptions of the classical models. Deep learning presents itself as a good alternative given the properties of the problem.

DeepAR. DeepAR [17] is a probabilistic forecast method that applies auto-regressive recurrent networks to a group of time series. By training one model on the grouped time series, generating forecast for all time series is less time-consuming, has less need for manual feature engineering needed to capture certain behaviours and also captures similarities between common product time series. This feature can be useful for products that were introduced later and do not have enough data to fit models like ETS, ARIMA or Prophet. An open-source implementation of this model is available in the PyTorch Forecasting package [18].

To facilitate learning time-dependent patterns, such as spikes during weekends, DeepAR automatically creates feature time series based on the frequency of the target time series. For example, DeepAR creates two feature time series (day of the month and day of the year) for a weekly time series frequency [19]. To capture seasonality patterns, DeepAR also automatically feeds lagged values and trains a model by randomly sampling several training examples from each of the time series in the training dataset. Instead of creating a model for each time series, DeepAR creates a single model for all the time series.

In the DeepAR model, it is possible to associate each time series with categorical features [20]. This is useful for the model to learn common behaviours between time series that share the same features. We extend this to incorporate the information from the calendar events. To explain this with an example, the Super Bowl can be considered; around this date it is expected that people consume more of certain products, and this information might help the model better react to the increases in demand.

4 RESULTS

In the previous sections the models used in this work, along with their characteristics were presented. Here, the different results and the best performing model are discussed. The models are compared by type (Baseline/Advanced/Deep Learning) and evaluated on the test set (*sales_test_validation_afcs2021.csv*) and the Kaggle competition.

Table 1, presents the results according to the RMSE, MAE and MASE metrics measured on the test set. For Prophet and DeepAR it was not possible to calculate the MASE, as these were implemented in Python and, to the best of our knowledge, required the function to be developed from scratch. As this was not the main scope of this project and due to time limitation constraints, we decided to only compare the MASE score for the Mean, Seasonal Naive, ETS and ARIMA. Table 2 contains the results from the Kaggle submissions. Table 2 also describes processing time for each model applied in the time series. 'Low' for less than 15 minutes, 'Medium' for less than one hour (60 minutes) and 'High' for any processing time above one hour (60 minutes).

Table 1: Results of the different models on the validation set given.

Model	RMSE	MAE	MASE
Mean	2.113	1.659	1.540
Seasonal Naïve	2.322	1.697	1.724
ETS	1.893	1.458	1.516
Arima	1.826	1.399	1.496
Prophet	2.024	1.492	-
Prophet *Events	2.126	1.557	-
DeepAR	3.101	1.445	-
DeepAR *Events	3.036	1.456	-

Table 2: Results of the different models on the Kaggle competition.

Model	RMSE	Processing Time
Mean	3.685	Low
Seasonal Naïve	8.151	Low
ETS	4.059	High
Arima	3.800	High
Prophet	3.958	High
Prophet *Events	4.067	High
DeepAR	3.394	Medium
DeepAR *Events	**3.359**	Medium

We first explore the baseline models. The Mean model performed better that the Seasonal Naïve method. This was expected, as the Seasonal Naïve model is being applied on inconsistent seasonality patterns. The Mean model, on the other hand, is averaging over past values to predict the future. For sales forecasting this can offer moderate results, despite the model being incapable of dealing with changes in demand. We define our baseline model as the Mean, and all future models are compared against the accuracy scores of the Mean and the RMSE score obtained from kaggle which is 3.685.

While comparing advanced models, the ARIMA outperforms the ETS model on the validation provided based on the RMSE, MAE, and MASE scores. The ARIMA model also performed better than the ETS model according to the RMSE obtained in the Kaggle competition. Prophet did not yield lower errors than the previous two models in any of the evaluations. Furthermore, it was expected that including meta-data such as events would allow the model to adapt to possible increases in demand given certain event such as Sports (NBA finals or Super Bowl); this was, however, not the case, and the RMSE and MAE increased slightly when including this type of data. As mentioned in Advanced Models' section, the data in the Prophet model was transformed using a cumulative sum. The forecasts were evaluated on the back-transformed data, obtained by taking the difference between t and $t-1$. Despite the increased complexity of these models, the results on the Kaggle competition did not outperform our base model (Mean). Moreover, one of the main challenges encountered when fitting these models on the 823 time series was the expensive processing time. Fitting one model per series proved to be unscalable as the process is very time-consuming.

Lastly we explore the Deep Learning model: DeepAR. As discussed previously when the describing the model, this approach has many advantages when it comes to scalability. In the training process of this models the 823 series were considered in parallel. Categorical data such as events were also considered when experimenting with DeepAR. In this case, event data was helpful on achieving a lower error score. Analysing table 1 it is seen that the DeepAR RMSE errors are the highest, however, in the Kaggle competition it was the best performing model. This might indicate that models such as ETS, ARIMA and Prophet are overfitting the data.

Overall the DeepAR model proved to have a good trade-off between error and processing time. It performed significantly better than our baseline model, the Mean, and the processing time was reasonable as it was possible to explore parallel computation through a Graphical Processing Unit (GPU).

5 DISCUSSION

According to Amazon's time series forecasting principles, forecasting is a hard problem for two reasons. First, incorporating large volumes of historical data, which can lead to missing important information about the past of the target data dynamics. Second, incorporating related yet independent data (holidays/events, locations, marketing promotions). Besides these, one of the central aspects of sales forecasting is that accuracy is key. If the forecast is too high it may lead to over-investing and therefore losing money. If the forecast is too low it may lead to under-investing and therefore losing opportunity [21].

The data set was challenging to work with since it includes many time series, each with different properties. Some time series do not show seasonality, whereas others do, even if it is just a slight pattern. This data property made it challenging to select an appropriate model that would fit the data as a whole. Even though this was our approach, the accuracy measures could have been improved by iterating different models over the data set and saving the best-performing model for each time series. For example, this could lead to the mean being the most suitable model for some time series and the DeepAR for others.

Another challenge regarding the data was the property that some products are not or hardly sold at certain times, which affects the models as a result. For example, we saw that the original sales did not lead to a good forecast with the prophet model. In order to address this problem the same model was provided with cumulative sales of products over time yielding in far better results.

6 CONCLUSION

We evaluated three types of models and their performance on a subset of the M5 Kaggle competition data set. The three types of models we investigated are – Baseline (Mean, Seasonal Naïve), Advanced (ETS, ARIMA, Prophet), and Deep Learning (DeepAR). The accuracy metrics utilized were the RMSE, MASE, and MAE, and the overall best performing model was the DeepAR with 'Medium' processing time. The deep learning model generates accurate forecasts with less than an hour of processing time. As hypothesized, it captures similarities between time series and hidden relations with related yet independent data. Deep learning enables the development of sophisticated, customized forecasting models that incorporate unstructured retail data sets, therefore it makes sense to use them when the data is complicated [21].

Traditional methods can only account for the dynamics of one-dimensional data they are trained on. Generality and flexibility seem to be the key factors that permeate successful sales forecasting models. Our investigation points to a future of hybrid models where multiple time series can be accounted for and categorical variables can be included in the forecasting pipeline. Some other approaches to sales forecasting include leveraging Natural Language Processing models with entity embeddings and language of product descriptions. Additionally, WaveNet – a fully convolutional neural network model – seems to hint at a trend of using generative models with deep learning allowing for predictions of likelihoods of different scenarios.

Keeping the qualitative and quantitative aspects in mind, as part of further research, Walmart could potentially experiment with other sales forecasting techniques. One is to review the sales quotas for each salesperson and weigh their performance based on historical results. Another method is the rule of three. Walmart's sales force will likely fall into three categories; a third will over-perform, a third will just meet the expectation and a third will under-perform. Another method is to look at historical growth patterns and extrapolate forward or to look at the growth in the industry that is applicable and assume the company will grow at the same rate [22].

REFERENCES

[1] Statista. *Topic: Walmart.* Dec. 2020. URL: https://www.statista.com/topics/1451/walmart/.

[2] JungleTopp. *25 Walmart Stats You Need to Know in 2020.* Dec. 2020. URL: https://jungletopp.com/walmart-stats/.

[3] Skyword. *Sales Forecasting.* Dec. 2020. URL: https://www.skyword.com/marketing-dictionary/sales-forecasting/.

[4] Harvard Business Review. *How to Choose the Right Forecasting Technique.* Dec. 2021. URL: https://hbr.org/1971/07/how-to-choose-the-right-forecasting-technique.

[5] Medium. *Sales Forecasting: from Traditional Time Series to Modern Deep Learning*. Dec. 2021. URL: https://towardsdatascience.com/sales-forecasting-from-time-series-to-deep-learning-5d115514bfac.

[6] MOFC. *The M5 Competition*. Dec. 2020. URL: https://mofc.unic.ac.cy/m5-competition/.

[7] Jaswanth Badvelu. *M5 Forecasting Accuracy*. Dec. 2020. URL: https://towardsdatascience.com/m5-forecasting-accuracy-24d7f42130de.

[8] JSTOR. *The Use of Cumulative Sum (Cusum) Techniques for ... - JSTOR*. Dec. 1963. URL: https://www.jstor.org/stable/167932.

[9] Robin John Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. English. 2nd. Australia: OTexts, 2018.

[10] Ivan Svetunkov. *Forecasting and Analytics with ADAM*. OpenForecast. (version: [current date]). 2021. URL: https://openforecast.org/adam/.

[11] Munich Personal RePEc Archive. *3/RT/98 - Forecasting Irish Inflation Using ARIMA Models*. Dec. 2021. URL: https://mpra.ub.uni-muenchen.de/11359/1/cbi_3RT98_inflationarima.pdf.

[12] Machine Learning Plus. *ARIMA Model - Complete Guide to Time Series Forecasting in Python: ML+*. Dec. 2021. URL: https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/.

[13] Medium. *What is Auto-ARIMA?* Dec. 2021. URL: https://medium.com/featurepreneur/what-is-auto-arima-b8025c6d732d.

[14] 8.10 ARIMA vs ETS. *Forecasting: Principles and Practice (2nd ed)*. Dec. 2021. URL: https://otexts.com/fpp2/arima-ets.html.

[15] Sean J Taylor and Benjamin Letham. "Forecasting at scale". In: *The American Statistician* 72.1 (2018), pp. 37–45.

[16] *Prophet: forecasting at scale*. 2017. URL: https://research.fb.com/blog/2017/02/prophet-forecasting-at-scale/.

[17] David Salinas et al. "DeepAR: Probabilistic forecasting with autoregressive recurrent networks". In: *International Journal of Forecasting* 36.3 (2020), pp. 1181–1191.

[18] *Prophet Forecasting: Documentation*. 2020. URL: https://pytorch-forecasting.readthedocs.io/en/latest/#.

[19] Amazon. *Machine Learning for Business: using Amazon SageMaker and Jupyter*. Dec. 2021. URL: https://docs.aws.amazon.com/sagemaker/latest/dg/deepar_how-it-works.html.

[20] *DeepAR Forecasting Algorithm: Documentation*. 2021. URL: https://docs.aws.amazon.com/sagemaker/latest/dg/deepar_how-it-works.html.

[21] Medium. *Sales Forecasting: from Traditional Time Series to Modern Deep Learning*. Dec. 2021. URL: https://towardsdatascience.com/sales-forecasting-from-time-series-to-deep-learning-5d115514bfac.

[22] TGG Accounting. *Importance of Sales Forecasting*. Dec. 2021. URL: https://tgg-accounting.com/importance-of-sales-forecasting-2/.