*applied system innovation*

MDPI

*Article*

# Robust Sales forecasting Using Deep Learning with Static and Dynamic Covariates

**Patrícia Ramos [1,2,*] and José Manuel Oliveira [2,3]**

[1] CEOS.PP, ISCAP, Polytechnic of Porto, Rua Jaime Lopes Amorim, 4465-004 S. Mamede de Infesta, Portugal
[2] INESC TEC, Campus da FEUP, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
[3] Faculty of Economics, University of Porto, Rua Dr. Roberto Frias, 4200-464 Porto, Portugal; jmo@fep.up.pt
* Correspondence: patricia@iscap.ipp.pt

**Abstract:** Retailers must have accurate sales forecasts to efficiently and effectively operate their businesses and remain competitive in the marketplace. Global forecasting models like RNNs can be a powerful tool for forecasting in retail settings, where multiple time series are often interrelated and influenced by a variety of external factors. By including covariates in a forecasting model, we can often better capture the various factors that can influence sales in a retail setting. This can help improve the accuracy of our forecasts and enable better decision making for inventory management, purchasing, and other operational decisions. In this study, we investigate how the accuracy of global forecasting models is affected by the inclusion of different potential demand covariates. To ensure the significance of the study's findings, we used the M5 forecasting competition's openly accessible and well-established dataset. The results obtained from DeepAR models trained on different combinations of features indicate that the inclusion of time-, event-, and ID-related features consistently enhances the forecast accuracy. The optimal performance is attained when all these covariates are employed together, leading to a 1.8% improvement in RMSSE and a 6.5% improvement in MASE compared to the baseline model without features. It is noteworthy that all DeepAR models, both with and without covariates, exhibit a significantly superior forecasting performance in comparison to the seasonal naïve benchmark.

**Keywords:** deep neural networks; time series forecasting; covariates; retailing



**Citation:** Ramos, P.; Oliveira, J.M. Robust Sales forecasting Using Deep Learning with Static and Dynamic Covariates. *Appl. Syst. Innov.* **2023**, *6*, 85. https://doi.org/10.3390/asi6050085

Academic Editor: Georgios Th Papadopoulos

Received: 25 July 2023
Revised: 8 September 2023
Accepted: 25 September 2023
Published: 28 September 2023



## 1. Introduction

Accurate sales forecasts are of paramount importance to retailers as they heavily depend on them to effectively manage their supply chains and make crucial decisions related to marketing, logistics, finance, and human resources [1]. Accurate sales forecasts help retailers determine how much inventory they need to purchase from their suppliers in order to meet customer demand. If the forecast is too low, they risk running out of stock and losing sales. If the forecast is too high, they risk overstocking and tying up cash in excess inventory. Sales forecasts also help retailers plan their logistics operations, such as determining how much warehouse space they need, how many trucks they need to transport goods, and how much labor is required to handle incoming and outgoing shipments. They also help retailers plan their marketing campaigns, such as determining which products to promote, which channels to use, and how much to spend on advertising. By having a clear understanding of expected sales volumes, retailers can more effectively allocate their marketing budgets. Accurate sales forecasts are also important for financial planning, such as budgeting and forecasting cash flow. Retailers need to know how much revenue they can expect in order to plan for expenses, investments, and debt repayment. Finally, sales forecasts are used by retailers to plan their staffing needs. They need to know how many employees they will need in their stores and warehouses in order to meet customer demand, and how much labor they will need to handle incoming and outgoing shipments [2].

A global forecasting model, such as a recurrent neural network (RNN), is a model that uses information from multiple time series to make predictions [3,4]. This is in contrast to a univariate forecasting model like autoregressive integrated moving average (ARIMA) or exponential smoothing (ETS) [5], which only use information from a single time series to make predictions. In a global forecasting model, the RNN is trained on multiple time series at once. Each time series is considered a separate input sequence, and the RNN is trained to capture the patterns and relationships between the different time series. This allows the model to make predictions for all of the time series simultaneously. One advantage of using a global forecasting model like an RNN is that it can capture complex dependencies between the different time series. For example, the sales of one product in a store may be influenced by the sales of a complementary product, or sales at one store may be influenced by sales at nearby stores. By using a global model that takes into account all of the relevant time series, we can better capture these relationships and make more accurate predictions [6]. Another advantage of using a global model is that it can help reduce uncertainty in the individual time series. In some cases, individual time series may be noisy or exhibit erratic behavior. By using a global model that incorporates information from multiple time series, we can reduce the impact of these individual anomalies and improve the overall accuracy of the forecasts [7].

In retail forecasting, covariates (also known as exogenous variables or features) can be used to help improve the accuracy of forecasts [8]. Retail sales can be influenced by the time of year, as well as holidays, weekends, and other special events. Including calendar variables such as the day of week, month, and year can help capture these effects. Changes in price can also have a significant impact on sales. Including variables such as regular price, discount percentage, and promotional price can help capture these effects. Promotions, advertising, and other marketing activities can also influence sales. Including variables such as the type and frequency of marketing activities can help capture these effects. Weather conditions can affect the demand for certain products. Including variables such as temperature, precipitation, and wind speed can help capture these effects. The characteristics of a store's local area can also influence sales. Including variables such as population density, median income, and age distribution can help capture these effects. Overall economic conditions can also influence sales. Including variables such as unemployment rate, GDP, and consumer confidence can help capture these effects.

The objective of this research was to explore how the accuracy of global forecasting models is influenced by incorporating various potential demand covariates, considering their potential effects on operational decisions. The subsequent structure of this paper is organized as follows: In Section 2, we outline the developed forecasting framework designed for the evaluation study, while Section 3 delves into its implementation details. Moving forward, Section 4 unveils and discusses the obtained results, and finally, in Section 5, we offer concluding remarks along with identifying potential areas for further research.

## 2. Autoregressive Neural Network Model

In this research, we employ a deep learning RNN sequence-to-sequence model known as DeepAR [9], which represents a state-of-the-art algorithm specifically designed to tackle the intricate challenges inherent in time series forecasting. Developed by Amazon Web Services (AWS), DeepAR has demonstrated excellent performance across diverse domains such as finance, healthcare, and supply chain management. This success can be attributed to its inherent capability to capture complex temporal patterns and dependencies within data, which may remain obscured when using traditional forecasting methods [10,11]. A notable advantage of DeepAR lies in its built-in ability to model uncertainty, offering probabilistic forecasts [12,13]. This feature proves particularly crucial in decision-making processes that demand a nuanced assessment of both risks and opportunities. Furthermore, DeepAR's flexibility in accommodating varying data characteristics and its adaptability to incorporate external covariates make it a versatile tool for enhancing the accuracy and reliability of predictions.

Let $z_{i,t}$ denote the sales of product $i$ at time $t$. The primary objective of DeepAR model is to forecast the conditional probability $P$ of future sales $z_{i,t_0:T}$ using past sales $z_{i,1:t_0-1}$ and additional information in the form of covariates $\mathbf{x}_{i,1:T}$, where $t_0$ represents the first time instant of the future and $T$ represents the last time instant of the future [14]:

$$P(z_{i,t_0:T}|z_{i,1:t_0-1}, \mathbf{x}_{i,1:T}). \tag{1}$$

It is important to note that the time index $t$ is relative, meaning that $t = 1$ may not correspond to the initial time point of the time series. During training, we have access to $z_{i,t}$ in both the conditioning range $[1, t_0 - 1]$ and the prediction range $[t_0, T]$. The former is used for encoding, while the latter is used for decoding in the sequence-to-sequence model. However, during inference (when we make predictions), $z_{i,t}$ is not available in the prediction range.

At each time step $t$, the model produces an output represented by $\mathbf{h}_{i,t}$:

$$\mathbf{h}_{i,t} = h(\mathbf{h}_{i,t-1}, z_{i,t-1}, \mathbf{x}_{i,t}; \Theta). \tag{2}$$

This output is obtained by applying a multi-layer RNN with long short-term memory (LSTM) cells [15] parameterized by $\Theta$. The model is considered autoregressive because it takes the sales value from the previous time step $z_{i,t-1}$ as input. Additionally, it is considered recurrent, as the output from the network at the previous time step $\mathbf{h}_{i,t-1}$ is given back as input at the next time step.

During training, we learn the model parameters by maximizing the log-likelihood of a chosen probability distribution using the equation:

$$L = \sum_{i=1}^{N} \sum_{t=0}^{T} \log L(z_{i,t}|\theta(\mathbf{h}_{i,t})). \tag{3}$$

Here, $L$ denotes the likelihood of the distribution, $N$ corresponds to the number of products, and $\theta$ represents a linear mapping from the function $\mathbf{h}_{i,t}$ to the parameters of the distribution. DeepAR uses the entire time range to calculate the loss since the encoder model is identical to the decoder. DeepAR forecasts a single value at each step. However, during the inference phase, to predict several steps, the model repeatedly obtains forecasts for subsequent periods until the forecast horizon is reached. The model starts by generating samples from a probability distribution that has been trained on past sequences. The first prediction is made using these samples, and this prediction is then used as input to the model to make the next prediction. This process is repeated for each subsequent period. Because the predictions are derived from samples taken from the trained distribution, the model's output is probabilistic. This means that the model does not produce a single deterministic value for each prediction, but rather a distribution of possible values. This distribution can be used to assess the forecasting accuracy of the model by providing a measure of the uncertainty associated with each prediction. The sampling mechanism also allows the model to be used to generate different forecasting scenarios. By sampling from the distribution of predictions, the model can be used to generate a range of possible outcomes, which can be used to inform decision making.

Sales data often exhibit a zero-inflated distribution, meaning that there are a significant number of observations that equal zero [16]. This can pose a challenge for forecasting models, as they are typically not designed to handle zeros [17–20]. To address this challenge, we used the negative log-likelihood of the Tweedie distribution as our loss function [21,22]. The Tweedie distribution is a flexible distribution that can accommodate zero-inflated data, and the negative log-likelihood is a well-established loss function for Tweedie models. This

approach allowed us to develop a forecasting model that was able to accurately predict both zero and non-zero sales.

$$f(y;\mu,\phi,p) = \frac{y^{p-1}\exp\left(\frac{y\mu^{1-p}}{\phi(1-p)}\right)}{\phi(1-p)y^{p}\Gamma\left(\frac{1}{1-p}\right)}, \quad y > 0. \tag{4}$$

Here, $\Gamma$ represents the gamma function, and $\mu$, $\phi$, and $p$ denote the mean, dispersion, and power parameters, respectively. When $p$ lies between 1 and 2, the distribution takes on the form of a compound Poisson-gamma distribution, which is frequently used for datasets displaying positive skewness and a significant number of zeros. The dispersion parameter $\phi$ regulates the level of diversity or heterogeneity in the data. A small value of $\phi$ suggests high dispersion in the data, whereas a large $\phi$ value indicates homogeneity.

## 3. Empirical Study

### *3.1. Dataset and Exploratory Analysis*

In this study, we used the M5 competition dataset, which is a well-established and openly available dataset of hierarchical unit sales data from Walmart. The M5 dataset is widely used for forecasting research because it is credible and reproducible. The M5 dataset comprises 3049 items categorized under Hobbies, Foods, and Household. These categories are further divided into a total of seven departments. Specifically, the Foods category is subdivided into three distinct departments (Foods1, Foods2, and Foods3), while both the Hobbies and Household categories are each subdivided into two departments (Hobbies1, Hobbies2, Household1, and Household2) [23]. These items are available for sale across 10 stores located in three states: California (CA), Texas (TX), and Wisconsin (WI). The state of California encompasses four stores (CA1, CA2, CA3, and CA4), whereas the states of Texas and Wisconsin each have three stores (TX1, TX2, and TX3; WI1, WI2, and WI3, respectively). The dataset covers a period of 5.4 years, from 29 January 2011 to 19 June 2016, on a daily basis, totaling 1969 days.

In addition to sales data, the M5 dataset also includes the regular price of each item, supplemental nutrition assistance program (SNAP) days, and special events that may impact sales. Approximately 8% of days in the dataset are marked by a special event, which is equivalent to around 160 events in the span of 1969 days. Of these events, around one-third are religious, such as Orthodox Christmas, while another one-third are national holidays, like Independence Day. The remaining third is divided into two-thirds encompassing cultural events, such as Valentine's Day, and one-third sporting events, such as the Super Bowl (please see Figure 1).

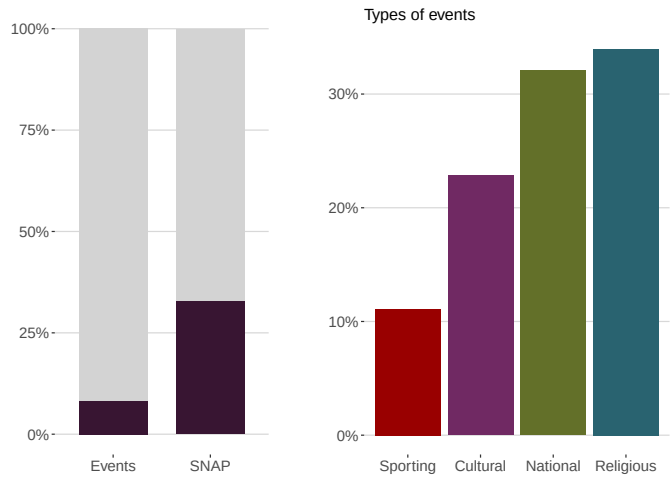


**Figure 1.** Proportion of days corresponding to events and SNAP (on the **left**) and distribution of event types (on the **right**).

The SNAP is a federally funded initiative in the United States aiming to help individuals and families with low incomes to buy food. It was previously referred to as food stamps and is managed by the U.S. Department of Agriculture. As the largest nutrition assistance program in the nation, SNAP plays a crucial role in providing support for those in need of food assistance. The M5 dataset provides information on the SNAP in each state for each day. When we examine the proportion of days on which Walmart stores permit purchases with SNAP benefits, we observe that it is consistent across all three states: 650 days or 33%. SNAP benefits are available for exactly 10 days each month in all states, and these days occur on fixed dates that are the same for every month in every state. In California, SNAP benefits are available in the first 10 days of the month, while in Texas, benefits are available on the 1st, 3rd, 5th, 6th, 7th, 9th, 11th, 12th, 13th, and 15th days. In Wisconsin, SNAP benefits are available on the 2nd, 3rd, 5th, 6th, 8th, 9th, 11th, 12th, 14th, and 15th days. Notably, SNAP days occur in the first half of the month for all states. Figure 2 provides a comprehensive overview of the sales volume during SNAP and non-SNAP days in the three states: California, Texas, and Wisconsin. Although the daily time series are visible in the background, it is more informative to examine the smoothed representations. Our analysis shows that sales volumes are significantly higher on SNAP days compared to non-SNAP days in every state. The largest difference is observed in Wisconsin, while the variations over time are relatively minor. Specifically, the two curves in Wisconsin appear to reach their biggest difference at approximately 2013. However, as with all smoothing fits, it is important to exercise caution when examining data at the edges. In such cases, the results may be less reliable and should be interpreted with caution.

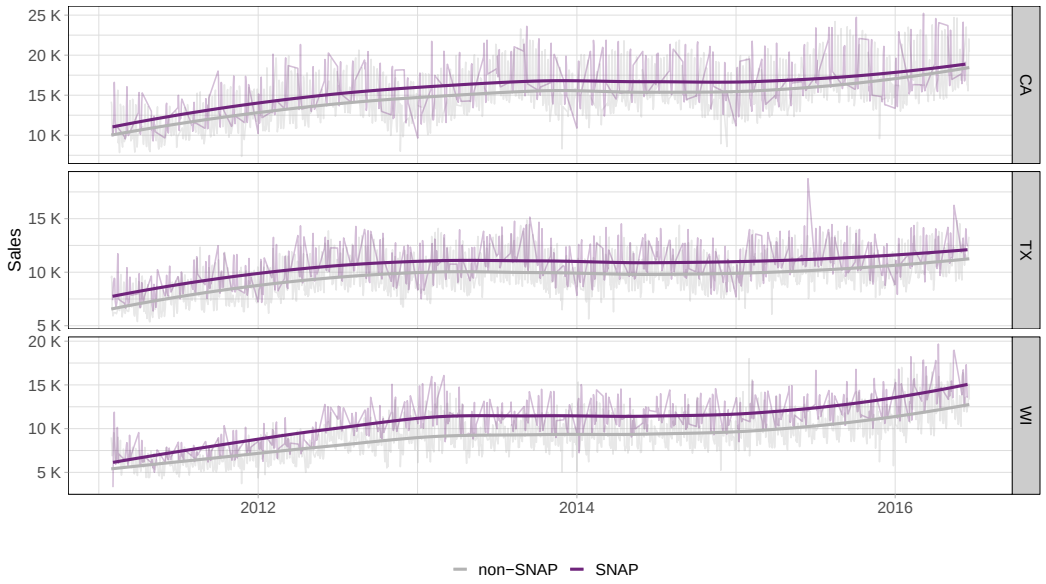


**Figure 2.** Sales per state on SNAP and non-SNAP days.

We also analyzed sales volumes during special event days versus non-event days in the three states (please see Figure 3). Our findings suggest that special events slightly outsell non-event days in Texas before 2014, while afterward, their sales are similar. In California and Wisconsin, there is a drop in sales around the same time, but here it is from similar sales to lower sales. This pattern appears to be common, starting from 2013. Our analysis of event types is particularly interesting, especially for Wisconsin, where national events result in a considerable adverse effect on sales figures (Figure 4). Additionally, Wisconsin stands alone as the sole state where cultural events experience lower sales figures, particularly when compared to Texas. On the other hand, religious events show a relatively minor but still unfavorable effect in Wisconsin, while sporting events have positive impacts in all three states.

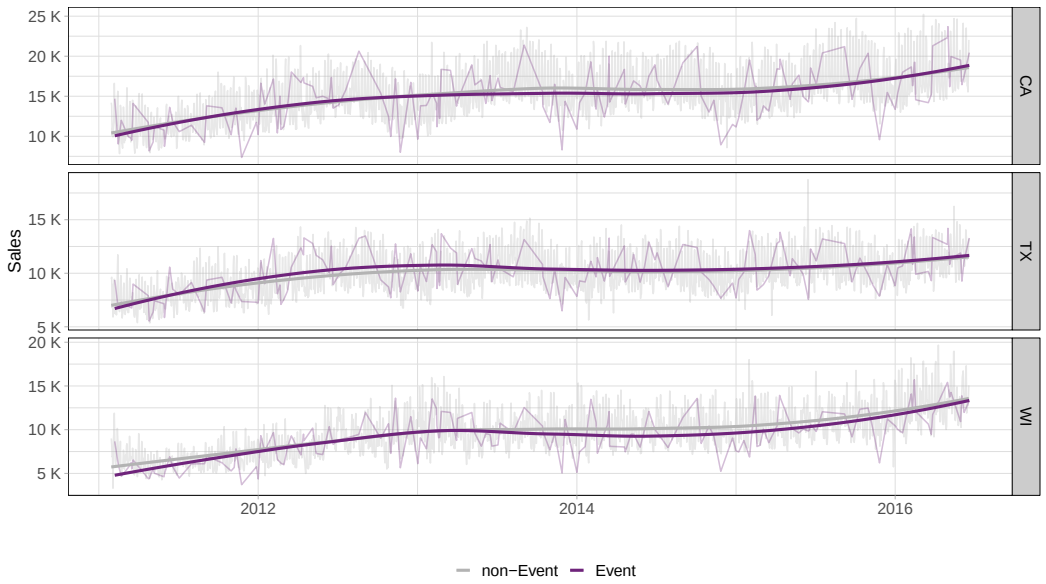


**Figure 3.** Sales per state on event and non-event days.

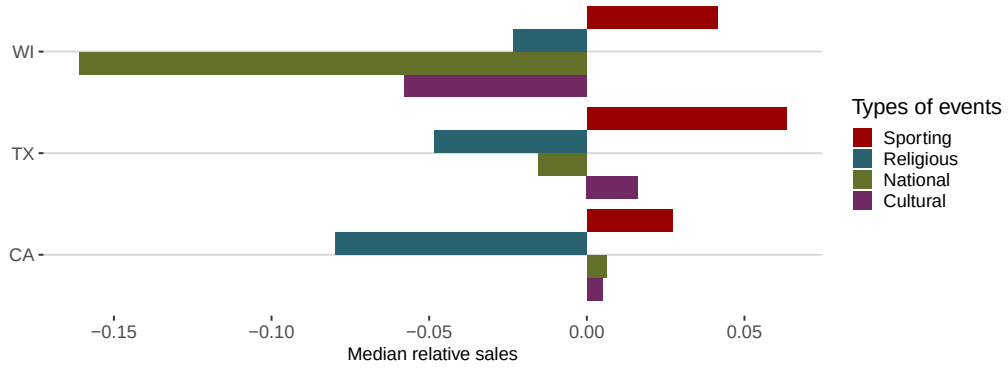


**Figure 4.** Median relative sales per state and per event type.

The "Everyday Low Prices" policy has been a key driver of Walmart's success, helping the company to establish a strong position in the highly competitive retail industry. The idea behind this policy is to provide customers with affordable pricing on a wide range of products throughout the year, rather than relying on the typical retail model of offering higher prices and occasional sales. By maintaining low prices every day, Walmart aims to attract and retain customers who value affordability and reliability in their shopping experience. Figure 5 depicts overlapping density plots for weekly average price distributions within each category's departments for each year from 2011 to 2016. As expected, the price distributions have remained relatively steady, experiencing only slight increases, likely attributed to inflation. However, distinct variations exist among the categories. On average, Foods items tend to be more affordable compared to Household items, whereas Hobbies items exhibit a broader range of prices, even displaying a secondary peak at lower price points. Within each category, there are also substantial differences. For instance, in the Foods category, department 3 (Foods3 in dark green) does not have a high-price tail. The Hobbies category exhibits the greatest diversity, with both departments displaying wide-ranging distributions. The Hobbies2 department (depicted in light green) showcases a bimodal structure, encompassing almost all items priced below USD 10, whereas the Hobbies1 department (in pink) exhibits notably higher prices. The price distributions in the Household category are quite similar, but the Household2 department (in light green) has a peak at clearly higher prices than the Household1 department (in pink). An interesting

trend is visible in the Hobbies2 department, which becomes increasingly bimodal over time. The second peak at USD 1 is growing in importance, almost reaching the level of the main peak just above USD 2. Meanwhile, the small secondary peak at half a dollar in the Hobbies1 department (in pink) becomes flatter after 2012. Conversely, the Household departments remain very stable, while the Foods category shows small changes such as the relative growth of the USD 1 peak in the Foods1 department.

Furthermore, the M5 dataset encompasses details regarding the item ID (ranging from 1 to 3,049), the category ID (Foods, Hobbies, or Household), the department ID (Foods1, Foods2, Foods3, Hobbies1, Hobbies2, Household1, or Household2), the store ID (CA1, CA2, CA3, CA4, TX1, TX2, TX3, WI1, WI2, or WI3), as well as the state ID (CA, TX, or WI) for each individual item.

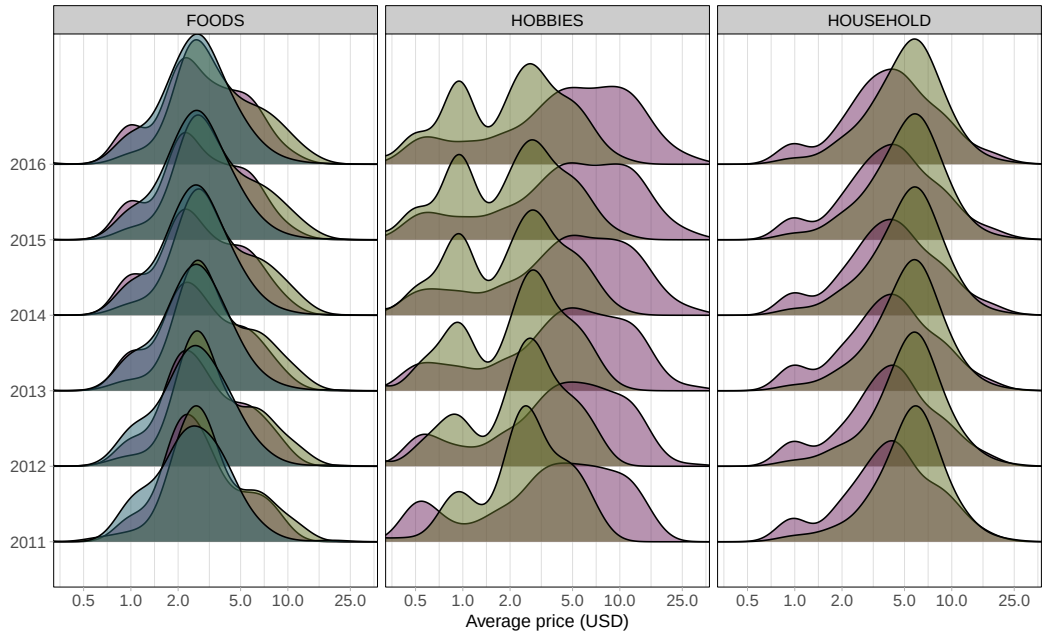


**Figure 5.** Distribution of the weekly average prices within each category's departments for each year from 2011 to 2016 ($X$ axes on a logarithmic scale). From left to right, we have the Foods category's departments: Foods1 in pink, Foods2 in light green, Foods3 in dark green; Hobbies departments: Hobbies1 in pink, Hobbies2 in light green; Household's departments: Household1 in pink, Household2 in light green.

### *3.2. Features Engineering*

There are two distinct types of features: sequential and categorical. Sequential features are represented as two-dimensional real-value data, encompassing both the time and channel dimensions. These sequential features are directly fed into the network without any additional manipulation.

On the other hand, categorical features undergo an embedding process before being fed into the network. During this process, the network learns embeddings through an embedding layer, which serves as a lookup table, mapping input values to trainable embeddings [24].

Four distinct feature groups were taken into consideration: time-related features, price-related features, event-related features including SNAP days, and identification-related (ID) features. The first three groups of features are sequential, while the last group is categorical. Regarding time-related features, eight different value types were used: day, month, year, week of the month, week of the year, day of the year, weekday, and weekend indicator. Following the recommendation by [9], all of these values, excluding the final one, were encoded to the range of $[-0.5, 0.5]$. The weekend indicator was encoded as a binary feature. For instance, in terms of the day values, day 1 is represented as the encoded

value −0.5, day 2 as −0.4666667, day 29 as 0.4333333, day 30 as 0.4666667, and day 31 as 0.5. Similarly, for the weekday, Monday is encoded as −0.5, Tuesday as −0.3333333, Wednesday as −0.1666667, Thursday as 0, Friday as 0.1666667, Saturday as 0.3333333, and Sunday as 0.5.

Three normalized price features were utilized, with normalization achieved through a standard score, i.e., price values were divided by the standard deviation after the mean value was subtracted. The first normalized price value was obtained for each item across all time, considering the disparities from the mean price. The second normalized price value was calculated within each item group that belonged to the same department, allowing for comparisons of the relative price of each item. In each item group associated with the same store, we computed the third normalized price value. This calculation allowed us to make relative price comparisons for each item.

We used three values representing the SNAP days from the M5 dataset, which is a binary feature indicating whether the three states permit SNAP purchases on specific dates, without making any changes to them. The calendar events, characterized by two-dimensional features with varying values based on time, underwent a similar encoding process, scaled to fit within the range of $[-0.5, 0.5]$ prior to being input into the network. Each event is defined by both its name and type. In total, thirty distinct events were considered, each with a corresponding encoded value: days without events were represented as −0.5, Chanukah End as −0.4666667, Christmas as −0.4333333, Cinco de Mayo as −0.4, Columbus Day as −0.3666667, Easter as −0.3333333, Eid al-Fitr as −0.3, Eid Al-Adha as −0.2666667, Father's Day as −0.2333333, Halloween as −0.2, Independence Day as −0.1666667, Labor Day as −0.1333333, Lent Start as −0.1, Lent Week 2 as −0.06666667, Martin Luther King Day as −0.03333333, Memorial Day as 0, Mother's Day as 0.03333333, NBA Finals End as 0.06666667, NBA Finals Start as 0.1, New Year as 0.1333333, Orthodox Christmas as 0.1666667, Orthodox Easter as 0.2, Pesach End as 0.2333333, Presidents' Day as 0.2666667, Purim End as 0.3, Ramadan starts as 0.3333333, St. Patrick's Day as 0.3666667, Super Bowl as 0.4, Thanksgiving as 0.4333333, Valentines Day as 0.4666667, and Veterans Day as 0.5. Each event falls within one of the following four types: days without events are encoded as −0.5, cultural events as −0.25, national events as 0, religious events as 0.25, and sporting events as 0.5. Given that some days have two calendar events, four event-related features were incorporated alongside the existing three SNAP features.

We considered five distinct identification-related features: the item ID, the category ID, the department ID, the store ID, and the state ID. Each of these features was encoded using integers ranging from 0 to one less than its cardinality. As these are one-dimensional features with constant values regardless of time, we replicated them within the time dimension after embedding, ensuring their dimensions matched with other features.

### *3.3. Evaluation Design*

DeepAR global models were trained using the M5 dataset, consisting of 30,490 sales of products across the 10 stores.

The information about the short-term sales trend was derived using the sales data from the previous 28 days as inputs. We adopted the M5 competition's framework, where the last 28 days of each time series (from 23 May 2016 to 19 June 2016) were reserved as a testing set for out-of-sample evaluation. The remaining data, spanning from 29 January 2011 to 22 May 2016 (a total of 1941 days), was used for training the models. In order to reach good accuracy results, it is essential to identify a high-performing model during the testing phase. Usually, a validation set is utilized to assess the most appropriate model. The success of a deep learning model is significantly impacted by various factors, including the values for hyperparameters and the values for the initial weights. In order to identify the best model, we used the final 28 days of in-sample training data, covering the period from 25 April 2016 to 22 May 2016, as the validation set. The hyperparameter optimization process was carried out using the Optuna optimization framework [25], with the root mean squared error (RMSE) [7] serving as the accuracy metric for model selection. To evaluate the

significance of the different types of features, we included them individually and in various combinations, and compared their performance to the baseline case where no features were used.

We assessed the performance of the DeepAR models using two metrics commonly employed in the forecasting literature [26]: the root mean squared scaled error (RMSSE) and the mean absolute scaled error (MASE):

$$\mathrm{RMSSE}_i = \sqrt{\frac{\frac{1}{h}\sum_{t=n+1}^{n+h}(z_{i,t}-\hat{z}_{i,t})^2}{\frac{1}{n-1}\sum_{t=2}^{n}(z_{i,t}-z_{i,t-1})^2}}, \tag{5}$$

$$\mathrm{MASE}_i = \frac{\frac{1}{h}\sum_{t=n+1}^{n+h}|z_{i,t}-\hat{z}_{i,t}|}{\frac{1}{n-1}\sum_{t=2}^{n}|z_{i,t}-z_{i,t-1}|} \tag{6}$$

where $z_{i,t}$ is the sales of item $i$ at time $t$ and $\hat{z}_{i,t}$ is its forecast, $n$ is the length of the training set and $h$ is the forecast horizon. In this particular case study, the time frame considered for the forecast horizon is 28 days. Once calculated for each item, the RMSSE and the MASE were summarized across all items.

The RMSSE was the metric used to assess the accuracy of point forecasts in the M5 competition [27]. RMSSE and MASE are two scale-independent measures that can be utilized to compare forecasts across different items with different scales and units. This is because they scale the forecast errors based on the MSE and MAE of the 1-step ahead in-sample naïve forecast errors, respectively. This ensures that the errors are measured in terms of their absolute or squared magnitude, which allows for fair comparisons across different products. Additionally, RMSSE and MASE differ in how they weight the errors. RMSSE uses squared errors, which emphasizes forecasts that closely adhere to the mean of the target series. MASE uses absolute errors, which emphasizes forecasts that closely adhere to the median of the target series. This difference in weighting allows for different perspectives on the underlying structure of the data.

## 4. Results and Discussion

Table 1 displays the results obtained from an empirical study employing the DeepAR model across various feature combinations, organized by the number of feature groups included. The seasonal naïve model is used as a benchmark. This simple time series forecasting model assumes that the future value of a series is equal to the most recent observed value from the same season. In this context, assuming a seasonal period of 7 days with daily data, as is the case here, the forecast for all future Monday values, for instance, mirrors the last observed Monday value, and so forth. Table 1 highlights the most effective combination of features in boldface within the RMSSE and MASE columns.

Let us highlight the key observations in the results. Firstly, regardless of the error measure used, the DeepAR model with or without features consistently exhibits a significantly better performance compared to the univariate benchmark. The improvements in forecast accuracy are substantial, amounting to approximately 25.8% based on RMSSE. This underscores the advanced capabilities of the DeepAR model in time series forecasting.

Secondly, it is important to note that there are differences in the results between RMSSE and MASE. This difference is understandable since these error measures concentrate on distinct aspects of the distribution of the target variable.

Thirdly, the incorporation of time-, event-, and ID-related features consistently enhances the accuracy of the baseline DeepAR model, which lacks any covariates. This

improvement aligns with expectations, given the significance of special events, sales dates, and item attributes in retail forecasting.

Fourthly, in general, the inclusion of price-related features does not lead to performance improvement in DeepAR models across various feature combinations. This result is unsurprising, especially in the context of a retailer like Walmart, where price distributions tend to remain relatively stable over the years.

Finally, the usefulness of ID-related features significantly surpasses those of time- and event-related features. However, the best performance is achieved when all three of these groups of features are used together, suggesting that the individual relevance of each type of feature is emphasized when the information is given jointly. This model's performance (DeepAR + Events + Time + IDs) improved by 1.8% for RMSSE and 6.5% for MASE when compared to the baseline model without features.

**Table 1.** Performance of DeepAR global models and benchmark evaluated with respect to RMSSE and MASE.

| Model | RMSSE | MASE |
|---|---|---|
| DeepAR | 0.78245 | 0.5718 |
| DeepAR + Prices | 0.78493 | 0.5829 |
| DeepAR + Events | 0.78247 | 0.5692 |
| DeepAR + Time | 0.78190 | 0.5742 |
| DeepAR + IDs | 0.77356 | 0.5404 |
| DeepAR + Prices + Events | 0.78402 | 0.5776 |
| DeepAR + Prices + Time | 0.78330 | 0.5740 |
| DeepAR + Prices + IDs | 0.77461 | 0.5466 |
| DeepAR + Events + Time | 0.78511 | 0.5766 |
| DeepAR + Events + IDs | 0.77221 | 0.5393 |
| DeepAR + Time + IDs | 0.76990 | 0.5359 |
| DeepAR + Prices + Events + Time | 0.78471 | 0.5777 |
| DeepAR + Prices + Events + IDs | 0.77231 | 0.5438 |
| DeepAR + Prices + Time + IDs | 0.76971 | 0.5360 |
| DeepAR + Events + Time + IDs | 0.76866 | **0.5344** |
| DeepAR + Prices + Events + Time + IDs | **0.76864** | 0.5354 |
| Seasonal Naïve | 1.03543 | 0.5889 |

Additionally, in Figure 6, we offer a comparison of how features related to prices and IDs affect the performance of DeepAR models across different feature combinations. Hollow dots represent errors from models without prices/IDs, while filled dots represent errors from models with prices/IDs. These figures unequivocally demonstrate that, regardless of the error measure used, the inclusion of price-related features never enhances the accuracy of any DeepAR model, regardless of the feature combination. Conversely, the addition of ID-related features consistently improves the forecasting performance of a DeepAR model, regardless of the feature combination used. The influence of time- and event-related features closely resembles that of the IDs, albeit with a less pronounced improvement (figures not displayed).

In the M5 dataset, we only had five distinct ID-related features: item ID, category ID, department ID, store ID, and state ID. However, in a real retail setting, there would be numerous additional attributes and features associated with each product. These may include details such as the product type, brand, size, and attributes that are relevant within specific subcategories (for example, "white vs. red" matters for wine but less so for beer, even though both fall under the category of "alcoholic beverages"). Leveraging these additional attributes is likely to result in even more significant improvements compared to solely relying on the five item IDs provided by the M5 dataset.

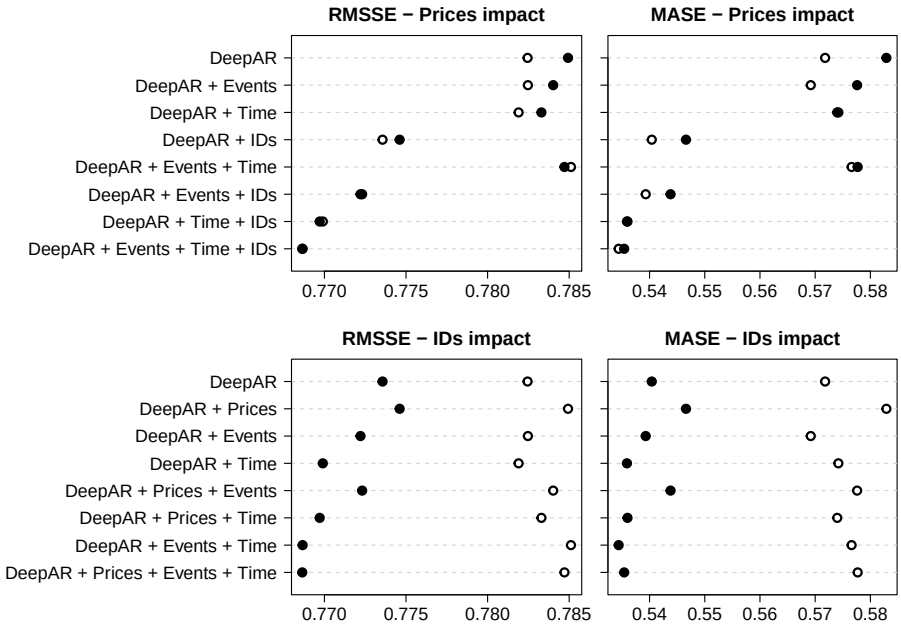


**Figure 6.** Influence of prices and IDs' inclusion in DeepAR models. Hollow dots represent errors from models without prices/IDs, while solid dots represent errors from models with prices/IDs.

It is worth emphasizing that the importance of each feature type hinges on the specific application domain. It is expected that the significance of these specific features may vary in other application domains, and different features may assume a more prominent role.

## 5. Conclusions

Retailers heavily depend on accurate sales forecasts to effectively manage their supply chains and make informed decisions about purchasing, logistics, marketing, finance, and human resources. An advantage of using a global forecasting model, such as an RNN, is its ability to capture intricate dependencies and relationships between different time series. For instance, the sales of a complementary product can affect the sales of another product in a store, or the sales of a particular store can be influenced by the sales at nearby stores. By utilizing a comprehensive model that incorporates all relevant time series, we can more effectively capture these interrelationships and make more accurate predictions.

In retail forecasting, covariates such as calendar events, changes in pricing, and weather conditions can be employed to enhance the forecast accuracy. The objective of this study was to examine how the accuracy of global forecasting models is influenced by the inclusion of various possible demand covariates, considering their potential impact on operational decision making. To ensure the significance of our findings, we employed the widely recognized and openly accessible dataset from the M5 competition. We trained DeepAR global models using the complete M5 dataset, which comprises 30,490 product sales across ten stores.

To assess the significance of the different feature types, we included them both individually and in various combinations, comparing their performance against the baseline case where no features were used. The findings reveal that the DeepAR model, whether with or without additional features, consistently demonstrates a significantly superior performance compared to the univariate seasonal naïve benchmark. The inclusion of time, event, and ID-related features consistently enhances the accuracy of the baseline DeepAR model, which lacks any covariates. In general, the inclusion of price-related features does not lead to performance improvements in DeepAR models across various feature combinations. This observation is not surprising, especially in the context of a retailer like Walmart, which follows an "Everyday Low Prices" strategy, resulting in relatively stable price distributions. The utility of ID-related features greatly surpasses that of time and event-related features.

However, the best performance is achieved when all three groups of features are used in combination, suggesting that the individual relevance of each feature type is accentuated when information from all sources is considered jointly. Overall, the best-performing model demonstrates a 1.8% improvement in RMSSE and a 6.5% improvement in MASE compared to the model without any additional features.

**Author Contributions:** Conceptualization, J.M.O. and P.R.; methodology, J.M.O. and P.R.; software, J.M.O. and P.R.; validation, J.M.O. and P.R.; formal analysis, J.M.O. and P.R.; investigation, J.M.O. and P.R.; resources, J.M.O. and P.R.; data curation, J.M.O. and P.R.; writing—original draft preparation, J.M.O. and P.R.; writing—review and editing, J.M.O. and P.R.; visualization, J.M.O. and P.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** A publicly available dataset was used in this study. The data can be found here: https://www.kaggle.com/competitions/m5-forecasting-accuracy/data (accessed on 6 January 2023).

**Acknowledgments:** We would like to extend our heartfelt gratitude to Stephen Kolassa for their invaluable contributions during the revision process. Stephen's meticulous attention to detail, insightful feedback, and dedication to improving the quality of our work were instrumental in shaping the final version of this paper.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Ramos, P.; Santos, N.; Rebelo, R. Performance of state space and ARIMA models for consumer retail sales forecasting. *Robot. Comput.-Integr. Manuf.* **2015**, *34*, 151–163. [CrossRef]
2. Ramos, P.; Oliveira, J.M. A procedure for identification of appropriate state space and ARIMA models based on time-series cross-validation. *Algorithms* **2016**, *9*, 76. [CrossRef]
3. Bandara, K.; Hewamalage, H.; Liu, Y.H.; Kang, Y.; Bergmeir, C. Improving the accuracy of global forecasting models using time series data augmentation. *Pattern Recognit.* **2021**, *120*, 108148. [CrossRef]
4. Januschowski, T.; Gasthaus, J.; Wang, Y.; Salinas, D.; Flunkert, V.; Bohlke-Schneider, M.; Callot, L. Criteria for classifying forecasting methods. *Int. J. Forecast.* **2020**, *36*, 167–177. [CrossRef]
5. Petropoulos, F.; Apiletti, D.; Assimakopoulos, V.; Babai, M.Z.; Barrow, D.K.; Ben Taieb, S.; Bergmeir, C.; Bessa, R.J.; Bijak, J.; Boylan, J.E.; et al. Forecasting: Theory and practice. *Int. J. Forecast.* **2022**, *38*, 705–871. [CrossRef]
6. Wang, Y.; Smola, A.; Maddix, D.; Gasthaus, J.; Foster, D.; Januschowski, T. Deep Factors for Forecasting. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Chaudhuri, K., Salakhutdinov, R., Eds.; Machine Learning Research Press; 2019; Volume 97, pp. 6607–6617.
7. Ramos, P.; Oliveira, J.M.; Kourentzes, N.; Fildes, R. Forecasting Seasonal Sales with Many Drivers: Shrinkage or Dimensionality Reduction? *Appl. Syst. Innov.* **2023**, *6*, 3. [CrossRef]
8. Oliveira, J.M.; Ramos, P. Assessing the Performance of Hierarchical Forecasting Methods on the Retail Sector. *Entropy* **2019**, *21*, 436. [CrossRef]
9. Salinas, D.; Flunkert, V.; Gasthaus, J.; Januschowski, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.* **2020**, *36*, 1181–1191. [CrossRef]
10. Oliveira, J.M.; Ramos, P. Cross-Learning-Based Sales Forecasting Using Deep Learning via Partial Pooling from Multi-level Data. In *Proceedings of the Engineering Applications of Neural Networks*; Iliadis, L., Maglogiannis, I., Alonso, S., Jayne, C., Pimenidis, E., Eds.; Springer Nature: Cham, Switzerland, 2023; pp. 279–290. [CrossRef]
11. Oliveira, J.M.; Ramos, P. Investigating the Accuracy of Autoregressive Recurrent Networks Using Hierarchical Aggregation Structure-Based Data Partitioning. *Big Data Cogn. Comput.* **2023**, *7*, 100. [CrossRef]
12. Rangapuram, S.S.; Werner, L.D.; Benidis, K.; Mercado, P.; Gasthaus, J.; Januschowski, T. End-to-End Learning of Coherent Probabilistic Forecasts for Hierarchical Time Series. In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021; Meila, M., Zhang, T., Eds.; Machine Learning Research Press; 2021; Volume 139, pp. 8832–8843.
13. Rangapuram, S.S.; Kapoor, S.; Nirwan, R.S.; Mercado, P.; Januschowski, T.; Wang, Y.; Bohlke-Schneider, M. Coherent Probabilistic Forecasting of Temporal Hierarchies. In Proceedings of the 26th International Conference on Artificial Intelligence and Statistics, Valencia, Spain, 25–27 April 2023; Ruiz, F., Dy, J., van de Meent, J.W., Eds.; Machine Learning Research Press; 2023; Volume 206, pp. 9362–9376.
14. Alexandrov, A.; Benidis, K.; Bohlke-Schneider, M.; Flunkert, V.; Gasthaus, J.; Januschowski, T.; Maddix, D.C.; Rangapuram, S.; Salinas, D.; Schulz, J.; et al. GluonTS: Probabilistic and Neural Time Series Modeling in Python. *J. Mach. Learn. Res.* **2020**, *21*, 1–6.
15. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]

16. Kourentzes, N. Intermittent demand forecasts with neural networks. *Int. J. Prod. Econ.* **2013**, *143*, 198–206. [CrossRef]
17. Lolli, F.; Gamberini, R.; Regattieri, A.; Balugani, E.; Gatos, T.; Gucci, S. Single-hidden layer neural networks for forecasting intermittent demand. *Int. J. Prod. Econ.* **2017**, *183*, 116–128. [CrossRef]
18. Gutierrez, R.S.; Solis, A.O.; Mukhopadhyay, S. Lumpy demand forecasting using neural networks. *Int. J. Prod. Econ.* **2008**, *111*, 409–420. [CrossRef]
19. Zhang, G.; Xia, Y.; Xie, M. Intermittent demand forecasting with transformer neural networks. *Ann. Oper. Res.* **2023**, 1–22. [CrossRef]
20. Babai, M.Z.; Tsadiras, A.; Papadopoulos, C. On the empirical performance of some new neural network methods for forecasting intermittent demand. *IMA J. Manag. Math.* **2019**, *31*, 281–305. [CrossRef]
21. Zhou, H.; Qian, W.; Yang, Y. Tweedie gradient boosting for extremely unbalanced zero-inflated data. *Commun. Stat.-Simul. Comput.* **2022**, *51*, 5507–5529. [CrossRef]
22. Muhaimin, A.; Prastyo, D.D.; Horng-Shing Lu, H. Forecasting with Recurrent Neural Network in Intermittent Demand Data. In Proceedings of the 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 28–29 January 2021; pp. 802–809. [CrossRef]
23. Makridakis, S.; Spiliotis, E.; Assimakopoulos, V. The M5 competition: Background, organization, and implementation. *Int. J. Forecast.* **2022**, *38*, 1325–1336. [CrossRef]
24. Jeon, Y.; Seong, S. Robust recurrent network model for intermittent time-series forecasting. *Int. J. Forecast.* **2022**, *38*, 1415–1425. [CrossRef]
25. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-Generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19), Anchorage, AK, USA, 4–8 August 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 2623–2631. [CrossRef]
26. Hyndman, R.J.; Koehler, A.B. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. [CrossRef]
27. Makridakis, S.; Spiliotis, E.; Assimakopoulos, V. M5 accuracy competition: Results, findings, and conclusions. *Int. J. Forecast.* **2022**, *38*, 1346–1364. [CrossRef]