ADVANCING TIME SERIES FORECASTING: HIERARCHICAL METHODS, PROBABILISTIC MODELS, AND DOMAIN KNOWLEDGE INTEGRATION FROM POWER SYSTEMS TO RETAIL

A Dissertation
Presented to
The Academic Faculty

By

Hanyu Zhang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering
College of Engineering

Georgia Institute of Technology

May 2025

ADVANCING TIME SERIES FORECASTING: HIERARCHICAL METHODS, PROBABILISTIC MODELS, AND DOMAIN KNOWLEDGE INTEGRATION FROM POWER SYSTEMS TO RETAIL

Thesis committee:

Dr. Pascal Van Hentenryck
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. B. Aditya Prakash
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. Yao Xie
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Terrence Mak
Department of Data Science & AI
Monash University

Dr. Siva Theja Maguluri
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Date approved: January 1, 2025

Observe the present to grasp the hidden truth; study the past to foresee the unfolding future.

Liezi: Chapter on Explaining Conjunctions

For my parents

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Professor Pascal Van Hentenryck, for his invaluable guidance, unwavering support, and inspiring enthusiasm throughout my Ph.D. journey. His vision and insights have fundamentally shaped my research perspective and academic growth.

I am especially grateful to my mentors, Dr. Mathieu Tanneau and Dr. Reza Zandehshahvar, whose dedication and expertise have been instrumental in my research development. Their patience and willingness to share knowledge have truly enriched my doctoral experience.

My sincere appreciation goes to my dissertation committee members for their thoughtful feedback and contributions to improving this work.

The support from my fellow researchers in the lab has been invaluable. I am particularly thankful to Rose, Rosemary, Haoruo, Rabab, Jihye, and Meen-Sung for their friendship, encouragement, and the countless discussions that helped me navigate both research challenges and life as a Ph.D. student. Their presence and strength as women in engineering has been truly inspiring, creating an empowering environment in our research group. I also want to thank my collaborators and lab mates - Tejas, Anthony, Chaofan, Hongzhao, Terrence, and Wenbo - for their contributions to our joint research efforts and for creating an engaging academic environment.

Finally, I would like to express my heartfelt gratitude to my family and those who have become family to me. Your unwavering support, endless encouragement, and unconditional love have been my foundation throughout this journey. I could not have reached this milestone without you by my side.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Time series forecasting plays a critical role in modern power systems and supply chains, where accurate predictions enable efficient operations and resource allocation. This thesis advances the field through three novel methodological contributions.

First, the thesis introduces the Bundle-Predict-Reconcile (BPR) framework for hierarchical wind power forecasting. BPR integrates asset bundling with machine learning and forecast reconciliation techniques to improve prediction accuracy at both individual and system-wide levels. This approach demonstrates significant improvements over traditional methods when tested on large-scale industry data.

Second, the dissertation develops a weather-informed probabilistic forecasting framework that combines Temporal Fusion Transformers with Gaussian copula methods. This approach effectively captures spatio-temporal dependencies in renewable energy systems while providing robust uncertainty quantification. The framework shows superior performance in predicting load, wind, and solar power generation across the MISO system.

Finally, this work presents LLMForecaster, a novel approach that leverages large language models to incorporate unstructured textual information into time series forecasts. Applied to retail demand prediction, LLMForecaster significantly improves forecast accuracy for products with seasonal patterns by capturing domain knowledge from product descriptions and attributes.

Together, these contributions advance the state-of-the-art in time series forecasting by effectively integrating hierarchical structures, probabilistic methods, and domain knowledge across different applications.

CHAPTER 1
INTRODUCTION AND BACKGROUND

This thesis addresses three fundamental challenges in modern time series forecasting: hierarchical structure, uncertainty quantification, and domain knowledge integration. While these challenges appear distinct, they share a common theme: the need to capture and leverage complex relationships in data to improve forecast accuracy. Through novel methodological contributions in each area, this work demonstrates how advanced machine learning techniques can be combined with domain-specific insights to create more robust and accurate forecasting systems.

The thesis makes three main methodological contributions, each addressing a critical aspect of time series forecasting. First, it introduces the Bundle-Predict-Reconcile (BPR) framework, which improves hierarchical forecasting by learning optimal groupings of related time series. Second, it develops a novel probabilistic forecasting approach that combines weather information with copula methods to better capture uncertainties in renewable energy systems. Finally, it presents LLMForecaster, which leverages large language models to incorporate unstructured textual information into time series forecasts. Together, these contributions advance the state-of-the-art in time series forecasting while maintaining practical applicability to real-world problems.

The increasing integration of intermittent renewable energy sources (RES), such as wind and solar, into US power grids introduces significant operational uncertainties. These uncertainties arise primarily due to the variable nature of these energy sources, especially wind power, which is difficult to predict due to its high variability. Accurate and reliable forecasts are essential to manage these uncertainties and stabilize grid operations effectively.

This thesis presents novel methodologies to enhance forecast precision and provides robust uncertainty quantification and scenario generation techniques. A key innovation is the development of the Bundle-Predict-Reconcile (BPR) framework, detailed in chapter 2. This framework aims to improve the accuracy of time series predictions by integrating asset bundling with advanced machine learning and forecast reconciliation techniques. Specifically, the BPR framework strategically bundles assets to create an intermediate hierarchy level—enhancing the predictive performance by introducing an auxiliary learning task that focuses on predicting bundle-level time series. This is complemented by a reconciliation process that ensures consistency across predictions at the asset, bundle, and fleet levels. Additionally, the proposed framework also introduces new asset-bundling criteria that capture the spatio-temporal dynamics of wind power time series.

The stochastic and unpredictable nature of RES contributes to uncertainty in forecasts of power generation and demand, complicating efforts towards grid stability and optimization. Addressing these challenges, chapter 3 explores the development of advanced probabilistic forecasting methods that are capable of handling the high-dimensional variability of RES. Utilizing weather data as covariates, this work enhances the accuracy of power demand and generation forecasts for wind and solar energies by preserving spatio-temporal correlations among the predicted variables. The methodologies developed herein lay the groundwork for reliable scenario generation, which is pivotal for enabling stochastic optimization in grid planning and risk assessment.

Beyond power systems, the thesis also explores novel approaches to demand forecasting in retail settings where seasonal patterns and special events significantly impact demand. Chapter 4 introduces LLMForecaster, an innovative framework that leverages large language models to incorporate unstructured textual information like product descriptions into time series forecasts. This approach particularly improves forecast accuracy for products with seasonal demand patterns tied to holidays and special events, where traditional time series models often fail to anticipate demand surges. By utilizing LLMs to process product metadata and contextual information, LLMForecaster demonstrates how modern language models can enhance traditional forecasting techniques in retail applications.

1.1 Background

1.1.1 Time Series Forecasting

Historically, the domain of time series prediction was dominated by statistical methods, with Autoregressive Integrated Moving Average (ARIMA) [1] model being a prominent example. However, these methods often fall short in capturing the non-linear and non-stationary nature of the time series and might not be reliable for forecasting RES. The advent of Recurrent Neural Networks (RNNs) marked a significant shift, demonstrating superior capabilities in capturing complex temporal dependencies inherent in time series data [2, 3]. Notably, the sequence-to-sequence (S2S) RNN architecture [2], initially developed for neural machine translation, has inspired the development of innovative time series forecasting models such as multi-quantile RNN [4], Temporal Fusion Transformer (TFT) [5], and DeepAR model [6]. By leveraging cross-series and cross-temporal learning, these methods have gained promising performance across diverse fields, which motivates their utilization for RES forecasting [7, 8, 9]. For a comprehensive survey of the recent advancements in RNN-based and general deep learning architectures in time series forecasting, readers are referred to [10, 11].

Despite their significant advancements, RNN-based models and their derivatives encounter substantial challenges in accurately forecasting the dynamics of modern power systems. These challenges stem largely from the intermittency, uncertainty, and stochasticity associated with the integration of RES and distributed energy resources into existing power networks [12]. Such complexities highlights the urgent need to pivot from the deterministic forecasting approaches to uncertainty quantification of forecast errors [13, 14]. Many studies have adopted parametric approaches to model the distribution of forecast errors, utilizing distributions such as the beta distribution [15] and the logit-normal distribution [16]. These approaches, while valuable, often presume a specific functional form for the forecast errors, which may not always align with the observed data. In contrast, non-parametric methods, particularly quantile regression [13, 17], offer a more flexible solution. Quantile regression excels in providing a comprehensive quantification of uncertainty without the need to assume a predetermined functional form or distribution of forecast errors. This methodological flexibility is especially pertinent in addressing the complexities of stochastic power systems, thereby enhancing decision-making processes in critical operations such as risk management, unit commitment, economic dispatch, and optimal decision-making [18, 19].

1.1.2 Hierarchical Forecasting

Wind power forecasts are often produced in a hierarchical fashion, with different models predicting different levels of the hierarchy. For instance, one model predicts the output of individual wind farms, and another model predicts the fleet's total output. Forecast reconciliation [20, 21] can then be used to ensure consistency of hierarchical wind forecasts [22, 23, 24, 25, 26]. Zhang et al. [22] compare the performance of several reconciliation approaches for very short-term ($\leq$1 hour) predictions. Bai et al [23] propose a distributed algorithm to solve the reconciliation problem, and an online reconciliation approach is presented in [24]. An end-to-end learning approach for hierarchical wind power forecast is introduced in [25] with a special focus on handling missing values. Hansen et al. [26] find that hierarchical reconciliation leads to improved day-ahead wind forecasts, especially for fleet-level predictions. In all these works [22, 23, 24, 25, 26], the hierarchical structure is fixed and given a priori. In contrast, the BPR framework proposed in chapter chapter 2 improves the forecast accuracy by *learning a hierarchical structure through asset bundling.*

Asset bundling is a popular technique in portfolio optimization, tracing back to the mean variance model of Markowitz [27]. In the context of power systems, a similar approach is known as the geographical smoothing effect, wherein the aggregated output of multiple wind farms exhibits lower variability [28, 29, 30, 31]. The geographical smoothing effect has been employed in prior literature on optimal design, planning, and re-powering of wind farms [32, 33, 34, 35, 36, 37]. References [38, 39, 40, 41, 42] use a similar approach, but consider a conditional value at risk (CVaR) objective. *Unlike the above portfolio selection-based approaches, chapter 2 proposes to use asset bundling to improve the quality of wind power forecasts.*

1.1.3 Scenario Generation in RES

Scenario generation is power systems is the process of creating diverse and possible future states of the system, accounting for uncertainties caused by different factors such as temperature and other weather-related variable fluctuations, which affects the power consumption and generation within the RES. Various methods have been proposed for scenario generation in power systems, including sampling-based approaches (e.g., Monte Carlo sampling, Latin Hypercube Sampling) and optimization-based methods (e.g., moment matching, distance matching). While these methods are foundational, they often require extensive datasets and are computationally demanding, particularly for generating high-dimensional scenarios [43]. Moreover, when these methods are applied to individual dimensions (i.e., marginal distributions), they struggle to capture the critical spatio-temporal correlations present in the system.

Recent advancements in generative adversarial networks (GANs) have been introduced in power systems scenario generation, especially wind scenario generation in some recent studies [44, 45, 46]. This method has demonstrated its utility in capturing the dynamic and intricate patterns inherent in RES, thereby providing a robust framework for the simulation of realistic and varied energy production scenarios. Dong et al. [45] applied data-driven GAN model on wind and solar scenario generation, however, the model does not consider the weather information data, which is necessary, to learn the wind and solar power data distribution. Zhang et al. [44] consider a point-prediction forecasting model and use GAN to learn the distribution of the forecast errors, and then generate scenarios by generating the residuals and adding to the point forecasts. Thus, the model performance depends highly on the accuracy of the point-forecast model. Chen et al. [47] uses GAN for solar and wind scenario generation. Their proposed model can be conditioned on certain weather events, but may result in poor accuracy without using weather time series such as wind speed and solar irradiation. In addition to these, training GANs can become very challenging, specially when dealing with high-dimensional forecasting problems in RES.

The integration of the Copula method with time series forecasting presents a promising solution to the aforementioned challenges by facilitating the modeling of the joint distribution. For example, the DeepVAR model [48] utilizes a Gaussian Copula to model the possibly time-varying mutual-dependence structure, incorporating a low-rank structure on the Gaussian Copula covariance to enhance computational efficiency. However, DeepVAR assumes stationarity in the time series, an assumption often violated in the context of RES, particularly in wind forecasting. An alternative strategy involves using Copula methods as a post-processing step to combine predicted marginal distributions, where the covariance matrix of the Gaussian Copula captures temporal dependencies [49]. The application of copulas effectively bridges the gap between probabilistic forecasting models and the generation of realistic spatio-temporal scenarios. Gaussian Copulas, in particular, have been extensively explored in the realm of wind power scenario generation, employing an exponential covariance function to address situations with limited historical observations [50, 51]. Additionally, recursive methods have been employed in wind power to refine scenario generation processes [52].

Recent studies, such as [53], have utilized multivariate time series prediction approaches for forecasting load, wind, solar power, and price in power systems. However, these studies typically focus on zonal-level forecasting, which diverges from the generator-level analysis prevalent in most power flow problems. Furthermore, these approaches often overlook the correlations between different forecasting dimensions, a critical factor for effective scenario generation. The integration of weather forecast covariates, which can significantly influence forecasting accuracy, is also absent in their model. Another limitation of the previous works is that they have been focused on small-scale problems and data. chapter 3 aims to address these issues through combination of Copula method with multivariate forecasting approaches on a large-scale problem.

1.1.4 LLMs for Time Series Forecasting

Recent research shows promise in using large language models (LLMs) for time series forecasting [54, 55, 56, 57]. While traditional models excel at incorporating numerical features, they often struggle to utilize unstructured text data that could provide valuable context. LLMs offer a solution through their ability to process and understand textual information.

Most existing work focuses on feeding time series data directly into LLMs through tokenization [58, 59] or specialized prompting [57, 55]. However, using LLMs to integrate descriptive textual features with time series remains largely unexplored. This gap is particularly relevant in retail forecasting, where product descriptions and attributes could help anticipate seasonal demand patterns. Recent advances in LLM fine-tuning techniques, particularly Low-Rank Adaptation (LoRA) [60], have made it more feasible to adapt these models for specialized tasks like time series forecasting while maintaining their text understanding capabilities. This creates new opportunities for combining the strengths of traditional forecasting models with LLMs' natural language processing abilities. This addition provides necessary background while maintaining conciseness and connecting to the thesis contributions in chapter 4.

CHAPTER 2
ASSET BUNDLING FOR HIERARCHICAL FORECASTING OF WIND POWER GENERATION[1]

2.1 Introduction

The sustained growth of wind power generation in US power grids is causing operational and reliability challenges for Transmission System Operators (TSOs). In particular, the intrinsic variability and intermittency of wind resources creates significant operational uncertainty, which must be managed to ensure safe and reliable operations. This, in turn, requires accurate wind power forecasts at various temporal and geographical scales. Indeed, market participants use wind power forecasts at the asset level to inform their participation in day-ahead and real-time markets, while TSOs primarily use aggregated, fleet-wide forecasts to operate the grid. Forecasts must be accurate at both levels while remaining consistent, i.e., the sum of individual forecasts should be equal to the forecast for the overall fleet. As a result, this chapter considers the task of producing high-dimensional, short-term and day-ahead, wind power forecasts that satisfy three requirements:

1. the forecasts predict the output of individual asset and the total output of the fleet;
2. the forecasts are consistent, i.e., the total forecast is the sum of the individual forecasts;
3. the forecasts are accurate both at the asset level and at the fleet level.

To carry this task, the chapter proposes a novel hierarchical approach, the Bundle-Predict-Reconcile (BPR) framework, which (1) learns an additional forecast level (the bundles)

[1]This chapter is based on the following publication: Zhang, Hanyu, Mathieu Tanneau, Chaofan Huang, V. Roshan Joseph, Shangkun Wang, and Pascal Van Hentenryck. "Asset bundling for hierarchical forecasting of wind power generation." *Electric Power Systems Research* 235 (2024): 110771.

between the assets and the fleet, (2) predicts outputs at the asset, bundle, and fleet levels, and (3) reconciles all levels for ensuring consistency. The rest of this section surveys the relevant literature and presents the chapter's contributions.

2.1.1 Contributions

The BPR framework is a three step-approach that first learns a prediction hierarchy by bunding assets, before forecasting time series at the asset, bundle, and fleet levels, before reconciling the forecasts at all levels. A key innovation of the BPR framework is its ability *to learn the "best" bundles*. In other words, BPR is not given a forecasting hierarchy; rather it learns a hierarchy by choosing the bundles that will help improve the asset and fleet forecasts. As a result, BPR can be viewed as a multi-task learning approach: it introduces an auxiliary learning task, i.e., forecasting the bundles, to help the main learning tasks. A second innovation of BPR is the criteria used to select the bundles, leveraging the spatio-temporal correlations that exist in time series of wind power.

The paper presents a comprehensive evaluation of BPR that considers both short-term and day-ahead predictions, includes weather forecasts as covariates, and evaluates a large variety of forecasting models for the individual prediction tasks. Experimental results for industry-size test cases from NREL [61] clearly show the benefits of BPR. They also give significant insights on the impact of bundling and ML architecture on the quality of the forecasts. The learning architectures include recurrent neural networks, transformer models, temporal convolution transformers, and temporal fusion transformers. The bundling criteria include geographical distance, seasonal-adjusted variance, and the intermittency index.

In summary, the paper makes three main contributions:

1. It introduces a new approach, Bundle-Predict-Reconcile (BPR), that learn a forecasting hierarchy to improve the prediction accuracy at the asset and fleet level.

2. It proposes new bundling criteria that exploit the spatio-temporal structure of the time

series for wind power.

3. It evaluates BPR on several forecasting tasks (short-term and day-ahead), numerous learning architectures, and covariate configurations, demonstrating its state-of-the-art performance for both short-term and day-ahead wind power forecasts. In particular, BPR improves fleet-level accuracy by about 25% over a strong baseline in short-term forecasting and about 10% over the best models in day-ahead forecasting.

The rest of the paper is organized as follows. section 2.2 presents the problem formulation and summarizes the proposed approach. section 2.3 presents the proposed asset-bundling methodology. Section section 2.5 describes the hierarchical forecasting and reconciliation framework, and the ML architectures used in the paper. section 2.6 presents numerical experiments, and section 2.7 concludes the paper.

2.2 Problem Formulation

Let $\mathcal{N} = \{1, ..., N\}$ denote the set of wind farms in the system. The output of wind farm $i \in \{1, ..., N\}$ at time $t \in \mathbb{Z}$ is denoted by $\mathbf{x}_{i,t}$. The following notations are used in the paper for $i \in \mathcal{N}$, $t \in \mathbb{Z}$, and $\mathcal{T} \subseteq \mathbb{Z}$:

$$\mathbf{x}_t = (x_{i,t})_{i \in \mathcal{N}} \in \mathbb{R}^N, \qquad \mathbf{x}_{\mathcal{T}} = (\mathbf{x}_\tau)_{\tau \in \mathcal{T}} \in \mathbb{R}^{N \times |\mathcal{T}|}, \qquad \mathbf{x}_{i,\mathcal{T}} = (x_{i,\tau})_{\tau \in \mathcal{T}} \in \mathbb{R}^{|\mathcal{T}|}$$

The paper takes the perspective of a system operator that has full visibility over the output of every wind farm in the system, thus eliminating data privacy issues. Note that BPR may be extended to settings where asset-level data must be kept private via, e.g., separate asset-level models, federated learning to train a joint asset-level forecasting model [62, 63], and distributed forecast reconciliation [23].

2.2.1 Wind Power Forecasting

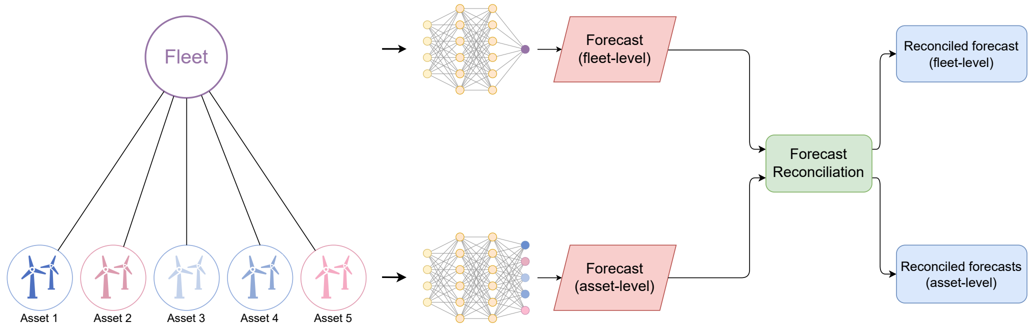
Let $t \in \mathbb{Z}$ denote the current time period, and denote by $\mathcal{H} = \{t-H+1, ..., t\}$ and $\mathcal{T} = \{t+1, ..., t+T\}$ the *historical* and *forecast* windows, respectively. Given past observations $\mathbf{x}_{\mathcal{H}}$, the goal of wind power forecasting is to produce a forecast $\hat{\mathbf{x}}_{\mathcal{T}}$ that is as close as possible to the actual (future) realization $\mathbf{x}_{\mathcal{T}}$. While the paper considers point forecasts for ease of presentation, the BPR methodology naturally extend to probabilistic forecasting.

TSOs and market participants use wind power forecasts as input for market-clearing and to conduct reliability studies. In that context, it is important to note that spatial correlations (between wind farms) and temporal correlations (between time periods) have an impact on congestion management and flexibility (ramping) requirements. Therefore, forecasts should capture spatio-temporal dynamics, which is achieved by jointly predicting all $N \times T$ components of $\hat{\mathbf{x}}_{\mathcal{T}}$. The paper considers joint forecasts for hundreds wind farms and time horizons containing 24 to 48 time steps, giving prediction tasks with over 10,000 output dimensions. Such large output dimensions create scalability challenges and have an adverse impact on forecast accuracy.

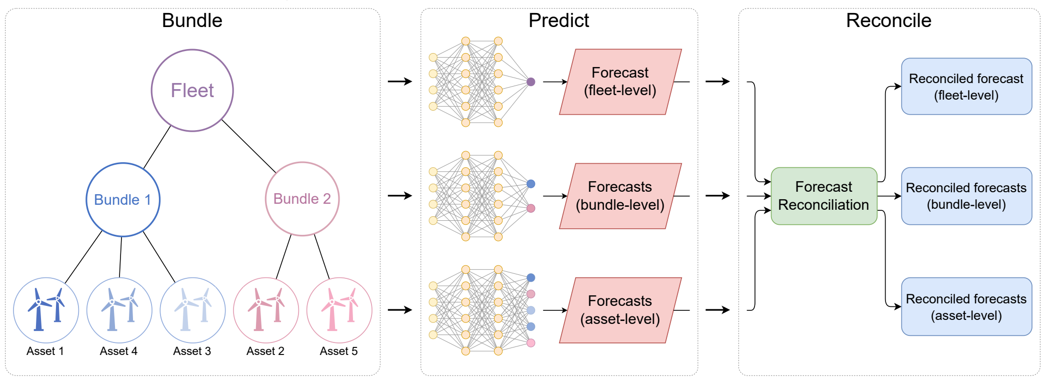
2.2.2 Overview of BPR

Figure 2.1 sketches the BPR framework. A baseline approach to wind power forecasting is illustrated in Figure 2.1a: it trains a model to predict asset-level and fleet-level generations; a two-level reconciliation step ensures consistency. BPR is depicted in Figure 2.1b. First, assets, (i.e., wind farms), are grouped into K bundles based on spatial and temporal features (see Section section 2.3). Second, machine-learning models are trained to deliver forecasts at the asset, bundle, and fleet levels. Third, a hierarchical reconciliation step produces consistent forecasts.

While the presentation of BPR in this chapter considers a 3-level hierarchy (asset, bundle, fleet), the proposed methodology directly extends to any hierarchy. The main difference between BPR and existing hierarchical forecasting approaches is the initial bundling



(a) Baseline hierarchical wind power forecasting. Left: the initial two-level (asset, fleet) hierarchy; each asset is an individual wind farm. Middle: two models are trained for the asset and fleet levels. Right: consistent forecasts are obtained via reconciliation.



(b) The proposed Bundle-Predict-Reconcile (BPR) framework. Left: assets are grouped into bundles. Middle: a model is trained for each level (asset, bundle, fleet) of the hierarchy. Right: consistent forecasts are obtained via reconciliation.

Figure 2.1: Illustration of the BPR Framework.

step that *learns* a hierarchy based on spatio-temporal features, so as to improve forecast accuracy.

2.3 Asset Bundling

This section describes the asset-bundling methodology of BPR, from its problem formulation to its solution algorithm. The presentation assumes a set of N wind farms with given location, nameplate capacity, and historical output $\mathbf{x}_{\mathcal{T}}$ where $\mathcal{T} = \{1, ..., T\}$. The distance between wind farms i and j is denoted by D_{ij}.

2.3.1 Problem Formulation

The asset bundling problem consists in grouping N wind farms into $K \in \{1, .., N\}$ groups, referred to as *bundles*, so as to optimize a certain criterion. Setting $K = 1$ aggregates all wind farms into a single bundle, whereas $K = N$ considers all wind farms individually. The asset-bundling problem can be modeled as a combinatorial optimization problem.

Model 1 The Asset-Bundling Problem

Input: $\mathbf{x}_{\mathcal{T}}$, D, $\bar{D}$
Variables: $\lambda_{k,i}$

$$\min_{\lambda} \quad f(\mathbf{z}_{\mathcal{T}}, \Lambda) \tag{2.1a}$$

$$\text{s.t.} \quad \mathbf{z}_{\mathcal{T}} = \Lambda \mathbf{x}_{\mathcal{T}} \tag{2.1b}$$

$$\sum_{i \in \mathcal{N}} \lambda_{k,i} \geq 1 \qquad \forall k \in \mathcal{K} \tag{2.1c}$$

$$\sum_{k=1}^{K} \lambda_{k,i} = 1 \qquad \forall i \in \mathcal{N} \tag{2.1d}$$

$$D_{ij}(\lambda_{k,i} + \lambda_{k,j}) \leq 2\bar{D} \qquad \forall (i,j) \in \mathcal{N}, k \in \mathcal{K} \tag{2.1e}$$

$$\Lambda \in \{0,1\}^{K \times N} \tag{2.1f}$$

Model 1 presents an optimization model for the asset-bundling problem and a criterion f. Binary variable $\lambda_{k,i}$ takes value 1 if wind farm i belongs to bundle k. Constraints (Equation 2.1b) capture the bundled time series $\mathbf{z}_{\mathcal{T}} \in \mathbb{R}^{K \times T}$. Namely, the output of bundle k at time t is $\mathbf{z}_{k,t} = \sum_i \lambda_{k,i} \mathbf{x}_{i,t}$, i.e., it is the combined output of all wind farms assigned to bundle k. Constraints (Equation 2.1c) and (Equation 2.1d) ensure that each bundle is non-empty and that each wind farm is assigned to exactly one bundle. Constraints (Equation 2.1e) enforce that two wind farms cannot be assigned to the same bundle if the distance between them exceeds a maximum diameter $\bar{D}$. Finally, the objective (Equation 2.1a) captures properties of the bundled time series $\mathbf{z}_{\mathcal{T}}$ and of the bundling assignment Λ; possible choices for the objective function f are described next.

2.3.2 Bundling Criteria

Variance

A natural bundling criterion is to minimize the total variance of the bundles, i.e.,

$$f(\mathbf{z}_{\mathcal{T}}, \Lambda) = \sum_{k=1}^{K} \mathrm{Var}(\mathbf{z}_{k,\mathcal{T}}) = \mathrm{tr}(\Lambda \boldsymbol{\Sigma} \Lambda^{\top}), \tag{2.2}$$

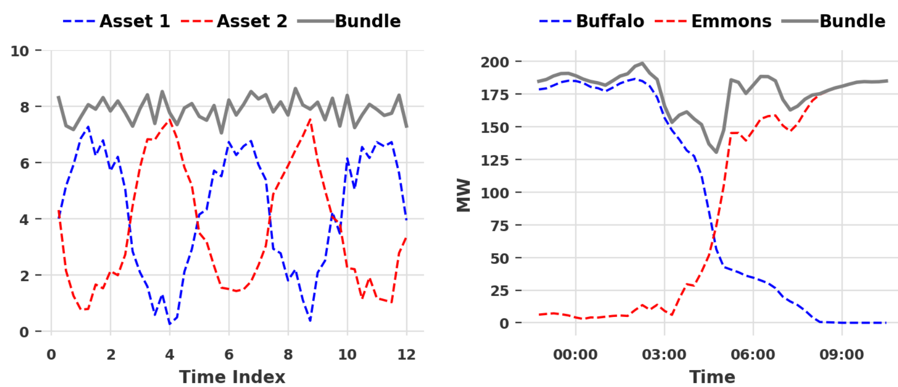


Figure 2.2: Illustration of variance-based bundling.
Left: synthetic example. Right: real example with two MISO wind farms from [61]. In each case, the two assets are negatively correlated, and bundling yields a reduction in variance.

where $\boldsymbol{\Sigma} \in \mathbb{R}^{N \times N}$ is the empirical covariance matrix of $\mathbf{x}_{\mathcal{T}}$, tr stands for the trace of a matrix, and $\top$ represents the matrix (or vector) transpose. Variance-based bundling is illustrated in Figure 2.2. Note that bundling reduces the total variance only if negatively correlated assets are bundled together. Therefore, a key factor for its success is the proportion of negative correlations among the original assets.

Seasonal-Adjusted Variance

A large proportion of wind farms may exhibit positive correlations over a given study period due to, e.g., common daily patterns. This is detrimental to variance-based bundling, which relies on negative correlations. To mitigate this effect, the seasonal-adjusted variance is defined as$\mathrm{SAVar}(\mathbf{x}_{\mathcal{T}}) = \sum_{i \in \mathcal{N}} \mathrm{Var}(\mathbf{x}_{i,\mathcal{T}} - \boldsymbol{\mu}_{\mathcal{T}})$, where the average $\boldsymbol{\mu}_{\mathcal{T}} = \frac{1}{N} \sum_{i \in \mathcal{N}} \mathbf{x}_{i,\mathcal{T}}$

captures the seasonal trend. This yields the objective function

$$f(\mathbf{z}_{\mathcal{T}}, \Lambda) = \mathrm{SAVar}(\mathbf{z}_{\mathcal{T}}) = \mathrm{tr}(\Lambda \tilde{\mathbf{\Sigma}} \Lambda^\top), \tag{2.3}$$

where $\tilde{\mathbf{\Sigma}}$ is the covariance matrix of the seasonal-adjusted time series $\tilde{\mathbf{x}}_{\mathcal{T}} = \mathbf{x}_{\mathcal{T}} - \boldsymbol{\mu}_{\mathcal{T}}$. Seasonal-adjusted variance-based bundling promotes bundles that behave similarly to the mean of the assets, which is expected to improve the quality of the forecast when the mean exhibits simple patterns.

Intermittency Index

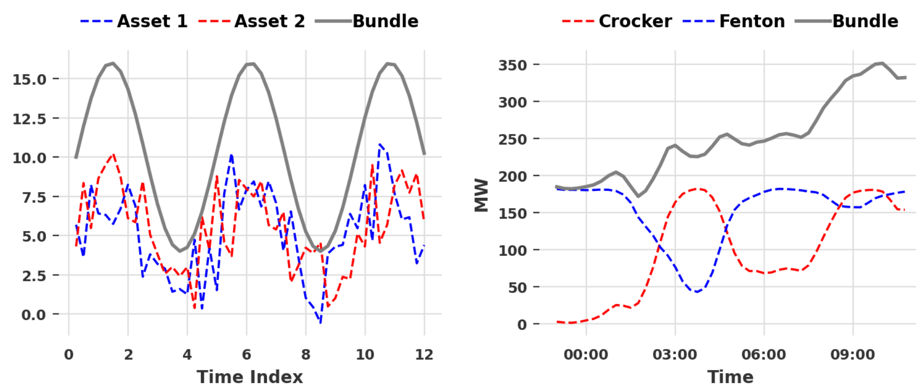


Figure 2.3: Illustration of intermittency index-based bundling.
Left: synthetic example. Right: real example with two MISO wind farms from [61]. In each case, bundling yields a reduction in intermittency: the bundled time series exhibits fewer high-frequency oscillations and clearer long-term patterns.

The *intermittency index* of $\mathbf{x}_{\mathcal{T}}$ is

$$\mathrm{Imcy}(\mathbf{x}_{\mathcal{T}}) = \sum_{i \in \mathcal{N}} \mathrm{Var}(\dot{\mathbf{x}}_{i,\mathcal{T}}),$$

where $\dot{\mathbf{x}}_{i,t} = \mathbf{x}_{i,t} - \mathbf{x}_{i,t-1}$; $\dot{\mathbf{z}}$ is defined similarly. Substituting $\dot{\mathbf{z}} = \Lambda \dot{\mathbf{x}}$, the objective function (Equation 2.1a) becomes

$$f(\mathbf{z}_{\mathcal{T}}, \Lambda) = \mathrm{Imcy}(\mathbf{z}_{\mathcal{T}}) = \mathrm{tr}(\Lambda \dot{\mathbf{\Sigma}} \Lambda^{\top}), \tag{2.4}$$

where $\dot{\mathbf{\Sigma}}$ is the covariance matrix of $\dot{\mathbf{x}}_{\mathcal{T}}$. Intermittency-based bundling is illustrated in Figure 2.3 where the bundled time series exhibit long-term patterns that are easier to learn. In the context of power systems operations, the ability of intermittency-based bundling to identify long-term trends allows to better anticipate and manage ramping needs.

2.4 The Bundling Algorithm

The objective functions in Equation 2.2, Equation 2.3 and Equation 2.4 share the same intrinsic structure, differing only in the choice of the covariance matrix $\mathbf{\Sigma}$, $\tilde{\mathbf{\Sigma}}$, or $\dot{\mathbf{\Sigma}}$. As a result, for each proposed bundling criterion, Problem in Equation 2.1 is a convex mixed-integer quadratic programming (MIQP) problem, which can be solved using MIQP solvers like Gurobi or CPLEX. However, for realistic test cases, the asset-bundling problem is intractable for MIQP solvers, mainly due to its intrinsic complexity and the symmetry of set-partitioning constraints (Equation 2.1d). The latter are notoriously detrimental to the performance of MIQP solvers.

To remedy these scalability issues, the BPR framework uses a fast greedy algorithm presented in Algorithm 1. For simplicity, Algorithm 1 is presented for the case of Variance-based bundling. The algorithm takes as input the time series $\mathbf{x}$ and the number of bundles K, and initializes a set of N bundles, each corresponding to an individual wind farm. Then, at each step, the algorithm aggregates the two most negatively correlated bundles, subject to the maximum diameter constraint. Algorithm 1 is computationally fast, and was found to produce close-to-optimal solutions in early experiments.

Algorithm 1 The Greedy Asset-Bundling Algorithm

Input: Time series $\mathbf{x}_{\mathcal{T}}$, distance matrix D, maximum diameter $\bar{D}$, number of bundles K
Initialize: $\mathcal{K} \leftarrow \{\{1\}, \dots, \{N\}\}$
while $|\mathcal{K}| > K$ **do**
  for $k = 1$ to $|\mathcal{K}|$ **do**
    $\mathbf{z}_k \leftarrow \sum_{i \in \mathcal{K}_k} \mathbf{x}_{i,\mathcal{T}}$
  end for
  for $k = 1$ to $|\mathcal{K}|$ **do**
    for $l = k + 1$ to $|\mathcal{K}|$ **do**
      $\tilde{D}_{k,l} \leftarrow \max\{D_{ij} \mid i \in \mathcal{K}_k,\ j \in \mathcal{K}_l\}$
    end for
  end for
  $(k^*, l^*) \leftarrow \underset{\tilde{D}_{k,l} \leq \bar{D}}{\arg\min}\ \mathrm{Cov}(\mathbf{z}_k, \mathbf{z}_l)$
  $\mathcal{K} \leftarrow (\mathcal{K} \setminus \{\mathcal{K}_{k^*}, \mathcal{K}_{l^*}\}) \cup \{\mathcal{K}_{k^*} \cup \mathcal{K}_{l^*}\}$
end while
Return set of bundles $\mathcal{K}$

2.5 Forecasting

This section presents the hierarchical forecasting and reconciliation component of BPR, the architecture of the ML models considered in the experiments, and the metrics for evaluating the quality of the forecasts. The section assumes a fixed bundling of N wind farms into K bundles, denoted by $\Lambda \in \{0,1\}^{K \times N}$. For $t \in \mathbb{Z}$, $\mathbf{z}_t = \Lambda \mathbf{x}_t \in \mathbb{R}^K$ and $\mathbf{X}_t = e^\top \mathbf{x}_t \in \mathbb{R}$ denote the bundle-level and total-level wind power output, respectively. The forecast horizon at time t is $\mathcal{T}_t = \{t+1, ..., t+T\}$.

2.5.1 Reconciliation

BPR works on a three-level hierarchical structure illustrated in Figure 2.1b: asset-level ($\mathbf{x}$), bundle-level ($\mathbf{z}$), and total-level ($\mathbf{X}$). The paper leverages this structure by training three models that predict each level of this hierarchy: at time t, three forecasts $\hat{\mathbf{x}}_{\mathcal{T}_t}, \hat{\mathbf{z}}_{\mathcal{T}_t}, \hat{\mathbf{X}}_{\mathcal{T}_t}$ are produced that predict the future asset-level, bundle-level, and total-level time series $\mathbf{x}_{\mathcal{T}_t}, \mathbf{z}_{\mathcal{T}_t}, \mathbf{X}_{\mathcal{T}_t}$, respectively.

The reconciliation aims at making these predictions consistent. Given forecasts $\hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\mathbf{X}}$

at the asset, bundle, and fleet levels, reconciliation produces new forecasts $\tilde{\mathbf{x}}, \tilde{\mathbf{z}}, \tilde{\mathbf{X}}$ such that

$$\forall k, \quad \tilde{\mathbf{z}}_{k,\mathcal{T}_t} = \sum_{i \in \mathcal{N}} \lambda_{k,i} \tilde{\mathbf{x}}_{i,\mathcal{T}_t}, \qquad \qquad \tilde{\mathbf{X}}_{\mathcal{T}} = \sum_{i \in \mathcal{N}} \tilde{\mathbf{x}}_{i,\mathcal{T}} = \sum_{k} \tilde{\mathbf{z}}_{k,\mathcal{T}}$$

More formally, forecast reconciliation [20] takes as input incoherent forecasts, and produces the most accurate consistent forecasts. Let $\hat{\mathbf{h}} = (\hat{\mathbf{X}}, \hat{\mathbf{z}}, \hat{\mathbf{x}})$ and S be the summing matrix $S = \begin{bmatrix} e \Lambda^\top I \end{bmatrix}^\top \in \mathbb{R}^{(N+K+1)\times N}$. Reconciliation searches for a matrix $G \in \mathbb{R}^{N\times(N+K+1)}$ such that $\tilde{\mathbf{h}}_\tau = SG\hat{\mathbf{h}}_\tau$. In addition, to improve accuracy further, BPR computes T matrices $G_1, ..., G_T$ such that $\forall \tau \in \{1, ..., T\}$, $\tilde{\mathbf{h}}_\tau = SG_\tau \hat{\mathbf{h}}_\tau$ to capture the variances at different lead times τ. BPR uses the MinT optimal reconciliation approach from [21], with a weighted least squares (WLS) estimator using variance scaling. Thereby, each G_τ has the form $G_\tau = (S^\top W_\tau^{-1} S)^{-1} S^\top W_\tau^{-1}$, where $W_\tau = \text{Diag}(\hat{W}_\tau)$, $\hat{W}_\tau = \frac{1}{M} \sum_{t \in \mathcal{M}} \epsilon_{t,\tau} \epsilon_{t,\tau}^\top$ and $\epsilon_{t,\tau} = (\hat{\mathbf{h}}_{t,\tau} - \mathbf{h}_{t,\tau})$. For the case at hand,WLS using variance scaling used in [64] and [21] achieved better performance than other MinT-based reconciliation approaches such as Ordinary least squres in [20], MinT(sample) [21], and WLS with structure scaling in [64]. Therefore, only WLS using variance scaling is reported in section 2.6.

2.5.2 Forecasting Models

BPR was instantiated with various learning architectures for the experimental evaluation. If a model supports future-known covariates, a weather forecasts are given to the model for 48 hours prediction tasks (see [65] for model's specification on types of input).

The co-variates may include static information about each wind farm (e.g., its location and model), temporal information such as hour of the day or month of the year, and past and forecasted weather-related information such as temperature, humidity, wind speed and direction. Covariates have been shown to improve forecast accuracy [66, 67].

- **Persistence Model**: The Persistence model (`Per`) is a naive baseline whose prediction for the entire forecasting horizon is equal to the last observed value.
- **Linear Model**: The linear regression model is trained using ridge regression. Without loss of generality, a separate model is trained for each bundle, which makes training more efficient.

- **Recurrent Neural Network**: The Recurrent Neural Network (RNN) architecture follows the state-of-the-art DeepAR model [6]. It is applied in an auto-regressive way to compute multi-step predictions. The RNN models are trained with mean squared error (MSE) loss.
- **Transformer models** The transformer model follows the approach of [68]. Namely, it employs a multi-head attention mechanism inside an encoder-decoder, sequence-to-sequence architecture.
- **Temporal Convolution Transformer (TCN)** The TCN architecture follows the model from [69]. Unlike the baseline transformer model, TCN uses stacked dilated causal convolutional neural networks to extract features from historical observations.
- **Temporal Fusion Transformer (TFT)** The TFT architecture [70] is a deep learning architecture designed for multi-horizon forecasting tasks with a mixed-type inputs, e.g., past observations and past, future, and static covariates. This allows the integration of weather-based, wind speed forecasts into the model, which improves longer-term predictions. The TFT architecture uses a combination of recurrent layers and self-attention layers to learn temporal relationships at different scales and provide interpretable insights into the temporal dynamics. The model utilizes specialized components to select relevant features, and gating layers to suppress unnecessary components.

2.5.3 Evaluation Metrics

The experimental evaluation uses various metrics to evaluate the quality of wind power forecasts. Each metric is presented for asset-level times series $\mathbf{x}$ in what follows, but they are similar for the bundle and fleet levels. See [71] for a detailed discussion of wind power forecast evaluation.

Normalized Mean Absolute Error

The Normalized Mean Absolute Error (NMAE) is defined as

$$\text{NMAE} = \frac{1}{MNT} \sum_{t \in \mathcal{M}} \sum_{i \in \mathcal{N}} \frac{\|\mathbf{x}_{i,\mathcal{T}_t} - \hat{\mathbf{x}}_{i,\mathcal{T}_t}\|_1}{\bar{\mathbf{x}}_i} \tag{2.5}$$

where $\bar{\mathbf{x}}_i$ is the maximum capacity of wind farm i and $\|\cdot\|_1$ is the vector ℓ_1 norm.

Root-mean-square Error

The Root-mean-square Error (RMSE) is expressed as

$$\text{RMSE} = \sqrt{\frac{1}{MTN} \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{M}} \|\mathbf{x}_{\mathcal{T}_t} - \hat{\mathbf{x}}_{\mathcal{T}_t}\|_2^2} \tag{2.6}$$

where $\|\cdot\|_2$ is the vector ℓ_2 norm. RMSE better captures large prediction errors than NMAE.

Variogram Score

The Variogram Score (VS) of order $p > 0$ is given by

$$\text{VS}_p = \frac{1}{M} \sum_{t \in \mathcal{M}} \sum_{i,j \in \mathcal{N}} \sum_{\tau,\tau' \in \mathcal{T}_t} \left(\mathbf{\Delta x}^p_{i,j,\tau,\tau'} - \mathbf{\Delta \hat{x}}^p_{i,j,\tau,\tau'} \right)^2 \tag{2.7}$$

where $\mathbf{\Delta x}^p_{i,j,\tau,\tau'}$ and $\mathbf{\Delta \hat{x}}^p_{i,j,\tau,\tau'}$ are defined as

$$\mathbf{\Delta x}^p_{i,j,\tau,\tau'} = |\mathbf{x}_{i,\tau} - \mathbf{x}_{j,\tau'}|^p, \qquad \mathbf{\Delta \hat{x}}^p_{i,j,\tau,\tau'} = |\hat{\mathbf{x}}_{i,\tau} - \hat{\mathbf{x}}_{j,\tau'}|^p.$$

The paper uses $p = 1/2$, as recommended in [72]. Note that computing VS_p requires $\mathcal{O}(MN^2T^2)$ operations, which is computationally expensive for the case at hand. Therefore, the paper only reports VS for total wind predictions, which reduces the complexity to $\mathcal{O}(MT^2)$.

Energy Distance

The Energy Distance (ED) is

$$\text{ED} = \frac{2}{M} \sum_{t \in \mathcal{M}} \|\mathbf{x}_{\mathcal{T}_t} - \hat{\mathbf{x}}_{\mathcal{T}_t}\|_F \tag{2.8}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

2.6 Experimental Evaluation

2.6.1 Data

The results are presented on the NREL dataset [61] that contains 283 time series of existing wind farms in MISO system for 2018 and 2019 at different time granularities (1 hour and 15 minutes). This dataset is complemented by a separate dataset of weather forecasts from the European Center for Medium-Range Weather Forecasts (ECMWF) [73]. These forecasts are collected on a 1-degree by 1-degree grid for the MISO region and used as covariates in the learning models. They are generated twice a day, at noon and midnight in UTC time, and have a 6-hour granularity. The weather forecasts time series are interpolated from 6-hour granularity down to the desired granularity of the forecasts.

2.6.2 Experimental Setting

The experiments consider both short-term and day-ahead forecasting tasks. Short-term forecasts predict wind power generation over the next 6 hours with a granularity of 15 minutes; it is the forecasting task needed for the Look-Ahead Commitment (LAC) in the MISO market clearing pipeline. Short-term forecasts are purely data-driven, because the original ECMWF weather forecasts have a granularity of 6 hours and are produced only twice a day. Day-ahead forecasts predict wind power generation over the next 48 hours with a granularity of 1 hour; it is the forecasting task needed for the day-ahead reliability commitment (FRAC) in the MISO market clearing pipeline. The day-ahead forecasting task is data-driven as well but it also uses co-variates given by the latest available ECMWF weather forecasts.

The evaluation uses a test set comprising five weeks in 2019, i.e., the weeks of January 8th, March 8th, May 8th, July 8th, and September 8th. For short-term predictions, the training dataset consists of 90 days of historical data immediately preceding each test week. The training for day-ahead forecasting uses a more extensive dataset spanning a year. This historical data is used to compute the bundles, train the forecasting models, and estimate the reconciliation matrices (see subsection 2.5.1). Note that, in practice, BPR can be executed on a rolling basis, e.g., every few weeks, effectively re-computing bundles and re-training models using the most recent information. This allows to capture changes in the underlying dynamics, and incorporate new assets as they are added to the system.

The evaluation uses two bundle sizes: 10 and 50 bundles. The diameter constraints are set to 500 kilometers for 50 bundles and 800 kilometers for 10 bundles. The evaluation trains three types of models for (i) fleet level ($K = 1$), (ii) bundle level ($K = 10$ and $K = 50$), and (iii) asset level ($K = 283$). These parameters were determined based on preliminary experiments, and allow to assess the impact of the number of bundles (K) on overall performance. K-means clustering is used as a baseline, corresponding to existing forecasting methodologies such as [25, 26]. Recall that the proposed BPR can support an arbitrary forecasting hierarchy, and an arbitrary number of bundles. Finally, forecast reconciliation is performed using MinT [74] with WLS scaling as

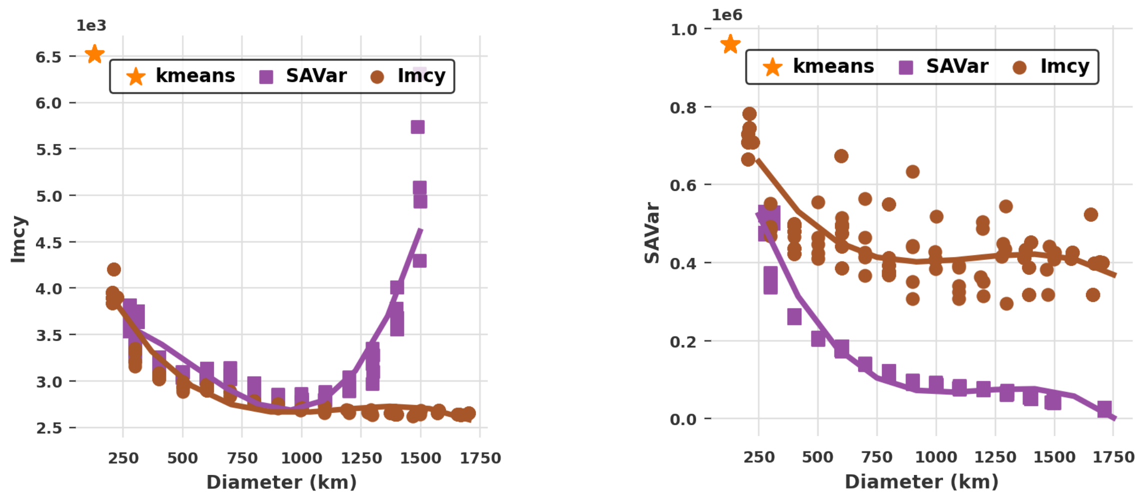


Figure 2.4: Sensitivity of Imcy and SAVar with respect to diameter for three methods when considering 50 bundles.

A cubic trend line is fitted for better visualization. Different bundles are obtained by varying the distance cutoff $\bar{D}$ in Algorithm 1. A single point is displayed for K-means ($\star$) because it does not support a distance cutoff.

Figure 2.4 depicts the tradeoffs between the diameter constraints for the bundles and the quality of the bundling criteria (Imcy and SAVar). Recall that the baseline K-means clustering groups assets purely based on their pair-wise distances, i.e., it does not take any temporal information into account. Therefore, Figure 2.4 displays a single point for the K-means bundling algorithm, which exhibits low diameter by design (since nearby assets are bundled together), but high variance and intermittency index due to the lack of temporal information in K-means. Figure 2.4 shows that it is possible to obtain low values for the intermittency and the SAVar criteria with a reasonable diameter constraint. This informed the selection of the diameter constraints specified earlier.

2.6.3 Prediction Performance

Preliminary experiments, whose results are not reported for lack of space, showed that, for the two-level hierarchy (K=1), forecast reconciliation improved overall performance. For instance, reconciliation improved the fleet-level accuracy of the RNN model by roughly 10% for short-term predictions. Accordingly, all results presented next are for reconciled forecasts. Table 2.1 and Table 2.2 present the results obtained by BPR for the short-term and day-ahead forecasting tasks, respectively. All results are averaged over the five selected weeks. For each model architecture, each table reports: the number of bundles (K), the bundling criterion, and performance metrics of fleet-level forecasts (NMAE, RMSE, ED, VS) and asset-level forecasts (NMAE, RMSE, ED). Rows with $K = 1$ correspond to the baseline two-level hierarchy illustrated in Figure 2.1a, i.e., the corresponding forecasts are obtained by reconciling an asset-level and a fleet-level forecasts,without any bundling. A detailed analysis of each forecasting task is presented next.

Short-Term Predictions Observe first that, for short-term forecasting, the persistence model emerges as a strong baseline, achieving a 7.6% NMAE at the fleet level, and a 13% NMAE and a 25.2MW RMSE at the asset level. Only the Ridge regression, transformer, and TFT models are capable of improving over the baseline. For the TFT model with 10 SAVar bundles, BPR achieves a NMAE of 5.7% at the fleet level and a 15.3% NMAE and a 23.5MW RMSE at the asset level. The best results come from reconciling asset level forecasts from the Ridge regression and bundle level forecasts from TFT (`TFT+Lin`, last row of the table). The latter was found to produce the most accurate asset-level forecast, and the former to produce the most accurate bundle-level and fleet-level forecast. In this case, BPR achieves a NMAE of 5.8% at the fleet level and a 18.8% NMAE and a 21.7MW RMSE at the asset level. When correlations are taken into account, as captured by ED and VS metrics, BPR achieves consistently better results, especially for VS at the fleet level. *In summary, BPR improves the accuracy by about 25% (NMAE, RMSE and ED) at the fleet level and by about 14% (RMSE and ED) at the asset level over the baseline, and reduces VS by 75% at the fleet level.*

When looking at a specific model (e.g., TFT), the main benefit of BPR for short-term forecasting is at the fleet level. For instance, BPR on TFT with SAVar and 10 bundles improves the fleet level accuracy by about 10% (NMAE, RMSE and ED) over the base setting ($K = 1$); VS is also improved by 6%. The improvements of BPR at the asset level through reconciliation is about 2% (RMSE), again for TFT with SAVar and 10 bundles. Note also that the bundling criteria achieve rather similar results: SAVar achieves consistently strong results overall, but there is no clear overall winner. Similarly, there is no significant difference between 10 and 50 bundles for short-term forecasting.

Day-Ahead Predictions The persistence model is not relevant for the day-ahead setting, so the discussion focuses on the benefits of BPR for specific models. Nonetheless, the short-term forecasting results provide a good reference point, indicating the high quality of the day-ahead forecasts. For day-ahead forecasting, the `Imcy` criterion dominates the others (almost always) and BPR with 50 bundles is typically superior to BPR with 10 bundles. In addition, BPR always improves against the baseline (K=1), especially when considering ED and VS metrics, which capture correlations. BPR over the TFT model with 50 bundles and the `Imcy` criterion achieves the best results overall. *At the fleet level, BPR yields a 6–8% improvement in NMAE and RMSE, and a 20% improvement in ED and VS*

Table 2.1: BPR for 6 Hours Forecasting: Accuracy Results.

			Fleet				Asset		
Model	K	criterion	NMAE	RMSE	ED	VS	NMAE	RMSE	ED
`Per`	–	–	7.6	2444	18797	631229	13.9	25.3	3911
`Lin`	**1**	–	6.5	2038	16370	171212	14.0	22.4	3524
	10	Imcy	6.2	1958	15637	165624	13.9	22.0	3469
		SAVar	6.1	1955	15631	167449	13.9	22.0	3471
		kmeans	6.2	1959	15662	166057	13.9	22.1	3471
	50	Imcy	6.0	1925	15200	163855	**13.7**	**21.3**	**3355**
		SAVar	6.1	1940	15407	163822	13.8	21.9	3437
		kmeans	6.0	1930	15348	165105	13.7	21.7	3406
`RNN`	**1**	–	8.1	2524	20275	199126	15.6	24.8	3928
	10	Imcy	6.7	2114	16927	175617	15.2	23.5	3752
		SAVar	7.4	2284	18374	191918	15.4	23.9	3807
		kmeans	8.1	2463	20057	196443	15.6	24.6	3904
	50	Imcy	8.6	2549	20919	199591	15.6	24.4	3863
		SAVar	7.1	2199	17534	175830	15.2	23.8	3781
		kmeans	7.7	2387	19316	202821	15.3	24.0	3807
`Tfm`	**1**	–	7.2	2054	17026	173000	15.3	22.7	3615
	10	Imcy	6.4	1878	15348	168573	15.2	22.3	3554
		SAVar	6.5	1893	15508	167283	15.2	22.3	3559
		kmeans	6.8	1956	16121	170853	15.3	22.5	3591
	50	Imcy	6.5	1885	15517	170166	15.2	22.3	3547
		SAVar	6.7	1931	15775	161601	15.2	22.4	3561
		kmeans	6.8	1950	16029	162314	15.2	22.5	3572
`TCN`	**1**	–	10.6	2811	24388	211609	19.3	27.7	4428
	10	Imcy	10.3	2725	23763	202882	19.3	27.4	4390
		SAVar	10.4	2738	23851	209404	19.3	27.5	4405
		kmeans	10.3	2730	23682	209485	19.3	27.4	4397
	50	Imcy	10.0	2672	23182	213046	19.1	27.0	4328
		SAVar	10.1	2691	23375	207217	19.1	27.3	4365
		kmeans	10.2	2735	23552	213180	19.2	27.3	4374
`TFT`	**1**	–	6.4	1899	15432	152863	15.3	23.7	3755
	10	Imcy	5.8	1753	14163	142315	15.3	23.5	3723
		SAVar	**5.7**	**1721**	**13891**	142824	15.3	23.4	3708
		kmeans	6.0	1780	14387	**140360**	15.3	23.5	3725
	50	Imcy	6.0	1778	14472	145387	15.3	23.5	3724
		SAVar	6.1	1797	14689	148588	15.3	23.5	3729
		kmeans	5.9	1746	14256	142929	15.2	23.3	3697
`TFT` + `Lin`	**10**	SAVar	5.8	1806	14574	152841	13.8	21.7	3423

K=1 denotes the baseline two-level hierarchy. All forecasts are reconciled.

Table 2.2: BPR for 48 hours Forecasting: Accuracy Results.

			Fleet				Asset		
Model	K	criterion	NMAE	RMSE	ED	VS	NMAE	RMSE	ED
RNN	**1**	–	10.5	2848	38389	1370109	19.5	28.8	6673
	10	lmcy	9.8	2655	35722	1228135	19.3	28.0	6478
		SAVar	9.8	2662	35818	1242135	19.3	28.1	6514
		kmeans	10.2	2743	36973	1283293	19.3	28.3	6562
	50	lmcy	9.2	2444	33102	1167797	18.9	26.7	6189
		SAVar	10.3	2749	37135	1296811	19.2	28.2	6525
		kmeans	10.1	2720	36611	1331087	19.1	28.1	6516
TFT	**1**	–	8.8	2480	33236	1214637	16.1	25.7	5918
	10	lmcy	8.4	2364	31586	1117189	16.0	25.3	5819
		SAVar	8.6	2391	31915	1139472	16.0	25.5	5867
		kmeans	8.6	2408	32227	1153448	16.0	25.4	5862
	50	lmcy	**8.1**	2324	30835	1099162	15.9	**24.9**	**5726**
		SAVar	8.3	2330	31178	1119034	15.9	25.2	5797
		kmeans	8.2	**2302**	**30615**	**1092823**	**15.8**	25.0	5753

$K=1$ denotes the baseline two-level hierarchy. All forecasts are reconciled.

over the baseline. At the asset level, BPR yields a 3% improvement in RMSE and ED.

2.6.4 Comparison with NREL Forecasts

The original NREL dataset [61] includes intra-day and day-ahead forecasts. To the best of the authors' knowledge, these are produced by NREL using a combination of weather forecasts, physical models [75], and unspecified machine learning models.[2] NREL's intra-day forecasts are produced every 6 hours, with a 6-hour lead time, 6-hour horizon, and hourly granularity. They are therefore not comparable to the short-term forecasts considered in the paper.

NREL's day-ahead forecasts are produced once a day (at midnight UTC), with an 11-hour lead time, 48-hour horizon, and hourly granularity. While not directly comparable to the day-ahead setting considered here, which has no lead time and is updated hourly, the accuracy of NREL's day-ahead forecasts nonetheless provides a sense of the quality of the BPR forecasts. Namely, NREL's day-ahead forecasts achieve, at the fleet level, a NMAE of 10.5%, an RMSE of 2954MW, an ED of 37589, and a VS of 854604 and, at

[2]The forecast methodology used by NREL in [61] has not been published.

the asset level, NMAE of 14.4%, an RMSE of 24.5MW, and an ED of 5469. Overall, the data-driven BPR forecasts compare favorably in terms of NMAE, RMSE and ED at the fleet level, while NREL's asset-level forecasts achieve a slightly lower NMAE and similar RMSE and ED. Recall that NREL's forecasts use advanced physical models [75], which are known to improve long-term forecasts. These results confirm the overall potential of BPR.

2.7 Conclusion

The work has proposed a novel *Bundle-Predict-Reconcile* framework for hierarchical wind power forecasting, that integrates asset bundling, machine learning and forecast reconciliation techniques. In contrast with existing hierarchical forecasting approaches, BPR leverages asset-bundling techniques to identify a hierarchy that yields improved forecast accuracy. The proposed asset-bundling methodology captures spatio-temporal characteristics of the underlying time series and is computationally fast, thus allowing to update bundles and re-train models periodically. The paper has conducted extensive numerical experiments on a large industry-size dataset, thereby evaluating the impact of BPR across different forecasting tasks (short-term and day-ahead), bundling criteria, and machine learning architectures. The results demonstrate that, compared to a baseline hierarchical setting, BPR consistently improves forecast accuracy at the fleet and asset levels, for both short-term and day-ahead forecasts.

CHAPTER 3

BENCHMARKING WEATHER-INFORMED PROBABILISTIC FORECASTING AND SCENARIO GENERATION IN POWER SYSTEMS[1]

3.1 Introduction

The adoption of Renewable Energy Sources (RES), especially wind and solar generation, is at the forefront of modern power systems' transition towards sustainable and carbon-free electricity generation. This transition also promises to improve the affordability and accessibility of energy systems worldwide [76, 77]. However, integrating high levels of RES into existing power grids introduces significant challenges, particularly in forecasting. Indeed, RES, whose output depends on weather conditions, are intrinsically intermittent and stochastic. This operational uncertainty thus complicates grid stability and optimization efforts [76] which, in turn, motivates the need for advanced forecasting tools.

In that context, academics and practitioners have increasingly moved from deterministic to probabilistic forecasting, which better captures the uncertainty in load and RES output. In addition, probabilistic forecasts provide the ability to sample multiple scenarios, which is essential for risk quantification [78] and uncertainty-aware optimization [79]. Despite their popularity, probabilistic forecasts present substantial scalability challenges. Indeed, they must capture spatial (between system elements located at different locations), and temporal (between different time periods) correlations. Typical forecasting tasks comprise hundreds of renewable generators and dozens of time steps (e.g. 48 hours), with output dimensions in tens of thousands, and correlation matrices with hundreds of millions of coefficients.

[1]This chapter is based on the following manuscript currently under revision:: Zhang, Hanyu, Reza Zandehshahvar, Mathieu Tanneau, and Pascal Van Hentenryck. “Weather-informed probabilistic forecasting and scenario generation in power systems.” *arXiv preprint arXiv*:2409.07637 (2024).

To address these challenges, this chapter focuses on the development of high-dimensional probabilistic forecasting methods for load and RES generation. By utilizing weather information as covariates and restoring the spatio-temporal correlations among the prediction variables, this chapter sets the foundation for reliable scenario generation using high-dimensional multivariate time series forecasting in RES.

3.2 Contributions

To address the existing gaps within high-dimensional forecasting in RES, this chapter introduces a comprehensive probabilistic multivariate time series forecasting framework that leverages Gaussian copulas for enhanced forecasting and realistic scenario generation of load, and wind and solar power generation. The proposed model is augmented with both historical and forthcoming weather forecast covariates, aiming at improving forecasting and scenario generation performance across various temporal and spatial dimensions. The principal contributions of this chapter are threefold:

- Integration of multivariate time series forecasting with copula method for high dimensional problem in RES to effectively capture the complex spatial and temporal dependencies. Particularly, the largest problem is a 48-hour ahead prediction for 376 solar farms, with an output dimension of 10,848.
- Showcasing the importance of incorporating weather forecast data, which significantly enhances the accuracy and reliability of forecasting and scenario generation processes, especially for higher prediction lead times.
- Comprehensive benchmarking and comparison of various time series prediction methodologies, including ARIMA, DeepAR, NLinear, DLinear, and TFT, combined with Gaussian copula. Different metrics are considered to evaluate the efficacy of these models in multi-dimensional forecasting and scenario generation within power systems.

The analysis encompasses time series forecasting methods for the prediction and scenario generation of load, and wind and solar power generation within the Midcontinent Independent System Operator (MISO) system. In contrast with previous methods, which are mainly on small and/or low-dimensional datasets, this work considers probabilistic forecasting of a large-scale system with 6 load zones, 286 wind farms and 376 solar farms. The proposed Weather-Informed (WI) TFT model is highlighted for its superior performance across multiple evaluation metrics, in both deterministic forecasting and scenario generation. The findings of this chapter showcase the comparative effectiveness of different forecasting techniques within the context of energy forecasting and uncertainty quantification, thus providing critical insights for risk-aware decision making in power system operations.

3.3 Problem Formulation

This section outlines the problem formulation and introduces the notations used for the probabilistic multivariate time series forecasting of load, and wind and solar power generation, that incorporates weather data.

3.3.1 Problem Setting

Let $Z_{i,t}$ denote a random variable corresponding to the target $i \in \mathcal{D} = \{1, .., D\}$ (i.e., a load, a wind farm, or a solar farm) at time $t \in \mathbb{Z}$, and denote its realization as $z_{i,t} \in \mathbb{R}$. The vector realization of all joint observations at time t is defined as:

$$\mathbf{z}_{.,t} = (z_{1,t}, ..., z_{D,t}) \in \mathbb{R}^D \tag{3.1}$$

For instance, in the context of wind farms, $z_{i,t}$ represents the output generated power of wind farm i at time t, measured in megawatts (MW), and $\mathbf{z_{.,t}}$ is the vector representing output of all wind farms at time t. Similarly, for an individual target i, the realization of the

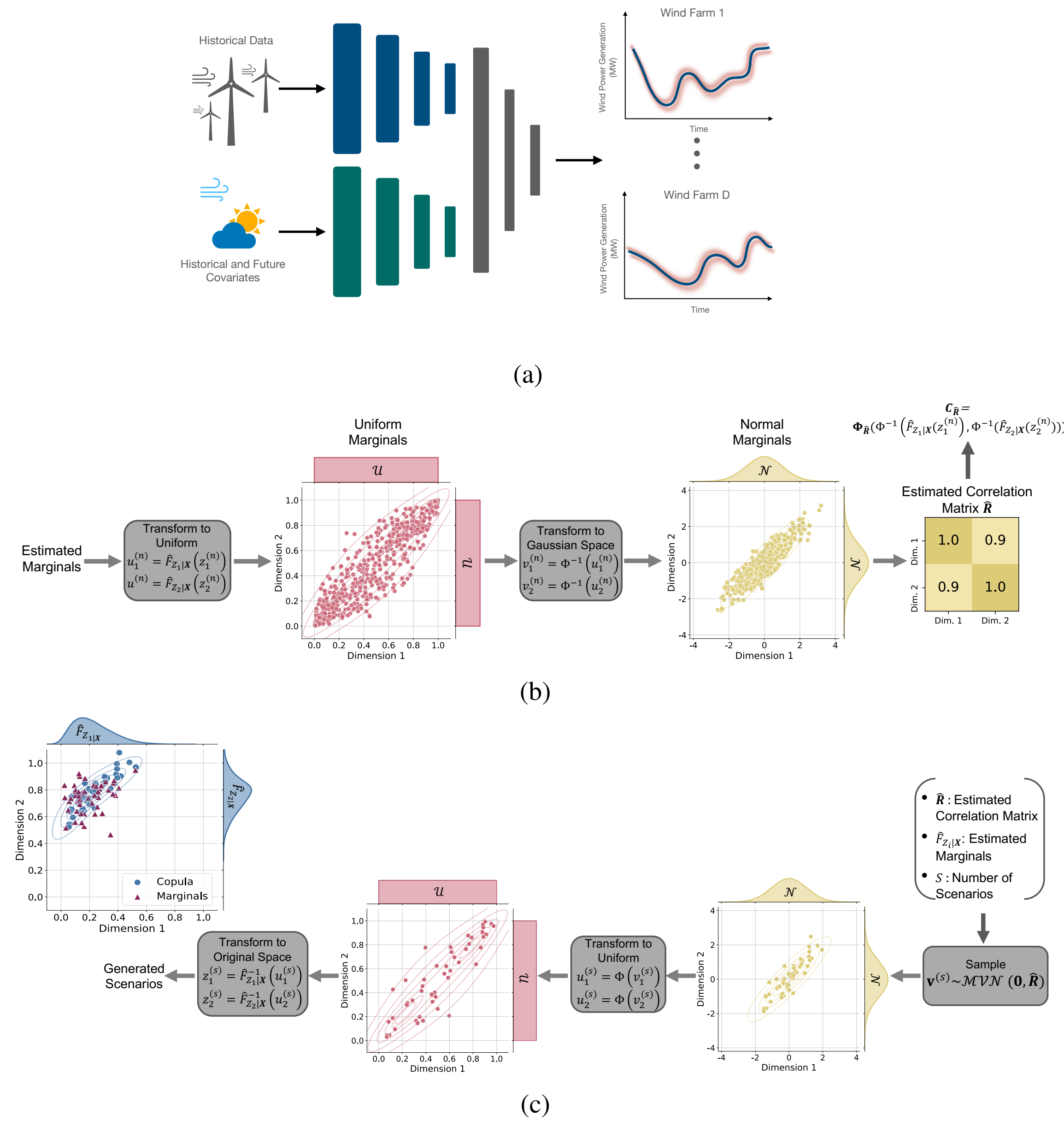


Figure 3.1: Schematic of the proposed method for scenario generation in RES.
(a) Schematic of the weather informed multivariate probabilistic forecasting for wind farms, given the historical wind power generation and the past and future weather covariates. (b) Schematic of estimating the Gaussian copula for a 2D example. (c) Schematic of the scenario generation steps given the estimated copula and estimated marginals for a 2D problem. Examples of generated scenarios using marginals are also presented in (c) with red triangles.

Table 3.1: Nomenclature

Sets	
$\mathcal{D}$	Set of target variables (i.e., load zones, wind farms, or solar farms)
$\mathcal{D}'$	Set of covariates
$\mathcal{T}$	Time window
$\mathcal{H}$	Forecasting horizon time window
$\mathcal{W}$	Past observation time window
Ω	Dataset of input-output pairs
Indices	
$i \in \mathcal{D}$	Spatial index
$t \in \mathbb{Z}$	Time index of forecast start times
$\tau \in \mathcal{H}$	Time index representing a specific time step within the forecast horizon
Variables	
$Z_{i,t}$	Random variable for target at location i and time t
$z_{i,t}$	Realization of the random variable $Z_{i,t}$
$\mathbf{z}_{\cdot,t}$	Vector realization of all joint observations in locations within $\mathcal{D}$ at time t
$\mathbf{z}_{i,\mathcal{T}}$	Vector realization of observations at location i within time interval $\mathcal{T}$
$\mathbf{Z}_{\cdot,\mathcal{T}}$	Matrix of random variables for target locations within $\mathcal{D}$ over time horizon $\mathcal{T}$
$\mathbf{z}_{\cdot,\mathcal{T}}$	Matrix of all observations in $\mathcal{D}$ over time interval $\mathcal{T}$ (i.e., realization of $\mathbf{Z}_{\cdot,\mathcal{T}}$)
$\mathbf{x}_{\cdot,t}$	Vector realization of all joint covariates within $\mathcal{D}'$ at time t
$\mathbf{x}_{\cdot,\mathcal{T}}$	Matrix of all covariates in $\mathcal{D}'$ within time interval $\mathcal{T}$
Z_i, z_i	Random variable and realization in target space with CDF F_{Z_i}
U_i, u_i	Random variable and realization in uniform space $\mathcal{U}[0,1]$
V_i, v_i	Random variable and realization in Gaussian space $\mathcal{N}(0,1)$
$F_{i,\tau}(z_{i,\tau})$	Marginal CDF for target at location $i \in \mathcal{D}$ at time $\tau \in \mathcal{H}$
$\mathbf{F}_{\cdot,\mathcal{H}}(\mathbf{z}_{\cdot,\mathcal{H}})$	Joint CDF for all targets within $\mathcal{D}$ over time window $\mathcal{H}$
$\hat{F}, \hat{\mathbf{F}}$	Empirical marginal and joint CDFs
$\mathbf{0}, \mathbf{1}$	Vectors of all zeros and ones, respectively
$\mathbf{R}$	Correlation matrix
$\mathbf{C}$	Copula function
Φ	Standard normal CDF
$\mathbf{\Phi_R}$	Multivariate normal CDF with standard normal marginal CDFs and correlation matrix $\mathbf{R}$
Operators	
vec	Matrix vectorization
max	Largest value operator

univariate time series for time interval $\mathcal{T} = \{t+1, ..., t+T\} \subseteq \mathbb{Z}$ is denoted by:

$$\mathbf{z}_{i,\mathcal{T}} = (z_{i,t+1}, ..., z_{i,t+T}) \in \mathbb{R}^T \tag{3.2}$$

Hence, the multivariate time series realization corresponding to the targets in the system from time $t+1$ to $t+T$ is denoted as:

$$\mathbf{z}_{\cdot,\mathcal{T}} = \begin{pmatrix} — & \mathbf{z}_{1,t+1:t+T} & — \\ & \vdots & \\ — & \mathbf{z}_{D,t+1:t+T} & — \end{pmatrix} \in \mathbb{R}^{D\times T} \tag{3.3}$$

The paper also considers so-called *covariates*, denoted by $\mathbf{x}_{\cdot,t}$ with support in $\mathbb{R}^{D'}$. Thereby, for $j \in \mathcal{D}' = \{1, ..., D'\}$, $x_{j,t}$ represents the observed value of the j-th covariate at time t, and the vector representation of all covariates at time t is denoted by $\mathbf{x}_{\cdot,t} = (x_{1,t}, ..., x_{D',t})$. The corresponding covariates in the paper include time encoding, e.g., day of the week, and the weather-related quantities such as temperature, humidity, cloud coverage, wind speed, etc. Hence, the matrix realization of the covariates is denoted as below:

$$\mathbf{x}_{\cdot,\mathcal{T}} = \begin{pmatrix} — & \mathbf{x}_{1,t+1:t+T} & — \\ & \vdots & \\ — & \mathbf{x}_{D',t+1:t+T} & — \end{pmatrix} \in \mathbb{R}^{D'\times T} \tag{3.4}$$

The details corresponding to each target and the covariates are discussed in section 3.6.

3.4 Multivariate Forecasting

Let $t \in \mathbb{Z}$ and consider the forecasting horizon $\mathcal{H} = \{t+1, ..., t+H\}$ of length H, and the past observation window of length W as $\mathcal{W} = \{t-W+1, .., t-1, t\}$. A time series forecasting model predicts the target $\mathbf{z}_{\cdot,\mathcal{H}} \in \mathbb{R}^{D\times H}$ as $\hat{\mathbf{z}}_{\cdot,\mathcal{H}} \in \mathbb{R}^{D\times H}$. Figure 3.1(a) shows schematic of the DL-based model for multivariate probabilistic forecasting of the wind power generation given historical data and covariates. More generally, probabilistic forecasting can be considered as modeling the joint cumulative distribution function (CDF) of the target variables conditioned on the past observations $\mathbf{z}_{\cdot,\mathcal{W}} \in \mathbb{R}^{D\times W}$, past covariates

$\mathbf{x}_{\cdot,\mathcal{W}} \in \mathbb{R}^{D' \times W}$, and the future covariates $\mathbf{x}_{\cdot,\mathcal{H}} \in \mathbb{R}^{D' \times H}$ as

$$\mathbf{F}_{\mathcal{H}}(\mathbf{z}_{\cdot,\mathcal{H}}) = \mathbb{P}\left(\mathbf{Z}_{\cdot,\mathcal{H}} \leq \mathbf{z}_{\cdot,\mathcal{H}} \mid \mathbf{z}_{\cdot,\mathcal{W}}, \mathbf{x}_{\cdot,\mathcal{W}}, \mathbf{x}_{\cdot,\mathcal{H}}\right) \tag{3.5}$$

Modeling this joint CDF directly is challenging due to the high-dimensionality of the problems at hand. An alternative, in the multivariate forecasting setting, is to jointly predict the following marginal distributions:

$$F_{i,\tau}(z_{i,\tau}) = \mathbb{P}(Z_{i,\tau} \leq z_{i,\tau} \mid \mathbf{z}_{\cdot,\mathcal{W}}, \mathbf{x}_{\cdot,\mathcal{W}}, \mathbf{x}_{\cdot,\mathcal{H}}), \tag{3.6}$$

where $i \in \mathcal{D}$ is the spatial index indicating the location of the target and $\tau \in \mathcal{H}$ represents the time index within the forecasting horizon. In other words, the multivariate forecasting model takes as input the historical data and covariates, which are shared across different targets, and predicts the marginal distributions. Then, given the estimated marginals as the outputs of the multivariate forecasting, the next critical step is to restore the spatial and temporal correlations among the target variables. This is necessary for realistic scenario generation, particularly within the RES where the targets (e.g., electricity demand and outputs of multiple wind farms or solar farms at different locations and time horizons) are highly correlated. this chapter restores the joint distribution of the random variable $\mathbf{Z}_{\cdot,\mathcal{H}}$ using a copula method, as discussed in section 3.5.

3.5 Restoring Correlations via Copula

The copula method provides a robust statistical tool for modeling the underlying dependencies and correlations in high-dimensional probabilistic forecasting. This method enables the estimation of the joint distribution by combining the marginal distributions with a copula function. In the context of RES, the copula method facilitates the generation of realistic scenarios that accurately reflect the true variability and spatial-temporal interdependencies.

3.5.1 Copula Method

A function $\mathbf{C} : [0,1]^d \to [0,1]$ is a *copula* if it is the joint CDF of a d-dimensional random vector $\mathbf{U} = (U_1, \ldots, U_d)$ with uniformly distributed marginals on the interval $[0,1]$. Sklar's Theorem [80] shows that any multivariate CDF can be expressed as the product of its marginal distributions and a copula function.

Theorem 1 (Sklar's Theorem) *Let* $\mathbf{Z}$ *be a* d*-dimensional random variable with joint CDF* $\mathbf{F_Z}$ *and marginal CDFs* $F_{Z_1}, \ldots, F_{Z_d}$*. There exists a copula* $\mathbf{C}$ *such that*

$$\mathbf{F_Z}(z_1, ..., z_d) = \mathbf{C}\left(F_{Z_1}(z_1), ..., F_{Z_d}(z_d)\right), \tag{3.7}$$

and the random variables $U_i = F_{Z_i}(Z_i)$ *are uniformly distributed on interval* $[0,1]$ *(i.e.,* $U_i \sim \mathcal{U}[0,1]$*). The copula* $\mathbf{C}$ *is unique, given continuous marginals* $F_{Z_1}, \ldots, F_{Z_d}$*.*

Sklar's theorem thus enables the decomposition of any multivariate CDF into its marginal CDFs and a copula, without specific assumptions on the marginals. Hence, given the estimated marginal CDFs, it is possible, via a Copula function, to reconstruct the joint CDF and represent the underlying dependencies and correlations among different dimensions. This is particularly crucial in probabilistic forecasting in RES where spatial and temporal correlations should be restored in high dimensions.

3.5.2 Conditional Copula

Sklar's theorem can be extended to modeling conditional distributions, which is particularly important in regression and time series forecasting [81, 82]. Given a target random vector $\mathbf{Z} = (Z_1, \ldots, Z_d)$ conditioned on a set of input variables $\mathbf{X}$, the conditional copula aims at modeling $\mathbf{F_{Z|X}}$ from its marginals $F_{Z_1|\mathbf{X}}, \ldots, F_{Z_d|\mathbf{X}}$ as follows:

$$\mathbf{F_{Z|X}} = \mathbf{C}(F_{Z_1|\mathbf{X}}(z_1|\mathbf{x}), \ldots, F_{Z_d|\mathbf{X}}(z_d|\mathbf{x})) \tag{3.8}$$

It is important to note that, in general, the input variables $\mathbf{X}$ should be consistent across different marginals and the copula. For further details and considerations regarding different input variable sets under specific assumptions, readers are referred to [83].

3.5.3 Gaussian Copula

One of the most popular and effective ways of linking the joint distribution into its marginals is the Gaussian copula. Considering Φ as the CDF of the standard normal distribution and $\mathbf{\Phi_R}$ as the multivariate normal CDF with mean vector of $\mathbf{0}$ and covariance matrix of $\mathbf{R}$, the Gaussian copula $\mathbf{C_R} : [0,1]^d \to [0,1]$ is defined as:

$$\mathbf{C_R} = \mathbf{\Phi_R}\big(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d)\big) \tag{3.9}$$

As a result, the multivariate joint CDF $\mathbf{F_{Z|X}}$ can be decomposed as:

$$\mathbf{F_{Z|X}} = \mathbf{\Phi_R}\Big(\Phi^{-1}\big(F_{Z_1|\mathbf{X}}(z_1|\mathbf{x})\big), \ldots, \Phi^{-1}\big(F_{Z_d|\mathbf{X}}(z_d|\mathbf{x})\big)\Big) \tag{3.10}$$

The two main pillars to estimate the joint CDF under the Gaussian copula framework are: 1) the marginal CDFs $F_{Z_i|\mathbf{X}}$ for $i \in \{1, \ldots, d\}$ and 2) the correlation matrix $\mathbf{R} \in \mathbb{R}^{d \times d}$, which is responsible for modeling the correlations among different dimensions.

Figure 3.1(b) shows the schematic of the approach for estimating the Gaussian copula for a 2D (i.e., $d = 2$) problem. First, the marginal CDFs are estimated as $\hat{F}_{Z_i|\mathbf{X}}$ using a probabilistic forecasting model and a training dataset Ω_{train} with $N = |\Omega_{\text{train}}|$ samples. Then, given the training covariate and target pairs (i.e., $(\mathbf{x}^{(n)}, \mathbf{z}^{(n)}) \in \Omega_{\text{train}}, n \in \{1, \ldots, N\}$), and the estimated marginals $\hat{F}_{Z_i|\mathbf{X}}$, each of the target dimensions $z_i^{(n)}$ is transformed into $u_i^{(n)} = \hat{F}_{Z_i|\mathbf{X}}$. The transformed variables $u_i^{(n)}$ are distributed uniformly on $[0, 1]$ if the underlying model that estimates marginals is well-calibrated (refer to the probability integral transform in the Appendix for more details). Following this step, the corresponding data is transformed through the inverse of the standard normal CDF to $v_i^{(n)} = \Phi^{-1}(u_i^{(n)})$. This results in a space with standard normal marginal CDFs. Finally, the correlation matrix is estimated as $\hat{\mathbf{R}}$, using Pearson correlation over $\mathbf{v}^{(n)} = (v_1^{(n)}, \ldots, v_d^{(n)}), n \in \{1, \ldots, N\}$. The estimated correlation matrix is then used along with the estimated marginals to form the Gaussian copula as presented in Equation (Equation 3.10). In short, given a dataset with dimension $d \times N$, where d corresponds to the dimension of the copula and N is the number of training data points, i.e., $N = |\Omega_{train}|$, the correlation matrix is estimated as:

$$\hat{\mathbf{R}} = \mathrm{corr}\left(\begin{bmatrix} - \Phi^{-1}\left(\hat{F}_{z_1|\mathbf{x}}\left(z_1^{(n)} \mid \mathbf{x}^{(n)}\right)\right) - \\ \vdots \\ - \Phi^{-1}\left(\hat{F}_{z_d|\mathbf{x}}\left(z_d^{(n)} \mid \mathbf{x}^{(n)}\right)\right) - \end{bmatrix}_{d \times N} \right) \tag{3.11}$$

3.5.4 Scenario Generation

The combination of the multivariate probabilistic forecasting and copula method provides the estimated joint distribution of the target variables, incorporating the correlations among different dimensions. Consequently, it is possible to generate probabilistic scenarios by sampling from the estimated joint distribution that captures the underlying correlations between different dimensions.

The scenario generation steps are outlined in Algorithm 2 and are depicted in Figure 3.1(c) for a 2D (i.e., $d = 2$) example. The algorithm takes as input the estimated correlation matrix $\hat{\mathbf{R}}$, the estimated marginals $\hat{F}_{Z_i|\mathbf{X}}$ for $i \in \{1, \ldots, d\}$, and the number of required scenarios (i.e., the number of samples from the joint distribution) S, and it outputs $\hat{\mathbf{z}}^{(s)} = (\hat{z}_1^{(s)}, \ldots, \hat{z}_d^{(s)})$ for $s \in \{1, ..., S\}$. The algorithm starts by sampling $\mathbf{v}^{(s)} = (v_1^{(s)}, \ldots, v_d^{(s)})$ from a multivariate normal distribution with $\mathbf{0}$ mean and estimated covariance matrix $\hat{\mathbf{R}}$ (i.e., $\mathcal{MVN}(\mathbf{0}, \hat{\mathbf{R}})$). The samples are then transformed using the CDF of the standard normal into $u_i^{(s)} = \Phi(v_i^{(s)})$, which has uniform marginals (as shown in Figure 3.1(c)). Finally, the samples are mapped into the original space via $\hat{z}_i^{(s)} = \hat{F}^{-1}_{Z_i|\mathbf{X}}(u_i^{(s)})$. The resulting vector $\hat{\mathbf{z}}^{(s)}$ corresponds to the samples generated using the copula method,

Algorithm 2 Scenario Generation Pseudo-code

Inputs: $\hat{\mathrm{R}}$ (Estimated Correlation Matrix),
$\hat{F}_{Z_i|\mathbf{X}}$ (Estimated Marginals for $i \in \{1, \ldots, d\}$),
S (Total Number of Required Scenarios)
Output: Generated Scenarios: $\hat{\mathbf{z}}^{(s)} = (\hat{z}_1^{(s)}, \ldots, \hat{z}_d^{(s)})$ for $s \in \{1, \ldots, S\}$
for $s = 1$ to S **do** *(Can be done in parallel)*
  Sample $\mathbf{v}^{(s)} = (v_1^{(s)}, \ldots, v_d^{(s)})$ from $\mathcal{MVN}(\mathbf{0}, \hat{\mathbf{R}})$
  for $i = 1$ to d **do**
    Compute $u_i^{(s)} = \Phi(v_i^{(s)})$
    Map to the original space:
      $\hat{z}_i^{(s)} = \hat{F}^{-1}_{Z_i|\mathbf{X}}(u_i^{(s)})$
  end for
end for

which captures the interdependencies and correlations among the different dimensions.

3.5.5 Copula for Time Series Scenario Generation

This section extends the Gaussian copula method to the context of time series forecasting in RES, where the objective is to model a joint CDF for all sites and time steps, and generate scenarios that capture both spatial and temporal dependencies across the forecasting horizon. In this setting, the joint conditional CDF $\mathbf{F}_{\mathcal{H}}(\mathbf{z}_{\cdot,\mathcal{H}})$ is over $d = D \times H$ dimensions (recall that $D = |\mathcal{D}|$ is the number of locations and $H = |\mathcal{H}|$ is the number of time steps in the forecast horizon). Similar to what is presented in subsection 3.5.3, the joint conditional CDF can be derived from: 1) the estimated conditional marginals $\hat{F}_{i,\tau}(z_{i,\tau})$ (where $i \in \mathcal{D}$ and $\tau \in \mathcal{H}$) and 2) a Gaussian copula with an estimated correlation matrix $\hat{\mathbf{R}} \in \mathbb{R}^{(D\times H)\times(D\times H)}$ that captures the spatio-temporal correlations as follows:

$$\hat{\mathbf{F}}_{\mathcal{H}}(\mathbf{z}_{\cdot,\mathcal{H}}) = \mathbf{\Phi}_{\hat{\mathbf{R}}}\Big(\Phi^{-1}\big(\hat{F}_{1,t+1}(z_{1,t+1})\big), \ldots, \Phi^{-1}\big(\hat{F}_{1,t+H}(z_{1,t+H})\big),$$
$$\ldots, \tag{3.12}$$
$$\Phi^{-1}\big(\hat{F}_{D,t+1}(z_{D,t+1})\big) \ldots, \Phi^{-1}\big(\hat{F}_{D,t+H}(z_{D,t+H})\big)\Big)$$

Note that $\hat{\mathbf{F}}_{\mathcal{H}}$ and $\hat{F}_{i,\tau}$ (defined in (Equation 3.5) and (Equation 3.6), respectively) are conditioned on the inputs $\mathbf{z}_{\cdot,\mathcal{W}}$, $\mathbf{x}_{\cdot,\mathcal{W}}$, and $\mathbf{x}_{\cdot,\mathcal{H}}$, which are omitted in Equation (Equation 3.12) for simplicity. Also, τ depends on t, which corresponds to the starting of the forecasting time window. The marginals are estimated using multivariate probabilistic time series forecasting given a training dataset and the corresponding covariates as:

$$\Omega_{\text{train}} = \left\{ \left(\{\mathbf{z}^{(t)}_{\cdot,\mathcal{W}}, \mathbf{x}^{(t)}_{\cdot,\mathcal{W}}, \mathbf{x}^{(t)}_{\cdot,\mathcal{H}}\}, \mathbf{z}^{(t)}_{\cdot,\mathcal{H}}\right) \middle| \begin{aligned} t \in& \{W, W+1, \ldots, N-H\}, \\ \mathcal{H} =& \{t+1, \ldots, t+H\}, \\ \mathcal{W} =& \{t-W+1, \ldots, t\} \end{aligned} \right\},$$

where N is the total length of the training time series dataset and t is the time index in the rolling time window.

Next, using the estimated marginal CDFs and following the steps outlined in subsection 3.5.3, the correlation matrix $\hat{\mathbf{R}} \in \mathbb{R}^{(D\times H)\times(D\times H)}$ can be estimated using Pearson correlation, with $N' = N - W - H + 1$ samples (i.e., the training dataset excluding the few first and last time steps).

The scenario generation follows the process outlined in subsection 3.5.4 and Algorithm 1. Samples are drawn from $\mathcal{MVN}(\mathbf{0}, \hat{\mathbf{R}})$, producing S vectors with dimensions $1 \times (D \times H)$. Each sample is then reshaped into a matrix of size $D \times H$, where the rows and the columns correspond to locations and time, respectively.

3.6 Experiment Setup

A comprehensive performance evaluation was conducted with large-scale data using various time series forecasting methods. Both deterministic and probabilistic forecasting methods are employed to assess their efficacy under different metrics. Additionally, the experiments incorporate weather information to evaluate its importance in predicting RES. The forecast horizon $\mathcal{H}$ is set to 48 hours, which aligns with the common practice for day-ahead market-clearing and reliability assessment in power systems [84]. However, the methodology can be extended to longer or shorter time horizons depending on the specific requirements.

Table 3.2: Dataset Summary

This table presents the overall minimum and maximum values across all time series. For wind and solar, the maximum values represent the capacity of the largest wind or solar farms. For the load time series, which does not have a predefined capacity, the capacity is defined by the highest observation in the training dataset. The minimum value for each type corresponds to the lowest recorded observation across all time series.

Type	D	Level	Total(GW)	Min (MW)	Max (MW)
load	6	zone	124.2	9424.6	32148.5
wind	286	unit	21.4	0.6	495.0
solar	376	unit	1.8	0.2	138.0

3.6.1 Dataset

Table 3.2 provides summary statistics of the MISO dataset that is used in this study which contains time series data for the years 2018 and 2019 with hourly resolution within the MISO system [61]. This dataset includes zone-level time series for loads and unit-level time series of wind and solar power generation. Additionally, it includes forecasts generated by NREL's System Advisor Model (SAM) for the year 2019, which is considered for comparison with the developed models in this study.

To explore the effect of weather data, this study incorporates an additional dataset of weather forecasts for the MISO system. The weather forecasts are gathered from the

Table 3.3: Description of Meteorological Variables [85].

Variable	Description
10u	- Eastward component of the 10 metre wind (*U-component*): Measures the horizontal speed of air moving towards the east at a height of ten metres above the Earth's surface, in metres per second.
10v	- Northward component of the 10 metre wind (*V-component*): Measures the horizontal speed of air moving towards the north at a height of ten metres above the Earth's surface, in metres per second.
2t	- 2 metre temperature: Temperature of the air measured at 2 metres above the surface of the Earth, in kelvins (K).
skt	- Skin temperature: Temperature at the interface between the Earth and the atmosphere, in kelvins (K). Measured directly at the Earth's surface.
sp	- Surface pressure: Atmospheric pressure exerted on the Earth's surface by the air above, measured in pascals (Pa). Reflects the weight of the air vertically above a point.
ssr	- Net surface solar radiation: Total solar radiation (shortwave radiation) reaching a horizontal plane at the Earth's surface, adjusted for the albedo, measured in joules per square metre (J/m^2).
tcc	- Total cloud cover: Proportion of the sky occluded by clouds, measured on a scale from 0 (no cover) to 1 (complete cover).

publicly available datasets produced by the European Center for Medium-Range Weather Forecasts (ECMWF), particularly from the THORPEX Interactive Grand Global Ensemble (TIGGE) dataset [73, 85]. This dataset provides essential meteorological variables including surface pressure, wind speed, cloud cover, skin temperature, solar irradiation, as outlined in Table 3.3. Issued twice everyday at 12:00 and 00:00 UTC, the forecasts from the TIGGE database are provided in 6-hour intervals [85]. The forecast data is collected on a 1° latitude by 1° longitude grid covering the MISO territory. The 6-hour temporal resolution time series from the ECMWF are then interpolated to hourly granularity to align with the load, and wind and solar power time series data used in this study.

3.6.2 Methods

In this study, different methods are employed to determine their efficacy for 48-hour forecasting horizon prediction in RES and load. The models encompass both traditional statistical methods and advanced DL-based techniques with and without incorporation of weather data, to provide a comprehensive performance comparison. The models include:

- ARIMA: This model is implemented in a univariate framework due to the shortcoming of the autoregresive methods in handling high-dimensional input data [1].
- Dlinear: This is a linear model tailored for time series forecasting. The implementation in this work considers the spatio-temporal correlations by regressing the forecasts over the previous observations and past and future covariates [86].
- Nlinear: This model normalizes each input by subtracting the last value in the sequence and then adds it back after passing through the linear layer to address distribution shift. In this chapter, the model is adapted to probabilistic setting by using quantile regression objective function [86].
- DeepAR: This is an RNN-based model designed to capture the temporal dependencies and provide multi-step ahead forecast by recursively applying the model, using its own predictions as inputs for subsequent steps. In this work, the model's output parameters are parametrized using a Beta distribution to account for the capacity factors in RES, which are inherently bounded between 0 and 1 [6].
- TFT: This is a transformer-based model designed for multi-horizon forecasting with mixed-type inputs, including static covariates, known future inputs, and past exogenous time series. Quantile loss is utilized for training the model in this work [5].

The models are trained using historical data of the targets from 2018 within the MISO system, with the remaining data reserved for testing. At the beginning of each month, a new model is trained using all available observations since Jan. 1st, 2018 up to midnight on the first day of that month. The first week of each month is designated for validation purposes (i.e., for fine tuning the model parameters), and the models are subsequently evaluated during the second week of each month. The DL models are trained both with and without weather data to assess the importance of weather covariates. Evaluations are conducted only for the first nine months, as the ECMWF data was not available from October 2019 onward.

Weather features are utilized as additional covariates in the forecasting tasks for the WI-based models. These covariates include both historical weather data and weather forecasts, which are available in advance as discussed in the previous section. For each target variable, weather information from the closest region on a 1° by 1° grid is used. For load forecasting, skin temperature and total cloud cover are incorporated as key variables. For wind power generation forecasting, the model uses wind speed, surface pressure, and both V and U wind components, which indicate horizontal airflow towards the north and east, respectively. In forecasting solar power generation, total cloud cover and surface solar radiation are key variables. When generating forecasts, the actual weather information from the past and weather forecasts for the upcoming forecasting horizon are fed into the WI-based models..

The ARIMA model parameters are optimized using the open-source Statsforecasts package [87], which features an AutoARIMA function for parameter selection. Training of the DL-based models are performed in Pytorch using Adam optimizer [88]. Early stopping with patience of 10 steps in conjunction with a learning rate scheduler (ReduceLROnPlateau) with patience of 5 steps is used for all models. The remaining hyperparameters for each model are optimized using Optuna [89]. The details about the training process are provided in subsection 3.6.4.

3.6.3 Evaluation Metrics

This section outlines various metrics employed to quantify the models for both deterministic and the probabilistic forecasting (i.e., scenario generation). Recall that, for a given target $i \in \mathcal{D}$ and each time step within the forecasting time horizon ($\tau \in \mathcal{H}$), the model produces output in the form of marginal CDFs, whereas S forecasts (i.e., scenarios) are generated for the scenario generation using Algorithm 1. This essentially gives the prediction one more

Table 3.4: Summary and Description of Evaluation Metrics.

Metric	Formula	Description
$\text{NMAE}^{**}_{\texttt{ind}}$	$\frac{1}{N_{\text{test}}DT}\sum_{t=1}^{N_{\text{test}}}\sum_{i\in\mathcal{D},\tau\in\mathcal{H}}\left\lvert\frac{\hat{z}_{i,\tau}-z_{i,\tau}}{\max\{\mathbf{z}_{i,\mathcal{T}_{\text{train}}}\}}\right\rvert$	Computed on individual space-time dimensions, normalized by generator capacity or maximum value in the first year of training dataset, measuring model's accuracy relative to the scale of the time series.
$\text{NMAE}_{\texttt{s-sum}}$	$\frac{1}{N_{\text{test}}T}\sum_{t=1}^{N_{\text{test}}}\sum_{\tau\in\mathcal{H}}\left\lvert\frac{\sum_{i=1}^{D}\hat{z}_{i,\tau}-\sum_{i=1}^{D}z_{i,\tau}}{\sum_{i=1}^{D}\max\{\mathbf{z}_{i,\mathcal{T}_{\text{train}}}\}}\right\rvert$	Computed on the summation over space, measuring model's accuracy relative to the scale of the time series.
$\text{RMSE}_{\texttt{ind}}$	$\sqrt{\frac{1}{N_{\text{test}}DT}\sum_{t=1}^{N_{\text{test}}}\sum_{i\in\mathcal{D},\tau\in\mathcal{H}}\left(\hat{z}_{i,\tau}-z_{i,\tau}\right)^2}$	Root Mean Squared Error computed on individual level over all space and time dimensions, measuring model's accuracy in MW scale.
$\text{RMSE}_{\texttt{s-sum}}$	$\sqrt{\frac{1}{N_{\text{test}}T}\sum_{t=1}^{N_{\text{test}}}\sum_{\tau\in\mathcal{H}}\left(\sum_{i=1}^{D}\hat{z}_{i,\tau}-\sum_{i=1}^{D}z_{i,\tau}\right)^2}$	Root Mean Squared Error computed on the summation of wind power, solar power, or load over space, measuring model's accuracy in MW scale.
$\text{ED}^{*}_{\texttt{ind}}$	$\frac{1}{N_{\text{test}}}\sum_{t=1}^{N_{\text{test}}}\hat{\text{ED}}(\{\text{vec}(\hat{z}_{\cdot,\mathcal{H}})^{(s)},s\in\mathcal{S}\},\text{vec}(z_{\cdot,\mathcal{H}}))$	Average energy distance between the actual and probabilistic forecasts, capturing the joint distribution over space and time.
$\text{ED}^{*}_{\texttt{s-sum}}$	$\frac{1}{N_{\text{test}}}\sum_{t=1}^{N_{\text{test}}}\hat{\text{ED}}(\{\hat{z}^{(s)}_{\cdot,\mathcal{H}},s\in\mathcal{S})\},\mathbf{1}^{\text{T}}z_{\cdot,\mathcal{H}})$	Energy distance computed on space-sum level, capturing the joint distribution over time on the space-sum level.
$\text{VS}_{\texttt{s-sum}}$	$\frac{1}{N_{\text{test}}}\sum_{t=1}^{N_{\text{test}}}\sum_{\tau_1\in\mathcal{H}}\sum_{\tau_2\in\mathcal{H}}\left(\left\lvert\sum_{i=1}^{D}z_{i,\tau_1}-\sum_{i=1}^{D}z_{i,\tau_2}\right\rvert^{\frac{1}{2}} - \frac{1}{S}\sum_{s=1}^{S}\left\lvert\sum_{i=1}^{D}\hat{z}^{(s)}_{i,\tau_1}-\sum_{i=1}^{D}\hat{z}^{(s)}_{i,\tau_2}\right\rvert^{\frac{1}{2}}\right)$	Evaluated for summation of load and renewable generation predictions over space, capturing temporal correlations.
$\text{VS}_{\texttt{t-sum}}$	$\frac{1}{N_{\text{test}}}\sum_{t=1}^{N_{\text{test}}}\sum_{i_1\in\mathcal{D}}\sum_{i_2\in\mathcal{D}}\left(\left\lvert\sum_{\tau\in\mathcal{H}}z_{i_1,\tau}-\sum_{\tau\in\mathcal{H}}z_{i_2,\tau}\right\rvert^{\frac{1}{2}} - \frac{1}{S}\sum_{s=1}^{S}\left\lvert\sum_{\tau\in\mathcal{H}}\hat{z}^{(s)}_{i_1,\tau}-\sum_{\tau\in\mathcal{H}}\hat{z}^{(s)}_{i_2,\tau}\right\rvert^{\frac{1}{2}}\right)$	Evaluated for summation over time periods, capturing spatial correlations.

* $\hat{\text{ED}}(\mathbf{x},\mathbf{y})=\frac{2}{S}\sum_{i=1}^{S}\lVert\mathbf{x}_i-\mathbf{y}\rVert-\frac{1}{S^2}\sum_{i=1}^{S}\sum_{j=1}^{S}\lVert x_i-x_j\rVert$, more details in appendix.

** $\max\{\mathbf{z}_{i,\mathcal{T}_{\text{train}}}\}$ refers to the capacity for wind/solar farm i, which is the maximum observation in the training dataset for time series i.

dimension, which indiced by superscript (s).

The evaluation of the deterministic forecasts are performed at the individual and space-sum levels. Additionally, the probabilistic forecasts metrics are computed over time-sum levels to quantify the ability of the models in capturing temporal dependencies. Let $\mathbf{1}$ be the column vector of all ones (with proper dimension), the comparison will be:

- individual (`ind`) level — directly comparing the forecast vectors $\hat{\mathbf{z}}_{i,\tau}^{(s)}, s \in \{1, ..., S\}$ and the ground truth $\mathbf{z}_{i,\tau}$;
- space-sum (`s-sum`) level — comparison after summation over all locations, i.e., comparing $\mathbf{1}^{\mathrm{T}}\hat{\mathbf{z}}_{\cdot,\mathcal{H}}^{(s)}$ with $\mathbf{1}^{\mathrm{T}}\mathbf{z}_{\cdot,\mathcal{H}}$;
- time-sum (`t-sum`) level — comparison after summation over all time steps, i.e., comparing $\hat{\mathbf{z}}_{\cdot,\mathcal{H}}^{(s)}\mathbf{1}$ with $\mathbf{z}_{\cdot,\mathcal{H}}\mathbf{1}$.

Table 3.4 outlines the metrics used to evaluate deterministic and probabilistic forecasts. The precision of the deterministic forecasts is evaluated using normalized absolute error (NMAE) and root mean squared error (RMSE) over the individual forecasts and space-sum levels. These metrics are calculated for the load, wind power, and solar power generation over unseen test data across the different models discussed in subsection 3.6.2, with and without the inclusion of the weather covariates.

To evaluate the efficacy of the models and the significance of integrating the copula method for realistic scenario generation, the results report the energy distance (ED) and variogram score (VS). ED is a proper scoring rule that measures the distance between the predicted CDF and the actual observations. It generalizes the continuous ranked probability score (CRPS) and is particularly useful for vector-valued forecasts in high-dimensional spaces [90]. This chapter considers ED due to its generalization capabilities for high-dimensional data. However, in scenario generation, both ED and CRPS fail to detect differences in correlation structures [91]. Therefore, VS is introduced to quantify differences between the correlations of the scenarios and the ground truth. More details regarding the metrics are presented in Appendix A in the Appendix.

3.6.4 Training Process

The models were trained using the Adam optimizer, and a learning rate scheduler was employed to adaptively adjust the learning rate during training. Specifically, we utilized the `ReduceLROnPlateau` scheduler with a reduction factor of 0.1, a threshold of 0.0001, a patience of 5 epochs, and a minimum learning rate of $\mathrm{lr} \times 10^{-4}$. Early stopping criteria were also implemented based on the validation loss, with a patience of 10 epochs and a minimum delta of 0.0001 to prevent overfitting. All experiments were conducted on a single node equipped with one NVIDIA A100 GPU using the PACE cluster infrastructure [92].

3.7 Results

This section presents the experimental results for the deterministic and probabilistic forecasting of the load, and the wind and solar power generation within MISO. Initially, the deterministic forecasting results for different models discussed in Section subsection 3.6.2 are presented to compare the performance with and without the inclusion of weather data. Following this, the probabilistic forecasting results are discussed, highlighting the rational for incorporating the copula technique to restore the correlations. The experimental results are reported on distinct individual and aggregate levels.

3.7.1 Deterministic Prediction

To evaluate the performance of the different models discussed in Section subsection 3.6.2 for point-predictions and quantify the importance of weather covariates, deterministic forecasts are performed for each model. For each probabilistic forecasting method, and given the predicted marginal distributions of each space-time dimension, the expected values on each dimension are used as deterministic forecasts. For models that trained with quantile

Table 3.5: Deterministic Forecast Evaluation.

Performance of different models are evaluated on MISO dataset with and without inclusion of the weather covariates. The metrics are calculated at individual (i.e., `ind`) and space-sum (i.e., `s-sum`) levels.

	load				wind				solar			
	s-sum		ind		s-sum		ind		s-sum		ind	
Model	NMAE	RMSE	NMAE	RMSE	NMAE	RMSE	NMAE	RMSE	NMAE	RMSE	NMAE	RMSE
ARIMA	4.2%	6316.1	5.2%	1302.4	16.6%	4136.8	28.5%	25.4	9.0%	190.3	12.4%	0.7
Dlinear	2.8%	4211.2	4.2%	1003.1	16.7%	4134.2	26.9%	24.6	7.2%	181.0	10.0%	0.7
Nlinear	3.3%	4964.5	3.9%	994.9	16.2%	4030.6	26.0%	23.9	7.4%	182.5	10.2%	0.7
DeepAR	5.1%	7261.4	6.1%	1428.8	16.8%	4357.0	27.2%	25.3	5.2%	168.9	8.2%	0.7
TFT	2.9%	4473.8	3.7%	954.5	16.7%	4335.3	27.0%	25.7	5.5%	173.3	8.9%	0.7
WI-Dlinear	2.7%	4149.3	3.8%	947.3	9.6%	2615.7	18.5%	18.5	4.8%	130.2	7.3%	0.6
WI-Nlinear	3.0%	4572.3	3.9%	984.6	9.7%	2633.1	18.4%	18.4	5.0%	134.9	7.5%	0.6
WI-DeepAR	1.6%	2461.2	2.3%	596.4	10.3%	2778.3	18.7%	19.4	4.0%	139.7	6.3%	0.6
WI-TFT	**1.1%**	**1702.6**	**1.9%**	**500.4**	**7.9%**	**2152.1**	**16.1%**	**17.0**	**3.1%**	**106.1**	**5.8%**	**0.5**

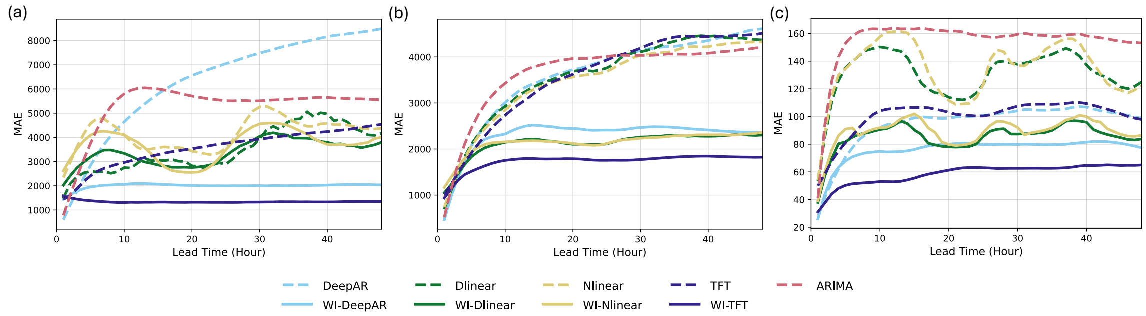


Figure 3.2: Total-level forecasting error (i.e., MAE) in MW versus lead time for the time series models for (a) load, (b) wind power generation, and (c) solar power generation.

loss, the middle quantiles are used as deterministic forecasts, as they are optimized for L1 loss [93, 4]. The forecasting accuracy of each model is evaluated using NMAE`ind` and RMSE`ind` to measure the average forecast accuracy across all individual series. Additionally, NMAE`s-sum` and RMSE`s-sum` are used to assess the forecasts for total load, and solar and wind power generation at the aggregate space-sum level.

Table Table 3.5 presents the forecast errors for MISO system on the test dataset. The performance of models with and without weather forecast information is compared to examine the impact of weather covariates. The models that utilize weather data are denoted with the prefix *WI* (i.e., weather informed) in Table Table 3.5.

The comparison of deterministic forecasting methods shows how the DL-based models improve over the traditional ARIMA model. These improvements are primarily attributed to the transition from ARIMA's univariate framework to a DL-based multivariate approach, which more accurately captures the complex dynamics within energy systems. The capacity of these models to effectively utilize cross-series information significantly enriches the forecasting process, offering a deeper understanding of energy patterns. The integration of weather forecast information into these models further enhances their accuracy. It is observed that WI models consistently outperform those that do not include weather data, highlighting the importance of weather covariates in predicting the variability of RES. This is most pronounced in the TFT model, where incorporating weather covariates yields a significant reduction in prediction error by 49% for load forecasting, 40% for wind energy prediction, and 34% for solar energy prediction on NMAE at an individual asset level. Similarly, adding covariantes improves RMSE at individual asset level by 30% to 50%.

Table 3.6: Probabilistic Forecast Evaluation.

Performance of the weather-informed (WI) models are evaluated on MISO dataset with and without inclusion of copula (i.e., mar. and cop., respectively) method for scenario generation. The metrics are calculated at space-sum (i.e., `s-sum`), time-sum (i.e., `t-sum`), and total levels. Lower values are better.

		ED`s-sum`		VS`s-sum`		VS`t-sum`		ED	
Target	Model	mar.	cop.	mar.	cop.	mar.	cop.	mar.	cop.
Load	WI-DeepAR	25632.1	23882.4	655547.1	629964.7	27518.5	**26209.7**	15834.4	15791.6
	WI-Dlinear	46952.4	43919.8	1253299.7	1222333.8	65596.7	63073.6	27159.3	27086.2
	WI-Nlinear	50969.3	46696.5	1452783.1	1406618.6	63242.8	60534.6	27566.3	27528.2
	WI-TFT	17928.8	**17553.7**	343455.7	**341995.2**	31011.6	30331.8	14299.4	**14277.0**
Wind	WI-DeepAR	29751.4	28845.1	1150740.6	1124880.2	7657926.0	7551511.4	5061.9	5036.3
	WI-Dlinear	32004.6	26113.7	1329397.4	1152174.8	5038233.2	4877483.1	4571.7	4574.6
	WI-Nlinear	32098.8	26033.3	1346814.8	1160315.4	4956035.6	4800996.2	4522.9	4526.2
	WI-TFT	25695.3	**24469.8**	1062437.8	**1028368.5**	4396148.6	**4268520.6**	**4452.2**	4455.5
Solar	WI-DeepAR	1509.4	1389.2	39688.5	39176.7	274696.7	275591.4	278.4	280.6
	WI-Dlinear	1598.7	1275.8	51144.3	49122.6	235188.6	**229894.0**	266.0	265.5
	WI-Nlinear	1657.9	1319.3	57429.9	54939.6	236787.0	230477.4	270.5	269.7
	WI-TFT	1231.4	**1145.9**	27296.2	**26915.7**	259657.2	253595.5	263.9	**263.4**

Moreover, the advantage of incorporating weather features becomes increasingly apparent for forecasts that extend further into the future, as depicted in Figure Figure 3.2. This trend is especially notable in wind energy forecasting: while models exhibit comparable accuracy for the initial 5-hour period, the performance of WI models remains robust beyond this window. In contrast, models lacking weather data experience a significant decline in performance over time. This pattern suggests that short-term predictions primarily rely on the current state of the system, whereas the accuracy of extended forecasts is increasingly contingent upon reliable weather information. A notable limitation of the linear models, such as DLinear and NLinear, is their inability to cope with the daily seasonality present in the data, which presents as a periodic pattern in the errors for load and solar power forecasting, as shown in Figure 3.2. Conversely, the WI-TFT model maintains consistent performance across different targets and various lead times, highlighting its superiority in reliable forecasting for RES.

The NREL dataset also comes with predictions [61]. Therefore, a comparison of prediction accuracy is made between the results of this chapter and NREL. Every effort has been made to ensure a fair comparison. However, it should be noted that NREL forecasts have an 11-hour lead time and are updated once a day. In contrast, the forecasts in this chapter are updated hourly without a lead time. This difference means that Table Table 3.5 contains 24 times more samples than the NREL forecasts. In the realm of load prediction, NREL forecasts exhibit a NMAE of 2.3% and a RMSE of 3708.0 at the aggregate level, and a NMAE of 3.3% alongside a RMSE of 885.6 at the individual level. The WI-TFT model for load prediction demonstrates a notable 52% improvement in NMAE at the aggregate level and a 4.4% improvement at the individual level over NREL predictions. When evaluating wind power predictions, the WI-TFT and NREL models show comparable performance at the individual level, with NREL registering a RMSE of 16.9. However, at the aggregate level, the WI-TFT model achieves a 44% better NMAE than the 11.4% from NREL. For solar power predictions, the WI-TFT model performs 70% worse at the individual level and 94% worse at the aggregate level in terms of NMAE compared to NREL. Despite this, both models exhibit high accuracy regarding the magnitude of errors, as evidenced in Table Table 3.5, with the WI-TFT model averaging only a 0.5 MW error per generator based on RMSE at individual level.

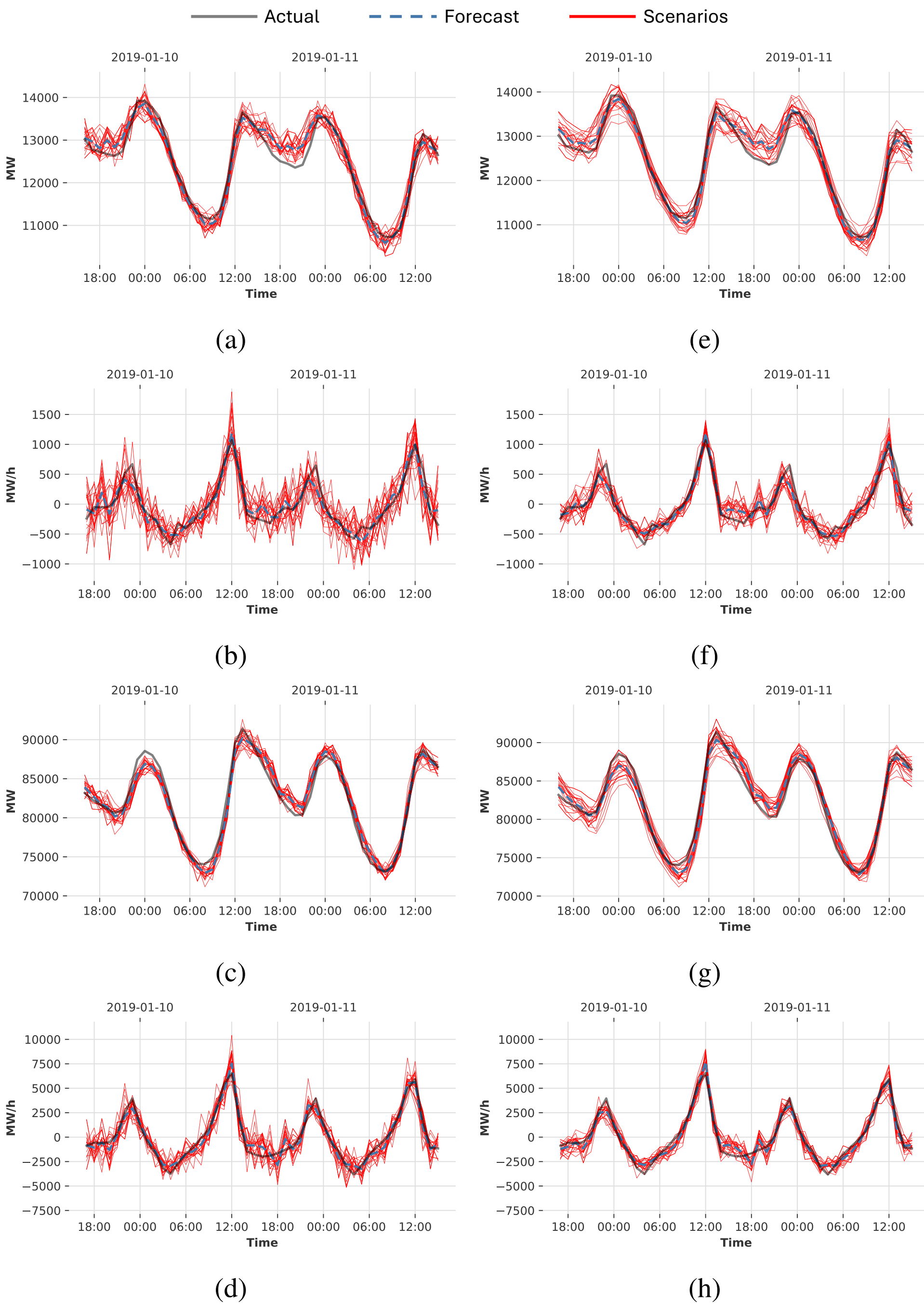


Figure 3.3: Comparison of Marginal and Copula Scenarios.
Generated scenarios for a load (LRZ1) using (a) marginals (i.e., no inclusion of copula) and (e) with the proposed copula method. (b) The ramps corresponding to the actuals and generated scenarios from marginals in (a). (f) The ramps for the actuals and generated scenarios using copula from (e). (c, d) The generated scenarios and the ramps for at total aggregated level using marginals and (g, h) using the proposed copula method.

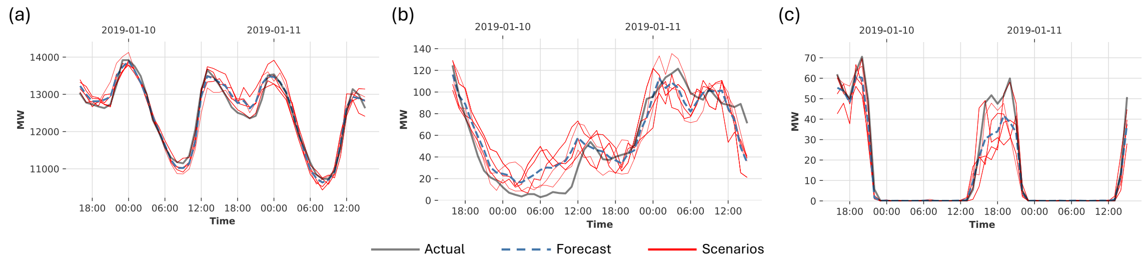


Figure 3.4: WI-TFT + Copula for Scenario Generation.
Examples of the forecast and 5 generated scenarios for (a) load, (b) wind, and (c) solar power generation. For simplicity, only 5 scenarios are plotted for each case.

3.7.2 Scenario Generation

This section presents the results for the probabilistic forecasting of RES using the DL-based WI models discussed in the previous sections. Models that do not include weather covariates are omitted, as it was demonstrated that they consistently underperform compared to the WI models. Specifically, the performance of the generated scenarios for load, wind power, and solar power generation using different models are compared with and without integration of Gaussian copula. The evaluation metrics are computed for scenarios generated from both marginal distributions (mar.) and joint distributions represented by copula (cop.). For each model and individual target, 200 scenarios are generated (i.e., $S = 200$) with and without the copula restoration. The performance of the models for generating scenarios are presented in Table Table 3.6.

The ED quantifies the distance between the distributions of the forecasted scenarios and the actual data. Lower ED values with the copula indicate improved alignment of generated scenarios with the true distribution. The ED for joint distribution and (s−sum)-level are calculated and presented in Table Table 3.6. Notably, as the sample sizes remain constant and the dimension tends to infinity, ED can only identify the equality of means and the traces of covariance matrices [94]. This accounts for the absence of differences in ED in Table Table 3.6, since the copula does not affect the means and variances of the marginal distribution. Comparing the $\text{ED}_{\texttt{s-sum}}$ with and without inclusion of the copula (i.e., cop. versus mar., respectively) shows that the copula model is better at capturing the joint distribution of the overall demand for the load and the overall wind and solar power generation with an improvement from 2% to 20% across different models.

The VS evaluates the correlation structure within the forecast scenarios. Due to the computational expense of VS for large dimensions, $\text{VS}_{\texttt{s-sum}}$ and $\text{VS}_{\texttt{t-sum}}$ are computed to capture the correlations across the spatial and temporal dimensions, respectively. The VS values are reduced by 0.4% to 5% upon integration of the copula, demonstrating its ability in improving the spatial and temporal correlations.

The WI-TFT model performs the best overall due to its superior ED and sufficiently low VS values. The lower ED indicates that the scenarios generated by WI-TFT are more accurate and closer to the true distribution, which is crucial for effective forecasting and decision making. It is important to note that VS does not capture the accuracy of the prediction and ED is more critical than VS in this context. Hence, ED and VS should be considered together for a comprehensive assessment of both the accuracy and reliability of the generated scenarios.

Figure 3.3 illustrates the comparative scenario analysis confirmed by the statistical data in Table 3.6. The left side of each plot displays various characteristics of scenarios generated from marginal distributions, while the right side depicts these characteristics after incorporating estimated correlations via the copula method. At the top of the figure, the plots Figure 3.3a and Figure 3.3e provide a 48-hour prediction for one load zone at the individual (`ind`)-level. The coverage of the scenarios remains consistent before and after the application of the copula. However, the copula-generated scenarios exhibit smoother temporal correlations, aligning more closely with the ground truth. This smoother correlation is particularly evident in the Figure 3.3b and Figure 3.3f, which shows the ramp rates of these scenarios. Here, the ramps of the copula models closely match the actual ramps, whereas those derived from marginal distributions show significant deviations.

Figures Figure 3.3g, Figure 3.3c, Figure 3.3d, and Figure 3.3h illustrate the same instance but at the space-sum (`s-sum`)-level. Initially, the copula scenarios display wider coverage compared to the marginal scenarios, effectively capturing some essential spatial correlations. This enhanced coverage ensures that the copula scenarios more accurately encompass the actual total-level outputs. Similar to the (`ind`)-level, the temporal correlations at the (`s-sum`)-level are more accurately represented in the copula scenarios, as demonstrated by the improved ramp alignment in the final row of the figure.

The conclusions drawn from the load scenarios apply similarly to wind and solar. Figure Figure 3.4 shows the effectiveness of WI-TFT with copula in generating realistic scenarios for individual load zones, wind farms, and solar farms. The copula method improves temporal and spatial correlations, better aligning the generated scenarios with the actual data compared to marginal distributions alone.

3.8 Conclusion

This chapter proposed a new method based on combination of TFT and Gaussian copula for high-dimensional probabilistic forecasting in RES. Extensive experiments were conducted to compare different time series forecasting methods on a real-world forecasting problem within the MISO system. The results demonstrated superiority of the TFT model compared to other statistical and DL-based methods, and the significant impact of including weather information on the accuracy and effectiveness of the forecasts. The integration of weather data as covariates also improved the precision of the models over higher lead times. Additionally, the paper highlighted the efficacy of the Gaussian copula method in restoring spatio-temporal correlations, which is crucial for generating realistic scenarios in RES forecasting. This methodological advancement provides a robust framework for addressing the complexities associated with high-dimensional forecasting problems in RES.

Future work will focus on studying the effect of noise in covariates on the performance of the forecasts and scenarios. Investigating how inaccuracies in weather data influence the model's predictions is essential for further improving the robustness of the forecasting method. Additionally, uncertainty quantification will be considered using conformal prediction to provide confidence-aware predictions in high-dimensional RES forecasting problems. This will involve developing techniques to quantify and incorporate prediction uncertainty, thereby offering more reliable and actionable insights for system operators.

CHAPTER 4

LLMFORECASTER: IMPROVING SEASONAL EVENT FORECASTS WITH UNSTRUCTURED TEXTUAL DATA [1]

4.1 Introduction

Time series forecasting has a broad variety of uses across industry today, including in transportation, energy, and retail settings. In modern retail settings, accurate demand forecasts are key to running an efficient supply chain. Improvements in forecast quality directly affect inventory efficiency, reducing stock shortages, and improving customer satisfaction.

In recent years, deep neural networks have become a powerful tool for forecasting at scale. Deep learning models such as Recurrent Neural Networks (RNNs) [95, 96], Convolutional Neural Networks (CNNs) [97], and attention-mechanisms [98], have shown promising results as they extract complex features and adapt to various time series patterns. These architectures are often designed to learn auto-correlations and cross-correlations from data [99, 100, 58]. These types of models have successfully integrated exogenous features (e.g., past covariates, known future information, and static covariates) and have achieved remarkable success in real-world problems, including traffic forecasting [101, 102], retail demand prediction [103, 6], power generation prediction [104], and energy consumption modeling [105].

While these models can effectively incorporate numerical or categorical exogenous features, other valuable features like product descriptions or customer reviews exist only as unstructured text. Because this information exists only as unstructured text which is difficult to featurize, these sources of information have been largely neglected or featurized in

[1]This chapter is based on the following preprint: Zhang, Hanyu, Chuck Arvin, Dmitry Efimov, Michael W. Mahoney, Dominique Perrault-Joncas, Shankar Ramasubramanian, Andrew Gordon Wilson, and Malcolm Wolff. "LLMForecaster: Improving seasonal event forecasts with unstructured textual data." *arXiv preprint arXiv*:2412.02525 (2024).

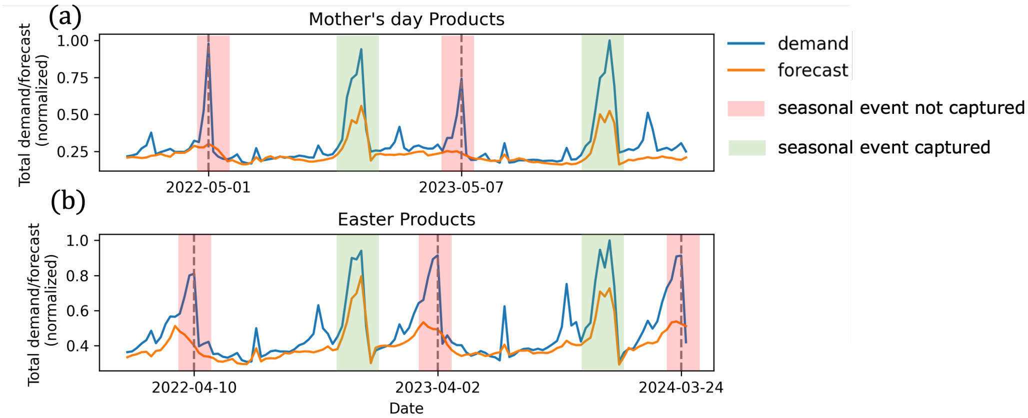


Figure 4.1: Aggregated demand and forecast for groups of products.
(a) Mother's Day products; and (b) Easter products. The vertical dashed lines mark the week prior to the holiday in question. In green, we show time segments where the production model anticipates event-driven demand surges — specifically large shopping events like Christmas. In red, we show time segments where the production model fails to anticipate event-driven demand surges.

simple ways [58, 106, 59, 4].

Nonetheless, descriptive and contextual information about the time series may meaningfully enhance forecast quality. In the retail setting, unstructured text describing the use and design of a product contains valuable information about whether this product will see surges in demand for upcoming seasonal events (holidays, back-to-school, etc.). This information is often difficult or impossible to learn from the time series themselves, for several reasons. Many products are new, with perhaps only one (or *fewer*) years of sales history. Sales history at the product level is also noisy, subject to spurious spikes and stockouts which distort the historical sales. These factors mean that the available historical sales are often insufficient to predict upcoming seasonal patterns reliably.

In Figure 4.1, we illustrate this problem, using forecasts from an MQ-Transformer model [103] trained on a retail dataset. We focus on products which are relevant to two key holiday related seasonal events: Mother's Day and Easter. For both groups of products, we see that the existing forecasting model appropriately anticipates surges in demand during the holiday season (between Black Friday and Christmas). The model responds reasonably to the holiday season (green bands), where we see elevated sales across many products, high customer traffic and numerous price discounts. By contrast, the model fails to anticipate surges in demand during the Mother's Day and Easter holiday periods themselves (red bands). Notably, the demand surges during Mother's Day and Easter are localized to small groups of products, aggregate customer traffic does not spike, and products are infrequently discounted. The failure to anticipate these localized surges in demand increases the risk of stockouts for key products during these events. Today, these defects must be addressed through human intervention, relying on human analysts to use their judgment and knowledge to identify products relevant to an upcoming event and adjust the forecasts accordingly.

Recent advancements in Large Language Models (LLMs) offer a promising avenue to address these challenges, enhancing predictive accuracy by combining rich textual information with traditional covariates. Gruver et al. [54] first showed that even LLMs with text-based pre-training can perform impressive zero-shot time series forecasting. Moreover, pre-trained LLMs have the capability to perform domain-specific predictive tasks by directly querying them with domain-specific instructions and knowledge [107, 108, 109, 110]. Xue et al. [55] extended this approach by generalizing prediction tasks to time series data, incorporating context and semantic information from historical data. LLM4TS[56] utilizes a two-stage fine-tuning process to improve the model's ability to handle time series data, even with limited data availability. Similarly, TEMPO [57] applies a Generative Pre-trained Transformer (GPT) to time series forecasting, using a prompt-based approach that tailors the model to complex temporal patterns and non-stationary data. Most existing research in time series forecasting has focused on using time series data as input to LLMs, exploring methods like tokenization of time series data [58, 59].

Despite these advancements, it remains an open question how to develop LLMs to integrate descriptive information about the time series themselves — for example, the description of the product corresponding to the sales in question. Further, these methods are often stand-alone forecasting models. This can be powerful, but in other real-world use cases we may already have a "good enough" forecasting solution in place. In those cases, we do not want to completely replace the existing solution — instead we only want to modify the forecasts to fix areas where the existing models has clear deficiencies.

Our work addresses both of these areas. Here we introduce a procedure, the *LLMForecaster*, to incorporate unstructured textual information and recommend forecast adjustments to improve the accuracy of an existing forecasting pipeline. Our procedure utilizes fine-tuned LLMs that incorporate both historical forecasts as well as unstructured information. This allows us to systematically improve forecast quality in cases where the existing model fails to anticipate holiday related demand surges. We show that our approach enhances the accuracy of demand forecasts in retail environments, empowering businesses to better manage seasonal fluctuations and optimize their operations.

4.2 Methodology

We introduce the LLMForecaster, designed to automate forecast adjustments to correct biases in existing demand forecasting models. Here, the existing model is an MQ-Transformer model trained on a large dataset of retail sales. This existing model generates initial predictions, denoted as $f_{i,t}$ for product i and target date t. We then train an additional model, incorporating unstructured text information such as product titles and descriptions ($\mathbf{x}_{i,t,\text{text}}$) and numeric features such as the price or forecast ($\mathbf{x}_{i,t,\text{num}}$) via prompts to an LLM. The model architecture is shown in Figure 4.2.

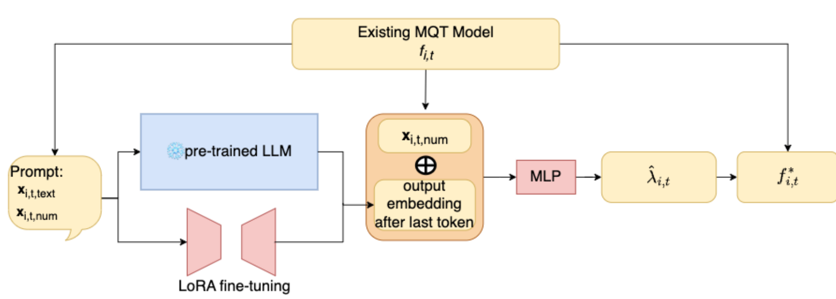


Figure 4.2: LLMForecaster incorporates text and numeric information through an LLM to rescale the raw prediction $f_{i,t}$, producing a better forecast $f^*_{i,t}$.

This additional model is trained to predict the scaling factor $\lambda_{i,t} = \log(\frac{y_{i,t}}{f_{i,t}})$, where $y_{i,t}$ represents the actual observed demand. We choose to predict a scaling factor, as it allows us to make improvements on top of an existing primary model, and helps deal with heavy-tailed response data. [111]. The model minimizes the absolute error between the true and predicted scaling factors:

$$\min_{\hat{\lambda} \in \Lambda} \sum_{i=1}^{n} \left| \lambda_{i,t} - \hat{\lambda}(\mathbf{x}_{i,t,\text{text}}, \mathbf{x}_{i,t,\text{num}}) \right| \tag{4.1}$$

where n is the number of samples, and Λ is the model space.

We then use this scaling factor to rescale the original predictions $f_{i,t}$ into the adjusted forecast $f^*_{i,t}$, using the following transformation.

$$f^*_{i,t} = e^{\hat{\lambda}(\mathbf{x}_{i,t,\text{text}}, \mathbf{x}_{i,t,\text{num}})} f_{i,t} \tag{4.2}$$

The model relies on an input prompt that contains product information, forecast values, and other contextual information formatting using a predefined template. An example of such a prompt is shown in subsection 4.2.1. The fine-tuned LLM generates an embedding vector, which is then adapted to the specific forecasting task using Low-Rank Adaptation (LoRA) [60]. This adapted embedding is concatenated with numerical features and fed into a Multi-Layer Perceptron (MLP) head, which outputs the scaling factor $\hat{\lambda}_{i,t}$. By fine-tuning the LLMForecaster on historical forecasts and demand, the model learns to identify patterns in forecast errors and provide accurate adjustments to the primary model's predictions, especially for products with significant holiday demand surges.

4.2.1 Prompting details

The following template is used to apply on the provided text features and numerical features. The part that is being inserted based on the data is indicated with blue square brackets, []. As discussed in subsection 4.2.2, we also include information if the target date is in the near vicinity of a particular holiday.

> Pretend you are a sales analyst. You need to adjust the current model's prediction for a specific product's sales week of [forecast target date].
>
> ### Instruction:
>
> Today's date: [forecast creation date].
>
> Product Title: [product title]
>
> List Price: $[list price].
>
> Item Created At: [item creation date].
>
> Product Group Type: [product group]
>
> Description: [product description]
>
> Bullet Points: [bullet points]
>
> The current prediction for week [forecast target date]'s sales is [p50] units with a 90th percentile [p90] units. Please provide your adjusted prediction for next week's sales volume considering today's date, the season, the holiday, and the product. Explain your reasoning.
>
> ### Response Format: - Prediction: [Your Adjusted Prediction] units - Reasoning: [Explanation]
>
> ### Response:

4.2.2 Holiday encoding prompt

To improve the LLMForecaster's ability to capture holiday-driven demand patterns, we have implemented a "Holiday-Encoding Prompt". This prompt provides the model with contextual information about the temporal relationship between the target forecast date and surrounding holidays. For example, for a forecast targeting the week of June 1, 2024, the prompt would include:

> The mother's day at 2024-05-12 is 3 weeks before 2024-06-01.
>
> The father's day at 2024-06-16 is 2 weeks after 2024-06-01.

By including these details about the proximity of the target date to relevant holidays, we aim to help the model better identify the appropriate demand patterns. This helps by ensuring the LLM knows precisely when a given event will take place. This is especially important for ”moving holidays” like Easter, where the exact date can vary by as much as 35 days from year to year. This Holiday-Encoding Prompt is expected to significantly improve the LLMForecaster’s performance in accurately predicting sales for various holiday periods. As shown in Figure 4.3, the proposed LLMForecaster with the Holiday-Encoding Prompt is able to identify the Easter spike at the end of March 2024, while removing the prompt fails to capture the Easter-related demand surge.

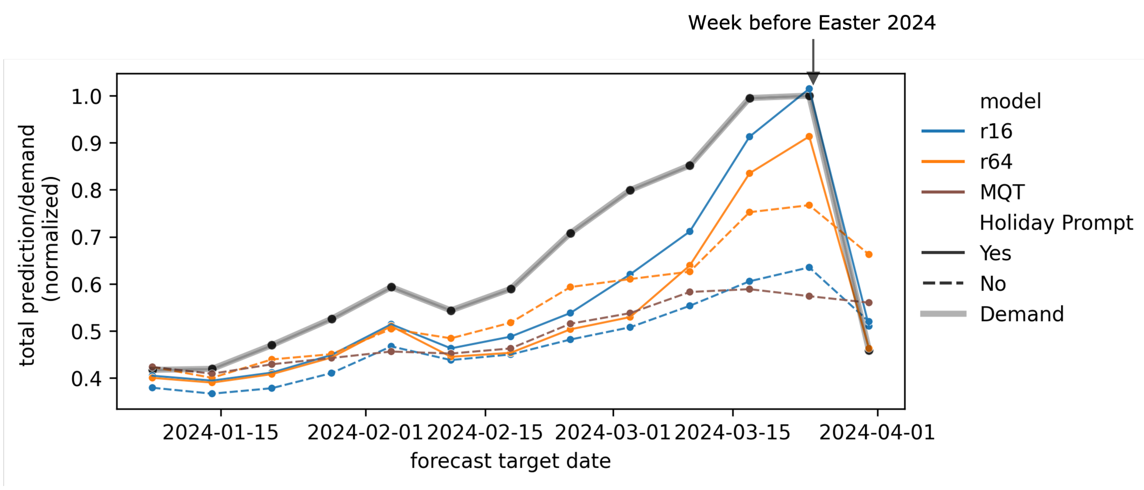


Figure 4.3: Total demand and prediction for Easter products with and without *Holiday-Encoding Prompt*

4.3 Experiment Results

We apply the LLMForecaster to refine predictions made by the existing global model, `MQ-Transformer` (`MQT`), at a lead time of 12 weeks. This lead time is selected to provide sufficient time to procure inventory in advance of holiday surges. The various features are processed into a prompt and fed into the pre-trained `MPT7b-Instruct` model[112], which is further fine-tuned for the forecasting task. Performance is evaluated using the weighted p_{50} quantile loss (wQL), defined as: wQL $= \sum_{i,t} 0.5|f_{i,t} - y_{i,t}|/\sum_{i,t} y_{i,t}$ where

i is index of product and t is the forecast target date.

In this experiment, we train a single model capable of calibrating demand predictions across multiple holidays. We focus on five holidays known for strong seasonality: Halloween, Easter, Mother's Day, Father's Day, and Valentine's Day. The training dataset spans 88 weeks, from August 29th, 2021, to April 30th, 2023, including both holiday-related and non-holiday products. A holiday-related product is one with the holiday name in the item name or product description. The test period covers 48 weeks from May 7th, 2023, to March 3rd, 2024, and is divided into five distinct test sets, one for each target holiday. To ensure the LLM knows when events happen, we use a "Holiday-Encoding Prompt" that provides the LLM with the proximity of the target date to the relevant holiday (subsection 4.2.2).

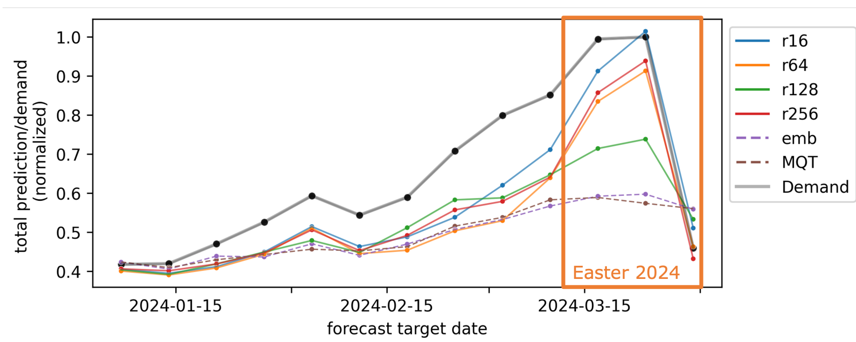


Figure 4.4: Example of aggregated forecasts on Easter products.

Figure 4.4 compares the aggregated forecasted demand with actual demand for Easter products. It shows several LLMForecaster models (`r16`, `r64`, `r128`, and `r256`, representing varying LoRA ranks), the `emb` baseline (where we do not apply LoRA fine-tuning) and the original `MQT` baseline. All iterations of the LLMForecaster approach anticipate the Easter demand surge, while our two baselines fail to do so. Table 4.1 presents wQL improvement results for the 48-week test sets demonstrating that the fine-tuned LLMForecaster models (`r16`, `r64`, `r128`, and `r256`) consistently outperform the baseline `MQT` and `emb` models across all five holiday datasets. We also conduct statistical significance testing of the improvement throughout the year - most of these improvements are statistically significant.

By contrast, the `emb` model, in which we do not do the LoRA fine-tuning, shows no improvement over the `MQT` baseline. This underscores the importance of the LLMForecaster's sophisticated fine-tuning approach, leveraging LoRA fine-tuning to effectively learn the holiday-specific demand patterns.

Table 4.1: wQL improvement over the `MQT` baseline (in basis points) for different testsets.

Significance Levels: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

Model	Halloween	Easter	Father's Day	Mother's Day	Valentine's Day
`r16`	**105*****	51***	**93*****	68***	**58*****
`r64`	103***	**60*****	88***	**70*****	33
`r128`	77***	32**	39***	10**	18
`r256`	70***	34*	48***	52**	17*
`emb`	8	−5	11**	−14	−16

4.3.1 Accuracy results throughout the year

Here we show how the LLMForecaster changes forecast accuracy for different groups of products throughout the year. First, we show the results from a t-test measuring the weekly change in quantile loss, over the 48 weeks in the test period. Across the various product sets, we generally see statistically significant improvements over the `MQT` baseline. The only exception is for Valentine's Day products, which show improvements which are not statistically significant.

Table 4.2: pairwise t-tests of wQL for different testsets and models.

Significance Levels: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

	Halloween		Easter		Father's day		Mother's day		Valentine's day	
	`MQT`	`emb`	`MQT`	`emb`	`MQT`	`emb`	`MQT`	`emb`	`MQT`	`emb`
`r16`	−5.4***	−5.5***	−3.6***	−3.7***	−5.0***	−3.6***	−3.8***	−4.2***	−3.6***	−4.4***
`r64`	−5.3***	−5.7***	−3.8***	−4.1***	−4.7***	−3.6***	−3.7***	−4.0***	−1.7	−2.2*
`r128`	−4.2***	−4.4***	−3.4**	−3.6***	−3.9***	−2.3*	−2.4**	−2.7**	−1.9	−2.7**
`r256`	−4.2***	−4.3***	−2.3*	−2.5*	−3.5***	−2.2*	−3.5**	−3.9***	−2.4*	−3.1**
`emb`	−1.2	-	0.6	-	−3.1**	-	0.2	-	2.0	-

Next, we show the change in forecast accuracy from our LLMForecaster model versus the `MQT` baseline. Positive numbers indicate weeks in which the LLMForecaster model improved accuracy over the existing baseline. For each group examined, the LLMForecaster generally improves accuracy throughout the year.

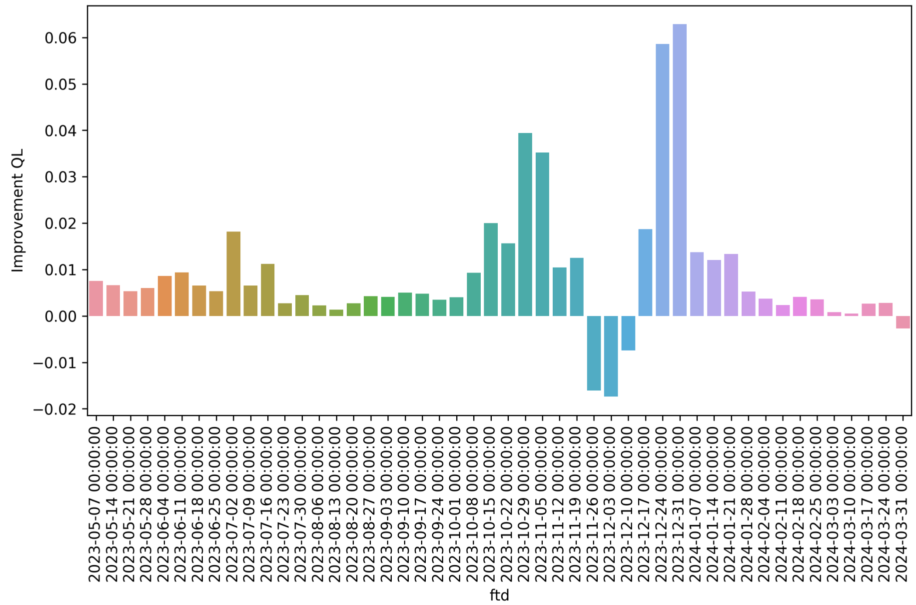


Figure 4.5: Forecast accuracy change for Halloween products

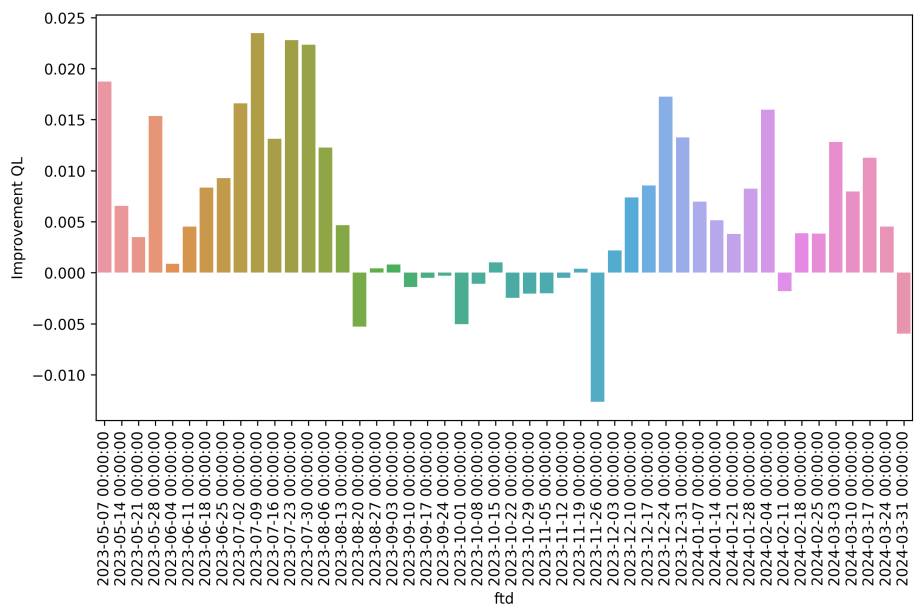


Figure 4.6: Forecast accuracy change for Mother’s Day products

4.3.2 Valentine’s Day

The results shown in Table 4.1 and Table 4.2 indicate a relatively small and sometimes not statistically significant improvement for the Valentine’s Day products compared to the

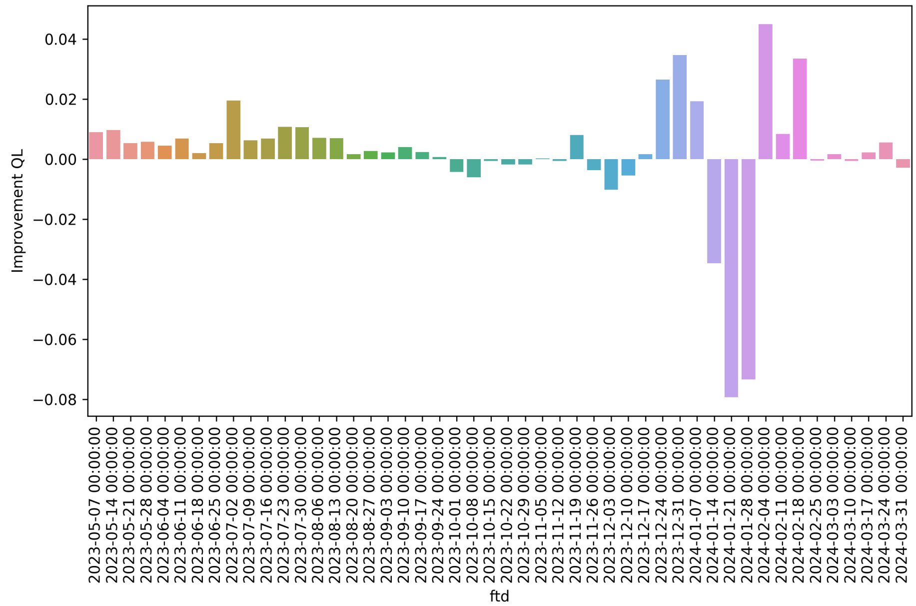


Figure 4.7: Forecast accuracy change for Valentine's Day products

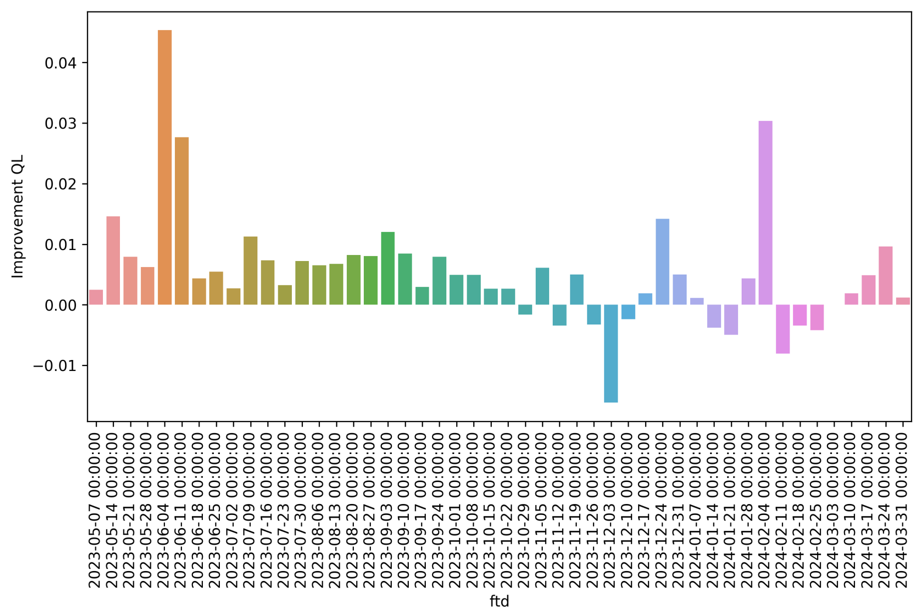


Figure 4.8: Forecast accuracy change for Father's Day products

other holiday categories. This can be attributed to the unique nature of the Valentine's Day holiday and its shifting position within the calendar week. For other holidays like Halloween, Easter, Mother's Day, and Father's Day, the dates are fixed on either Saturdays or Sundays, so the overall demand distribution during the holiday period remains relatively consistent. By contrast, Valentine's Day is fixed on February 14th, which can fall on differ-

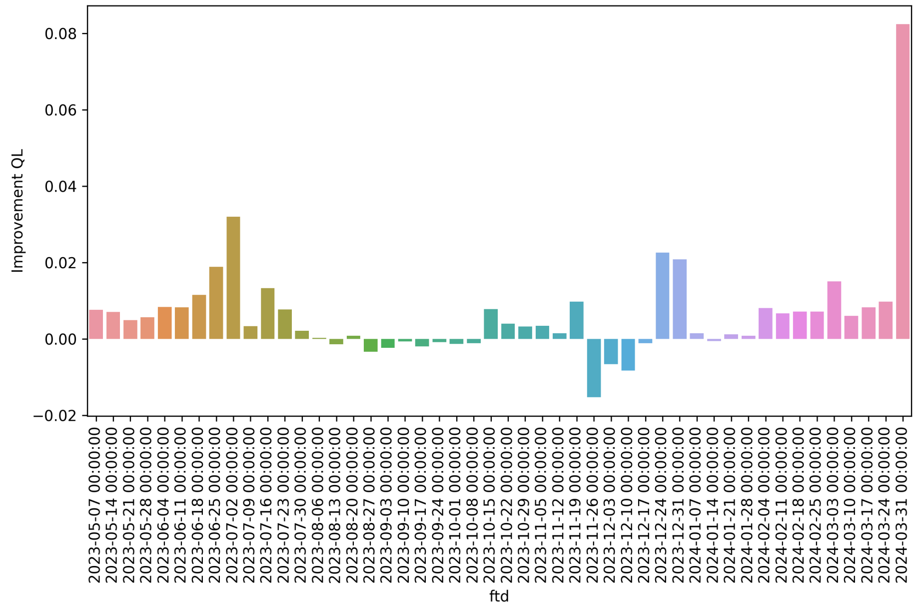


Figure 4.9: Forecast accuracy change for Easter products

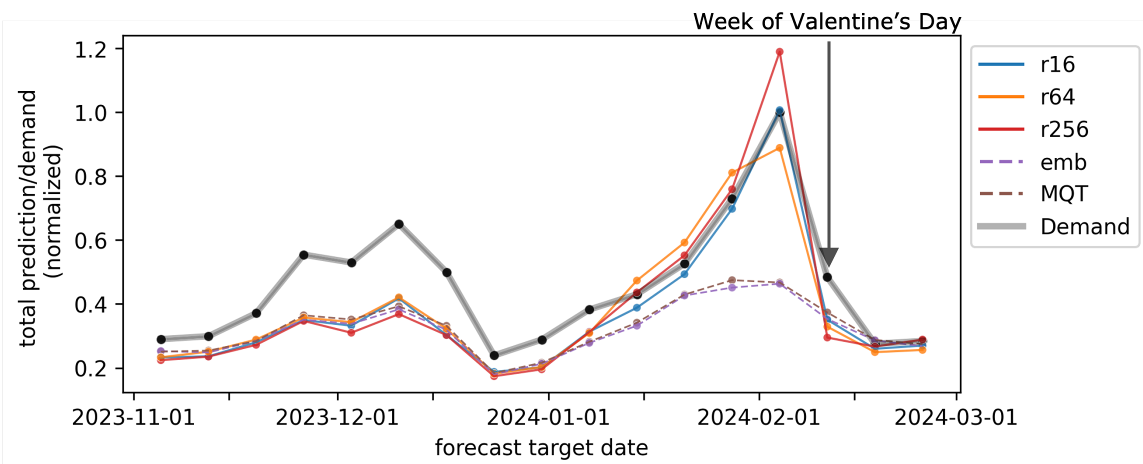


Figure 4.10: Total demand prediction for Valentine's Day products

ent days of the week. That means that last-minute shopping, for example, may take place in the week of Valentine's Day or the prior week. In this experiment, the training dataset contained Valentine's Day falling on Monday or Tuesday - with minimal time to shop during the week of the holiday, last-minute shopping took place primarily in the prior week. In the test set, Valentine's Day occurring on a Wednesday, so consumers had more time to shop for the holiday during the week itself. When plotting the total prediction and demand on Figure 4.10, this effect is clearly observed. While the LLMForecaster models are able to capture the Valentine's Day demand spike compared to the baseline models, they tend to over-predict the weeks before Valentine's Day and significantly under-predict the demand during the actual Valentine's Day week in 2024. This issue could potentially be addressed by training the model with data spanning multiple years, or by incorporating daily demand patterns into the training process. This would help the LLMForecaster better account for the shifting position of Valentine's Day within the calendar week and improve its ability to accurately predict the associated demand patterns.

4.4 Conclusion and Future Work

This chapter introduced LLMForecaster, a novel framework that bridges the gap between unstructured textual data and traditional time series forecasting. By leveraging large language models to interpret product-level information and implementing targeted forecast adjustments, LLMForecaster addresses a critical limitation in existing forecasting approaches. Our extensive empirical evaluation demonstrates that this approach yields statistically significant improvements in demand forecasting accuracy, particularly for products with strong seasonal patterns and holiday-driven demand surges.

The success of LLMForecaster opens several promising directions for future research. First, the framework could be extended to incorporate more sophisticated prompting techniques that better capture complex relationships between product descriptions and demand patterns. Second, there is potential to expand the role of LLMs from post-processing to feature engineering, potentially using them to generate rich embeddings of textual data as inputs to deep learning models. Finally, the framework could be enhanced to handle multimodal inputs, such as product images and customer reviews, providing a more comprehensive understanding of factors affecting demand. These extensions would further advance the integration of unstructured information into quantitative forecasting systems, potentially improving accuracy across a broader range of applications and domains.

Appendices

APPENDIX A

EVALUATION METRICS

In this section, the details of evaluation metrics are provided.

A.1 Deterministic Forecast

In the case of point forecast, $S = 1$ and only one scenario/forecast is generated. For notational simplicity, the scenario index s is dropped in this subsection.

Normalized Mean Absolute Error (NMAE) For wind and solar power, the ground truth can sometimes become zero or close to zero. To avoid dividing by zeros, NMAE is considered. The first metric considers the forecast on each space-time dimension:

$$\text{NMAE}_{\texttt{ind}} = \frac{1}{N_{\text{test}} D T} \sum_{t=1}^{N_{\text{test}}} \sum_{i \in \mathcal{D}, \tau \in \mathcal{H}} \left| \frac{\hat{z}_{i,\tau} - z_{i,\tau}}{\max\{\mathbf{z}_{i,\mathcal{T}_{\text{train}}}\}} \right|, \tag{A.1}$$

expressed as a percentage. The $\text{NMAE}_{\texttt{ind}}$ reflects the average performance on individual time series.

The normalization value $z_{\max\{\mathbf{z}_{i,\mathcal{T}}\}}$ is the capacity of the generator if given. If the capacity is not given, it is the maximum value of the first year in the training dataset.

The second metric considers summation over space and is defined as:

$$\text{NMAE}_{\texttt{s-sum}} = \frac{1}{N_{\text{test}} T} \sum_{t=1}^{N_{\text{test}}} \sum_{\tau \in \mathcal{H}} \left| \frac{\sum_{i=1}^{D} \hat{z}_{i,\tau} - \sum_{i=1}^{D} z_{i,\tau}}{\sum_{i=1}^{D} \max\{\mathbf{z}_{i,\mathcal{T}_{\text{train}}}\}} \right|. \tag{A.2}$$

Root Mean Square Error (RMSE) The RMSE is a commonly used metric to measure the accuracy of a model by calculating the square root of the average squared differences between the predicted and actual values. It provides a measure of the average magnitude of the error. The first metric considers the forecast on each space-time dimension:

$$\mathrm{RMSE}_{\texttt{ind}} = \sqrt{\frac{1}{N_{\text{test}} DT} \sum_{t=1}^{N_{\text{test}}} \sum_{i \in \mathcal{D}, \tau \in \mathcal{H}} \left(\hat{z}_{i,\tau} - z_{i,\tau}\right)^2}, \tag{A.3}$$

expressed in megawatts (MW). The $\mathrm{RMSE}_{\texttt{ind}}$ reflects the average performance on individual dimension.

The second metric considers summation over space and is defined as:

$$\mathrm{RMSE}_{\texttt{s-sum}} = \sqrt{\frac{1}{N_{\text{test}} T} \sum_{t=1}^{N_{\text{test}}} \sum_{\tau \in \mathcal{H}} \left(\sum_{i=1}^{D} \hat{z}_{i,\tau} - \sum_{i=1}^{D} z_{i,\tau} \right)^2}, \tag{A.4}$$

also expressed in MW. The $\mathrm{RMSE}_{\texttt{s-sum}}$ measures the model's accuracy in predicting the summation of wind power, solar power, or load over space.

A.2 Probabilistic Forecast

Energy Distance The energy distance is a popular metric for evaluating the quality of probabilistic forecasts [113]. Given two probability distributions $\mathcal{F}$ and $\mathcal{G}$ with support in $\mathbb{R}^D$, the energy distance between $\mathcal{F}$ and $\mathcal{G}$ is defined as

$$\mathrm{ED}(\mathcal{F}, \mathcal{G}) = 2\mathbb{E}\left\|X - Y\right\| - \mathbb{E}\left\|X - X'\right\| - \mathbb{E}\left\|Y - Y'\right\|, \tag{A.5}$$

where $\|\cdot\|$ is the Euclidean norm, and $X \sim \mathcal{F}, X' \sim \mathcal{F}, Y \sim \mathcal{G}, Y' \sim \mathcal{G}$ are independent random variables. In particular, when $\mathcal{F}$ and $\mathcal{G}$ are empirical probability distributions with realizations $\mathbf{x} = \{x_1, ..., x_n\}$ and $\mathbf{y} = \{y_1, ..., y_m\}$, an estimator of the energy distance using the U-statistics is given by:

$$\hat{\text{ED}}(\mathbf{x},\mathbf{y}) = \frac{2}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}\|x_i - y_j\| - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|x_i - x_j\| - \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\|y_i - y_j\|. \quad \text{(A.6)}$$

Note that, for $m = 1$ case, the third term in the above summation is zero.

Here, two energy distance metrics are considered: (i) At time step t, the D-by-T matrix $z_{\cdot,\mathcal{H}}$ is vectorized by stacking the columns of the matrix on top of one another. The resulting DT-dimensional vector is denoted by $\text{vec}(z_{\cdot,\mathcal{H}})$. There are S generated forecasts, $\hat{z}^{(s)}_{\cdot,\mathcal{H}}, s = 1, \dots, S$ and the quality of those forecasts are measured by

$$\hat{\text{ED}}(\text{vec}(z_{\cdot,\mathcal{H}}), \{\text{vec}(\hat{z}_{\cdot,\mathcal{H}})^{(s)}, s = 1, \dots, S\}) = \frac{2}{S}\sum_{s=1}^{S}\left\|z_{\cdot,\mathcal{H}} - \hat{z}^{(s)}_{\cdot,\mathcal{H}}\right\|_F - \frac{1}{S^2}\sum_{s_1,s_2=1}^{S}\left\|\hat{z}^{(s_1)}_{\cdot,\mathcal{H}} - \hat{z}^{(s_2)}_{\cdot,\mathcal{H}}\right\|_F,$$

where $\|\cdot\|_F$ denotes the matrix F-norm and $\|\text{vec}(\cdot)\| = \|\cdot\|_F$. In the following experiments, given NS probabilistic forecasts produced, the average over $t = 1, \dots, N$ is taken as the first metric, which is also called the energy score

$$\text{ED}_{\texttt{ind}} = \frac{1}{N_{\text{test}}}\sum_{t\in\mathcal{N}_{\text{test}}}\hat{\text{ED}}(\text{vec}(z_{\cdot,\mathcal{H}}), \{\text{vec}(\hat{z}_{\cdot,\mathcal{H}})^{(s)}, s = 1, \dots, S\}). \quad \text{(A.7)}$$

(ii) The energy distance computed on space-sum level is given by

$$\hat{\text{ED}}(\mathbf{1}^{\text{T}}z_{\cdot,\mathcal{H}}, \{\mathbf{1}^{\text{T}}\hat{z}^{(s)}_{\cdot,\mathcal{H}}, s = 1, \dots, S\}) = \frac{2}{S}\sum_{s=1}^{S}\left\|\mathbf{1}^{\text{T}}z_{\cdot,\mathcal{H}} - \mathbf{1}^{\text{T}}\hat{z}^{(s)}_{\cdot,\mathcal{H}}\right\| - \frac{1}{S^2}\sum_{s_1,s_2=1}^{S}\left\|\mathbf{1}^{\text{T}}\hat{z}^{(s_1)}_{\cdot,\mathcal{H}} - \mathbf{1}^{\text{T}}\hat{z}^{(s_2)}_{\cdot,\mathcal{H}}\right\|.$$

The second metric, referred to as average space-sum energy score:

$$\text{ED}_{\texttt{s-sum}} = \frac{1}{N_{\text{test}}}\sum_{t\in\mathcal{N}_{\text{test}}}\hat{\text{ED}}(\mathbf{1}^{\text{T}}z_{\cdot,\mathcal{H}}, \{\mathbf{1}^{\text{T}}\hat{z}_{\cdot,\mathcal{H}}(s), s = 1, \dots, S\})$$

Recall that forecasts are produced for multiple time series and time periods. Therefore, the energy distance in Eq. (Equation A.7) captures both the spatial and temporal quality of the forecasts.

Variogram Score The variogram score (VS) captures the correlations among multivariate quantities [114]. Given a probabilistic forecast at time t, the corresponding variogram score of order p is defined as

$$\mathrm{VS}_p(t) = \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{\tau_1 \in \mathcal{T}} \sum_{\tau_2 \in \mathcal{T}} \left(|z_{i,\tau_1} - z_{j,\tau_2}|^p - \frac{1}{S} \sum_{s=1}^{S} \left| \hat{z}_{i,\tau_1}^{(s)} - \hat{z}_{j,\tau_2}^{(s)} \right|^p \right). \tag{A.8}$$

In all the experiments that follow, the variogram score is evaluated with $p = 0.5$, following the recommendations in [114].

As can be seen from Equation A.8, computing the variogram score for one time period has complexity $\mathcal{O}(D^2T^2S)$. This is computationally intractable for high dimensional setting, where $D \geq 300$, $T = 48$, and $S = 200$ for two days ahead solar power forecast. Therefore, variogram scores are reported for summation of load and renewable generation predictions over space. This captures temporal correlations (between time periods), at the expense of spatial correlations (between time series).

$$\mathrm{VS}_{p,\texttt{s-sum}}(t) = \sum_{\tau_1 \in \mathcal{T}} \sum_{\tau_2 \in \mathcal{T}} \left(\left| \sum_{i=1}^{D} z_{i,\tau_1} - \sum_{i=1}^{D} z_{i,\tau_2} \right|^p - \frac{1}{S} \sum_{s=1}^{S} \left| \sum_{i=1}^{D} \hat{z}_{i,\tau_1}^{(s)} - \sum_{i=1}^{D} \hat{z}_{i,\tau_2}^{(s)} \right|^p \right). \tag{A.9}$$

Averaging over the test set, the average space-sum variogram score is given as

$$\mathrm{VS}_{\texttt{s-sum}} = \frac{1}{N_{\text{test}}} \sum_{t=1}^{N_{\text{test}}} \mathrm{VS}_{p,\texttt{s-sum}}(t). \tag{A.10}$$

Another metric inter-changes the order of the summation and is defined as follows:

$$\mathrm{VS}_{p,\texttt{t-sum}}(t) = \sum_{i=1}^{D} \sum_{j=1}^{D} \left(\left| \sum_{\tau \in \mathcal{T}} z_{i,\tau} - \sum_{\tau \in \mathcal{T}} z_{j,\tau} \right|^p - \frac{1}{S} \sum_{s=1}^{S} \left| \sum_{\tau \in \mathcal{T}} \hat{z}_{i,\tau}^{(s)} - \sum_{\tau \in \mathcal{T}} \hat{z}_{j,\tau}^{(s)} \right|^p \right). \tag{A.11}$$

Averaging over the test set, the average time-sum variogram score is given as

$$\mathrm{VS}_{\texttt{t-sum}} = \frac{1}{N_{\text{test}}} \sum_{t=1}^{N_{\text{test}}} \mathrm{VS}_{p,\texttt{t-sum}}(t). \tag{A.12}$$

A.3 Marginal Distributions

When estimating marginal distributions, researchers typically employ either parametric or non-parametric approaches. This section introduces two prevalent methods: quantile regression and parameterized distribution estimation.

Non-Parametric Estimation: Quantile Regression The quantile regression approach, as outlined in Wen et al.[4], utilizes the concept of quantile loss in multistep multivariate settings. Consider $\mathcal{Q}$ to represent a predefined set of quantiles. For any given target quantity in space-time, denoted by $z_{i,\tau}$, our model generates a corresponding set of quantile estimates $\hat{z}_{i,\tau}^{(q)}$, where $q \in \mathcal{Q}$. The primary goal is to minimize the overall quantile loss function across all estimated quantiles, $\hat{z}_{i,\tau}^{(q)}$, formulated as follows:

$$\min \sum_{t} \sum_{\tau \in \mathcal{H}} \sum_{q \in \mathcal{Q}} \sum_{i \in \mathcal{D}} \mathrm{QL}_q \left(z_{i,\tau}, \hat{z}_{i,\tau}^{(q)} \right),$$

where t iterates over all forecast creation times.

The quantile loss (QL) is defined as

$$\mathrm{QL}_q \left(z_{i,\tau}, \hat{z}_{i,\tau}^{(q)} \right) = q \left(z_{i,\tau} - \hat{z}_{i,\tau}^{(q)} \right)^{+} + (1-q) \left(\hat{z}_{i,\tau}^{(q)} - z_{i,\tau} \right)^{+}, \tag{A.13}$$

where $(\cdot)^{+} = \max(0, \cdot)$. This paper considers the quantiles $q \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

Once the quantiles $\hat{z}_{i,\tau}^{(q)}$ are available, the CDF function $\hat{F}_{i,\tau}(z_{i,\tau})$ can be estimated from fitting a smooth curve across the predicted quantiles.

Parametric Estimation: Maximum Likelihood Estimation (MLE) The Beta MLE approach is utilized for estimating the parameters of a Beta distribution, typically denoted as α (alpha) and β (beta). This method is particularly beneficial when the data is understood to follow a Beta distribution, often applicable to variables constrained between 0 and 1. For RES, it is common to scale the time series based on their capacity to capacity factors, therefore, beta distribution is chosen to demonstrate the parametric estimation.

For a set of observations $z_{i,\tau}$ corresponding to each forecast time τ within the forecast horizon $\mathcal{H}$ and across all dimensions $i \in \mathcal{D}$, assuming these observations are drawn from a Beta distribution, the aim of Beta MLE is to find the parameter values that maximize the likelihood of the given sample.

The likelihood function for the Beta distribution, given the observations, is:

$$\max_{\alpha_{i,\tau},\beta_{i,\tau}} \prod_{t} \prod_{\tau \in \mathcal{H}} \prod_{i \in \mathcal{D}} \frac{z_{i,\tau}^{\alpha_{i,\tau}-1}(1-z_{i,\tau})^{\beta_{i,\tau}-1}}{B(\alpha_{i,\tau},\beta_{i,\tau})},$$

where t iterates all forecast creation times and $B(\alpha_{i,\tau},\beta_{i,\tau})$ is the Beta function.

Optimization is typically performed on the log-likelihood,

$$\max_{\alpha_{i,\tau},\beta_{i,\tau}} \sum_{t} \sum_{\tau \in \mathcal{H}} \sum_{i \in \mathcal{D}} [(\alpha_{i,\tau}-1)\ln(z_{i,\tau}) + (\beta_{i,\tau}-1)\ln(1-z_{i,\tau})] - DH\ln(B(\alpha_{i,\tau},\beta_{i,\tau})). \tag{A.14}$$

The $\hat{\alpha}_{i,\tau}$ and $\hat{\beta}_{i,\tau}$ are found using gradient decent. Then the estimated marginal CDF functions can be written as $\hat{F}_{i,\tau}(z_{i,\tau}) = I_{z_{i,\tau}}(\alpha_{i,\tau}\beta_{i,\tau})$ where $I_{z_{i,\tau}}(\alpha_{i,\tau}\beta_{i,\tau})$ is the regularized incomplete beta function.

APPENDIX B
BASELINE MODELS

B.1 Autoregressive Integrated Moving Average (ARIMA)

The ARIMA model is a popular and widely used statistical method for time series forecasting. It combines the concepts of autoregressive (AR), differencing (I), and moving average (MA) models to forecast future values based on past values. The AR component considers the linear relationship between an observation and a certain number of lagged observations. The I component involves differencing the series to make it stationary, removing trends or seasonality that can affect the model. However, the non-stationarity can be sometimes caused by a non-structured dynamic background [115]; The method proposed therein can only handle time series retrospectively, and fails to forecast the dynamic background. The unstructured non-stationarity is out of the scope of the current work and is left for future discussion. The MA component accounts for the linear dependency between an observation and a residual error from a moving average model applied to lagged observations.

The ARIMA model is a univariate model. The multivarite extention of ARIMA model is the Vector Autoregressive Integrated Moving Average (VARIMA) model. VARIMA models capture the dependencies between the lagged values of all the variables in the system. However, in the high-dimensional time series problem. The VARIMA model is unsuitable in practice because of the number of parameters in the linear model. Therefore, each time series is modeled independently using ARIMA model using the existing Auto-ARIMA package [116].

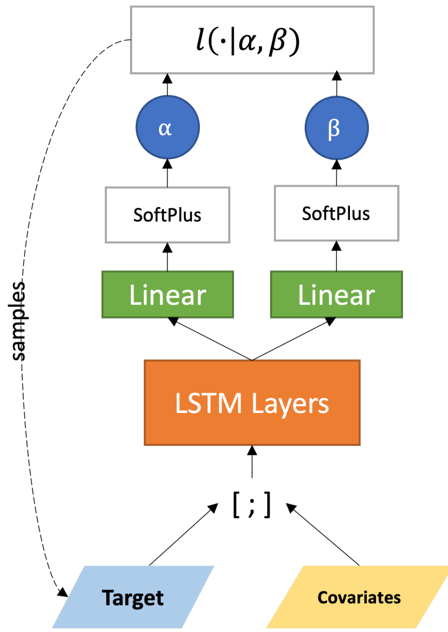


Figure B.1: DeepAR

B.2 DeepAR

DeepAR model is a deep learning-based approach that uses RNNs to capture temporal dependencies and can handle multiple related time series together. Similar to ARIMA model, DeepAR deals with multi-step ahead forecast by applying the model recursively, utilizing its own predictions as inputs for future predictions. As shown in Figure Figure B.1, the model takes observations prior to time t as input, and its predictions for the prediction horizon starting at time t are used as inputs for the model at the next time step. The model's outputs are the parameters of a beta distribution, α and β, rather than a Gaussian distribution as used in the original paper. This is because the DeepAR model outputs capacity factors of RES that are bounded between 0 and 1, making the Gaussian distribution unsuitable.

To estimate the model likelihood, the joint conditional distribution of $Z_{\cdot,\mathcal{T}}$ can be decomposed into a product of marginal conditional probabilistic distributions via Bayes rule as follows:

$$\prod_{t=W}^{N-H} \prod_{i\in\mathcal{D},\tau\in\mathcal{H}} F_{i,\tau}^{(t)}\left(z_{i,\tau}^{(t)} \mid \mathbf{z}_{\cdot,\mathcal{W}}^{(t)}, \mathbf{x}_{\cdot,\mathcal{W}}^{(t)}, \mathbf{x}_{\cdot,\mathcal{H}}^{(t)}\right). \tag{B.1}$$

The marginal conditional distribution takes the following form

$$F_{i,\tau}^{(t)}\left(z_{i,\tau}^{(t)} \mid \mathbf{z}_{\cdot,\mathcal{W}}^{(t)}, \mathbf{z}_{\cdot,\mathcal{W}}^{(t)}, \mathbf{x}_{\cdot,\mathcal{H}}^{(t)}\right) = \ell\left(z_{i,\tau}^{(t)} \mid \theta(\mathbf{h}_{i,t}^{(t)};\Theta)\right),$$

where the likelihood $\ell(\cdot)$ is parameterized by θ, where θ depends on trainable parameter Θ and the output $\mathbf{h}_{i,t}^{(t)}$ of an autoregressive RNN:

$$\mathbf{h}_{i,t}^{(s)} = h\left(\mathbf{h}_{i,t-1}^{(t)}, \mathbf{z}_{\cdot,\mathcal{W}}^{(t)}, \mathbf{z}_{\cdot,\mathcal{W}}^{(t)}, \mathbf{x}_{\cdot,\mathcal{H}}^{(t)};\Theta\right), \tag{B.2}$$

and $h(\cdot)$ is a multi-layer RNN with LSTM cells. Here, Θ are typically associated with the neural network model architecture. This model is autoregressive in the sense that it exploits the observation at the last step $z_{i,t-1}$ and the previous output of the network $\mathbf{h}_{i,t-1}^{(t)}$ as an input at the next time step.

In our work, because the amount of electricity generated by the wind farm is capped by the wind power capacity per wind farm, by normalizing with respect to the wind power capacity, the historical wind power time series can be represented within the range of $[0,1]$. Thus, the beta distribution for the target parametric distribution is more suitable. The likelihood of the beta distribution reads

$$\ell(z;\alpha,\beta) = \frac{z^{\alpha-1}(1-z)^{\beta-1}}{B(\alpha,\beta)},$$

where

$$B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}, \tag{B.3}$$

and $\Gamma(\cdot)$ is the Gamma function. In this beta distribution example, at time t, the model estimation can be done by maximizing the log-likelihood as follows:

$$\max_{\mathbf{\Theta}} \frac{1}{(N-W-H)HD} \sum_{t=W}^{N-H} \sum_{i\in\mathcal{D},\tau\in\mathcal{H}} \log \ell\left(z_{i,\tau}^{(t)};\alpha_{i,\tau}^{(t)},\beta_{i,\tau}^{(t)}\right),$$

with $\theta(\mathbf{h}_{i,t}^{(t)};\Theta) = \left(\alpha_{i,\tau}^{(t)}, \beta_{i,\tau}^{(t)}\right)$.

B.3 NLinear and DLinear

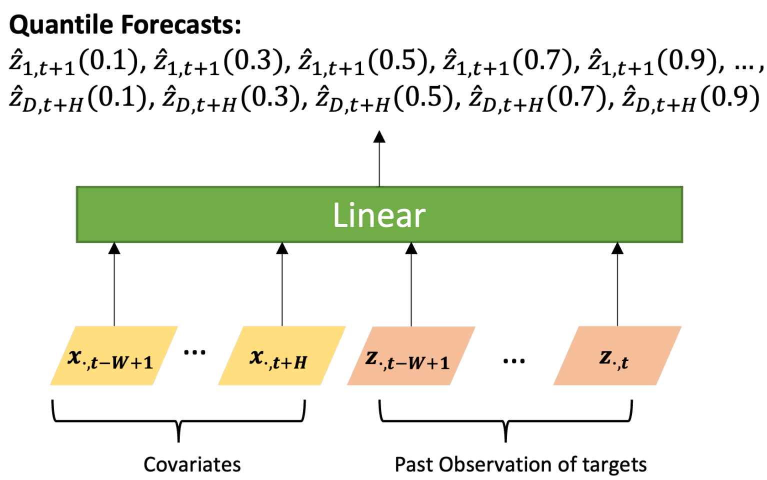


Figure B.2: Linear Model

The NLinear and Dlinear models are variants of the model described in [86]. The original paper does not consider the spatial correlations and covariates. And the weights are shared for every time series. To account for covariates (weather forecast), every future dimension in space and time is regressed on features of all past observations for all time series as well as covariates that extends to future.

Figure Figure B.2 provides a visual representation of the models inputs and outputs for a vanilla linear (one-layer) model. Dlinear and Nlinear correspond to two preprocessing methods build upon the vanilla linear model. Dlinear uses decomposition scheme, which decomposes the input into trend and seasonal component. NLinear normalizes each input by subtracting the last value in the sequence, and then adds it back after passing through the linear layer to address distribution shift. Initially, the models were only used for point predictions. However, in this paper, it is adapted to a probabilistic setting by using quantile regression as the objective function, as described in the TFT [5] model.

B.4 Temporal Fusion Transformer (TFT)

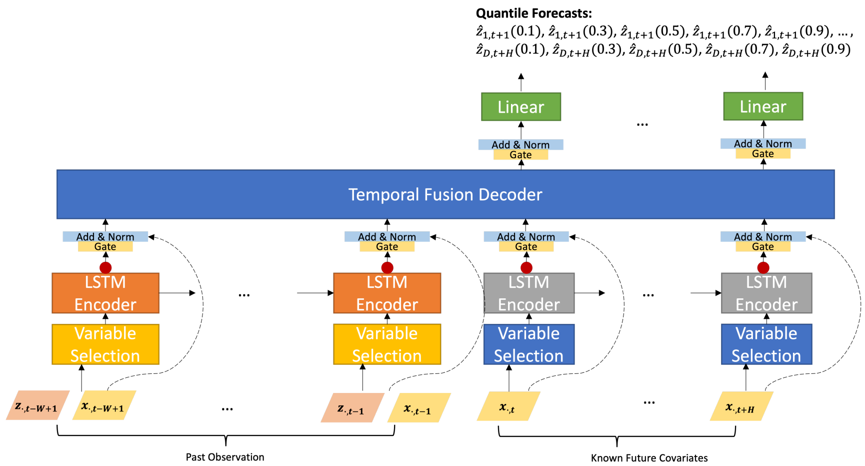


Figure B.3: Temporal Fusion Transformer (TFT)

TFT is a deep learning model designed for multi-horizon forecasting tasks with mixed-type inputs, including static covariates, known future inputs, and past exogenous time series. TFT uses a combination of recurrent layers and self-attention layers to learn temporal relationships at different scales and provide interpretable insights into temporal dynamics. The model utilizes specialized components to select relevant features and gating layers to suppress unnecessary components, resulting in high performance in a wide range of scenarios. Figure Figure B.3 presents an illustration of TFT [5] architecture.

TFT models output prediction intervals via a quantile regression approach. For a chosen set of quantiles $\mathcal{Q}$ with $|\mathcal{Q}| = Q$, the TFT model outputs forecast quantiles $\hat{z}_{i,\tau}^{(t)}(q),\ q \in \mathcal{Q}$. The objective is to minimize the quantile loss function with respect to all quantile outputs $\hat{z}_{i,\tau}^{(t)}(q)$'s, i.e.,

$$\min \frac{1}{(N-W-H)HD} \sum_{t=W}^{N-H} \sum_{i \in \mathcal{D}, \tau \in \mathcal{H}} \sum_{q \in \mathcal{Q}} \mathrm{QL}\left(z_{i,\tau}^{(t)}, \hat{z}_{i,\tau}^{(t)}(q)\right),$$

where

$$\begin{aligned}
&\text{QL}\left(z_{i,\tau}^{(s)}, \hat{z}_{i,\tau}^{(s)}(q)\right) \\
&\quad = q\left(z_{i,\tau}^{(s)} - \hat{z}_{i,\tau}^{(s)}(q)\right)^{+} + (1-q)\left(\hat{z}_{i,\tau}^{(s)}(q) - z_{i,\tau}^{(s)}\right)^{+},
\end{aligned}$$

and $(\cdot)^{+} = \max(0, \cdot)$. All experiments use values of $q \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. In the inference phase, scenarios are sampled by first generating uniformly-distributed noised in $[0, 1]$, then by applying the inverse CDF defined by the quantiles produced by the TFT model.

APPENDIX C
LIST OF PUBLICATIONS

This appendix presents publications completed during my Ph.D. studies at Georgia Institute of Technology from 2020 to 2025. While some of these works form the core chapters of this thesis, others represent collaborative research in related areas of power systems, forecasting, supply chains, and machine learning. The publications are organized by type: journal articles, workshop papers, and preprints.

C.1 Journal Publications

1. **Zhang, H.**, Tanneau, M., Huang, C., Joseph, V.R., Wang, S. and Van Hentenryck, P., 2024. Asset bundling for hierarchical forecasting of wind power generation. *Electric Power Systems Research*, 235, p.110771.

2. Santanam, T., Trasatti, A., Van Hentenryck, P., and **Zhang, H.**, 2024. Public Transit for Special Events: Ridership Prediction and Train Scheduling. *IEEE Transactions on Intelligent Transportation Systems*.

3. Auad, R., Dalmeijer, K., Riley, C., Santanam, T., Trasatti, A., Van Hentenryck, P., and **Zhang, H.**, 2021. Resiliency of on-demand multimodal transit systems during a pandemic. *Transportation Research Part C: Emerging Technologies*, 133, p.103418.

C.2 Workshop Papers

1. **Zhang, H.**, Arvin, C., Efimov, D., Mahoney, M.W., Perrault-Joncas, D., Ramasubramanian, S., Wilson, A.G., and Wolff, M., 2023. LLMForecaster: Improving Seasonal Event Forecasts with Unstructured Textual Data. In *NeurIPS Workshop on Time Series in the Age of Large Models*.

C.3 Preprints

1. **Zhang, H.**, Zandehshahvar, R., Tanneau, M., and Van Hentenryck, P., 2024. Weather-Informed Probabilistic Forecasting and Scenario Generation in Power Systems. In preparation for *Applied Energy*.

2. Ojha, R., Chen, W., **Zhang, H.**, Khir, R., Erera, A., and Van Hentenryck, P., 2023. Optimization-based Learning for Dynamic Load Planning in Trucking Service Networks. In preparation for *Transportation Science*.

3. Dong, Z., **Zhang, H.**, Zhu, S., Xie, Y., and Van Hentenryck, P., 2021. Multi-resolution spatio-temporal prediction with application to wind power generation. *arXiv preprint arXiv:2108.13285*.

REFERENCES

[1] S. L. Ho and M. Xie, "The use of arima models for reliability forecasting and analysis," *Computers & industrial engineering*, vol. 35, no. 1-2, pp. 213–216, 1998.

[2] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[3] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *International Journal of Forecasting*, vol. 36, no. 1, pp. 75–85, 2020.

[4] R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka, "A multi-horizon quantile recurrent forecaster," *arXiv preprint arXiv:1711.11053*, 2017.

[5] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[6] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[7] D. Dong, Z. Sheng, and T. Yang, "Wind power prediction based on recurrent neural network with long short-term memory units," in *2018 International Conference on Renewable Energy and Power Engineering (REPE)*, 2018, pp. 34–38.

[8] H. Wilms, M. Cupelli, A. Monti, and T. Gross, "Exploiting spatio-temporal dependencies for rnn-based wind power forecasts," in *2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia)*, 2019, pp. 921–926.

[9] T. Srivastava, Vedanshu, and M. Tripathi, "Predictive analysis of rnn, gbm and lstm network for short-term wind power forecasting," *Journal of Statistics and Management Systems*, vol. 23, no. 1, pp. 33–47, 2020.

[10] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021.

[11] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20 200 209, 2021.

[12] Z. Shi, H. Liang, and V. Dinavahi, "Direct interval forecast of uncertain wind power based on recurrent neural networks," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 3, pp. 1177–1187, 2018.

[13] P. Pinson, H. A. Nielsen, J. K. Møller, H. Madsen, and G. N. Kariniotakis, "Non-parametric probabilistic forecasts of wind power: Required properties and evaluation," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 10, no. 6, pp. 497–516, 2007.

[14] J. B. Bremnes, "Probabilistic wind power forecasts using local quantile regression," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 7, no. 1, pp. 47–54, 2004.

[15] H. Bludszuweit, J. A. Dominguez-Navarro, and A. Llombart, "Statistical analysis of wind power forecast error," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 983–991, 2008.

[16] P. Pinson, "Very-short-term probabilistic forecasting of wind power with generalized logit–normal distributions," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 61, no. 4, pp. 555–576, 2012.

[17] C. Wan, J. Lin, J. Wang, Y. Song, and Z. Y. Dong, "Direct quantile regression for nonparametric probabilistic forecasting of wind power generation," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2767–2778, 2017.

[18] H. Holttinen *et al.*, "Methodologies to determine operating reserves due to increased wind power," *IEEE transactions on sustainable energy*, vol. 3, no. 4, pp. 713–723, 2012.

[19] R. Sioshansi and D. Hurlbut, "Market protocols in ercot and their effect on wind generation," *Energy Policy*, vol. 38, no. 7, pp. 3192–3197, 2010.

[20] R. J. Hyndman, R. A. Ahmed, G. Athanasopoulos, and H. L. Shang, "Optimal combination forecasts for hierarchical time series," *Computational Statistics & Data Analysis*, vol. 55, no. 9, pp. 2579–2589, 2011.

[21] S. L. Wickramasuriya, G. Athanasopoulos, and R. J. Hyndman, "Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization," *Journal of the American Statistical Association*, vol. 114, no. 526, pp. 804–819, 2019.

[22] Y. Zhang and J. Dong, "Least squares-based optimal reconciliation method for hierarchical forecasts of wind power generation," *IEEE Transactions on Power Systems*, pp. 1–1, 2018.

[23] L. Bai and P. Pinson, "Distributed reconciliation in day-ahead wind power forecasting," *Energies*, vol. 12, no. 6, 2019.

[24] C. Di Modica, P. Pinson, and S. Ben Taieb, "Online forecast reconciliation in wind power prediction," *Electric Power Systems Research*, vol. 190, p. 106 637, 2021.

[25] A. Stratigakos, D. van der Meer, S. Camal, and G. Kariniotakis, "End-to-end Learning for Hierarchical Forecasting of Renewable Energy Production with Missing Values," in *2022 17th International Conference on Probabilistic Methods Applied to Power Systems*, 2022, pp. 1–6.

[26] M. E. Hansen, N. Peter, J. K. Møller, and M. Henrik, "Reconciliation of wind power forecasts in spatial hierarchies," *Wind Energy*, vol. 26, no. 6, pp. 615–632, 2023.

[27] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[28] W. Katzenstein, E. Fertig, and J. Apt, "The variability of interconnected wind plants," *Energy policy*, vol. 38, no. 8, pp. 4400–4410, 2010.

[29] M. M. Bandi, "Spectrum of wind power fluctuations," *Physical review letters*, vol. 118, no. 2, p. 028 301, 2017.

[30] M. Shahriari and S. Blumsack, "Scaling of wind energy variability over space and time," *Applied Energy*, vol. 195, pp. 572–585, 2017.

[31] C. Wu, X.-P. Zhang, and M. Sterling, "Wind power generation variations and aggregations," *CSEE Journal of Power and Energy Systems*, vol. 8, no. 1, pp. 17–38, Jan. 2022.

[32] L. M. Hansen, "Can wind be a firm resource-a north carolina case study," *Duke Envtl. L. & Pol'y F.*, vol. 15, p. 341, 2004.

[33] B. Drake and K. Hubacek, "What to expect from a greater geographic dispersion of wind farms?—a risk portfolio approach," *Energy Policy*, vol. 35, no. 8, pp. 3999–4008, 2007.

[34] F. Roques, C. Hiroux, and M. Saguan, "Optimal wind power deployment in europe—a portfolio approach," *Energy policy*, vol. 38, no. 7, pp. 3245–3256, 2010.

[35] Y. Degeilh and C. Singh, "A quantitative approach to wind farm diversification and reliability," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 2, pp. 303–314, 2011.

[36] N. S. Thomaidis, F. J. Santos-Alamillos, D. Pozo-Vázquez, and J. Usaola-García, "Optimal management of wind and solar energy resources," *Computers & Operations Research*, vol. 66, pp. 284–291, 2016.

[37] F. Santos-Alamillos, N. Thomaidis, J. Usaola-García, J. Ruiz-Arias, and D. Pozo-Vázquez, "Exploring the mean-variance portfolio optimization approach for planning wind repowering actions in spain," *Renewable Energy*, vol. 106, pp. 335–342, 2017.

[38] O. Grothe and J. Schnieders, "Spatial dependence in wind and optimal wind power allocation: A copula-based analysis," *Energy policy*, vol. 39, no. 9, pp. 4742–4754, 2011.

[39] T. Ma and C. Li, "The electricity portfolio decision-making model based on the cvar under risk conditions," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 7, no. 3, pp. 570–575, 2014.

[40] G. Gersema and D. Wozabal, "Risk-optimized pooling of intermittent renewable energy sources," *Journal of banking & finance*, vol. 95, pp. 217–230, 2018.

[41] A. Vinel and E. Mortaz, "Optimal pooling of renewable energy sources with a risk-averse approach: Implications for us energy portfolio," *Energy Policy*, vol. 132, pp. 928–939, 2019.

[42] C. Han and A. Vinel, "Reducing forecasting error by optimally pooling wind energy generation sources through portfolio optimization," *Energy*, vol. 239, p. 122 099, 2022.

[43] J. Li, J. Zhou, and B. Chen, "Review of wind power scenario generation methods for optimal operation of renewable energy systems," *Applied Energy*, vol. 280, p. 115 992, 2020.

[44] Y. Zhang, Q. Ai, F. Xiao, R. Hao, and T. Lu, "Typical wind power scenario generation for multiple wind farms using conditional improved wasserstein generative adversarial network," *International Journal of Electrical Power & Energy Systems*, vol. 114, p. 105 388, 2020.

[45] W. Dong, X. Chen, and Q. Yang, "Data-driven scenario generation of renewable energy production based on controllable generative adversarial networks with interpretability," *Applied Energy*, vol. 308, p. 118 387, 2022.

[46] J. Tan, G. Qi, D. Sun, W. Li, H. Yang, and Z. Li, "Forecasting renewable energy generation scenarios based on multi-agent diverse gans," in *2020 IEEE Sustainable Power and Energy Conference (iSPEC)*, 2020, pp. 180–186.

[47] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3265–3275, 2018.

[48] D. Salinas, M. Bohlke-Schneider, L. Callot, R. Medico, and J. Gasthaus, "High-dimensional multivariate forecasting with low-rank gaussian copula processes," *Advances in neural information processing systems*, vol. 32, 2019.

[49] F. Golestaneh, H. B. Gooi, and P. Pinson, "Generation and evaluation of space–time trajectories of photovoltaic power," *Applied Energy*, vol. 176, pp. 80–91, 2016.

[50] X.-Y. Ma, Y.-Z. Sun, and H.-L. Fang, "Scenario generation of wind power based on statistical uncertainty and variability," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 4, pp. 894–904, 2013.

[51] R. J. Bessa, "On the quality of the gaussian copula for multi-temporal decision-making problems," in *2016 Power Systems Computation Conference (PSCC)*, IEEE, 2016, pp. 1–7.

[52] P. Pinson, H. Madsen, H. A. Nielsen, G. Papaefthymiou, and B. Klöckl, "From probabilistic forecasts to statistical scenarios of short-term wind power production," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 12, no. 1, pp. 51–62, 2009.

[53] A. Mashlakov, T. Kuronen, L. Lensu, A. Kaarna, and S. Honkapuro, "Assessing the performance of deep learning models for multivariate probabilistic energy forecasting," *Applied Energy*, vol. 285, p. 116 405, 2021.

[54] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large language models are zero-shot time series forecasters," in *Advances in Neural Information Processing Systems*, 2023.

[55] H. Xue and F. D. Salim, "Promptcast: A new prompt-based learning paradigm for time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[56] C. Chang, W.-C. Peng, and T.-F. Chen, "Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms," *arXiv preprint arXiv:2308.08469*, 2023.

[57] D. Cao *et al.*, "Tempo: Prompt-based generative pre-trained transformer for time series forecasting," *arXiv preprint arXiv:2310.04948*, 2023.

[58] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.

[59] A. F. Ansari *et al.*, “Chronos: Learning the language of time series,” *arXiv preprint arXiv:2403.07815*, 2024.

[60] E. J. Hu *et al.*, “Lora: Low-rank adaptation of large language models,” *arXiv preprint arXiv:2106.09685*, 2021.

[61] B. Sergi *et al.*, “ARPA-E PERFORM datasets,” Aug. 2022.

[62] A. Ahmadi, M. Talaei, M. Sadipour, A. M. Amani, and M. Jalili, “Deep federated learning-based privacy-preserving wind power forecasting,” *IEEE Access*, vol. 11, pp. 39 521–39 530, 2023.

[63] Y. Li, R. Wang, Y. Li, M. Zhang, and C. Long, “Wind power forecasting considering data privacy protection: A federated deep reinforcement learning approach,” *Applied Energy*, vol. 329, p. 120 291, 2023.

[64] G. Athanasopoulos, R. J. Hyndman, N. Kourentzes, and F. Petropoulos, “Forecasting with temporal hierarchies,” *European Journal of Operational Research*, vol. 262, no. 1, pp. 60–74, 2017.

[65] J. Herzen *et al.*, “Darts: User-friendly modern machine learning for time series,” *Journal of Machine Learning Research*, vol. 23, no. 124, pp. 1–6, 2022.

[66] O. F. Eikeland, F. D. Hovem, T. E. Olsen, M. Chiesa, and F. M. Bianchi, “Probabilistic forecasts of wind power generation in regions with complex topography using deep learning methods: An arctic case,” *Energy Conversion and Management: X*, vol. 15, p. 100 239, 2022.

[67] Q. Cao, B. T. Ewing, and M. A. Thompson, “Forecasting wind speed with recurrent neural networks,” *European Journal of Operational Research*, vol. 221, no. 1, pp. 148–154, 2012.

[68] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.

[69] S. Bai, J. Z. Kolter, and V. Koltun, *An empirical evaluation of generic convolutional and recurrent networks for sequence modeling*, 2018.

[70] B. Lim, S. Arik, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting. arxiv,” *arXiv preprint arXiv:1912.09363*, 2019.

[71] J. W. Messner, P. Pinson, J. Browell, M. B. Bjerregård, and I. Schicker, “Evaluation of wind power forecasts—an up-to-date view,” *Wind Energy*, vol. 23, no. 6, pp. 1461–1481, 2020.

[72] M. Scheuerer and T. M. Hamill, “Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities*,” *Monthly Weather Review*, vol. 143, no. 4, pp. 1321–1334, 2015.

[73] European Centre for Medium-Range Weather Forecasts, *TIGGE Data Retrieval*, https://apps.ecmwf.int/datasets/data/tigge/levtype=sfc/type=cf/, [Online; accessed March 2023], 2023.

[74] G. A. Shanika L. Wickramasuriya and R. J. Hyndman, “Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization,” *Journal of the American Statistical Association*, vol. 114, no. 526, pp. 804–819, 2019.

[75] J. M. Freeman *et al.*, “System advisor model (sam) general description (version 2017.9. 5),” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2018.

[76] U.S. Department of Energy, *Renewable systems integration*, https://www.energy.gov/eere/wind/renewable-systems-integration, Accessed: 2024-06-10, n.d.

[77] U.S. Department of Energy, *Windexchange: Distributed wind energy*, https://windexchange.energy.gov/markets/distributed, Accessed: 2024-06-10, n.d.

[78] O. Stover, P. Karve, and S. Mahadevan, “Reliability and risk metrics to assess operational adequacy and flexibility of power grids,” *Reliability Engineering & System Safety*, vol. 231, p. 109 018, 2023.

[79] B. Knueven *et al.*, “Stochastic look-ahead commitment: A case study in miso,” in *2023 IEEE Power & Energy Society General Meeting (PESGM)*, 2023, pp. 1–5.

[80] M. Sklar, “Fonctions de repartition an dimensions et leurs marges,” *Publ. inst. statist. univ. Paris*, vol. 8, pp. 229–231, 1959.

[81] A. J. Patton, “Modelling asymmetric exchange rate dependence,” *International economic review*, vol. 47, no. 2, pp. 527–556, 2006.

[82] A. J. Patton, “Copula–based models for financial time series,” in *Handbook of financial time series*, Springer, 2009, pp. 767–785.

[83] J.-D. Fermanian and M. H. Wegkamp, “Time-dependent copulas,” *Journal of Multivariate Analysis*, vol. 110, pp. 19–29, 2012.

[84] T. Werho, J. Zhang, V. Vittal, Y. Chen, A. Thatte, and L. Zhao, “Scenario generation of wind farm power for real-time system operation,” *arXiv preprint arXiv:2106.09105*, 2021.

[85] D. Varela Santolalla and R. Mladek, *Tigge archive*, https://confluence.ecmwf.int/display/TIGGE, Accessed: 2024-04-03, Feb. 2024.

[86] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" *arXiv preprint arXiv:2205.13504*, 2022.

[87] NIXTLA, *Autoarima model*, Nixtlaverse StatsForecast Documentation, Accessed: 2024-04-03, 2024.

[88] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[89] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 2623–2631, ISBN: 9781450362016.

[90] T. Gneiting and A. E. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American statistical Association*, vol. 102, no. 477, pp. 359–378, 2007.

[91] M. B. Bjerregård, J. K. Møller, and H. Madsen, "An introduction to multivariate probabilistic forecast evaluation," *Energy and AI*, vol. 4, p. 100 058, 2021.

[92] PACE, *Partnership for an Advanced Computing Environment (PACE)*, 2017.

[93] R. Koenker and K. F. Hallock, "Quantile regression," *Journal of economic perspectives*, vol. 15, no. 4, pp. 143–156, 2001.

[94] S. Chakraborty and X. Zhang, "A new framework for distance and kernel-based metrics in high dimensions," *Electronic Journal of Statistics*, vol. 15, no. 2, pp. 5455–5522, 2021.

[95] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[96] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European journal of operational research*, vol. 270, no. 2, pp. 654–669, 2018.

[97] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[98] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[99] H. Zhou *et al.*, “Informer: Beyond efficient transformer for long sequence time-series forecasting,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 11 106–11 115.

[100] H. Wu, J. Xu, J. Wang, and M. Long, “Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting,” *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.

[101] H. Zhang, Y. Zou, X. Yang, and H. Yang, “A temporal fusion transformer for short-term freeway traffic speed multistep prediction,” *Neurocomputing*, vol. 500, pp. 329–340, 2022.

[102] Y. Zhou, “Temporal fusion transformers model for traffic flow prediction,” in *Proceedings of the 2nd International Conference on Big Data Economy and Digital Management, BDEDM 2023, January 6-8, 2023, Changsha, China*, 2023.

[103] C. Eisenach, Y. Patel, and D. Madeka, “Mqtransformer: Multi-horizon forecasts with context dependent and feedback-aware attention,” *arXiv preprint arXiv:2009.14799*, 2020.

[104] H. Zhang, M. Tanneau, C. Huang, V. R. Joseph, S. Wang, and P. Van Hentenryck, “Asset bundling for hierarchical forecasting of wind power generation,” *Electric Power Systems Research*, vol. 235, p. 110 771, 2024.

[105] P. Zheng, H. Zhou, J. Liu, and Y. Nakanishi, “Interpretable building energy consumption forecasting using spectral clustering algorithm and temporal fusion transformers architecture,” *Applied Energy*, vol. 349, p. 121 607, 2023.

[106] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, “Timesnet: Temporal 2d-variation modeling for general time series analysis,” 2023.

[107] Y. Jiang *et al.*, “Empowering time series analysis with large language models: A survey,” *arXiv preprint arXiv:2402.03182*, 2024.

[108] X. Yu, Z. Chen, Y. Ling, S. Dong, Z. Liu, and Y. Lu, “Temporal data meets llm–explainable financial time series forecasting,” *arXiv preprint arXiv:2306.11025*, 2023.

[109] X. Wang, M. Fang, Z. Zeng, and T. Cheng, “Where would i go next? large language models as human mobility predictors,” *arXiv preprint arXiv:2308.15197*, 2023.

[110] X. Liu *et al.*, "Large language models are few-shot health learners," *arXiv preprint arXiv:2305.15525*, 2023.

[111] N. Tripuraneni, D. Madeka, D. Foster, D. Perrault-Joncas, and M. I. Jordan, "Meta-analysis of randomized experiments with applications to heavy-tailed response data," 2023.

[112] MosaicML NLP Team, *Introducing mpt-7b: A new standard for open-source, commercially usable llms*, Accessed: 2023-05-05, 2023.

[113] J. W. Messner, P. Pinson, J. Browell, M. B. Bjerregård, and I. Schicker, "Evaluation of wind power forecasts—an up-to-date view," *Wind Energy*, vol. 23, no. 6, pp. 1461–1481, 2020.

[114] M. Scheuerer and T. M. Hamill, "Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities," *Monthly Weather Review*, vol. 143, no. 4, pp. 1321–1334, 2015.

[115] S. Wei, Y. Xie, and D. Rahnev, "Inferring serial correlation with dynamic backgrounds," in *International Conference on Machine Learning*, PMLR, 2021, pp. 11 047–11 057.

[116] F. Garza, M. Mergenthaler Canseco, C. Challú, and K. G. Olivares, *StatsForecast: Lightning fast forecasting with statistical and econometric models*, PyCon Salt Lake City, Utah, US 2022, 2022.