Erasmus University Rotterdam

Erasmus School of Economics

Proposal Econometrics and Management Science - BA & QM

Uncertainty Estimation Temporal Fusion Transformer - An online supermarket case study

Florien Muntinga (662395)

Supervisor:	Professor K. Gruber
Second assessor:	Professor F. Frasincar
Company supervisor:	Henk Peterse
Date final version:	March 5, 2024

The content of this thesis is the sole responsibility of the author and does not reflect the view of the supervisor, second assessor, Erasmus School of Economics or Erasmus University.

Abstract

This master's thesis explores the enhancement of demand forecasting within the context of time-series analysis, specifically focusing on integrating uncertainty estimation into the Temporal Fusion Transformer (TFT) model through the implementation of Monte Carlo (MC) dropout. Employing this novel approach, the thesis aims to adapt the TFT architecture to produce probability density functions alongside its predictions, thereby offering a more nuanced understanding of forecast uncertainty. The study conducts a detailed case study using a dataset from Picnic, an innovative app-based online grocery store. Empirical results from the study indicate a promising direction for incorporating Monte Carlo (MC) dropout into the forecasting process. The inclusion of MC dropout has shown a modest yet significant improvement in forecast precision, evidenced by a 1.50% reduction in the Weighted Absolute Percentage Error (WAPE) and a notable enhancement in the Weighted Percentage Error (WPE) by 36.25%. These improvements highlight the value of integrating uncertainty into demand forecasting models. Moreover, the thesis presents predictive distributions across various forecasting scenarios, demonstrating the MC dropout-enhanced TFT model's ability to produce distributions that reflect its overall performance accurately. This approach not only advances beyond point predictions but also provides a comprehensive evaluation of the forecast's reliability and the inherent uncertainty of each outcome. By generating and examining these distributions, the study offers insights into the model's robustness and a more refined understanding of the variability in forecasted outcomes. This master's thesis lays the groundwork for a significant shift in demand forecasting methodologies by integrating uncertainty estimation into the forecasting process.

Keywords: Uncertainty quantification; demand forecasting; probability density forecasting
JEL classification: C53, C55, D81, L81

Contents

1 Introduction

Time-series forecasting has been a key area of academic research and application in domains such as economics and finance [Cao et al., 2019, Nunnari and Nunnari, 2017], as well as meteorology [Karevan and Suykens, 2020], medicine [Bui et al., 2018], energy [Alvarez et al., 2010], and supply chain management [Mircetic et al., 2022]. Time-series forecasting involves the use of historical data to forecast future events or trends. In recent decades, a variety of approaches have been introduced by researchers to tackle the challenge of time-series forecasting: achieving accurate predictions. These approaches broadly fall into three categories: traditional statistical models, machine learning approaches, and deep learning.

An example of the first category is Autoregressive Moving Average Model (ARIMA) [Junior et al., 2014]. Many statistical techniques, such as ARIMA, construct time series models based on historical data, primarily capturing linear characteristics. These methods excel at short-term or one-step-ahead predictions. In practical applications of time series forecasting, traditional machine learning methods such as Support Vector Machine [Pai et al., 2010], Random Forest [Dudek, 2015], and XGBoost [Lv et al., 2021] are commonly used. These methods have proven their effectiveness in a range of prediction tasks. However, they have limitations in capturing temporal relationships within data. This limitation arises from the assumption that data points in the time dimension are equally relevant at every time step, which restricts the extraction of meaningful temporal insights. Deep learning incorporates multiple layers to systematically extract increasingly sophisticated features from raw input data. Deep learning has attracted tremendous attention from researchers because its capacity to effectively handle data in the time dimension, an aspect challenging for standard machine learning algorithms [Feng et al., 2022]. Within the extensive range of deep learning methods designed for complex data analysis, notable examples include the Long Short-Term Memory Neural Network (LSTM) [Sagheer and Kotb, 2019] and the Transformer [Vaswani et al., 2017]. The Transformer model has been shown to significantly outperform the LSTM in capturing extended temporal dependencies, a key advantage in many analytical tasks [Feng et al., 2022].

The Transformer model, introduced by Vaswani et al. [2017], quickly became popular across various domains for its effective handling of sequential data. At its core, the Transformer model consists of an encoder to process historical data and a decoder for predicting future values, employing an autoregressive approach. This design allows the model to pay selective attention to historical data segments crucial for accurate future predictions, and thus, enhances its predictive capabilities. In the realm of time series forecasting, the Transformer model has proven especially beneficial for its adeptness at capturing extended temporal relationships.

One notable advancement in this area is the Temporal Fusion Transformer (TFT) [Lim et al., 2021], which has garnered attention for its specialized approach to time series data. Unlike the previous mentioned models, TFT is designed to adeptly manage diverse input types—static, known future, and observed inputs—thereby optimizing prediction performance across a range of scenarios. This adaptability addresses previous models' limitations, such as their handling of exogenous inputs or the assumption that all such inputs are known in advance. Furthermore, the TFT distinguishes itself through its inherent interpretability, a contrast to the "black-box" nature of conventional deep learning models. By incorporating well-defined components that construct detailed feature representations for different types of inputs, the TFT not only demonstrates superior forecasting accuracy but also provides insights into the underlying decision-making process, enhancing its utility for multi-horizon time series forecasting. By integrating the Transformer's strengths with the specific requirements of time series forecasting, the TFT model represents a significant step forward.

The architecture of TFT comprises five key elements: gating mechanisms, variable selection networks, static covariate encoders, temporal processing modules, and prediction intervals. Each of these components plays a crucial role in optimizing the model's forecasting ability, allowing for a nuanced understanding of time series data and contributing to the model's overall interpretability and effectiveness in predicting future events. The practical utility of the Temporal Fusion Transformer (TFT) model is well-documented, having been confirmed through various studies [Zhang et al., 2022, Huy et al., 2022, Wu et al., 2022]. The TFT model will serve as the foundational model for analysis in this master's thesis.

Uncertainty impacts the predictive capabilities of Deep Neural Networks (DNNs), hence also of TFT model. Since numerous factors contribute to uncertainty, elimination of the complete uncertainty within the predictions is impossible, particularly when handling real-world data. In practice, training data typically represents a subset of all input data, leading to domain misalignment between the DNN and the actual data domain. Nevertheless, achieving precise representation of DNN prediction uncertainty remains difficult due to the inherent complexity of accurately modeling multiple uncertainties, many of which are unidentified.

Within this context, two types of uncertainty have been identified: aleatoric and epistemic. Aleatoric uncertainty, also known as data uncertainty, emerges from the natural randomness or variability inherent to the phenomenon being observed. Conversely, epistemic uncertainty, or model uncertainty, is rooted in incomplete knowledge or the absence of sufficient data. The research focus on developing methods for estimating uncertainty in DNN predictions has gained significant importance and attention.

To address uncertainty of deep learning context, there has been a significant shift in the literature from point forecasts to probabilistic forecasting [Huy et al., 2022]. Point forecasts are deterministic in nature, so are the quantiles of the TFT model, and do not capture the underlying uncertainty. Probabilistic forecasting can be categorized based on its approach to quantifying the uncertainty of predictions. Where point forecasts only provide the values of the predicted points, interval predictions predict a certain point and its confidence level with a certain probability. This thesis focuses on another type of probabilistic forecasting: probability density forecasting. This methodology crafts a probability density function that encapsulates forecasting outcomes and provides substantial insights, for example, by enabling the assessment of risk and uncertainty in future events more precisely. Unlike conventional interval-based forecasts that offer production intervals (PIs) at specific confidence levels, probability density forecasting directly outputs continuous probability density functions, surpassing the fragmented nature of PIs. In the domain of probabilistic forecasting, several methodologies have been recognized for their ability to quantify the uncertainty of neural network predictions, including the delta method [Hwang and Ding, 1997, De Vleaux et al., 1998], Bayesian methods [Yang et al., 2013], mean variance estimation [Khosravi et al., 2012], the bootstrap method [Wan et al., 2013], and Gaussian processes [Yang et al., 2018]. This thesis, however, will concentrate on Bayesian Neural Networks (BNNs), leveraging their capacity to incorporate uncertainty directly into the model architecture, thus providing a sophisticated framework for probability density forecasting.

BNNs incorporate uncertainty into deep learning models by applying Bayesian principles. Instead of seeking a single point estimate for the network's parameters, BNNs aim to determine the posterior distribution of these parameters by giving a prior to the network's parameters, based on the initial assignment of prior distributions. This approach is called posterior inference in classical Bayesian models. However, the inherent complexity and the absence of compatibility in deep learning models often hinder exact posterior inference. Furthermore, the large amount of parameters in modern neural networks challenges traditional approximate Bayesian inference techniques due to scalability issues. In response to these challenges, recent advancements have introduced several methods for approximate inference in BNNs, offering scalable solutions to approximate the posterior distribution of the network's parameters in these complex models (see Gal et al. [2016] for a comprehensive review on approximate inference techniques). Among these methods, a significant subset employs variational inference to optimize the variational lower bound. This subset includes approaches such as variational Bayes [Kingma and Welling, 2013], probabilistic backpropagation [Hernández-Lobato and Adams, 2015], and 'Bayes by Backprop' [Blundell et al., 2015]. Furthermore, there are algorithms that expand upon this framework through the optimization of α-divergence, examples of which can be found in Hernandez-Lobato et al. [2016] and Li and Gal [2017].

The previously mentioned algorithms require different training strategies for neural networks, involving specific modifications to the loss function to address different optimization challenges and complex alterations to the training methodology. Moreover, many of the existing inference algorithms introduce additional parameters, sometimes nearly doubling the parameter count, which poses scalability issues due to the already large parameter sets typical in neural network models. Monte Carlo Dropout (MC dropout) [Gal and Ghahramani, 2016] emerges as a compelling and practical approach, recognized for its simplicity in implementation and effectiveness. This method considers dropout layers as Bernoulli-distributed random variables, framing training with dropout layers as approximating variational inference [Gal and Ghahramani, 2016, Gal et al., 2017]. MC dropout enables the computation of predictive uncertainty by implementing dropout not only during training but also at test time. Meaning that implementation efforts are minimal once the model is trained with dropout layers. It is important to note that MC dropout primarily addresses model uncertainty, enabling a more robust representation of the uncertainty associated with the model's parameters. The practical significance of MC dropout is evident, as it has been validated across multiple works [Eaton-Rosen et al., 2018, Loquercio et al., 2020, Rußwurm et al., 2020], including those with large-scale data.

Practically, these variational Bayesian methods often find themselves outperformed by the impressive Deep Ensembles, as demonstrated in Lakshminarayanan et al. [2017]'s study. Deep

Ensembles, a straightforward and non-Bayesian approach, involve the training of multiple deep models with varying initializations and distinct data set arrangements. Ovadia et al. [2019] supported this point, illustrating the consistent superiority of Deep Ensembles over Bayesian neural networks trained using variational inference. However, this elevated performance comes at a computational cost. Deep Ensembles, both during training and testing, exhibit linear scaling in terms of memory and compute utilization as the number of ensemble elements increases. Given the practical constraints of computational power and memory capacity, MC dropout is the method employed in this master's thesis in order to obtain probabilistic forecasts.

To summarize, this master thesis is fundamentally focused on uncertainty estimation within the context of time-series analysis, particularly employing a Temporal Fusion Transformer (TFT) model. In critical decision-making scenarios, understanding uncertainty is important. A TFT model provides outputs in the form of quantiles, lacking the capability for probability density forecasting. Consequently, this master's thesis aims to modify the TFT architecture, enabling it to produce probability density functions around its predictions. This modification not only aims to provide a more nuanced and comprehensive view of uncertainty but also suggests that by averaging over a range of probabilistic forecasts, it may improve overall prediction accuracy. This modification thus seeks to enhance the TFT model, offering a more nuanced and comprehensive understanding of the associated uncertainty. By providing probability density forecasts, the master thesis aims to empower decision-makers with more insightful information for informed and robust decision-making processes.

In this master thesis, a case study will be conducted. The dataset employed originates from Picnic, an app-based online grocery store determined to revolutionize the online supermarket industry. For retail companies article demand prediction is the core of supply chain decision making. Article demand prediction fundamentally relies on time-series forecasting. As mentioned before, estimating uncertainty plays an important role in facilitating safe decision-making, especially in the retail context where fluctuations in demand can significantly impact profitability and operational efficiency. Enhanced forecast accuracy and a detailed understanding of uncertainty can lead to more informed stock management decisions, minimizing the risk of overstocking or stockouts, and thereby ensuring that customer demand is met efficiently without excess expenditure on unused inventory. If successful, the probability density forecasts generated by this improved model will provide critical insights into forecast uncertainty, substantially advancing the safety and reliability of decision-making processes within the retail industry. This advancement could be a significant step forward in achieving the delicate balance between supply and demand, ultimately contributing to the sustainability and profitability of retail operations.

The *baseline* of this master thesis is the TFT model with its general outputs, the quantiles. To enhance the precision of uncertainty, MC dropout will be incorporated, hereafter called *MC dropout* model, aiming to provide density function forecasts. This master thesis seeks to investigate whether the integration of MC dropout into the TFT model can improve article demand forecast accuracy and the precision of its uncertainty within the retail sector. To the best of our knowledge, this is the first attempt at incorporating MC dropout with a TFT model to enhance demand predictions in the retail sector. Such an approach promises to bridge a crucial gap in the literature and offer new perspectives in predictive analytics.

The contributions of this research are as follows:

- An improved Temporal Fusion Transformer approach for article demand forecasting that enables the TFT model to construct a probability density function, so that it can be used in safe decision-making.
- An extensive evaluation on a large real-world data set to demonstrate the effectiveness and applicability of the proposed method.

The integration of MC dropouts into our forecasting model has led to significant improvements in two key areas of article demand forecasting. Firstly, the adoption of MC dropout has sharpened prediction accuracy, as demonstrated by a 1.50% reduction in Weighted Absolute Percentage Error (WAPE) and a 36.25% improvement in Weighted Percentage Error (WPE). These results highlight an improvement in the precision of our forecasts. Secondly, the ability of the model to encapsulate uncertainty has been substantially enhanced, with the MC dropout model consistently producing probability density functions that are well-aligned with observed outcomes. This confirms the effectiveness of the MC dropout method in generating more reliable forecasts. Overall, these findings affirm that the incorporation of MC dropouts into the forecasting process not only improves accuracy but also provides a deeper and more actionable understanding of uncertainty, offering considerable value to the field.

This research is structured as follows. At first, Section 2, details the dataset used and identifies the primary variable of interest. Thereafter, Section 3, delves into the research methods employed, particularly highlighting the application of Monte Carlo dropout to the Temporal Fusion Transformer model. Section 4 presents the empirical findings derived from these methods. After this, we proceed to Section 5, where the key conclusions of the study are drawn. Lastly, Section 6 entails the discussion, which offers a reflective analysis on the implications and future directions of the research.

2 Data

This section starts with a brief introduction to the company from which the data is obtained. Thereafter the variable of interest will be explained in detail. Next, the features used in our model are briefly discussed. Lastly, the filters from which the final dataset is obtained is discussed.

2.1 Introduction to Picnic Technologies B.V.

The dataset used in this master's thesis originates from the Data WareHouse (DWH) of Picnic Technologies B.V, hereafter referred to as Picnic. As an e-commerce grocery retailer, Picnic manages a product inventory of approximately 10000 stock keeping units (SKU) categorized across different hierarchical levels. For instance, products like "Konings Magere Franse kwark 1kg" are categorized into levels such as 'Diary & Eggs' (1st level), 'Quark' (2nd level), 'Low-fat' (3rd level), and 'Quark naturel low-fat' (4th level). Picnic's supply chain contains distribution centers (DCs), fulfilment centers (FCs), where customer orders are assembled and hubs, from where the orders are distributed to the customers. Operating across the Netherlands, Germany, and France, Picnic guarantees next-day delivery if orders are placed before 22:00h, referred to as the cutoff time. Given that customers can place orders until 22:00h and receive them the next day, Picnic must accurately forecast demand to ensure optimal inventory management and timely supplier orders. Picnic's demand is forecasted using Article Delivery Rates (ADRs), the variable of interest in this master's thesis.

2.2 Article demand rate

ADRs are computed by dividing the number of customer units sold for a specific article by the number of deliveries, providing an average count of units sold per delivery. These rates are predicted per fulfillment center *F*, slot picking group *SPG*, article *I*, and delivery date *D*, constituting the granularity of the dataset used in our analysis. The SPG represents the time slot during which groceries are delivered, categorized as either 'morning' or 'evening'. Each article corresponds to a product identified by a unique Stock Keeping Unit (SKU), such as 'Coca-Cola Regular 4 x 1.5L'.

Handling unavailability is crucial for predicting ADRs accurately. Unavailability occurs when articles are out of stock, preventing customers from adding them to their baskets. Consequently, the reported ADR in such instances does not reflect the actual demand. In reality, the ADR for these articles would have been higher if customers could have placed orders for them. Therefore, we use *'clean'* ADR values, representing the ADR if stock were unlimited, to mitigate the impact of unavailability on forecasting accuracy. To estimate the lost sales due to unavailability, we assess the quantity of units customers would have purchased if the article had been available, in other words the potential customer demand for a specific article. Picnic logs instances where customers attempt to add a product to their basket but encounter an unavailability message. However, we cannot simply count these messages due to two reasons. Firstly, some customers who initially encounter an unavailability message may ultimately purchase the product, as it may become available again before their next delivery. Secondly, some customers may intend to purchase multiple units of a product and receive the unavailability message only once. To address the former scenario, we define unavailability as the count of customers who view an unavailability message *and* do not purchase the article. For the latter case, we multiply the number of unavailability events by the *commonality*, which reflects the average units per delivery for a product. To conclude, clean ADR is computed as follows:

$$\text{clean } ADR_{D,F,S,I} = \frac{\sum_{d,f,s,i}(\text{units\_sold}_{d,f,s,i} + \text{unavailability}_{d,f,s,i} \times \text{commonality})}{\sum_{d,f,s} \text{deliveries}_{d,f,s}},$$
$$d \in D, f \in F, s \in S, i \in I.$$

2.3 Features

The model incorporates a combination of static and time-varying features, listed in Table 1, with the time-varying features computed up to 28 days prior to the prediction date.*Confirmed demand* refers to the known demand resulting from orders already placed at the time of prediction, while

Confirmed deliveries indicates the number of deliveries for a FC and SPG corresponding to the confirmed demand. Additionally, as elaborated in Section 2.2, *ADR* represents the clean article delivery rate.

Table 1: Overview of the TFT features

Feature name	**Definition**	**Type**
FC ID	Fulfilment centre ID	Static
Slot picking group	Part of day for customer delivery	Static
Article category level 1	High-level category	Static
Article category level 2	More granular category	Static
Weather sensitive level 3	Indicates weather sensitivity	Static
Confirmed ADR	Confirmed demand, strong predictor	Time-varying
Confirmed deliveries	Portion of demand confirmed	Time-varying
Confirmed rate	Confirmed deliveries / last week	Time-varying
Minutes until slot cutoff	Minutes until order deadline	Time-varying
avg_wind_speed	Impact of weather on behavior	Time-varying
Max temperature	Predicted maximum temperature	Time-varying
Min temperature	Predicted minimum temperature	Time-varying
Average cloud cover	Impact of weather on behavior	Time-varying
Vacation period	Different vacation periods for FCs	Time-varying
Special days	Christmas, Saint Nicholas, Easter	Time-varying
Weekday	Capture weekly patterns	Time-varying
Month	Capture yearly patterns	Time-varying
Promo mechanism	Types of promo mechanisms	Time-varying
promo is superdeal	If promo is a superdeal	Time-varying
promo adr confirmed	Confirmed demand by promotions	Time-varying
discount percentage	Influence of discount on demand	Time-varying
number of recipes	Number of recipes the article has	Time-varying
number of articles in promotion	Descriptive for promo success	Time-varying
number of art p cat lev 2 in promotion	Capture cannibalism of promotions	Time-varying
day of month	Capture patterns like payday	Time-varying
calendar week	Capture yearly cycles of behavior	Time-varying
ADR	Article Delivery Rate	Time-varying

2.4 Filters

The dataset provided by Picnic is extensive, requiring the application of various filters to reduce its size for the purposes of this master's thesis. Initially, we opted to focus solely on one fulfillment center located in the Netherlands. This approach allowed us to incorporate a diverse range of articles to enhance the representativeness of our model. However, even with this filter in place, the dataset remained excessively large. Consequently, we further refined the dataset by only considering articles with an even article ID. This random selection approach ensured that our dataset remained representative. After filtering the data, we ensured that only informative records are retained by discarding observations with null values for any of the variables listed in Table 1. Subsequently, we conducted sanity checks to verify that ADR values fell within the acceptable range, ensuring they were neither negative nor greater than one.

3 Methodology

The methodology section focuses on integrating Monte Carlo (MC) dropouts into the Temporal Fusion Transformer (TFT) model architecture, aiming to enhance model performance and provide uncertainty estimates. However, before delving into this specific method, it is important to establish a foundational understanding of the TFT model. This involves the fundamental concepts and exploring how the architecture incorporates uncertainty quantification. Consequently, Section 3.1 provides a brief overview of the general architecture of a TFT model. Subsequently, Section 3.2 discusses the principles of *standard dropouts*. Following this, Section 3.3 delves into *MC dropouts*. In Section 3.4, the derivation of a predictive distribution is explained. Finally, the practical implementation is outlined in Section 3.5.

3.1 General architecture TFT model

At the core of our method lies a TFT model, as described by Lim et al. [2021]. Using well-established components, TFT constructs feature representations for different input types, including static, known future, and observed inputs. This ensures high prediction performance across various forecasting scenarios. TFT includes five major constituents, namely, gating mechanisms, variable selection networks, static covariate encoders, temporal processing, and prediction intervals. Figure 1 shows an overview of the architecture of a TFT model.

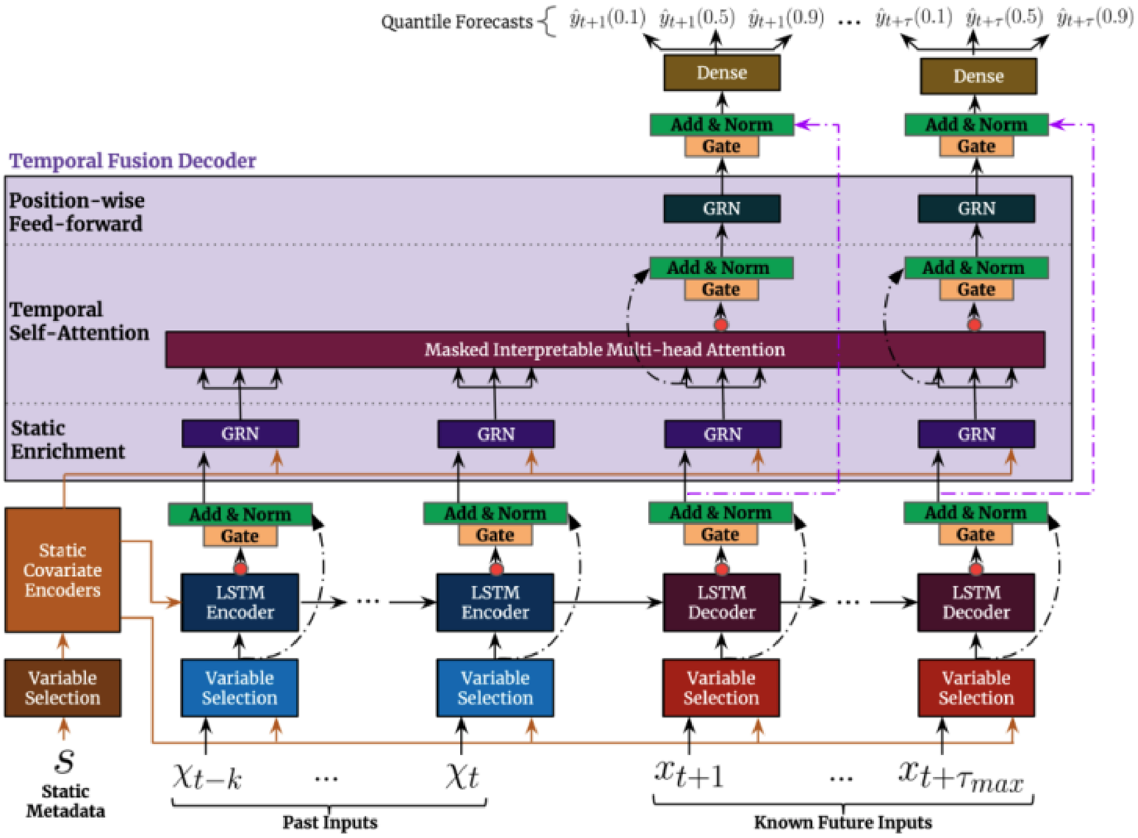


Figure 1: Architecture TFT model as proposed in Lim et al. [2021]

Important for this research is that TFT not only generates a single point prediction, but also provides a prediction interval around this point to account for uncertainty. The general TFT architecture with quantiles as uncertainty quantification serves as the baseline for the research, hereafter referred to as baseline TFT model. This interval captures the model's expectation of the possible variation in the actual outcome. TFT generates quantiles by simultaneously predicting various percentiles at each time step, using a linear transformation output by the temporal fusion decoder. The training involves jointly minimizing quantile loss, and the resulting quantile outputs are summed using the given formula:

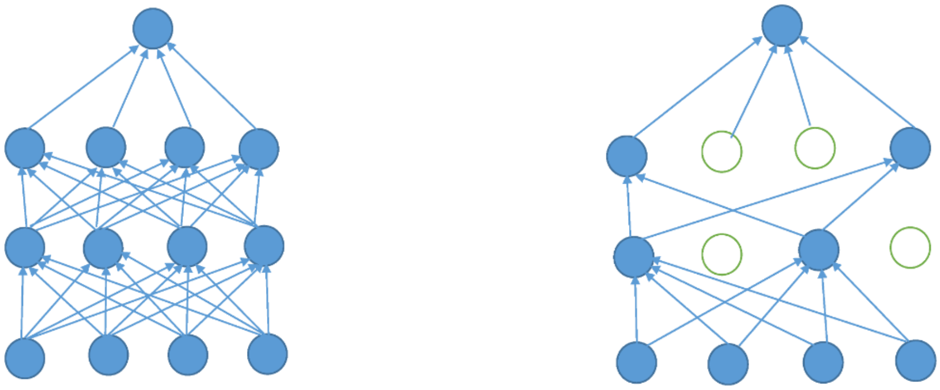
$$L(\Omega, \mathrm{W}) = \sum_{y_t \in \Omega} \sum_{q \in \varrho} \sum_{T=1}^{T_{\max}} \frac{QL\left(y_t, \widehat{y}(q, t-T, T), q\right)}{M T_{\max}}$$

$$QL\left(y, \widehat{y}, q\right) = q\left(y - \widehat{y}\right)_+ + (1-q)\left(\widehat{y} - y\right)_+ ,$$

where Ω refers to the domain of the training data containing, comprising M samples, ϱ represents the set of output quantiles, W corresponds to the weights of TFT, and $(.)_+$ denotes max(0,.). The specific characteristics of our quantile loss function will be clarified in Section 3.5.3.

3.2 Standard dropouts

With the significant evolution and expansion of deep learning architectures in recent years, new challenges have emerged. The adoption of larger and more complex deep learning architectures has heightened the risk of overfitting during training. To address these challenges, researchers have introduced various regularization techniques, among which dropout stands out prominently. Initially proposed by Hinton et al. [2012], dropout, in thesis referred to as *standard dropout*, offers a solution to mitigate overfitting by randomly deactivating neurons in both input and hidden layers within the neural network during training. During this process, all connections, both forward and backward, associated with the excluded neuron are temporarily eliminated, resulting in the derivation of a modified network architecture from the original network. The nodes undergo exclusion based on a dropout probability denoted as p_i for the i-th layer. Below, we provide an illustration of dropout as a regularization technique is provided by Figure 2. The principles of *standard dropout* are fundamental to those of *MC dropouts*.



(a) Without dropout

(b) With dropout

Figure 2: Illustration of dropout in a standard neural network by Salehin and Kang [2023]

3.3 Monte Carlo dropouts

MC dropouts can be used to approximates the principles of a Bayesian Neural Networks (BNNs) [Gal and Ghahramani, 2016]. BNNs introduce uncertainty to deep learning models from a Bayesian perspective. Before explaining MC dropouts it is thus important to understand the fundamentals of BNNs.

3.3.1 Bayesian Neural Network

Let a neural network be denoted as $f_\omega(\cdot)$, with f containing the network architecture, and ω representing the model parameters. BNNs introduce a prior for the weight parameters, aiming to fit the optimal posterior distribution. A commonly employed prior is the Gaussian prior:

$$\omega \sim N(0, I).$$

The probability distribution that characterizes the data generation is defined as $p(y|f_\omega(x))$. In the context of regression, for a certain noise level σ, this probability distribution is frequently assumed to be:

$$y|\omega \sim N\left(f^\omega(x), \sigma^2\right).$$

Bayesian inference, involving finding the posterior distribution over model parameters $p(\omega|X, Y)$ given a set of N observations $X = x_1, ..., x_N$ and $Y = y_1, ..., y_N$, is important for estimating model uncertainty. Let x denote a new data point. The prediction distribution is obtained by marginalizing out the posterior distribution:

$$p(y^\star|x^\star) = \int_\omega p(y^\star|f^\omega(x^\star))p(\omega|X, Y)d\omega.$$

However, Bayesian inference is hard in deep learning models. In this paper, the idea of Gal and Ghahramani [2016] is used which states that MC dropouts can be used as an approach to approximate model uncertainty.

3.3.2 MC dropouts as Bayesian approximate

The principles of Gal and Ghahramani [2016] will be summarized here before showing how it is used to obtain uncertainty estimation, for the exact proofs we refer to the appendix of the paper itself. Gal and Ghahramani [2016] show that the dropout objective, in effect, minimises the Kullback-Liebler divergence between an approximate distribution and the posterior of a deep Gaussian process.

MC dropout leverages dropout as a regularization technique, explained in Section 3.2 to estimate prediction uncertainty. The dropout objective function, designed for minimization and incorporating L_2 regularization, can be expressed as follows:

$$\mathcal{L}_{\text{dropout }}:=\frac{1}{N} \sum_{i=1}^{N} E\left(\mathbf{y}_{i}, \widehat{\mathbf{y}}_{i}\right)+\lambda \sum_{i=1}^{L}\left(\left\|\mathbf{W}_{i}\right\|_{2}^{2}+\left\|\mathbf{b}_{i}\right\|_{2}^{2}\right) .$$

Where, the output of a neural network and the actual is denoted by $\hat{y}$ and y respectively, and L is the number of layers of the neural network. $E(.,.)$ denotes a loss function, in our case the quantile loss. W_i denotes the weights matrices, and b_i denotes the bias vectors.

While non-probabilistic NNs lack the capability to model distributions over functions, the deep Gaussian process (GP) excels in this regard. In line with Gal and Ghahramani [2016], the predictive probability of the deep GP model integrated w.r.t the finite rank covariance function parameters ω given some precision parameter $\tau > 0$ can be expresses as follows:

$$\begin{aligned}
p(\mathbf{y} \mid \mathbf{x}, \mathbf{X}, \mathbf{Y}) &= \int p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega} \mid \mathbf{X}, \mathbf{Y}) \mathrm{d} \boldsymbol{\omega} \\
p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\omega}) &= \mathcal{N}\left(\mathbf{y} ; \widehat{\mathbf{y}}(\mathbf{x}, \boldsymbol{\omega}), \tau^{-1} \mathbf{I}_{D}\right) \\
\widehat{\mathbf{y}}\left(\mathbf{x}, \boldsymbol{\omega}=\left\{\mathbf{W}_{1}, \ldots, \mathbf{W}_{L}\right\}\right) &= \sqrt{\frac{1}{K_{L}}} \mathbf{W}_{L} \sigma\left(\cdots \sqrt{\frac{1}{K_{1}}} \mathbf{W}_{2} \sigma\left(\mathbf{W}_{1} \mathbf{x}+\mathbf{m}_{1}\right) \ldots\right),
\end{aligned}$$

where $\sigma(.)$ represents some element-wise non-linearity, W_i a stochastic matrix with dimensions $K_i \times K_{i-1}$ for each layer i, and $\omega = \{W\}_{i=1}^{L}$. Moreover, vectors m_i with dimensions K_i for each layer in the Gaussian process are considered.

The posterior distribution $p(\omega|X, Y)$ is challenging to compute directly. Hence, $q(\omega)$, a distribution over matrices where columns are stochastically zeroed out, is employed to provide and approximation for the posterior. Specifically, Gal and Ghahramani [2016] define $q(\omega)$ as:

$$\begin{aligned}
&\mathbf{W}_{i}=\mathbf{M}_{i} \cdot \operatorname{diag}\left(\left[\mathbf{z}_{i, j}\right]_{j=1}^{K_{i}}\right) \\
&\mathbf{z}_{i, j} \sim \operatorname{Bernoulli}\left(p_{i}\right) \text { for } i=1, \ldots, L, j=1, \ldots, K_{i-1},
\end{aligned}$$

given certain probabilities p_i and matrices M_i as variational parameters. The binary variable $z_{i,j} = 0$ signifies that unit j in layer $i - 1$ is omitted as an input to layer i.

Gal and Ghahramani [2016] minimize the Kullback-Leibler (KL) divergence between the approximate posterior $q(\omega)$ and the true posterior of the complete deep Gaussian process, $p(\omega|X, Y)$. The KL divergence is the primary minimization objective, formulated as:

$$-\int q(\boldsymbol{\omega}) \log p(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\omega}) \mathrm{d} \boldsymbol{\omega}+\mathrm{KL}(q(\boldsymbol{\omega}) \| p(\boldsymbol{\omega})).$$

The first term of this objective can be further expressed as a sum:

$$-\sum_{n=1}^{N} \int q(\boldsymbol{\omega}) \log p\left(\mathbf{y}_{n} \mid \mathbf{x}_{n}, \boldsymbol{\omega}\right) \mathrm{d} \boldsymbol{\omega}.$$

Each term in the sum is approximated using Monte Carlo integration with a single sample $\omega \sim q(\omega)$, providing an unbiased estimate $\log p\left(\mathbf{y}_n \mid \mathbf{x}_n, \boldsymbol{\omega}\right) \mathrm{d}\boldsymbol{\omega}$. The second term is approximated as $\sum_{i=1}^{L}\left(\frac{p_i l^2}{2}\left\|\mathbf{M}_i\right\|_2^2+\frac{l^2}{2}\left\|\mathbf{m}_i\right\|_2^2\right)$, where l denotes prior length. Given model precision τ, the result is scaled by $\frac{1}{\tau N}$ to obtain the objective:

$$\mathcal{L}_{\text{GP-MC}} \propto \frac{1}{N} \sum_{n=1}^{N} \frac{-\log p\left(\mathbf{y}_n \mid \mathbf{x}_n, \widehat{\boldsymbol{\omega}}_n\right)}{\tau}+\sum_{i=1}^{L}\left(\frac{p_i l^2}{2 \tau N}\left\|\mathbf{M}_i\right\|_2^2+\frac{l^2}{2 \tau N}\left\|\mathbf{m}_i\right\|_2^2\right).$$

By defining $E\left(\mathbf{y}_n, \widehat{\mathbf{y}}\left(\mathbf{x}_n, \widehat{\boldsymbol{\omega}}_n\right)\right)=-\log p\left(\mathbf{y}_n \mid \mathbf{x}_n, \widehat{\boldsymbol{\omega}}_n\right) / \tau$, the objective mirrors the dropout objective for an appropriate choice of the precision hyper-parameter τ and length scale l.

In essence, this realization demonstrates that dropout can be viewed as a Bayesian approximation, and consequently, uncertainty estimates for dropout neural networks can be derived.

3.3.3 Obtaining model uncertainty estimates

The approximate predictive distribution proposed by Gal and Ghahramani [2016] is expressed as:

$$q\left(\mathbf{y}^* \mid \mathbf{x}^*\right)=\int p\left(\mathbf{y}^* \mid \mathbf{x}^*, \boldsymbol{\omega}\right) q(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega}.$$

Here, $\omega=\{W_i\}_{i=1}^{L}$ represent the set set of random variables in a model with L layers.

The predictive mean can be estimated by:

$$\mathbb{E}_{q\left(\mathbf{y}^* \mid \mathbf{x}^*\right)}\left(\mathbf{y}^*\right) \approx \frac{1}{T} \sum_{t=1}^{T} \widehat{\mathbf{y}}^*\left(\mathbf{x}^*, \mathbf{W}_1^t, \ldots, \mathbf{W}_L^t\right),$$

and it is referred to as MC dropout by Gal and Ghahramani [2016]. Practically, this is equivalent to performing T stochastic forward passes through the network and averaging the results, which in turn is equivalent to the concept of standard dropout during test time.

To summarize, the algorithm of Gal and Ghahramani [2016] involves computing the neural network output with stochastic dropouts at each hidden layer repeated T times to yield $\hat{y}^{\star}(1), \ldots, y^{\star}(T)$, which can be used to approximate model uncertainty. The mathematical representation of this procedure can be summarized as follows:

$$q\left(\mathbf{y}^* \mid \mathbf{x}^*\right)=\int p\left(\mathbf{y}^* \mid \mathbf{x}^*, \boldsymbol{\omega}\right) q(\boldsymbol{\omega}) \mathrm{d}\boldsymbol{\omega} \approx \frac{1}{T} \sum_{t} p\left(\mathbf{y}^* \mid \mathbf{x}^*, \boldsymbol{\omega}_{\boldsymbol{t}}\right), \quad \boldsymbol{\omega}_{\boldsymbol{t}} \sim q(\omega).$$

3.4 Predictive distribution

In the preceding section, we introduced our quantile forecasting approach within the TFT with MC dropout framework, which iteratively generates predictions to account for model uncertainty. This method computes quantiles repeatedly for a single ADR prediction concerning a specific article, FC, SPG, and prediction date. By doing so, it yields an array of potential outcomes which, when averaged, form a robust ADR estimate.

This iterative quantile prediction process allows us to construct a comprehensive predictive distribution for each ADR prediction. We combine the multiple quantile predictions associated with a single FC, SPG, prediction date, and article instance to create an empirical distribution. This aggregated collection of quantiles offers a discretized approximation of what would be a continuous distribution in the theoretical setting. The resulting empirical distribution captures a spectrum of possible outcomes and provides deeper insight into the predictive capabilities and uncertainty inherent in the model.

To visualize this distribution, we employ a histogram where the quantiles are categorized into bins, with the frequency of occurrences dictating the height of each bin. This histogram reveals the concentration and variability of the predicted values, granting immediate perception of the most probable outcomes as well as the range and variability of predictions.

We enhance the histogram with a Kernel Density Estimate (KDE), which overlays a smooth curve to approximate the probability density function of the underlying distribution. The KDE smooths the discrete nature of the histogram, providing a visual representation of the distribution's shape, central tendency, and dispersion. It allows us to discern the modes, assess the skewness, and identify other distributional properties that characterize the predictive performance of the model.

3.5 Implementation

After providing a concise overview of the TFT model components and clarifying the methodology of MC dropouts, this section delves into the practical implementation. The implementation is carried out using Python 3.10, alongside the PyTorch library[1]. The initiates with a description of the essential data pre-processing steps. Subsequently, it outlines the hyperparameters used and the procedures employed for their tuning. Next, it elaborates on the model training process to attain the desired output. Lastly, it addresses the evaluation of the model's output.

3.5.1 Pre-processing

To obtain unbiased estimates, the dataset, as explained in Section 2, is divided into three distinct sets: a training, validation, and test set. Each of these sets contains multiple time series data points, where each time series includes a 28-day lookback period and a 14-day forecast horizon. Since we are dealing with time series data, it is crucial to split the datasets carefully to prevent data leakage. Data leakage occurs when the model is trained on data points that overlap with the forecasting period, leading to an overestimation of model performance [Kaufman et al., 2012]. To address data leakage, the training dataset includes prediction dates from March 19, 2023, to September 24, 2023. The validation set covers the period from October 8, 2023, to October 21, 2023, and finally, the test set comprises data from November 5, 2023, to November 19, 2023. In total, we adopted an 8-month time window, spanning from March 19,

[1] https://pytorch-forecasting.readthedocs.io/en/stable/api/pytorch_forecasting.models.temporal_fusion_transformer.TemporalFusionTransformer.html

2023, to November 19, 2023, excluding December, January, and February due to their distinct characteristics. This partitioning ensures that the model's performance is assessed rigorously and without any contamination from future data.

As described in Section 2, several filters were applied to the dataset. However, despite these filters, issues with high computational time and memory constraints persisted during training. To address these challenges, a sampling approach was implemented. During the sampling process, a random subset of the dataset was created, consisting of a specified percentage of all possible combinations of FC, SPG, and prediction date. When a particular combination of FC, SPG, and prediction date was selected, all associated products were included in the random sample. The impact of reducing the training dataset size was investigated. Different proportions of the training data were sampled, including 1%, 2%, 5%, and 10%. The results revealed that training time scaled linearly with dataset size. However, in terms of validation loss, there was minimal improvement beyond a 5% sample size. Based on these findings, a 5% sample of the training set was chosen as the optimal compromise between computational efficiency and model performance. The validation and test sets are not sampled and contain the full range of observations available within their respective time periods. This approach ensures maximal data utilization, allowing for a thorough evaluation of the model's performance across a wide array of scenarios without any sampling bias.

3.5.2 Hyperparameters

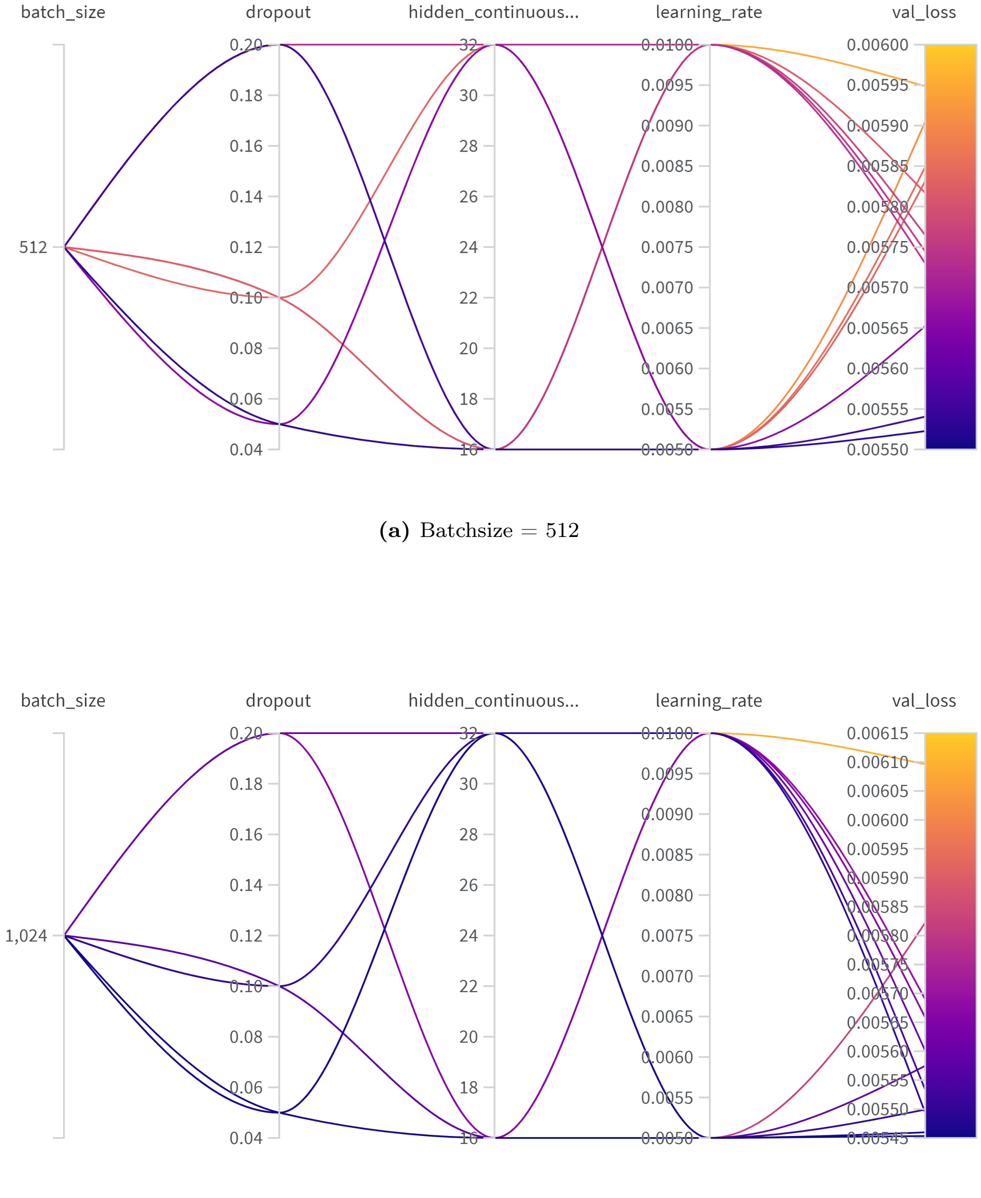
The Temporal Fusion Transformer (TFT) model includes several hyperparameters that play a crucial role in achieving accurate predictions. These hyperparameters require careful tuning to optimize the model's performance. In the table below, we present the key hyperparameters of interest in this study, along with concise descriptions:

Table 2: TFT hyperparameter settings

Hyperparameter	**Definition**
attention_head_size	Number of attention heads
batch_size	Number of time series processed before updating the model weights
dropout	Dropout rate
hidden_continuous_size	Hidden size used for processing continuous variables
hidden_size	Size of the hidden layers
learning_rate	Rate at which the model adjusts its weights during training
lstm_layers	Number of LSTM layers
max_epochs	Maximum number of training epochs before stopping
patience	Number of checks with no improvement before training is stopped
num_samples[1]	Number of MC samples drawn from the model during inference

[1] This is only a hyperparameter for the MC dropout model. It is fixed at 10 due to practical constraints of computational power and memory capacity.

The hyperparameters of both the baseline TFT model and the MC dropout TFT model have seperate tuning processes. This separation is necessary because these models have distinct architectures. The tuning process involves two rounds of random searches, followed by a grid search to identify the optimal hyperparameters. The results, in terms of validation loss, of the grid search of the baseline is shown in Figure 3. Due to a time limit of 24-hours run in Google Colab Pro +, generating similar graphs for the TFT model with MC dropouts was not possible. However, the same approach was applied to tune this model. The resulting optimal hyperparameters are provided in Table 3 of the baseline model and the MC dropout model, respectively. Despite the slight edge in reducing validation loss with a batch size of 1024, as illustrated in Figure 3, we ultimately selected a batch size of 512 for our models. This decision was influenced by the negligible difference in evaluation metrics (detailed in Section 3.5.4) when compared to a batch size of 1024. Moreover, utilizing a larger batch size frequently led to time-outs due to memory limitations. Therefore, we opted for batch size 512 to ensure a balance between performance and computational feasibility. Default values, as defined by the PyTorch package, are used for parameters not displayed.



(a) Batchsize = 512

(b) Batchsize = 1024

Figure 3: Grid search baseline model

Additionally, we include Picnic's TFT model hyperparameters in the last column for reference. A difference between our baseline hyperparameters and Picnic's TFT model is observed, particularly in batch size and learning. This difference could be driven by variations in dataset size, given that our study employs a smaller dataset. The difference in the dropout rate between our baseline TFT model and the MC dropout model is logical. This deviation is primarily due to the architectural distinction between the two models and the specific role of dropout in each. While both models may perform optimally under similar conditions, the MC dropout model incorporates dropout during the prediction phase, a feature absent in the baseline TFT model. Consequently, the dropout rate needs to be adjusted to suit the unique requirements of each model's architecture. Therefore, the slight difference in dropout rates ensures that each model is optimized for its specific design and functionality.

Table 3: Values of hyperparameter optimization

Hyperparameter	**Random Search**	**Grid search**	**Baseline**	**MC dropout**	**Picnic**
max_epochs	1, 3, 5	3	3	3	3
attention_head_size	2, 4	2	2	2	2
batch_size	512, 1024	512, 1024	512	512	1024
dropout	0.05, 0.1, 0.2	0.05, 0.1, 0.2	0.1	0.2	0.1
hidden_continuous_size	16, 32	16, 32	16	16	16
hidden_size	32, 64, 128	32	32	32	32
learning_rate	0.005, 0.01, 0.02	0.005, 0.01	0.005	0.005	0.01
lstm_layers	2	2	2	2	2
patience	3	3	3	3	3

3.5.3 Training

In the model training process, we employ a loss function known as Quantile Loss (QL), defined by the equation:

$$QL_q = \sum_{i=1}^{N} w_i \times \left(\max(q \times (A_i - P_i), (1-q) \times (P_i - A_i))\right).$$

Within the given equation, q signifies the quantile, A_i corresponds to the actual value, P_i represents the predicted value, w_i denotes the weight assigned to the i-th observation, and N stands for the total count of observations. Our model does not differentiate between overestimation and underestimation. Picnic, in its order processing procedures, employs distinct methods to establish safety margins.

Within our model, each observation receives a weight, w_i, which is contingent upon two critical factors: the forecast horizon and specific conditional factors. The forecast horizon refers to the number of days into the future for which the forecast is generated. The specific conditional factors are promotion related and those factors enhance the relevance of the forecasted observation.

To compute both the training and validation losses, each observation's weight is applied to the loss as per the quantile loss function outlined above.This strategic approach theoretically guides the model's focus, emphasizing short-term predictions with specific promotional characteristics.

Table 4: Weighting of observations in two steps

(a) First step weights

Forecast horizon (days)	Weight tw_i
1	10
2	8
3, 4, 5	6
6, 7	4
8, 9, 10	2
Else	1

(b) Second step weights

Case	Weight w_i
`promo_dr_confirmed > 0`	$tw_i \times 3$
`promo_mechanism != 'NOPROMO'`	$tw_i \times 3$
`number_of_recipes > 0`	$tw_i \times 3$
Else	tw_i

3.5.4 Evaluation

In evaluating both the baseline TFT model and the MC dropout TFT model, two key metrics are employed: Weighted Absolute Percentage Error (WAPE) and Weighted Average Percentage Error (WPE). These metrics are defined as follows:

$$WAPE_{D,F,S,I,H} = \frac{\sum_{d,f,s,i,h} \left|A_{d,f,s,i} - P_{d,f,s,i,h}\right|}{\sum_{d,f,s,i} \left|A_{d,f,s,i}\right|}, \quad d \in D, f \in F, s \in S, i \in I, h \in H$$

$$WPE_{D,F,S,I,H} = \frac{\sum_{d,f,s,i,h} \left(A_{d,f,s,i} - P_{d,f,s,i,h}\right)}{\sum_{d,f,s,i} A_{d,f,s,i}}, \quad d \in D, f \in F, s \in S, i \in I, h \in H$$

Here, A represents the actual value, and P is the predicted value at delivery date *d*, FC *f*, slot picking group *s*, article *i*, and forecast horizon *h*. These metrics are highly flexible and can be computed for specific subsets of predictions by customizing the included sets of delivery dates *D*, FCs *F*, SPGs *S*, articles *I*, and forecast horizons *H*. WAPE is our primary evaluation metric, providing an effective measure of overall forecast accuracy. In contrast, WPE offers valuable insights into forecast bias, as it considers the direction of errors without taking their absolute values. A positive WPE indicates overforecasting on average, while a negative WPE indicates underforecasting on average. One notable advantage of WAPE and WPE over their unweighted counterparts, like MAPE and MPE, is their ability to remain robust in the presence of relatively large errors for products with low sales volumes. This makes them more robust and informative in practical demand forecasting retail scenarios.

4 Results

This section delves into a comparative analysis of forecast accuracy, illustrating both the quantitative gains in prediction accuracy and the qualitative improvements in modeling uncertainty achieved through the incorporation of MC dropout. The section starts with showing the accuracy metrics detailed in 3.5.4, examining the performance of both the baseline TFT and the MC dropout TFT model across various operational subsets to demonstrate the improvements attributed to MC dropout. Furthermore, it underscores the ability of MC dropout to generate predictive distributions. Prior to delving into predictive distributions, the general predictive performance of the MC dropout model is discussed. Subsequently, it presents predictive distributions for various forecasting scenarios, showcasing the MC dropout TFT model's ability to produce predictive distributions that are consistent with expectations based on its overall predictive performance. This demonstrates the model's effectiveness in not only enhancing prediction accuracy but also in accurately estimating uncertainty across different scenarios.

4.1 Comparative analysis of forecast accuracy

The total evaluation of forecast accuracy for this thesis' models is summarized in Table 5. The 'Baseline' column represents the performance metrics for the standard TFT model, while the 'MC dropout' column reflects the results of the TFT model with MC dropouts implemented. It can be seen that including MC dropout demonstrates a slight decrease in WAPE from 0.3335 to 0.3285 (-1.50%), suggesting a modest improvement in forecast accuracy. Similarly, the WPE has shown an improvement, with a reduction in the absolute value from -0.1189 to -0.0758 (-36.25%). This indicates a potential decrease in forecast bias when incorporating MC dropout, which is designed to account for model uncertainty. These results provide an initial indication that incorporating uncertainty through MC dropout can positively affect the model's predictive performance, albeit marginally. It is observed that the implementation of MC dropout does not lead to a deterioration in the error metrics used, and thus maintains the model's integrity while addressing uncertainty.

Total	**Baseline**		**MC dropout**	
	WAPE	**WPE**	**WAPE**	**WPE**
All	0.3335	-0.1189	0.3285	-0.0758

Table 5: Total evaluation of forecast accuracy using Weighted Absolute Percentage Error (WAPE) and Weighted Percentage Error (WPE).

In this research, the forecast accuracy is not only assessed on the whole dataset, but also across various subsets, including article category level 1, article shelf life, delivery weekday, forecast horizon, ordered ADR bucket, and SPG. Ordered ADR bucket reflects the sales velocity of the article. This multifaceted evaluation approach allows for a nuanced analysis, revealing how the baseline TFT model, and its Monte Carlo (MC) dropout enhanced counterpart perform under diverse operational conditions. The format for presenting this data follows the structure introduced in Table 5, as previously described.

Table 6 shows diverse changes in WAPE and WPE across different product categories when MC dropout is applied. Some categories like '*Baby & kind*' and '*Gezondheid*' exhibit a significant decrease in WAPE with MC dropout, suggesting that the uncertainty modeling could be particularly beneficial for these categories. The WPE values are generally less negative with MC dropout across most categories, suggesting a reduction in overestimation of forecasts or a more balanced prediction that accounts for uncertainty.

The performance metrics per article shelf life category, see Table 7, show that the '*Very short (2 or fewer days)*' category benefits from a notable decrease in both WAPE and WPE with MC dropout, which may indicate that the uncertainty modeling is effectively capturing the volatility inherent in products with a shorter shelf life.

Article Category Level 1	**Baseline**		**MC dropout**	
	WAPE	**WPE**	**WAPE**	**WPE**
Aardappelen & groente	0.1698	-0.0658	0.1737	0.0035
Baby & kind	0.7551	-0.1335	0.7372	-0.1225
Bakkerij	0.2888	-0.1390	0.2582	-0.0897
Bier & aperitieven	0.6720	-0.2352	0.6782	-0.0917
Diepvries	0.4300	-0.1371	0.4374	-0.0800
Dier	0.5190	-0.1954	0.5181	-0.1511
Drinken	0.3958	-0.1099	0.4062	-0.0474
Drogist	0.6492	-0.1066	0.6800	-0.0649
Fruit	0.2356	-0.1342	0.2041	-0.0505
Gezondheid	0.9890	-0.2017	0.8322	-0.4633
Huishouden	0.3532	-0.0440	0.3558	-0.0923
Kaas	0.3349	-0.0605	0.3291	-0.0951
Koek, snoep & snacks	0.4340	-0.1357	0.4336	-0.1795
Koffie & thee	0.4805	-0.1477	0.4663	-0.2114
Koken, tafelen & vrije tijd	0.7202	-0.4110	0.7470	-0.4124
Maaltijden & gemak	0.4628	-0.1589	0.4514	-0.1293
Ontbijt & zoet beleg	0.3387	-0.0852	0.3320	-0.1145
Pasta, rijst & internationaal	0.4087	-0.0805	0.4132	-0.0826
Vega & vegan	0.3846	-0.0533	0.3862	-0.0137
Vlees & vis	0.2947	-0.0972	0.2924	-0.1032
Vleeswaren, spreads & tapas	0.3636	-0.1335	0.3502	-0.1036
Voorraadkast	0.4661	-0.1386	0.4646	-0.1278
Wijn & bubbels	0.6451	-0.2372	0.6210	-0.2953
Zuivel & eieren	0.3012	-0.1410	0.2866	-0.1129

Table 6: Evaluation of forecast accuracy by article category level 1 using Weighted Absolute Percentage Error (WAPE) and Weighted Percentage Error (WPE).

Article Shelf Life	**Baseline**		**MC Dropout**	
	WAPE	**WPE**	**WAPE**	**WPE**
Very short (2 or fewer days)	0.3297	-0.1039	0.3023	-0.0397
Short (3 to 5 days)	0.2430	-0.0920	0.2369	-0.0448
Long (6 or more days)	0.3842	-0.1346	0.3806	-0.0946

Table 7: Evaluation of forecast accuracy by article shelf life using Weighted Absolute Percentage Error (WAPE) and Weighted Percentage Error (WPE).

The impact of MC dropout varies by day of the week, depicted in Table 8, with certain days like '*Thursday*' and '*Saturday*' showing more pronounced improvements in both WAPE and WPE than others. In Table 9, a comparative evaluation of the forecast accuracy over a 14-step forecast horizon is presented. The MC dropout model generally performs better, as it has lower WAPE and WPE values than the baseline model. The stabilization of errors with increasing forecast horizon indicates that both models are robust and retain their predictive quality over time. The improvements brought by MC dropout are especially noteworthy, as they demonstrate the model's capability to maintain precision and handle uncertainty effectively across varying forecast lengths.

Delivery Weekday	**Baseline**		**MC dropout**	
	WAPE	**WPE**	**WAPE**	**WPE**
Monday	0.3108	-0.1048	0.3022	-0.0954
Tuesday	0.3528	-0.1114	0.3528	-0.1108
Wednesday	0.3628	-0.1181	0.3640	-0.0844
Thursday	0.3733	-0.1275	0.3711	-0.0547
Friday	0.3169	-0.1402	0.3036	-0.0348
Saturday	0.3201	-0.1070	0.3192	-0.0478
Sunday	0.3052	-0.1224	0.2951	-0.1055

Table 8: Evaluation of forecast accuracy by delivery weekday using Weighted Absolute Percentage Error (WAPE) and Weighted Percentage Error (WPE).

Forecast Horizon	**Baseline**		**MC dropout**	
	WAPE	**WPE**	**WAPE**	**WPE**
1	0.1692	-0.0645	0.1409	-0.0144
2	0.2845	-0.0978	0.2737	-0.0504
3	0.3201	-0.1157	0.3111	-0.0763
4	0.3351	-0.1215	0.3277	-0.0848
5	0.3435	-0.1222	0.3367	-0.0783
6	0.3497	-0.1267	0.3443	-0.0794
7	0.3530	-0.1237	0.3510	-0.0849
8	0.3550	-0.1320	0.3560	-0.0864
9	0.3581	-0.1340	0.3575	-0.0870
10	0.3592	-0.1318	0.3585	-0.0857
11	0.3594	-0.1272	0.3593	-0.0836
12	0.3595	-0.1249	0.3598	-0.0844
13	0.3593	-0.1216	0.3595	-0.0831
14	0.3605	-0.1206	0.3590	-0.0818

Table 9: Evaluation of forecast accuracy by forecast horizon using Weighted Absolute Percentage Error (WAPE) and Weighted Percentage Error (WPE).

The ADR bucket evaluation, displayed in Table 10, demonstrates improved accuracy over the baseline, as indicated by the lower WAPE scores in all ADR buckets. While WPE scores also reflect better performance for the MC dropout model in the initial categories, an anomaly is observed in the *'Extreme fast mover'* bucket, where the MC dropout model's WPE becomes positive. This suggests that for items with high transaction frequencies, the model may be over-estimating demand to a certain extent. Both models struggle with the *'Extreme slow mover'* category, suggesting a common difficulty in forecasting items with very low transaction frequencies.

Ordered ADR Bucket	**Baseline**		**MC dropout**	
	WAPE	**WPE**	**WAPE**	**WPE**
1 - Extreme slow mover	0.8425	-0.1841	0.8101	-0.0677
2 - Slow mover	0.4234	-0.1123	0.4163	-0.0868
3 - Fast mover	0.2301	-0.1103	0.2229	-0.0799
4 - Extreme fast mover	0.1387	-0.1071	0.1202	0.0154
5 - ignore_token	0.5663	-0.2660	0.5305	-0.2316

Table 10: Evaluation of forecast accuracy by ordered ADR bucket using Weighted Absolute Percentage Error (WAPE) and Weighted Percentage Error (WPE).

Lastly, in Table 11, the WAPE and WPE are evaluated across the two different SPGs. There is a slight improvement in WAPE for both '*Morning*' and '*Evening*' SPGs with MC dropout, indicating a consistent benefit across different operational time frames. The WPE improvements for the SPG are quite uniform, suggesting that MC dropout provides a consistent reduction in forecast bias irrespective of the time of day.

Slot Picking Group	**Baseline**		**MC dropout**	
	WAPE	**WPE**	**WAPE**	**WPE**
Morning	0.3572	-0.1154	0.3523	-0.0758
Evening	0.3106	-0.1224	0.3054	-0.0759

Table 11: Evaluation of forecast accuracy by slot picking group using Weighted Absolute Percentage Error (WAPE) and Weighted Percentage Error (WPE).

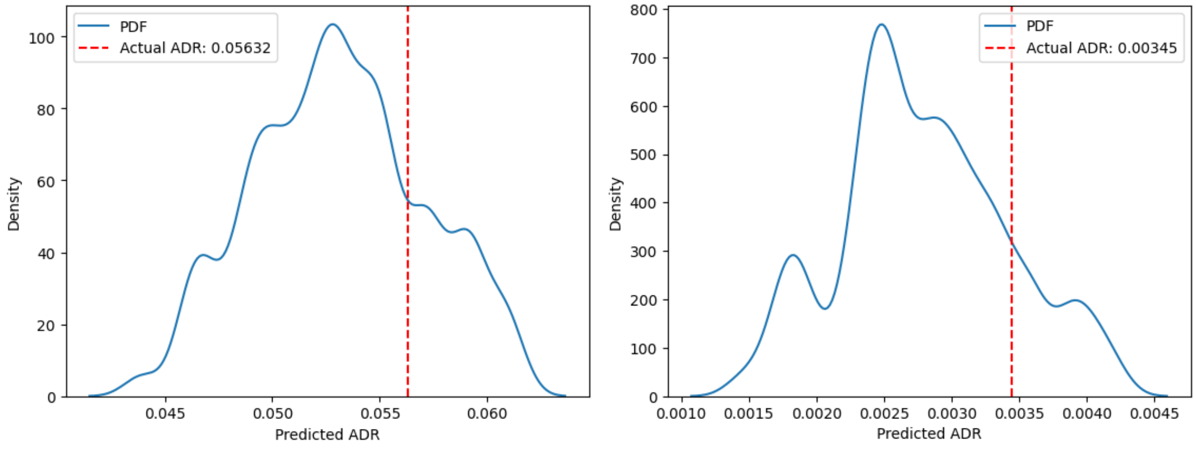
4.2 General predictive performance

In terms of predictive performance, both models exhibit similar trends, as outlined in Tables 5 to 11. Firstly, the '*Aardappelen & groente*' article category performs the best, while '*Gezondheid*' shows the weakest performance. Additionally, articles with a '*Short*' shelf life outperform those with a '*Long*' shelf life. When considering delivery weekdays, *Monday* and *Sunday* demonstrate relatively better predictive performance compared to *Wednesday* and *Thursday.* Analyzing the forecast horizon, we observe that predictive performance tends to stabilize with longer horizons, with marginal changes in WAPE and WPE from horizon 10 onwards. Notably, the models excel in predicting *extreme fast movers* compared to *extreme slow movers*, as evident from the ordered ADR bucket analysis. Finally, the model exhibits a slight performance variation between deliveries scheduled for the '*morning*' and '*evening*', with a slightly better performance observed for the '*evening*' slot.

4.3 Predictive distribution analysis

Following our examination of the predictive performance of both models, we aim to further investigate the predictive capabilities of our MC dropout model. In this section, we present the probability density function of the ADR for a specific article, SPG, prediction date, and forecast horizon. This article belongs to a category level 1, has a specific article shelf life, and falls within a certain ordered ADR bucket. The predictive distribution, which includes the actual predictions, is detailed in Appendix A. As demonstrated in Section 4.2, these characteristics influence predictive performance. By creating predictive distributions for different forecast scenarios, we aim to examine how the model's predictions align with observed trends. These scenarios will illustrate the impact of changes in individual characteristics while *keeping others constant*. To this end, we compare the probability density functions using the Coefficient of Variation (CV), a standardized measure of the dispersion of the probability distribution. The CV is calculated as the ratio of the standard deviation (σ) to the mean (μ) of the distribution, expressed as a percentage: $\text{CV} = \frac{\sigma}{\mu} \times 100\%$. This metric allows us to quantitatively assess the spread of predicted ADR values, offering insights into the precision and reliability of the model's forecasts under different conditions. Through generating and analyzing probability density functions, we can evaluate not only the point predictions but also the associated uncertainty of each forecast. This enables us to assess the robustness of our model's predictions and gain insights into the variability of forecasted outcomes.

In Figure 4, the probability density function of an article with a *short* shelf life is compared to an article with a *long* shelf life keeping other characteristics constant. The article with a *long* shelf life exhibit a higher Coefficient of Variation (CV) of 22.14%, signaling a greater relative variability and consequently, less certainty in the model's predictions for these products. On the other hand, *short* shelf life items present a CV of 7.46%, indicating a more compact distribution of predicted values in relation to their mean, and suggesting increased confidence in the model's forecasts for these items. This perspective is confirmed by the data in Table 7, which shows a marginally greater WAPE for *long* shelf life items compared to *short* shelf life items.

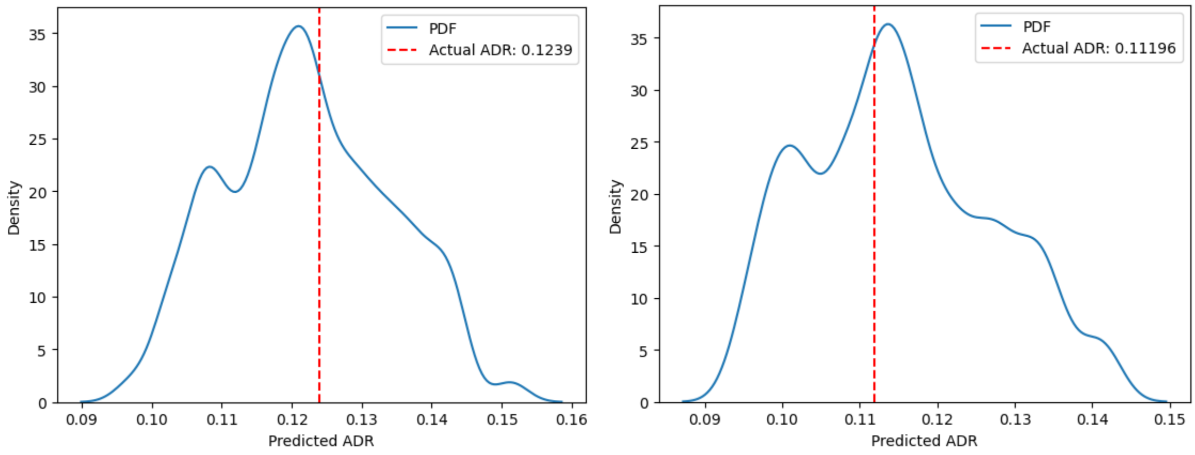


(a) Article shelf life = short

(b) Article shelf life = long

Figure 4: Comparison of ADR probability density functions for articles with short versus long shelf life, keeping other characteristics constant.

In Figure 5, the probability density function of an article with a *Monday* as delivery weekday is compared to an article with a *Wednesday* as delivery weekday keeping other characteristics constant. The prediction for *Wednesday* exhibits a CV of 10.37%, indicating a modest spread of predicted ADR values. This spread, as depicted in Figure 5b, shows that while there is some variability in the model's forecasts, the actual ADR is well within the central region of the probability density function, suggesting an overall reliable forecast for this weekday. In comparison, the predictions for *Monday* demonstrate a lower CV of 9.67%, which is indicative of a tighter clustering of forecasted ADRs around the actual value, as shown in Figure 5a. This tighter probability density function signifies a higher degree of precision and less variability in the model's forecast for *Monday*, aligning with a lower WAPE and a more positive WPE for Mondays in the MC dropout model, as detailed in Table 8.



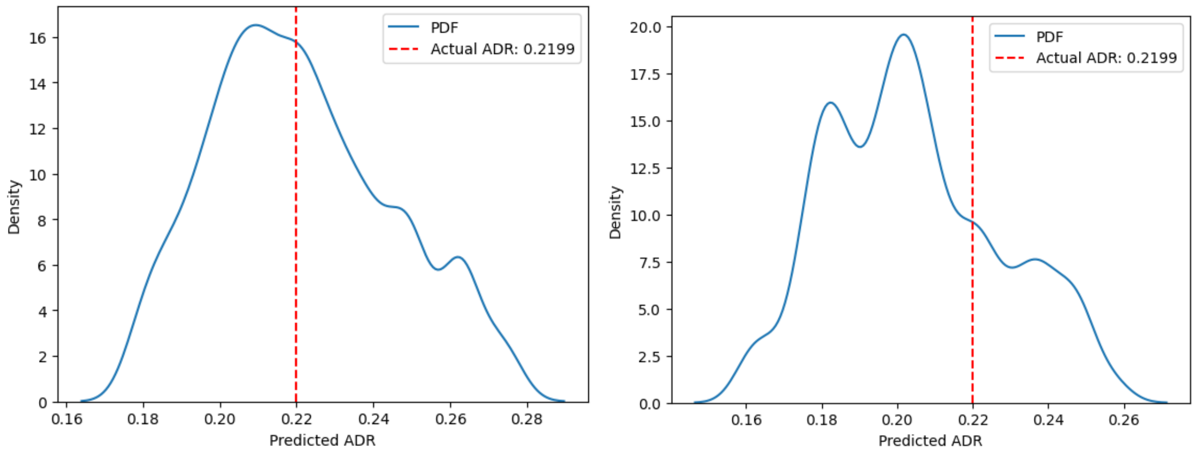
(a) Delivery weekday = Monday

(b) Delivery weekday = Wednesday

Figure 5: Comparison of ADR probability density functions for weekday delivery on Monday versus Wednesday, keeping other characteristics constant.

As shown in Table 9, there is an increasing trend of uncertainty as the forecasting period extends since the WAPE increases. In Figure 6 the probability density function of an article with a *short forecast horizon* is compared to the same article with a *long forecast horizon* keeping other characteristics constant. Indeed, an increased level of uncertainty with the extension of the forecast period is observed. The CV for forecast horizon 1 is 10.74%, which, while indicative of some degree of variability, is relatively low. This is visually supported by the histogram in Figure 6a, where the predicted ADR probability density function is tightly clustered around the actual ADR value, demonstrating a strong confidence in the short-term forecast accuracy. In contrast, for forecast horizon 10, as depicted in Figure 6b, the CV slightly rises to 11.14%, suggesting an incremental increase in forecast uncertainty. The histogram for forecast horizon 10 displays a wider dispersion of predicted values, implying a broader range of potential outcomes and a higher degree of uncertainty. Despite this increase, the predictions for both horizons remain in proximity to the actual ADR value, with the wider probability density function for horizon 10 indicating the model's appropriate accommodation for the increased uncertainty that is typically characteristic of longer-term forecasting. These observations from the probability density functions underline the model's adeptness in yielding precise short-term forecasts and its systematic accommodation for the heightened uncertainty in long-term predictions, as quantified by the respective CVs.

In Figure 7, the probability density functionn of an extreme slow mover article is compared to the same extreme fast mover article keeping other characteristics constant. The prediction

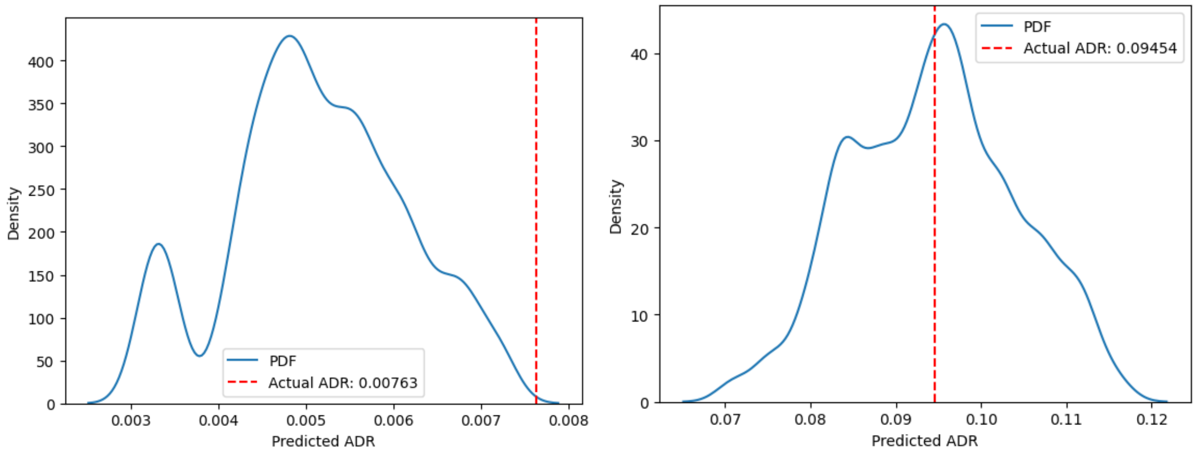


(a) Forecast horizon = 1

(b) Forecast horizon = 10

Figure 6: Comparison of ADR probability density functions for short versus long forecast horizons, keeping other characteristics constant.

accuracy of the 'extreme fast mover' versus 'extreme slow mover' categories reveals significant contrasts in Table 10, which can also clearly be seen in 7. The 'extreme fast mover' group, which has a CV of 10.26%, demonstrates a relatively high degree of forecast confidence. This is visually evidenced by the tight histogram in Figure 7b, where the predicted ADR closely aligns with the actual ADR, suggesting a high level of model accuracy. Conversely, the 'extreme slow mover' category, with a notably higher CV of 19.60%, displays a wider spread of predicted ADRs as seen in Figure 7a. This wider probability density function indicates a greater degree of uncertainty and a reduced level of precision in the model's forecasts for these items. The distinction in predictive accuracy between the two categories is also reflected in the reported WAPE and WPE metrics from Table 10, where 'extreme slow movers' register higher errors, reaffirming the trends observed in the predictive distributions.

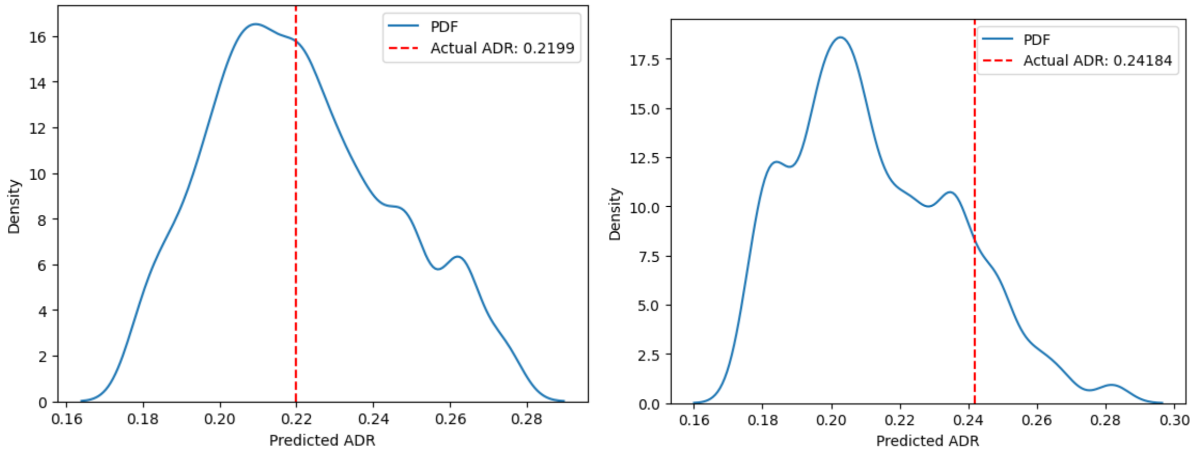


(a) Ordered ADR bucket = extreme slow mover

(b) Ordered ADR bucket = extreme fast mover

Figure 7: Comparison of ADR probability density functions for ADR bucket extreme slow mover versus extreme fast mover, keeping other characteristics constant.

In Table 11, the probability density function of a *morning* SPG is compared to an *evening* SPG keeping other characteristics constant. The *evening* SPG displays a distribution with a CV of 10.74%, which suggests a commendable level of precision in the model's predictions during these hours. This is visually represented in Figure 8a where the predicted ADRs are narrowly concentrated within the probability density function, reflecting a tighter concentration of forecasts around the actual ADR. In contrast, the *morning* SPG, with a CV of 11.14%, shows a marginally broader spread of predicted ADRs as seen in Figure 8b. Despite the slight increase in variability, the actual ADR is still proximate to the peak of the probability density function, indicating that the model's forecasts remain reliably accurate even with the increased spread. Although the '*morning* forecasts demonstrate a slightly higher degree of uncertainty compared to the *evening*, both time slots exhibit probability density functions that center near the actual ADR values, affirming the model's adeptness in capturing the SPG that may influence the forecast accuracy.



(a) SPG = evening

(b) SPG = morning

Figure 8: Comparison of ADR probability density functions for SPG evening versus morning, keeping other characteristics constant.

5 Conclusion

The core aim of this thesis, as detailed in Section 1, was to advance the TFT methodology for the forecasting of article demand. This work sought to enable the TFT model to not only predict demand but to also generate a predictive distribution that could serve as the foundation for decision-making under uncertainty. Through the integration of Monte Carlo dropouts, this research has not only quantitatively enhanced prediction accuracy but has also qualitatively improved the model's capacity to encapsulate uncertainty.

The empirical results signal a promising direction, with the inclusion of MC dropout leading to a modest yet notable improvement in forecast precision, as evidenced by a decrease in WAPE by 1.50%. Additionally, the improvement in WPE by 36.25% underscores the value of integrating uncertainty into the forecasting process. While the WAPE and WPE generally improved across different subsets, there are certain categories that stood out due to their significant performance enhancements. Particularly, categories such as 'Baby & kind' and 'Gezondheid' have shown significant enhancements in forecast accuracy, suggesting that the uncertainty modeling is especially advantageous for certain product types. The improvement is further accentuated in the ’Very short (2 or fewer days)’ category, where the model adeptly captures the volatility associated with products with shorter shelf lives. While the enhanced model exhibits proficiency across various forecast lengths, it continues to face challenges within the ’Extreme slow mover’ category, highlighting an area where both the standard and the MC dropout-enhanced TFT models encounter limitations. This observation invites further investigation into the intricacies of forecasting items with low transaction frequencies and opens up avenues for future research to refine models for such challenging categories.

Moreover, the thesis presents predictive distributions across a spectrum of forecasting scenarios. The MC dropout TFT model’s ability to yield distributions that align with its overall performance further substantiates the efficacy of this approach. By generating and examining these distributions, the research moves beyond point predictions to a more holistic evaluation of the forecast’s reliability and the inherent uncertainty of each outcome. Consequently, this allows for a deeper understanding of the model’s robustness and a nuanced perspective on the variability of forecasted outcomes.

Given that predictions must be executed multiple times for MC dropout method, the runtime for this method extends accordingly. This extended runtime is an important factor to consider in the practical application of the MC dropout architecture. One potential solution to mitigate this issue is to execute predictions in parallel, provided that the computational infrastructure supports it. Such an approach addresses operational efficiency, further enhancing the model’s applicability in real-world scenarios.

To conclude, this thesis contributes to the field of demand forecasting by demonstrating that the careful modeling of uncertainty by incorporating MC dropouts can yield significant benefits. This thesis establishes a foundation for more robust decision-making in supply chain and inventory management, especially valuable in contexts where precise predictions are scarce yet critical for operational efficiency.

6 Discussion

To set the stage for our comprehensive discussion, it’s essential to outline the methodology that underpins this thesis. The core of our approach involved integrating MC dropout with the TFT model, specifically tailored for the retail demand forecasting context. This integration aimed to address two primary objectives: enhancing the predictive accuracy of demand forecasts and improving the model’s ability to quantify and encapsulate uncertainty in its predictions.

The integration of MC dropout with the TFT for retail demand forecasting represents a significant advancement in the field, addressing the critical need for models that balance predictive accuracy with the capability to quantify uncertainty. To the best of our knowledge, this thesis is the first to incorporate MC dropout within the TFT model architecture in the context of retail, illustrating its potential to enhance decision-making under uncertain conditions. However, the journey towards refining this model and fully realizing its implications for retail and beyond is accompanied by several challenges and areas ripe for further research.

One of the primary limitations encountered in this study stems from the constraints on computational resources, which necessitated a limited grid search for hyperparameter tuning. This approach introduces a degree of uncertainty regarding the optimality of the chosen hyperparameters, potentially affecting the model's performance and generalizability. Additionally, the dataset's scope was limited to a single FC and included only articles identified by even numbers, representing a fraction of the comprehensive data available from Picnic. These constraints highlight the need for caution when extrapolating the study's findings, as they may not fully capture the model's capabilities across a broader dataset.

Despite these limitations, our findings underscore the benefits of integrating MC dropout with the TFT model, particularly in enhancing predictive accuracy and the model's ability to encapsulate uncertainty. This not only improves forecast reliability but also provides a more solid foundation for decision-making in uncertain environments. In light of these advantages, we advocate for the adoption of this approach in retail demand forecasting, emphasizing the exploration of parallel computation techniques to counterbalance the increased runtime associated with MC dropout. Such strategies aim to preserve the model's operational efficiency without compromising the benefits derived from enhanced uncertainty quantification.

The study also identifies potential paths for enhancing the TFT model's performance further. Currently, the model utilizes article category levels in a manner that reflects human intuition rather than quantitative properties. By revisiting this strategy and potentially grouping articles according to quantitative characteristics like historical sales figures, the model's precision could see considerable enhancement. Additionally, incorporating features like geographical location, placement within the application, and insights from promotional emails sent to customers could further refine the model's effectiveness.

Looking ahead, the scope for future research is vast and varied. Exploring hybrid models, adopting advanced techniques such as transfer learning, and extending the application of the enhanced TFT model to other domains like energy or healthcare, could provide valuable insights into its effectiveness. Moreover, improving model interpretability stands out as an essential objective, particularly as the complexity added by MC dropouts complicates the understanding of variable importance. Developing methods to better visualize variable impacts or quantify the contribution of input features, even in the presence of stochasticity, could make the model more accessible and useful to practitioners.

Finally, this thesis underscores the fundamental shift introduced by quantifying uncertainty in forecasts. Providing a probabilistic understanding of future demand allows businesses to prepare for a range of outcomes, bolstering their resilience against market volatility. However, the challenge remains in translating these probabilistic forecasts into actionable strategies, emphasizing the need for effective communication of uncertainty principles to decision-makers. This underscores the importance of bridging the gap between advanced forecasting techniques and practical business applications, ensuring that the benefits of uncertainty quantification are fully leveraged in the decision-making process.

References

F. M. Alvarez, A. Troncoso, J. C. Riquelme, and J. S. A. Ruiz. Energy time series forecasting based on pattern sequence similarity. *IEEE Transactions on Knowledge and Data Engineering*, 23(8):1230–1243, 2010.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

C. Bui, N. Pham, A. Vo, A. Tran, A. Nguyen, and T. Le. Time series forecasting for healthcare diagnosis and prognostics with the focus on cardiovascular diseases. In *6th International Conference on the Development of Biomedical Engineering in Vietnam (BME6) 6*, pages 809–818. Springer, 2018.

J. Cao, Z. Li, and J. Li. Financial time series forecasting model based on ceemdan and lstm. *Physica A: Statistical mechanics and its applications*, 519:127–139, 2019.

R. D. De Vleaux, J. Schumi, J. Schweinsberg, and L. H. Ungar. Prediction intervals for neural networks via nonlinear regression. *Technometrics*, 40(4):273–282, 1998.

G. Dudek. Short-term load forecasting using random forests. In *Intelligent Systems' 2014: Proceedings of the 7th IEEE International Conference Intelligent Systems IS'2014, September 24-26, 2014, Warsaw, Poland, Volume 2: Tools, Architectures, Systems, Applications*, pages 821–828. Springer, 2015.

Z. Eaton-Rosen, F. Bragman, S. Bisdas, S. Ourselin, and M. J. Cardoso. Towards safe deep learning: accurately quantifying biomarker uncertainty in neural network predictions. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*, pages 691–699. Springer, 2018.

G. Feng, L. Zhang, F. Ai, Y. Zhang, and Y. Hou. An improved temporal fusion transformers model for predicting supply air temperature in high-speed railway carriages. *Entropy*, 24(8): 1111, 2022.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Y. Gal, J. Hron, and A. Kendall. Concrete dropout. *Advances in neural information processing systems*, 30, 2017.

Y. Gal et al. Uncertainty in deep learning. 2016.

J. Hernandez-Lobato, Y. Li, M. Rowland, T. Bui, D. Hernández-Lobato, and R. Turner. Black-box alpha divergence minimization. In *International conference on machine learning*, pages 1511–1520. PMLR, 2016.

J. M. Hernández-Lobato and R. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, pages 1861–1869. PMLR, 2015.

G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

P. C. Huy, N. Q. Minh, N. D. Tien, and T. T. Q. Anh. Short-term electricity load forecasting based on temporal fusion transformer model. *IEEE Access*, 10:106296–106304, 2022.

J. G. Hwang and A. A. Ding. Prediction intervals for artificial neural networks. *Journal of the American Statistical Association*, 92(438):748–757, 1997.

P. R. Junior, F. L. R. Salomon, E. de Oliveira Pamplona, et al. Arima: An applied time series forecasting model for the bovespa stock index. *Applied Mathematics*, 5(21):3383, 2014.

Z. Karevan and J. A. Suykens. Transductive lstm for time-series prediction: An application to weather forecasting. *Neural Networks*, 125:1–9, 2020.

S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6 (4):1–21, 2012.

A. Khosravi, S. Nahavandi, D. Creighton, and R. Naghavizadeh. Uncertainty quantification for wind farm power generation. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2012.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Y. Li and Y. Gal. Dropout inference in bayesian neural networks with alpha-divergences. In *International conference on machine learning*, pages 2052–2061. PMLR, 2017.

B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

A. Loquercio, M. Segu, and D. Scaramuzza. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160, 2020.

C.-X. Lv, S.-Y. An, B.-J. Qiao, and W. Wu. Time series analysis of hemorrhagic fever with renal syndrome in mainland china by using an xgboost forecasting model. *BMC infectious diseases*, 21:1–13, 2021.

D. Mircetic, B. Rostami-Tabar, S. Nikolicic, and M. Maslaric. Forecasting hierarchical time series in supply chains: an empirical investigation. *International Journal of Production Research*, 60(8):2514–2533, 2022.

G. Nunnari and V. Nunnari. Forecasting monthly sales retail time series: a case study. In *2017 IEEE 19th conference on business informatics (CBI)*, volume 1, pages 1–6. IEEE, 2017.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

P.-F. Pai, K.-P. Lin, C.-S. Lin, and P.-T. Chang. Time series forecasting by a seasonal support vector regression model. *Expert Systems with Applications*, 37(6):4261–4265, 2010.

M. Rußwurm, M. Ali, X. X. Zhu, Y. Gal, and M. Körner. Model and data uncertainty for satellite time series forecasting with deep recurrent models. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 7025–7028. IEEE, 2020.

A. Sagheer and M. Kotb. Time series forecasting of petroleum production using deep lstm recurrent networks. *Neurocomputing*, 323:203–213, 2019.

I. Salehin and D.-K. Kang. A review on dropout regularization approaches for deep neural networks within the scholarly domain. *Electronics*, 12(14):3106, 2023.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong. Probabilistic forecasting of wind power generation using extreme learning machine. *IEEE Transactions on Power Systems*, 29(3): 1033–1044, 2013.

B. Wu, L. Wang, and Y.-R. Zeng. Interpretable wind speed prediction with multivariate time series and temporal fusion transformers. *Energy*, 252:123990, 2022.

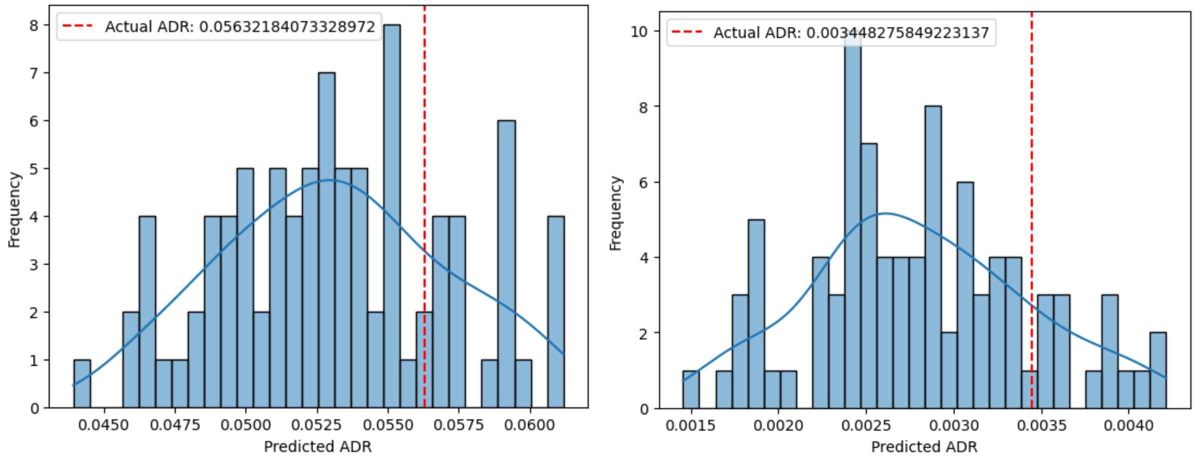
M. Yang, S. Fan, and W.-J. Lee. Probabilistic short-term wind power forecast using componential sparse bayesian learning. *IEEE Transactions on Industry Applications*, 49(6):2783–2792, 2013.

Y. Yang, S. Li, W. Li, and M. Qu. Power load probability density forecasting using gaussian process quantile regression. *Applied Energy*, 213:499–509, 2018.

H. Zhang, Y. Zou, X. Yang, and H. Yang. A temporal fusion transformer for short-term freeway traffic speed multistep prediction. *Neurocomputing*, 500:329–340, 2022.

A Predictive distributions

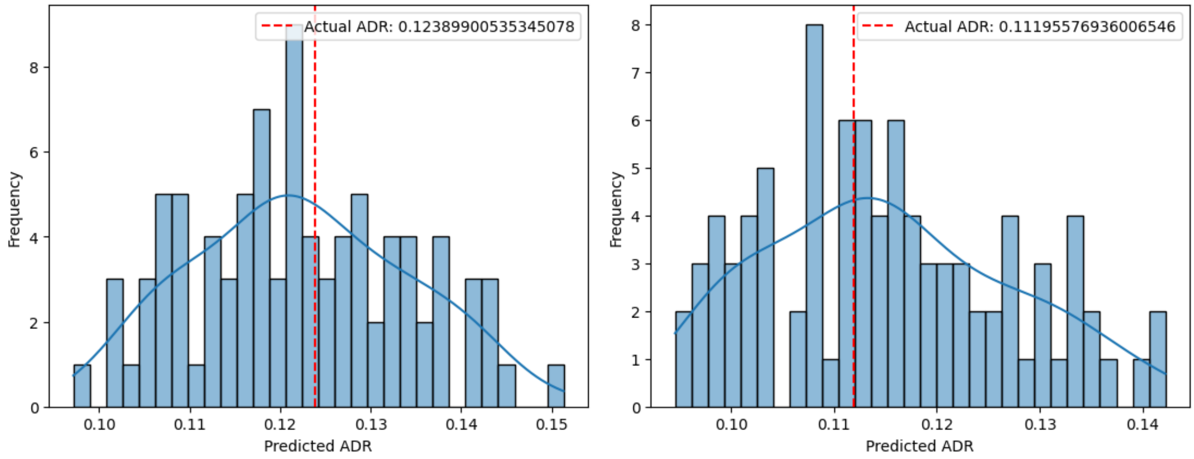
In this Appendix, the predictive distributions corresponding to the probability density functions detailed in Section 3.4 are provided for reference.



(a) Article shelf life = short

(b) Article shelf life = long

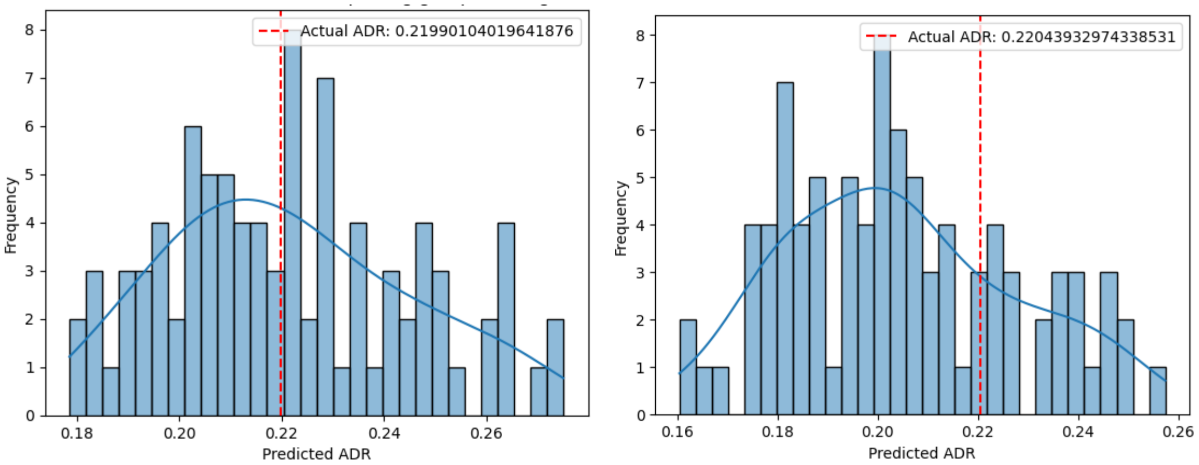
Figure 9: Comparison of ADR predictive distributions for articles with short versus long shelf life, keeping other characteristics constant.



(a) Delivery weekday = Monday

(b) Article shelf life = long

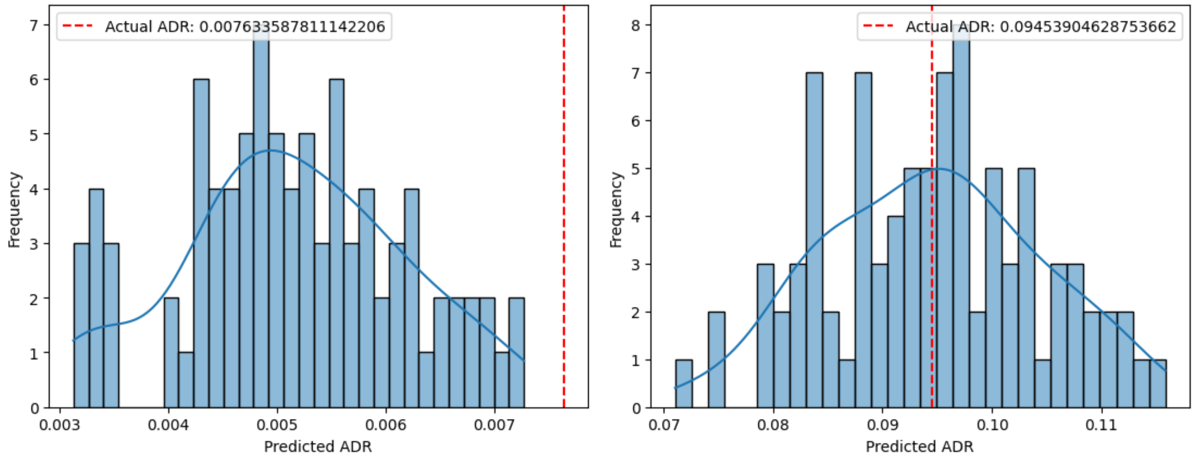
Figure 10: Comparison of ADR predictive distributions compared for weekday delivery on Monday versus Wednesday, keeping other characteristics constant



(a) Forecast horizon = 1

(b) Forecast horizon = 10

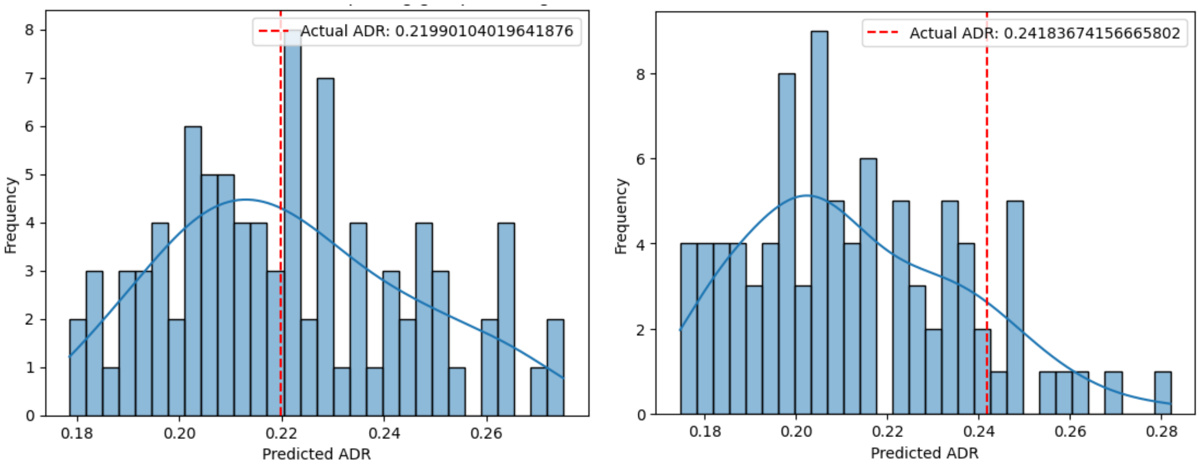
Figure 11: Comparison of ADR predictive distributions for short versus long forecast horizons, keeping other characteristics constant.



(a) Ordered ADR bucket = extreme slow mover

(b) Ordered ADR bucket = extreme fast mover

Figure 12: Comparison of ADR predictive distributions for ADR bucket extreme slow mover versus extreme fast mover, keeping other characteristics constant.



(a) SPG = evening

(b) SPG = morning

Figure 13: Comparison of ADR predictive distributions for SPG evening versus morning, keeping other characteristics constant.