ZOLTÁN NAGY – JÁCINT JUHÁSZ

DEMAND PLANNING FOR BUILDING ENGINEERING PRODUCTS – A CASE STUDY WITH TRANSFORMER-BASED NEURAL NETWORKS

ÉPÜLETGÉPÉSZETI TERMÉKEK KERESLETTERVEZÉSE – ESETTANULMÁNY TRANSZFORMER ALAPÚ NEURÁLIS HÁLÓZATOKKAL

Efficient demand planning holds critical significance for businesses. In this research, the authors investigate the applicability of the Temporal Fusion Transformer, a neural network-based model, to address demand planning challenges. Specifically, they explore the potential benefits of incorporating additional information related to product characteristics and sales channel types. The primary objective of this study is to assess the advantages gained by incorporating these supplementary variables. The dataset utilized in this analysis originates from a company predominantly engaged in the sale of building engineering products. The authors initially focus on static attributes such as product groupings and time-varying attributes such as sales channel variations. This paper's contribution lies in its comprehensive case study, which applies the Temporal Fusion Transformer model to a real-world demand planning problem of the company, including all its specifications and customizations.

Keywords: **time series forecasting, demand planning, neural networks**

A hatékony keresletttervezés kritikus jelentőségű a vállalkozások számára. Ebben a kutatásban a Temporal Fusion Transformer neurális hálózat alapú modell alkalmazhatóságát vizsgálják a szerzők a keresletttervezési kihívások kezelésére. Konkrétan megvizsgálják a termékjellemzőkkel és értékesítési csatornatípusokkal kapcsolatos további információk beépítésének lehetséges előnyeit. Kutatásuk elsődleges célja e kiegészítő változók beépítésével nyert előnyök felmérése. Az elemzésben felhasznált adatállomány egy túlnyomórészt épületgépészeti termékek értékesítésével foglalkozó cégtől származik. Kezdetben a statikus tulajdonságokra, például a termékcsoportokra, majd az időben változó tulajdonságokra, például az értékesítési csatornaváltozásokra összpontosítanak. A tanulmány fő hozzájárulása az átfogó esettanulmány, amely alkalmazza a Temporal Fusion Transformer modellt a vállalat keresletttervezési problémájára, az összes specifikációjával és testreszabásával együtt.

Kulcsszavak: **idősor előrejelzés, keresletttervezés, neurális hálózatok**

Funding/Finanszírozás:
This paper was financially supported by the Project "Network of excellence in applied research and innovation for doctoral and postdoctoral programs / InoHubDoc", a project co-funded by the European Social Fund financing agreement no. POCU/993/6/13/153437.
Ezt a tanulmányt a „Kiválósági hálózat az alkalmazott kutatásban és innovációban doktori és posztdoktori programokban / InoHubDoc" elnevezésű projekt anyagilag támogatta és társfinanszírozta az Európai Szociális Alap (POCU/993/6/13/153437).

Authors/Szerzők:

Dr. Zoltán Nagy[a] (Zoltan.Nagy@aut.utcluj.ro) teaching assistant; Dr. Jácint Juhász[b] (jacint.juhasz@econ.ubbcluj.ro) assistant professor

[a]Technical University of Cluj-Napoca (Kolozsvári Műszaki Egyetem) Romania (Románia); [b]Babeş-Bolyai University (Babeş-Bolyai Egyetem) Romania (Románia)

The article was received: 02. 10. 2023, revised: 10. 10. 2023, accepted: 11. 10. 2023.
A cikk beérkezett: 2023. 10. 02-án, javítva: 2023. 10. 10-én, elfogadva: 2023. 10. 11-én.

Forecasting holds a central position within supply chain management due to its profound significance and the broad-ranging effects it has on various viewpoints of the supply chain. Accurate analysis and forecasting yield valuable insights into future demand, empowering companies to make well-informed decisions concerning production scheduling, inventory control, and resource allocation. By foreseeing customer requirements and market trends, organizations can optimize their operations, cut down expenses, and elevate customer satisfaction levels. The practice of forecasting empowers businesses to synchronize their supply with demand, averting situations like stockouts or excessive stockpiling. Moreover, it fosters effective collaboration with suppliers, facilitating the timely procurement of materials and components, trimming lead times, and cultivating stronger supplier relationships. Also, forecasting is a crucial risk management tool, identifying disruptions and enabling proactive responses.

The importance of forecasting accuracy is highlighted in the J.P. Morgan Working Capital Index Report 2022 (Shah et al., 2022). The report analyzed financial data from S&P 1500 companies spanning 20 industries, collectively accounting for approximately 90% of the total U.S. market capitalization.

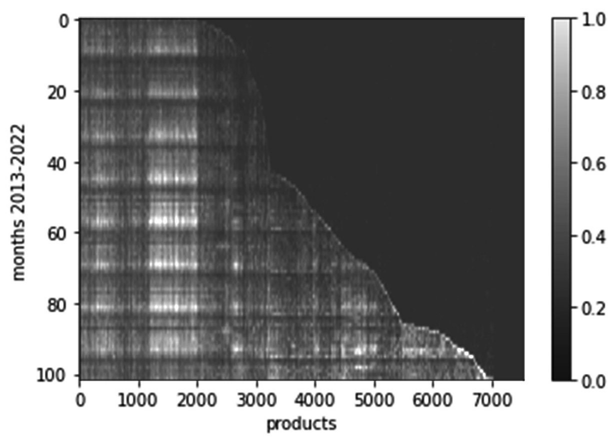
The companies were categorized into three groups based on their working capital metrics: the top 25%, the middle 50%, and the bottom 25%. The analysis indicated that the bottom 25% needed to enhance their performance to reach the industry average. This enhancement uncovered an overinvestment in working capital amounting to $523 billion by the end of 2021. Since working capital is defined as the combination of inventory and accounts receivable, adjusted for accounts payable (which are nearly identical), this suggests a close relationship between inventory and working capital. Consequently, optimizing working capital aligns closely with optimizing inventory, underscoring the significance of effective working capital management. Additionally, PWC's report from 2021 (Windaus & Tebbett, 2021) found that global overinvestment in working capital over 5 years amounted to $1.2 trillion. U.S. companies demonstrated inventory performance indicators that were 33% better than those in Europe and 46% better than in Asia.

The effect of working capital on small and medium-sized companies is more pronounced, as (Nobanee, 2017) highlighted in a study of 5802 US non-financial firms. His finding reveals a negative and highly correlated relationship between the cash conversion cycle and profitability for small firms, in contrast with a weak positive relationship for medium and large firms. This relationship is the focus of much empirical research in various regions and countries all around the world. (Deloof, 2003) studied 1000 Belgian firms between 1992-1996 suggesting value creation possibility in reducing inventory and shortening receivables period. Another research paper, (Lazaridis & Tryfonidis, 2006), came to the same conclusion for companies in Greece, highlighting the value creation by optimizing the components of the cash conversion cycle. A broader study (Garcia et al., 2012) analyzed 2974 non-financial companies from 11 European countries for 12 years (1998-2009) and concluded that working capital decisions have a significant effect on profitability regardless of country, industry, and fiscal year.

On the other hand, the correlation between forecast accuracy and service level remains a topic of discussion, frequently reliant on management choices about inventory management strategies. A lot of companies opt for a fixed-cycle service level approach. In these scenarios, forecast accuracy and service level work in conjunction to establish the safety stock, consequently impacting the overall cost associated with the inventory policy.

(Barros et al., 2021) assessed nine distinct methods for calculating safety stock, each of which demonstrated an exponential relationship involving the service level and the standard deviation of the forecast error, incorporating different adjustment factors. Conversely, certain management strategies favor an optimization-based approach to service level, aiming to find an equilibrium point between inventory expenses and potential sales loss. Even when examining a limited set of variables, the connection between inventory and service level follows an exponential trend. This was also highlighted in the research conducted by (Jeffery et al., 2008) where a boost of 10% in service level resulted in a 100% increase in inventory.

Figure 1
Normalized portfolio over the months from 2013 to 2022



Source: own compilation

Our study focuses on demand planning based on time series forecasting using historical demand data, product characteristics, and external factors. In these situations, is crucial to understand the difference between sales and demand data, as in general the company rarely possesses demand information and often there is confusion with sales data, replacing sales with demand. Such confusion leads to shifting the forecast from demand to sales target and there are significant differences between these two. Demand planning aims to predict future customer demand for a company's products or services. It provides insights into what customers are likely to purchase in terms of quantity and timing, while sales target setting involves establishing specific, measurable sales goals or targets that a company aims to achieve within a defined period and often incorporates specific business logic. For example, a business logic might forbid decreasing trends, but demand planning could indicate such trends.

Drawing upon the previously mentioned concepts, we have emphasized the significance of accurate forecasts. In the upcoming section, we introduce the specific issue we investigated in our case study.

Problem statement

In this paper, we address the demand planning problem faced by a company specializing in building engineering products. With an inventory of approximately 8,000 critical items, maintaining their availability while avoiding overstocking is essential. Thus, accurate forecasting is crucial for the company's seamless operation. To achieve this, we conducted a thorough analysis of a dataset obtained from the company aiming to develop a tailored and precise forecasting model.

On the other hand, the company is experiencing rapid expansion, diversifying its product offerings significantly. This trend is clearly illustrated in Figure 1, which displays the normalized extension of the product portfolio from 2013 to 2022, showcasing the quantities sold. In Figure 1, the dark values represent nearly zero values, while the light values approach or reach one. This portfolio analysis reveals several noteworthy insights. Firstly, the portfolio has expanded in 2022 more than threefold from its initial state in 2013. Secondly, some products exhibit distinct seasonal patterns, repeating every twelve months.

In our specific study, we have focused on conducting an analysis exclusively on products that have remained in the portfolio for at least six years. This selection process yielded a total of 2625 products that formed the basis of our analysis.

The company places a strong emphasis on the 6-month forecast horizon, viewing it as the most critical timeframe for its operations. To ensure the accuracy of these forecasts is closely monitored, they conduct biweekly evaluations. It's worth highlighting that while long-term forecasts, such as those for 12 or 18 months, are indeed significant, they are carried out just once a year and require extensive manual adjustments and speculative analysis. In such cases, even the most advanced methods have limited usability.

This paper's primary contributions encompass three key elements: Firstly, it entails an in-depth case study conducted on the company's dataset. Secondly, it examines the company's portfolio with a neural network model. Lastly, the paper demonstrates the potential benefits achievable through the incorporation of various supplementary products and sold quantity information.

The findings of the current work hold significant relevance for both practitioners and researchers. For practitioners, both the model itself and the resultant outcomes demonstrate their utility in real-world business environments, underscoring the practical applications and opportunities they offer. Simultaneously, for researchers engaged in this field, these findings are valuable as they demonstrate the potential of incorporating covariates in the modeling process of the time series.

In the subsequent sections, we aim to conduct a literature review concerning the cutting-edge models currently employed in the creation of time series forecasting models.

Literature review

In recent years, the field of time series modeling and forecasting has gained substantial attention, owing to its increasing importance and relevance across various domains. One pioneering work in this field, (Anderson, 1977), introduced the Auto-Regressive Integrated Moving Average (ARIMA) model, revolutionizing the approach to time series analysis. ARIMA establishes a vital link between present and past data points, particularly excelling in modeling stationary time series.

Another notable contribution, (Gardner Jr, 1985), presented exponential smoothing as an alternative to the traditional ARIMA model, well-suited for handling non-stationary time series. Extensions and adaptations of these models have since evolved, such as seasonal ARIMA (Hipel et al., 1977), designed for seasonality, and Holt-Winter's model (Chatfield, 1978), extending exponential smoothing to handle seasonality.

Despite their utility, it's important to highlight that fine-tuning these models often requires expert knowledge for optimal parameter selection, a crucial consideration in practical applications.

The Generalized Additive Model (GAM) represents another valuable approach to time series modeling and forecasting, as discussed in (Hastie & Tibshirani, 1987). A GAM model decomposes a time series into its fundamental components, including elements like trend and seasonality. These models can be conceptualized as curve-fitting tools, with regression techniques employed to determine the associated parameters, as highlighted in (Hong & Wang, 2014). Although GAMs have enjoyed recognition for some time, they were recently rediscovered (Taylor & Letham, 2018) with the introduction of a novel modeling methodology known as *Prophet*. Their inherent simplicity and transparency set these models apart, as they define a time series by breaking it down into its constituent elements. Also, GAM models offer the advantage of analyzing each component individually, facilitating the identification of changes in trends and their correlation with specific events or factors.

Over the past few years, there have been new methods created that take advantage of neural networks for predicting and modeling time series. As neural networks have improved, they have become more commonly used for analyzing time series. While Long Short-Term Memory (LSTM) networks have existed for a while, they were limited by the tools available for machine learning. But now, with the development of simple and easy-to-use frameworks like TensorFlow, Keras, and PyTorch, researchers can apply neural network-based solutions more efficiently to forecast and analyze time series.

Various models employing specific LSTM network architectures have been explored for time series forecasting. For instance, (Chimmula & Zhang, 2020) treated the COVID-19 transmission as a time series problem and used an LSTM network for modeling. Additionally, (Altan & Karasu, 2022) focused on crude oil prediction and employed LSTM in their model. Similarly, in the work by (Nguyen et al., 2021), both forecasting and anomaly detection were addressed within the context of supply chain management.

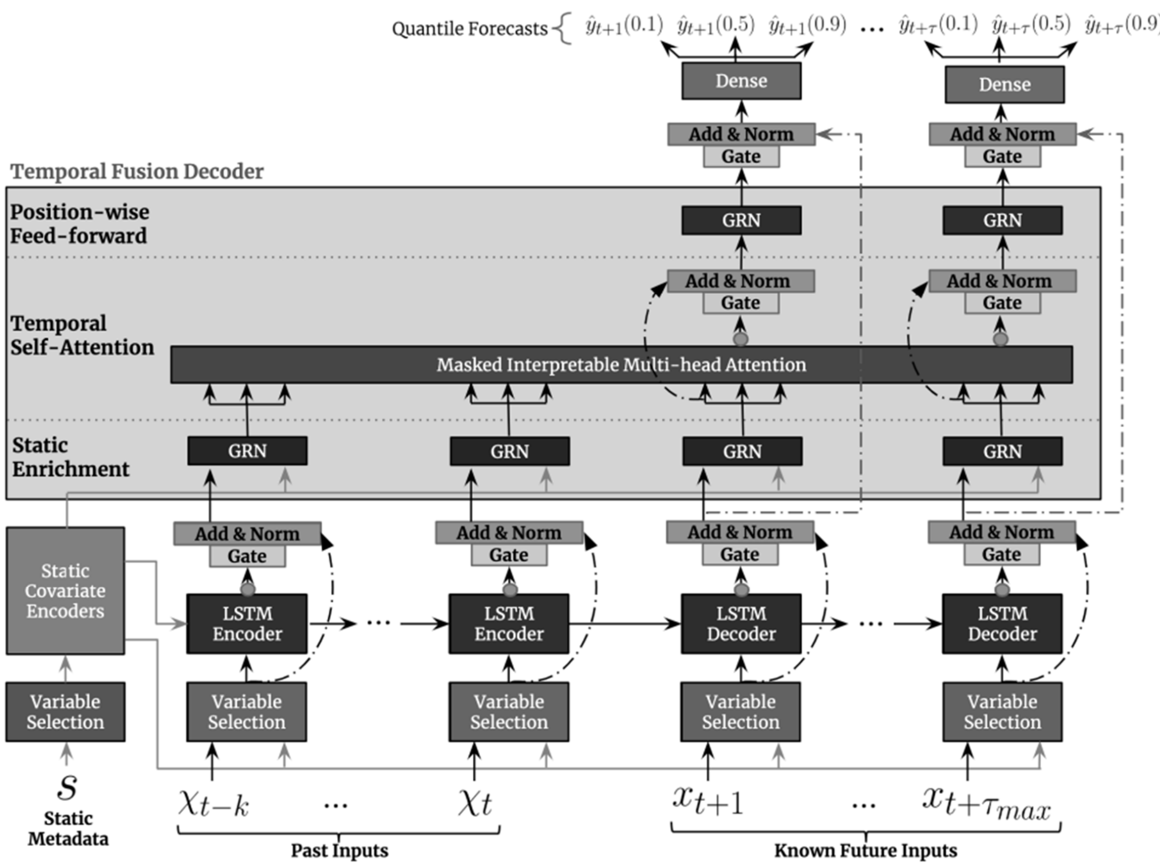
It's important to highlight that the models mentioned earlier primarily focus on univariate modeling, meaning they handle individual time series in isolation without considering their interconnections. However, a recent and intriguing model known as the Temporal Fusion Transformer (TFT) was introduced in (Lim et al., 2021). TFT is an attention-based model that employs the novel encoder-decoder strategy called the transformer, as originally presented in (Vaswani et al., 2017). One notable advantage of TFT is its ability to model multi-horizon and multi-input data, allowing it to learn from other data points beyond its own. TFT can also improve forecasting accuracy by incorporating additional covariates, such as historical and geographical information or other additional information that could help improve forecasting accuracy. With its dual capabilities of generating forecasts for datasets and effectively utilizing supplementary information, the TFT model emerges as a compelling choice for addressing demand planning challenges. For example, (Wu et al., 2023) successfully employed this approach in an application where they leveraged the benefits of covariates within the TFT to predict and model tourism demand during the COVID-19 pandemic.

Several alternatives to the TFT exist in the field of time series forecasting. One such alternative is DeepAR (Salinas et al., 2020), developed by Amazon, which employs an autoregressive model using LSTM-based neural networks. This model also integrates covariates and delivers probabilistic forecasts. However, it falls short in one important aspect when compared to TFT: it cannot provide feedback regarding the significance of covariates, a feature offered by TFT through its attention-based layers. Another similar model is N-BEATS (Oreshkin et al., 2020), which employs a deep neural architecture based on both backward and forward residual links, combined with a deep stack of fully-connected layers. A drawback of N-BEATS is its inability to handle extra covariates. The NHITS model (Challu et al., 2023) is specifically designed for forecasting time series with long forecasting horizons and can also handle covariates. However, it falls short of providing probabilistic or quintile forecasts.

Overall, when comparing these recent models, the

Figure 2

TFT architecture



Source: Lim et al. (2021, p. 1751)

TFT model stands out as the most promising option due to its comprehensive feature set. These include its capability to produce quintile forecasts, manage additional covariates, and simultaneously handle multiple time series. Our primary objective in this study is to thoroughly investigate the potential of the TFT model. Our initial hypothesis is that by using these extra covariates the forecasting accuracy could be improved.

Next, we present briefly the TFT model which is followed by the case study developed with the dataset obtained from the company. Note that, we refer to a time-stamped quantity vector as a *time series* and the term *dataset* to refer to a set of time series. Additionally, the terms *covariate* and *feature* are used interchangeably in the manuscript, referring to extra information about the time series.

Temporal Fusion Transformer (TFT) models

A recent breakthrough in this field is the introduction of a transformer-based solution, as highlighted in the study (Lim et al., 2021). The TFT model introduces an innovative attention-based structure that combines advanced multi-horizon forecasting with interpretable insights into temporal dynamics. Compared to other neural network architectures, TFT offers several unique features.

In addition to the traditional time-value vectors, this model allows for the incorporation of covariates, which are supplementary inputs that enhance its performance. These covariates fall into several categories: time-dependent or static, real or categorical, and known or unknown. Time-dependent covariates refer to variables that change with time; for instance, weather conditions can be considered time-dependent, whereas the material composition of a product remains static, such as being made of copper for example. Moreover, weather can be represented as a real value due to its numerical nature, while months can be categorized as discrete features since their possible values are predefined (e.g., January, February, etc.). Lastly, the known or unknown classification relates to whether we possess future knowledge about a covariate. Depending on a company's policies, we may or may not have information about the future price of a product, but one certainty is that we lack knowledge about future sales quantities.

Based on (Lim et al., 2021), the architecture of the TFT model can be observed in Figure 2.

Furthermore, the TFT also helps us to understand the importance of several of these covariates. The model gives feedback on which features were the most important in the generation of the forecasts. It gives a ranking with percentages based on the importance of these features. In comparison to other neural network-based methodologies, such as the example presented in (Nguyen et al., 2021), where the importance and impact of individual factors on the forecast remain undisclosed, TFT offers a distinctive advantage. With TFT, this information is available after the training, empowering us to make more informed decisions when considering the incorporation of new features. Moreover, the model also highlights the significance of the temporal vector, pinpointing precisely which segment of the historical time series played the most pivotal role in the forecast.

Another notable benefit of this model is its capacity to provide more than a mere point forecast; it also generates a quantile forecast, thereby furnishing valuable metrics that quantify the level of confidence associated with the forecast.

Before starting the model training, a series of hyperparameters can be configured regarding the model. While we won't delve into the intricate specifics here, we would like to emphasize a few of them. For an in-depth understanding, we recommend referring to (Lim et al., 2021).

Among the hyperparameters available for fine-tuning, the learning rate stands as a pivotal parameter. It plays a critical role in determining the initial pace at which the model progresses towards approaching its near-optimal parameters. To elaborate further, if the learning rate is set excessively low, the model may struggle to converge to a local optimum, potentially stalling its training progress. Conversely, if the learning rate is overly large, the model might exhibit erratic behavior, oscillating around local minima without effectively settling on an optimal solution.

Another crucial hyperparameter is the dropout rate. Dropout is a regularization technique employed to prevent overfitting in neural networks. This rate determines the proportion of randomly selected neurons that are temporarily dropped out during each training iteration. It helps enhance model generalization by introducing an element of uncertainty, preventing any single neuron from becoming overly specialized on the training data. In the case of our analysis to find the near-optimal hyperparameters we used an automatic tuning process, the *optuna* (Akiba et al., 2019).

Based on the advantages of the TFT, we consider that this model architecture represents an interesting modeling approach capable of offering valuable insights into our data. In the subsequent section, we delve into the details of the training process.

Training process

To guarantee a comprehensive and reliable evaluation of our models, we divided the dataset into three distinct segments: the training set, the validation set, and the testing set. Each of these segments plays a crucial role in ensuring the quality and effectiveness of our models. The training dataset primarily serves the purpose of deriving model parameters, while the validation dataset plays a crucial role in evaluating the model's performance and mitigating overfitting risks.

Furthermore, an epoch represents a single complete pass through the entire training dataset during the training of a neural network. It is a fundamental unit of iteration in training, helping the model update its parameters by learning from each data point.

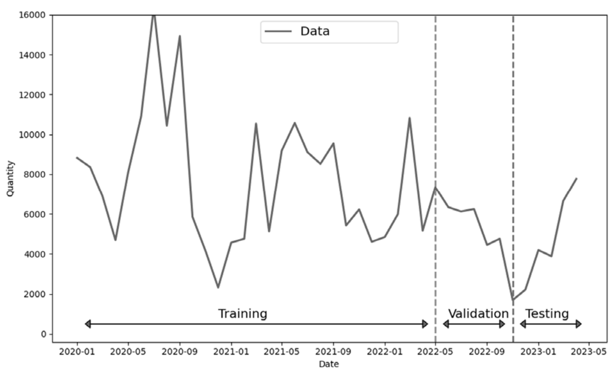
Overfitting occurs when a model is exclusively trained on the training dataset, gradually improving its performance in each epoch without any external reference or control. To prevent this scenario, we incorporate the validation dataset into the process. We continuously assess the model's performance on this independent dataset, ensuring that it doesn't overly adapt to the noise present in the training data. This precaution is essential because the noise encountered in the training data may differ from that in the validation data, and by monitoring both, we aim to create a more robust and generalizable model.

Lastly, the testing dataset remains untouched during the model development phase. It serves as an objective benchmark, providing an unbiased measure of the true performance of our models. This dataset offers a critical assessment of the models' capabilities on unseen data, assuring the reliability and generalizability of the results.

It's worth noting that similar approaches to data partitioning have been employed in several studies (El Bourakadi et al., 2023; Lim et al., 2021). For a visual representation of this data partitioning strategy for a specific product, please refer to Figure 3.

Figure 3

Splitting the data to training, validation, and test



Source: own compilation

Numerous software packages are available for constructing a TFT model. For our study, we harnessed the Python environment, leveraging the PyTorch package, more precisely, the PyTorch Lightning and PyTorch Forecasting packages.

To maintain consistency in initializing the model's parameters and eliminate randomness, we employ identical initialization with random values across all iterations. Furthermore, we ran all of our training for 50 epochs, from which we selected the best-performing parameter set with respect to the validation.

Performance measures

To evaluate the forecast's accuracy, we use the Symmetric Mean Absolute Percentage Error (SMAPE) metric (Koutsandreas et al., 2022). The SMAPE formula is as follows:

$$SMAPE = \frac{1}{N}\sum_{k=1}^{N}\frac{2|y(k)-\hat{y}(k)|}{|y(k)|+|\hat{y}(k)|}$$

$$SMAPE(\%) = \frac{1}{N}\sum_{k=1}^{N}\frac{2|y(k)-\hat{y}(k)|}{|y(k)|+|\hat{y}(k)|}100(\%)$$

In this equation, represents the true value at point , and denotes the estimated value. SMAPE provides symmetrical results, treating scenarios where the true value is, for instance, 5 and the forecasted value is 100, the same as cases where the true value is 100 and the forecasted value is 5.

Moreover, SMAPE inherently yields values within the range of 0% to 200%. However, it's important to note that the SMAPE function itself is not symmetric in the sense that it assigns higher values for underestimation and lower values for overestimation. For instance, with a true value of 100 and a forecasted value of 140, the resulting SMAPE is 16%. Conversely, with a true value of 100 and a forecasted value of 60, the SMAPE becomes 25%, despite both cases having the same absolute difference between the true and forecasted values.

This distinction between underestimation and overestimation is useful from our business perspective as lost sales due to understocking do more harm than incremental costs generated by overstocking. The relation between lost sales and increased inventory cost generally has a convex form, based on (Janakiraman & Roundy, 2004), and it is difficult to express in a simple formula. In our study, we believe that by using SMAPE, we can effectively highlight the significance of preventing stockouts, thereby favoring the outcome of overstocking.

Data preprocessing

In this section, we delve into the details of the problem. Our analysis encompasses the portfolio, resulting in a dataset comprising 2,625 products. These products have been included in our analysis because they possess data spanning at least six years. The data we are examining covers the period from January 2013 to October 2022 (see Figure 1).

Predictions in the TFT model are based on a collection of input values known as encoders. The choice of encoder length is a hyperparameter, and in our case, we employed a 12-month encoder length due to the yearly periodicity of the data and its monthly granularity. Moreover, the prediction horizon is set at 6 months, as it represents the primary and frequently used forecasting timeframe of the company. In the subsequent sections, we will outline our approach to addressing missing values within the dataset and provide a comprehensive exploration of the dataset's features.

Handling missing and negative values

Given the extensive dataset, some months lack sales for certain products. Regression models handle this well, but models like TFT rely heavily on prior values. While various strategies exist in the literature for addressing this challenge, see e.g., (Bashir & Wei, 2018; Liu et al., 2020), we won't delve into these details here. We refrain from delving into extensive details about handling missing values because it hinges on understanding the underlying reasons for those gaps, necessitating a deeper grasp of the business logic and product portfolio. To handle missing data, we set sales to 0 for dates with no recorded sales, ensuring forecasts closely align with reality.

On the other hand, in certain instances, negative data points may appear, indicating anomalies within the system. These anomalies should also be considered in the context of business logic. However, given the main topic of this manuscript, we address this issue by replacing negative values with zeros. In our future research endeavors, we intend to delve deeper into these cases and explore them more thoroughly. Nevertheless, for the current paper, this topic falls outside its scope.

Dataset

The dataset consists of three primary columns: *Date, ItemID,* and *OrderedQty*. The *ItemID* column is used for product identification, while the *OrderedQty* column records the quantity of the product sold. The *Date* column indicates the month in which the sales quantity is recorded. In addition to the aforementioned columns, we also have a set of static covariate columns, namely *MGName, GroupName, SubGroupName*, and *ReqGroupID*. *MGName* designates the primary category to which the product belongs, such as Installation products, Interior design products, Electronic products, DIY products, and more. The *GroupName* column provides details about the specific category within the main category. For instance, within the Installation products' main category, we have the *Pipes* group. *SubGroupName* further narrows down the specification within the group; for instance, it might specify *copper pipes*. Lastly, *ReqGroupID* indicates the product's priority level.

Furthermore, we also consider time-varying covariates in our analysis. Initially, the *OrderedQty* represents the aggregated quantity sold, consolidating sales across all distribution channels. To introduce additional features, we incorporate information about the specific channels through which these products were sold. In this company, three primary channels exist: *Shop* for in-store purchases accessible to any walk-in customer, *Eshop* representing sales on the company's website, and *B2B* referring to sales to other retailers who subsequently sell the products to end clients.

In our analysis, we have identified both static and time-dependent variables that we have included as covariates. Our objective is to explore the benefits derived from incorporating these additional covariates. To achieve this, we will investigate four specific scenarios in the subsequent sections:

- Case 1 - Using base information: product id, date, and quantity
- Case 2 - Using base information and static covariates
- Case 3 - Using base information and time-varying covariates
- Case 4 - Using base information with both static and time-varying covariates

Now we are ready to present the main findings of the paper.

Main results

Within this section, we will comprehensively explore each distinct scenario, commencing from Case 1 and progressing through to Case 4. Our objective is to provide a thorough account of our observations and findings for each case, ensuring a comprehensive presentation of our results.

To ensure reproducibility, we employed the following model hyperparameters, which were derived from the base information. You can find these hyperparameters outlined in Table 1.

Table 1

Hyperparameters

Max epoch	*50*
Gradient clip value	*0.8033*
Learning rate	*0.0111*
Hidden size	*70*
Attention head size	*2*
Dropout	*0.2903*
Hidden continuous size	*16*

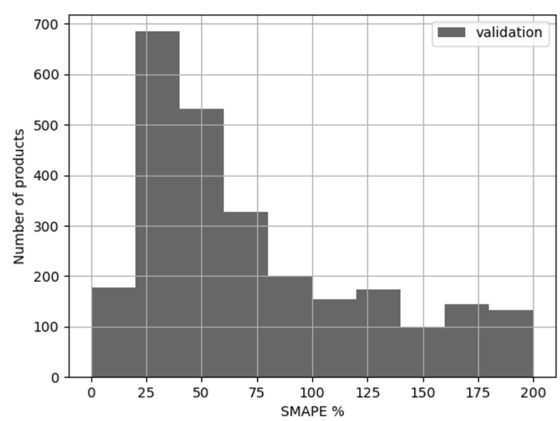
Soure: own compilation

Case 1

In Case 1, we used the training data to determine the parameters while using the validation set to prevent overfitting. The model underwent 50 training epochs, during which we selected parameters from the epoch that exhibited the lowest quintile loss on the validation dataset. Our findings from this process are reflected in Figure 4, where we present a histogram of these SMAPE percentages on the validation data.

Figure 4

Histogram of SMAPE % of the products on the validation data



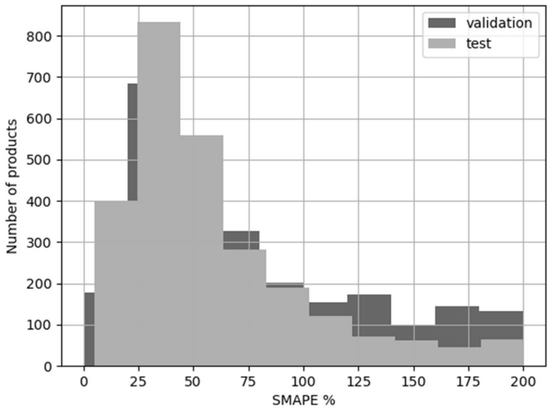
Source: own compilation

We can observe that nearly half of the products exhibit SMAPE values below 50%. The median SMAPE stands at 58.74%, while the mean SMAPE is 75.93%.

Additionally, Figure 5 illustrates the SMAPE histogram for the untouched test data along with the validation data. In this scenario, the median SMAPE is 47.02%, with a mean of 60.14%. Notably, an interesting pattern emerges: the model's performance on the test data surpasses that on the validation data. This phenomenon could potentially be attributed to the selling seasonality, primarily occurring in August, September, October, and November. As the test data spans from May 2022 to October 2022, it encompasses the peak of this season, likely contributing to the improved performance.

Figure 5

Histogram of SMAPE % of the products on the validation and test data

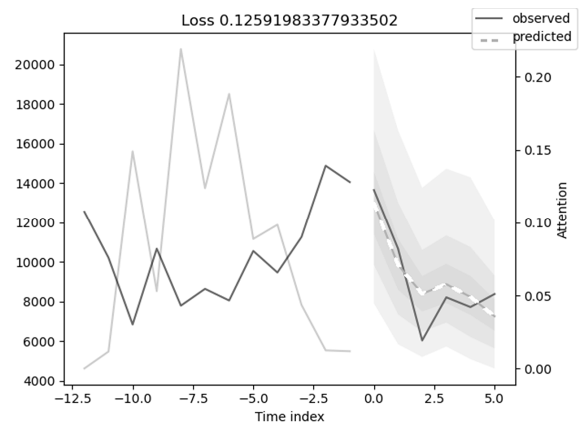


Source: own compilation

Moreover, we delve into a more detailed examination of one of the top-performing predictions, as displayed in Figure 6. In this visual representation, the true values are depicted in black, while the forecasted values are presented with dashed grey color. Notably, the plot also includes shaded regions of grey, indicating the forecast's quantiles. These quantiles serve the purpose of maintaining confidence in the forecast, proving particularly valuable in scenarios like "what if" analyses, where decision-makers, such as managers, may seek to enhance the sales of specific product types.

Figure 6

Prediction compared to validation – product id: anon101



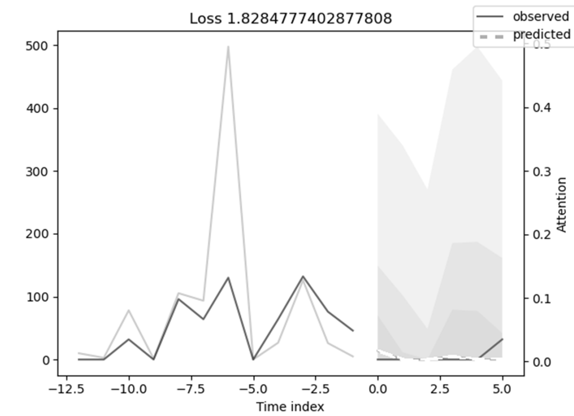
Source: own compilation

It's worth noting that the data used for generating the forecast comprises the most recent 12 months of training data, with a forward projection spanning 6 months (prediction length). Additionally, a light grey line is visible, underscoring the significance of specific data points that played a pivotal role in shaping the forecast. These significance values are presented on the y-axis of the right-hand side. Furthermore, the displayed loss corresponds to the SMAPE quantile loss, represented in decimal format. To clarify, a value such as 0.1259 should be understood as 12.59%, encompassing not only the point loss but also accounting for the level of confidence in the forecast.

Conversely, we can examine one of the least accurate forecasts, as depicted in Figure 7.

Figure 7

Base features, prediction compared to validation - product id: anon801

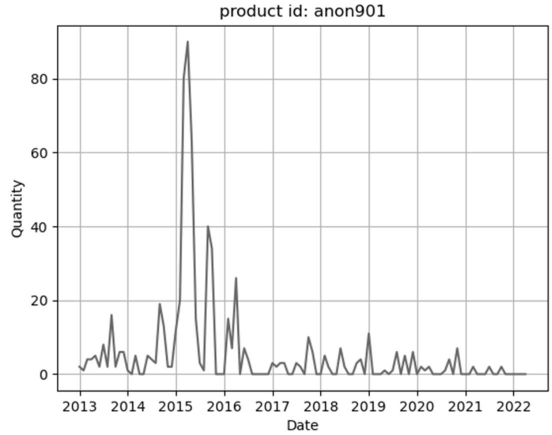


Source: own compilation

It's worth noting that the actual prediction doesn't appear significantly flawed. However, when considering the quantile loss, which accounts for the inclusion of quantiles, the outcome is notably unfavorable, reaching 182.84%, approaching the upper limit of 200% of the SMAPE.

Figure 8

Product data with no clear pattern

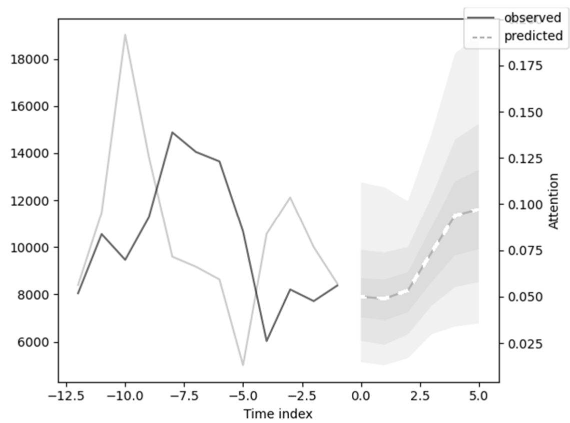


Source: own compilation

Henceforth, we categorize these products as challenging to forecast, as they do not exhibit a clearly repeatable pattern. Notably, these are the products that once had substantial sales quantities in the past but are presently less popular. This trend is evident in Figure 8. Moreover, a group of products shares this characteristic, and this pattern is also apparent in Figure 4, where we observe a significant cluster of products with forecast errors approaching 200%.

Comparable findings are evident when extending the forecast into the later stages of the test data period, as illustrated in Figure 9. Notably, the forecast closely replicates the initial behavior of the first few months, characterized by an upward trend.

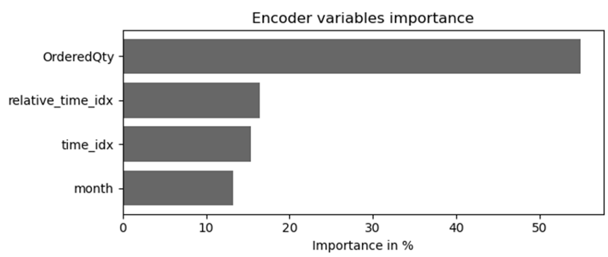
Figure 9
Forecast of product anon101



Source: own compilation

The significance of the variables can be seen in Figure 10. From this visualization, we can assert that the most influential variable is the quantity sold (OrderedQty), followed by the *relative_time_index*, which for each sampled sequence spans from -*encoder_length* (12 in our case) to *prediction_length* (6 in our case). The subsequent crucial variable is the *time_idx*, signifying the time index stretching from the initial date and time in the dataset to the final date and time. In this context, January 2013 corresponds to 0, while October 2022 corresponds to 117.

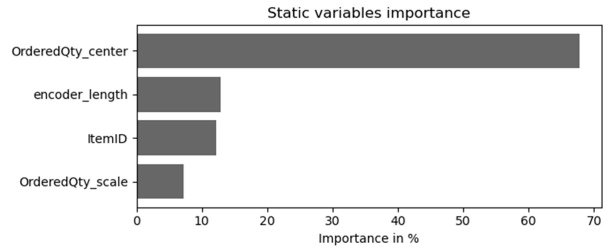
Figure 10
Encoder variables importance – Case 1.



Source: own compilation

Moreover, the model produces certain static variables, as depicted in Figure 11. From this representation, we can discern that when generating forecasts, the most pivotal element is the mean value of sold quantities (*OrderedQty_center*), whereas the scale factor exerts a comparatively lesser influence.

Figure 11
Static variable importance - Case 1

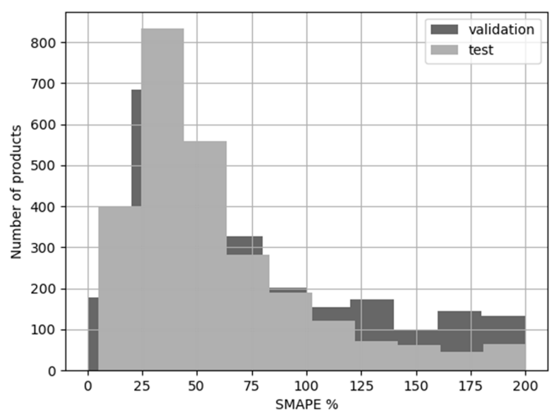


Source: own compilation

Case 2

In this scenario, we incorporated the static variables, specifically *MGName, GroupName, SubGroupName,* and *ReqGroupID*, into the training process. By introducing these additional covariates while maintaining the same setup as in *Case 1*, we observed enhancements in the validation data results, with a median improvement of 56.54% and a mean improvement of 74.18%. Similarly, there was an improvement in the test data, with the median rising to 46.49% and the mean reaching 59.46%. The histograms of the test and validation data are presented in Figure 12.

Figure 12
Histogram of the validation and test measures obtained using the base and the static covariates

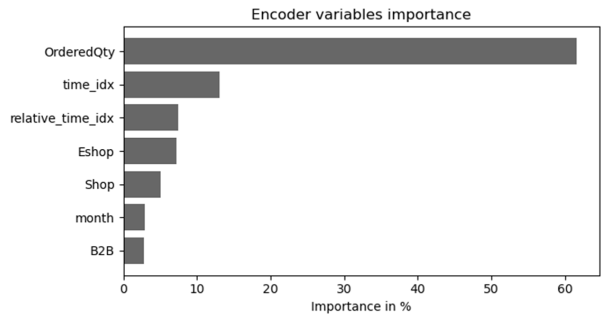


Source: own compilation

Conversely, we can delve into the importance of the static elements as depicted in Figure 13. Notably, *OrderedQty_scale* emerges as the most significant, whereas *OrderedQty_center* ranks among the least significant. Furthermore, it's worth noting that *MGgroup* is the second most significant element, although its importance in the overall context is approximately 10%. As a result of the seemingly erratic significance of these variables, we get the impression that the inclusion of static variables may not necessarily enhance the forecasting performance, despite the improved percentages observed on both the validation and test datasets.

Figure 13

Significance of static variables including the static covariates

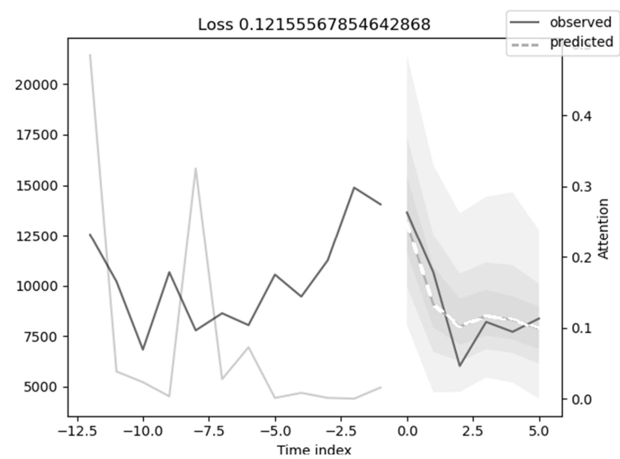


Source: own compilation

On the other hand, when we examine the good example from *Case 1*, we also observe a slight enhancement in the quintile loss, as depicted in Figure 14.

Figure 14

Base and static features, prediction compared to validation - product id: anon101



Source: own compilation

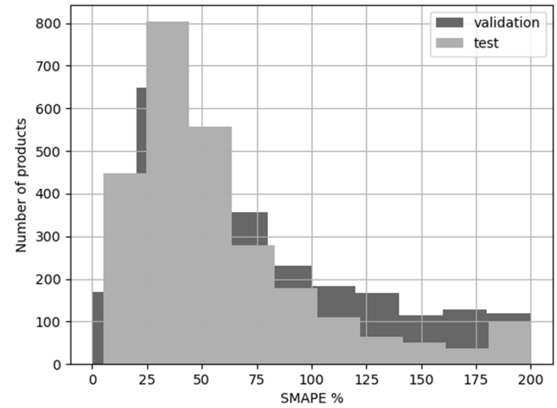
Case 3

In *Case 3*, we incorporated time-varying covariates that segment the sales quantity based on the channels of sale: *B2B, Eshop*, and *Shop*. This additional information led to an enhanced performance on the validation and the test data. We can visualize this improvement in Figure 15.

The median on the validation data is 59.46%, while the mean is 74.47%. Furthermore, the median on the test data is 46.38%, and the mean is 59.42%. The importance of the time-varying covariates can be seen in Figure 16. Unfortunately, our expectations were not met, as the newly introduced additional variables turned out to be the least significant. However, there is an improvement in the model's performance compared to both validation and test data.

Figure 15

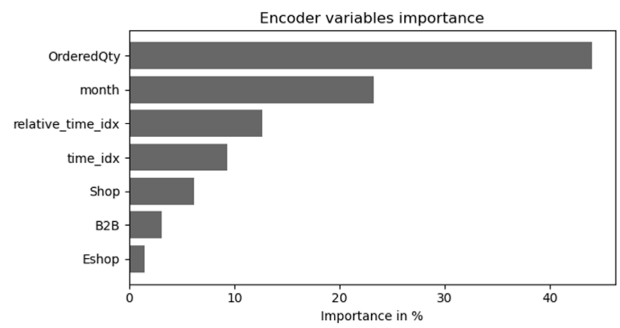
Histogram of the validation and test measures obtained using the base and the time-varying features



Source: own compilation

Figure 16

Time-varying features importance



Source: own compilation

Case 4

In this case, we incorporated both the static and the time-varying features that were examined in previous cases. The expectation was that the inclusion of these additional features would lead to improved forecasting accuracy, and to some extent, this expectation held true. On the validation dataset, we observed a median SMAPE of 53.5% and a mean of 70.06%, marking an enhancement compared to both Case 1, Case 2, and Case 3. However, the median SMAPE for the test data, at 48.66%, and the mean, at 63.42%, were somewhat higher than anticipated, indicating a less favorable outcome in this regard.

Table 2

Performance measures of Case 1-4

	Validation (median)	Validation (mean)	Test (median)	Test (mean)
Case 1	58.74%	75.93%	47.02%	60.14%
Case 2	56.54%	74.18%	46.49%	59.46%
Case 3	59.46%	74.47%	46.38%	59.42%
Case 4	53.50%	70.06%	48.66%	63.42%

Source: own compilation

According to the data in Table 2, it is apparent that the supplementary features generally lead to improvement; however, the obtained results suggest that there may still be room for further enhancements.

In this regard, we ensured comparability by maintaining a consistent set of hyperparameters across all cases. This approach was intended to facilitate a fair comparison among the various training scenarios. However, the outcomes did not align with our expectations, as the addition of extra information did not uniformly enhance forecast accuracy. To address this challenge, we once again employed the *optuna toolbox* to fine-tune the model's hyperparameters, this time considering the features from Case 4. We expect that this hyperparameter tuning, performed with the training and validation dataset of Case 4, will result in improved forecast accuracy. Next, we present this scenario.

Case 4, with improved hyperparameters

For this instance, we incorporated all available features during the model-tuning process. We can find the hyperparameters obtained for this case in Table 3, with the previously used values shown in parentheses for reference. The most notable alteration is in the *gradient_clip_value*, a hyperparameter designed to prevent exploding gradients. Gradient clipping ensures that gradient values remain within specified ranges. In the initial case, this value was substantial, whereas now it is significantly smaller. Furthermore, the *attention_head_size* has been reduced to 1, signifying that the model performs more effectively with a single attention head rather than employing multiple ones and subsequently aggregating their results.

Table 3
Hyperparameters obtained considering all covariates

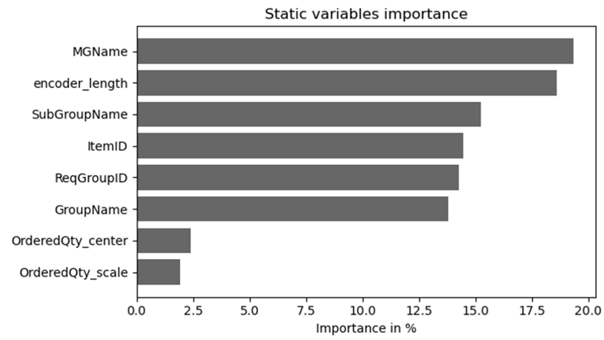
Max epoch	*50*
Gradient clip value	*0.0164 (0.8033)*
Learning rate	*0.0179 (0.0111)*
Hidden size	*47 (70)*
Attention head size	*1 (2)*
Dropout	*0.1686 (0.2903)*
Hidden continuous size	*14 (16)*

Source: own compilation

The validation dataset yielded a median SMAPE of 53.88% and a mean SMAPE of 69.66%, while the test dataset produced a median SMAPE of 45.80% and a mean SMAPE of 58.71%. To facilitate comparison, please refer to Table 4. Furthermore, in Figure 17 we can observe the importance of the static variables.

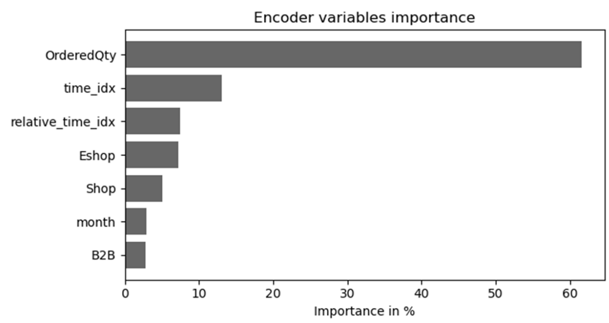
Now, we can observe that all the additional static features contribute to more than 10% of the overall importance. Interestingly, the least significant static variables are the center and scale of the ordered quantity. This suggests that the added static variables are indeed crucial, and the model recognizes their significance.

Figure 17
Static variables importance – Case 4 improved



Source: own compilation

Figure 18
Importance of time-varying features – Case 4 improved



Source: own compilation

In Figure 18, we explore the importance of the time-varying features. The most vital feature is *OrderedQty*, as expected, followed by time-related features: *time_idx* and *relative_time_idx*, which align with their importance for forecasting purposes. *Eshop* and Shop sales also exhibit noteworthy importance, indicating their relevance as features, while *B2B* sales are the least significant. Regarding the time-varying features, *Eshop, Shop*, and *B2B*, it's worth noting that from a business standpoint, the majority of sales (around 70-80%) occur through the *B2B* channel. Consequently, the *B2B* channel exhibits the strongest correlation with *OrderedQty*, explaining its lower importance. On the other hand, *Eshop* and Shop sales are considered somewhat less predictable, making their information valuable for enhancing the overall prediction of *OrderedQty* values.

Table 4
Performance measures with improved hyperparameters on Case 4

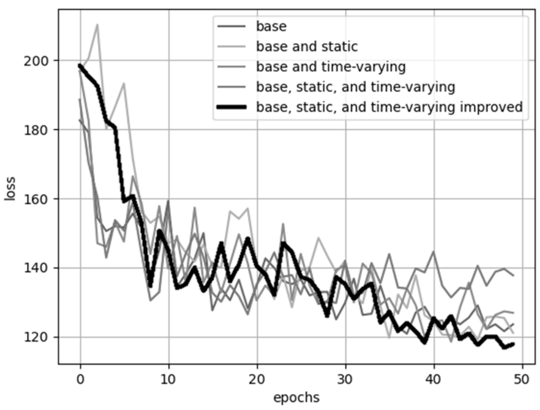
	Validation (median)	Validation (mean)	Test (median)	Test (mean)
Case 1	58.74%	75.93%	47.02%	60.14%
Case 2	56.54%	74.18%	46.49%	59.46%
Case 3	59.46%	74.47%	46.38%	59.42%
Case 4	53.50%	70.06%	48.66%	63.42%
Case 4 imp	53.88%	69.66%	45.80%	58.71%

Source: own compilation

Although the validation results closely resemble those of Case 4, there is a marked improvement in performance on the test data. To further examine the performance of the model, we present the loss function for all five cases in Figure 19. Note that the loss function employed here is the quintile loss, which means it evaluates not only point errors but also the corresponding quintile losses. It's noteworthy that the smallest loss is achieved when all features are used and the model's hyperparameters are tuned accordingly. This underscores a compelling conclusion: retraining the hyperparameters when introducing new features is essential.

Figure 19

Loss function evolution on the validation data across the epochs



Source: own compilation

Conclusions and Future Work

In this paper, the problem of demand forecasting was examined. The main objective of the paper was to present a neural network-based forecasting model. The considered model was one of the most recently developed, the Temporal Fusion Transformer (Lim et al., 2021). The problem under study was the demand forecasting problem of a company that mainly sells building engineering products. We used the available data from the company from January of 2013 to October of 2022. First, we examined the data, having several information from which we identified 4 cases. In Case 1 we used only the base data, having only the product id, the date, and the sold quantity of that product. Furthermore, in Case 2 we added static variables, and in Case 3 we added time-varying variables. Finally, in Case 4 we used both static and time-varying features. In the current study, we compared these four cases and we examined each of them in depth. We observed that in most of the cases, the model's performance was improved by using these extra features. Additionally, fine-tuning the model's hyperparameters to align with the dataset's features can lead to further enhancements in the model's performance. Therefore, the overarching conclusion is that augmenting the model with additional covariates can indeed boost its performance. This conclusion aligns with the findings of the related research as it is presented in (Lim et al., 2021; Wu et al., 2023).

There are several potential avenues for enhancing the model's performance in the future. Firstly, we can consider a more rigorous data separation, distinguishing between cases with reliable forecasts and those exhibiting random behavior, as indicated in Figure 8. Some products, for instance, may have once had high sales but now demonstrate declining quantities. Filtering out such cases can help mitigate their disruptive impact on the model's accuracy.

Another avenue for improvement involves a deeper exploration of forecast quantiles, enabling the generation of 'what if' analyses based on these forecasts. For instance, we can assess the probability that increased demand planning for certain products would meet customer needs. From a technical perspective, further enhancements can be achieved by incorporating additional features and fine-tuning hyperparameters accordingly. These represent just a few of the potential future directions stemming from our work.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–2631. https://doi.org/10.1145/3292500.3330701

Altan, A., & Karasu, S. (2022). Crude oil time series prediction model based on LSTM network with chaotic Henry gas solubility optimization. *Energy*, *242,* 122964. https://doi.org/10.1016/j.energy.2021.122964

Anderson, O.D. (1977). The Box-Jenkins approach to time series analysis. *RAIRO-Operations Research*, *11*(1), 3–29. https://doi.org/10.1051/ro/1977110100031

Barros, J., Cortez, P., & Carvalho, M.S. (2021). A systematic literature review about dimensioning safety stock under uncertainties and risks in the procurement process. *Operations Research Perspectives*, *8*, 100192. https://doi.org/10.1016/j.orp.2021.100192

Bashir, F., & Wei, H.L. (2018). Handling missing data in multivariate time series using a vector autoregressive model-imputation (VAR-IM) algorithm. *Neurocomputing*, *276*, 23–30. https://doi.org/10.1016/j.neucom.2017.03.097

Challu, C., Olivares, K.G., Oreshkin, B.N., Ramirez, F.G., Mergenthaler-Canseco, M., & Dubrawski, A. (2023). NHITS: Neural Hierarchical Interpolation for Time Series Forecasting. *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI 2023*, *37.* https://doi.org/10.1609/aaai.v37i6.25854

Chatfield, C. (1978). The Holt-winters forecasting procedure. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *27*(3), 264–279. https://doi.org/10.2307/2347162

Chimmula, V.K.R., & Zhang, L. (2020). Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos, Solitons and Fractals*, *135,* 109864.
https://doi.org/10.1016/j.chaos.2020.109864

Deloof, M. (2003). Does working capital management affect profitability of Belgian firms? *Journal of Business Finance and Accounting*, *30*(3–4), 573-588.
https://doi.org/10.1111/1468-5957.00008

El Bourakadi, D., Ramadan, H., Yahyaouy, A., & Boumhidi, J. (2023). A robust energy management approach in two-steps ahead using deep learning BiLSTM prediction model and type-2 fuzzy decision-making controller. *Fuzzy Optimization and Decision Making*, 22, 645-667.
https://doi.org/10.1007/s10700-022-09406-y

Garcia, J.L., Martins, F.V., & Brandão, E. (2012). The Impact of Working Capital Management Upon Companies' Profitability: Evidence from European Companies. *SSRN Electronic Journal*.
https://doi.org/10.2139/ssrn.2165210

Gardner Jr, E.S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting*, *4*(1), 1–28.
https://doi.org/10.1002/for.3980040103

Hastie, T., & Tibshirani, R. (1987). Generalized additive models: some applications. *Journal of the American Statistical Association*, *82*(398), 371–386.
https://doi.org/10.2307/2289439

Hipel, K.W., McLeod, A.I., & Lennox, W.C. (1977). Advances in Box-Jenkins modeling: 1. Model construction. *Water Resources Research*, *13*(3), 567–575.
https://doi.org/10.1029/WR013i003p00567

Hong, T., & Wang, P. (2014). Fuzzy interaction regression for short term load forecasting. *Fuzzy Optimization and Decision Making*, *13*, 91–103.
https://doi.org/10.1007/s10700-013-9166-9

Janakiraman, G., & Roundy, R.O. (2004). Lost-sales problems with stochastic lead times: Convexity results for base-stock policies. *Operations Research, 52*(5), 795–803.
https://doi.org/10.1287/opre.1040.0130

Jeffery, M.M., Butler, R.J., & Malone, L.C. (2008). Determining a cost-effective customer service level. *Supply Chain Management: An International Journal*, *13*(3), 225–232.
https://doi.org/10.1108/13598540810871262

Koutsandreas, D., Spiliotis, E., Petropoulos, F., & Assimakopoulos, V. (2022). On the selection of forecasting accuracy measures. *Journal of the Operational Research Society*, *73*(5), 937–954.
https://doi.org/10.1080/01605682.2021.1892464

Lazaridis, I., & Tryfonidis, D. (2006). The relationship between working capital management and profitability of listed companies in the Athens Stock Exchange. *Journal of Financial Management and Analysis*, *30*(76), 1-12. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=931591

Lim, B., Ark, S.Ö., 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

Liu, A., Lu, J., & Zhang, G. (2020). Concept drift detection: dealing with missing values via fuzzy distance estimations. *IEEE Transactions on Fuzzy Systems*, *29*(11), 3219–3233.
https://doi.org/10.1109/TFUZZ.2020.3016040

Nguyen, H.D., Tran, K.P., Thomassey, S., & Hamad, M. (2021). Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. *International Journal of Information Management*, *57,* 102282.
https://doi.org/10.1016/j.ijinfomgt.2020.102282

Nobanee, H. (2017). Working Capital Management of Small Firms. *SSRN Electronic Journal*.
https://doi.org/10.2139/ssrn.2970031

Oreshkin, B.N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *8th International Conference on Learning Representations, ICLR 2020*. https://openreview.net/attachment?id=r1ecqn4YwB&name=original_pdf

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

Shah, G., Fraser, J., Mandhana, V., & Verma, V. (2022). *Working Capital Index Report 2022*. https://www.jpmorgan.com/content/dam/jpm/treasury-services/documents/working-capital-report-2022.pdf

Taylor, S.J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, *72*(1), 37–45.
https://doi.org/10.1080/00031305.2017.1380080

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*, *30*.
https://doi.org/10.48550/arXiv.1706.03762

Windaus, D., & Tebbett, S. (2021). *Working Capital Report 2019/20: Creating value through working capital*. https://www.pwc.com/gx/en/working-capital-management-services/assets/working-capital-report-final.pdf

Wu, B., Wang, L., & Zeng, Y.R. (2023). Interpretable tourism demand forecasting with temporal fusion transformers amid COVID-19. *Applied Intelligence*, *53*(11), 14493–14514.
https://doi.org/10.1007/s10489-022-04254-0