SCHOOL OF
ECONOMICS
AND MANAGEMENT
UNIVERSITY OF PORTO

USING GLOBAL FORECASTING TO PREDICT SALES AT COMMUNITY PHARMACIES

André Rodrigues Vicente

Dissertation

Master in Health Care Economics and Management

Supervised by

Prof. Dr. Mário Amorim Lopes

Co-supervised by

Prof. Dr. José Manuel Soares Oliveira

2024

Acknowledgments

First and foremost, I express my deepest gratitude to my supervisors, Dr. Mário Amorim Lopes and Dr. José Manuel Oliveira, for their invaluable guidance, unwavering support, and continuous encouragement throughout this research. Their expertise, insightful feedback, and patience have been instrumental in shaping this work.

I would also like to thank the School of Economics and Management of the University of Porto, especially the Master in Health Care Economics and Management's coordinator, Dr. Susana Oliveira, and the dedicated teaching staff, for their support throughout this journey. Their commitment to imparting knowledge, skills, and opportunities has been crucial in completing this Master's degree, significantly contributing to my personal and professional growth.

Finally, I extend my heartfelt thanks to my family, who have always believed in me in every possible way.

Abstract

Introduction: Forecasting plays a pivotal role in retail, optimizing supply chain management, mitigating the bullwhip effect, and enabling just-in-time delivery. While local forecasting models have been standard, recent literature highlights the success of global forecasting models (GFMs). Based on an empirical study, this work aims to determine the most suitable GFM for predicting pharmaceutical sales; to compare the performance between global and local forecasting approaches, and to better understand the performance of GFMs. Additionally, it explores the impact of additional features on forecasting accuracy.

Methods: The empirical study employed three forecasting processes that commonly involved data collection, data preprocessing, model selection, fitting, and evaluation using MAE, RMSE, WQL, and execution time. GFMs such as LightGBM, DeepAR, TFT, DLinear, PatchTST, and Weighted Ensemble, as well as local models like Seasonal Naïve, Theta, and AutoETS (serving as the benchmark) were employed using AutoGluon-TimeSeries. Forecasting process A identified the best GFM and compared global to local models. Forecasting process B investigated whether GFMs perform better by capturing sales patterns of the same product across different pharmacies or of different products within the same pharmacy. Forecasting process C evaluated the impact of additional features on GFMs performance.

Results: The final dataset comprised 200 637 time series of 85 pharmacies. Weighted Ensemble and TFT emerged as the top-performing models with fast execution times. Generally, global models outperformed local approaches consistently for at least 8% on MAE, 9% on WQL, and 10% on RMSE. GFMs seem to primarily enhance performance by capturing sales patterns of all products within the same pharmacy. Feature integration yielded varied impacts across models.

Conclusion: In summary, this study contributes to the comparative analysis between global and local models, additionally shedding light on the efficacy of adding features, in a specific retail sales setting within the healthcare sector.

Keywords: forecasting, retail, global forecasting model, local forecasting model, pharmaceutical sales

Resumo

Introdução: A previsão da procura é crucial no retalho, otimizando a gestão da cadeia de abastecimento, mitigando o efeito-chicote e garantindo entregas pontuais. Embora modelos de previsão locais sejam comuns, literatura recente tem destacado o sucesso dos modelos globais. Baseado num estudo empírico, pretende-se determinar o modelo global mais adequado, comparar o desempenho entre modelos globais e locais e contribuir para a melhor compreensão dos modelos globais. Adicionalmente, é explorado o impacto da integração de atributos no seu desempenho.

Métodos: O estudo empírico utilizou três processos de previsão que envolveram recolha de dados, pré-processamento, seleção, treino dos modelos e avaliação usando MAE, RMSE, WQL e tempo de execução. Foram utilizados modelos globais (LightGBM, DeepAR, TFT, DLinear, PatchTST, Weighted Ensemble) e locais (Seasonal Naïve, Theta, AutoETS, este último como referência) na plataforma AutoGluon-TimeSeries. O processo A identificou o melhor modelo global e comparou modelos globais com locais. O processo B avaliou se os modelos globais capturam melhor padrões de vendas de um produto entre diferentes farmácias ou de todos os produtos numa farmácia. O processo C avaliou o impacto da integração de atributos no desempenho dos modelos globais.

Resultados: Os dados finais compreenderam 200 637 séries temporais de 85 farmácias. Weighted Ensemble e TFT emergiram como os melhores modelos globais com tempos de execução baixos. Em geral, os modelos globais superaram os locais (no mínimo em 8% no MAE, 9% no WQL e 10% no RMSE). Os modelos globais melhoraram o desempenho principalmente ao capturar padrões de vendas de todos os produtos numa farmácia. A integração de atributos produziu impactos diversos.

Conclusão: Este estudo contribuiu para a análise comparativa entre modelos globais e locais, fornecendo adicionalmente pistas sobre a eficácia da integração de atributos num ambiente específico de vendas do retalho no setor da saúde.

Palavras-chave: previsão, retalho, modelo de previsão local, modelo de previsão global, vendas farmacêuticas

Index

Figure Index

Table Index

Introduction

The importance of forecasting in retail is critical since generating accurate sales forecasts at the Stock Keeping Unit (SKU) level improves the effectiveness of supply chain management and subsequently attenuates the bullwhip effect and enables just-in-time delivery[1,2]. If there are forecasting errors or no method at all for forecasting, the probability of out-of-stock conditions increases, which can create customer dissatisfaction and loss of profits for missing sales[3]. Retailers may try to overtake this by overstocking, but that strategy significantly raises inventory costs (e.g., capital cost, warehousing, and deterioration).

As proposed by Petropoulos et al[4], forecasting retail demand happens in a three-dimensional structure: *length*, which refers to the position in the supply chain (store, distribution center, or chain); *depth*, which relates to the level in the product hierarchy (SKU, brand, category or total) and *time* (day, week, month, quarter or year).

Univariate local forecasting models are the usual methods retailers use to forecast demand, such as simple moving averages, Auto-Regressive Integrated Moving Average (ARIMA), and Exponential Smoothing (ETS) models[5]. Since recent literature showed the success of global forecasting models (GFMs) in time series forecasting, it could potentially be useful for community pharmacies' demand prediction, despite the lack of literature specific to the field. Therefore, based on an empirical study, the main objectives of this work are to determine the most suitable GFM for predicting pharmaceutical sales, to compare the performance between global and local forecasting approaches, and to better understand the performance of GFMs. The secondary objective was to evaluate the GFMs performance improvements using features. The empirical study is based on an organization that is currently using a Holt-Winters or Triple Exponential Smoothing variant of ETS to predict sales at the SKU level for around 100 community pharmacies. The predictions are generated on a weekly basis starting on Monday for the entire upcoming month. However, due to the intermittent, trend, and seasonal nature of the data, including products with different seasonal patterns, errors still persist.

The dissertation is structured into six main sections: Introduction (provides a brief description of the problem and its scope), Literature Review (which is a summary of the current state of knowledge), Methodology (explains the objectives and methods employed in the research), Results and Discussion (presents the research findings, discussion, and answer to RQs), Conclusion (summary with conclusions, limitations, and future research suggestions) and References.

Literature Review

Business Analytics

Organizations, regardless of private or public, from healthcare to finance, need to respond quickly because the environment is constantly changing. To take the best course of action, organizations can employ Business Analytics, which encompasses Business Intelligence (BI) and Advanced Analytics[7]. It is important to note that BI is a content-free expression, which means it has different meanings according to different authors. However, it is safe to assume that while in the past BI was used as synonymous with Business Analytics, nowadays BI is used to define the early stage of Business Analytics[7].

Business Analytics is used as a term to describe the whole process that assesses what is happening, predicts or forecasts what is most likely to happen, and, consequently, the decision process is more data-driven[7]. In other words, it is a process by which a team helps an organization make better decisions through the analysis of data[8].

There are three levels of analytics: Descriptive, Predictive, and Prescriptive, as shown in Figure 1. There is a natural progression in the level of insight provided (and typically mathematical sophistication as well) from descriptive to prescriptive analytics. The descriptive part is related to BI[7,8].

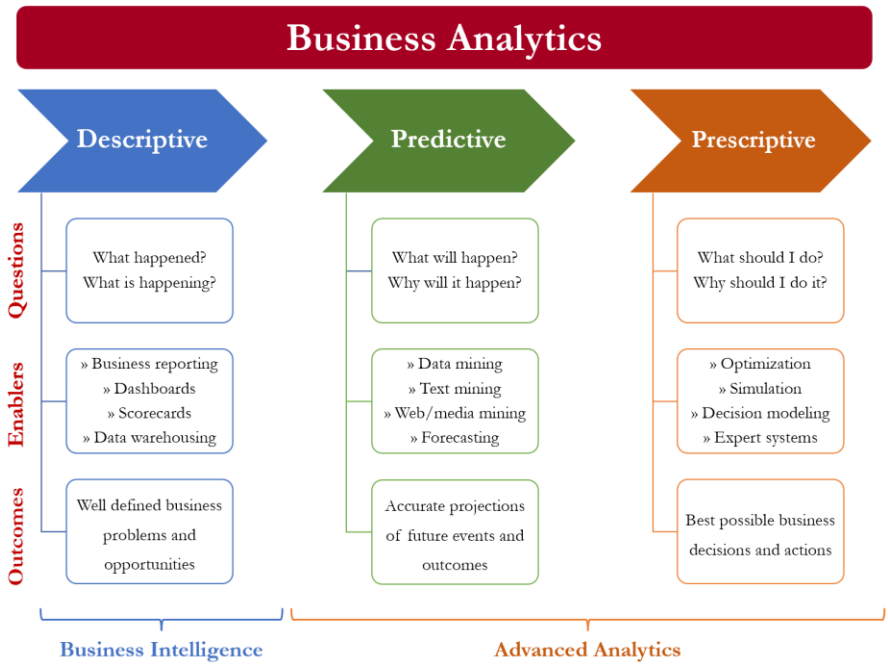


Figure 1: Types of business analytics. Adapted from Sharda[7].

Predictive analytics focuses on determining what is likely to happen in the future. Data mining is a very important concept in this field which integrates several techniques, including

statistical, mathematical, and artificial intelligence, to discover ("mine") knowledge from the data[7]. One might assume that an accurate descriptive model is enough to make predictions. However, this assumption is not entirely correct. Correlation does not imply causation, so making good forecasts implies more sophisticated modeling approaches and more rigorous validation procedures[8].

The patterns that data mining seeks to find may be divided into four categories: **associations** (find correlations or co-occurrences between datasets); **predictions** (predict or forecast the nature of future occurrences based on what happened in the past); **clusters** (identify natural groupings based on their characteristics so objects in a particular group will be similar to one another) and **sequential relationships** (find time-ordered events, like a prediction of sequential events)[7].

The prediction being made can be named more specifically as **classification** (if the result is a class label), **regression** (if the result is a real number), or **time series** (if the result is an extrapolation of future values of the same variable over time). Each data mining method can be classified as supervised (training data includes both the descriptive and class attributes) or unsupervised (training data only includes the descriptive attributes)[7]. Some examples of supervised popular learning processes are shown in Figure 2.

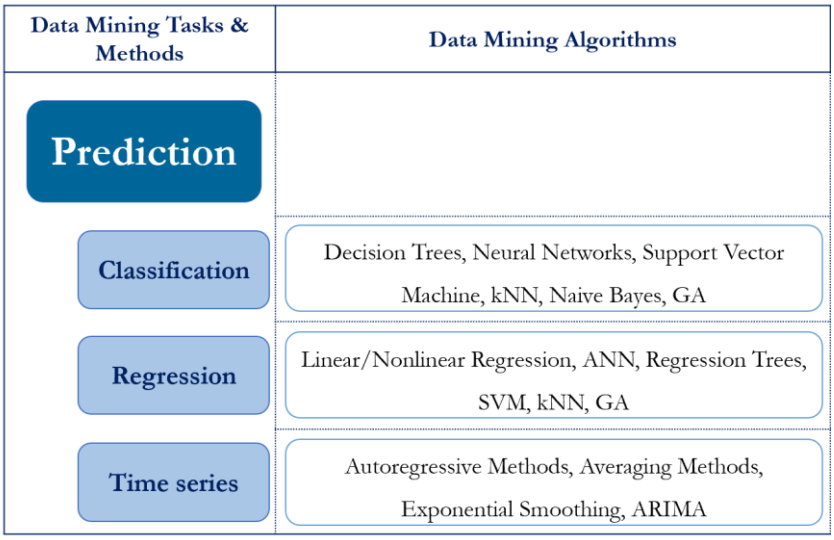


Figure 2: Prediction methods and processes. Adapted from Sharda[7].

Time Series

A **time series** is a set of observations on a variable of interest in a chronological or time-oriented sequence[6]. Time series can be classified as **regular** if the observations occur at consistent time intervals (every day for example) or **irregular** if not[9].

There can be four types of patterns on a time series[6,9,10]:

- **Trend**: typically seen when there is a long-term change in data. If there is an increase, it is an upward trend. If there is a decrease, it is a downward trend.
- **Seasonality**: when a time series varies on a specific, fixed, and known period (time of the year or the day of the week) because of seasonal factors.
- **Cycle**: occurs when the data shows an up-and-down pattern that is not of a fixed frequency. These are often related to the "business cycle". Unlike seasonality, the frequency is not fixed, the average length is typically longer, and the magnitude tends to be more variable.
- **Irregular/Residual/Error term:** what is left when we remove the trend, seasonality, and cycle. It is considered unpredictable, although modern machine learning (ML) tries to capture it or part of it using features.

Three components comprise a time series: a trend-cycle component (a combination of trend and cycle that can be solely termed trend for simplicity), a seasonal component, and a remainder component (same as irregular or residual that contains anything else), as shown in **Figure 3**. There can be more than one seasonal component since some time series can have different seasonal periods. These components can be mixed commonly as **additive** (Y = Trend-cycle + Seasonality + Remainder) or **multiplicative** (Y = Trend-cycle * Seasonality * Remainder)[10].

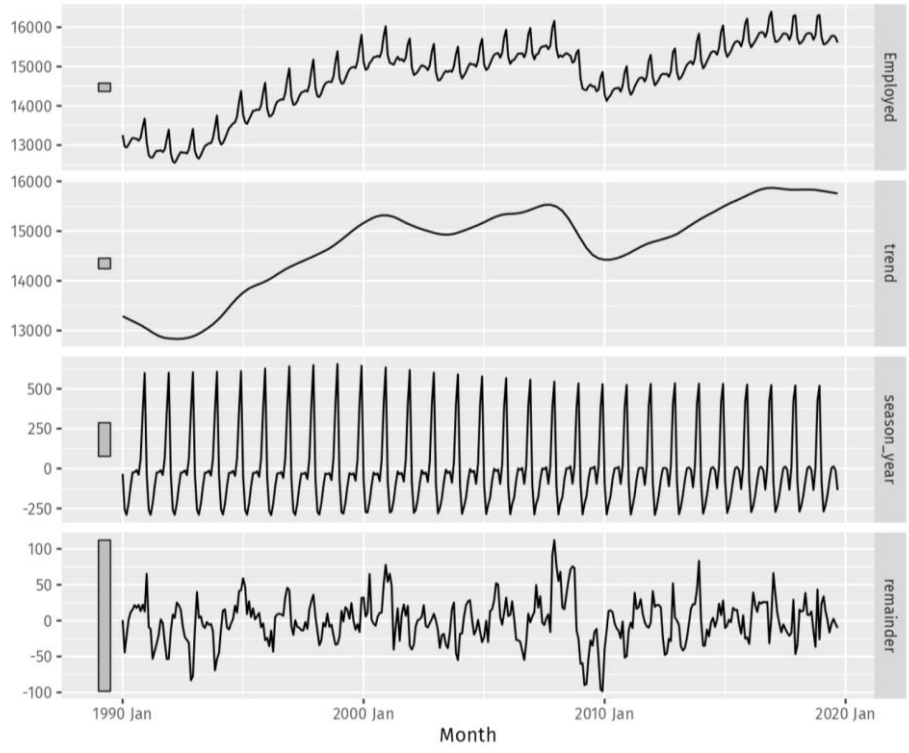


Figure 3: Additive components of a time series: total number of persons employed in US retail. Original from Hyndman & Athanasopoulos[10].

Time series can also be classified as **stationary** when the probability distribution remains the same throughout time or **non-stationary** if the data changes in variance or in mean over time, which is the most common in the real world. In other words, stationary time series typically have no trend (roughly horizontal) or seasonality (no patterns predictable in the long term)[9].

Intermittent time series are characterized not only by multiple non-demand intervals, i. e. long runs of zero demand are observed, but also by variability in the quantity demanded and the intervals between demands. This combination of factors makes forecasting particularly challenging[11].

Time Series Forecasting

The theoretical premise behind time series forecasting is that it is possible to identify those historical patterns previously discussed and implement them in the process of predicting future values[4].

It is important to have in mind some concepts, such as **forecast horizon**: the number of periods forecasts need to be produced; and **forecast interval**: the frequency of new forecasts being made[10]. **Point forecasting** just provides a single, best prediction based on an error metric while **probabilistic forecasting** allows the predictive uncertainty to be taken into account by quantifying it[12].

Hyndman & Athanasopoulos defined the forecastability factors as the predictability aspects that make it easier to forecast: to have a good understanding of the factors that contribute to it; to have a lot of data available; to have a future that is somewhat similar to the past and to have forecasts that can't affect what it is being forecasted[10].

Montgomery, Jennings & Kulahci defined the forecasting process as follows[6]:

1. *Problem definition*: understand how the forecasts will be used in order to meet the expectations of the forecast user.
2. *Data collection*: obtain the relevant data.
3. *Data analysis*: preliminary step to pre-process data to facilitate the next phase (for example decompose a time series, detect outliers and missing values).
4. *Model selection and fitting*: select one or more forecasting models and fit it/them to the data. Fitting means estimating the unknown model parameters using a train set. Usually, data is randomly divided into train and test sets in a process called **data splitting** but, although not straightforward in time series forecasting, a better practice in many fields is to use cross-validation. Basically, it is a resampling

procedure where different portions of the dataset are sampled to train and test in multiple iterations[13,14]. A good rule of thumb is that 75 to 80% of the data should be used for training and 20 to 25% for testing[15,16]. However, this depends on the size of the dataset. The size of the test set should be at least the maximum forecast horizon required[10]. This data splitting can dramatically reduce the risk of overfitting (when a model is too adjusted to the data and has poor performance predicting future values)[8]. To further reduce the risk of overfitting, it is common practice to do a holdout validation: hold out part of the train set (called validation or development set) that is used to select the model with the best parameters on new unseen data[15,16].

5. *Model validation*: evaluate the forecasting model in a "fresh" data environment, usually called the test set (obtained as explained in the previous phase).
6. *Forecasting model deployment*: ensure that the forecast user understands how to use, maintain, and benefit from the model.
7. *Monitoring forecasting model performance*: to prevent performance deterioration over time, it is important to keep track of model performance by monitoring forecast errors.

In what concerns the data pre-processing in phase 3, it's important to review some concepts. **Decomposing time series** is very helpful to improve understanding of the time series and to improve forecast accuracy. The general approach for decomposing time series comprises detrending and deseasonalizing. Detrending, or removing the trend, is commonly done in two different ways: Moving averages or Locally estimated scatterplot smoothing (LOESS) regression. Deseasonalizing, or removing the seasonality, is commonly done using period-adjusted averages or a Fourier series[9,10].

If there is an observation that lies at an abnormal distance from the rest, it is termed an **outlier**. Detecting outliers and properly treating them is important to make forecasting models perform better. There are different techniques to identify outliers. For high seasonal data, deseasonalizing is crucial to be done before using any of these techniques due to the risk of flagging a seasonal peak as an outlier[9]:

- **Standard deviation**: a popular rule of thumb rooted in statistics that defines an outlier as anything that falls beyond $\mu \pm 3\,\sigma$, being μ the mean of the time series and σ the standard deviation.
- **Interquartile range (IQR)**: a similar technique to standard deviation but uses IQR (difference between 3rd quartile - Q3 - and 1st quartile – Q1) instead of standard deviation to identify the bounds. Applying the rule of thumb, the upper bound is

equal to Q3 + 1.5*IQR, and the lower bound is equal to Q1 – 1.5*IQR.

- **Isolation Forest**: an unsupervised process based on decision trees that highlight points outside the norm. A key parameter is contamination which specifies what percentage of the dataset we expect to be anomalous and can range between 0 to 0.5.
- **Extreme studentized deviation (ESD) and seasonal ESD (S-ESD):** a statistics-based technique that uses Grubbs's test to iteratively identify and remove an outlier at each step, adjusting the critical value considering the observations left. S-ESD is especially different because it doesn't need the user to tweak some parameters. It works independently and the user just sets an upper bound on the number of expected outliers.

After outliers are identified, the question is whether to correct them. If the case is a handful of time series, it is good practice to try to anchor them to reality, analyze their causes, and then decide. However, automated techniques are needed if the user faces thousands of time series. Common practices include considering the outliers as missing data and imputing them. It is important to note that outlier correction is not always necessary in forecasting, especially when ML or deep learning are used[9].

Forecasting Techniques

Despite the numerous situations in various fields that require forecasting, there are two broad types of forecasting techniques. **Qualitative** forecasting techniques are often subjective and employed when there is no available data or if it is irrelevant to the forecasts. Examples of methods used include market research, the Delphi method, or panel consensus. **Quantitative** forecasting techniques are usually employed when relevant data is available and include different methods, the most widely used being smoothing, regression, and general time series models[6,10]. The quantitative methods can be divided into **univariate** if the forecasts are developed using only the information of the time series itself; or **multivariate** if other variables are used to produce the forecasts or there are more than one forecasting variable being predicted simultaneously[4].

There is a common practice of classifying forecasting methods as "classical/statistical" (including for example Moving Averages, ETS, ARIMA, SARIMA, and TBATS) or "ML" (such as LightGBM, XGBoost, and Random Forest) and a subtype designated deep learning (includes for example LSTM). Januschowski et al[18] argued that this classification between

classical/statistical or ML methods can be misleading and create artificial boundaries. They proposed distinctions along both subjective and objective dimensions that are more useful for drawing valid conclusions as summarized in **Table 1**.

Table 1: Summary of dimensions for classifying forecasting methods. Adapted from Januschowski et al[18]

Category	***Dimension***
Objective	Global *vs* Local Methods
	Probabilistic *vs* Point Forecasts
	Computational Complexity
	Linearity & Convexity
Subjective	Data-driven *vs* Model-driven
	Ensemble *vs* Single Models
	Discriminative *vs* Generative
	Statistical Guarantees
	Explanatory/Interpretable *vs* Predictive

As highlighted in **Table 1**, there is a distinction that can be made between two extremes: models that forecast independently for each time series (**local**) and methods that forecast jointly from all available time series (**global**). Sometimes there can be a source of confusion since some models can be used in both global and local settings without modifications[18]. In addition, traditional (statistical) forecasting univariate models cannot be trained globally such as ETS, ARIMA, Theta, and Prophet for example[19,20]. Consequently, these models are unable to exploit the cross-series information available in a group of related time series as global models can[20].

There's also a promising idea that is receiving increased attention in the forecasting community, which is to combine forecasting techniques, especially if that ensembles diverse methods so that the forecast errors aren't highly correlated[4]. The ensemble between global (that capture the common patterns across time series) and local forecasting models (that capture the local behaviors of a time series) have been successfully employed[21].

The model currently being used at the organization chosen as the benchmark for this dissertation work is based on ETS, so it is important to get familiar with this method. **Exponential smoothing (ETS)** has been one of the most popular methods since the 1950s. It combines the intuition that history is important but recent data holds greater significance. Therefore, the forecasts produced are weighted averages of past observations, with the weights decaying exponentially as the observations become older. Due to this exponential decay of the influence of older data, the process is termed exponential smoothing[4,9,10]. There

are a few variants of ETS, each tailored to different types of time series data.

Simple exponential smoothing (SES) is suitable for forecasting data with no clear trend or seasonal pattern. Being f_{t+k} the forecast for $t+k$ period, L_t is the level equation to smooth value at time t, α is the smoothing factor, y_t is the observed value at time t, the forecast equation and level smoothing equation are (1) and (2) respectively[4,9,10].

$$f_{t+k} = L_t \quad (1)$$

$$L_t = \alpha \cdot y_t + (1-\alpha)L_{t-1} \quad (2)$$

Nowadays, the best optimal parameters are chosen by minimizing the Sum of Squared Errors (SSE)[4,10].

Double exponential smoothing (DES) or Holt Linear Trend extends the SES to model trend (T). Therefore, the trend equation represented by T_t (5), with a smoothing parameter for trend (β), is introduced to the forecast equation (3) and the level equation is updated (4), as highlighted[9,10].

$$f_{t+k} = L_t + k \cdot T_t \quad (3)$$

$$L_t = \alpha \cdot y_t + (1-\alpha)(L_{t-1} + T_{t-1}) \quad (4)$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \quad (5)$$

This method can result in over-forecasts over the long term because real-world time series don't increase at a constant rate forever. Thus, it has been introduced an addition to this model that dampens the trend by a factor (0<Φ<1). When Φ=1, it is identical to DES. Figure 4 shows an example[9,10].

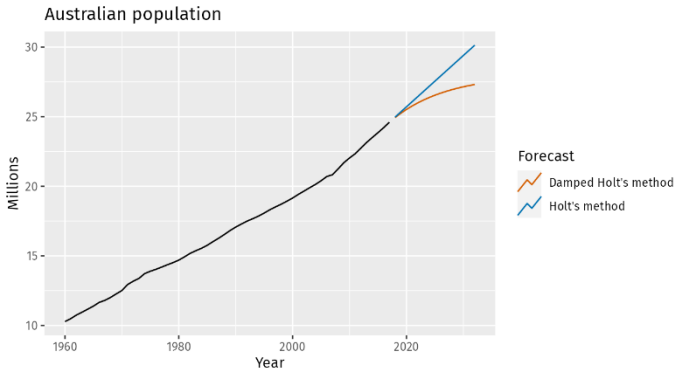


Figure 4: Forecasting annual Australian population (millions) over 2018-2032. For the damped trend method, Φ=0.9. Original from Hyndman[10].

Triple exponential smoothing (TES) or Holt-Winters (HW) extends the DES to capture seasonality (S_t) with a corresponding smoothing parameter (γ) and period of the seasonality (m)[9,10]. There is also an integer part of (k−1)/m (i) that ensures the estimates used for calculating seasonality indices come from the final year of the sample[10]. There are two types

of this model depending on the nature of the seasonality component. **Additive method is** used when the seasonal variations are roughly constant through the series which means that, in each m, the seasonal component will add up to approximately zero. To do this, the seasonal component is subtracted from the level equation (7) and the seasonality is added to the forecast (6), as highlighted[9,10].

$$f_{t+k} = L_t + k \cdot T_t + S_{t+k-m(i+1)} \quad (6)$$

$$L_t = \alpha \cdot (y_t - S_{t-m}) + (1-\alpha)(L_{t-1} + T_{t-1}) \quad (7)$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \quad (8)$$

$$S_t = \gamma(y_t - L_{t-1} - T_{t-1}) + (1-\gamma)S_{t-m} \quad (9)$$

Multiplicative method is used when the seasonal variations are changing proportionally to the level of the series. In this case, the seasonality is expressed in relative terms (percentages) (13); the series is seasonally adjusted by dividing through by the seasonal component in the level equation (11) and adding the seasonality as a multiplication in the forecast equation (10), as highlighted[9,10].

$$f_{t+k} = (L_t + k \cdot T_t) \cdot S_{t+k-m(i+1)} \quad (10)$$

$$L_t = \alpha \cdot \frac{y_t}{S_{t-m}} + (1-\alpha)(L_{t-1} + T_{t-1}) \quad (11)$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \quad (12)$$

$$S_t = \gamma \cdot \left(\frac{y_t}{L_{t-1} + T_{t-1}}\right) + (1-\gamma)S_{t-m} \quad (13)$$

In both additive and multiplicative methods damping can be used as the example previously seen in DES[10]. **Figure 5** is a simple visualization of how the types of ETS discussed work.

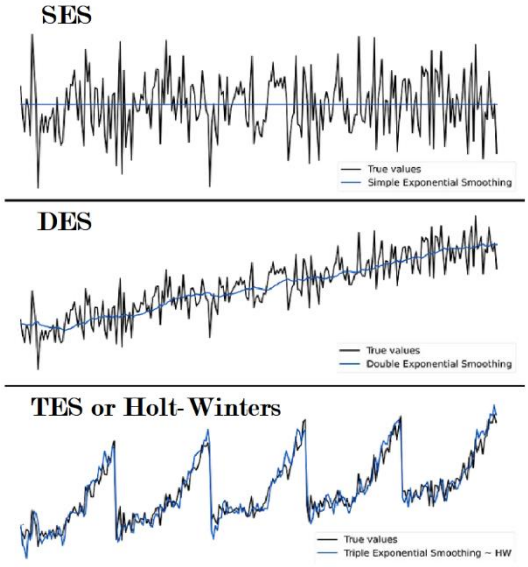


Figure 5: Visualization of how the different types of ETS work. Adapted from Szablowski[17].

Global Forecasting Models for Time Series Forecasting

As discussed, this global forecasting approach fits a single forecasting function to all the related time series instead of fitting one for each individually as local models do[9]. A major difference between global and multivariate models is that global models are not required to have all time series to align with each other and interdependence between time series is not accounted for, while in multivariate forecasting such relationships are directly modeled, and the predictions can be produced for multiple time series variables simultaneously[18].

To build robust data-driven models, large quantities of data are necessary. However, this poses a challenge in time series data as it is constrained by time. Therefore, a theoretical premise of global forecasting is that users can increase the width instead of the length of the time series and thereby increase the amount of data the model is getting trained with (**Figure 6**)[9,22]. The strong assumption is that all series come from the same process and, consequently, they perform better than local ones only when the time series are related. However, the empirical studies showed better performance even when the time series are not related (although there's a lack of theoretical understanding of the causes) which can open the possibility of adopting data-driven techniques that local forecasting can't take advantage of[22,23].

The other advantage is scalability since it drastically reduces the number of models needed to maintain if there are several time series to forecast[9]. Global models can afford to be more complex without being prone to overfitting. Even though the model complexity of a single model is high, when employed as a global model, the total model complexity remains the same[20,22,24]. The main disadvantage is underfitting some specific patterns[9].

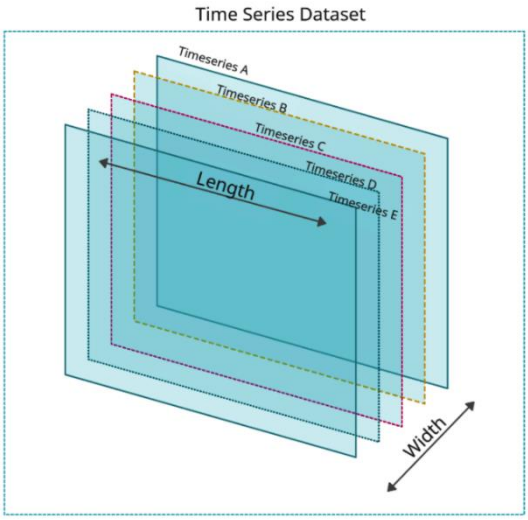


Figure 6: The length and width of time series datasets. Original from Manu[9].

The first use of GFM dates back to 2001[25]. Since then, many models have been proposed over the years. The popular models for time series can be summarized in **Table 2**, to better understand the following part of the literature review.

Table 2: Popular global forecasting methods for time series

Regression Models	Regression Model	
	Linear Regression Model	
	Random Forest[26]	
	Gradient-boosted decision trees (GBDT)	XGBoost (2016)[27]
		LightGBM (2017)[28]
		CatBoost (2017)[29]
Neural Networks (NN)	**Recurrent Neural Networks (RNN)**	Long Short-Term Memory (LSTM) (1997)[30]
		Gated Recurrent Unit (GRU) (2014)[31]
		DeepState (2018)[32]
		DeepAR (2020)[33]
	Convolutional Neural Networks (CNN)	Temporal Convolutional Neural Network (TCN) (2018)[34]
Specialized Deep Learning Architectures	Transformer (2017)[35]	
	Neural Basis Expansion Analysis for Interpretable Time Series Forecasting (N-BEATS) (2019)[36]	
	Temporal Fusion Transformer (TFT) (2021)[37]	
	Neural Hierarchical Interpolation for Time Series Forecasting (N-HiTS) (2022)[38]	
	N-Linear (2022)[39]	
	D-Linear (2022)[39]	
	Time-Series Dense Encoder (TiDE) (2023)[40]	
	PatchTST (2023)[41]	
Ensemble Models		

Related Work

The GFMs have gained popularity because of their better performance compared to local approaches in extensive empiric studies[42] and in multiple recent competitions, such as

CIF2016 Forecasting challenge[43], Kaggle (Rossman Store Sales[44], Wikipedia WebTraffic[45], Corporación Favorita Grocery Sales[46], Intermarché[47]), and notably in M5 "Accuracy" competition (the 5th edition of the also-called Makridakis Competition), where for the first time all of the top-performing models were "pure" ML[48].

As reviewed by Bojer & Meldgaard[49], even though a significant proportion of participants haven't revealed their methods, the most recent Kaggle competitions have been dominated by global and ensemble approaches:

- Rossman Store Sales (2015): The objective was to forecast daily sales by store. The accuracy measure was Root Mean Squared Percent Error (RMSPE). The winner and most of the top performers used ensembles of global XGBoost models, which improved the performance by around 5% compared to the best single model. This was the first competition where a NN placed in the top three.
- Wikipedia WebTraffic (2017): The objective was to forecast daily Wikipedia page visits and the accuracy measure was Symmetric Mean Absolute Percentage Error (SMAPE). The winning solution used an ensemble of global RNN with identical structures without much feature engineering. The other top performers used different NN architectures, including RNN, CNN, and FeedForward Neural Networks (FFNN), being that feature complex engineering was not a requirement for its' high performance.
- Corporación Favorita Grocery Sales (2018): The objective was to forecast daily unit sales by product and store, the accuracy measure was Normalized Weighted Root Mean Squared Logarithmic Error (NWRMSLE). This competition showed that participants learn and improve upon the solutions obtained in previous competitions, since both the GBDT approaches of Rossman and NN of Wikipedia were employed by top performers, including the winner who used a relatively complex ensemble of those models. One advance from the earlier competitions was the application of the new and significantly faster gradient-boosting library LightGBM, which makes it easier to experiment with different features and parameters.
- Recruit restaurant visitor forecasting (2018): The objective was to forecast daily restaurant visits by restaurant and the accuracy measure was Root Mean Squared Logarithmic Error (RMSLE). The winner of the competition was an ensemble comprising the average of their models based on LightGBM, XGBoost, and FFNN. The RNN and CNN variants used successfully in the Wikipedia and Corporación Favorita competitions generally performed slightly worse than models based on

GBDT.

- In the M5 "Accuracy" competition, the objective was to forecast unit sales by product and store related to Walmart and the accuracy measure was Weighted Root Mean Square Scaled Error (WRMSSE). Despite many participants haven't shared their methods, most of the top performers used LightGBM not only because it allows effective handling of multiple features, but also because it is fast to compute compared with other gradient boosting methods and doesn't depend on data pre-processing or transformation. The winning submissions provided better accuracy compared to benchmark models like ETS. However, these methods do not guarantee better performance since when the whole sample of participants is considered, the vast majority of ML models (92.5%) failed to outperform the top benchmark[48].

This better performance of global models is in line with the results of a simulation study in three real-world data sets (one of them heterogeneous) that showed a better performance of global methods over local ones (i.e. ARIMA). Within global methods, LightGBM achieved better performance compared to FFNN and RNN, using SMAPE and Mean Absolute Scaled Error (MASE) metrics, with considerably better efficiency in terms of computational time (9.99 seconds versus 655.09 seconds for FFNN and 10793.98 seconds for RNN)[50]. LightGBM with adequate optimization has also shown better accuracy at predicting stock prices at the Moroccan market, during a period of market stress, compared to LSTM, GRU, and CNN, using Mean Absolute Error (MAE), Root Mean Squared error (RMSE) and R-squared metrics. Additionally, LightGBM was significantly faster (8.23 seconds). LSTM and GRU took the longest time and CNN had an intermediate performance (37 seconds)[51]. In line with this work, LightGBM also has shown better performance on Corporación Favorita Grocery sales and Jollychic datasets over LSTM and XGBoost, using the NWRMSLE metric. However, on the other dataset (Rossman), LSTM performed slightly better. When efficiency was compared, LightGBM had the best prediction time compared to LSTM. For example, on the Corporación Favorita dataset, LightGBM needed 1.76 seconds and LSTM 54 seconds to forecast. Further details can be found in Annex I – **Figure 26**, **Figure 27**, and **Figure 28**, respectively.

The other important finding of the M5 "Accuracy" competition is that deep learning methods like DeepAR and N-BEATS have shown potential since the second and third winning submissions were built on variants of those models[48,52]. For load forecasting in distribution networks, global N-BEATS performed better than local LSTM and N-BEATS, in terms of MASE. DeepAR trained globally outperformed global N-BEATS

(0.7422<0.7564)[53]. Further details can be found in Annex I – **Figure 29**.

DeepAR has also been showing potential in other scientific work that used three datasets related to retail (two of them based on weekly item sales from Amazon and the other based on monthly sales of different items by a US automobile company), outperforming some state-of-the-art methods, including ETS and two types of RNN. On average the greatest difference was from ETS with minor impact versus RNN[33]. A similar work comparing global DeepAR with local benchmarks (such as ETS, ARIMA, and Seasonal Naïve) on forecasting retail demand showed superior performance by a significant margin, using MASE and Root Mean Squared Scaled Error (RMSSE) as accuracy metrics[23]. Further details can be found in Annex I – **Figure 30** and **Figure 31**, respectively. An empirical study on predicting city-wide electricity consumption using global forecasting demonstrated that DeepAR and DeepState models are slightly inferior to more generic deep learning models such as LSTM and GRU in terms of their point forecasting performance, considering the metrics SMAPE, Normalized Deviation (ND) and Normalized Root Mean Squared Error (NRMSE), and yet provide a better probabilistic forecasting performance for longer prediction horizons. Even though the local forecasting methods (such as Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors – SARIMAX - and Naïve Baseline) provided the worst predictions, the performance gains of global models weren't that meaningful mainly due to the relatively simple structure of the dataset[54]. Further details can be found in Annex I – **Figure 32**. In contrast, DeepAR trained locally for groundwater level prediction was better than global DeepAR, measured by RMSSE[55].

Laptev et al used global LSTM on Uber data with better performance compared to the current proprietary method comprising a univariate time series and ML model (Mean SMAPE 26.66<32.44). They highlighted that there are three criteria for picking up a NN for time series over a classic method: (a) high number of time series (b) high length of time series and (c) high correlation among the time series[56]. The first research that employed a GFM-based methodology (LSTM) to conduct causal inference outperformed local methods such as ARIMA, ETS, and LSTM trained as local, using SMAPE and MASE metrics, on two datasets: emergency attendances related to alcohol intoxication reported in Australia and 911 emergency call demand in the Montgomery County, Pennsylvania[57]. Further details can be found in Annex I – **Figure 33**.

When used as a GFM to predict the quarterly production volume of 325 crops across 83 provinces in the Philippines, a Transformer global model performed better in terms of forecast accuracy, using mSMAPE, ND, and NRMSE metrics, when compared against tree-

based ML models (LightGBM, Random Forest and Decision Tree) and traditional local forecasting approaches, such as ARIMA and seasonal naïve. However, the Transformer has the worst training time (1529 seconds) compared to other ML methods (LightGBM 320 seconds, Random Forest 182 seconds, and Decision Tree 21 seconds, which was the best) and to the traditional local methods (ARIMA 280 seconds)[58]. Further details can be found in Annex I – **Figure 34**.

Another work used this Transformer method, which outperformed an LSTM local forecasting approach for energy consumption forecasting on 40 buildings in the Melbourne Bundoora campus, using the RMSE error metric[59]. A similar study extended the work to compare to other local and multivariate models and an LSTM global model. The Transformer as global showed a strong performance better than multivariate and local Transformers, LSTM as global, and other local methods. A type of global transformer with a better hyperparameter configuration (PatchTST) was the best model in five out of six cases. However, LSTM trained as global was significantly faster[60]. In a case study related to online fashion retail prediction of demand, a Transformer approach outperformed LightGBM, DeepAR, and a naïve method using RMSE and Mean Absolute Percentage Error (MAPE)[61]. Further details can be found in Annex I – **Figure 35**, **Figure 36**, and **Figure 37**, respectively.

For forecasting soil moisture in smart agriculture, the use of TFT as global with covariates showed promising results outperforming a naïve method using MAE, RMSE, and q-Risk metrics[62]. TFT as a GFM outperformed a TCN approach in predicting the daily cases of COVID-19 in the United States of America by a considerable margin, using MAE, SMAPE, and Coefficient of Variation metrics[63]. Further details can be found in Annex I – **Figure 38** and **Figure 39**, respectively.

Abhimanyu Das et al proposed a TiDE model that empirically showed a strong performance on multivariate long-term forecasting, along 7 datasets, outperforming DLinear, N-HiTS, and Transformer based models, using MAE and Mean Squared Error (MSE) metrics. TiDE model is also 5-10x faster than Transformer based baselines. The scientific work also concluded that using TiDE on M5 forecasting competition had significantly better performance over DeepAR using all covariates (as much as 20% through WRMSSE metric). Furthermore, although there is a degradation without covariates, it still performed better than DeepAR with covariates ($0.637<0.789$)[40]. Further details can be found in Annex I – **Figure 40**.

As in Kaggle competitions, the M5 "Accuracy" competition confirmed the value of combining different forecasts obtained with different methods[48,64]. Another scientific work

applied on 3 datasets (that included Rossmann and Corporación Favorita Grocery datasets) confirmed this advantage of combining methods (using NWRMSLE as metric), specifically LSTM and LightGBM, although the difference for LSTM and LightGBM best single models was not that meaningful (on average 0,96%). In terms of computational time, the combined model had the worst result on all datasets (55.75 seconds), even though the difference to the LSTM single model was minor. The times displayed were related to the Corporación Favorita Grocery dataset with the same conclusion for both remaining datasets[65]. Further details can be found in Annex I – **Figure 28**.

After conducting a literature review, no scientific work was found regarding the application of global forecasting methods for predicting demand in the specific context of community pharmacies.

Improving Global Forecasting Models' Accuracy

There are many strategies to improve global forecasting methods' accuracy although very little research has been done to examine why they work so well from a theoretical perspective. Montero-Manso & Hyndman's work gives some insights motivating the use of larger complexity to improve accuracy, highlighting three principles: adding features, increasing memory, and partitioning[42]. The strategies to improve GFMs were widely used in the literature review ("Related Work" section) and can be summarized in five topics: increasing memory, using meta-features, adding covariates, partitioning, and tuning hyperparameters.

Increasing memory: longer memory leads global models to improve and outperform local ones[66]. This can be achieved by adding lag, rolling, or Exponentially Weighted Moving Average (EWMA) features. Lag features are values of a variable at previous time points[9] (**Figure 7**). The example used by Montero-Manso & Hyndman showed empirically that adding lag features improved accuracy[42]. In line with this study, Ma and Fildes found that individual forecasting models, such as ETS and Autoregressive Integrated Moving Average with eXogenous factors (ARIMAX), had higher accuracy when using one lag compared to three lags. However, GFMs (such as GBDT) showed better performance when using seven lags instead of three[24].

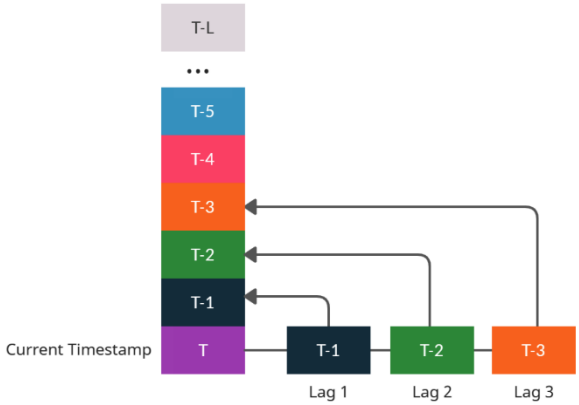


Figure 7: Lag features. Original from Joseph Manu[9].

Rolling features encode the information of time windows using an aggregate descriptive statistic (such as mean or max). It is termed rolling since it is continually updated with new data points as the window "rolls"[9] (**Figure 8**).

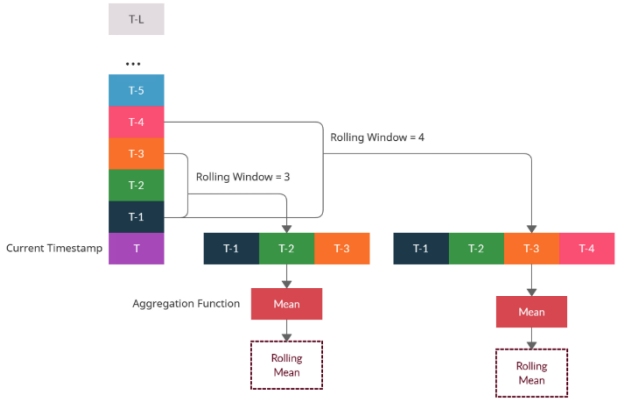


Figure 8: Rolling window aggregation features. Original from Manu[9].

While the moving average considers a rolling window and each item in the window equally, the EWMA takes the weighted average according to the α that it is setted, resulting in different kinds of memory that are encoded as a single feature (**Figure 9**). The larger the value of α, the more the average is skewed toward recent values[9].

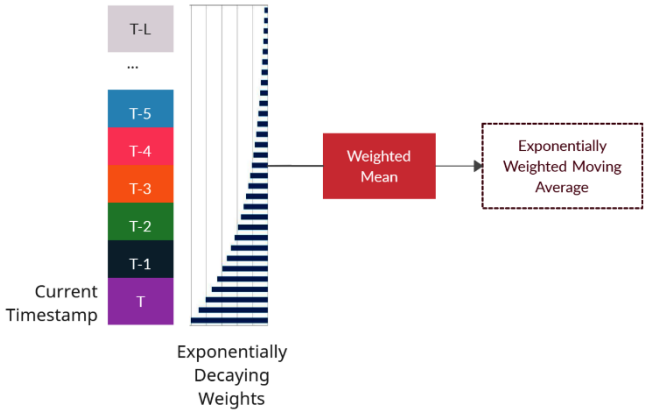


Figure 9: EWMA features. Original from Manu[9].

Using Meta-features: meta-feature is information about the time series itself. In a retail context, it can be the product ID, the category of products, the store number, and so on. However, these are often categorical values that are not appropriate for ML models. To include categorical features in ML models they must be encoded into numerical form[9]. Using a meta-learner framework that, among other attributes, automatically learns meta-features, Ma and Fildes demonstrated that applying it only to a pool of global forecasting methods (M4) had a very limited improvement. The greatest impact was observed when it was applied to a mixed pool of base-forecasters (M0). It displayed superior accuracy over the best-performing base-forecaster (3,2% improvement of Average Relative MAE - AvgRelMAE - over GDBT) and even compared to ensembles, being particularly effective during promotion weeks. Further details can be found in Annex I – **Figure 41**[24].

Adding Covariates: also known as exogenous variables or simply features, covariates refer to external data that can be used as inputs to global models to improve accuracy. Calendar events, changes in pricing, weather conditions, and characteristics of a store's local area can be very important in the retail sales domain. For example, considering retail sales are influenced by the time of year, holidays, weekends and other special events, including those calendar variables can help capture these effects. The same logic applies to changes in pricing (regular price, discount percentage, promotional price, type and frequency of marketing activities for example), to weather conditions (temperature, precipitation, wind speed), and to characteristics of a store's local area (unemployment rate, gross domestic product, and consumer confidence). Using DeepAR on the M5 competition dataset, Ramos et al found that adding those time and event-related features and meta-features (ID-related) led to a 1.8% improvement in RMSSE and a 6.5% improvement in MASE compared to the baseline model without features, being the meta-features the most relevant to that improvement. The inclusion of price-related features did not lead to performance improvement mainly because the dataset was related to Walmart, where price distributions tend to remain relatively stable over the years[52]. Further details can be found in Annex I – **Figure 42**.

When covariates are incorporated, the global model Neural Prophet showed the best performance for groundwater level prediction measured by RMSSE but, unexpectedly, without them, global Neural Prophet performed worse than any other Prophet-based configuration and the local linear model achieved the best performance[67].

Kaggle competitions' review was not clear on the benefits of adding features, other than the meta-features since it led to significant improvements in some competitions but minor or

not at all in others. Available information known in advance (i.e. promotions, holidays, events, and reservations) was highly useful in most of the competitions, however, variables that need to be forecasted (i.e. weather and macroeconomic variables) appeared to provide no significant benefits[64].

In the M5 "Accuracy" competition, all winning submissions utilized covariates to improve the forecasting accuracy[68]. Abhimanyu Das et al used the same dataset to demonstrate the better performance of TiDE and showed the degradation of that model without covariates[69]. Further details can be found in Annex I – **Figure 40**. When authors used viral load as a covariate in forecasting daily cases of COVID-19, it led to significant improvements in TCN and TFT models. Further details can be found in Annex I – **Figure 39**.

Partitioning: normally, it would be expected that the model worked better with more data but partitioning or splitting the dataset into multiple sub-groups and fitting separate global models for each sub-group improved the accuracy of the model. However, the theoretical explanation is still not clear[9,18,62]. The simplest method is random partitioning, where the dataset is randomly split and then the separate models are trained for each partition. Judgmental partitioning is a method that splits the dataset using an attribute (for example a meta-feature) that largely depends on the judgment of the user. Bandara et al used the natural grouping available in a sales product hierarchy (for example product department/category/sub-category) with improvements in accuracy over the baseline LSTM performance (using a modified version of MAPE: mMAPE) on a real-world E-commerce database from Walmart.com[63]. Further details can be found in Annex I – **Figure 43**. Another work, previously referred to, compared judgmental partitioning approaches and concluded that the performance gain over simple global DeepAR is not significant, with an improvement of less than 1% based on RMSSE and up to 3.6% based on MASE. The authors suggest that there's a trade-off between data availability and relatedness. Data partitioning increases relatedness at the cost of a reduced sample size[23]. Further details can be found in Annex I – **Figure 31**. Algorithm partitioning is a method based on an unsupervised clustering approach and then the user trains a model for each cluster. The work of Bandara cited above showed improvements in accuracy when the K-Means clustering approach is used, outperforming baseline LSTM and even the natural grouping approach on the group of products with greater business impact (using mMAPE)[63]. Further details can be found in Annex I – **Figure 43**. Another work from the same author used a fixed set of time series features extracted from each series first and then applied traditional clustering

processes (i.e., K-Means, DB-Scan, PAM, DBSCAN, and Snob) on two competition datasets (CIF2016 and NN5). Using SMAPE, the accuracy of LSTM Cluster variants achieved improvements over the baseline LSTM. Additionally, the clustering approaches had a better model training time compared to the baseline LSTM[19]. Further details can be found in Annex I – **Figure 44.** Another work from Grabner and Blazic compared the use of clustering on an ensemble global approach and concluded that K-Means led to a better result than DB-Scan, in terms of MASE (0,7221<0,7326)[64].

Tuning hyperparameters: the process of finding the best parameters to optimize the performance of an ML model. There are three techniques for doing it, besides manual trial and error[9]. Grid search is a systematic optimization technique that involves discretizing the hyperparameter space into a grid of predefined values. At each intersection point of this grid, the performance of the ML model is evaluated using a chosen objective function to identify the combination of hyperparameter values that yields the best model performance. The negative aspects of this approach are the possibility of another solution that the predefined grid hasn't covered and the potential for computational inefficiency. In random search, the search space is also defined, but instead of defining specific points, probability distributions are defined to explore the different regions of space. While in grid search the number of trials (combinations of hyperparameters) are determined by the predefined grid, in random search users have the flexibility to decide how many trials to run and consequently the extent of computational resources they are willing to allocate. The main disadvantage is the pure exploratory nature since this approach is unaware of what is happening as it evaluates different trials. Bayesian optimization is similar to random search. This technique also defines the space as probability distributions and gives users the ability to choose the number of trials. However, Bayesian optimization is aware of past trials and does extended searches in regions that are giving better results, which is termed exploitation instead of the exploration used in random search. This method is the preferred choice for users who prioritize computational efficiency.

Evaluating and Monitoring Performance

George Box, a British statistician, once said that "all models are wrong, some are useful". No matter what technique is used, these forecasts are almost always wrong. In other words, there is going to be a **forecast error** (difference between an observed value and its' forecast) so it is good practice to provide an estimate of how large that type of error might be experienced, in the form of a prediction interval, for example. Forecast errors are different from residuals.

Residuals are calculated on training data sets when the model is fitted while forecast errors are calculated on the test dataset[6,10].

Some popular forecasting metrics to evaluate forecasting models can be summarized as follows[6,9,65]:

Absolute (scale dependent):

- **MAD** (Mean Absolute Deviation) or **MAE**: calculates the mean error along all forecasts (14). The smaller, the better the model is.

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|(Y_t - \hat{Y}_t)| \tag{14}$$

- **MSE**: calculates the mean of the squared error along all forecasts (15). It is a good metric if users may find it acceptable to experience minor discrepancies while avoiding larger ones.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2 \tag{15}$$

- **RMSE**: the square root of MSE (16). Often preferred over MSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2} \tag{16}$$

- **WQL** (Weighted Quantile Loss): indicated for probabilistic forecasting, this metric is defined as total quantile loss divided by the sum of absolute time series values in the forecast horizon (17). The τ is a quantile in the set, $q_{i,t}^{(\tau)}$ the quantile that the model predicts and $Y_{i,t}$ the observed value at a certain point.

$$WQL = 2\frac{\Sigma_{i,t}[\tau\max(Y_{i,t} - q_{i,t}^{(\tau)}, 0) + (1-\tau)\max(q_{i,t}^{(\tau)} - Y_{i,t}, 0)]}{\Sigma_{i,t}|Y_{i,t}|} \tag{17}$$

Relative (unit-free):

- **MASE**: the quotient of MAE for the average absolute difference between consecutive observations in the time series (18) and thus it accounts for seasonal and trend patterns. If the value is < 1, the method being tested is better than the one-step ahead naïve method in-sample.

$$MASE = \frac{MAE}{\frac{1}{n-1}\sum_{t=2}^{n}|Y_t - Y_{t-1}|} \quad (18)$$

- **MAPE**: calculates the mean error percentage along all forecasts (19). As a disadvantage, it puts a heavier penalty on positive errors than on negative ones.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{Y_t - \hat{Y}_t}{Y_t}\right| \times 100 \quad (19)$$

- **SMAPE**: the symmetric approach (20) is better when dealing with data that has small actual values since it uses the average of the real and the forecast values as the denominator while MAPE just considers the actual values. Even though it's called symmetric, it penalizes the underestimates more than the overestimates, which is arguably even less symmetric.

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{2 \times |Y_t - \hat{Y}_t|}{|Y_t| + |\hat{Y}_t|} \times 100 \quad (20)$$

The selection of metrics is still very debatable, especially for global forecasting methods since the user wants to compare models across datasets or select models that perform generally well on many datasets, each having many series and not the traditional case limited to one time series[65]. Hyndman & Koehler argued that MASE is the best available measure of forecast accuracy when there are different scales including data close to zero or negative. However, if all data are positive and much greater than zero, MAPE may still be preferred for reasons of simplicity. In the case that all series are on the same scale, then the MAE may be preferred also for its' simplicity[66]. Despite that, the measures used in the different competitions are diverse[48,49].

Independently from the chosen comparative metric, there are other measures important to keep track of structural over- or under-forecasting problems over time[6,9]. **Cumulative Forecast Error (CFE)** is the sum of all the errors, including the sign. A positive CFE indicates over-forecasting and a negative CFE under-forecasting. So, the closer to 0, the better. Since CFE is scale-dependent, it is possible to scale it by the actual observations creating the **forecast bias**. The interpretation is the same as CFE. **Tracking signals** is the ratio between CFE and MAD/MAE. Although a thumb rule is to use 3.75, it is defined by the users what is the best threshold. A positive tracking indicates over-forecasting and a negative under-forecasting.

Methodology

Objectives

Based on an empirical study, the main objectives of this work are to determine the most suitable GFM for predicting pharmaceutical sales, to compare the performance between global and local forecasting approaches, and to better understand the performance of GFMs. The following research questions address these objectives:

RQ1: Which GFM is the best fit for the inventory management system to predict the pharmaceuticals' sales?

RQ2: How do the GFMs perform in comparison to local forecasting approaches, including the one currently used by the organization?

RQ3: If GFMs perform better than local models, is it primarily because they better capture the sales patterns of the same product across different pharmacies, or because they better capture the sales patterns of all products within the same pharmacy?

The secondary objective was to evaluate the improvements on GFMs performance using features. This objective was summarized by the following research question:

RQ4: To what extent do adding meta-features and covariates contribute to the improvement of GFMs?

To address the research questions, three methodologies were followed, namely forecasting processes, adapted from the forecasting process defined by Montgomery et al[6], using the following tools.

Tools

Spyder IDE is a free and open-source scientific environment for Python, which was used to follow the forecasting processes using version 5.5.0 and Python version 3.10.11 on a computer that had 8 Central Processing Units and no Graphics Processing Units. AutoGluon version 1.0.0 was the open-source library used for forecasting. First introduced in 2020, AutoGluon framework is a popular automated and open-source ML package for tabular data[75]. Building on the foundation of AutoGluon, AutoGluon-TimeSeries was launched in 2023 focusing on probabilistic time series forecasting, showing better performance than other AutoML libraries in both point and probabilistic forecasting. It adheres to the same principles as AutoGluon framework (such as ease of use, fastness, and robustness), allowing

it to empower users with limited expertise. The user specifies for training the prediction length, the target variable, the evaluation metric, the presets (the only ones applicable to GFMs are best_quality or high_quality, being the difference a greater number of cross-validation windows on best_quality), and the time limit. By default, random search is used for hyperparameter tuning. Trained models are then ranked based on their performance on an internal validation set. This validation set has the same length as the prediction length and typically precedes the test set. The test set comprises the last prediction length of each time series. This means that the training data needs to have a length of at least twice the prediction length plus one to accommodate the internal validation set. Finally, models are evaluated on unseen data (test data) to estimate their performance and ranked according to the same metric used for training[76].

Forecasting Processes

There are three forecasting processes (A, B, and C) that were followed, all adapted from the forecasting process defined by Montgomery et al[6], to address the research questions (**Figure 21** in Appendix I). **Forecasting process A** intends to address RQ1 and RQ2 and it consists of the following steps:

1. *Problem definition*: as previously explained in the introduction section.

2. *Data collection*: the historical sales data of the organizations this project focuses on was retrieved from a database created by the supervisor after pseudonymization. It was accessed using MySQL Workbench through Python with the proper credentials. Due to the significant impact of the COVID-19 pandemic on sales, which was an extraordinary event, the sales were restricted to 04-01-2022 to 01-01-2024. The particular day of the week was defined that way because the forecasts were supposed to be weekly aggregated starting on Monday to be comparable with the current local approach the organization is using. The main variables of interest were the 'sku', 'dim_calendar_key' (sale day on the format yyyymmdd), 'dim_pharmacy_key' (pharmacy identifier), and 'quantity' (sales).

3. *Data analysis*: The complete raw dataset had 15 185 025 rows of data, 72 091 unique SKU's and 117 pharmacies. The following steps were followed to analyse and pre-process data:

3.1. It was created a column defined as 'sku_dim_pharmacy' to represent the concatenated columns 'sku' and 'dim_pharmacy_key' (e.g. 1001875_4) and those columns were dropped from the dataset. This step involved creating a time series representative of each SKU for each community pharmacy. Then, the number of unique time series was 821 462. The sum of sales by day of the week was plotted to analyse the sales behavior throughout the week. This analysis indicates a noticeable drop during the weekend, particularly on Sundays, which is likely related to the normal operating hours of community pharmacies (**Figure 10**).

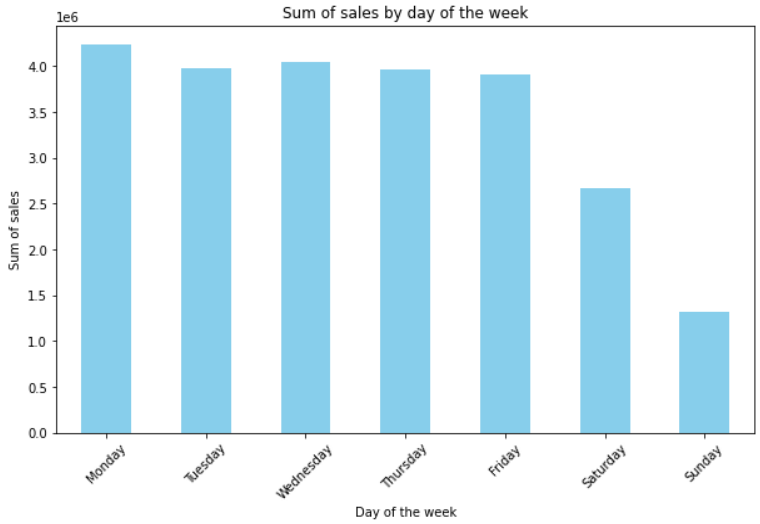
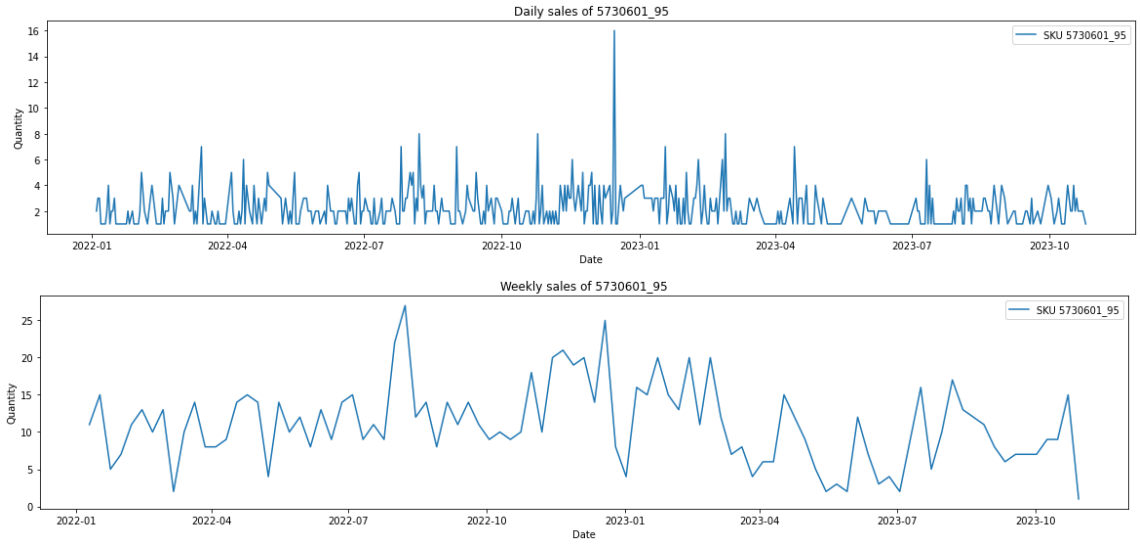


Figure 10: Sum of sales by day of the week.

3.2. Sales were weekly aggregated starting the week on Monday (freq='W-MON'). After this step, the dataset consisted of 9 976 276 rows, 34,3% less than the raw dataset. It was tested with a few examples and the plots were smoother, as shown in **Figure 11** for the 'sku_dim_pharmacy' 5730601_95. It was also checked if the data range was accurate with



a length of 104.

3.3. Since the data contained SKUs that could be already discontinued and the organization did not have a list of those, the dataset was filtered to include only 'sku_dim_pharmacy' with sales on December 2023. After this, the dataset consisted of 6 183 094 rows (-38% than the previous) and the number of unique time series was 239 093 (-71% than the previous).

3.4. Only data from sales was registered on the database so there were abundant missing values when sales of each 'sku_dim_pharmacy' were not sold. These missing values were filled with zeros on 'quantity' for each 'dim_calendar_key' only after the beginning of each time series. This step took about 3 days to be executed and the dataset consisted of 15 463 774 rows (2.5 times the previous) with the same 239 093 time series. It was tested with a few examples of the filling of zeros through tables and plots, as shown in **Figure 12** for 'sku_dim_pharmacy' 10000032_120.

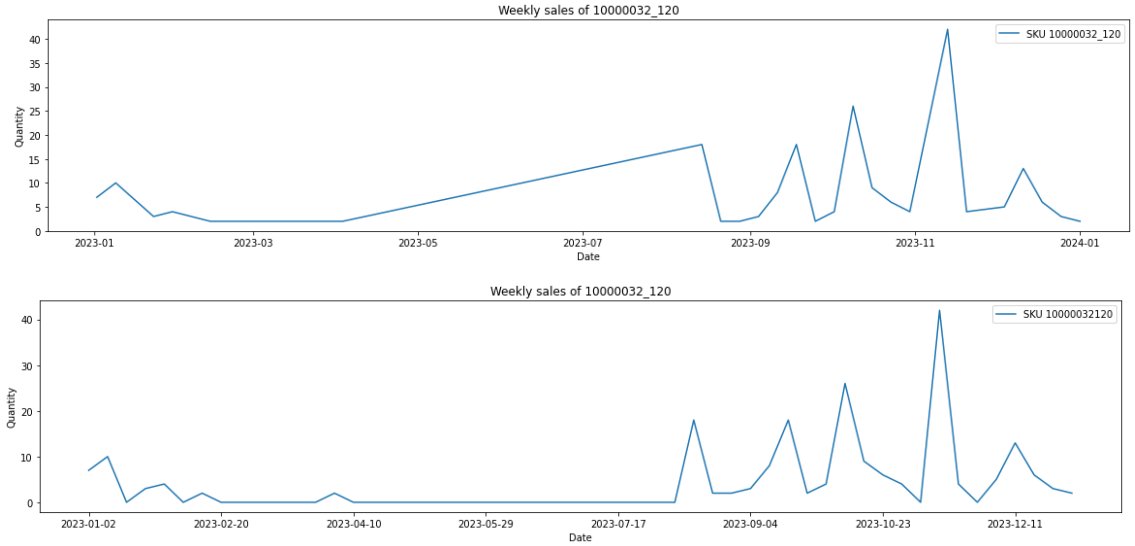


Figure 12: Example of filling with zeros for 'sku_dim_pharmacy' 10000032_120.

3.5. Since the prediction length was defined as 4 to align with the business requirements of the community pharmacies, the dataset was filtered to exclude time series that only had sales after the 4th of December of 2023, since those SKUs represented new products that models could not forecast based on the prediction length. Then, 14 162 time series were identified and removed from the dataset.

3.6. To enhance the robustness of model testing, it was assured that the minimum percentage of the test set was at least 20% of the time series length when using an expanding window cross-validation strategy with 4 windows and a stepsize of 1. Therefore, the minimum length requirement for a meaningful evaluation was established for all time series based on this formula: 3*prediction length + validation set + test set = 3 * 4 + 4 + 7, which amounts to a length equal or greater than 23. Then, the minimum length of the test set was 4 out of 19 (21%) on each expanding step (**Figure 13**). Consequently, 24 294 time series that did not meet that criterion were removed. This refinement resulted in a final dataset comprising 200 637 time series.

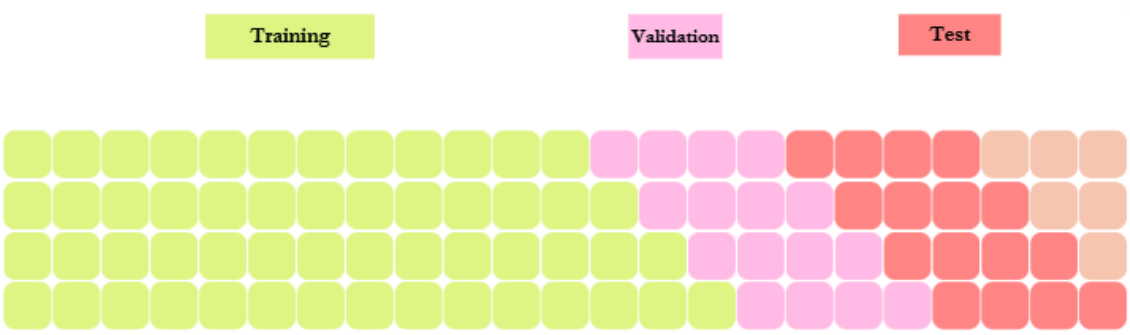


Figure 13: Expanding cross-validation window strategy.

3.7. When the dataset was analysed, it was noted that some time series had a few erratic points on December 2023 (**Figure 14**). It was decided to only identify the extremely erratic outliers using the rule of 80 times the IQR and replace them with the mean of the adjacent points.

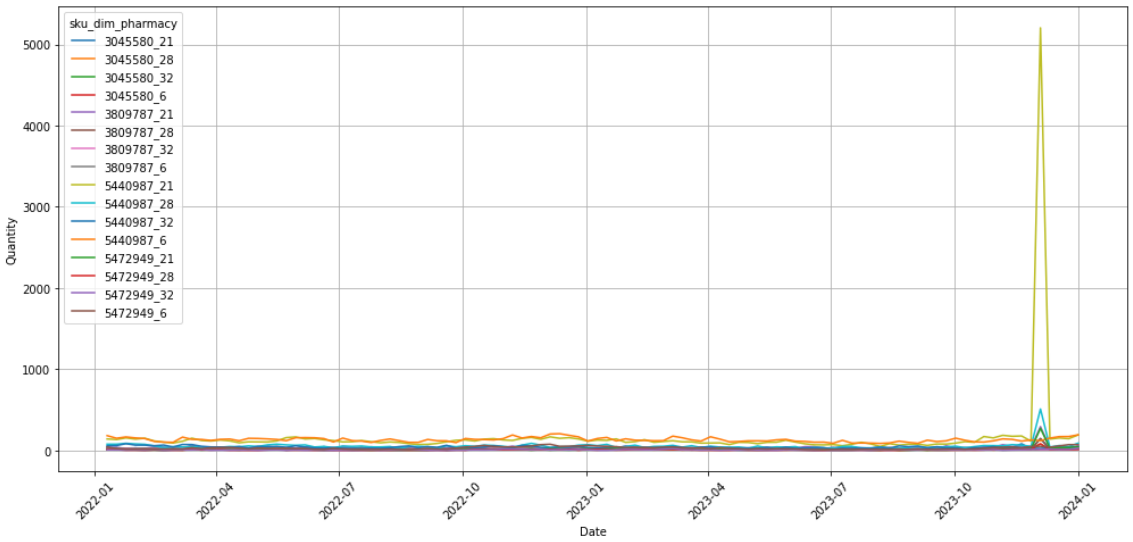


Figure 14: Examples of top sales time series with erratic points on December 2023.

4. *Model selection and fitting*: the following GFMs were chosen to ensure coverage of the most popular and diverse from those available at AutoGluon-TimeSeries: a GBDT (LightGBM), an RNN (DeepAR), a few deep learning architectures (TFT, D-Linear, PatchTST) and a Weighted Ensemble combining these. The three local models chosen were AutoETS, Theta, and Seasonal Naïve.

    4.1. The final dataset was fitted after specifying that the prediction length or forecast horizon was 4, the evaluation metric was MAE, no time limit was set and the preset configuration was defined to 'best_quality' with 4 cross-validation windows and stepsize 1 to perform an expanding window strategy. This strategy has advantages over a rolling window when weekly data is used since the number of historical points is limited[13]. The results of the validation set, based on MAE, and the training and validation times were automatically displayed by AutoGluon-TimeSeries after training.

5. *Model validation*: benchmarks are commonly used by researchers and practitioners as standards against which the models proposed are being compared. The benchmark for this work was the best proxy to the model the organization is employing (AutoETS) and two other local models available on AutoGluon-TimeSeries (Theta and Seasonal Naïve).

    5.1. Probabilistic forecasts were predicted using the final dataset.

    5.2. Using 'predictor.leaderboard', the leaderboard was displayed based on the metric defined for training (MAE) with the corresponding prediction time for each model.

    5.3. To further evaluate the forecasts, it was used the 'predictor.evaluate' with a few extra metrics (WQL and RMSE). The accuracy metrics are very debatable but these three (MAE, WQL and RMSE) are well-suited for dealing with intermittent time series that have a lot of zeros, which is the case of the dataset. RMSE is an essential metric for commercial organizations such as community pharmacies since they operate in a highly sensitive environment where it is important to identify and mitigate the greater errors that lead to significant losses. WQL is specially indicated to evaluate the accuracy of probabilistic forecasting.

Forecasting process B addresses RQ3 to better understand the improved performance of GFMs proposing a new approach not found in literature review. This process mirrors the one previously described but includes specific variations in the data analysis process (step 3). It introduces two different approaches after the outlier correction (3.8). A vertical approach that focuses on specific pharmacies, allowing the models to analyze sales performance across different SKUs for the same pharmacy. To provide a balanced distribution, the pharmacies selected were the two pharmacies with sales figures closest to the overall average sales across all pharmacies and the two pharmacies with the highest total sales. A horizontal approach that focuses on specific products rather than pharmacies, allowing the models to analyze sales performance across different pharmacies for the same SKU. The SKUs selected were the common top 4 SKUs of the pharmacies previously selected on the vertical approach (**Figure 15**), excluding the bestseller ('5440987') since it had an erratic behavior even after the outlier correction (**Figure 16**).

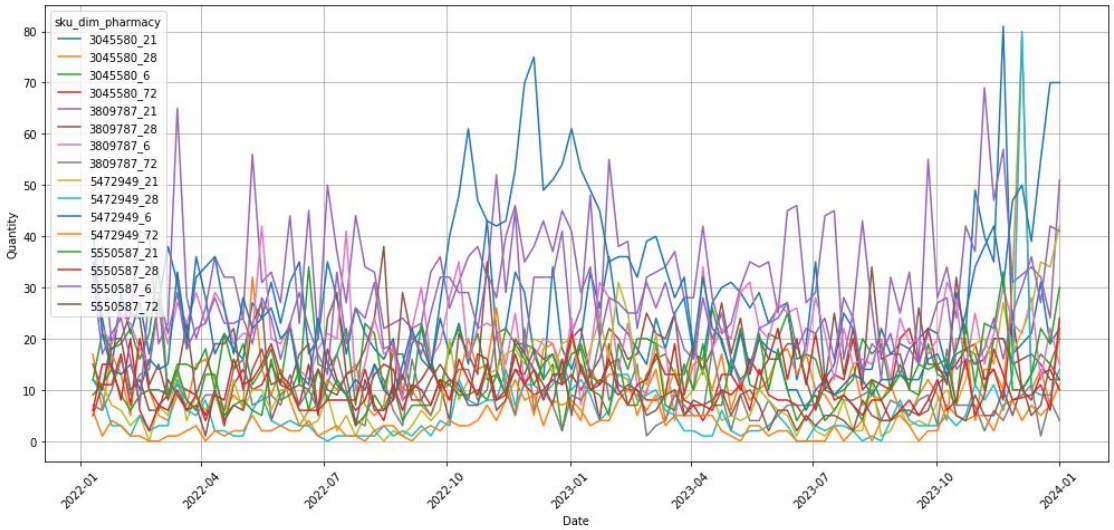


Figure 15: The common top 4 SKUs of the pharmacies previously selected on the vertical approach after outlier correction, excluding SKU '5440987'.

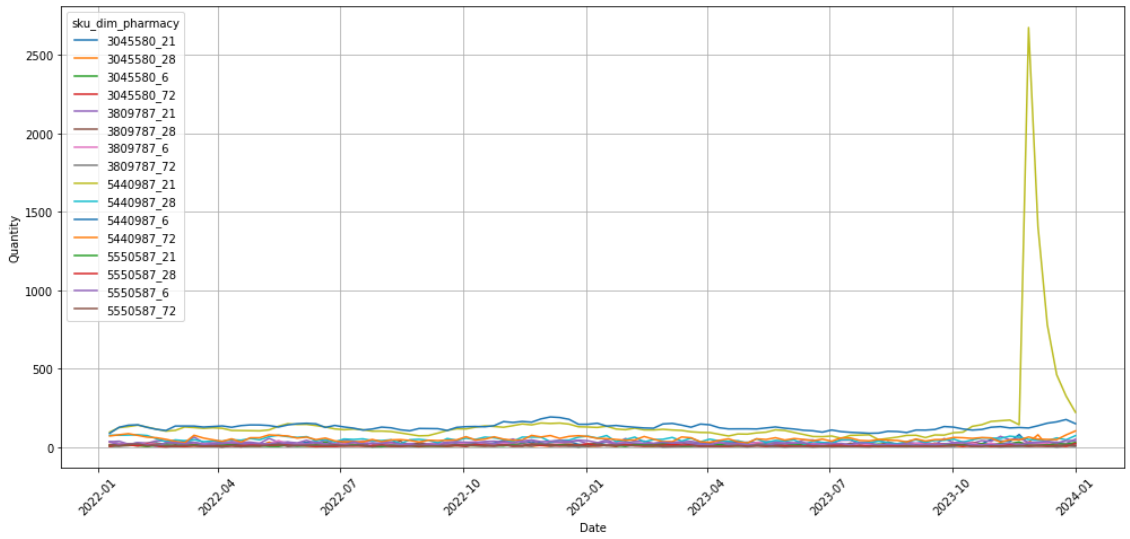


Figure 16: The common top 4 SKUs of the pharmacies previously selected on the vertical approach after outlier correction.

Each unique pharmacy/SKU was fitted on the next step maintaining the same assumptions defined for forecasting process A.

Forecasting process C addresses RQ4 to assess the improvements related to features. It also mirrors process A but it has a few variations. In step 3 ("Data Analysis"), after the last step, the meta-features (the product identifier and the pharmacy identifier) were extracted from the final dataset and two covariates were retrieved and properly pre-processed: publicly online data of daily emergency cases of Portugal from "Portal da Transparência SNS" and holidays. An additional step was inserted ("Data Integration"), where those features were iteratively added to the final dataset. The meta-features as 'static_features', emergency data (ER data) as a 'past_covariate', and holidays as a 'known_covariate'. Different combinations of these four variables were tested to properly assess the best approach for improvement.

Throughout the three processes, careful documentation of each step was an important aspect to ensure reproducibility and future refinement of the forecasting processes.

Ethical Considerations

The data underwent a process of pseudonymization, during which organizationally identifiable information was encrypted to ensure confidentiality. The data was stored in a secure environment with restricted access granted only to authorized personnel.

Results and Discussion

Dataset

The final dataset was characterized. It had 200 637 time series of 85 pharmacies. Descriptive statistics of sales can be found in **Table 3**. Sales were aggregated by each pharmacy and analyzed using descriptive statistics (Appendix II - **Table 6**).

Table 3: Descriptive statistics of sales

Mean	Min	Max	Standard deviation	Total Sales	Nr of unique SKUs	Most sold SKU
1,07	0	2674	3,72	16 128 840	20 219	5440987

Figure 17 shows a general view of the sum of sales across months considering the two years of data. It suggests a seasonal trend of higher sales during the winter months (October to January), which is comprehensible due to the peak of cold-related illnesses, such as respiratory infections.

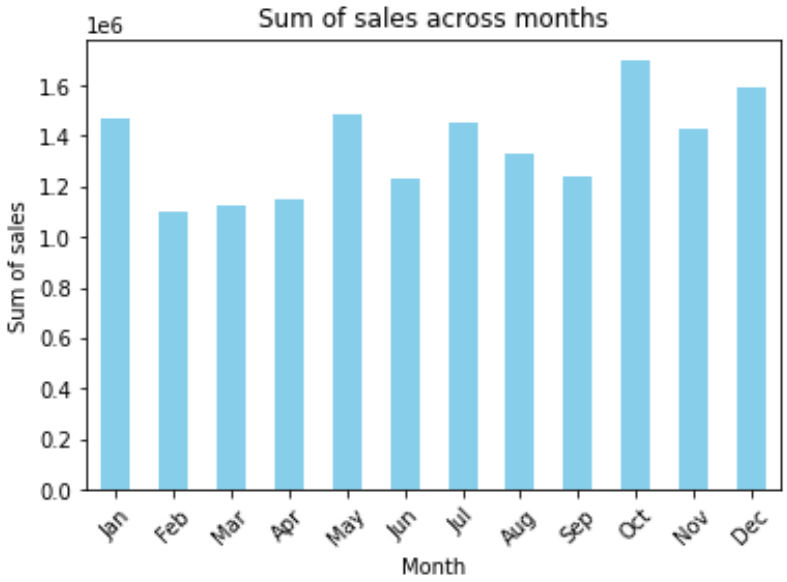


Figure 17: Sum of sales across months (2022 and 2023).

Figure 18 illustrates the sum of sales (represented by bars) and the corresponding number of partnering pharmacies (depicted by lines) on each month over two years. Throughout 2023, there was a consistent surge in sales month by month compared to the previous year. This uptrend can be attributed to the expanding network of pharmacies.

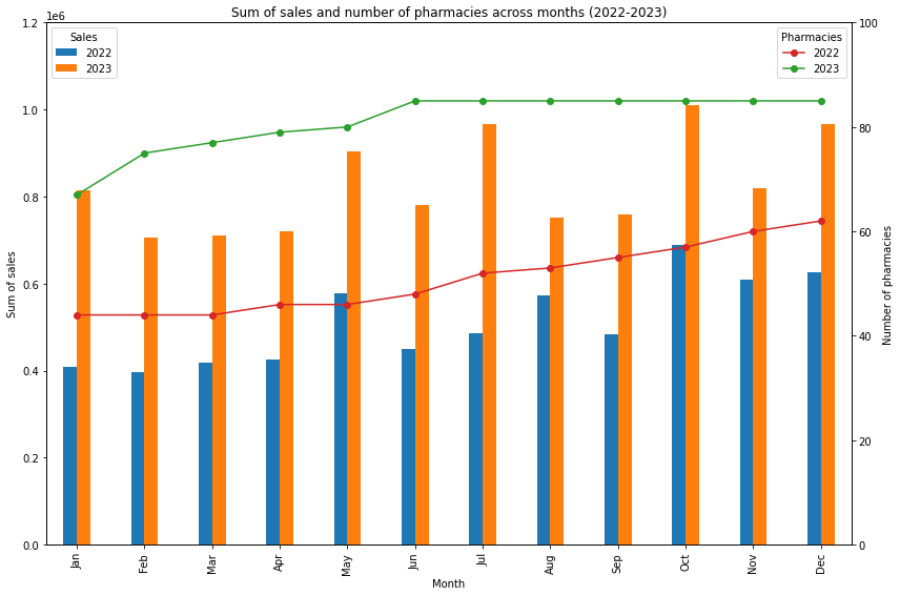


Figure 18: Sum of sales (bars) and number of unique pharmacies (lines) across months (2022 and 2023).

Empirical Results and Discussion

Table 4 displays the training and validation times related to the training phase on AutoGluon-TimeSeries. Distinct differences in training times between global and local models were observed. Notably, global models generally trained faster compared to local models, with a few exceptions: TFT and DeepAR. Among the local models, excluding Seasonal Naïve, Theta was the most efficient, completing training in just 22 minutes, while AutoETS was the most time-consuming, requiring around 32 minutes. For global models, as anticipated from the literature review, DeepAR took the longest time to train (almost 50 minutes), with TFT being the second worst (around 32 minutes). Contrary to the initial anticipation that LightGBM would be the fastest, DLinear outperformed it with a training time of only 8 minutes, which was around 1 minute faster than LightGBM.

Table 4: Training and validation times for each model, excluding Seasonal Naïve

Model	Training (min)	Validation (min)	Total (min)
DLinear	7,34	1,31	8,65
LightGBM	9,11	0,76	9,87
PatchTST	10,29	1,42	11,72
Theta	16,67	5,48	22,15
Weighted Ensemble	12,56	10,74	23,30
AutoETS	23,53	8,06	31,59

TFT	29,63	2,68	32,31
DeepAR	40,33	9,49	49,82

Table 5 shows the leaderboard based on MAE with the extra metrics added along with the corresponding prediction time and displayed in **Figure 19** are the scatterplots comparing MAE vs RMSE and MAE vs WQL for each model, excluding Seasonal Naïve. In line with the empirical studies cited in the literature review, Weighted Ensemble was the best model according to all metrics. In turn, the best single global model (TFT) had an equivalent performance in MAE and WQL, a slightly poorer performance on RMSE (+0,3%), requiring less time to predict but longer to train. In the third position, DeepAR showed a strong performance with significantly greater computational time needed. Theta was the best single local model considering MAE and RMSE. Even though AutoETS performed better considering WQL, it was a neglectable difference from Theta. Seasonal Naïve was the worst local model performer and DLinear was the worst global model performer.

Table 5: Leaderboard based on MAE. Highlighted in **bold** are the best results in each category with the percentage differences compared to the best-performing model enclosed in parentheses

Model	MAE	WQL	RMSE	Prediction (min)
Weighted Ensemble	**0,849**	**0,590**	**2,746**	7,97
TFT	**0,849**	**0,590**	2,755 (+0,3%)	2,75
DeepAR	0,872 (+3%)	0,638 (+8%)	2,840 (+3%)	8,72
Theta	0,919 (+8%)	0,649 (+10%)	3,425 (+25%)	5,26
LightGBM	0,929 (+9%)	0,960 (+63%)	2,776 (+12%)	**0,73**
AutoETS	0,931 (+10%)	0,646 (+9%)	4,056 (+48%)	5,22
PatchTST	0,949 (+12%)	0,714 (+21%)	3,331 (+21%)	1,41
DLinear	1,060 (+25%)	0,797 (+35%)	3,720 (+35%)	1,32
Seasonal Naïve	1,208 (+42%)	0,965 (+64%)	4,329 (+58%)	1,87

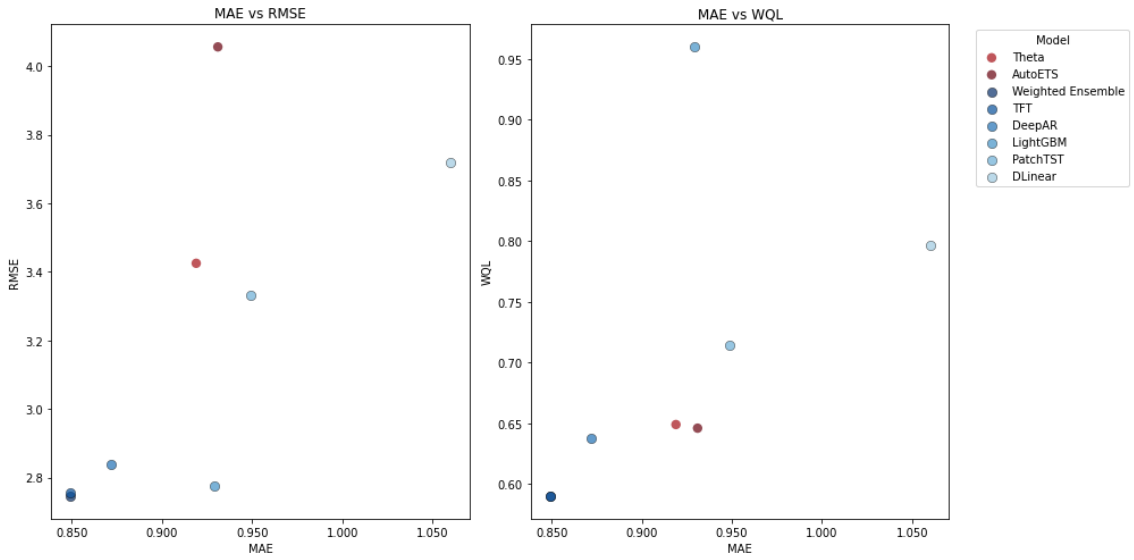


Figure 19: Scatterplots comparing MAE vs RMSE and MAE vs WQL for each model, excluding Seasonal Naïve. Local models are marked in **red** and global models in **blue**.

Figure 20 presents the results of the innovative two approaches (vertical *vs* horizontal) to better understand the improved performance of GFMs: the vertical approach (Pharmacy) and the horizontal approach (SKU) considering the best GFM on each of the eight datasets. It appears that the better performance of GFMs is related to the vertical approach. The complete results are shown in Appendix III - **Figure 22**, **Figure 23**, **Figure 24,** and **Figure 25**.

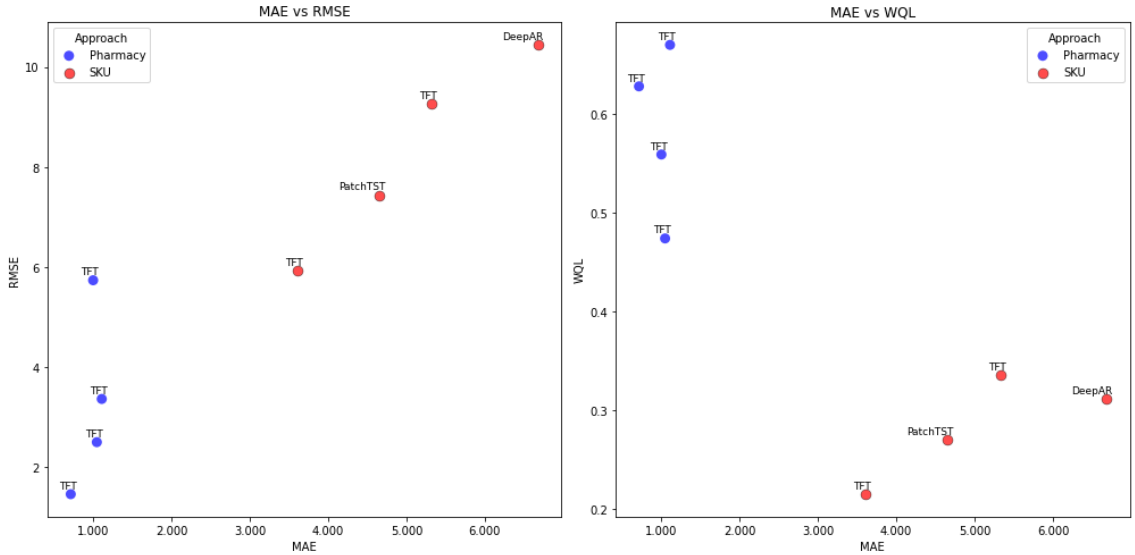


Figure 20*:* Scatterplots comparing MAE vs RMSE and MAE vs WQL for each best GFM on the 8 datasets (4 pharmacies – **vertical approach** - and 4 SKUs – **horizontal approach**).

Table 7 in Appendix IV displays the complete results of adding meta-features and covariates to each model. It yielded varied impacts across models, being PatchTST and DLinear the ones that most benefited from feature integration.

The answers to the RQs are explored bellow.

RQ1: Which GFM is the best fit for the inventory management system to predict the pharmaceuticals' sales?

To answer RQ1, considering the execution time is relevant for the organization, Weighted Ensemble and TFT would be adequate options for the difficult balance between execution time and accuracy. Even though they had the worst performance in terms of training time, the time required was less than 33 minutes, which is quite practicable for the business environment. Besides, the difference in terms of accuracy is extremely relevant. TFT and Weighted Ensemble outperformed the models that trained faster (with at least a difference of 8% considering MAE, 10% considering WQL, and a difference ranging from less than 12% to 58% considering RMSE) and even DeepAR, which required more time to train (with a difference of around 3% considering MAE, 8% considering WQL and 3% considering RMSE).

It is important to note that the machine used in this work had some limitations, such as not having a Graphics Processing Unit. The use of a more powerful machine to run the models can diminish the execution time to a point where those differences can be even less important.

RQ2: How do the GFMs perform in comparison to local forecasting approaches, including the one currently used by the organization?

As displayed in **Figure 19** and in line with the results of the empirical studies and recent competitions cited in the literature review, when comparing local to global models, the overall performance favored GFMs, with a few exceptions. DLinear exhibited poor performance despite having the best total training time. LightGBM also showed poorer performance compared to local models when WQL is considered likely due to its focus on point rather than probabilistic forecasting. When the proxy to the local model the organization is currently using (AutoETS), defined as the benchmark, is compared to GFMs, GFMs outperformed consistently (+10% of MAE, +9% of WQL, and +48% of RMSE compared to the best performing GFM). The only exceptions were DLinear and PatchTST, which lagged in all metrics excluding RMSE, and LightGBM, in the specific case of WQL. In terms of training time, only TFT and DeepAR took more time than AutoETS with a difference of less than 1 minute and 20 minutes, respectively. When it comes to prediction time, only Weighted Ensemble and DeepAR performed worse than AutoETS (+2,75 min and +3,5 minutes, respectively).

RQ3: If GFMs perform better than local models, is it primarily because they better capture the sales patterns of the same product across different pharmacies, or because they better capture the sales patterns of all products within the same pharmacy?

The empiric findings (**Figure 20**) indicate that the superior performance of global models is predominantly associated with the vertical approach rather than the horizontal one across all metrics considered, except for WQL (but the difference was minimal). In other words, GFMs seem to enhance their performance primarily since they can better capture the sales patterns of all products within the same pharmacy, likely due to the larger amount of time series data available. Specifically, in the vertical approach, the minimum number of time series ranged between 2094 and 4907, while in the horizontal approach, the number of time series ranged between 83 and 85. When the complete results (Appendix III) are compared to the baseline in **Figure 19**, it is evident that these approaches generally resulted in worse performance for the models, with a more pronounced difference observed in the horizontal approach. This indicates that the improvements seen in GFMs could be related to the greater number of time series available, essentially scaling up the vertical approach. However, that may not be the only explanation.

RQ4: To what extent do adding meta-features and covariates contribute to the improvement of GFMs?

Overall, GFMs struggled to fully capture the impact of features in a non-negligible way and it even made the models perform worse in some instances, which was not expected. LightGBM was particularly the worst model in capturing those effects. In contrast, PatchTST and DLinear emerged as the frontrunners in effectively leveraging introduced features, showcasing great improvements over baseline. Notably, these were the only GFMs that performed better when all features were integrated across all metrics.

Specifically, TFT demonstrated its greatest improvement in MAE and WQL, when Product ID was added (-1,53%); in RMSE when Holidays + ER data was added (-2,29%) and in WQL for both the combinations Holidays + ER data and Pharmacy ID + Holidays (-1,53%). Notably, RMSE was always negatively affected by introducing features, except for Holidays + ER data and Product ID. Despite holidays being the only feature that showed negative results compared to the baseline when added individually, Holidays + ER data had the best impact on the model when all metrics were considered. For Weighted Ensemble the greatest improvement was -1,41% for MAE and -1,36% for WQL when Product ID was added and

-2,18% for RMSE with the combination of Holidays + ER data. The results were very similar to TFT since it played an important role in every ensemble (79 to 99%).

In contrast, LightGBM failed to capitalize on the impact of features across all metrics. The neglictable improvements were observed on MAE when Holidays and Holidays + ER data were added to the baseline (-0,21%) and on WQL when Pharmacy ID + Holidays were added (-0,21%).

Interestingly, Pharmacy ID emerged as pivotal feature for DeepAR since it was the only feature that alone or in combination with others (except for Product ID), led to improvements over baseline, even though they were very slight. With the addition of Pharmacy ID alone and with the combination Pharmacy ID + ER data, the improvement was -0,69% for MAE; with the addiction of Holidays + ER data the improvement was -1,27% for RMSE, and with the combinations Pharmacy ID + Holidays and Pharmacy ID + Holidays + ER data it was -0,47% for WQL. As for TFT and Weighted Ensemble, Holidays + ER data had the best impact on the model when all metrics were considered.

Overall, DLinear and PatchTST were able to capture the effects of all meta-features and covariates, including all combinations, with great improvements over baseline. For PatchTST, even though Product ID alone caused the worst performance across all metrics, Product ID + Holidays led to the best results over baseline (-8,32% for MAE, -11,56% for RMSE and -9,38% for WQL). For DLinear, the best results were achieved with the combination Pharmacy ID + ER data + Holidays (-6,89% for MAE, -7,98% for RMSE and -9,41% for WQL).

Finally, the incorporation of features had minimal impact on prediction times across all models, suggesting that computational efficiency remained consistent regardless of feature complexity. For further insights and detailed analysis, refer to **Table 7** in Appendix IV.

Managerial insights

The empirical results of this study demonstrated that global models consistently outperformed local approaches for at least 8% considering MAE, 9% for WQL, and 10% for RMSE. There is no simple formula to quantify the importance of these improvements, in terms of cost reductions (by, for example, lower inventory levels and fewer dead stocks) or increased profits (for instance because of higher service levels and no out-of-stock situations). Although it is complex to quantify the economic impact on the specific setting of community pharmacies, industry benchmarks provide useful guidance. According to the McKinsey Global Institute, a 10-20% improvement in forecasting accuracy translates into a

potential 5% reduction in inventory costs and a 2-3% increase in revenue[77]. Similarly, Gartner reported that for every 1% of forecast improvement, companies could achieve a 2,7% reduction in finished goods inventory (days), a 3,2% reduction in transportation costs, and a 3,9% reduction in inventory obsolescence[78]. The Institute of Business Forecasting also stated that a 15% forecasting accuracy improvement can deliver a 3% or higher pre-tax improvement[79].

In the specific context of community pharmacies, improved forecasting can translate into lower inventory levels, fewer dead stocks, and higher service levels, thereby minimizing out-of-stock situations. The robustness, speed, and ease of use of the AutoGluon-TimeSeries framework make it particularly suitable for business environments, allowing organizations to deploy and leverage these tools effectively for inventory management and profit maximization.

Moreover, the integration of data-driven simulations, also called digital twins, empowers managers to simulate and assess the business impacts of different scenarios using real historical data. This capability enables the exploration of what-if scenarios, aiding in strategic decision-making. However, it is essential to recognize that accurate forecasting alone is insufficient. Effective inventory planning requires appropriate inventory policies to complement the use of forecasting models.

Conclusion

In summary, this study provides valuable insights into forecasting retail sales in the context of community pharmacies, marking a significant advancement in comparing the performance of global and local models and offering practical insights for inventory management. The empirical results demonstrated the superior performance of GFMs and highlighted the varying computational efficiencies of different forecasting models, leveraging the practical advantages of the AutoGluon-TimeSeries framework for the business environment. A deeper understanding of GFMs' superior performance indicated that their advantage derives from an enhanced ability to capture the sales patterns of all products within the same pharmacy, likely due to the larger amount of time series data available. The integration of additional features yielded various impacts.

However, certain limitations have been identified. The pseudonymization process, which excluded meta-features such as the location of each community pharmacy, likely constrained forecasting accuracy. Additionally, the reliance on a specific dataset and limited computational resources may restrict the generalizability of the findings.

There remain numerous avenues for future research in this area. Quantifying the economic impact of these performance improvements would be a valuable topic for future research. Further exploration of feature engineering techniques, experimenting with different ways of partitioning the dataset, and overcoming the limitation of random search for hyperparameter tuning could enhance the models' performance. Furthermore, the investigation of new types of global models (pre-trained models) offered by AutoGluon-TimeSeries, namely Chronos, could provide valuable insights into improving forecasting accuracy. Continued research in this area will further refine these models, expanding their applicability and enhancing their impact on business operations.

References

1. Ouyang, Y. (2007). The effect of information sharing on supply chain stability and the bullwhip effect. *European Journal of Operational Research*, *182*(3), 1107–1121. https://doi.org/10.1016/j.ejor.2006.09.037
2. Sodhi, M. S., & Tang, C. S. (2011). The incremental bullwhip effect of operational deviations in an arborescent supply chain with requirements planning. *European Journal of Operational Research*, *215*(2), 374–382. https://doi.org/10.1016/j.ejor.2011.06.019
3. Corsten, D., & Gruen, T. (2003). Desperately seeking shelf availability: An examination of the extent, the causes, and the efforts to address retail out-of-stocks. *International Journal of Retail & Distribution Management*, *31*(12), 605–617. https://doi.org/10.1108/09590550310507731
4. 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., Browell, J., Carnevale, C., Castle, J. L., Cirillo, P., Clements, M. P., Cordeiro, C., Cyrino Oliveira, F. L., De Baets, S., Dokumentov, A., … Ziel, F. (2022). Forecasting: theory and practice. *International Journal of Forecasting*, *38*(3), 705–871. https://doi.org/10.1016/j.ijforecast.2021.11.001
5. Huang, T., Fildes, R., & Soopramanien, D. (2019). Forecasting retailer product sales in the presence of structural change. *European Journal of Operational Research*, *279*(2), 459–470. https://doi.org/10.1016/j.ejor.2019.06.011
6. Montgomery, Douglas; Jennings, Cheryl; Kulahci, M. (2016). *Introduction to time series analysis and forecasting (second edition)* (2nd ed.). Wiley.
7. Sharda, R., Delen, D., & Turban, E. (2018). Business Intelligence, Analytics, and Data Science: A Managerial Perspective. In *Winning with Data* (4th ed.). Pearson Education Limited.
8. INFORMS. (2019). *INFORMS Analytics Body of Knowledge*. Wiley.
9. Manu, J. (2022). *Modern Time Series Forecasting With Python* (1st ed.). Packt Publishing.
10. Hyndman, R.J.; Athanasopoulos, G. (2021). *Forecasting: principles and practice (3rd ed.)*. http://otexts.com/fpp3
11. Croston, J. D. (1972). Forecasting and Stock Control for Intermittent Demands. *Operational Research Quarterly (1970-1977)*, *23*(3), 289–303. https://doi.org/10.2307/3007885
12. Januschowski, T., Gasthaus, J., Wang, Y., Salinas, D., Flunkert, V., Bohlke-Schneider, M., & Callot, L. (2020). Criteria for classifying forecasting methods. *International Journal of*

Forecasting, *36*(1), 167–177. https://doi.org/10.1016/j.ijforecast.2019.05.008

13. Manu, J. (2022). *Modern Time Series Forecasting With Python* (1st ed.). Packt Publishing.
14. 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., Browell, J., Carnevale, C., Castle, J. L., Cirillo, P., Clements, M. P., Cordeiro, C., Cyrino Oliveira, F. L., De Baets, S., Dokumentov, A., … Ziel, F. (2022). Forecasting: theory and practice. *International Journal of Forecasting*, *38*(3), 705–871. https://doi.org/10.1016/j.ijforecast.2021.11.001
15. Müller, A. C., & Guido, S. (2017). *Introduction to Machine Learning with Python* (D. Schanafelt, Ed.; 1st ed.). O'Reilly Media.
16. Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow* (N. Tache, Ed.; 2nd ed.). O'Reilly Media, Inc.
17. Szabłowski, B. (2023). *Forecast Multiple Time Series Like a Master*. Towards Data Science. https://medium.com/towards-data-science/forecast-multiple-time-series-like-a-master-1579a2b6f18d
18. Januschowski, T., Gasthaus, J., Wang, Y., Salinas, D., Flunkert, V., Bohlke-Schneider, M., & Callot, L. (2020). Criteria for classifying forecasting methods. *International Journal of Forecasting*, *36*(1), 167–177. https://doi.org/10.1016/j.ijforecast.2019.05.008
19. Bandara, K., Bergmeir, C., & Smyl, S. (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, *140*. https://doi.org/10.1016/j.eswa.2019.112896
20. Bandara, K. (2023). Forecasting with Big Data Using Global Forecasting Models. In M. Hamoudia, S. Makridakis, & E. Spiliotis (Eds.), *Forecasting with Artificial Intelligence. Palgrave Advances in the Economics of Innovation and Technology.* (pp. 107–122). Palgrave Macmillan, Cham. https://doi.org/https://doi.org/10.1007/978-3-031-35879-1_5
21. Godahewa, R., Bandara, K., Webb, G. I., Smyl, S., & Bergmeir, C. (2021). Ensembles of localised models for time series forecasting. *Knowledge-Based Systems*, *233*. https://doi.org/10.1016/j.knosys.2021.107518
22. Montero-Manso, P., & Hyndman, R. J. (2021). Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting*, *37*(4), 1632–1653. https://doi.org/10.1016/j.ijforecast.2021.03.004
23. Oliveira, J. M., & Ramos, P. (2023). Investigating the Accuracy of Autoregressive Recurrent Networks Using Hierarchical Aggregation Structure-Based Data Partitioning. *Big Data and Cognitive Computing*, *7*(100). https://doi.org/10.20944/preprints202304.0222.v1

24. Ma, S., & Fildes, R. (2021). Retail sales forecasting with meta-learning. *European Journal of Operational Research*, *288*(1), 111–128. https://doi.org/10.1016/j.ejor.2020.05.038

25. Duncan, G. T., Gorr, W. L., & Szczypula, J. (2001). *Forecasting Analogous Time Series*. 195–213. https://doi.org/10.1007/978-0-306-47630-3_10

26. Breiman, L. (2001). Random Forests. *Machine Learning*, *45*, 5–32. https://doi.org/https://doi.org/10.1023/A:1010933404324

27. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-Augu*, 785–794. https://doi.org/10.1145/2939672.2939785

28. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye1, Q., & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*. https://doi.org/10.1109/ICCSE.2019.8845529

29. CatBoostTeam. (n.d.). *CatBoost*. https://catboost.ai/en/docs/

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

31. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

32. Rangapuram, S. S., Seeger, M., Gasthaus, J., Stella, L., Wang, Y., & Januschowski, T. (2018). Deep state space models for time series forecasting. *Advances in Neural Information Processing Systems*, *2018-Decem*(NeurIPS), 7785–7794.

33. Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, *36*(3), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

34. Bai, S., Kolter, J. Z., & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *ArXiv*.

35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*. https://doi.org/https://doi.org/10.48550/arXiv.1706.03762

36. Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-Beats: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting. *ArXiv*, 1–31.

37. Lim, B., Arık, S., Loeff, N., & Pfister, T. (2021). Temporal Fusion Transformers for

interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, *37*(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

38. Challu, C., Olivares, K. G., Oreshkin, B. N., Ramirez, F. G., Mergenthaler-Canseco, M., & Dubrawski, A. (2023). N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting. *ArXiv*, *37*, 6989–6997. https://doi.org/10.1609/aaai.v37i6.25854
39. Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are Transformers Effective for Time Series Forecasting? *ArXiv*, *37*, 11121–11128. https://doi.org/10.1609/aaai.v37i9.26317
40. Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., & Yu, R. (2023). *Long-term Forecasting with TiDE: Time-series Dense Encoder*. 1–21.
41. Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. .
42. Montero-Manso, P., & Hyndman, R. J. (2021). Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting*, *37*(4), 1632–1653. https://doi.org/10.1016/j.ijforecast.2021.03.004
43. Štěpnička, M., & Burda, M. (2017). On the results and observations of the time series forecasting competition CIF 2016. *IEEE International Conference on Fuzzy Systems*. https://doi.org/10.1109/FUZZ-IEEE.2017.8015455
44. Knauer, F., & Cukierski, W. (2015). *Rossmann Store Sales*. Kaggle. https://kaggle.com/competitions/rossmann-store-sales
45. Maggie, Anava, O., Kuznetsov, V., & Cukierski, W. (2017). *Web Traffic Time Series Forecasting*. Kaggle. https://kaggle.com/competitions/web-traffic-time-series-forecasting
46. He, W., Liu, P., & Qian, Q. (2023). Case Study : Corporación Favorita Grocery Sales Forecasting. In *Machine Learning Contests: A Guidebook*. Springer. https://doi.org/https://doi.org/10.1007/978-981-99-3723-3_11
47. Cortial, K. (2021). *Challenge Intermarché: Prédiction des volumes de ventes des produits de grande consommation*. OpenStudio. https://www.openstudio.fr/2021/06/29/challenge-intermarche-prediction-des-volumes-de-ventes-des-produits-de-grande-consommation/
48. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, *38*(4). https://doi.org/10.1016/j.ijforecast.2021.11.013
49. Bojer, C. S., & Meldgaard, J. P. (2021). Kaggle forecasting competitions: An overlooked

learning opportunity. *International Journal of Forecasting*, *37*(2), 587–603. https://doi.org/10.1016/j.ijforecast.2020.07.007

50. Hewamalage, H., Bergmeir, C., & Bandara, K. (2022). Global models for time series forecasting: A Simulation study. *Pattern Recognition*, *124*. https://doi.org/10.1016/j.patcog.2021.108441
51. Guennioui, O., Chiadmi, D., & Amghar, M. (2023). Global Stock Price Forecasting during a Period of Market Stress using LightGBM. *International Journal of Computing and Digital Systems*, *14*(1). https://doi.org/https://dx.doi.org/10.12785/ijcds/XXXXXX
52. Ramos, P., & Oliveira, J. M. (2023). *Robust Sales Forecasting Using Deep Learning with Static and Dynamic Covariates*. 1–13.
53. Grabner, M., Wang, Y., Wen, Q., Blazic, B., & Struc, V. (2023). A Global Modeling Framework for Load Forecasting in Distribution Networks. *IEEE Transactions on Smart Grid*, 1–12. https://doi.org/10.1109/TSG.2023.3264525
54. Rafayal, S., Cevik, M., & Kici, D. (2022). An empirical study on probabilistic forecasting for predicting city-wide electricity consumption. *Proceedings of the Canadian Conference on Artificial Intelligence*, *October*. https://doi.org/10.21428/594757db.8e8477a9
55. Mbouopda, M. F., Guyet, T., Labroche, N., & Henriot, A. (2023). Experimental Study of Time Series Forecasting Methods for Groundwater Level Prediction. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *13812 LNAI*, 34–49. https://doi.org/10.1007/978-3-031-24378-3_3
56. Laptev, N., Yosinski, J., Erran Li, L., & Smyl, S. (2017). Time-series Extreme Event Forecasting with Neural Networks at Uber. *International Conference on Machine Learning - Time Series Workshop*, 1–5.
57. Grecov, P., Bandara, K., Bergmeir, C., Ackermann, K., Campbell, S., Scott, D., & Lubman, D. (2021). Causal Inference Using Global Forecasting Models for Counterfactual Prediction. In K. K (Ed.), *Advances in Knowledge Discovery and Data Mining* (Vol. 12713, pp. 282–294). Springer Nature. https://doi.org/10.1007/978-3-030-75765-6_29
58. Ibañez, S. C., & Monterola, C. P. (2023). A Global Forecasting Approach to Large-Scale Crop Production Prediction with Time Series Transformers. *Agriculture*, *13*(9). https://doi.org/https://doi.org/10.3390/agriculture13091855
59. Rathnayaka, P., Moraliyage, H., Mills, N., De Silva, D., & Jennings, A. (2022). Specialist vs Generalist: A Transformer Architecture for Global Forecasting Energy Time Series. *International Conference on Human System Interaction, HSI.*

https://doi.org/10.1109/HSI55341.2022.9869463

60. Hertel, M., Beichter, M., Heidrich, B., Neumann, O., Schäfer, B., Mikut, R., & Hagenmeyer, V. (2023). Transformer Training Strategies for Forecasting Multiple Load Time Series. *ArXiv*, 1–15. https://doi.org/10.1186/s42162-023-00278-z

61. Kunz, M., Birr, S., Raslan, M., Ma, L., Li, Z., Gouttes, A., Koren, M., Naghibi, T., Stephan, J., Bulycheva, M., Grzeschik, M., Kekić, A., Narodovitch, M., Rasul, K., Sieber, J., & Januschowski, T. (2023). *Deep Learning based Forecasting: a case study from the online fashion industry*. http://arxiv.org/abs/2305.14406

62. Deforce, B., Baesens, B., & Diels, J. (2022). Forecasting Sensor-data in Smart Agriculture with Temporal Fusion Transformers. In *Transactions on Computational Science & Computational Intelligence* (pp. 1–10). Springer Nature.

63. Fazli, M., & Shakeri, H. (2022). Leveraging Wastewater Monitoring for COVID-19 Forecasting in the US: a Deep Learning study. *ArXiv*.

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

65. Weng, T., Liu, W., & Xiao, J. (2020). Supply chain sales forecasting based on lightGBM and LSTM combination model. *Industrial Management and Data Systems*, *120*(2), 265–279. https://doi.org/10.1108/IMDS-03-2019-0170

66. Montero-Manso, P., & Hyndman, R. J. (2021). Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting*, *37*(4), 1632–1653. https://doi.org/10.1016/j.ijforecast.2021.03.004

67. Mbouopda, M. F., Guyet, T., Labroche, N., & Henriot, A. (2023). Experimental Study of Time Series Forecasting Methods for Groundwater Level Prediction. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *13812 LNAI*, 34–49. https://doi.org/10.1007/978-3-031-24378-3_3

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

69. Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., & Yu, R. (2023). *Long-term Forecasting with TiDE: Time-series Dense Encoder*. 1–21. http://arxiv.org/abs/2304.08424

70. Hewamalage, H., Bergmeir, C., & Bandara, K. (2022). Global models for time series forecasting: A Simulation study. *Pattern Recognition*, *124*. https://doi.org/10.1016/j.patcog.2021.108441

71. Bandara, K., Shi, P., Bergmeir, C., Hansika, H., Quoc, T., & Seaman, B. (2019). Sales Demand Forecast in E-commerce using a Long Short-Term Memory Neural Network Methodology. *ArXiv*. https://doi.org/https://doi.org/10.48550/arXiv.1901.04028

72. Grabner, M., Wang, Y., Wen, Q., Blazic, B., & Struc, V. (2023). A Global Modeling Framework for Load Forecasting in Distribution Networks. *IEEE Transactions on Smart Grid*, 1–12. https://doi.org/10.1109/TSG.2023.3264525

73. Hewamalage, H., Ackermann, K., & Bergmeir, C. (2023). Forecast evaluation for data scientists: common pitfalls and best practices. *Data Mining and Knowledge Discovery*, *37*(2), 788–832. https://doi.org/10.1007/s10618-022-00894-5

74. Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679 – 688. https://doi.org/https://doi.org/10.1016/j.ijforecast.2006.03.001

75. Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., & Smola, A. (2020). *AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data*. http://arxiv.org/abs/2003.06505

76. Shchur, O., Turkmen, C., Erickson, N., Shen, H., Shirkov, A., Hu, T., & Wang, Y. (2023). *AutoGluon-TimeSeries: AutoML for Probabilistic Time Series Forecasting*. http://arxiv.org/abs/2308.05566

77. Chui, M., Manyika, J., Miremadi, M., Chung, R., Nel, P., Malhotra, S., & Henke, N. (2018). *Notes from the AI frontier insights: insights from hundreds of use cases*. https://www.mckinsey.com/~/media/mckinsey/featured%20insights/artificial%20intelligence/notes%20from%20the%20ai%20frontier%20applications%20and%20value%20of%20deep%20learning/notes-from-the-ai-frontier-insights-from-hundreds-of-use-cases-discussion-paper.ashx

78. Steutermann, S. (2017). *Magic Quadrant for Data Science and Machine Learning Platforms*. https://www.gartner.com/en/documents/3701817

79. Wilson, E. (2018, July 12). *How much does forecasting software cost, & how much will it save?* https://demand-planning.com/2018/07/12/how-much-does-forecasting-software-cost/

80. Guennioui, O., Chiadmi, D., & Amghar, M. (2023). Global Stock Price Forecasting during a Period of Market Stress using LightGBM. *International Journal of Computing and Digital Systems*, *14*(1). https://doi.org/https://dx.doi.org/10.12785/ijcds/XXXXXX

81. Weng, T., Liu, W., & Xiao, J. (2020). Supply chain sales forecasting based on lightGBM and LSTM combination model. *Industrial Management and Data Systems*, *120*(2), 265–279.

https://doi.org/10.1108/IMDS-03-2019-0170

82. Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, *36*(3), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

83. Rafayal, S., Cevik, M., & Kici, D. (2022). An empirical study on probabilistic forecasting for predicting city-wide electricity consumption. *Proceedings of the Canadian Conference on Artificial Intelligence*, *October*. https://doi.org/10.21428/594757db.8e8477a9

84. Grecov, P., Bandara, K., Bergmeir, C., Ackermann, K., Campbell, S., Scott, D., & Lubman, D. (2021). Causal Inference Using Global Forecasting Models for Counterfactual Prediction. In K. K (Ed.), *Advances in Knowledge Discovery and Data Mining* (Vol. 12713, pp. 282–294). Springer Nature. https://doi.org/10.1007/978-3-030-75765-6_29

85. Ibañez, S. C., & Monterola, C. P. (2023). A Global Forecasting Approach to Large-Scale Crop Production Prediction with Time Series Transformers. *Agriculture*, *13*(9). https://doi.org/https://doi.org/10.3390/agriculture13091855

86. Rathnayaka, P., Moraliyage, H., Mills, N., De Silva, D., & Jennings, A. (2022). Specialist vs Generalist: A Transformer Architecture for Global Forecasting Energy Time Series. *International Conference on Human System Interaction, HSI*. https://doi.org/10.1109/HSI55341.2022.9869463

87. Hertel, M., Beichter, M., Heidrich, B., Neumann, O., Schäfer, B., Mikut, R., & Hagenmeyer, V. (2023). Transformer Training Strategies for Forecasting Multiple Load Time Series. *ArXiv*, 1–15. https://doi.org/10.1186/s42162-023-00278-z

88. Deforce, B., Baesens, B., & Diels, J. (2022). Forecasting Sensor-data in Smart Agriculture with Temporal Fusion Transformers. In *Transactions on Computational Science & Computational Intelligence* (pp. 1–10). Springer Nature.

89. Fazli, M., & Shakeri, H. (2022). Leveraging Wastewater Monitoring for COVID-19 Forecasting in the US: a Deep Learning study. *ArXiv*. http://arxiv.org/abs/2212.08798

90. Ramos, P., & Oliveira, J. M. (2023). *Robust Sales Forecasting Using Deep Learning with Static and Dynamic Covariates*. 1–13. www.preprints.org

91. Bandara, K., Shi, P., Bergmeir, C., Hansika, H., Quoc, T., & Seaman, B. (2019). Sales Demand Forecast in E-commerce using a Long Short-Term Memory Neural Network Methodology. *ArXiv*. https://doi.org/https://doi.org/10.48550/arXiv.1901.04028

92. Bandara, K., Bergmeir, C., & Smyl, S. (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach.

Expert Systems with Applications, *140*. https://doi.org/10.1016/j.eswa.2019.112896

Appendix I

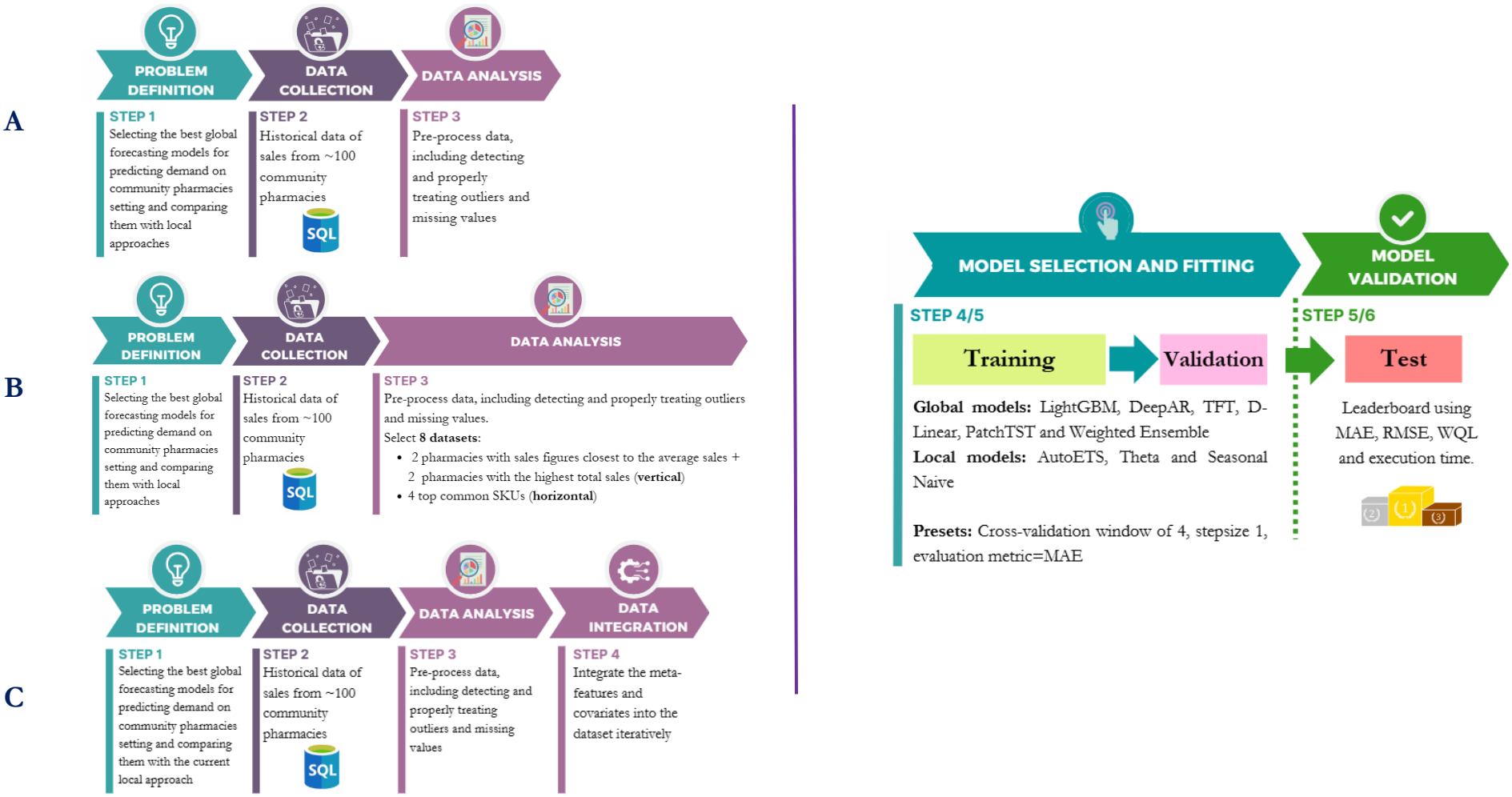


Figure 21: The three forecasting processes that were followed.

Appendix II

Table 6: Descriptive statistic of sales aggregated by each pharmacy

Pharmacy identifier	Mean	Min	Max	Total Sales	Number of SKUs	Most sold SKU
21	1,30	0	2674	592844	4907	1026948
6	1,57	0	691	550414	3784	6057463
113	1,43	0	245	532599	4081	5440987
7	1,43	0	705	472236	3607	2000200
108	1,99	0	1884	453592	2333	5440987
80	1,26	0	253	438523	3717	5440987
25	1,11	0	271	383293	3760	5440987
37	1,11	0	748	374738	3651	6399477
67	1,56	0	141	355503	4021	6399469
9	1,13	0	261	332492	3162	1065564
46	1,18	0	138	327579	3400	5440987
103	1,04	0	394	322374	3337	5440987
52	1,43	0	132	320569	2998	5440987
36	1,31	0	122	313437	2497	5440987
49	1,06	0	208	310166	3137	5440987
101	1,07	0	273	307035	3049	5440987

91	1,06	0	633	302919	3080	1000031
17	1,09	0	175	298549	2977	5440987
11	0,98	0	149	288979	3172	1111111
44	1,16	0	150	285477	2676	1100097
45	1,14	0	203	252101	2605	1011510
58	1,25	0	134	237782	2667	5440987
126	0,95	0	204	231964	2606	5440987
38	0,91	0	292	227344	2672	5440987
14	1,02	0	112	224480	2350	5440987
55	1,14	0	148	223501	2751	1006254
102	1,11	0	126	211888	2089	5440987
69	1,14	0	162	207727	3269	5440987
4	0,98	0	105	205261	2270	5440987
26	0,86	0	430	205255	2559	1017962
74	1,20	0	172	202801	3493	5440987
32	1,01	0	272	194293	2083	5440987
28	0,87	0	80	192609	2375	5440987
72	0,98	0	104	191829	2094	5440987
3	1,01	0	136	184925	1959	5440987
59	0,87	0	134	184860	2357	5440987

23	1,06	0	202	182698	1867	5440987
27	0,90	0	104	177849	2122	5440987
119	0,77	0	116	177468	2457	5440987
5	0,89	0	77	177440	2164	5440987
71	1,17	0	138	168793	2775	5440987
31	0,77	0	316	165472	2273	6399469
90	0,89	0	154	160692	1964	159672
100	0,84	0	79	155339	1977	5440987
68	0,99	0	219	155260	2866	5440987
78	1,20	0	80	145990	2544	5118534
76	1,09	0	118	144412	2730	5440987
48	0,84	0	78	143775	1805	5440987
40	1,01	0	344	143755	1554	1000000
61	0,94	0	76	140916	2357	5440987
112	1,53	0	182	140041	3321	5440987
60	1,06	0	97	133421	1927	5440987
79	1,18	0	103	132211	2453	1019984
8	0,89	0	314	132073	1660	1002393
19	0,81	0	68	123715	1617	5440987
33	0,83	0	67	123366	1629	5440987

97	1,33	0	148	121552	2336	5440987
53	0,82	0	77	114578	1914	1009720
42	0,71	0	90	111941	1667	1008516
88	1,09	0	126	110832	2351	5440987
111	0,79	0	77	109064	1510	5440987
120	1,27	0	81	108558	1724	5280045
86	0,95	0	98	104174	2507	5440987
73	0,90	0	975	98268	2187	5440987
104	1,29	0	225	89613	2282	5440987
50	0,73	0	96	87420	1562	5440987
89	1,06	0	84	84544	1841	5440987
110	0,56	0	112	84306	1727	5440987
98	1,14	0	136	82181	1863	5440987
81	1,03	0	109	77874	1703	5440987
54	0,73	0	77	76470	1509	5288725
63	0,91	0	65	66747	1186	5440987
105	1,00	0	124	65807	2119	5440987
43	0,51	0	69	64020	1374	1005330
66	0,72	0	60	63654	1512	5440987
107	1,29	0	1282	61420	1655	5440987

65	0,68	0	66	56099	1403	5440987
87	0,86	0	118	54720	1490	5440987
92	0,74	0	58	51708	1731	5440987
85	0,82	0	48	41683	1148	3854585
82	0,75	0	59	39913	1252	1009340
99	0,75	0	88	37406	1295	5440987
109	0,85	0	90	33696	1444	5440987
83	0,54	0	38	32715	1406	5440987
106	0,82	0	68	31253	1332	5440987

Appendix III

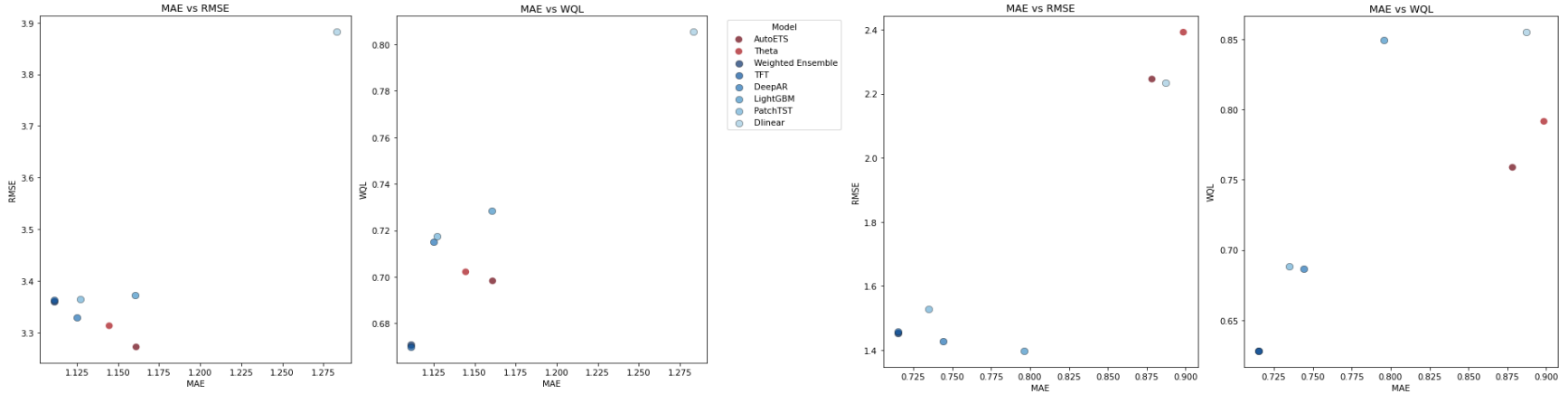


Figure 22: Scatterplots comparing MAE vs RMSE and MAE vs WQL for each model, excluding Seasonal Naïve, on pharmacy 72 dataset (left) and pharmacy 28 dataset (right). Local models are marked in **red** and global models in **blue**.

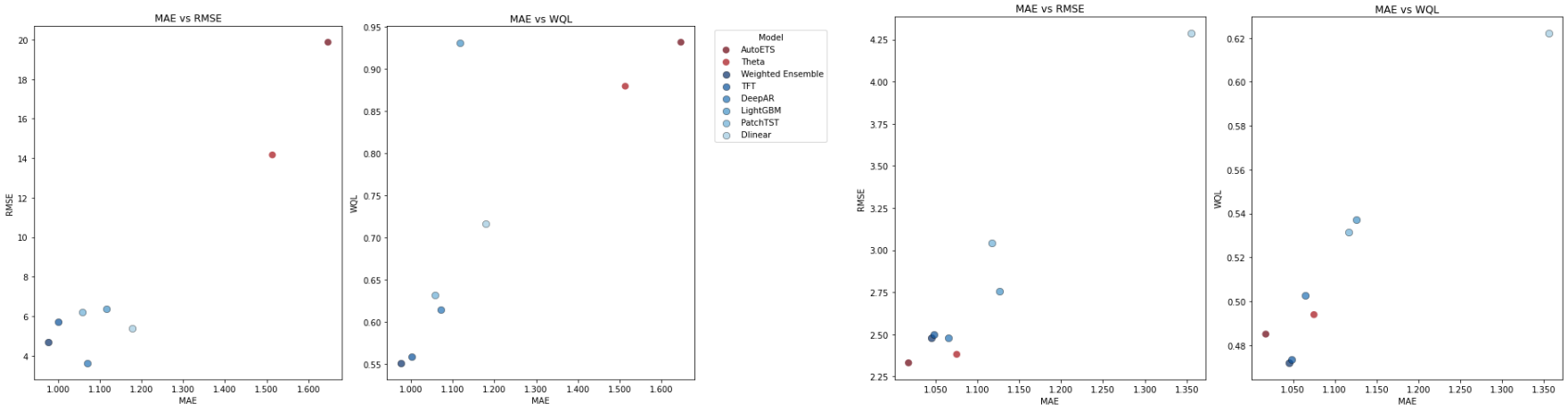


Figure 23: Scatterplots comparing MAE vs RMSE and MAE vs WQL for each model, excluding Seasonal Naïve, on pharmacy 21 dataset (left) and pharmacy 6 dataset (right). Local models are marked in **red** and global models in **blue**.

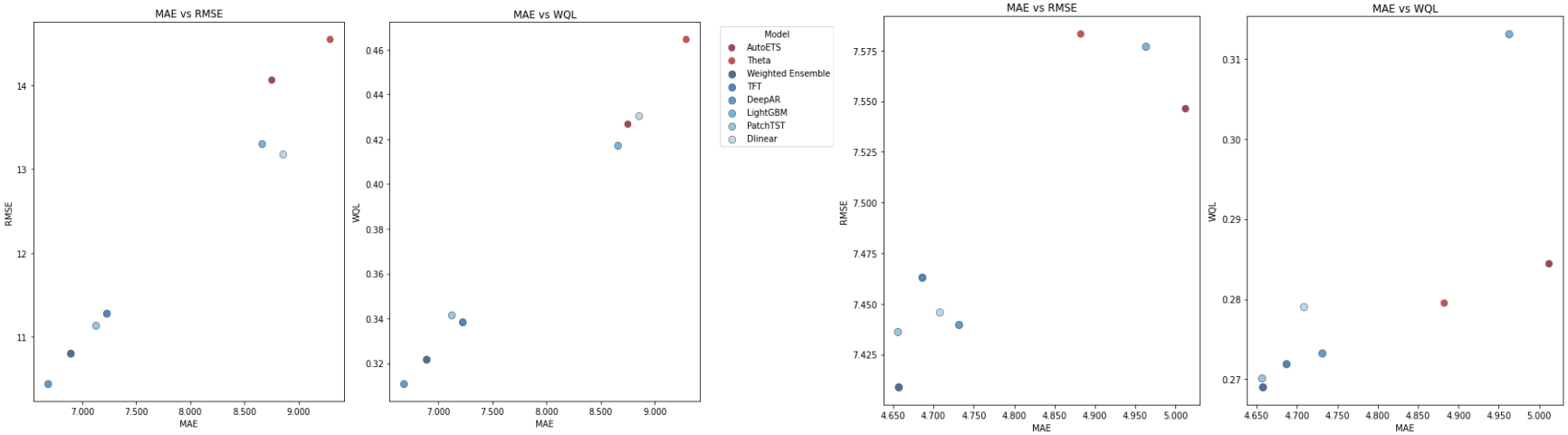


Figure 24: Scatterplots comparing MAE vs RMSE and MAE vs WQL for each model, excluding Seasonal Naïve, on SKU 5472949 dataset (left) and SKU 3809787 dataset (right). Local models are marked in **red** and global models in **blue**.

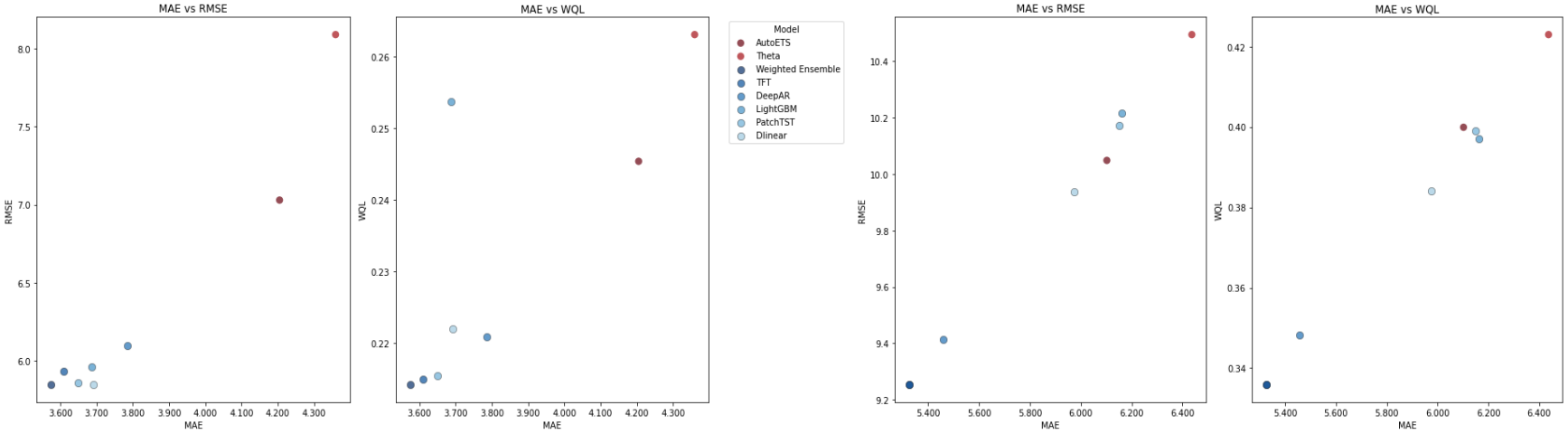


Figure 25: Scatterplots comparing MAE vs RMSE and MAE vs WQL for each model, excluding Seasonal Naïve, on SKU 3045580 dataset (left) and SKU 5550587 dataset (right). Local models are marked in **red** and global models in **blue**.

Appendix IV

Table 7: Results of adding meta-features (Product ID and Pharmacy ID) and covariates (Holidays and ER data) on each model. The better result for each model-metric is marked in **bold** and the worst results compared to baseline are marked in **red**

	Weighted Ensemble				TFT				DeepAR				LightGBM				PatchTST				DLinear			
	MAE	RMSE	WQL	PT (min)	MAE	RMSE	WQL	PT (min)	MAE	RMSE	WQL	PT (min)	MAE	RMSE	WQL	PT (min)	MAE	RMSE	WQL	PT (min)	MAE	RMSE	WQL	PT (min)
Baseline	0,849	2,746	0,590	7,97	0,849	2,755	0,590	2,75	0,872	2,840	0,638	8,72	0,929	**2,776**	0,960	0,73	0,949	3,331	0,714	1,41	1,060	3,720	0,797	1,32
Product ID	**0,837**	2,746	**0,582**	8,91	**0,836**	2,753	0,582	2,90	1,154	4,130	0,999	8,71	0,928	2,802	0,960	0,77	0,990	3,487	0,764	1,46	1,047	3,664	0,779	1,36
Pharmacy ID	0,845	2,767	0,591	22,53	0,847	2,809	0,590	3,81	**0,866**	2,810	0,637	11,24	0,933	2,805	0,960	1,03	0,875	2,959	0,656	1,93	1,007	3,490	0,745	1,97
ER data	0,849	2,821	0,591	20,27	0,849	2,840	0,590	3,51	0,867	2,847	0,644	9,03	0,929	**2,776**	0,960	0,97	0,887	3,047	0,659	1,91	1,025	3,573	0,763	1,77
Holidays	0,851	2,763	0,596	20,38	0,856	2,793	0,595	3,78	0,871	2,818	0,636	9,88	**0,927**	2,788	0,960	0,86	0,883	3,035	0,663	1,98	1,039	3,630	0,779	1,70
Product ID + Pharmacy ID	100% TFT				0,838	2,776	0,582	2,72	1,116	3,959	0,861	8,44	0,935	2,866	0,959	0,69	0,903	3,142	0,677	1,86	1,051	3,677	0,801	1,25
Product ID + ER data	0,848	2,831	0,590	16,35	0,849	2,849	0,591	4,23	1,154	4,130	0,999	10,43	0,928	2,802	0,960	0,85	0,999	3,543	0,744	1,50	1,030	3,590	0,770	1,38
Product ID + Holidays	0,845	2,767	0,586	7,93	0,844	2,795	0,586	2,80	1,142	4,070	0,932	8,36	0,929	2,810	0,960	0,66	**0,870**	**2,946**	**0,647**	1,40	1,023	3,552	0,752	1,27
Pharmacy ID + ER data	0,853	2,802	0,600	12,27	0,855	2,837	0,599	3,38	**0,866**	2,814	0,636	8,88	0,933	2,805	0,960	0,83	0,942	3,299	0,706	1,51	1,062	3,723	0,796	1,37
Pharmacy ID + Holidays	100% TFT				0,838	2,768	**0,581**	3,04	0,870	2,812	**0,635**	8,78	0,930	2,781	**0,958**	0,85	0,966	3,394	0,724	1,47	1,000	3,461	0,732	1,35
Holidays + ER data	0,840	**2,686**	0,583	27,75	0,839	**2,692**	**0,581**	4,85	0,870	**2,804**	0,636	13,33	**0,927**	2,788	0,960	1,03	0,950	3,324	0,707	1,39	1,041	3,628	0,773	1,88
Product ID + Pharmacy ID + ER data	0,842	2,802	0,587	110,07	0,843	2,824	0,588	3,35	1,115	3,949	0,861	8,39	0,935	2,866	0,959	0,64	0,898	3,123	0,671	1,90	1,010	3,503	0,742	1,33
Product ID + Pharmacy ID + Holidays	0,846	2,808	0,586	8,27	0,846	2,840	0,586	3,08	1,148	4,097	0,947	8,23	0,934	2,863	0,960	0,82	0,931	3,273	0,695	1,40	1,022	3,564	0,756	1,27

Product ID + ER data + Holidays	0,846	2,805	0,587	10,03	0,845	2,815	0,587	3,85	1,143	4,071	0,932	9,53	0,929	2,810	0,960	0,76	0,928	3,247	0,696	1,54	1,020	3,551	0,758	1,40
Pharmacy ID + ER data + Holidays	0,850	2,813	0,596	23,20	0,854	2,869	0,594	4,93	0,870	2,812	**0,635**	8,93	0,930	2,781	0,959	0,75	0,881	3,021	0,657	1,52	**0,987**	**3,423**	**0,722**	1,81
All features	0,842	2,797	0,587	8,94	0,842	2,813	0,588	3,35	1,147	4,096	0,947	8,28	0,934	2,863	0,960	0,80	0,894	3,085	0,682	1,42	1,034	3,614	0,779	1,30

Annex I

Model	NN5		WWT		Mixture	
	SMAPE	MASE	SMAPE	MASE	SMAPE	MASE
ARIMA	24.95	0.97	50.36	1.48	41.56	1.26
PR(70)	22.52	0.89	49.30	1.23	45.20	1.44
LGBM(70)	**21.11**	**0.83**	**44.87**	**1.17**	**36.97**	**1.14**
FFNN(70)	21.99	0.87	45.03	**1.17**	38.03	1.16
RNN(70)	22.05	0.87	44.94	**1.17**	37.79	1.16

Table 15
Computational times comparison of the forecasting models (in seconds).

Model	Data Preprocessing	Model Training & Testing	Total
RNN(15)	5.36	10788.62	10793.98
FFNN(15)	5.36	649.73	655.09
LGBM(15)	5.36	4.58	9.94
AR(15)	-	111.08	111.08
SETAR	-	3.03	3.03
PR(15)	0.05	0.40	0.45

Figure 26: Global models for time series forecasting: A Simulation study. Original from Bandara et al [70].

Method	RMSE	MAE	R2 score
LightGBM	0.043	0.031	0.820
Optimized LightGBM	0.0370	0.026	0.863
Optimized LightGBM with correlated indices	**0.033**	**0.020**	**0.989**
LSTM	0.038	0.028	0.662
GRU	0.039	0.029	0.642
CNN	0.0375	0.027	0.692

Figure 27: Global Stock Price Forecasting during a Period of Market Stress using LightGBM. Original from Guennioui et al[80]

Table IV.
NWRMSLE values of model results

Model	Corporacion Favorita Grocery	Jollychic	Rossmann
NWRMSLE			
Feature engineering + Ridge regress model	0.5553	1.7243	0.3580
Feature engineering + Svm regress model	1.7421	2.2138	0.5559
Feature engineering + Random forest model	0.5657	1.56365	0.1493
Raw data + LightGBM single model	0.52608	1.05542	–
Feature engineering + Xgboost	0.51927	0.81317	0.12168
Feature engineering + LightGBM single model	0.50624	0.77333	0.11589
Feature engineering + LSTM single model	0.50669	0.80186	0.11108
Feature engineering + Combined model	*0.50413*	*0.76501*	*0.10955*

Table V.
Prediction time of models

Model	Prediction time		
	Corporacion Favorita Grocery	Jollychic	Rossmann
LightGBM model	1.76 s	1.9 s	1.31 s
LSTM model	54 s	38 s	20.7 s
Combined model	55.75 s	39.92 s	21.7 s

Figure 28: Supply chain sales forecasting based on LightGBM and LSTM combination model. Original from Weng et al[81].

TABLE I: Performance evaluation of local and global models.

Modeling framework	Single	sTS	mTS	lTS	All
Local modeling	0.7983	0.7870	0.7553	0.7182	0.7647
Global modeling	0.8011	0.7546	0.7115	0.6784	0.7364
Improvement [in %]	-0.28	3.24	4.38	3.98	2.83

COMPARISON WITH STATE-OF-THE-ART FORECASTING MODELS (MASE). THE MASE (↓) OF NAIVE PREDICTIONS IS 1. THE BEST RESULT FOR EACH LOAD AGGREGATE IS MARKED IN BOLD, THE SECOND-BEST IS UNDERLINED

Modeling Approach	**Model**	**Single**	**sTS**	**mTS**	**lTS**	**All**
Local	Local-MLR [33]	0.9333	0.8933	0.9045	0.9069	0.9095
Local	Local-LSTM [2]	0.8334	0.8400	0.8133	0.8024	0.8223
Local	Local-N-BEATS†	0.7983	0.7870	0.7553	0.7182	0.7647
Global	N-BEATS [34]	0.8306	0.7775	0.7253	0.6920	0.7564
Global‡	K-means-LSTM [21]	0.7705	0.7858	0.7234	0.6828	0.7406
Global	DeepAR (w/o localization) [12]	0.7725	0.7891	0.7241	0.6832	0.7422
Global	DeepAR (w/ localization)	**0.7642**	0.7749	0.7174	0.6763	0.7332
Global	Ours (w/o localization)	0.8048	0.7574	0.7146	0.6829	0.7399
Global	Ours (w/ localization)	0.7698	**0.7444**	**0.7047**	**0.6694**	**0.7221**

† This model represents the local N-BEATS version from Table I.
‡ This pooling-based approach is formally a global procedure applied over a subset of time series data at the time.

Figure 29: A Global Modeling Framework for Load Forecasting in Distribution Networks. Original from Grabner et al[72].

Table 2
Accuracy metrics relative to the strongest previously published method (baseline).

	0.5-risk				0.9-risk				Average
	parts								
(L, S)	(0, 1)	(2, 1)	(0, 8)	all(8)	(0, 1)	(2, 1)	(0, 8)	all(8)	average
Snyder (baseline)	1.00	1.00	1.00	**1.00**	1.00	1.00	1.00	1.00	1.00
Croston	1.47	1.70	2.86	1.83	–	–	–	–	1.97
ISSM	1.04	1.07	1.24	1.06	1.01	1.03	1.14	1.06	1.08
ETS	1.28	1.33	1.42	1.38	1.01	1.03	1.35	1.04	1.23
rnn-gaussian	1.17	1.49	1.15	1.56	1.02	0.98	1.12	1.04	1.19
rnn-negbin	**0.95**	**0.91**	0.95	**1.00**	1.10	**0.95**	1.06	0.99	0.99
DeepAR	0.98	**0.91**	**0.91**	1.01	**0.90**	**0.95**	**0.96**	**0.94**	**0.94**
	ec-sub								
(L, S)	(0, 2)	(0, 8)	(3, 12)	all(33)	(0, 2)	(0, 8)	(3, 12)	all(33)	average
Snyder	1.04	1.18	1.18	1.07	1.0	1.25	1.37	1.17	1.16
Croston	1.29	1.36	1.26	0.88	–	–	–	–	1.20
ISSM (baseline)	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	1.00	1.00
ETS	0.83	1.06	1.15	0.84	1.09	1.38	1.45	0.74	1.07
rnn-gaussian	1.03	1.19	1.24	0.85	0.91	1.74	2.09	0.67	1.21
rnn-negbin	0.90	0.98	1.11	0.85	1.23	1.67	1.83	0.78	1.17
DeepAR	**0.64**	**0.74**	**0.93**	**0.73**	**0.71**	**0.81**	1.03	**0.57**	**0.77**
	ec								
(L, S)	(0, 2)	(0, 8)	(3, 12)	all(33)	(0, 2)	(0, 8)	(3, 12)	all(33)	average
Snyder	0.87	1.06	1.16	1.12	0.94	1.09	1.13	1.01	1.05
Croston	1.30	1.38	1.28	1.39	–	–	–	–	1.34
ISSM (baseline)	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	1.00	1.00
ETS	0.77	0.97	1.07	1.23	1.05	1.33	1.37	1.11	1.11
rnn-gaussian	0.89	0.91	0.94	1.14	0.90	1.15	1.23	**0.90**	1.01
rnn-negbin	0.66	0.71	**0.86**	**0.92**	0.85	1.12	1.33	0.98	0.93
DeepAR	**0.59**	**0.68**	0.99	0.98	**0.76**	**0.88**	**1.00**	0.91	**0.85**

Note: The best results are marked in bold (lower is better).

Figure 30: DeepAR: Probabilistic forecasting with autoregressive recurrent networks. Original from Salinas et al[82].

Table 3. Performance of global and local models evaluated with respect to MASE and RMSSE. Model complexity estimated by TNP (total number of parameters), TCMS (total compressed model size), and WNP (weighted average number of parameters) and WCMS (weighted average compressed model size), per model.

Forecasting methods	No. of pools	MASE		RMSSE		TNP	WNP	TCMS (Bytes)	WCMS (Bytes)
Partitioning approaches									
DeepAR-Total	1	0.572	—	0.78245	—	204,603	204,603	776,553	776,553
DeepAR-State	3	0.564	-1.38%	0.78060	-0.24%	580,729	203,571	2,178,371	763,894
DeepAR-Store	10	0.560	-1.98%	0.78241	-0.01%	2,121,070	212,107	7,921,985	792,199
DeepAR-Category	3	0.566	-1.08%	0.78094	-0.19%	678,089	296,591	2,595,707	1,136,317
DeepAR-Department	7	0.559	-2.15%	0.78138	-0.14%	1,294,621	226,679	4,821,767	843,542
DeepAR-State-Category	9	0.553	-3.23%	0.78080	-0.21%	1,340,627	168,267	5,015,850	629,156
DeepAR-State-Department	21	**0.551**	**-3.58%**	0.78064	-0.23%	2,419,623	131,080	9,042,778	490,142
DeepAR-Store-Category	30	0.556	-2.74%	0.78340	0.12%	3,708,210	134,902	13,867,558	504,447
DeepAR-Store-Department	70	0.554	-3.11%	0.78421	0.23%	10,310,650	155,903	38,339,800	579,556
DeepAR-Comb	154	0.558	-2.37%	**0.77620**	**-0.80%**	22,658,222	192,634	83,740,297	723,979
Local benchmarks									
ARIMA	1	0.798	39.51%	0.93436	19.42%	487,840*	16*	24,431,329	801
ETS	1	0.808	41.24%	0.92853	18.67%	426,860*	14*	24,411,454	801
Seasonal Naïve	1	0.905	58.35%	1.23763	58.17%	—	—	—	—

*Conservative estimate of the number of parameters in the models. In ARIMA they include the orders $0 \leq p \leq 5$, $0 \leq q \leq 5$, $0 \leq P \leq 2$ and $0 \leq Q \leq 2$, c if exists and the residuals variance, thus a maximum of 16 parameters. In ETS models they include the smoothing parameters α, β, γ and ϕ, the initial states l_0, b_0, $s_0, \ldots, s_6$ and the residuals variance, thus a maximum of 14 parameters.

Figure 31: Investigating the Accuracy of Autoregressive Recurrent Networks Using Hierarchical Aggregation Structure-Based Data Partitioning. Original from Oliveira et al[23].

Table 4. Point forecasting results for three different forecast horizons and three selected cities. Performance values are reported in terms of "average ± stdev" values (bold-faced colored entries exhibit the results for the best performing model).

(a) 6-hours prediction horizon

	ANTALYA			ISTANBUL			KARS		
Model	NRMSE	ND	SMAPE	NRMSE	ND	SMAPE	NRMSE	ND	SMAPE
NB	0.097 ± 0.12	0.093 ± 0.12	8.660 ± 9.87	0.079 ± 0.09	0.074 ± 0.08	7.249 ± 7.65	0.083 ± 0.09	0.078 ± 0.08	7.646 ± 8.13
SARIMAX	0.262 ± 0.13	0.249 ± 0.13	26.096 ± 14.34	0.068 ± 0.03	0.061 ± 0.03	5.993 ± 3.00	0.179 ± 0.12	0.167 ± 0.12	16.436 ± 10.02
GBR	0.072 ± 0.07	0.068 ± 0.07	6.678 ± 6.07	0.066 ± 0.07	0.063 ± 0.07	6.270 ± 6.22	0.061 ± 0.06	0.058 ± 0.065	5.759 ± 5.94
RF	0.072 ± 0.07	0.068 ± 0.07	6.670 ± 6.04	0.066 ± 0.07	0.063 ± 0.07	6.237 ± 6.22	0.061 ± 0.06	0.058 ± 0.06	5.769 ± 5.93
GRU	**0.036 ± 0.04**	**0.031 ± 0.03**	**3.000 ± 3.11**	**0.025 ± 0.01**	**0.020 ± 0.01**	**2.032 ± 0.97**	**0.036 ± 0.04**	**0.031 ± 0.04**	**3.148 ± 3.83**
LSTM	0.038 ± 0.04	0.033 ± 0.04	3.175 ± 3.47	0.039 ± 0.03	0.033 ± 0.03	3.318 ± 2.77	0.037 ± 0.04	0.031 ± 0.04	3.143 ± 3.83
DeepState	0.048 ± 0.03	0.042 ± 0.03	4.234 ± 2.73	0.049 ± 0.03	0.043 ± 0.03	4.220 ± 2.75	0.041 ± 0.01	0.037 ± 0.04	3.679 ± 3.68
DeepAR	0.043 ± 0.03	0.038 ± 0.03	4.008 ± 2.99	0.049 ± 0.03	0.043 ± 0.03	3.914 ± 1.90	0.047 ± 0.03	0.041 ± 0.03	4.308 ± 2.28

(b) 12-hours prediction horizon

	ANTALYA			ISTANBUL			KARS		
Model	NRMSE	ND	SMAPE	NRMSE	ND	SMAPE	NRMSE	ND	SMAPE
NB	0.102 ± 0.11	0.094 ± 0.11	8.892 ± 8.83	0.082 ± 0.07	0.074 ± 0.07	7.420 ± 6.95	0.086 ± 0.08	0.077 ± 0.08	7.798 ± 7.67
SARIMAX	0.289 ± 0.14	0.267 ± 0.14	27.811 ± 13.58	0.183 ± 0.08	0.159 ± 0.08	15.908 ± 7.39	0.190 ± 0.08	0.170 ± 0.08	17.084 ± 6.50
GBR	0.078 ± 0.06	0.072 ± 0.06	7.193 ± 5.69	0.069 ± 0.07	0.064 ± 0.07	6.375 ± 5.97	0.064 ± 0.06	0.060 ± 0.06	5.887 ± 5.75
RF	0.078 ± 0.06	0.072 ± 0.06	7.184 ± 5.67	0.069 ± 0.07	0.064 ± 0.07	6.343 ± 5.97	0.064 ± 0.06	0.060 ± 0.06	5.901 ± 5.74
GRU	**0.041 ± 0.02**	**0.034 ± 0.02**	**3.463 ± 2.05**	**0.036 ± 0.02**	**0.029 ± 0.01**	**2.893 ± 1.32**	**0.043 ± 0.04**	**0.037 ± 0.04**	**3.833 ± 4.15**
LSTM	0.045 ± 0.03	0.037 ± 0.02	3.782 ± 2.38	0.038 ± 0.02	0.031 ± 0.01	3.062 ± 1.39	0.046 ± 0.04	0.040 ± 0.03	4.046 ± 3.80
DeepState	0.054 ± 0.03	0.046 ± 0.03	4.568 ± 3.06	0.055 ± 0.02	0.048 ± 0.02	4.759 ± 2.13	0.054 ± 0.03	0.048 ± 0.03	4.784 ± 0.03
DeepAR	0.050 ± 0.04	0.042 ± 0.03	4.098 ± 2.79	0.048 ± 0.03	0.042 ± 0.03	4.215 ± 2.62	0.050 ± 0.04	0.044 ±0.03	4.265 ± 2.61

(c) 24-hours prediction horizon

	ANTALYA			ISTANBUL			KARS		
Model	NRMSE	ND	SMAPE	NRMSE	ND	SMAPE	NRMSE	ND	SMAPE
NB	0.098 ± 0.09	0.087 ± 0.09	8.387 ± 7.32	0.086 ± 0.08	0.075 ± 0.08	7.409 ± 6.75	0.090 ± 0.08	0.079 ± 0.08	7.932 ± 7.71
SARIMAX	0.283 ± 0.11	0.248 ± 0.11	26.555 ± 13.38	0.185 ± 0.06	0.158 ± 0.06	15.810 ± 4.55	0.180 ± 0.06	0.155 ± 0.06	15.712 ± 4.84
GBR	0.076 ± 0.06	0.069 ± 0.06	6.882 ± 5.50	0.074 ± 0.06	0.067 ± 0.06	6.647 ± 5.79	0.070 ± 0.07	0.065 ± 0.7	6.430 ± 6.01
RF	0.076 ± 0.06	0.068 ± 0.06	6.874 ± 5.48	0.074 ± 0.06	0.067 ± 0.06	6.634 ± 5.79	0.070 ± 0.06	0.065 ± 0.07	6.435 ± 6.00
GRU	0.047 ± 0.09	0.039 ± 0.02	4.007 ± 1.71	**0.043 ± 0.02**	**0.035 ± 0.02**	**3.517 ± 1.71**	**0.047 ± 0.03**	**0.038 ± 0.03**	**3.971 ± 3.23**
LSTM	**0.047 ± 0.02**	**0.039 ± 0.02**	**3.978 ± 1.92**	0.044 ± 0.02	0.036 ± 0.02	3.600 ± 1.60	0.047 ± 0.03	0.039 ± 0.03	3.986 ± 3.47
DeepState	0.070 ± 0.04	0.059 ± 0.04	5.872 ± 3.88	0.057 ± 0.02	0.046 ± 0.02	4.639 ± 1.71	0.057 ± 0.03	0.047 ± 0.02	4.825 ± 2.30
DeepAR	0.048 ± 0.02	0.039 ± 0.02	3.862 ± 1.60	0.051 ± 0.02	0.042 ± 0.02	4.239 ± 1.86	0.052 ± 0.02	0.043 ± 0.02	4.325 ± 1.74

Table 5. Probabilistic forecasting average performance values over three target cities obtained for different prediction horizons

Model	0.10	0.25	0.50	0.75	0.90	average
	6-hours prediction horizon					
SARIMAX	0.0586	0.0807	0.1060	0.1250	0.1335	0.1007
NB	0.0374	0.0452	0.0472	0.0370	0.0248	0.0383
RF	0.0209	0.0304	0.0372	0.0352	0.0268	0.0301
GBR	0.0209	0.0304	0.0373	0.0352	0.0268	0.0301
GRU	**0.0122**	**0.0210**	**0.0265**	**0.0218**	**0.0135**	**0.0190**
LSTM	0.0139	0.0229	0.0292	0.0241	0.0147	0.0209
DeepState	0.0389	0.0539	0.0584	0.0468	0.0309	0.0457
DeepAR	0.0256	0.0380	0.0430	0.0336	0.0196	0.0319
	12-hours prediction horizon					
SARIMAX	0.0530	0.0775	0.1080	0.1140	0.1154	0.0935
NB	0.0313	0.0493	0.0606	0.0482	0.0290	0.0436
RF	0.0214	0.0387	0.0550	0.0440	0.0296	0.0379
GBR	0.0215	0.0386	0.0551	0.0448	0.0296	0.0379
GRU	**0.0178**	**0.0340**	**0.0441**	**0.0348**	**0.0188**	**0.0299**
LSTM	0.0183	**0.0340**	0.0444	0.0356	0.0198	0.0304
DeepState	0.0193	0.0366	0.0472	0.0397	0.0233	0.0332
DeepAR	0.0198	0.0350	0.0442	0.0375	0.0238	0.0320
	24-hours prediction horizon					
SARIMAX	0.0440	0.0742	0.1053	0.1138	0.1075	0.0889
NB	0.0317	0.0555	0.0681	0.0490	0.0280	0.0464
RF	0.0271	0.0528	0.0638	0.0472	0.030	0.0441
GBR	0.0271	0.0528	0.0638	0.0472	0.0300	0.0440
GRU	0.0227	0.0471	0.0557	**0.0383**	**0.0199**	**0.0367**
LSTM	0.0228	0.0471	0.0555	**0.0383**	**0.0204**	0.0368
DeepState	0.0261	**0.0430**	**0.0506**	0.0423	0.0245	0.0373
DeepAR	**0.0226**	0.0437	0.0620	0.0607	0.0448	0.0467

Figure 32: An empirical study on probabilistic forecasting for predicting city-wide electricity consumption. Original from Rafayal et al[83].

Table 1. Results for the 79 monthly series of NASS.

Method	Mean sMAPE	Median sMAPE	Mean MASE	Median MASE
DeepCPNet				
- All LGAs	0.1527	0.1425	1.0628	0.9818
- Treated Group	0.1547	0.1477	1.0731	0.9790
- Control Group	0.1388	0.1375	0.9917	1.0117
DeepCPNet-ALI				
- All LGAs	0.1396	0.1341	0.9786	0.9467
- Treated Group	0.1405	0.1341	0.9812	0.9413
- Control Group	0.1332	0.1357	0.9604	0.9733

LSTM-univariate				
- All LGAs	0.1557	0.1537	1.0828	1.0205
- Treated Group	0.1570	0.1560	1.0900	1.0205
- Control Group	0.1464	0.1323	1.0331	0.9716
ARIMA				
- All LGAs	0.1560	0.1458	1.0888	0.9853
- Treated Group	0.1572	0.1427	1.0973	0.9643
- Control Group	0.1476	0.1490	1.0300	1.0069
ETS				
- All LGAs	0.1507	0.1441	1.0647	1.0078
- Treated Group	0.1515	0.1427	1.0705	0.9981
- Control Group	0.1450	0.1555	1.0244	1.0876

Table 2. Results for the 62 monthly series of 911 emergency calls.

	Methods							
	Control group				Treated group			
Error metric	DeepCPNet	LSTM-uni	ARIMA	ETS	DeepCPNet	LSTM-uni	ARIMA	ETS
Mean sMAPE	0.1899	0.1945	0.1847	0.1920	0.2525	0.2525	0.2508	0.2624
Median sMAPE	0.1740	0.1783	0.1717	0.1746	0.2290	0.2256	0.2250	0.2336
Mean MASE	0.8521	0.8698	0.8288	0.8616	1.3939	1.3926	1.3857	1.4498
Median MASE	0.9176	0.9370	0.9272	0.9430	1.3100	1.2842	1.2848	1.3334

Figure 33: Causal Inference Using Global Forecasting Models for Counterfactual Prediction. Original from Grecov et al[84].

Table 3. Recorded msMAPE, NRMSE, and ND metrics on the test set. The best metric is highlighted in boldface, while the next-best metric is underlined. The training time in *seconds* is shown for each method in the last column. Lower is better.

Model	msMAPE	NRMSE	ND	Training Time
Seasonal Naïve	13.5092	5.7848	0.1480	-
ARIMA	17.5130	4.8592	0.1450	280 s
DT	18.3116	7.6188	0.2235	21 s
RF	14.7366	5.7692	0.1598	182 s
LightGBM	15.1735	5.7227	0.1562	320 s
Transformer	**2.7639**	**0.7325**	**0.0280**	1529 s

Figure 34: A Global Forecasting Approach to Large-Scale Crop Production Prediction with Time Series Transformers. Original from Ibañez and Monterola[85].

TABLE I
MODEL PERFORMANCE COMPARISON OF SPECIALIST LSTM MODEL AND GENERALIST TRANSFORMER MODEL

Building	RMSE		Building	RMSE	
	Specialist	Generalist		Specialist	Generalist
Wildlife Reserve	1.64	**1.57**	Thomas Cherry Building	9.40	**7.87**
Menzies College Annexe	4.60	**3.80**	Indoor Sports Centre	7.71	8.33
CS Buildings CS1 CS2 CS3	4.53	**3.99**	Humanities 2 Building	7.25	8.41
Martin Building	3.69	4.34	LIMS2 Building	9.17	**8.56**
Physical Sciences 1 Building	4.10	4.44	BS2 Building	9.72	**8.72**
Albury-Wodonga - B8	4.48	5.00	The Learning Commons Building	8.29	9.05
Animal and Glass Houses	5.75	**5.37**	Beth Gleeson Building	11.28	**9.15**
Social Sciences Building	6.07	**5.40**	Menzies College Main Supply	10.11	**9.19**
Eastern Lecture Theatre	6.25	**5.48**	RLR Building	10.96	**9.23**
John Scott Meeting House	5.24	5.61	Agora East Building	10.57	**9.53**
Union	5.64	**5.62**	ED1 Building HEUD	10.55	**9.64**
BS1 Building	4.58	5.84	George Singer Building	10.23	**9.91**
Boilerhouse Building	7.13	**5.85**	Glenn College	11.91	**10.69**
Peribolos West Building	6.80	**6.07**	Sylvia Walton Building	13.10	**11.13**
Humanities 3 Building	6.73	**6.13**	ED2 Building	14.45	**12.65**
Peribolos East Building	6.77	**6.69**	Donald Whitehead Building	13.92	**13.73**
Agora Theatre	8.49	**7.22**	Chisholm College	18.16	**15.96**
SFP	7.60	**7.26**	LIMS 1	22.93	**20.54**
Health Sciences Clinic	6.89	7.29	David Myers Building	52.98	**44.53**
SPF Specialised Pathogen Facility	9.61	**7.73**	Library	67.17	**53.63**

Figure 35: Specialist vs Generalist: A Transformer Architecture for Global Forecasting Energy Time

Series. Original from Rathnayaka et al[86].

Table 2: MAE results on the two datasets, with 24, 96 and 720 hours forecast horizon. MV = multivariate, L = local, G = global. The best results are highlighted in bold and the best results per training strategy are highlighted in italic.

Model	strat-egy	input (days)	Electricity 24h	Electricity 96h	Electricity 720h	Ausgrid 24h	Ausgrid 96h	Ausgrid 720h
Informer [19]	MV	4	0.399	0.407	0.450	0.582	0.607	0.645
Autoformer [20]	MV	4	0.289	0.317	0.361	0.579	0.569	*0.592*
FEDformer [21]	MV	4	*0.284*	*0.297*	*0.343*	*0.560*	*0.566*	0.609
LSTM	MV	7	0.400	0.402	0.407	0.611	0.618	0.613
Transformer	MV	7	0.366	0.384	0.382	0.584	0.586	0.576
Persistence	L	-	0.279	0.279	0.447	0.647	0.647	0.717
Linear regression	L	14	0.203	*0.233*	*0.296*	*0.496*	*0.524*	*0.565*
MLP	L	7	*0.199*	0.236	0.308	0.499	0.532	0.567
LSTM	L	7	0.263	0.283	0.337	0.517	0.541	0.573
Transformer	L	7	0.256	0.289	0.354	0.535	0.563	0.583
LTSF-Linear [23]	G	14	0.209	0.237	0.301	0.490	0.515	0.553
PatchTST [22]	G	14	0.190	**0.222**	**0.290**	**0.468**	**0.494**	**0.522**
LSTM	G	7	0.207	0.239	0.302	0.491	0.525	0.559
Transformer	G	14	**0.184**	0.225	0.312	0.482	0.514	0.533

Table 3: Training times in hours, measured on a machine with a Nvidia 3090 RTX GPU.

Model	Electricity 24h	Electricity 96h	Electricity 720h	Ausgrid 24h	Ausgrid 96h	Ausgrid 720h
Linear regression (local)	0.02	0.03	0.08	0.02	0.03	0.07
MLP (local)	0.42	0.42	0.36	0.40	0.39	0.39
LSTM (multivariate)	0.06	0.08	0.30	0.03	0.04	0.10
LSTM (local)	8.25	7.61	7.20	4.27	3.55	3.49
LSTM (global)	1.09	0.82	0.98	1.11	0.84	0.71
Transformer (multivariate)	0.19	0.23	0.88	0.10	0.09	0.39
Transformer (local)	14.20	16.82	102.09	8.33	9.74	62.53
Transformer (global)	3.42	2.00	9.86	3.85	2.85	6.77

Figure 36: Transformer Training Strategies for Forecasting Multiple Load Time Series. Original from Hertel et al[87].

Forecaster	Demand Error		Demand Bias		RMSE		MAPE	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Naive Forecaster	0.603	0.135	0.079	0.161	11.754	4.038	0.088	0.011
lightgbm	0.489	0.066	0.296	0.127	9.737	2.941	0.142	0.005
DeepAR	0.593	0.126	0.090	0.156	12.443	4.224	0.092	0.011
Transformer	**0.449**	0.04	**- 0.047**	0.064	**9.588**	2.725	**0.081**	0.007

Table 2: Model Performance

Comparison of our transformer models with commonly used alternative approaches and a naive baseline. The metrics are calculated over 10 different weeks in 2022 and measure the performance of the predictions for the first week. We report here the mean and standard deviation over the 10 weeks.

Figure 37: Deep Learning based Forecasting: a case study from the online fashion industry. Original from Kunz et al[61].

Table 3. Performance of the TFT compared to a naive baseline on our real-world agricultural dataset. The TFT outperforms the baseline on all metrics for all datasets except for the *0.9-Risk*.

Data	**Naive baseline**				**Temporal Fusion Transformer**			
	MAE	*RMSE*	*0.5-Risk*	*0.9-Risk*	*MAE*	*RMSE*	*0.5-Risk*	*0.9-Risk*
Train	7.3813	14.0810	0.1636	0.1444	**5.3457**	**11.6358**	**0.1110**	**0.1429**
Val	9.4538	16.6170	0.1536	**0.1227**	**8.4557**	**15.8144**	**0.1295**	0.1466
Test	5.3452	10.5391	0.1029	**0.0899**	**4.6931**	**9.6979**	**0.0858**	0.1131

Figure 38: Forecasting Sensor-data in Smart Agriculture with Temporal Fusion Transformers. Original from Deforce et al[88].

(a) Training counties

Model	MAE	SMAPE	CV
TFT	**16.00**	**41.20**	**75.88**
TFT - no viral load	17.92	44.88	90.62
DeepTCN	25.95	59.30	100.80
DeepTCN - no viral load	29.94	61.32	119.79

(b) Holdout counties

Model	MAE	SMAPE	CV
TFT	10.73	**32.92**	56.10
TFT - no viral load	**9.92**	36.50	**42.88**
DeepTCN	20.98	51.17	83.35
DeepTCN - no viral load	21.87	46.24	110.06

TABLE 1: Performance comparison for the four models of TFT, TFT without viral load data, DeepTCN, and DeepTCN without viral load data.

Figure 39: Leveraging Wastewater Monitoring for COVID-19 Forecasting in the US: a Deep Learning study. Original from Fazli and Shakeri[89].

Models		TiDE		PatchTST/64		N-HiTS		DLinear		FEDformer		Autoformer		Informer		Pyraformer		LogTrans	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	0.166	0.222	**0.149**	**0.198**	0.158	**0.195**	0.176	0.237	0.238	0.314	0.249	0.329	0.354	0.405	0.896	0.556	0.458	0.490
	192	0.209	0.263	**0.194**	**0.241**	0.211	0.247	0.220	0.282	0.275	0.329	0.325	0.370	0.419	0.434	0.622	0.624	0.658	0.589
	336	0.254	0.301	**0.245**	**0.282**	0.274	0.300	0.265	0.319	0.339	0.377	0.351	0.391	0.583	0.543	0.739	0.753	0.797	0.652
	720	**0.313**	0.340	**0.314**	**0.334**	0.401	0.413	0.323	0.362	0.389	0.409	0.415	0.426	0.916	0.705	1.004	0.934	0.869	0.675
Traffic	96	**0.336**	0.253	0.360	**0.249**	0.402	0.282	0.410	0.282	0.576	0.359	0.597	0.371	0.733	0.410	2.085	0.468	0.684	0.384
	192	**0.346**	**0.257**	0.379	**0.256**	0.420	0.297	0.423	0.287	0.610	0.380	0.607	0.382	0.777	0.435	0.867	0.467	0.685	0.390
	336	**0.355**	**0.260**	0.392	0.264	0.448	0.313	0.436	0.296	0.608	0.375	0.623	0.387	0.776	0.434	0.869	0.469	0.734	0.408
	720	**0.386**	**0.273**	0.432	0.286	0.539	0.353	0.466	0.315	0.621	0.375	0.639	0.395	0.827	0.466	0.881	0.473	0.717	0.396
Electricity	96	**0.132**	0.229	**0.129**	**0.222**	0.147	0.249	0.140	0.237	0.186	0.302	0.196	0.313	0.304	0.393	0.386	0.449	0.258	0.357
	192	**0.147**	0.243	**0.147**	**0.240**	0.167	0.269	0.153	0.249	0.197	0.311	0.211	0.324	0.327	0.417	0.386	0.443	0.266	0.368
	336	**0.161**	0.261	0.163	**0.259**	0.186	0.290	0.169	0.267	0.213	0.328	0.214	0.327	0.333	0.422	0.378	0.443	0.280	0.380
	720	**0.196**	0.294	**0.197**	**0.290**	0.243	0.340	0.203	0.301	0.233	0.344	0.236	0.342	0.351	0.427	0.376	0.445	0.283	0.376
ETTh1	96	**0.375**	0.398	0.379	0.401	0.378	**0.393**	0.375	**0.399**	0.376	0.415	0.435	0.446	0.941	0.769	0.664	0.612	0.878	0.740
	192	**0.412**	0.422	0.413	0.429	0.427	0.436	**0.412**	**0.420**	0.423	0.446	0.456	0.457	1.007	0.786	0.790	0.681	1.037	0.824
	336	**0.435**	**0.433**	**0.435**	0.436	0.458	0.484	0.439	0.443	0.444	0.462	0.486	0.487	1.038	0.784	0.891	0.738	1.238	0.932
	720	0.454	**0.465**	**0.446**	**0.464**	0.472	0.561	0.501	0.490	0.469	0.492	0.515	0.517	1.144	0.857	0.963	0.782	1.135	0.852
ETTh2	96	**0.270**	**0.336**	0.274	**0.337**	0.274	0.345	0.289	0.353	0.332	0.374	0.332	0.368	1.549	0.952	0.645	0.597	2.116	1.197
	192	**0.332**	0.380	0.338	**0.376**	0.353	0.401	0.383	0.418	0.407	0.446	0.426	0.434	3.792	1.542	0.788	0.683	4.315	1.635
	336	**0.360**	0.407	0.363	**0.397**	0.382	0.425	0.448	0.465	0.400	0.447	0.477	0.479	4.215	1.642	0.907	0.747	1.124	1.604
	720	0.419	0.451	**0.393**	**0.430**	0.625	0.557	0.605	0.551	0.412	0.469	0.453	0.490	3.656	1.619	0.963	0.783	3.188	1.540
ETTm1	96	0.306	0.349	**0.293**	**0.346**	0.302	0.350	0.299	0.343	0.326	0.390	0.510	0.492	0.626	0.560	0.543	0.510	0.600	0.546
	192	**0.335**	**0.366**	**0.333**	0.370	0.347	0.383	0.335	**0.365**	0.365	0.415	0.514	0.495	0.725	0.619	0.557	0.537	0.837	0.700
	336	**0.364**	**0.384**	0.369	0.392	0.369	0.402	0.369	0.386	0.392	0.425	0.510	0.492	1.005	0.741	0.754	0.655	1.124	0.832
	720	**0.413**	**0.413**	0.416	0.420	0.431	0.441	0.425	0.421	0.446	0.458	0.527	0.493	1.133	0.845	0.908	0.724	1.153	0.820
ETTm2	96	**0.161**	**0.251**	0.166	0.256	0.176	0.255	0.167	0.260	0.180	0.271	0.205	0.293	0.355	0.462	0.435	0.507	0.768	0.642
	192	**0.215**	**0.289**	0.223	0.296	0.245	0.305	0.224	0.303	0.252	0.318	0.278	0.336	0.595	0.586	0.730	0.673	0.989	0.757
	336	**0.267**	**0.326**	0.274	0.329	0.295	0.346	0.281	0.342	0.324	0.364	0.343	0.379	1.270	0.871	1.201	0.845	1.334	0.872
	720	**0.352**	**0.383**	0.362	**0.385**	0.401	0.413	0.397	0.421	0.410	0.420	0.414	0.419	3.001	1.267	3.625	1.451	3.048	1.328

Table 2: Multivariate long-term forecasting results with our model. $T \in \{96, 192, 336, 720\}$ for all datasets. The best results including the ones that cannot be statistically distinguished from the best mean numbers are in **bold**. We calculate standard error intervals for our method over 5 runs. The rest of the numbers are taken from the results from (Nie et al., 2022)[1]. All metrics are reported on standard normalized datasets. We provide the standard errors for our method in Table 5 in Appendix B.

Model	Covariates	Test WRMSSE
TiDE	Static + Dynamic	$\mathbf{0.611 \pm 0.009}$
TiDE	Date only	0.637 ± 0.005
DeepAR	Static + Dynamic	0.789 ± 0.025
PatchTST	None	0.976 ± 0.014

Table 3: M5 forecasting results on the private test set. We report the competition metric (averaged across three runs) for each model. We also list the covariates used by all models.

Figure 40: Long-term Forecasting with TiDE: Time-series Dense Encoder. Original from Das et al[69].

Table 8
Forecasting performance of nine meta-learners and three ensemble benchmarks in the test data.

Meta-learner	Horizon								
	h=1		h=4		h=7		h=1-7		
	sMAPE	AvgRelMAE	sMAPE	AvgRelMAE	sMAPE	AvgRelMAE	sMAPE	AvgRelMAE	MPE
M0	**15.953**	**0.987**	**16.747**	**0.980**	**17.965**	**0.960**	**16.849**	**0.968**	**-0.170**
M1	15.980	0.989	16.765	0.981	17.972	0.964	16.865	0.970	-0.046
M2	15.980	0.989	16.771	0.983	17.981	0.963	16.870	0.970	-0.034
M3	16.183	0.997	17.114	1.001	18.514	0.993	17.231	0.995	-1.117
M4	16.140	0.999	16.950	0.992	18.125	0.971	17.053	0.982	0.394
M5	17.067	1.058	18.073	1.069	19.203	1.035	18.135	1.050	3.886
M6	16.128	1.002	16.939	0.994	18.050	0.965	17.006	0.979	-0.077
E1	16.171	1.004	16.985	0.997	18.161	0.977	17.078	0.986	-0.064
E2	16.103	1.001	16.900	0.990	18.079	0.968	16.994	0.978	-0.292
E3	16.237	1.001	17.169	1.006	18.501	0.992	17.247	0.996	-1.334
FFORMA1	16.060	0.992	16.868	0.985	18.110	0.969	16.975	0.977	-0.060
FFORMA2	16.023	0.992	16.824	0.983	18.049	0.965	16.928	0.974	0.254

The best performing methods are shown in bold: The GBRT-7 forecasts are used as the baseline for calculating AvgRelMAE

Table 9
Forecasting performance of eight meta-learners and three ensemble benchmarks in promotion and non-promotion periods.

	Promotion		Non-promotion	
	AvgRelMAE (to ETS)	AvgRelMAE (to GBRT -7)	AvgRelMAE (to ETS)	AvgRelMAE (to GBRT -7)
M0	**0.760**	**0.950**	**0.830**	**0.984**
M1	0.762	0.952	0.831	0.985
M2	0.762	0.951	0.832	0.986
M3	0.794	0.992	0.842	0.998
M4	0.770	0.962	0.843	0.999
M5	0.836	1.044	0.890	1.055
M6	0.772	0.964	0.838	0.993
E1	0.780	0.974	0.845	1.001
E2	0.770	0.962	0.838	0.993
E3	0.795	0.994	0.842	0.999
FFORMA1	0.771	0.963	0.834	0.989
FFORMA2	0.770	0.962	0.833	0.987

The best performing method is shown in bold.

Figure 41: Retail sales forecasting with meta-learning. Original from Ma and Fildes[24].

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

Figure 42: Robust Sales Forecasting Using Deep Learning with Static and Dynamic Covariates. Original from Ramos and Oliveira[90].

Table 4. Results for category level dataset

		mMAPE (All) k = 1724		mMAPE (G1) k = 549		mMAPE (G2) k = 544		mMAPE (G3) k = 631	
Model	Configuration	Mean	Median	Mean	Median	Mean	Median	Mean	Median
LSTM.ALL	LSTM-LS1/Bayesian/Adam	0.888	0.328	1.872	0.692	0.110	0.073	0.640	0.283
LSTM.ALL	LSTM-LS1/Bayesian/COCOB	0.803	0.267	1.762	0.791	0.070	0.002	0.537	0.259
LSTM.ALL	LSTM-LS2/Bayesian/Adam	0.847	0.327	1.819	0.738	0.103	0.047	0.582	0.326
LSTM.GROUP	LSTM-LS1/Bayesian/Adam	0.873	0.302	1.882	**0.667**	0.093	0.016	0.604	0.283
LSTM.GROUP	LSTM-LS1/Bayesian/COCOB	1.039	0.272	2.455	0.818	0.074	**0.000**	0.549	**0.250**
LSTM.GROUP	LSTM-LS2/Bayesian/Adam	0.812	0.317	1.818	0.738	0.091	0.022	0.587	0.314
LSTM.FEATURE	LSTM-LS1/Bayesian/Adam	1.065	0.372	2.274	0.889	0.135	0.100	0.738	0.388
LSTM.FEATURE	LSTM-LS1/Bayesian/COCOB	0.800	0.267	1.758	0.772	**0.069**	**0.000**	0.533	0.255
LSTM.FEATURE	LSTM-LS2/Bayesian/Adam	0.879	0.324	1.886	0.750	0.091	0.022	0.611	0.324
LSTM.CLUSTER	LSTM-LS1/Bayesian/Adam	0.954	0.313	2.109	0.869	0.135	0.110	0.625	0.322
LSTM.CLUSTER	LSTM-LS1/Bayesian/COCOB	**0.793**	0.308	**1.695**	0.748	0.077	0.005	0.562	0.302
LSTM.CLUSTER	LSTM-LS2/Bayesian/Adam	1.001	0.336	2.202	0.863	0.084	0.017	0.664	0.347
EWMA	_	0.968	0.342	1.983	1.026	0.107	0.021	0.762	0.412
ARIMA	_	1.153	0.677	2.322	0.898	0.103	0.056	0.730	0.496
ETS (non-seasonal)	_	0.965	0.362	2.020	0.803	0.113	0.060	0.713	0.444
ETS (seasonal)	_	0.983	0.363	2.070	0.804	0.116	0.059	0.713	0.445
Naïve	_	0.867	**0.250**	1.803	0.795	0.124	**0.000**	0.632	**0.250**
Naïve Seasonal	_	0.811	0.347	1.789	0.679	0.086	**0.000**	**0.523**	0.320
Prophet-Facebook		0.892	0.342	1.923	0.842	0.103	0.042	0.609	0.325

Table 5. Results for super-department level dataset

		mMAPE (All items) k = 18254		mMAPE (G1) k = 5682		mMAPE (G2) k = 5737		mMAPE (G3) k = 6835	
Model	Configuration	Mean	Median	Mean	Median	Mean	Median	Mean	Median
LSTM.ALL	LSTM-LS1/Bayesian/Adam	1.006	0.483	2.146	1.285	0.191	0.079	0.668	0.434
LSTM.ALL	LSTM-LS1/Bayesian/COCOB	0.944	0.442	2.041	1.203	0.163	0.053	0.614	0.394
LSTM.GROUP	LSTM-LS1/Bayesian/Adam	**0.871**	0.445	**1.818**	**1.009**	0.189	0.067	**0.603**	0.377
LSTM.GROUP	LSTM-LS1/Bayesian/COCOB	0.921	0.455	1.960	1.199	0.173	0.053	0.618	0.394
LSTM.FEATURE	LSTM-LS1/Bayesian/Adam	0.979	0.424	2.117	1.279	**0.151**	0.050	0.653	0.377
LSTM.FEATURE	LSTM-LS1/Bayesian/COCOB	1.000	0.443	2.143	1.282	0.215	0.092	0.676	0.398
EWMA	_	1.146	0.579	2.492	1.650	0.229	0.091	0.805	0.562
ARIMA	_	1.084	0.536	2.305	1.497	0.198	0.094	0.734	0.510
ETS (non-seasonal)	_	1.097	0.527	2.314	1.494	0.204	0.092	0.755	0.509
ETS (seasonal)	_	1.089	0.528	2.290	1.483	0.204	0.092	0.756	0.510
Naïve	_	0.981	**0.363**	2.008	1.122	0.204	**0.000**	0.713	**0.286**
Naïve Seasonal	_	1.122	0.522	2.323	1.513	0.219	0.050	0.803	0.475
Prophet-Facebook		1.087	0.554	2.266	1.400	0.210	0.113	0.765	0.534

Figure 43: Sales Demand Forecast in E-commerce using a Long Short-Term Memory Neural Network Methodology. Original from Bandara et al[91].

Table 3
Results for the 72 monthly series of CIF2016, ordered by the second column, which is the Mean sMAPE. For each column, the results of the best performing method(s) are marked in boldface.

Method	Mean sMAPE	Median sMAPE	Rank sMAPE	Mean MASE	Median MASE	Rank MASE
LSTM.Cluster.kMeans.Silhouette	**10.52**	7.30	12.34	0.88	0.59	12.17
LSTM.Cluster.Snob	10.53	7.34	13.21	0.89	0.59	13.17
LSTM.Cluster.Dbscan	10.57	7.30	13.13	0.91	0.59	13.08
LSTM.Cluster.kMeans.Elbow	10.60	7.20	12.09	0.89	0.56	12.03
LSTM.Horizon	10.61	7.92	13.19	0.90	0.60	12.60
LSTM.All	10.69	6.72	12.61	0.83	0.59	12.49
LSTMs and ETS	10.83	6.60	**10.88**	**0.79**	0.56	**10.82**
LSTM.Cluster.PAM.Silhouette	10.98	7.41	12.98	0.91	0.57	12.83
ETS	11.87	6.67	12.21	0.84	**0.53**	12.21
MLP	12.13	6.92	12.99	0.84	0.54	13.13
REST	12.45	7.57	14.61	0.90	0.59	14.90
ES	12.73	6.51	13.13	0.87	**0.53**	13.05
FRBE	12.90	6.77	12.78	0.89	0.57	12.78
HEM	13.04	7.32	14.44	0.90	0.59	14.49
Avg	13.05	8.02	16.49	0.99	0.67	16.48
BaggedETS	13.13	**5.98**	11.61	0.83	0.54	11.38
LSTM	13.33	8.20	17.07	0.95	0.68	17.05
Fuzzy c-regression	13.73	10.04	17.46	1.13	0.72	17.46
PB-GRNN	14.50	7.86	16.00	1.01	0.65	16.18
PB-RF	14.50	7.86	16.00	1.01	0.65	16.18
ARIMA	14.56	7.03	14.62	0.92	0.56	14.62
Theta	14.76	11.01	20.57	1.25	0.74	20.73
PB-MLP	14.94	8.05	16.57	0.99	0.68	16.58
TSFIS	15.11	10.18	19.74	1.27	0.91	19.71
Boot.EXPOS	15.25	6.92	14.60	0.93	0.61	14.51
MTSFA	16.51	9.69	18.24	1.13	0.71	18.19
FCDNN	16.62	8.71	20.01	1.14	0.82	20.26
Naïve Seasonal	19.05	12.72	23.12	1.29	0.95	23.46
MSAKAF	20.39	14.24	21.89	1.57	1.31	21.85
HFM	22.39	11.89	22.97	3.27	1.14	23.14
CORN	28.76	19.86	28.44	2.24	1.83	28.42

Table 5
Summary of computation times of the proposed LSTM variants for the CIF2016 dataset (in minutes).

LSTM variant	Pre-processing	Model training	Post-processing	Total time
LSTM.All	4.4	26.2	0.9	31.5
LSTM.Horizon	4.0	22.2	0.8	27.0
LSTM.Cluster.kMeans.Silhouette	3.6	20.4	0.8	24.8
LSTM.Cluster.kMeans.Elbow	3.4	15.2	0.7	19.3
LSTM.Cluster.Dbscan	3.0	16.2	0.5	19.7
LSTM.Cluster.Snob	3.1	16.4	0.6	20.1
LSTM.Cluster.PAM.Silhouette	2.0	12.2	0.4	14.6

Table 9
Results of the fixed origin evaluation for the 111 daily series of NN5, ordered by the first column, which is the Mean sMAPE. For each column, the results of the best performing method(s) are marked in boldface.

Method	Mean sMAPE	Median sMAPE	Rank sMAPE	Mean MASE	Median MASE	Rank MASE
ES	**21.44**	**20.29**	5.13	**0.86**	**0.80**	4.66
ETS	21.46	20.57	5.07	**0.86**	0.81	4.58
BaggedETS	21.46	20.57	5.07	0.87	0.84	5.37
LSTM.Cluster.Snob	21.66	20.71	6.00	0.94	0.89	7.29
LSTM.Cluster.kMeans.Elbow	22.57	20.89	5.50	0.90	0.84	5.13
LSTM.Cluster.PAM.Silhouette	22.67	20.54	**4.87**	0.90	0.83	**4.41**
LSTM.Cluster.kMeans.Silhouette	22.72	21.02	5.39	0.90	0.83	5.04
LSTM.Cluster.Dbscan	22.79	21.30	5.35	0.90	0.83	4.55
LSTM.All	23.46	21.75	7.46	0.96	0.93	7.84
ARIMA	25.29	21.74	7.29	0.97	0.90	6.78
Naïve Seasonal	26.49	23.31	8.86	1.01	0.94	10.34

Table 11
Summary of computation times of the proposed LSTM variants for the NN5 dataset (in minutes).

LSTM Variant	Pre-processing	Model training	Post-processing	Total time
LSTM.All	47.6	90.5	0.6	138.2
LSTM.Cluster.kMeans.Silhouette	34.3	57.0	0.8	92.1
LSTM.Cluster.kMeans.Elbow	24.2	47.2	0.7	72.1
LSTM.Cluster.Dbscan	24.3	50.5	0.7	75.5
LSTM.Cluster.Snob	20.4	35.4	0.4	56.2
LSTM.Cluster.PAM.Silhouette	36.6	60.6	0.8	98.0

Figure 44: Forecasting Across Time Series Databases using Recurrent Neural Networks on Groups of Similar Series: A Clustering Approach. Original from Bandara et al[92].