2023

# Probabilistic Multivariate Time Series Forecasting and Robust Uncertainty Quantification with Applications in Electricity Price Prediction

Jie Han

# Probabilistic Multivariate Time Series Forecasting and Robust Uncertainty Quantification with Applications in Electricity Price Prediction

by

JIE HAN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
December 2023

Dedicated to the my friends who provided unwavering support during the solitary days of the COVID era and the challenging research junctures.

# ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to everyone who contributed to the completion of this thesis.

First and foremost, I am deeply thankful to my advisor, Shouyi Wang, for his unwavering guidance, invaluable insights, and continuous support throughout the entire research process. His mentorship has been instrumental in shaping the direction of this thesis.

I extend my appreciation to the faculty members of the Department of Industrial, Manufacturing, and Systems Engineering for their constructive feedback and scholarly input. Their expertise has enriched the depth and quality of this work. I would like to acknowledge Dr. Lee Wei-jen from the Department of Electrical Engineering for his expertise in electricity markets which made this research possible.

I would like to acknowledge the National Science Foundation award ECCS-1938895 for their financial support. Their investment in my academic pursuits has been a significant catalyst for the successful completion of this thesis.

I am grateful to my family for their enduring encouragement and understanding during the highs and lows of this academic journey. Their love and support have been my anchor.

Special thanks to my friends and colleagues who provided encouragement, shared their experiences and offered a helping hand when needed. Your camaraderie made the challenges more manageable and the victories more joyful.

Lastly, I express my gratitude to all the participants and contributors who willingly shared their time and insights for this research. Your contributions have added depth and real-world relevance to the findings.

This thesis represents the culmination of a collective effort, and I am sincerely thankful to each and every person who played a role, big or small, in its realization.

Thank you.

Jie Han

November 28, 2023

# ABSTRACT

Probabilistic Multivariate Time Series Forecasting and Robust Uncertainty Quantification with Applications in Electricity Price Prediction

Jie Han, Ph.D.

The University of Texas at Arlington, 2023

Supervising Professor: Shouyi Wang

Electricity price forecasting (EPF) is a crucial task for market participants seeking informed decisions in day-ahead electricity markets. The increasing penetration of stochastic renewable energy and the deregulation of electricity markets pose challenges to electricity price forecasting. Given the dependence of electricity prices on stochastic factors such as weather conditions, market dynamics, and customer behaviors, deterministic forecasting methods offer limited insight into the potential future states of energy prices in highly stochastic markets.

In this study, a transformer-based electricity price forecasting (TDEPF) model was developed, utilizing a two-step training process and demonstrating superior performance compared to typical RNN models. Subsequently, three probabilistic forecasting models were introduced: quantile regression (QR-Transformer), transformer-based composite quantile regression (TCQR), and Gaussian Process combined with transformer (GP-Transformer). These models can produce multi-step ahead forecasts using multivariate time series as inputs. The application of these models was demonstrated in the context of day-ahead electricity price forecasting, utilizing five years of data from the Electric Reliability Council of Texas (ERCOT).

Among the probabilistic models, TCQR emerged as the most effective, as evidenced by metrics such as CRPS, pinball loss, and Winkler score. To evaluate the accuracy of peak time prediction, a novel metric called Winkler-peak score was introduced. Additionally, to regulate the convergence of quantiles near the 0% or 100% level, MSE-regularized pinball loss was proposed. Simultaneously, to prioritize peak time prediction, Winkler-peak score regularized pinball loss was proposed. Both regularization methods on pinball loss were demonstrated to effectively achieve their respective goals when applied in TCQR.

Given that deep learning methods operate as black boxes and lack a guarantee of coverage rate, a deeper investigation into uncertainty quantification was undertaken on probabilistic forecasting results. Conformal prediction has been established as capable of constructing prediction intervals with statistically guaranteed coverage rates. However, existing research has not explored applications in either multivariate time series forecasting or multi-step ahead forecasting. Hence, in this study, adaptive quantile random forest (AQRF) and adaptive conformal residual fitting (ACRF) were introduced. Both methods can attain the target coverage rate, and AQRF, in particular, can achieve a narrower bandwidth compared to benchmark models such as conformal prediction and quantile random forest methods. In conclusion, this research can furnish reliable day-ahead electricity price forecasts, aiding decision-makers in making informed decisions within the electricity market.

# TABLE OF CONTENTS

# LIST OF ILLUSTRATIONS

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

As a state with the highest energy production in the U.S.[2] and abundant in renewable energy resources[3], Texas adopts a deregulated market structure, which "minimizes subsidy requirements for clean energy technologies with lower penetration rates"[4]. Thus, electricity prices of the Electricity Reliability Council of Texas (ERCOT) in Texas are more sensitive to climate change and affected by the bidding policy adopted by brokers. To be more competitive in a deregulated market, such as ERCOT, accurate electricity price forecasting (EPF) models are needed in decision-making. In this study, a multivariate time series model for EPF is built and tested in the ERCOT market.

## 1.1 Multivariate Time Series (MTS)

A time series is sequential data observed over a period and can be continuous or discrete as well as regularly or irregularly spaced [5]. When more than one related time series is observed at the same time over a period, these time series are known as multivariate time series (MTS) while only one time series observed over time is univariate time series[6]. MTS is moreover extremely valuable while also being the most common time series data since changes of the target variable are frequently not only correlated with the variable's historical values but also with other variables. For instance, exploring simultaneous behaviors of energy demand, temperature, and humidity over time will enable us to gain more insight into change patterns of energy demand compared to only observing the time series of energy demand itself.

Electricity prices are also correlated with multi variables, such as temperatures, wind speeds, and load, so this study targeted models for multivariate time series. To better describe the properties and analyze patterns of MTS, some basic concepts of MTS are covered in Chapter 2.

## 1.2 Multistep Ahead Forecasting

Multi-step ahead forecasting, which predicts more than one value in the future, is increasingly important and extensively used in energy forecasting. For instance, as wind power has become the second-largest source of U.S. electricity generation on March 29, 2022[7], accurate day ahead wind output forecasting helps operators prepare for changes of wind power in next day and dispatch available electricity generation from different sources[8]. Similarly, well-performed day-ahead or hours-ahead electricity price forecasting can assist brokers with more beneficial bid in a deregulated day-ahead electricity market[9], such as ERCOT.

Compared to one-step-ahead forecasting, multi-step-ahead forecasting has been an arduous assignment. First, accumulated errors, increased uncertainties, and the lack of information impair the accuracy of multi-step ahead forecasting[10]. Second, different strategies are needed to produce multiple outputs, such as recursively generating one-step ahead prediction, directly generating each step ahead prediction with different models and so on[11], which adds complexity to the modeling process. Our work focused on generating multiple outputs with only one model, called multiple input multiple outputs (MIMO) or sequence to sequence model, which has a relatively concise modeling process and maintains the same level of or even higher accuracy.

### 1.3 Probabilistic Time Series Forecasting

As mentioned in [12], probabilistic forecasting, which appeared as applications of prediction intervals or prediction densities, became increasingly popular in 1980s because the information provided by point predictions was confined to one exact value. Nevertheless, probabilistic forecasting was employed in EPF rather late, until 2014, according to the number of related publications[13]. [14] did a comprehensive review of the advances of probabilistic EPF and offered a detailed tutorial to encourage more applications of efficient and statistically sound probabilistic models in EPF since EPF had become more crucial in the decision-making process in deregulated energy markets. Our work is dedicated to developing a statistically sound and state-of-the-art probabilistic model for EPF.

Electricity price forecasting (EPF) can be described as predicting the electricity price of prediction timesteps with data collected during history timesteps, and there is usually a lead time for conducting prediction and bidding. Prediction time length is 24 hours in day-ahead forecasting and 15 minutes or 1 hour in real-time forecasting. EPF has been a challenging task due to frequent occurrences of abrupt changes in electricity prices which can be caused by extreme weather conditions, fuel prices, sustainable energy outputs, maintenance plans, breakdown of generators, etc. Those unexpectedly high values in a time series are called spikes. Compared to other electricity markets, such as Pennsylvania-New Jersey-Maryland Interconnection (PJM), California Independent System Operator (CAISO), and Midcontinent Independent System Operator (MISO), ERCOT market is more deregulated and competitive which means higher variations in electricity prices.

Influences of spikes were not considered in many EPF research, and forecasting models were built directly on original data[15, 16, 17, 18]. Nevertheless, the influences cannot be ignored for EPF in the ERCOT market where the spikes of electricity prices can be as high as \$9000, while 90% of electricity prices are below \$40.5 according to the hourly real-time settlement point prices (SPP) from 2017 to 2021. These minority values exerted an enormous adverse influence on the models' training process and destroyed the models' abilities to predict. [19] compared forecasting results of omitting different levels of spikes of ERCOT SPP data, and the MAE was 4.528 when prices over 100 dollars were omitted, however, the MAE increased to 13.750 when prices over 500 dollars were omitted.

Some research handled spikes via classification or data-smoothing technologies before building models. Both [20] and [21] decomposed the electricity price series to sub-series with simpler patterns which were easier for models to catch using wavelet. [21] further classified predicted electricity prices as spikes and non-spikes with a compound classifier including relevance vector machine, decision tree, and probabilistic neural network according to the preliminary forecasting results based on sub-series from wavelet. Two different forecasting models were built for spike and non-spike data individually.

Our methodology predicted the probabilities of spikes with a probabilistic neural network and built another model for normal price based on the SPP transformed by linear saturation function. The probability of spikes' occurrence is more important than accurate values of spikes, so once the price is predicted as a spike, there is no need to predict the accurate value which is difficult to predict as well.

### 1.4 Uncertainty Quantification of Prediction Models

Since machine learning and deep learning models play more and more important roles in decision-making [22], and these models are black-box and hard to interpret, it is important to measure the reliability and efficiency of those models. Uncertainty quantification provides a statistically trustworthy measurement of the reliability of machine learning and deep learning models.

Uncertainty quantification (UQ) encompasses the investigation of all sources of error and uncertainty, such as systematic and stochastic measurement error, ignorance, limitations of theoretical models, numerical representations, accuracy and reliability of computations, approximations, algorithms, and human error. In a more precise sense, UQ involves a comprehensive examination of the reliability of scientific inferences. In the realm of uncertainty quantification (UQ), it is imperative to note that UQ does not ascertain the correctness or truthfulness of a model. Rather, its role lies in affirming that, given the acceptance of a model's validity, one must consequently acknowledge the validity of specific conclusions, albeit to a quantifiable extent [23].

Conformal prediction serves as an accessible framework for establishing statistically robust uncertainty sets or intervals for the predictions generated by these models [24]. It is characterized as a distribution-free, non-parametric forecasting approach grounded in minimal assumptions[25]. This method can straightforwardly generate prediction sets that maintain statistical validity, even in finite sample cases. In this manuscript, a hybrid approach comprising a conformal quantile regression model [26] and an adaptively adjusted estimations online method [27] is introduced. Through this method, there is an effective enhancement in both the coverage rates of the targets and the bandwidth. Additionally, an investigation into the amalgamation of conformal residual fitting with an adaptively adjusted estimations online method is conducted, leading to increased coverage rates of prediction intervals.

## 1.5 Outline of Contributions

(1) A novel reversible spike transformation was developed, facilitating the learning of continuous patterns in electricity time series by the deterministic forecasting model. The original values can be derived through the reverse transformation.

(2) A two-step training procedure was employed to train a transformer –based model, named transformer-based day-ahead electricity price forecasting (TDEPF) model.

(3) The TDEPF model was compared to RNN, LSTM, and GRU and shows a better performance.

(4) A case study was done on the ERCOT market settlement point prices (SPP) day-ahead forecasting and an innovative spike transformation was applied, effectively enhancing the accuracy of prediction.

(5) The comparison between one-hot and sin-cos encoding methods was conducted, and it was determined that the latter is more efficient.

(6) A Winkler-peak score was developed based on the Winkler score to assess the accuracy of peak time predictions. The predictive accuracy increases as the Winkler-peak score decreases.

(7) The effectiveness of incorporating resource nodes for predicting SPP of load zone was evaluated.

(8) Integration of Gaussian Process and Transformer Model (GP-Transformer) to assess prediction uncertainties.

(9) Developed a transformer-based quantile regression model (QR-Transformer) to provide probabilistic forecasting.

(10) Propose the Winkler-peak score, a new performance metric for evaluating peak time predictions.

(11) Proposed Adaptive Quantile Random Forest (AQRF) calibration approach, aiming to enhance the coverage rates of prediction intervals while achieving a narrower bandwidth.

(12) Proposed Adaptive Conformal Residual Fitting (ACRF) approach, designed to elevate the coverage rates of prediction intervals to the specified target rates.

# CHAPTER 2

# BACKGROUND

## 2.1 Basic Concepts of MTS

Before building an accurate forecasting model for multivariate time series, relationships between different time series and underlying structure and patterns of single series should be studied as understanding features of time series is the foundation of forecasting [28]. Stationarity and autocorrelation relationships of MTS are commonly examined before building a forecasting model.

To describe the properties of MTS conveniently, representations of MTS are introduced. If an observation of a k-dimensional multivariate time series at time t is denoted by $Y_t$, where $Y_t = (Y_t^1, Y_t^2, ..., Y_t^k)$, observations from $t_1$ to $t_n$ can be represented by $Y_{t_1}, Y_{t_2}, ..., Y_{t_n}$ and lagged observations would be $Y_{t_1+l}, Y_{t_2+l}, ..., Y_{t_n+l}$, where n is a positive integer and l is an integer.

### 2.1.1 Stationarity

For any $t_1, t_2, ..., t_n$ and all $l = 0, 1, 2, ...$, if $Y_{t_1}, Y_{t_2}, ..., Y_{t_n}$ and $Y_{t_1+l}, Y_{t_2+l}, ..., Y_{t_n+l}$ follow the same probability distribution, $Y_t$ is strictly stationary[6]. Therefore, $E(Y_t) = \mu$ and $Var(Y_t) = E[(Y_t - \mu)(Y_t - \mu)']$ are constant for all $t$ provided that $Y_t$ is a stationary MTS. Here, the definition of stationarity is based on the assumption that $Y_t$ is finite and has at least two moments. In practical applications, strict stationarity is hard to prove. Therefore, [6, 29] define finite $Y_t$ with at least two moments as weakly stationary MTS, which is more common, on conditions that $E(Y_t) = \mu$ is constant and $E[(Y_t - \mu)(Y_{t+l} - \mu)']$ depends only on $l$.

### 2.1.2 Autocorrelation

Autocorrelation is a characteristic of time series where the current value correlates with the previous values. If a correlation only exists between the current value and the previous value, the correlation is called the first order autoregressive, AR(1), and if a correlation exists between the current and the two preceding values, the correlation is called the second-order autoregressive, AR(2), and so on [30]. Assuming that $Y_t$ represents the observation of the variable Y at time t, $Y_{t-1}$ should be considered in the regression model if a time series is the first-order autoregressive. Similarly, $Y_{t-1}$ and $Y_{t-2}$ should be used for predicting $Y_t$ if a time series is the second autoregressive. $Y_{t-1}$ is also called the first lagged value of $Y_t$.

As one of the critical features of time series, autocorrelation makes time series forecasting distinct from other regression tasks. A model's ability to determine the appropriate autoregressive order greatly affects the accuracy of forecasting. If a much higher order of autoregression is considered, accuracy of short-term forecasting may be impaired and reversely, long term patterns may be neglected. Accordingly, common long-term and short-term patterns of time series should be identified before building a model.

Features of univariate time series, such as seasonality, cyclic patterns, and trends are also useful in multivariate time series analysis. These descriptive features should be observed through graphs before building a forecasting model as well.

## 2.2 Statistical Models For MTS Forecasting

The vector autoregressive (VAR) model is one of the most widely used models[29, 31, 32] because it is easy to estimate whether by least-squares, maximum likelihood, or Bayesian method, and it has been thoroughly studied over a long time. A simple bivariate VAR (1) model can be written as $Y_t = \beta_0 + \beta_1 Y_{t-1} + \epsilon$ [29], where $Y_t$, $\beta_0$, and $Y_{t-1}$ are $k \times 1$ vectors, $\beta_1$ is a $k \times k$ coefficient matrix and $\epsilon$ is a residual vector. From the equation, we can see that VAR models only consider linear relationships among time steps, and it is not applicable to complex MTS. Similarly, vector moving average (VMA) models also only take linear relationships into account, but they assume each time series has a constant mean, and linear relationships exist in residuals. For instance, the equation of VMA (1), $Y_t = \mu + a_t - \beta_1 a_{t-1}$ [29], assumes a $k \times 1$ constant mean vector $\mu$ and linear combination of $k$ residuals at time $t - 1$ (residuals are represented by a). Obviously, VMA models explain variations in MTS with linear combinations of residuals from different time steps while VAR models explain linear relationships among time steps. It is natural to combine them into one equation, that is a vector autoregressive moving average (VARMA) model. VARMA (p, q) is the combination of VAR(p) and VMA(q) which indicates p order autoregressive and q time steps' moving window. No matter which model, VAR, VMA, or VARMA, they are models for stationary time series, but stationary time series are not so common in reality, especially in the energy field. Furthermore, selecting suitable parameters p, and q takes time.

There are many variants of VAR and VARMA dedicated to improving their performances and expanding their scope of application. Vector autoregressive integrated moving average (VARIMA) models expands the application context of VARMA by differencing time series to achieve stationarity[33, 34]. Nevertheless, the order of difference needs to be decided through trial and error, and it is not guaranteed that stationarity exists in time differences. Seasonal ARIMA(SARIMA) multiplies non-stationary seasonal patterns with stationary patterns to fit seasonally changed time series[35], but each quarter of a year needs a different model. To boost the computational efficiency of the parameter selection process in VAR, VAR-LASSO utilized the property of LASSO regression[36], selecting and estimating parameters at the same time[37, 38]. On the other hand, for VAR models with lots of zero values in the coefficient matrix, sparse VAR (sVAR) identifies non-zero coefficients by estimating partial spectral coherence at first and then refines results which accelerates computation as well[39]. In addition, the generalized autoregressive conditional heteroskedasticity (GARCH) model, a variant of autoregressive moving average (ARMA), fits time-dependent error variances, instead of time series itself[40]. Multivariate GARCH models can deal with nonstationary MTS with time-varying volatility and have been successfully applied to the financial domain as a useful decision tool, such as risk management and asset pricing[41, 42]. Besides, exogenous factors that may contribute to time series changes are introduced to ARIMA models to improve forecasting accuracy and these models are known as ARIMAX models[43].

Other than models based on autocorrelations and moving averages, support vector machine (SVM) is also a classic model for time series forecasting and achieved excellent prediction results for non-linear time series[44]. Furthermore, a multioutput SVM for multistep ahead forecasting (MM-SVM) was put forward to cover its instable spatiotemporal forecasting deficiency [43]. Nevertheless, SVM is essentially designed to solve a quadratic programming problem, so heavy computation becomes prominent when the dimensionality of data expands[45].

As an explosive increase in data size and complexity of MTS, the forecasting ability of most traditional MTS models is restricted due to correspondingly increased computational cost and common assumptions of certain stationarity. Comparatively, deep learning methods have gained popularity for their competence in modeling non-linear relationships in MTS.

## 2.3 Deep Learning Models For MTS Forecasting

As early as 1991, Park presented his research about electric load forecasting with an artificial neural network (ANN)[46]. However, deep neural networks were not even mentioned in a review of neural network methods for load forecasting in 2001 yet. [47, 48] claimed that few of ANNs applied to time series forecasting had theoretical support according to papers published between 2006 to 2016. But it is noticeable that recurrent neural network (RNN) began to be used for time series forecasting[49, 50]. Actually, unlike convolutional neural network (CNN) or ordinary multilayer perceptron (MLP) networks, RNN are quite suitable for time series forecasting according to its capability of learning and storing information of sequence data which is explicitly exhibited by its structure[51].

### 2.3.1 RNN-type Models

A simple RNN model connects each time steps by hidden states $h_t$, where $h_t = g(Wx_t + Uh_{t-1} + b)$. $x_t$ is the input at time $t$, $h_{t-1}$ is the hidden state at previous time $t - 1$, $b$ is bias, $W$, $U$ are weights, and $g$ is an activation function. Through hidden states, RNN enables autocorrelation of time series to be learned, but gradients vanish and explode especially when it comes to long sequences[52]. Thus, Long Short-Term Memory (LSTM)[53, 54] was developed to overcome this issue. LSTM is still an RNN type of neural network and connects inputs from a sequence. Whereas, each memory cell of LSTM contains three multiplicative gates, input gate, output gate and forget gate. Forget gate tunes the ratio of input added to the current cell, the input gate multiplies information from hidden states with scaled input in the current cell, and the output gate produces the hidden state to be passed and the output of the current state. So, LSTM is able to selectively remember information from previous time steps. [55] presented a bidirectional LSTM (BiLSTM) that processes a sequence from both starting and ending directions accelerates the processing speed and outperforms RNN and MLP. Furthermore, [56] proposed a deep LSTM which stacked LSTM layers to forecast petroleum production. Different from LSTM models, GRU models[57] only have two gates, update and reset gates. The update gate calculates a linear combination of the current input and previous hidden state and then activates the result by sigmoid function. The output from update gate will be used to decide weights of the hidden state at the previous time step and candidate hidden state at current time step. So, the update gate can decide to keep history information to which extent. Reset gate follows the same procedure as update gate to generate outputs, but reset gate is involved in candidate hidden state calculation as an element-wise multiplier of the hidden state at previous time step. Both LSTM and GRU are dedicated to solve the gradient vanish problem. It is hard to tell which of LSTM and GRU is better in sequence modeling task[58] though deep LSTM outperforms GRU in some application[56]. In general, LSTM and GRU are still RNN type models, so they still suffer from gradients vanishing and exploding issues though better than simple RNNs.

### 2.3.2 Transformer Neural Networks

There are some hybrid models of RNN-type models and traditional autoregressive models. For instance, Kim proposed a hybrid model that accepted multiple GARCH-type models' parameters together with other explanatory variables as inputs for LSTM and proved improvements of the prediction accuracy of LSTM models[59].

After Vaswani et al. first proposed a novel NN structure with attention mechanism[60], known as Transformer, for translation tasks, research into attention-based NN models booms for the attention mechanism's capability of capturing global dependencies between outputs and inputs. In [60], attention is a mapping function transforming a query(Q) and a set of key-value pairs to an output, and the output is the weighted sum of values. The weights of values are calculated by scaled dot-product attention as equation2.1,

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}} V) \tag{2.1}$$

where$K$ and $V$ are used to represent matrices of keys and values, $d_k$ is the dimension of $K$. The transformer neural network consists of several multi-head attentions which are composed of scaled dot-product attentions individually.

$$MultiHead(Q, K, V) = Concat(head_1, head_2, ..., head_h) \tag{2.2}$$

where each head represents a single attention defined in equation2.2.

Before the Transformer was invented, a typical attention mechanism used in neural networks computes weights of historical time steps or of different time horizons to improve forecasting but still based on existing RNN structures. In [61], an attention mechanism was added before decoder layers in an LSTM-based encoder-decoder NN. Hidden states $h_j$, as outputs of the encoder layer, were assigned weights calculated by the following attention mechanism.

$$e_{i,j} = Align(s_{i-1}), h_j) = s_{i-1}^T \odot h_j' \tag{2.3}$$

$$\alpha_{(i,j)} = \frac{\exp{(e_{i,j})}}{\sum_{k=1}^{T} \exp{(e_{i,k})}} \tag{2.4}$$

where $j$ is the position in inputs, $i$ is the position in outputs, $T$ is the length of time series, $s_{i-1}$ is the hidden state in the decoder, and $\alpha_{i,j}$ is the weight of $h_j$. The alignment model used for calculating intermediate variable $e_{i,j}$ was proposed in [62].

$$Align(s_{i-1}, h_j) = v^T tanh(Ws_{i-1} + Uh_j) \tag{2.5}$$

where $v, W, U$ are weight matrices. [61] proved that LSTM with attention improved standalone LSTM by experiments on five stocks data, but accuracy was calculated on predicted one step ahead while the NN model was potentially capable of forecasting multiple steps ahead. Moreover, higher variability in results was discovered due to complexity of the attention-based model. [63] applied the same attention mechanism to BiLSTM based encoder decoder network, but the model explicitly considered multivariate time series as input and was evaluated with 6-time-step-ahead prediction tasks. Attention-based BiLSTM outperforms traditional forecasting models, such as ARIMA, SVR, RNN, LSTM, and so on. Other than adding a single attention layer between encoder and decoder layers, [64] proposed multimodal attention which concatenated weighted hidden states at different time horizons from a RNN encoder and further passed combinations of weighted information to a BiLSTM decoder and directly to output layer respectively. The structure was tested for the day ahead forecast on the GEFCom2014 electricity price dataset and 3 months ahead forecast on JD50K online-sales dataset. The results of a BiLSTM model with multimodal attention were more accurate than results of either a BiLSTM encoder decoder model with single attention or a BiLSTM encoder decoder model without attention. Compared to scaled dot-product attention described in equation 2.1, this attention mechanism is a linear combination, which is later activated by the tanh function, of hidden states from the encoder and decoder. The typical attention mechanism mentioned above did not consider the interdependencies between time series and multivariate time series was treated as univariate time series. So temporal pattern attention (TPA) LSTM proposed in [65] defined another attention mechanism, called scoring function, to not only extract dependencies among time steps but also choose relevant time series. Supposing $H_i^c$ to be the concatenated hidden states at time step $i$ from multiple time series, the proposed scoring function calculates the weights of $H_i^c$ as defined below.

$$f(H_i^c, h_t) = (H_i^c)^T W_a h_t \tag{2.6}$$

$$\alpha_i = sigmoid(f(H_i^c, h_t)) \tag{2.7}$$

Where $W_a$ is a weight matrix, $h_t$ is the hidden state at time $t$, and $\alpha_i$ is the weight of $H_i^c$. Outputs of the TPA LSTM are $h_t'$.

$$h_t' = W_h h_t + W_v v_t \tag{2.8}$$

$$v_t = \sum_{i=1}^{n} \alpha_i H_i^c \tag{2.9}$$

From equations 2.6, 2.7, 2.8, and 2.9, it is obvious that the scoring function in TPA LSTM is quite similar to transformer's attention mechanism. While Transformer's attention mechanism was first applied to translation tasks, it is also suitable for time series forecasting since both sentences and time series are essentially ordered sequences. Q, K and V are initialized as matrices of history times series in a time series forecasting case. Weights calculated in attention can be used to present target time steps' dependencies of past time steps. Hence, global interdependencies among time steps and time series can be captured.

Huang et al. [66] embedded dual scaled dot-product attention in a novel NN, named as Dual Self-Attention Network (DSANet), to model interdependencies among different time series for global and local convolutional layers. Experiment results of DSANet were more promising than TPA. [67] adapted the original Transformer for long-interval multivariate time series forecasting. First, soft max layer of output was removed as it was used for classification task. Second, sine-cosine based positional encoding was replaced by one-hot encoding because [67] thought positional encoding impaired accuracy of continuous forecasting. Third, the log value of the conditional probabilistic distribution of outputs was set as additional cost function other than the common mean squared error (MSE) function. The adapted model in [67] was proved to be more accurate in multivariate time series forecasting than original Transformer model. [68] put forward a Temporal Fusion Transformer (TFT) structure which was similar to [64], but used multi-head attention to calculate weights of time steps and series instead, static covariates encoder for condition temporal dynamics and a variable selection module named as Gated Residual Network (GRN) was implemented. As stated in the paper, TFT is an interpretable model and can be used to detect regime changes or important events by comparing averaged attention levels over forecasting horizons.

As shown in the equation 2.1, the computational complexity of scaled dot-product attention is $O(n^2 \cdot d)$, where n is sequence length and d is the number of time series, while the computational complexity of RNN is $O(n \cdot d^2)$. Thus, when it comes to long sequence forecasting and the dimension of features is relatively small, which is typical for multivariate time series, the computational cost of the Transformer is much higher than RNN. To solve the problem, Zhou et al.[69] designed a new transformer-based model, Informer, to overcome this issue. Informer reduced computational complexity to $O(n \cdot \ln(n))$ by a prob sparse self-attention mechanism and further accelerated the computation by self-attention distilling operation. Moreover, the Informer predicted long sequences in one step instead of a step-by-step way by a generative decoder design, improving the inference speed of long sequence prediction. These advantages were shown when tested on long sequence forecasting. On the other hand, [70] also proposed a Residual Attention Layer Transformer (RealFormer) model to make the attention matrix sparser for later layers and stabilize the training. RealFormer just added a direct path between multi-head attention and the next layer, skipping activation and norm transformation, which is simple and cheap.

To specifically address long-term forecasting of time series, [71] created an innovative Auto-Correlation mechanism, inspired by scaled dot-product attention, to decompose the time series as long-term trend-cyclical patterns and short-term seasonal patterns with embedded fast Fourier transformation. This model, named Autoformer, outperformed the original transformer in both efficiency and accuracy. [72] adopted another strategy, combining typical attention 2.4 and scaled-dot product attention 2.1 as a new attention mechanism, Pyramidal Attention Module (PAM). PAM, which is the core of the model (Pyraformer) proposed in [72], is claimed to be capable of obtaining temporal features at different scales, and it achieved higher accuracies than Informer in long-time forecasting tasks.

In general, how to improve training efficiency, and capture temporal patterns at different time horizons will be major topics for transformer-type models.

## 2.4 Probabilistic MTS Forecasting

Weron reckoned in 2014 that probabilistic forecasting would be one of the hotspots in electricity price forecasting for the next decades [13], and this has already become true [14]. Probabilistic forecasts can be expressed as the regression in equation 2.10 [14, 73],

$$y_t = \hat{y_t} + \epsilon_t \tag{2.10}$$

where $\hat{y_t}$ is the point prediction at time $t$, and $\epsilon_t$ is the error term. Common forecasting methods just provide the prediction of point value $\hat{y_t}$, while most probabilistic forecasting models provide a prediction interval (PI) of $\hat{y_t}$. Methods to construct a probabilistic forecasting problem can be classified into 4 categories [14],

(1) Calculating PIs based on historical data,

(2) Forecasting based on distributions,

(3) Forecasting bootstrapped PIs,

(4) Quantile regression (QR),

The first category just calculates the PIs of historical data and is independent of models [74, 75, 76, 77]. In the second category, it is often assumed that the error term $\epsilon_t$ or variance follows a certain distribution, and two models are built for $\epsilon_t$ and point values separately. Subsequently, PIs are generated by combining predicted point values and PIs of error, which is often assumed to be Gaussian distribution [14]. The second category was called distribution-based probabilistic forecasts, but it can be easily mistaken as a distribution of observations. So, this category is renamed as error density forecasts in this study. The third category, bootstrap PIs, is often used in NN models to recursively update the forecasted values based on preliminary guesses of parameters. The last category, QR, sets up a regression function for a certain quantile of the target variable distribution. While the first category is an empirical and simple method, literature and advances in the other three categories are elaborated in detail. Bayesian approach is widely used in probabilistic forecasting and it is often applied with deep learning methods to provide probabilistic forecasts [78, 79, 80], a section for Bayesian inference was added in this study.

### 2.4.1 Error Density Forecasts

Normally, models in this category fit the distribution of deviations or errors instead of point values. PIs of error are calculated according to the density function of the distribution. Error density forecasts are usually complementary to point value forecasts, especially to traditional time series forecasts where only time dependencies are captured. For instance, [80] applied VAR to capture autocorrelations of time series and proposed a model to fit noise with a skewed t distribution according to the distribution of spikes in electricity prices. This method has some limitations. First, the choice of error distribution is crucial to the accuracy of forecasts, but the choice depends on the researchers' empirical knowledge and statistical analysis. Secondly, if the regression model does not fit the data set well, the residuals of the model cannot be treated as pure noise.

### 2.4.2 Bootstrapped Prediction Intervals

Bootstrapped prediction intervals were calculated through an error sampling process. The process can be summarized as following[14, 81, 82, 83].

(1) Initialize a point value prediction model with parameters $\hat{\theta}$ and achieve true residuals $e^*$.

(2) Sample $e_j^*$ from an empirical noise distribution and obtain $b_j^* = e^* - e_j^*$.

(3) Repeat the second step for n times.

(4) Calculate prediction intervals with $\frac{\sum_{j=1}^{B} b_j^*}{n}$

This method considers the uncertainty that cannot be parameterized in real data, but computational cost increases with repetition number n, which should be close to the sample size[81].

### 2.4.3 Quantile Regression

The response of traditional regression is a conditional mean of the target $y_t$, however, the response of quantile regression (QR) is a conditional quantile function of target $y_t$. Correspondingly, the parameters of QR are estimated by pinball loss function (see section 2.5) which compares the actual value at a certain quantile of distribution and the predicted value at the same quantile of the distribution. QR has proved to be a robust regression method less sensitive to outliers and widely applied in economic and energy fields[78, 84, 85]. Quantile regression averaging (QRA) proposed in [14] is one of the well-known QR methods. QRA is a simply weighted combination of more than one quantile regression model. [79] proposed Bayesian regularized quantile regression which outperformed pure QR by adding whether lasso or elastic net penalty. To solve the problem of nearly constant intervals, conformalized quantile regression (CQR) was put forward[26]. The data was split into two disjoint sample sets at first, and then one of them was used for training quantile regressor for upper and lower quantiles, and another for calibrating prediction intervals to meet the predetermined threshold of interval length. Therefore, CQR achieved shorter intervals than NN, ridge regression, quantile NN, and quantile random forest. [86] utilized RNN to estimate the parameters of a predefined quantile function which consists of a family of linear splines and the objective function is CRPS. This method made quantile regression more flexible and easily adaptive to local changes.

### 2.4.4 Bayesian Approach

The Bayesian approach for MTS forecasting combines the principles of Bayesian inference with models designed to capture the interdependencies and dynamics among multiple time series. There are different ways to interpret the origins of uncertainty in probabilistic time series forecasting problems, so various conditional distributions were defined for probabilistic forecasting. Common ways to embed uncertainties include (1) Bayesian Vector Autoregressive models; (2) Bayesian State Space Models; (3) Multivariate Gaussian Process Regression; (4) Bayesian Multivariate time series forecasting with Copulas and (5) Bayesian deep learning models.

Bayesian vector autoregressive (BVAR) and similar models, such as Bayesian Vector Autoregressive moving average (VARMA), assume that prior knowledge is reflected in the parameters of autoregressive time series models[87, 88]. In contrast to traditional multivariate time series models like VAR and VARMA 2.2, where correlations between variables are constant, Bayesian linear regression models consider parameters in these linear time series models to follow distinct distributions. Bayesian linear regression models are particularly advantageous when there is existing prior information, allowing practitioners to gauge the credibility of the models. They also offer flexibility as each parameter has the potential to adhere to a distinct distribution [89, 90]. However, when dealing with a large number of dependent variables, the computational complexity increases. Moreover, the prior assumptions about parameter distributions play a crucial role. Finally, it's important to note that BVAR models are fundamentally based on linear relationships between variables, which may not always hold in real-world scenarios.

A Bayesian state-space model characterizes the evolution of a system over time using state variables. One of its strengths lies in the ability to independently model different aspects of the series and then integrate them into an overarching model. State-space models are highly adaptable and find applications in a wide range of scenarios, from autoregressive integrated moving average models to models with unobserved components and smoothing models incorporating penalties for irregularities [91, 92, 93]. However, selecting appropriate state transition and observation equations can be challenging, and the modeling outcomes are influenced by the chosen prior distributions for parameters. Additionally, assuming linear relationships between variables may not always be applicable in real-world contexts.

Gaussian Process (GP) regression, a well-known Bayesian nonparametric model, relies on parameters such as mean and covariance functions (also known as kernels) for multivariate normal distributed data [94, 95]. Multivariate Gaussian Process Regression offers a flexible and probabilistic framework for modeling intricate relationships between input and output variables, capable of capturing dependencies across multiple output dimensions [96, 97]. The choice of different kernels in GP allows for accommodating diverse correlation patterns in time series, enabling the handling of non-linear relationships between variables. However, as the volume of data increases, the computation of covariates becomes computationally demanding. To address this, scalable and variational methods have been developed to enhance GP efficiency by approximating kernels rather than calculating exact values [98, 99, 100]. It's worth noting that Gaussian process regression is tailored for continuous variables, necessitating additional steps for handling discrete variables.

Much like Gaussian process regression, the copula-based multivariate model assumes a collective distribution of variables. This distribution can be broken down into individual distributions, with the copula serving as the function that links these individual distributions to the joint distribution [101]. This methodology incorporates existing knowledge about the relationships between variables and is adaptable to model various dependencies beyond simple linear correlations. Conversely, in cases where prior information is unavailable, selecting appropriate copula models becomes challenging, and the choices of associated parameters become crucial. Moreover, copula models rely on the assumption of stationarity [102] and may face difficulty in accurately modeling extreme values. They can also be prone to overfitting due to the complexity of the models.

Among various Bayesian deep learning models, variational autoencoder (VAE) is particularly suitable for probabilistic forecasting. VAE returns a posterior distribution of a latent variable given the data and it can be used to provide probabilistic forecasts by just setting the loss function for distributions, such as negative Gaussian log-likelihood, and Kullback-Leibler (KL) divergence[103, 104]. Temporal latent Auto encoder proposed a scalable matric factorization method to decompose latent variables which boosts the model efficiency, and an LSTM layer was used to learn nonlinear relationships of factors[103]. Using Variational Autoencoders (VAEs) for time series forecasting comes with several challenges. VAEs may produce somewhat blurry outputs, limiting their ability to generate sharp predictions. These models might struggle with capturing highly complex or multimodal data distributions, potentially hindering their effectiveness in diverse datasets. The phenomenon of ”posterior collapse” can occur, leading to less diversity in generated samples. VAEs are also sensitive to hyperparameter choices, and achieving the right balance between reconstruction accuracy and regularization terms can be challenging. Additionally, VAEs may face difficulties in capturing long-term dependencies in sequential data, and their probabilistic nature might introduce challenges in interpreting learned representations. Despite their potential, VAEs demand careful consideration of these limitations and thoughtful parameter tuning for effective application in time series forecasting.

## 2.5 Performance Metrics of Probabilistic Forecasting

As probabilistic forecasting provides prediction intervals or values at specific quantiles rather than single point estimates, the metrics used to assess probabilistic prediction accuracy also differ. One widely used metric for quantile regression is the Pinball loss [14, 105, 106]. The Pinball loss is defined as follows: $y_t$ represents the actual value at the quantile $\alpha$, while $\hat{y_t}$ denotes the predicted value at the same quantile $\alpha$.

$$Q_\alpha(y_t, \hat{y_t}) = \begin{cases} \alpha(y_t - \hat{y_t}) & if\ y_t - \hat{y_t} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

[64] predicted different quantiles of JD50KSales or GEFCom2014 electricity price with pinball loss. It calculates the difference between actual quantile values and forecasted quantile values, so pinball can only be used for a certain quantile regression. If the quantile changes, a new quantile regression model should be built. TFT

paper [68] applied a modified quantile loss (QL) function, which was proposed in [107] and similar to pinball, that combined two segmented functions together and chose the maximum value between 0 and loss. In this way, predicted quantile values below the target quantile values are not considered. Unlike pinball or QL, Winkler score[108] compares prediction intervals (PIs) for probability forecasting and selects the narrowest one by comparing differences between point values and lower, upper bounds. In addition, the Continuous Ranked Probability Score (CRPS) is also a commonly compared measurement[109]. CRPS derived from [110] calculates the differences between cumulative distribution function (CDF) and indicator function directly, and it is the same as mean absolute error (MAE) when applied to deterministic forecasting. CRPS is one of the most popularly used metrics in probabilistic forecasting[86, 109, 103]. There are some other probabilistic evaluation methods, such as logarithmic score (LogS) and variogram score (VarS) [111] as well, but LogS is very sensitive to tails and VarS only evaluates correlation and variance, not the mean of a distribution.

## 2.6 Performance Metrics of Uncertainty Quantification

Following the derivation of prediction intervals from probabilistic forecasting, a calibration process is undertaken to ensure a statistically guaranteed coverage rate. As outlined in [112], an effective prediction interval should encompass the anticipated rate of targets while maximizing sharpness. In this investigation, the sharpness of prediction intervals will be assessed through bandwidth measurement. The coverage rate, also frequently refered to as prediction interval coverage probability (PICP) [113] is computed using Equation 5.12. $n$ is the number of samples, and $\theta_i^\alpha$ denotes whether the $i$th sample is covered by the prediction interval at $100(1-\alpha)\%$ level. $\theta_i^\alpha$ equals 1 if $i$th sample is covered by PI, and 0 otherwise.

$$PICP = \frac{1}{n}\sum_{i=1}^{n}\theta_i^\alpha \tag{2.12}$$

Bandwidth, or prediction interval average width (PIAW) [114], is determined by Equation 5.13. $n$ represents sample counts as well. $U_i^\alpha$ and $L_i^\alpha$ denote the upper bound and lower bound of the prediction interval at $100(1-\alpha)\%$ level.

$$PIAW = \frac{1}{n}\sum_{i=1}^{n}(U_i^\alpha - L_i^\alpha) \tag{2.13}$$

PICP and PIAW are used to measure the performances of conformal prediction.

# CHAPTER 3

# DAY-AHEAD ELECTRICITY PRICE FORECASTING WITH MULTIVARIATE TIME SERIES TRANSFORMER

## 3.1 Introduction

Electricity price forecasting (EPF) has been a prevalent research topic for years [115, 116, 117, 118, 119] due to its potential to bring significant value to electricity markets. According to statistics from U.S. Energy Information Administration (EIA) [120], the revenue generated from the sale of electricity to ultimate customers amounted to 393,639 million dollars in 2020 and increased by 7.3% to reach 422,323 million dollars in 2021. While there are tremendous benefits in electricity markets, the growing volatility of electricity prices puts market participants at higher risks. The emergence of renewable energy sources, such as wind and solar power, is one of the contributing factors to this volatility. According to the U.S. Energy Information Administration, renewable energy accounted for 13.7% of the total net electric power generation in 2021, representing a 13.2% increase from the previous year [120]. Additionally, the deregulation of the electricity markets in the United States, which began in the 1990s, has created a more competitive environment compared to regulated markets. While customers in deregulated markets may benefit from lower electricity prices most of the time, supply shortages can lead to unexpectedly high prices. Against this backdrop, this study focuses on the ERCOT market, a deregulated market that is increasingly impacted by fluctuations in wind power as opposed to electricity demand [121]. To assist market participants to hedge against fluctuations in electricity prices and make more knowledgeable decisions, a day-ahead EPF model was created for the wholesale power market run by the ERCOT in this study.

A comprehensive review published by Weron divided EPF approaches into five categories including game theory models, fundamental methods, reduced-form models, statistical models, and machine learning methods [13]. Nevertheless, in light of potential application scenarios and advances in these approaches, this study mainly discusses and compares three categories of state-of-the-art EPF methods as summarized by Lago [119], namely statistical, machine learning, and hybrid methods.

Most statistical methods are dedicated to building linear regressions based on correlations among time steps and usually assume that predictors are independent. But factors contributing to the changes in electricity prices are complex and not necessarily independent, such as factors in the form of multivariate time series (MTS). So, typical deep learning methods for time series forecasting, such as RNN, LSTM, and GRU, which can learn nonlinear relationships become popular nowadays [122, 123]. Nevertheless, RNN-type deep learning models face gradient vanishing or exploding issues in long-sequence time series forecasting. Aside from RNN-type models, support vector machine (SVM) is also one of the classic machine learning methods and it achieves excellent results for non-linear time series prediction [124, 125]. Howbeit, SVM is essentially designed to solve quadratic programming problems, so heavy computation becomes prominent when dimensions of data expand [45]. Hybrid methods which normally consist of more than one algorithm are one of the popular EPF topics in accord with the number of published papers as well [14]. Despite the popularity of hybrid methods, it is hard to measure their accuracy against other state-of-the-art methods since most of them either avoided the comparison or were compared with outdated methods [119].

To address these limitations, this work adopts a transformer neural network. This network is particularly adept at modeling intricate multivariate time series and forecasting multiple steps ahead. Transformer, a trendy neural network architecture, was proven to outperform regular RNNs in multilingual neural machine translation tasks[126] and it is a relatively new topic in the EPF field [127, 128, 129]. Even though there are comparisons of transformer and RNN-type models in existing works [127, 128], the hyperparameter optimization process is not mentioned and the datasets they used are rather regular compared to electricity prices in the ERCOT market. A fair comparison of the transformer with RNN-type neural networks based on hyperparameter optimization is presented in this work. Besides, supervised training combined with unsupervised fine-tuning which was proved to be superior to single training was applied in the study [130].

Furthermore, spikes, which are unexpectedly high values in electricity prices, are much higher in the ERCOT market than in other markets such as Pennsylvania-New Jersey-Maryland Interconnection (PJM), California Independent System operator (CAISO), and Midcontinent Independent System Operator (MISO). So, while some research did not specifically address the issue of spikes [15, 16, 17, 18], many case studies of the ERCOT market involved separate predictions of peak and off-peak prices [131, 132] or decomposition of time series of electricity prices [133]. In this study, the spikes are transformed into lower magnitudes which are still considered high relative to normal prices. By incorporating the transformed spikes into the inputs, the model we developed can not only focus on normal price prediction but also catch the pattern of continuous time series.

### 3.1.1 Motivation and Contributions

As mentioned above, it is difficult for traditional statistical methods to handle complex multivariate time series, and usually, more than one model is needed for multiple-step-ahead forecasting. Meanwhile, RNN-type models have the problem of gradient vanishing or exploding in long-sequence forecasting. So, our first motivation is to apply the transformer structure which is able to capture complicated correlations and avoids gradient vanishing or exploding problems by calculating the weight of each data point in both time and feature dimensions. In addition, past works lack a fair comparison of transformer and RNN-type models with hyperparameter optimization on the models. Our work tried to provide comparisons between common RNN-type models and transformer networks. Last but not the least, the number of case studies on the ERCOT market for EPF is comparatively small, and most of them predicted electricity prices with hybrid models due to frequent occurrences of challenging spikes [131, 132, 133]. The spikes of electricity prices in ERCOT can be as high as \$9000, while 90% of electricity prices are below \$40.5 according to the hourly real-time SPP from 2017 to 2021.

In the consideration of above motivations, this study contributed to the following aspects:

(1) A multivariate time series transformer for day-ahead Electricity Price Forecasting (EPF) through a two-step process: unsupervised pretraining followed by supervised finetuning was employed. The unsupervised pretraining phase enhances the convergence of the finetuned model, mitigates overfitting concerns, and enables the model to capture more generalized data patterns. Moreover, the pretraining process contributes to the efficiency of model updates.

(2) Fair comparisons of our model, RNN, LSTM, and GRU with a hyperparameter optimization method, Bayesian Optimization Hyperband (BOHB) were done.

(3) A case study was conducted on the challenging electricity market, ERCOT. In order to address the difficulty of modeling frequent spikes and maintaining continuous time series patterns simultaneously, an innovative spike transformation was applied, effectively enhancing the accuracy of predictions.

(4) The incorporation of resource nodes' SPP as predictors did not enhance the prediction of the load zone's SPP, potentially because spikes of resource nodes and the load zone occurred nearly simultaneously.

### 3.1.2 Chapter Structure

The chapter is structured as follows: section 2 presents a comprehensive review of state-of-the-art electricity price forecasting methods. Section 3 introduces the studied problem and provides a detailed illustration of our proposed model. Section 4 presents and discusses the details of the ERCOT case study, including the data, hyperparameter optimization, evaluation metrics, and results. Finally, in section 5, we draw conclusions based on our findings.

## 3.2 Literature Review

### 3.2.1 Statistical Methods

Various statistical methods are available, such as vector autoregressive (VAR) models, vector autoregressive moving average (VARMA) models, and their related variants. These models are designed to enhance their performance and broaden their scope of application. Vector autoregressive integrated moving average (VARIMA) models expand the application context of VARMA by differencing time series to achieve stationarity [33, 34]. Nevertheless, the order of difference needs to be decided through trial and error, and it is not guaranteed that stationarity exists in time differences. Seasonal ARIMA(SARIMA) multiplies nonstationary seasonal patterns with stationary patterns to fit seasonally changed time series [35], but each quarter of a year needs a different model. To boost the computational efficiency of the parameter selection process in VAR, VAR-LASSO utilized the property of lasso regression [36], selecting and estimating parameters at the same time [37, 38]. On the other hand, for VAR models with lots of zero values in the coefficient matrix, sparse VAR (sVAR) identifies non-zero coefficients by estimating partial spectral coherence at first and then refines results which accelerates computation as well [39]. In addition, the generalized autoregressive conditional heteroskedasticity (GARCH) model, a variant of autoregressive moving average (ARMA), fits time-dependent error variances, instead of the time series itself [40]. Multivariate GARCH models can deal with nonstationary MTS with time-varying volatility and have been successfully applied to the financial domain as useful decision tools, such as risk management and asset pricing [41, 42]. Besides, exogenous factors which may contribute to time series changes are introduced to ARIMA models to improve forecasting accuracy and these models are known as ARIMAX models [43].

Due to the exponential growth of data size and complexity in MTS, the forecasting capabilities of many conventional MTS models are limited by the subsequent increase in computational costs, the intricacy of multivariate time series, and the underlying assumptions of stationarity. Comparatively, deep learning methods gain popularity for their competence in modeling complicated nonlinear relationships in MTS.

### 3.2.2 Machine Learning Methods

As early as 1991, Park presented his research about electric load forecasting with an artificial neural network (ANN) [46]. However, deep neural networks were not even mentioned in a review of neural network methods for load forecasting in

2001 yet [47] and [48] claimed that few of ANNs applied to time series forecasting had theoretical support according to papers published between 2006 to 2016. However, it is noticeable that recurrent neural networks (RNNs) began to be used for time series forecasting [49, 50]. Actually, unlike convolutional neural networks (CNNs) or ordinary multilayer perceptron (MLP) networks, RNNs are quite suitable for time series forecasting according to their capability of learning and storing information of sequence data which is explicitly exhibited by their structure [51].

A simple RNN model connects each time step by hidden states. Through hidden states, RNN enables autocorrelation of time series to be learned, but the information in the history cells gradually disappears with the increase of inputs' length[52]. Thus, LSTM [53, 54] was developed to overcome this issue. LSTM is still an RNN-type neural network, whereas, each memory cell of LSTM contains three multiplicative gates, input gate, output gate, and forget gate. Forget gate tunes the ratio of input added to the current cell, the input gate multiplies information from hidden states with scaled input in the current cell, and the output gate produces the hidden state to be passed and output of the current state. So, LSTM is able to selectively remember information from previous time steps. [55] presented a bidirectional LSTM (BiLSTM) that processes a sequence from both starting and ending directions and accelerates the processing speed and outperforms RNN and MLP. Furthermore, [56] proposed a deep LSTM that stacked LSTM layers to forecast petroleum production.

Different from LSTM models, GRU models [57] only have two gates, update and reset gates. The update gate calculates a linear combination of the current input and the previous hidden state and then activates the result by the sigmoid function. The output from the update gate will be used to decide the weights of the hidden state at the previous time step and the candidate hidden state at the current time step. So, the update gate can decide to keep history information to which extent.

The reset gate follows the same procedure as the update gate to generate outputs, but the reset gate is involved in the candidate hidden state calculation as an element-wise multiplier of the hidden state at the previous time step. Both LSTM and GRU are dedicated to solving the gradient vanishing problem. It is hard to tell which of LSTM and GRU is better in sequence modeling tasks [58] though deep LSTM outperforms GRU in some applications [56]. In general, LSTM and GRU are still RNN-type models, so they still suffer from gradients vanishing and exploding issues though better than simple RNNs.

There are some hybrid models of RNN-type models and traditional autoregressive models. For instance, Kim proposed a hybrid model which accepted multiple GARCH-type models' parameters together with other explanatory variables as inputs for LSTM and proved improvements in the prediction accuracy of LSTM models [59]. But they are trapped in the limitation of RNN-type models.

Transformer neural networks are one of the most widely used neural network structures recently. They jump out of the scope of CNNs or RNNs and rely on the attention mechanism. After Vaswani et al. first proposed a novel NN structure with the attention mechanism [60], known as transformer neural networks, for translation tasks, research into attention-based NN models booms for the attention mechanism's capability of capturing global dependencies between outputs and inputs. While the transformer's attention mechanism was first applied to translation tasks, it is also suitable for time series forecasting since both sentences and time series are essentially ordered sequences.

Other than deep learning models, the support vector machine (SVM) is also a classic model for time series forecasting and achieved excellent prediction results for non-linear time series [44]. Furthermore, a multi-output SVM for multistep ahead forecasting (MM-SVM) was put forward to cover its unstable spatiotemporal forecasting deficiency [134]. Nevertheless, SVM is primarily intended to address a quadratic programming problem, and as the dimensionality of the data increases, significant computation becomes necessary [45].

### 3.2.3 Hybrid Methods

Since variations in electricity prices are quite irregular and originate from complicated factors, many hybrid methods were invented for the purpose of disintegrating EPF problems into simpler sub-problems. These hybrid methods usually contain more than one of the following modules [119]:

(1) data decomposition

(2) feature selection

(3) clustering

(4) one or more prediction models

(5) parameter or hyperparameter optimization

The complex EPF problem is decomposed into easier sub-tasks by those modules. For example, time series of electricity prices can be decomposed into simpler signals by popular signal processing methods, such as variational mode decomposition, then forecasting results of each signal are combined together to generate final forecasts[135, 136, 137]. A hybrid method can consist of different combinations of modules corresponding to specific datasets. Thus, it is hard to find the most representative hybrid methods and they are normally not compared to other SOTA methods.

## 3.3 Proposed Method

### 3.3.1 Problem Formulation

The objective of this study is to develop a model that can forecast the SPP for the next $p$ hours (from $t$ to $t+p-1$) based on $k$ observed features from the preceding $h$ hours (from $t-h-l$ to $t-l-1$). This is illustrated in Figure 3.1, where $l$ represents the lead time.

If we denote $y_t$ as the SPP for the north load zone and $x_t$ as a set of features with $k$ dimensions observed at time $t$, the multi-step ahead electricity price forecasting problem can be formulated as follows:

$$(y_t, y_{t+1}, \ldots, y_{t+p-1}) \;=\; f\,(x_{t-l-h}, x_{t-l-h+1},\; ..., x_{t-l-1}) + \varepsilon_t \tag{3.1}$$

In this context, $\varepsilon_t$ represents the random error term, $p$ signifies the duration of the prediction horizon, and $h$ denotes the extent of historical time steps.

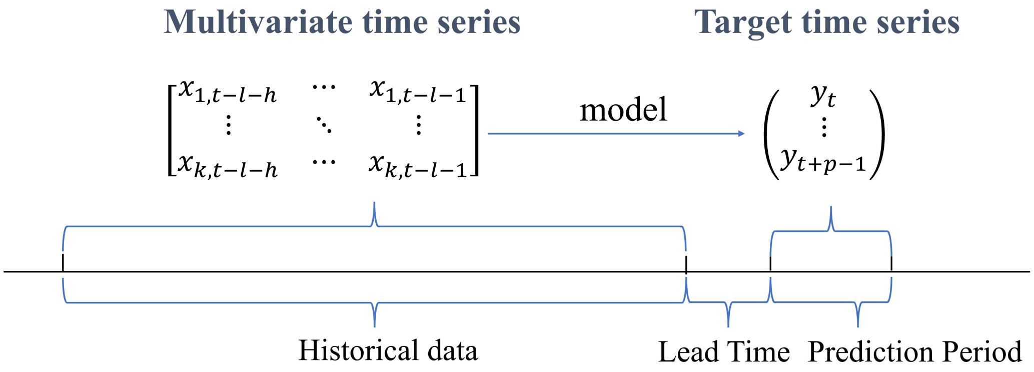

Multivariate time series
Target time series
$\begin{bmatrix} x_{1,t-l-h} & \cdots & x_{1,t-l-1} \\ \vdots & \ddots & \vdots \\ x_{k,t-l-h} & \cdots & x_{k,t-l-1} \end{bmatrix}$
model
$\begin{pmatrix} y_t \\ \vdots \\ y_{t+p-1} \end{pmatrix}$
Historical data
Lead Time
Prediction Period

Figure 3.1. The problem definition.

### 3.3.2 Workflow

To address the issue outlined in the preceding section, we have put forth a workflow depicted in Figure 3.2. The workflow incorporates a spike-focused continuous data transformation to minimize the impact of spikes on the SPP data. After this transformation, the resulting SPP and other variables, such as date, time, and temperature, were utilized as multivariate time series inputs for the TDEPF model. Finally, the spikes were reverse-transformed to obtain the day-ahead hourly SPP transformation.

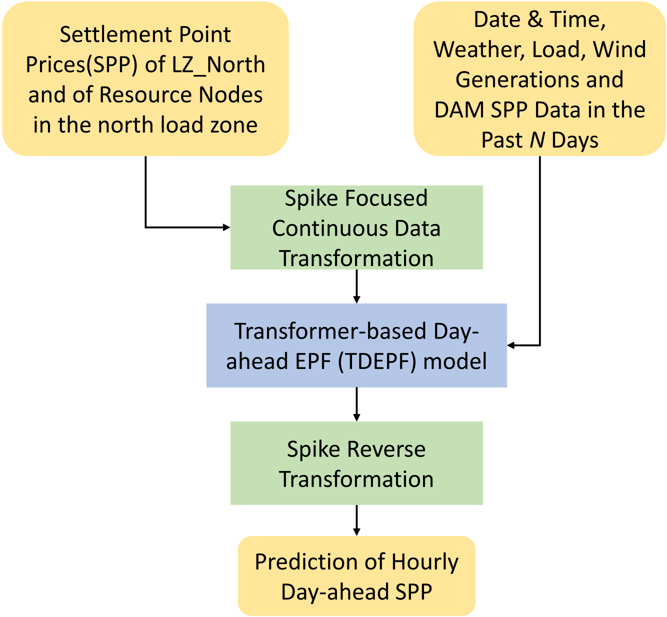

Settlement Point Prices(SPP) of LZ_North and of Resource Nodes in the north load zone
Date & Time, Weather, Load, Wind Generations and DAM SPP Data in the Past *N* Days
Spike Focused Continuous Data Transformation
Transformer-based Day-ahead EPF (TDEPF) model
Spike Reverse Transformation
Prediction of Hourly Day-ahead SPP

Figure 3.2. Workflow of transformer-based day-ahead electricity price forecasting.

### 3.3.3 Spike Transformation

Spikes refer to sudden and significantly high values in electricity prices that quickly return to their average levels. These spikes are inevitable due to the nature of electricity and the regulatory framework governing price bidding. Typically, strategies such as substituting with neighboring prices, using data from similar days, or applying user-defined thresholds are employed to address spikes [76]. In order to maintain the continuity of the time series data and reduce the impact of spikes, a logarithmic transformation, as demonstrated in Equation 3.2, was applied to the electricity prices. If the initial electricity price $y_t$ surpasses the upper limit $ub$, it will be subjected to a logarithmic adjustment and then multiplied by a factor that guarantees the adjusted value remains higher than $ub$. The selection of $ub$ will be discussed in the 3.4. This transformation effectively dampened the effect of spikes, allowing normal prices to remain within the same distribution. Consequently, the forecasting model for normal prices remains unaffected by the presence of spikes.

$$y'_t = \begin{cases} y_t & \text{if } y_t \leq ub \\ \frac{ub}{log_{10}(ub)} log_{10}(y_t) & \text{otherwise} \end{cases} \tag{3.2}$$

The benefit of employing this transformation is that the original values $y_t$ can be derived from the transformed values $y'_t$ using the inverse transformation equation 3.3.

$$y_t = \begin{cases} y'_t & \text{if } y'_t \leq ub \\ 10^{\frac{y'_t \cdot log_{10}(ub)}{ub}} & \text{otherwise} \end{cases} \tag{3.3}$$

The determination of $ub$ value is essential to the spike transformation, and two methods were explored. Two different approaches were experimented with for setting this upper bound: spike transformation method 1 (ST1), which utilizes the Interquartile Range (IQR) method, and spike transformation method 2 (ST2), which involves using the mean and standard deviation.

ST1 defines the upper bound as $1.5 \times (Q3 - Q1) + Q3$ as shown in Equation 3.4, where $Q3$ is the third quartile and $Q1$ is the first quartile of training SPP. So, ub defined by Equation 3.4 is 36.761\$/MWh. ST2 defines the upper bound as Equation 3.5, where $\hat{\mu}$ and $s$ represent the mean and standard deviation of training SPP. The upper limit is set as 148.680\$/MWh using ST2.

$$ub = 1.5(Q3 - Q1) + Q3 \tag{3.4}$$

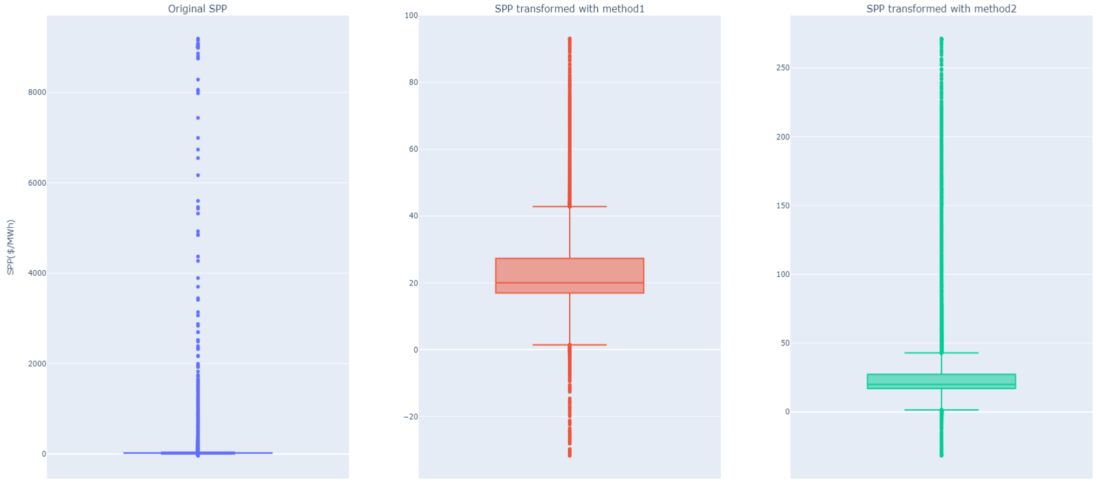


Figure 3.3. Comparisons of spike transformation with two upper bounds(the left with the original SPP, the middle with SPP transformed by ST1, and the right with SPP transformed by ST2).

$$ub = \hat{\mu} + s \tag{3.5}$$

In Figure 3.3, comparisons of the distributions of the original training SPP, training SPP transformed by ST1, and transformed by ST2 are presented. The boxplot of the original training SPP reveals numerous extremely high outliers, causing the majority of normal values to be compressed. In contrast, the SPP transformed by ST1 exhibits a relatively balanced distribution when compared to the original SPP and the SPP transformed by ST2.

### 3.3.4 Base Model

The fundamental structure of the TDEPF base model is a typical transformer structure which comprises an embedding layer, a positional encoding layer, multiple encoder layers, and a linear transformation layer, akin to component (a) illustrated in Figure 3.4. However, during the fine-tuning stage, there is a modification made to the final linear layer.

As shown in Figure 3.4 (b), the input and the output of the embedding layer are $x \in R^{h \times k}$ and $E \in R^{h \times d}$, where $h$ is the number of historical time steps, $k$ is the number of features, and $d$ is the number of embedded dimensions. Equation 3.6 describes the linear transformation in the embedding layer, where $W_e \in R^{k \times d}$ is a weight matrix and $b_e \in R^{h \times d}$ is a bias matrix. The embedding layer transforms the input $x$ from $k$ to $d$ dimensions which is a hyperparameter that can be tuned.

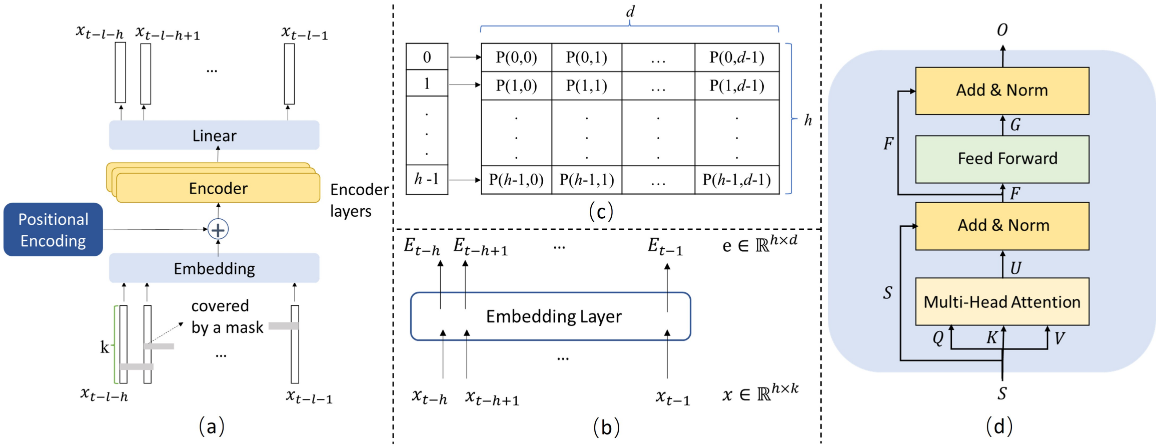


Figure 3.4. (a) TDEPF structure for unsupervised pretraining. (b) Inputs and outputs of the embedding layer. (c) Positional encoding of $h$ time steps and $d$ feature dimensions. (d) The encoder structure..

$$u_t = x \cdot W_e + b_e \tag{3.6}$$

The positional encoding layer stores the positions of each data point with sine and cosine functions as shown in Equation 3.7, where $2i$ and $2i+1$ represent even and odd positions in the feature dimension, $t$ represents the index in the time dimension, and $n$ is the number of training samples. Meanwhile, $i$ is an integer no less than 0 and smaller than half the embedded feature dimension, $d/2$. $c$ is a scalar and set as 10,000 in default. Figure 3.4 (c) illustrates the encoded matrix of positional encoding where $d$ is the feature dimension and $h$ is the length of history time steps.

$$\begin{gathered} pos\left(t,2i\right) = sin\left(\frac{t}{c^{2i/d}}\right) \\ pos\left(t,2i+1\right) = cos\left(\frac{t}{c^{2i/d}}\right) \\ i,t \text{ are intgers}, \quad i \in [0,d/2),\ t \in [0,h-1] \end{gathered} \tag{3.7}$$

Each encoder is composed of multi-head attention, 2 add and norm layers, and a feed-forward layer as illustrated in Figure 3.4 (d). The input of the first encoder, $S$, is the sum of outputs from the positional encoding layer and embedding layer which are $P$ and $E$ as shown in Equation 3.8. Thus, $S$ is the same dimension as $P$ and $E$. Through three different linear transformation, $S$ are transformed into $Q_i$, $K_i$, and $V_i$ as in Equations 3.9 - 3.10 and they are inputs of attention head $i(i = 1, 2, \ldots, m)$ as shown in Figure 3.4 (d). The number of attention heads $m$ is determined by the hyperparameter tuning process. As in Equation 3.12, for each attention head, the product of $Q_i$ and ${K_i}^T$ is divided by the squared root of the number of embedded features $d$. The scaled product matrix is converted into probabilities by the softmax function (Equation 3.13) as weights for values $V_i$.

$$S = pos + E, \quad S, P, E \in R^{h\times d} \tag{3.8}$$

$$Q_i = S \cdot W_{Q_i} + b_{Q_i}, \quad W_{Q_i} \in R^{d\times d}, \quad b_{Q_i} \in R^{h\times d} \tag{3.9}$$

$$V_i = S \cdot W_{V_i} + b_{V_i}, \quad W_{V_i} \in R^{d\times d}, \quad b_{V_i} \in R^{h\times d} \tag{3.10}$$

$$K_i = S \cdot W_{K_i} + b_{K_i}, \quad W_{K_i} \in R^{d\times d}, \quad b_{K_i} \in R^{h\times d} \tag{3.11}$$

$$head_i = softmax\left(\frac{Q_i \cdot {K_i}^T}{\sqrt{d}}\right) V_i, \quad i = 1, 2, \ldots, m \tag{3.12}$$

$$softmax\left(\alpha_i\right) = \frac{e^{\alpha_i}}{\sum_{j=1}^{n} e^{\alpha_j}} \tag{3.13}$$

Multi-head attention is a linear combination of multiple attention heads as Equation 3.14, where $Q$, $K$, and $V$ are ensembles of $Q_i$, $K_i$, and $V_i$ respectively and $U$ is the output of this layer. We use $F$ to represent outputs of add and norm layer as Equation 3.15, and the sum of outputs of multi-head attention $U$ and embedding layer $S$ can be normalized either within the layer (layer norm) or a batch (batch norm) in this layer. The next layer in an encoder is a feed-forward layer. Negative values in the linear transformed $F$ are truncated as zeros and then linear transformation was conducted again as in Equation 3.16. Dimension of $G$, noted as $d_{ff}$, is determined by the size of weight matrix $W_2$. $G$ and $F$ are inputs for the second add and norm layer as described in Equation 3.17, and $O$ is the final output of this encoder.

$$\begin{aligned} U &= \text{MultiHead}(Q, K, V) \\ &= \text{Linear}(\text{Concat}(head_1, head_2, \ldots, head_m)) \end{aligned} \tag{3.14}$$

$$F = \text{LayerNorm}(U + S) \tag{3.15}$$

$$G = \text{FFN}(F) = \max(0, F \cdot W_1 + b_1) \cdot W_2 + b_2 \tag{3.16}$$

$$O = \text{LayerNprm}(F + G) \tag{3.17}$$

The final linear layer transforms hidden features $O$ to the output and it is different in the unsupervised pretraining and supervised fine-tuning process.

### 3.3.5 Two-step Training

Two-step training, including unsupervised pretraining and supervised fine-tuning, achieved better performance compared to single supervised training in the natural language processing (NLP) field [138]. So, two-step training was applied to the time series forecasting field as well [130] and still outperformed purely supervised training.

In the unsupervised pretraining step, input is partially covered $x$ and output is uncovered $x$. Each feature vector was randomly covered 20% along the time dimension as shown in Figure 3.4. $x_{t-h}$ represents features observed during time $t-h$. Weights and biases of the embedding layer, positional encoding layer, and encoders learned in this step will be saved for the supervised learning step. This process enables the model to learn more useful representative information of training data before the training process.

[130] compared fixed weights and biases obtained in supervised training (Figure 3.4) and unfixed weights and biases, and found the fixed parameters tuned out to be better. Therefore, in this study, the structure in Figure 3.5 was adopted. Input and output of supervised finetune are uncovered $x$ and prediction $y$. This step applies the weights and biases trained in the first training step and the last linear transformation layer will be designed to convert the output to the same dimension as the target.

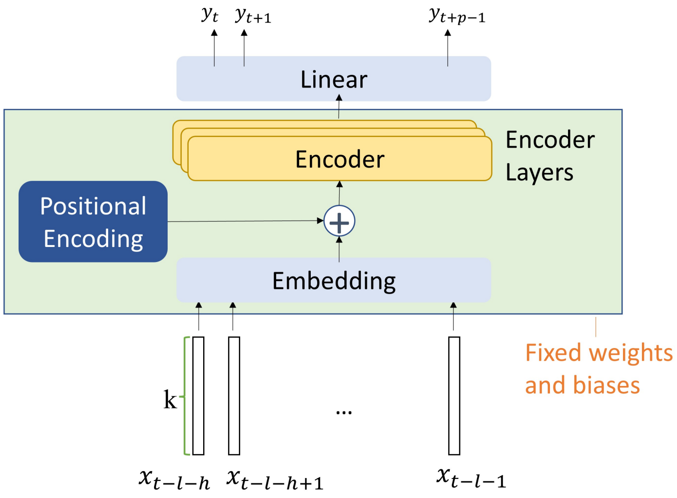


Figure 3.5. Supervised training structure of TDEPF.

The flowchart depicted in Figure 3.6 illustrates the two-step training process. Initially, the multivariate time series undergo transformation via an embedding layer and are encoded using a positional encoding layer. Subsequently, the combined features traverse through the transformer encoder. During the pretraining step, the layers preceding the latent feature matrix are trained. In the finetune step, only the multiple linear perception (MLP) prediction head undergoes updates.

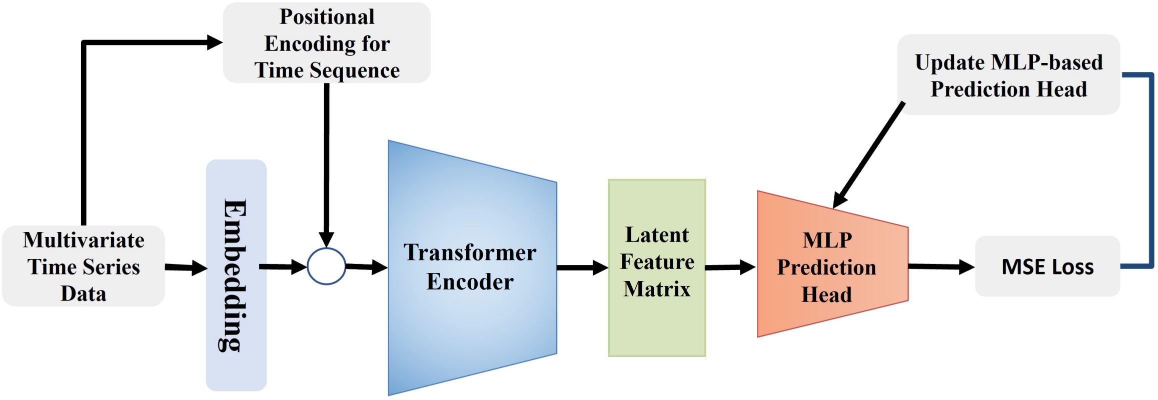

Positional Encoding for Time Sequence
Update MLP-based Prediction Head
Multivariate Time Series Data
Embedding
Transformer Encoder
Latent Feature Matrix
MLP Prediction Head
MSE Loss

Figure 3.6. Flowchart of TDEPF.

### 3.4 Experiments

#### 3.4.1 Data

The dataset employed in this research encompasses various variables including electricity prices, load, day-ahead-market (DAM) prices, and weather information. Specifically, the electricity price and load data were obtained from ERCOT, while the weather data was sourced from the integrated surface dataset available on the National Centers for Environmental Information website. In the case of intraday electricity price forecasting (EPF), the response variable was derived from 5-minute interval locational prices (LMP) in the real-time market. As for day-ahead EPF, the hourly electricity price was determined by averaging the 15-minute interval settlement point prices (SPP) in the real-time market. For instance, the SPP at 1:00 AM was computed as the average value of SPP at 00:00 AM, 00:15 AM, 00:30 AM, and 00:45 AM.

In ERCOT, there exists a diverse range of settlement points, numbering in the hundreds and categorized by type. For the purposes of forecasting, the Settlement Point Price (SPP) within the north load zone (LZ_north) was chosen as a representative region. Each load zone's SPP is determined by the weighted average of electricity prices from resource nodes (RN) located within that particular region, prompting an examination of the impact of including resource nodes' SPP in the analysis. The weather data encompasses temperature and dew points, providing an indication of air humidity. The original load and weather data were recorded on an hourly basis. Notably, while load data was provided in weather zones (see Figure 3.7), the SPP was attributed to the load zone (see Figure 3.8). As depicted in Figure 3.8, the north load zone is partially encompassed by the north-central, north, and east weather zones. Consequently, weather and load information from these three weather zones were considered as influential factors for SPP prediction within the north load zone. For day-ahead forecasting, data spans from 1:00 AM on January 1st, 2017, to 24:00 PM on December 31st, 2021. Any missing values within the time series were estimated using the nearest neighbor interpolation method.

In the modeling phase, the training dataset was defined to encompass data from 2017 to 2020, while the testing dataset consisted of data from the year 2021. Continuous variables within the dataset were standardized using the mean and standard deviation derived from their respective training sets. Categorical attributes like month, day of the year, and weekday were encoded for processing. We explored two different encoding methods: one-hot encoding and sin-cos encoding.

One-hot encoding, a widely utilized technique, transforms categorical features into a format suitable for machine learning models [139]. This approach assigns a binary column to each category of the variable, wherein all entries are set to zero except for those corresponding to samples that fall into that category. Circular encoding,

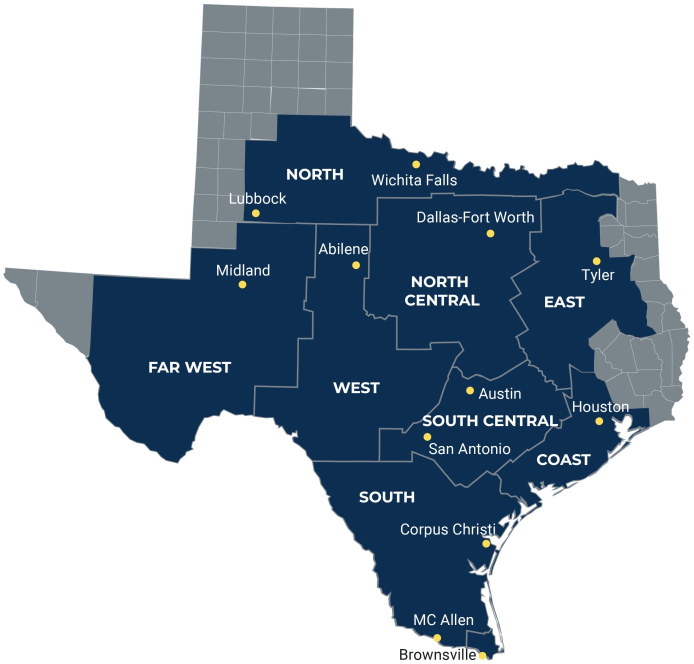


Figure 3.7. Weather zone map of ERCOT [1].

commonly referred to as sin-cos encoding, is a technique well-suited for representing cyclical variables. The sin-cos encoding equations, as depicted in Equations 3.18 and 3.19, involve the use of $x_i$ to denote the ith sample of the feature $x$.

$$\text{sin\_encoding}(x_i) = \sin\left(\frac{2\pi x_i}{\text{max(x)}}\right) \tag{3.18}$$

$$\text{cos\_encoding}(x_i) = \cos\left(\frac{2\pi x_i}{\text{max(x)}}\right) \tag{3.19}$$

### 3.4.2 Experiment Settings

The objective of the prediction task is to utilize the historical 144 hours' worth of features to forecast the day-ahead (24 hours') SPP of the north load zone, with a lead time of 24 hours (as illustrated in Figure 3.1). This means that the input $x \in R^{h \times k}$ and the output $y \in R^p$ where $h$ is 144, $p$ is 24, and k can vary depending on the specific experimental scenario.

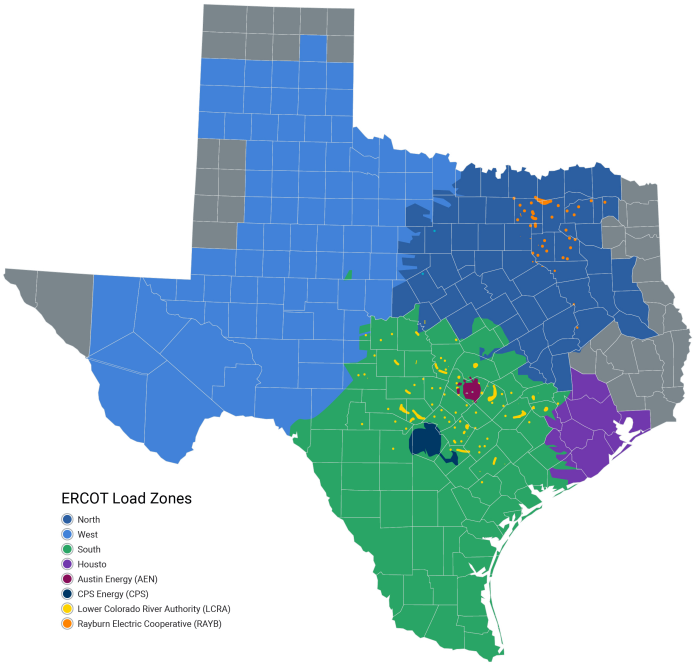

ERCOT Load Zones
North
West
South
Housto
Austin Energy (AEN)
CPS Energy (CPS)
Lower Colorado River Authority (LCRA)
Rayburn Electric Cooperative (RAYB)

Figure 3.8. Load zone map of ERCOT [1].

### 3.4.3 Hyperparameter Optimization

Hyperparameters of each model were optimized by a SOTA method, Bayesian optimization hyperband (BOHB) [140]. Searching space of hyperparameters for the TDEPF model is set as in Table 3.3.

Two common optimization methods, Adam and RAdam, were tested. The dropout rate is the ratio of information discarded after each training iteration and it is normally set to be between 0 to 0.9. When there is only one layer, the dropout rate is supposed to be 0. $d$ is the dimension d of embedded features as mentioned in section 3.3.4. $d_{ff}$ is the dimension of features of outputs of the feed-forward layer. The number of encoders is the number of stacked encoders in the TDEPF model. The number of heads is the parameter $m$ in the multi-head attention. Global regularization can be either applied or not, correspondingly to 'True' or 'False'. Learning rate determines step size that weights change in every iteration. When the hyperparameter

Table 3.1. Dimensions of continuous features

| Continuous Features | Dimensions |
|---|---|
| Temperatures, dewpoints, and wind speed of DFW, Wichita Falls, and Athens | 9 |
| Load of North, East, and Ncent zones | 3 |
| SPP of LZ_north | 1 |
| SPP of resource nodes inside LZ_north | 51 |
| DAM SPP of LZ_North | 1 |
| Total | 65 |

Table 3.2. Dimensions of encoded categorical features

| Features | Dimensions with One-hot | Dimensions with Sin-Cos |
|---|---|---|
| Month ID | 12 | 2 |
| Day of the year | 366 | 2 |
| Weekday | 7 | 2 |
| Hour ID | 24 | 2 |
| Total | 409 | 8 |

global regularization is ‘True’, the weight of l2 regularization will be tuned, otherwise, it will not be optimized. Common activation functions used in Transformer neural network, ReLU and GeLU, were compared. Normalization methods, normalizing per batch (batch norm) or per layer (layer norm) as described in section 3.3.4, were also explored. The optimized hyperparameters for the TDEPF model using the BOHB method are presented in Table 3.4.

The BOHB optimization was also utilized to optimize the hyperparameters of RNN-type models. The search space for each hyperparameter of RNN-type models, including 'num_layers,' which signifies the number of neural network layers, is illustrated in Table 3.5. Optimized hyperparameters of RNN models are listed in Table 3.6.

Table 3.3. Searching space of hyperparameters of TDEPF in the unsupervised step

| Hyperparameters | Bounds |
|---|---|
| Optimizer | {'Adam', 'RAdam'} |
| Drop out rate | [0, 0.9] |
| $d$ | {16,32,64,128} |
| $d_{ff}$ | {16,32,64,128} |
| Number of encoders | {1,2,3,4,5,6} |
| Number of heads | {1,2,4,6,8} |
| Global regularization | {True, False} |
| Learning rate | {0.0001, 0.001,0.01,0.1} |
| Weight of l2_reg | {0,0.0001,0.001,0.01,0.1} |
| Activation | {'ReLU', 'GeLU'} |
| Normalization choice | {'batch_norm','layer_norm'} |

### 3.4.4 Performance Evaluation Metrics

The measurements used in this deterministic forecasting in this study were MAE (mean absolute error), MAPE (mean absolute percentage error), MSE (mean squared error), and RMSE (root of mean squared error). MAE is used to present absolute residuals. MAPE measures relative error compared to original values. Compared to RMSE, MSE is more sensitive to outliers in prediction. Deterministic forecasting is more accurate when the values of these metrics are smaller. MAE, MAPE, MSE, and RMSE are defined in Equations 3.20 to 3.23, where $n$ is the number of observations, $y_t$ is the observation of SPP at time $t$, and $\hat{y}_t$ is the predicted value of SPP at time $t$.

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \tag{3.20}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \tag{3.21}$$

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \tag{3.22}$$

Table 3.4. Optimized hyperparameters of TDEPF transformer model

| Hyperparameters | Values |
|---|---|
| Optimizer | Adam |
| Drop out rate | 0.6 |
| $d$ | 64 |
| $d_{ff}$ | 64 |
| Number of encoders | 2 |
| Number of heads | 4 |
| Global regularization | False |
| Weight of l2_reg | 0 |
| Learning rate | 0.0006 |
| Activation | ReLU |
| Normalization choice | layer_norm |

Table 3.5. Hyperparameter Search Space for RNN-type models

| **Hyperparameters** | **Bounds** |
|---|---|
| batch size | 16, 32, 64, 128 |
| drop out | [0, 0.9] |
| hidden size | 32, 64, 128, 256, 512 |
| learning rate | 0.0001, 0.001, 0.01, 0.1 |
| num_layers | 1, 2, 3 |
| optimizer | 'Adam', 'RAdam' |

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{3.23}$$

### 3.4.5 Effects of Encoding Methods

Initially, the effect of two categorical encoding methods was investigated using the basic transformer model, without an unsupervised pretraining process (Section 3.3.5), which is referred to as one-step training. In one-step training, the dimension of the output from the final layer matches the targets and is $R^{24}$.

Table 3.6. Optimized Parameters of RNN-type models

| **Models** | **LSTM** | **GRU** | **RNN** |
|---|---|---|---|
| batch size | 128 | 128 | 128 |
| drop out | 0.5 | 0.5 | 0.4 |
| hidden size | 256 | 64 | 64 |
| learning rate | 0.0285 | 0.0006 | 0.0004 |
| num_layers | 3 | 1 | 3 |
| optimizer | Radam | Adam | Adam |

A comparison between sin-cos encoding and one-hot encoding was conducted. As indicated in Table 3.7, "Sincos" and "Onehot" in the settings column refer to whether the categorical features listed in Table 3.2 were encoded using sin-cos or one-hot methods. 'ST1' and 'ST2' refer to the spike transformation method ST1 and ST2 as described in section 3.3.3. As shown in Table 3.7, sin-cos encoding and one-hot encoding yield similar accuracy in terms of MSE, RMSE, MAPE, and MAE values. However, the computational time for sin-cos encoding settings is approximately one-third of that for one-hot encoding.

Table 3.7. Assessments of day-ahead electricity price prediction with one-step training evaluated on the original scale of electricity prices

| **Settings** | **MSE** | **RMSE** | **MAPE** | **MAE** | **Computational time (min)** |
|---|---|---|---|---|---|
| Sincos+ST1 | 961176.313 | 980.396 | 89.583 | 137.170 | 7.576 |
| Onehot+ST1 | 960801.250 | 980.205 | 89.044 | 136.344 | 25.570 |
| Sincos+ST2 | 960653.563 | 980.129 | 90.299 | 138.271 | 7.795 |
| Onehot+ST2 | 960073.375 | 979.833 | 89.276 | 136.705 | 25.569 |
| Sincos+no transformation | 962002.750 | 980.817 | 104.825 | 160.512 | 7.606 |
| Onehot no transformation | 962548.625 | 981.096 | 107.039 | 163.903 | 25.227 |

Upon reviewing the results presented in 3.7, it was noted that spike transformation methods ST1 and ST2 exhibit no substantial difference when all the predictions were transformed back to the original scale. But it is noticeable that forecasts displayed overall low accuracy. This is evidenced by the fact that the lowest MAPE remains close to 90, indicating that the error proportion relative to actual values is approximately 90%. The similarity in the performances of the two spike transformations could be attributed to the presence of extreme SPP values in the testing set, leading to errors being predominantly influenced by spike predictions. The prediction performances of data without spike transformation are captured in the last two rows of Table 3.7. Despite the generally low accuracy, the performances of experiments with spike transformation remain significantly superior to those with untransformed data.

Upon analyzing the results presented in Table 3.7, it can be deduced that sin-cos encoding outperforms one-hot encoding, taking into account the trade-off between accuracy and computational cost. Furthermore, spike transformation enhances prediction accuracy, although the determination of an optimal upper bound is still pending experimentation.

### 3.4.6 Comparison of Spike Tramsformation Methods

On inspecting the original SPP data, it was observed that exceptionally high spikes resulted from the Texas snowstorm in February 2021. As a result, we omitted the test data for the timeframe spanning from February 11 to February 19, 2021, during which there was an unusual surge in electricity prices. We subsequently reevaluated the metrics across various SPP data ranges, as illustrated in Table 3.8.

In Table 3.8, the performance of models trained using spike transformation ST1 and ST2 is depicted in the last two rows, with ST1 exhibiting superior performance across all four metrics. To ensure a fair comparison between the two spike transformations, predictions were inversely transformed to the original scale and assessed,

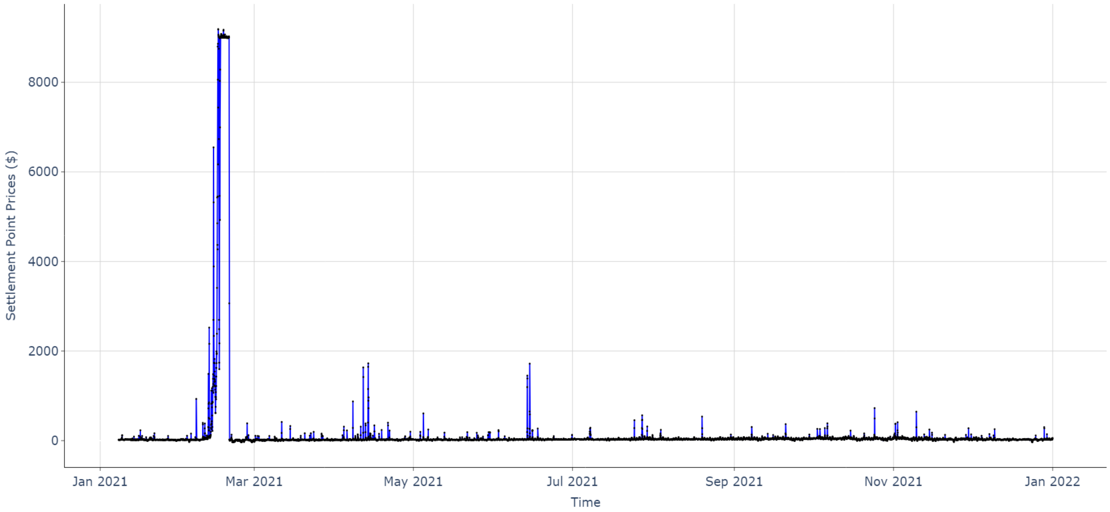


Figure 3.9. Settlement point prices of the north load zone from 01/08 to 12/31 in 2021 in the ERCOT.

as indicated in the first two rows. ST1 proves to be slightly more effective than ST2. Nevertheless, when assessing the prediction of SPP below 36.761$/MWh, a value that surpasses the majority of the target values based on the statistical definition of IQR outlier, ST1 demonstrates a clear superiority over ST2. Additionally, a spike transformation method labeled 'Spike Truncation', replacing original SPPs over 36.761$/MWh as 36.761$/MWh, was tested and results were included in the table. From the metrics in the table, it can be observed that simple truncation at spikes and replacement with the upper-bound value is not as effective as ST1.

### 3.4.7 One-step vs Two-step Training

As explained in Section 3.3.5, during the pretraining phase of the model, randomly selected 20% of each feature is masked, and the model is designed to reconstruct the original inputs. The parameters of the encoder layers are fixed after this stage. This pretraining step is advantageous as it enhances convergence during the

Table 3.8. Assessments of two spike transformation methods with one-step training and sin-cos encoding evaluated on different data ranges after removing abnormal days from 2/11/2021 to 2/19/2021

| **Data** | **Spike Transformation** | **MSE** | **RMSE** | **MAPE** | **MAE** |
|---|---|---|---|---|---|
| Original scale SPP | ST1 | 4293.620 | 65.526 | 54.798 | 19.213 |
| Original scale SPP | ST2 | 4301.299 | 65.584 | 58.153 | 20.390 |
| Original scale SPP under 36.761 $/MWh | ST1 | 103.289 | 10.163 | 34.605 | 7.511 |
| Original scale SPP under 36.761 $/MWh | ST2 | 169.016 | 13.001 | 45.627 | 9.903 |
| Original scale SPP under 36.761 $/MWh | Spike Truncation | 133.012 | 11.533 | 38.512 | 8.124 |
| SPP transformed with ST1 | ST1 | 198.989 | 14.106 | 40.889 | 10.963 |
| SPP transformed with ST2 | ST2 | 793.694 | 28.173 | 53.887 | 17.146 |

fine-tuning phase and facilitates the model in capturing more robust and generalized features. The outcomes presented in Table 3.9 indicate that the two-step training process contributes to an improvement in predictions across all four metrics. Specifically, the MAPE reduces from 40.889 to 27.376.

Table 3.9. One-step vs Two-step Training

| **Training Setting** | **MSE** | **RMSE** | **MAPE** | **MAE** |
|---|---|---|---|---|
| One-step | 198.989 | 14.106 | 40.889 | 10.963 |
| Two-step | 103.158 | 10.157 | 27.376 | 7.340 |

### 3.4.8 Inclusion of Resource Nodes

While the SPP of the north load zone is the weighted average of the SPP of resource nodes inside the load zone, the effect of including the SPP of resource nodes as inputs was examined as well. This test was done in order to find out if there was a maintenance or shutdown of a generation, the SPP spike that happened on the corresponding resource node could give a hint to the potential spike of the load zone. As presented in Table 3.10, the inclusion of either original SPP of resources nodes (”RN included”) or respectively transformed by ST1 were tested. It seems that the inclusion of resource nodes does not improve the performance of prediction in terms of the metrics listed in the table. By checking the occurrence time of spikes of resource nodes, it can be found that the occurrence time of spikes of resource nodes and load zone is almost the same. So, adding resource nodes in the feature sets does not enhance the accuracy of prediction. The predictive factors did not include the SPPs of resource nodes within the northern load zone.

Table 3.10. Inclusion VS Exclusion of Resource Nodes

| **RN** | **MSE** | **RMSE** | **MAPE** | **MAE** |
|---|---|---|---|---|
| No RN | 103.158 | 10.157 | 27.376 | 7.340 |
| RN included | 119.470 | 10.930 | 29.331 | 7.864 |
| Transformed RN included | 163.019 | 12.768 | 34.295 | 9.195 |

### 3.4.9 Comparisons of Time Lags

In the preceding experiments, the default lag time was set at 7 days, encompassing one day as lead time, resulting in $h$ being 144 hours. To investigate the impact of incorporating more extended historical information as inputs, experiments were conducted with 14 and 21 lag days, also including one day as lead time, leading to $h$ values of 312 and 480, respectively. As indicated in Table 3.11, without resource nodes, the accuracy of training with 7, 14, and 21 days as time lags is similar. Additionally, the augmentation of the time dimension results in prolonged training time and increased computational cost, without yielding any noticeable advantage in incorporating longer lags.

Table 3.11. Comparisons of Time Lags for Day-Ahead Electricity Price Forecasting

| **Time Lags (days)** | **MSE** | **RMSE** | **MAPE** | **MAE** |
|---|---|---|---|---|
| 7 | 103.158 | 10.157 | 27.376 | 7.340 |
| 14 | 107.713 | 10.378 | 27.559 | 7.412 |
| 21 | 112.254 | 10.595 | 28.416 | 7.669 |

### 3.4.10 Comparisons with RNN-type Models

After identifying the optimal configurations for day-ahead electricity price forecasting in the preceding experiments, a comparative analysis was conducted with RNN-type models, specifically RNN, GRU, and LSTM. The predictions of each model underwent evaluation, and the outcomes are presented in Table 3.12. Across all four metrics, whether assessed in terms of the transformed SPP or the original scale of SPP, the TDEPF model outperformed the other three models. Notably, the accuracies of the four models exhibit minimal differences when assessed on the transformed SPP. However, upon evaluation on the reverse-transformed SPP (the original scale), the MAPE of the TDEPF model is significantly smaller than that of the other three models. This suggests that the TDEPF model excels in capturing spikes compared to the alternative models.

Table 3.12. Comparisons of Proposed TDEPF and RNN-type Models

| **Evaluation Scale** | **Method** | **MSE** | **RMSE** | **MAPE** | **MAE** |
|---|---|---|---|---|---|
| Original SPP | GRU | 4157.748 | 64.481 | 72.013 | 18.057 |
| | LSTM | 4092.868 | 63.976 | 71.234 | 17.129 |
| | RNN | 4168.973 | 64.568 | 76.315 | 18.160 |
| | **TDEPF** | **4026.019** | **63.451** | **44.293** | **15.530** |
| Transformed SPP | GRU | 147.711 | 12.154 | 39.314 | 9.106 |
| | LSTM | 130.403 | 11.419 | 37.138 | 8.600 |
| | RNN | 156.684 | 12.517 | 41.793 | 9.470 |
| | **TDEPF** | **103.158** | **10.157** | **27.376** | **7.340** |

Figure 3.10, 3.11, 3.12, and 3.13 illustrate the predictions of four models against the original scale of the targets. The RNN-type models exhibit varying levels of overfitting, noticeable in their predictions. In contrast, TDEPF produces smoother predictions that demonstrate resilience against spikes. The TDEPF model effectively captures cyclic patterns within the time series, evident in its close alignment with the target values. Importantly, while predicting the precise values of spikes may be challenging, forecasting the timing of their occurrences remains feasible.

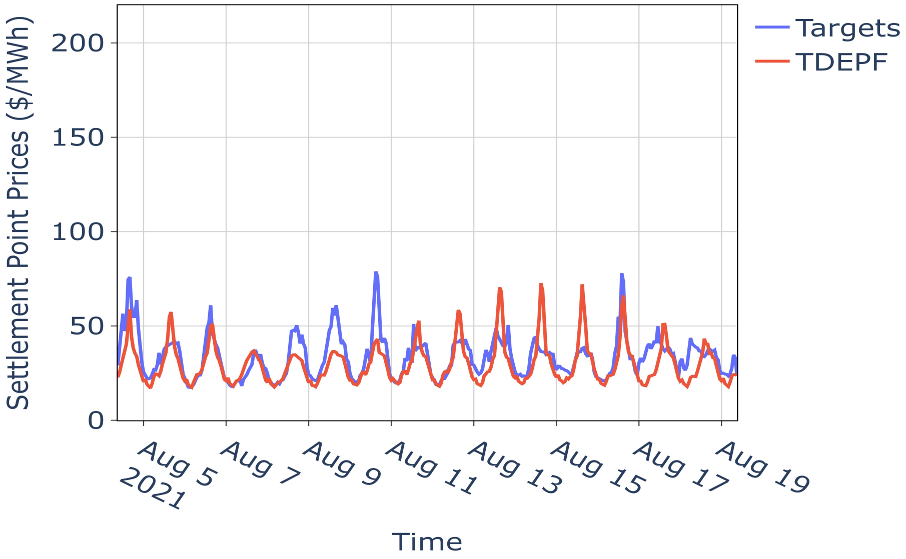


Figure 3.10. Settlement Point Prices Day-ahead Prediction by TDEPF.

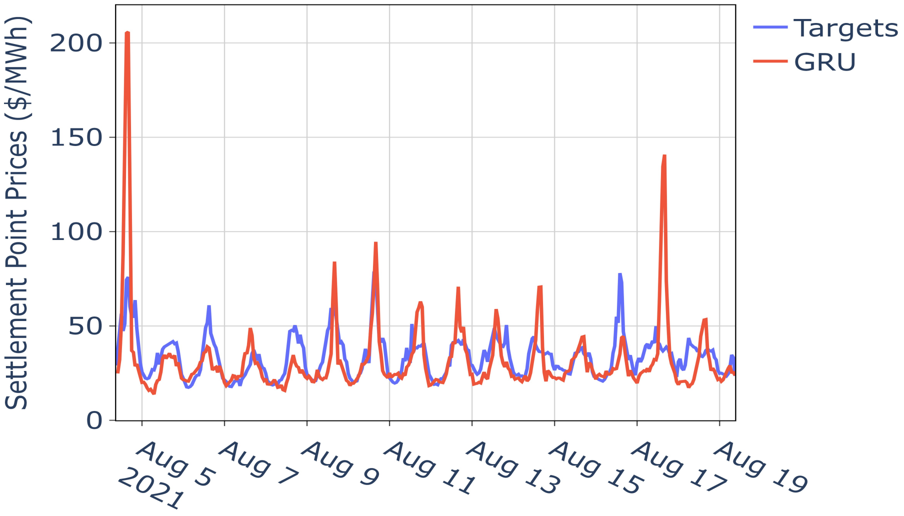


Figure 3.11. Settlement Point Prices Day-ahead Prediction by GRU.

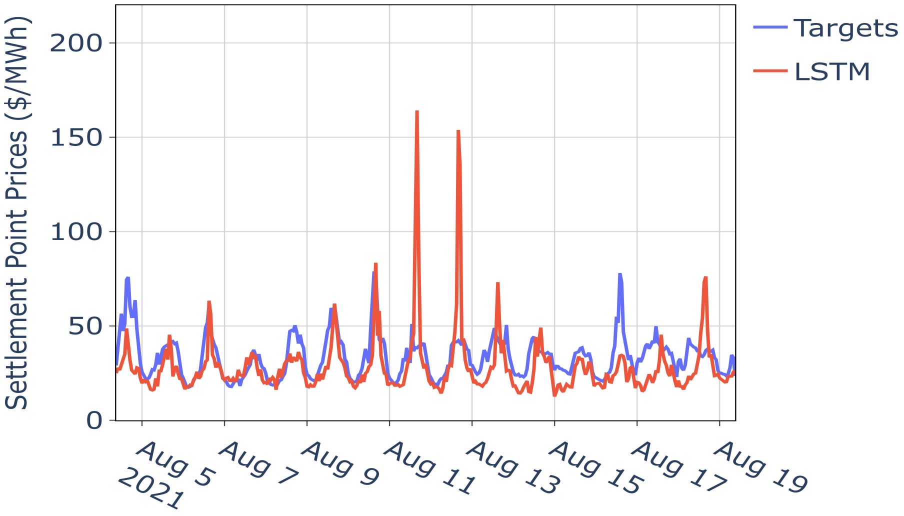


Figure 3.12. Settlement Point Prices Day-ahead Prediction by LSTM.

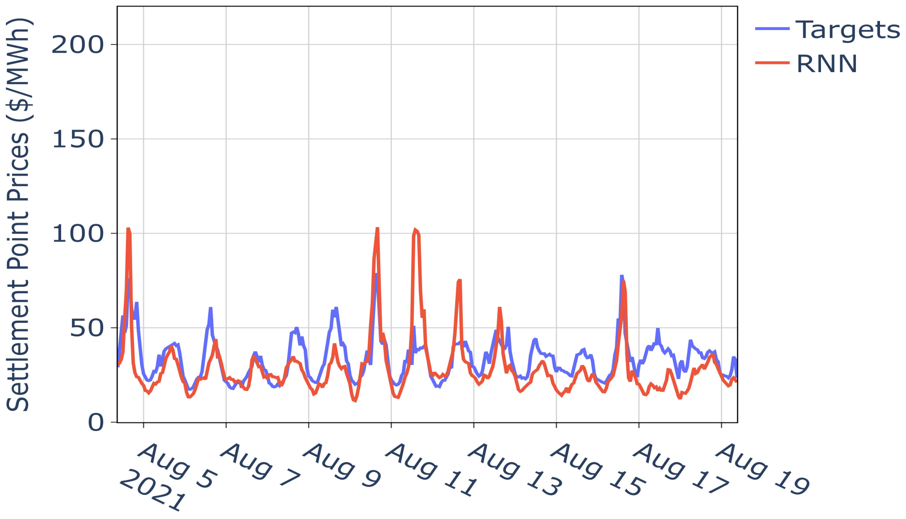


Figure 3.13. Settlement Point Prices Day-ahead Prediction by RNN.

## 3.5 Conclusions

In this chapter, the challenges associated with sudden spikes in electricity prices and the intricacies of modeling multivariate time series data were tackled. To address spikes in time series data, a new technique called ST1 was introduced, providing a novel spike transformation method. This method proved effective in directing the model's attention toward the accurate prediction of normal prices.

A transformer model with a two-step training process, the TDEPF model, was applied, designed to effectively handle the learning and forecasting of patterns in multivariate time series data. The inclusion of a pretraining step enhances model convergence and mitigates overfitting concerns.

The experiments included a comparison between Sin-cos encoding and one-hot encoding for representing time information. Sin-cos encoding demonstrated comparable performance to one-hot encoding but with fewer dimensions, resulting in reduced computational time compared to one-hot encoding.

The case study, utilizing ERCOT market data, demonstrated that the TDEPF model surpassed the performance of other widely used models such as LSTM, RNN, and GRU in the task of day-ahead electricity price forecasting. The experimental findings highlighted the effectiveness of the proposed spike transformation approach in mitigating spikes in electricity prices. Moreover, the model exhibited proficiency in capturing predictive patterns.

The proposed forecasting model bears significant potential to assist market participants in making more informed decisions regarding bidding on day-ahead electricity prices. Its capabilities contribute to fostering a more efficient and stable deregulated market.

# CHAPTER 4

# PROBABILISTIC DAY-AHEAD ELECTRICITY PRICE FORECASTING

## 4.1 Introduction

As discussed in the previous chapters, research into electricity price forecasting (EPF) can bring immense value to electricity markets. While there are tremendous benefits in electricity markets, the growing volatility of electricity prices puts market participants at higher risks. For one thing, the rise of renewable energy, such as wind and solar energy, leads to more dynamic electric power generation which further causes volatility in electricity prices. For another, the United States started deregulation in the 1990s, and deregulated electricity markets are naturally more competitive than regulated markets. The Electric Reliability Council of Texas (ERCOT) market, being one of the deregulated markets and gradually influenced more by changes in wind power than by electricity demand [2], was studied in this chapter. To assist market participants in hedging against fluctuations in electricity prices and make more knowledgeable decisions, probabilistic day-ahead EPF models were created for the wholesale power market run by ERCOT in this study.

Compared to deterministic EPF models, probabilistic EPF models developed later but gradually became prevailing [13, 14]. Unlike deterministic forecasting methods which just provide predictions of point values, probabilistic forecasting models provide prediction intervals (PIs) that offer more information and better assist decision-making. As summarized in 2.4, probabilistic modeling methods include Error density forecasts, bootstrapped prediction intervals, quantile regression, Bayesian approach. However, most of the probabilistic methods are based on certain distribution assumptions or preliminary knowledge of the parameter distributions. For the error density forecasting methods, complexity is increased due to modeling the error term and point values separately and the assumption of the error distribution depends on researchers' empirical knowledge and statistical analysis. Bootstrapped PIs repeatedly re-samples from the original data set to get the distribution of the variable. It is especially useful for small data sets when not enough data can be used for modeling, but computational cost increases with repetition number. Quantile regression is free of distribution assumptions and flexible to be combined with deep learning methods. In this study, a method combining quantile regression and the transformer neural network, QR-Transformer, was proposed. In addition, the widely used method, the Gaussian Process, is applied to estimate the prediction intervals based on the deterministic forecasting of electricity prices, the method is named GP-Transformer. Furthermore, composite quantile regression, which is designed to promote the efficiency of quantile regression, was combined with a transformer neural network as well. The comparisons of three models will be discussed in this study.

Two modifications of the pinball loss function were explored to achieve different prediction outcomes, aside from the model structure. The first modification is the Mean Squared Error (MSE) regularized pinball loss, and the second is the winkler-peak score, a metric proposed for measuring the accuracy of peak time prediction, regularized pinball loss. The MSE regularized pinball loss was formulated to impose constraints on the extreme quantiles which are close to 0% or 100%. The winkler-peak score regularized pinball loss was formulated to direct the model's focus toward peak time prediction. Discussion of these regularization methods will be presented in Section 4.4 as well.

### 4.1.1 Contributions

In this chapter, contributions are made to the following aspects.

1. The integration of Gaussian Process and Transformer Model (GP-Transformer) was suggested for evaluating prediction uncertainties.
2. A transformer-based quantile regression model (QR-Transformer) and a transformer-based composite quantile regression (TCQR) were formulated to offer probabilistic forecasting.
3. The Winkler-peak score, a novel performance metric for assessing peak time predictions of electricity prices, was proposed.
4. MSE regularized pinball loss was proposed to compel the convergence of quantiles near 0 and 1 boundaries.
5. A Winkler-peak score penalty was incorporated into the pinball loss to ensure that the TCQR model focuses on the timing of peaks in time series.

## 4.2 Problem Formulation

This study aims to find a model to predict the conditional distribution of next $p$ (from $t$ to $t+p-1$) hours' real-time settlement point prices (SPP) with $k$ features observed during previous $h$ (from $t-l-h$ to $t-l-1$) hours. Let $y_t$ represent the target SPP and $x_t$ represent $k$-dimension features observed at time t. Deterministic p-step ahead EPF problem can be expressed as $y_t, y_{t+1}, \ldots, y_{t+p-1} = f(x_{t-l-h}, x_{t-l-h+1}, \ldots, x_{t-l-1}) + \epsilon_t$ where $\epsilon_t$ is the noise term, p is the length of prediction horizon, and h is the length of historical time steps. Probabilistic forecasting is to find the conditional distribution of $P(y_t, y_{t+1}, \ldots, y_{t+p-1} \mid x_{t-l-h}, x_{t-l-h+1}, \ldots, x_{t-l-1})$ instead of $f(y_t, y_{t+1}, \ldots, y_{t+p-1})$. As shown in Figure 4.1, the targets are $m$ quantiles of the conditional probability over 24 hours. $Q_{q_1}$ to $Q_{q_m}$ represents $m$ quantiles, and

$X$ denotes the inputs of the model. The prediction interval band, as shown in Figure 4.2, is formed by these quantiles.

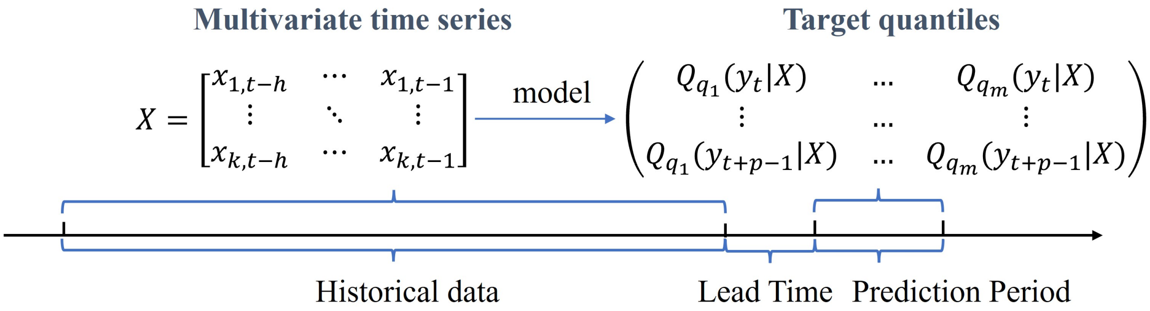

Multivariate time series
Target quantiles
$X = \begin{bmatrix} x_{1,t-h} & \cdots & x_{1,t-1} \\ \vdots & \ddots & \vdots \\ x_{k,t-h} & \cdots & x_{k,t-1} \end{bmatrix} \xrightarrow{\text{model}} \begin{pmatrix} Q_{q_1}(y_t|X) & \cdots & Q_{q_m}(y_t|X) \\ \vdots & \cdots & \vdots \\ Q_{q_1}(y_{t+p-1}|X) & \cdots & Q_{q_m}(y_{t+p-1}|X) \end{pmatrix}$
Historical data
Lead Time
Prediction Period

Figure 4.1. Definition of Probabilistic Forecasting.

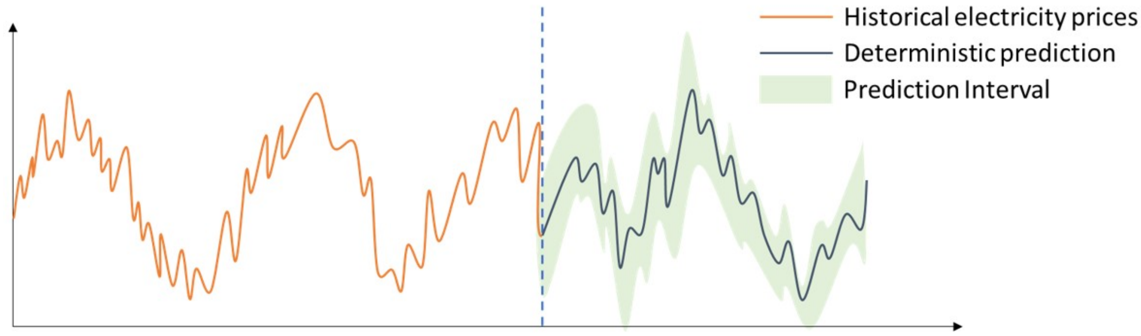


Figure 4.2. Probabilistic vs Deterministic Forecasting.

## 4.3 Related Works

A comprehensive examination of probabilistic forecasting methods for electricity price forecasting was conducted by [14] and [119]. It was observed, as reported in [14], that in the influential Global Energy Forecasting Competition (GEFCom2014), three out of the four winners employed quantile regression (QR). QR is proved to be a robust probabilistic regression method and less sensitive to outliers [141]. It is widely applied in economic and energy fields [142, 143, 144, 145, 146]. QR sets up a regression function for a certain quantile of the target variable distribution instead of a conditional mean of the target variable. Correspondingly, the parameters of QR are estimated by the pinball loss function which compares the actual value at a certain quantile of distribution and the predicted value at the same quantile of the distribution. What's more, QR is independent of distribution assumption, so it is flexible to model different data sets.

Bayesian approach is also widely used in probabilistic forecasting, and it is often combined with deep learning methods to provide probabilistic forecasts [78, 79, 80]. Bayesian approaches are classified as parametric and nonparametric methods corresponding to finite parameters and infinite parameters[147]. With the increase in data size and complexity of time series, it is harder to define a parametric probabilistic model. Bayesian nonparametric methods [148] are data-driven and the number of parameters increases with the data size, which sets practitioners free from model selection and parameter design. Gaussian process (GP) regression is one of the well-known Bayesian nonparametric models[94]. So, GP-based probabilistic forecasting was implemented for the comparison in this work.

The state-of-the-art (SOTA) deep learning structure, the transformer, was chosen as the foundational regression model for probabilistic forecasting. Traditional statistical methods such as VAR, VARMA, and GARCH are incapable of handling intricate relationships in multivariate time series. Conversely, common deep learning models for multivariate time series (MTS), including RNN, LSTM, and GRU, encounter issues of gradient vanishing or explosion as the forecasting sequence lengthens. The transformer computes weights in both time and feature dimensions, making it an excellent choice for MTS modeling. In this study, the transformer was combined independently with GP and QR, and their performance was compared with other RNN-type models in probabilistic MTS forecasting.

To effectively determine the parameters of various quantile regression models, [149] introduced the concept of composite quantile regression (CQR), allowing for the simultaneous estimation of parameters across multiple quantile regression models. Building upon this, [150] introduced the composite quantile regression neural network (CQRNN), a hybrid model merging CQR with a neural network. The CQRNN model proves to be adaptable and proficient in uncovering potential nonlinear associations among variables. Consequently, in this investigation, we delved into the combination of composite quantile regression with a transformer neural network, named transformer-based composite quantile regression (TCQR) as well.

When integrated with deep learning models, the quantile regression (QR) faces a challenge known as the quantile cross-over problem, as underscored by [151] and [152]. Additionally, the need to tailor quantile regression to focus on specific subsets of interest is emphasized by [153]. In this investigation, the utilization of Mean Squared Error (MSE) regularized pinball loss is explored to promote the convergence of quantiles close to 0 and 1. Furthermore, the Winkler-peak score is introduced as a metric for assessing the prediction of peak occurrence time. To enhance the QR-based model's ability to predict peak occurrence time, an exploration of Winkler-peak score penalized pinball loss is also conducted.

## 4.4 Proposed Methods

To generate probability predictions instead of deterministic values, we constructed two models: a Gaussian Process-based transformer and a quantile regression-based transformer. Prior to modeling, the SPP underwent preprocessing through a spike transformation, as detailed in Equation 3.2. The transformed SPP, along with other features (which will be discussed in Section 4), constitute the inputs for our models. The spike transformation is defined using the upper bound determined in Chapter 3, with spikes identified as values exceeding \$36.761/MWh, as established in Chapter **??**. The foundational structure of the two-step training process has been introduced in Chapter 3. The focus of this section will be on elucidating the modifications implemented in the model to enable probabilistic forecasting.

To achieve the probabilities of prediction instead of deterministic values, two models, a Gaussian Process-based transformer and a quantile regression-based transformer were built. Before modeling, SPP was preprocessed by a spike transformation. Then the transformed SPP and other features which will be mentioned in section 4 make up inputs for our models. The transformation of SPP is presented as Equation 3.2. $ub$ is the upper bound decided in Chapter 3. After trials in chapter **??**, spikes were defined as values exceeding 36.761 \$/MWh. As the basic structure of two-step training has already been introduced in the Chapter 3, we will focus on introducing the modification of the model for producing probabilistic forecasting in this section.

### 4.4.1 Base Model Structure

The core structure of the model remains in line with the description provided in Section 3.3. The subsequent experiments adopt a two-step training methodology. In these trials, the probabilistic model leverages the identical pre-trained model employed in deterministic forecasting. Modifications are introduced to the fine-tuned model structures to accommodate probabilistic forecasting. Two distinct structures for probabilistic forecasting are explored: one involving a Gaussian process transformer (GP_Transformer) and the other utilizing a Quantile regression transformer (QR_Transformer).

### 4.4.2 Gaussian Process-based Transformer Model

To enable the transformer-based model to furnish the distribution for each prediction, the Gaussian Process (GP) was applied to quantify the uncertainties of the point value predictions generated by the TDEPF model in Section 3.3. The structure of GP-Transformer prediction is illustrated in Figure 4.3, where predictions of training predictions $\hat{y}_{train}$, training targets $y_{train}$ and testing predictions $\hat{y}_{test}$ are the inputs for GP model. To derive the conditional multivariate normal distribution of $p(y_{test}|y_{train}, \hat{y}_{train}, \hat{y}_{test})$, the mean and variance of the probability function can be computed using Equations 4.1 and 4.2, where $\hat{y}_{test}$ denotes testing inputs, $\hat{y}_{train}$ represents training inputs, and the error of the model is assumed to adhere to a Gaussian distribution $\epsilon \sim N[0, \sigma_{obs}^2]$. Here, $y_{train}$ and $\mu$ denote targets and mean targets in the training set, respectively. $K_{\hat{y}_{train}\hat{y}_{train}}$ represents the kernel matrix of inputs $\hat{y}_{train}$ and $\hat{y}_{train}$, $k(\hat{y}_{test}, \hat{y}_{test})^T$ denotes the transposed kernel matrix of $\hat{y}_{test}$ and $\hat{y}_{test}$. Additionally, $I$ denotes the identity matrix.

$$\mu^* = \mu(\hat{y}_{test}) + k^T_{\hat{y}_{train}\hat{y}_{test}}(K_{\hat{y}_{train}\hat{y}_{train}} + \sigma_{obs}^2 I)^{-1}(y_{train} - \mu(y_{train})) \tag{4.1}$$

$$\begin{aligned} Var^* =& k(\hat{y}_{test}, \hat{y}_{test}) + \sigma_{obs}^2 \\ & - k^T_{\hat{y}_{train}\hat{y}_{test}}(K_{\hat{y}_{train}\hat{y}_{train}} + \sigma_{obs}^2 I)^{-1} k_{\hat{y}_{train}\hat{y}_{test}} \end{aligned} \tag{4.2}$$

GP-Transformer structure is shown in Figure 4.4. The outputs from TDEPF model will be the inputs for GP model and the parameters of GP are estimated by marginal log likelihood (MLL) loss.

### 4.4.3 Quantile Regression-based Transformer Model

The pinball loss, also referred to as the quantile loss, is a commonly used loss function in quantile regression. It is utilized for training machine learning models like linear regression or gradient boosting to estimate different quantiles of a target vari-

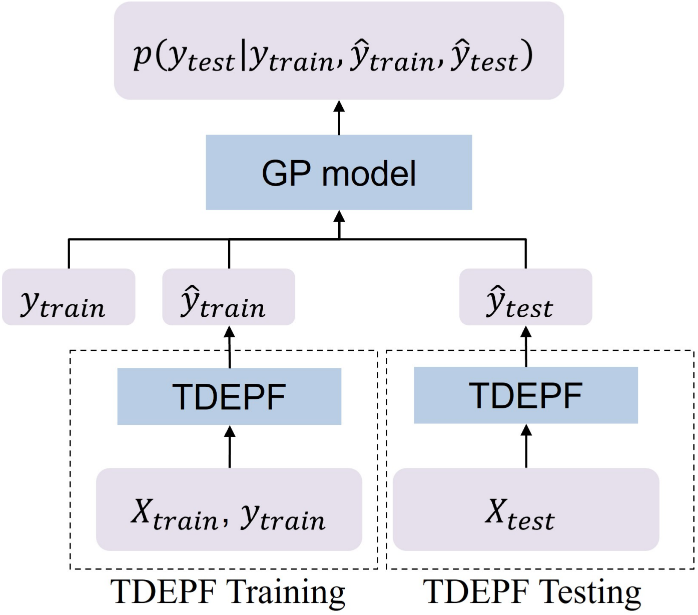


Figure 4.3. Gaussian process-based transformer (GP-Transformer) forecasting model.

able's distribution. Quantiles are statistical metrics that partition a data distribution into equally sized sections, offering valuable insights into the distribution's spread and central tendency. In the context of quantile regression (QR), which is illustrated in Figure 4.5, the same transformer structure as depicted in Figure 1 is employed. However, the objective is no longer centered on minimizing the mean squared error (MSE) of predictions; instead, it aims to minimize the Pinball loss, as defined in Equation 4.3.

$$L_q(\theta) = \begin{cases} q(y_i - f(\theta, x_i)) & \text{if } y_i \geq f(\theta, x_i) \\ (1-q)(f(\theta, x_i) - y_i) & \text{if } y_i < f(\theta, x_i) \end{cases} \tag{4.3}$$

In this equation, $\theta$ represents the model parameters, $y_i$ is the actual observed target value, and $f(\theta, x_i)$ is the predicted value by the model for input $x_i$. The parameter $q$ is a quantile level between 0 and 1 that determines the quantile of interest. The equation consists of two cases: (1) When $y_i$ is greater than or equal to the predicted

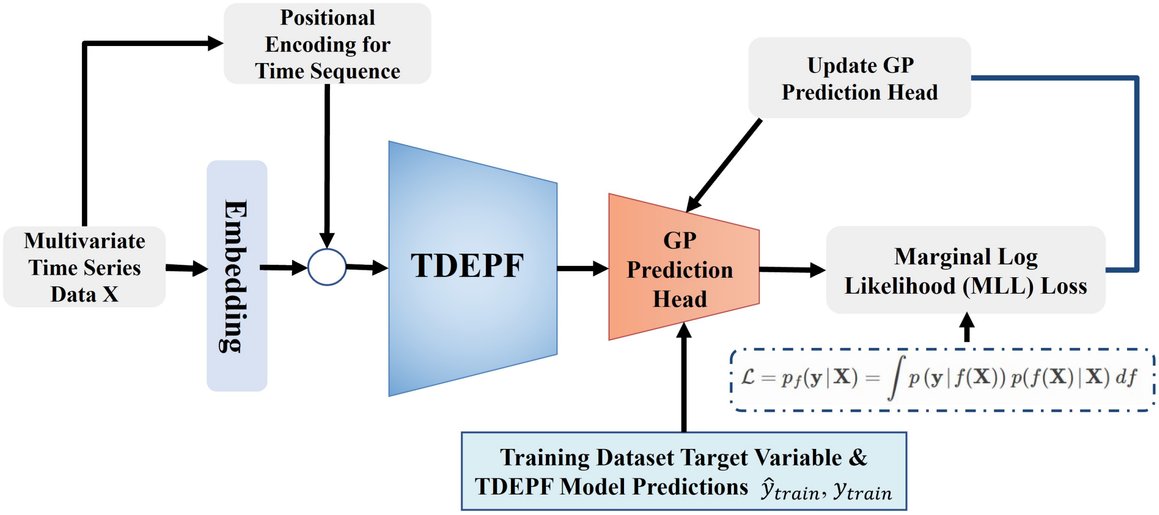

Positional Encoding for Time Sequence
Update GP Prediction Head
Multivariate Time Series Data X
Embedding
TDEPF
GP Prediction Head
Marginal Log Likelihood (MLL) Loss
$\mathcal{L} = p_f(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|f(\mathbf{X}))\,p(f(\mathbf{X})|\mathbf{X})\,df$
Training Dataset Target Variable & TDEPF Model Predictions $\hat{y}_{train}, y_{train}$

Figure 4.4. GP-Transformer Structure.

value $f(\theta, x_i)$, the loss is $q$ times the positive difference between $y_i$ and $f(\theta, x_i)$. (2) When $y_i$ is less than the predicted value $f(\theta, x_i)$, the loss is $(1 - q)$ times the positive difference between $y_i$ and $f(\theta, x_i)$.

This loss function is used in quantile regression to optimize models that estimate different quantiles of the conditional distribution of the target variable. The choice of $q$ determines the specific quantile being estimated, with smaller values of $q$ corresponding to lower quantiles (e.g., median for $q = 0.5$), and larger values corresponding to higher quantiles. Figure 4.5 shows the structure of QR-Transformer, where $100 \cdot (1 - \alpha)\%$ quantile prediction at time $t$ are denoted as $Q_{1-\alpha}(t)$. Here, $q = 1 - \alpha$.

### 4.4.4 Transformer-based Composite Quantile Regression Model

Transformer-based Composite Quantile Regression (TCQR) is a statistical method that enables the concurrent estimation of various quantiles, allowing for a more comprehensive exploration of the impact of predictors on distinct segments of the con-

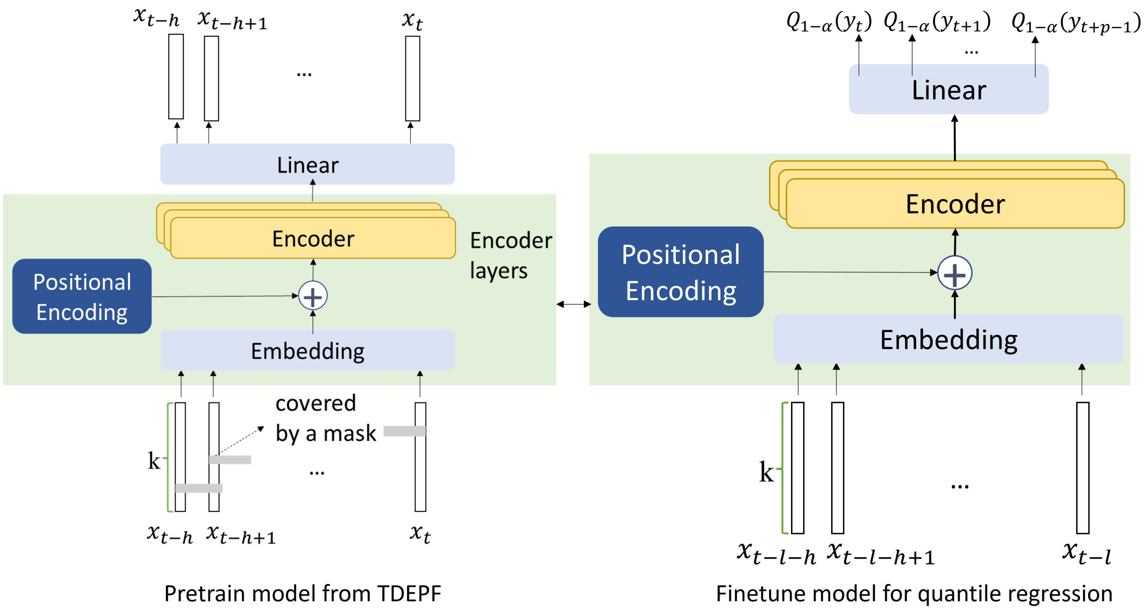


Figure 4.5. Quantile regression transformer (QR-Transformer) forecasting model.

ditional distribution. This simultaneous estimation feature contributes to a holistic understanding of how variables influence different parts of the distribution. One notable advantage of CQR is its robustness to outliers, surpassing mean-based regression models in this aspect. The incorporation of multiple quantiles in the estimation process enhances the model's resilience, providing a more thorough perspective on the relationship between variables, especially when faced with extreme observations. A composite quantile regression transformer framework (Composite QR-Transformer) was devised based on the existing transformer structure. The structure, depicted in Figure 4.6, features a final layer comprising several linear output layers identical to those in the linear layer of Figure 4.5. The loss function of CQR-Transformer, which predicts quantiles at intervals of 10% from 10% to 90%, is provided in Equation 4.4. The structure of the composite quantile regression is demonstrated in Fig-

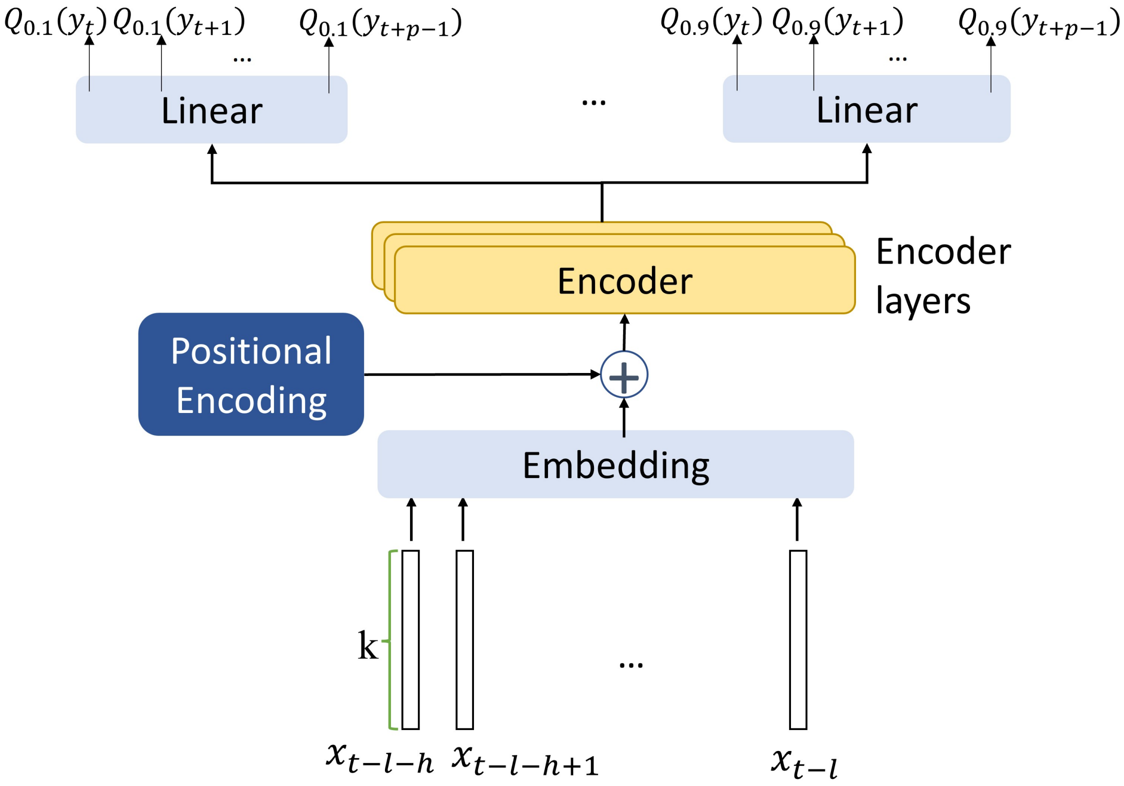

$Q_{0.1}(y_t)$ $Q_{0.1}(y_{t+1})$ ... $Q_{0.1}(y_{t+p-1})$
$Q_{0.9}(y_t)$ $Q_{0.9}(y_{t+1})$ ... $Q_{0.9}(y_{t+p-1})$
Linear
...
Linear
Encoder
Encoder layers
Positional Encoding
Embedding
k
...
$x_{t-l-h}$
$x_{t-l-h+1}$
$x_{t-l}$

Figure 4.6. Composite quantile regression transformer (CQR-Transformer) forecasting model.

ure 4.7, where only the parameters of the multilayer Perceptron (MLP) are updated according to the composite pinball loss function as shown in Equation 4.4.

$$L_{mean} = \frac{1}{M}\sum_{i=1}^{M} L_{q_i}(\theta) \ , \ q_i \in \{0.1, 0.2, 0.3, ..., 0.9\} \text{ and } M = 9 \tag{4.4}$$

The function $L_{mean}$ is defined as the mean of quantile loss functions $L_{q_i}(\theta)$ for a set of quantiles $q_i$ ranging from 10 to 90, where $M$ represents the number of quantiles (in this case, $M = 9$). The overall goal of this function is to compute the average quantile loss over the specified quantiles for a given parameter set $\theta$.

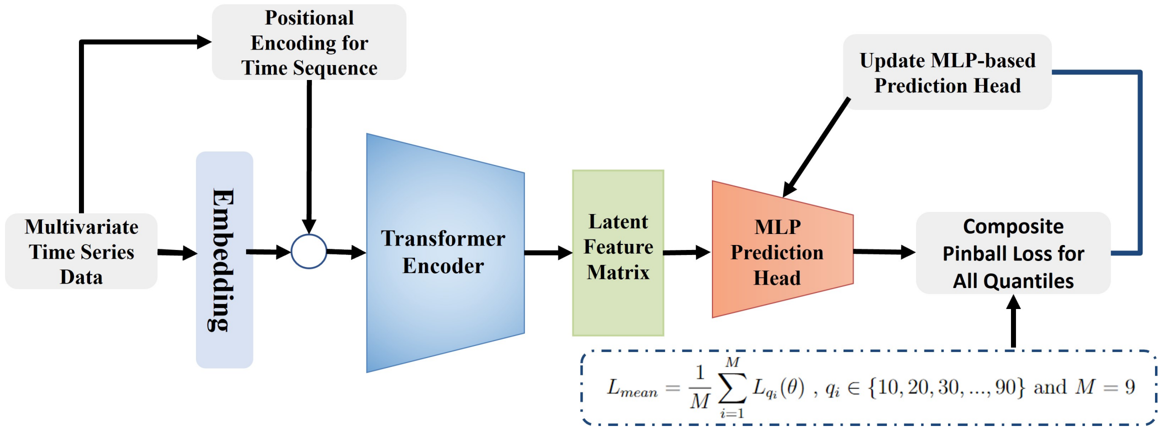

Positional Encoding for Time Sequence
Update MLP-based Prediction Head
Multivariate Time Series Data
Embedding
Transformer Encoder
Latent Feature Matrix
MLP Prediction Head
Composite Pinball Loss for All Quantiles
$L_{mean} = \frac{1}{M}\sum_{i=1}^{M} L_{q_i}(\theta)$ , $q_i \in \{10, 20, 30, ..., 90\}$ and $M = 9$

Figure 4.7. Composite QR-Transformer Structure.

### 4.4.5 Pinball Loss Regularized By MSE

As can be observed in Equation 4.3, when the quantile parameter $q$ is 0 or 1, the prediction is only one-side bounded. For example, when $q = 0$, as long as $f(\theta, x_i)$ is less than $y_i$, the loss is the constant 0, no matter how small the prediction $f(\theta, x_i)$ is. On the opposite, when $q = 1$, $f(\theta, x_i)$ could be extremely large since the loss is 0 as long as $f(\theta, x_i) > y_i$. In order to make the prediction interval bands smaller, a regularization term is added to the Pinball loss as Equation 4.5.

$$L_{reg} = \frac{1}{M}\sum_{i=1}^{M} (\omega \cdot L_{q_i}(\theta) + \beta_i MSE_i) \ , \ q_i \in \{0.1, 0.2, 0.3, ..., 0.9\} \text{ and } M = 9 \quad (4.5)$$

The regularized pinball loss function $L_{reg}$ is a combination of two terms. The first term involves the weighted mean of quantile loss functions $L_{q_i}$. The weight is given by the coefficient $\omega$. This part of the loss function measures the deviations between the predicted values and the actual values at different quantiles of interest.

The second term in the loss function is a regularization term, represented as $\beta_i MSE_i$, where $MSE_i$ represents the mean squared error associated with each quantile $q_i$, and $\beta_i$ are coefficients associated with each mean squared error term. This regularization term penalizes the model for deviations from the mean squared error at various quantiles. In summary, $L_{reg}$ combines quantile loss terms weighted by $\omega$ to capture the quantile-specific errors and a regularization term based on mean squared errors with $\beta_i$ coefficients.

### 4.4.6 Pinball Loss Regularized by Winkler-peak Score

Besides applying regularization using Mean Squared Error (MSE), we also incorporated regularization using Winkler-peak score into the pinball loss, as specified in Equation 4.6. The parameters remain consistent with those in Equation 4.5, except for $Wp_i$, which denotes the Winkler-peak score of the $i$th sample. The winkler-peak score was created to measure the peak time prediction and it will be explained in Section 4.5. The development of the pinball loss regularized by Winkler-peak score aimed to incentivize the quantile regression model to prioritize accurate peak time predictions.

$$L_w = \frac{1}{M} \sum_{i=1}^{M} (\omega \cdot L_{q_i}(\theta) + \beta_i W p_i \ , \ q_i \in \{0.1, 0.2, 0.3, ..., 0.9\} \text{ and } M = 9 \tag{4.6}$$

## 4.5 Performance Evaluation Metrics

To assess the uncertainty of probabilistic forecasting outcomes, one of the commonly used metrics is the Winkler score [154]. The $100(1-\alpha)\%$ prediction interval (PI) at time $t$ represented as $[L_t^\alpha, U_t^\alpha]$. In this context, the Winkler score is defined by Equation 4.7. If the observation $y_t$ falls within the $100(1-\alpha)\%$ PI, the Winkler score is the distance between the predicted upper and lower bounds. Conversely, if $y_t$ is outside the PI, the distance between $y_t$ and the closest bound of the PI is added to the width of the PI. Therefore, a smaller Winkler score indicates observations that are closer to the PIs.

$$Winkler\_Score = \begin{cases} U_t^\alpha - L_t^\alpha & \text{if } L_t^\alpha \le y_t \le U_t^\alpha \\ (U_t^\alpha - L_t^\alpha) + 2(L_t^\alpha - y_t)/\alpha & \text{if } y_t < L_t^\alpha \\ (U_t^\alpha - L_t^\alpha) + 2(y_t - U_t^\alpha)/\alpha & \text{if } y_t > U_t^\alpha \end{cases} \tag{4.7}$$

As previously discussed, while the Winkler score typically gauges the distance between observations and prediction intervals (PIs), it can be modified to measure the temporal difference between predicted spikes and actual spikes. In this scenario, the modified Winkler score, referred to as Winkler_Peak, is expressed by Equation 4.8. The final scores of day-ahead probabilistic forecasting Winkler Score or Winklear Peak are the averages of total testing Winkler Score and Winklear Peak respectively.

$$Winkler\_Peak = \begin{cases} \mid t - \hat{t} \mid & \text{if } t - e_1 \le \hat{t} \le t + e_2 \\ e_1 + \beta_1(t - e_1 - \hat{t}) & \text{if } \hat{t} < t - e_1 \\ e_2 + \beta_2(\hat{t} - t - e_2) & \text{if } \hat{t} > t + e_2 \end{cases} \tag{4.8}$$

The Continuous Ranked Probability Score (CRPS) serves as a widely employed statistical metric for assessing the accuracy of probabilistic predictions, particularly within the realm of probabilistic forecasting and prediction models [155]. In contrast to traditional point-wise evaluations that concentrate on individual predictions, CRPS evaluates the entire predictive distribution. It quantifies the disparity between the predicted cumulative distribution function (CDF) and the observed outcome, providing a holistic evaluation of a model's predictive performance across the entire range of potential outcomes. A lower CRPS value indicates better alignment between predicted and observed distributions.

Mathematically, CRPS is defined as the integral of the squared difference between the predicted cumulative distribution function and the Heaviside step function, representing the actual outcome, over the entire real line. Widely utilized in fields such as meteorology and hydrology, CRPS serves as a valuable tool for gauging the reliability and calibration of predictive models, offering insights into their ability to capture inherent uncertainty in predictions. Therefore, CRPS is employed to evaluate the probability distribution predicted by the CQR-Transformer model and multiple QR-transformer models. The CRPS calculation is expressed in Equation 4.9, where $F(x)$ denotes the cumulative distribution function of the prediction $\hat{y}$, $y$ represents the observed value, and $\mathbb{1}$ is an indicator function, signifying that $\mathbb{1}(\hat{y} \geq y) = 1$ if $\hat{y} \geq y$ and 0 otherwise.

$$CRPS(F, y) = \int_{-\infty}^{\infty} (F(\hat{y}) - \mathbb{1}(\hat{y} \geq y))^2 \, d\hat{y} \tag{4.9}$$

Because CRPS assesses the predicted cumulative distribution, the presence of cross-over quantiles in the predictions leads to a larger CRPS value. Therefore, if the cross-over quantile problem exists and the quantiles are manually sorted based on the predicted values, the CRPS value will experience a significant reduction. The CRPS calculated after sorting the quantiles is referred to as CRPS sorted in the experiments. Likewise, the pinball loss computed after sorting the quantiles is denoted as pinball loss sorted.

Additionally, the metrics employed to assess the coverage and width of the prediction intervals, namely PICP in Equation 5.12 and PIAW in Equation 5.13, are utilized to quantify the uncertainties.

## 4.6 Experiments

### 4.6.1 Experiment Settings

The experiments in probabilistic forecasting utilize the optimal configuration derived from deterministic forecasting. The pretrained model employed in this phase was the model previously trained in deterministic forecasting. Simultaneously, the case study focused on day-ahead settlement point price forecasting in the ERCOT market. Therefore, the identical dataset (see 3.4.1) and the same training and testing split were employed in this context.

Continuous features utilized in probabilistic forecasting are detailed in Table 3.1, excluding the SPP of resource nodes within the LZ_north load zone, as their inclusion was confirmed not to enhance the final predictions. Meanwhile, categorical features (3.2) are encoded using sin-cos encoding after comparing with one-hot encoding (refer to Section 3.4.5). The prediction scenario is set in line with Chapter 3, involving the prediction of day-ahead SPP for the north load zone with 7 lagged days, including a lead time of one day.

### 4.6.2 Results and Discussion

Tests were conducted to evaluate the model architectures mentioned earlier and to contrast the pinball loss function with its regularized counterpart. This evaluation encompassed the examination of QR models, composite QR models, and the GP-transformer model, focusing on their performance regarding prediction intervals, peak time estimation, and predicted probability distributions.

### 4.6.3 Evaluation of Prediction Intervals at Different Quantiles

The table presents the evaluation results of different models based on Winkler scores, with a focus on various percentile ranges (10-90%, 20-80%, 30-70%, and 40-

60%). The performance metrics are influenced by different loss functions and model architectures. Notably, the QR-LSTM model with the pinball loss function stands out, achieving the lowest scores across the majority of percentile ranges, especially excelling in the 20-80% and 30-70% intervals. This suggests that the QR-LSTM model with pinball loss effectively captures the desired quantile regression characteristics. On the other hand, the Composite QR-Transformer with pinball regloss also demonstrates competitive performance, particularly with the lowest scores in the 30-70% and 40-60% ranges. The GP-transformer, despite using the MLL loss function, exhibits relatively higher scores across all percentile ranges. The results underscore the importance of the choice of loss function and model architecture in quantile regression tasks, with QR-LSTM showing promise in this evaluation. Additionally, the Composite QR-Transformer with pinball regloss provides a compelling alternative with strong performance in specific percentile ranges.

Table 4.1. Evaluating Models Based on Winkler Scores

| **PI** | **Loss Function** | **10-90%** | **20-80%** | **30-70%** | **40-60%** |
|---|---|---|---|---|---|
| GP-Transformer | MLL | 169.416 | 105.587 | 30.715 | 22.514 |
| QR-GRU | pinball loss | 100.818 | 53.355 | 31.105 | 22.632 |
| QR-LSTM | pinball loss | **95.168** | **48.725** | 30.750 | 23.967 |
| QR-RNN | pinball loss | 104.731 | 61.007 | 27.346 | 24.057 |
| QR-Transformer | pinball loss | 117.134 | 67.335 | 26.644 | 21.658 |
| Composite QR-Transformer | pinball loss | 111.651 | 64.849 | 24.763 | **19.741** |
| Composite QR-Transformer | pinball regloss | **97.391** | 53.436 | **23.825** | **19.514** |

In summary, QR models generally outperform the GP-transformer, in terms of the Winkler score, which indicates the distance between targets and prediction intervals of QR models are smaller than the GP-transformer. RNN-type models based on quantile regression perform well in estimating larger prediction intervals, such as 10-90% and 20-80%, but exhibit lower accuracy in narrower intervals, like 30-70% and 40-60%. The composite QR-transformer model shows slight improvement over multiple QR-transformer models. Additionally, the use of pinball regloss reduces the width of the prediction interval compared to the basic pinball loss.

### 4.6.4 Evaluation of Peak Time Prediction Performance

The Winkler peak score, defined by Equation 4.8, was utilized to measure the timing of peak occurrences. Parameters $e_1 = e_2 = \beta_1 = \beta_2 = 1$ were set, indicating no preference for early or late peak predictions. Peaks were defined as the highest daily System Marginal Price (SPP) surpassing the upper bound of 36.761 \$/MWh specified in the spike transformation. The Winkler peak scores for each quantile in Table 4.2 reveal that the GP-transformer has the lowest score. Since the quantiles for the GP-transformer were derived from the same distribution, the score remains constant. The QR-transformer emerges as the second-best model for predicting spike times. In general, transformer-based models exhibit higher accuracy in predicting peak times

Table 4.2. Evaluating Models Based on Winkler Peak

| **Quantiles** | **GP-Transformer** | **QR-transformer** | **QR-GRU** | **QR-LSTM** | **QR-RNN** | **Composite QR-Transformer** | **Composite QR-Transformer (re-gloss)** |
|---|---|---|---|---|---|---|---|
| Q10 | 3.365 | 3.433 | 3.996 | 3.928 | 3.379 | 3.267 | 3.617 |
| Q20 | 3.365 | 3.350 | 3.827 | 3.726 | 3.773 | 3.722 | 3.531 |
| Q30 | 3.365 | 3.426 | 3.874 | 3.751 | 4.177 | 3.747 | 3.505 |
| Q40 | 3.365 | 3.531 | 3.599 | 3.780 | 3.614 | 3.650 | 3.466 |
| Q50 | 3.365 | 3.469 | 3.823 | 3.372 | 3.949 | 3.711 | 3.408 |
| Q60 | 3.365 | 3.390 | 3.975 | 3.773 | 4.585 | 3.588 | 3.487 |
| Q70 | 3.365 | 3.249 | 4.036 | 3.679 | 3.444 | 3.415 | 3.408 |
| Q80 | 3.365 | 3.231 | 3.859 | 4.090 | 4.386 | 3.422 | 3.430 |
| Q90 | 3.365 | 3.343 | 4.227 | 3.953 | 4.300 | 3.596 | 3.310 |
| Average | **3.365** | 3.380 | 3.913 | 3.783 | 3.956 | 3.569 | 3.462 |

### 4.6.5 Evaluation of Probabilistic Prediction Performance in terms of CRPS and Pinball Loss

Table 4.3 provides an assessment of different models using the Pinball loss metric. The models evaluated include GP-transformer, QR-GRU, QR-LSTM, QR-RNN, QR-Transformer, Composite QR-Transformer with Pinball loss, and Composite QR-Transformer with Pinball regloss. The results are organized into columns specifying the model, loss function employed, Pinball loss values, Pinball loss values sorted, and the difference between sorted and unsorted Pinball losses.

The pinball loss values are recorded for each model, with the QR-Transformer model achieving the lowest Pinball loss at 3.508. The Composite QR-Transformer with Pinball loss and GP-Transformer follow closely with values of 3.517 and 3.559, respectively. The table provides insights into the performance of each model based on the Pinball loss metric, highlighting the differences between unsorted and sorted Pinball losses.

Sorted pinball loss involves arranging quantiles based on prediction values to address the crossed quantiles issue. As the GP-Transformer's quantile predictions are derived from a multivariate normal distribution, the order of quantiles remains consistent. Consequently, the Pinball loss is identical, whether the quantile predictions are sorted or unsorted for the GP-Transformer. While the Pinball loss values for quantile regression models may not exhibit substantial differences, transformer-based quantile regression models demonstrate notably minor discrepancies between sorted and unsorted results. This suggests that transformer-based quantile regression models are capable of generating more organized quantile predictions.

Table 4.4 presents an evaluation of various models using the Continuous Ranked Probability Score (CRPS) metric. The CRPS values for each model are recorded, with the Composite QR-Transformer achieving the lowest CRPS at 6.333, closely followed

Table 4.3. Evaluating Models Based on Pinball Loss

| **Models** | **Loss Function** | **Pinball loss** | **Pinball loss sorted** | **Difference** |
|---|---|---|---|---|
| GP-Transformer | MLL | 3.559 | 3.559 | **0.000** |
| QR-GRU | pinball loss | 4.606 | 3.866 | 0.740 |
| QR-LSTM | pinball loss | 4.581 | 3.895 | 0.687 |
| QR-RNN | pinball loss | 4.194 | 3.672 | 0.522 |
| QR-Transformer | pinball loss | **3.508** | **3.496** | **0.012** |
| Composite QR-Transformer | pinball loss | 3.517 | 3.506 | **0.011** |
| Composite QR-Transformer | pinball regloss | 3.612 | 3.599 | **0.013** |

by the QR-Transformer at 6.351. GP-Transformer maintains its ranking as the third-best predictor based on the CRPS score, and this score remains unaffected by the sorting operation. Both QR-Transformer and composite QR-Transformer exhibit relatively small differences between sorted and unsorted CRPS results. This suggests that the Composite QR-Transformer and QR-Transformer are effective in generating consistent and accurate probabilistic forecasts, as indicated by the small discrepancies in CRPS values between sorted and unsorted outcomes. The utilization of Pinball regloss does not enhance the predictive performance in comparison to the standard Pinball loss, as indicated by the CRPS value.

Table 4.4. Evaluating Models Based on CRPS

| **Models** | **Loss Function** | **CRPS** | **CRPS sorted** | **Difference** |
|---|---|---|---|---|
| GP-Transformer | MLL | 6.476 | 6.476 | 0.000 |
| QR-GRU | pinball loss | 7.765 | 7.001 | 0.764 |
| QR-LSTM | pinball loss | 7.823 | 7.050 | 0.773 |
| QR-RNN | pinball loss | 6.995 | 6.647 | 0.347 |
| QR-Transformer | pinball loss | 6.351 | 6.342 | **0.009** |
| Composite QR-Transformer | pinball loss | **6.333** | **6.327** | **0.005** |
| Composite QR-Transformer | pinball regloss | 6.525 | 6.516 | **0.010** |

### 4.6.6 Evaluation of Uncertainty Quantification Performance

The PICP (see Equation 5.12) is defined as the ratio of the number of targets covered by prediction intervals to the total number of targets. PIAW (see Equation 5.13) is defined as the average distance between the upper bound and the lower bound. As shown in Table 4.5, GP-Transformer exhibits the highest coverage rate at the 80% prediction interval, but the corresponding prediction interval is excessively wide. An effective probabilistic forecasting model should cover a proportion of targets close to the expected prediction interval. In this context, the QR-Transformer falls short of achieving the anticipated 80% coverage rate.

Table 4.5. Evaluating Models Based on PICP and PIAW

| **Models** | **Loss Function** | **PI** | **PICP** | **PIAW** |
|---|---|---|---|---|
| GP-transformer | MLL | 80% | **0.903** | 34.755 |
| QR-GRU | pinball loss | 80% | 0.162 | 2.410 |
| QR-LSTM | pinball loss | 80% | 0.077 | 0.844 |
| QR-RNN | pinball loss | 80% | 0.332 | 7.839 |
| QR-Transformer | pinball loss | 80% | **0.703** | 17.801 |
| Composite QR-Transformer | pinball loss | 80% | **0.696** | 14.688 |
| Composite QR-Transformer | pinball regloss | 80% | 0.525 | 8.400 |

The 80% prediction intervals for a subset of the testing set, generated by the aforementioned models, are depicted in Figure 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, and 4.14. The 20% prediction intervals are represented by red bands, while the 80% prediction intervals are denoted by blue bands. Targets are represented by black lines. Similar observations can be made from these figures as those derived from the results presented in the previous tables. GP-Transformer covers a substantial portion of targets, but its prediction band is excessively wide, impacting accuracy.

In contrast, the prediction bands of RNN-type models are too narrow to encompass most targets, leading to pronounced boundary issues. While the prediction intervals of QR-Transformer do not cross as frequently as RNN models, the peaks are higher. The composite QR-Transformer model mitigates the peaks but widens the bands. Although Pinball regloss penalizes the width of prediction intervals, it simultaneously compromises accuracy.

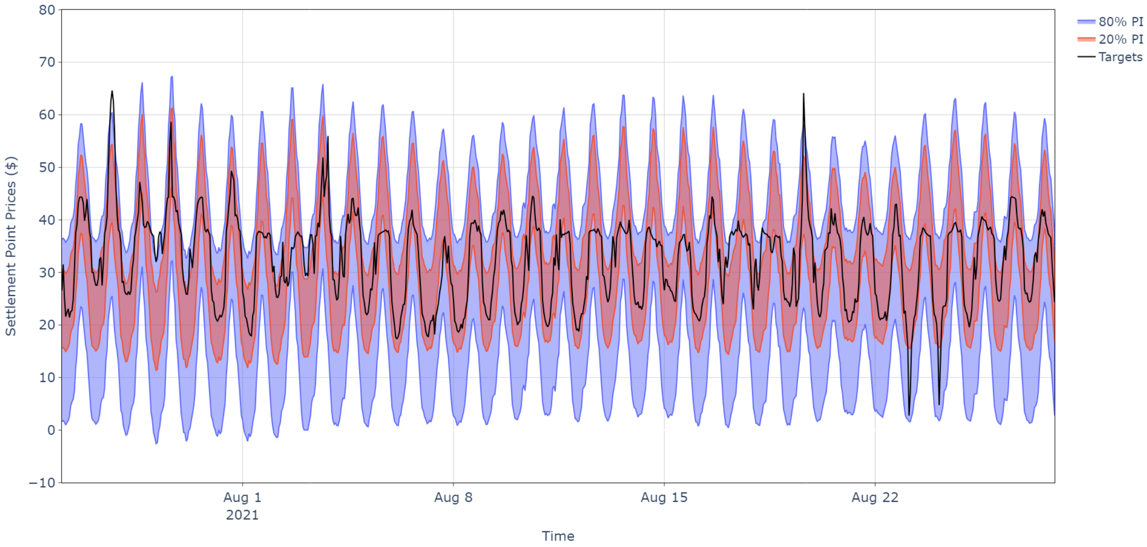


Figure 4.8. Prediction Intervals Comparison Between GP-Transformer Predictions and Targets for 80% and 20% Intervals.

### 4.6.7 Visualization of Probabilistic Predictions by Different Models

In Figure 4.15, 4.16, and 4.17, the composite QR-Transformer model predicts quantiles for various scenarios. In Figure 4.15, the prediction reflects a situation without spikes, successfully capturing the cyclic pattern of SPP. The majority of the targets (represented by the blue line) are covered by the predicted 90th quantile.

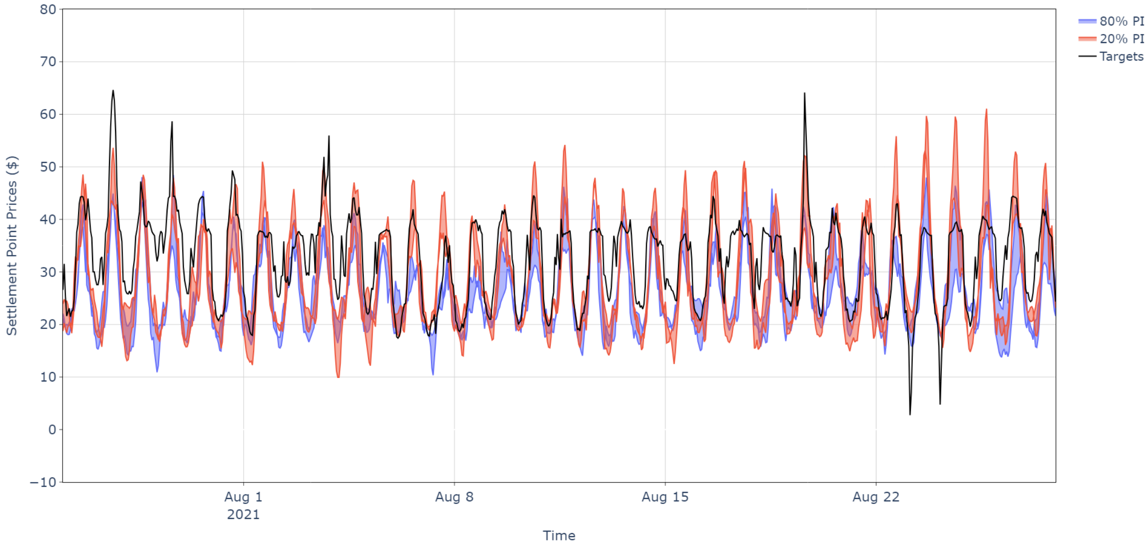


Figure 4.9. Prediction Intervals Comparison Between QR-GRU Predictions and Targets for 80% and 20% Intervals.

Regarding Figure 4.16, it depicts quantile predictions in the presence of spikes. As discussed in Chapter 3, the spike transformation method 1 is employed to handle spikes, making the actual spike values potentially higher than those visible in the figure. Quantile prediction encounters challenges in adequately covering these spikes.

Figure 4.17 showcases the quantile prediction for February 2021, marked by a hike due to a snowstorm. Despite the quantile prediction not fully capturing the SPP during hikes, a discernible upward trend is evident in the predictions during such events.

### 4.6.8 Evaluation of Prediction Performance by Winkler-Peak Score Regularized Pinball Loss

Table 4.6 displays the Winkler-peak scores for composite quantile regression using both pinball loss and Winkler-score penalized pinball loss. The table indicates that the regularization term effectively encourages the model to approach peak time

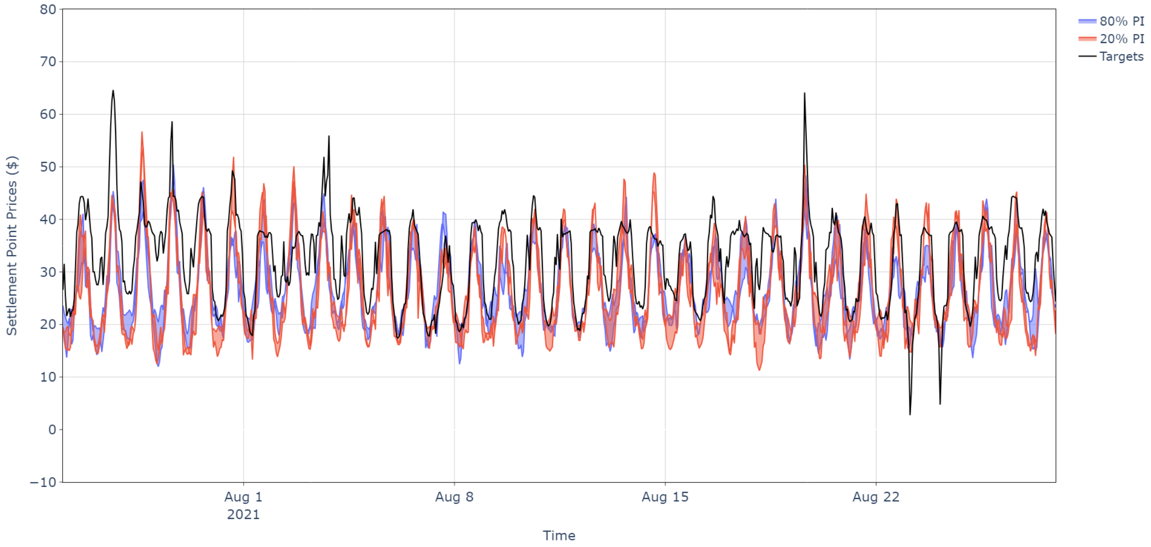


Figure 4.10. Prediction Intervals Comparison Between QR-LSTM Predictions and Targets for 80% and 20% Intervals.

predictions. However, it is crucial to optimize the regularization weight for optimal performance.

### 4.6.9 Evaluation of Prediction Performance by MSE Regularized Pinball Loss

The composite QR-Transformer can also be employed for load forecasting, with tests conducted on four sites: Amity, Donalsonville, San Antonio, and Waianae. Training data, gathered from 1 AM on January 1, 2022, to 11 PM on April 30, 2023, in their respective local time zones, were utilized. The model underwent testing on load data spanning from 1 AM on May 1, 2023, to 11 PM on May 30, 2023. In this particular application, the calculation of two extreme quantiles at 0% and 100% levels may encounter the issue of an unbounded boundary, as per the definition of pinball loss. As indicated in Table 4.7, pinball loss regularized by MSE (referred to as pinball regloss) resulted in more accurate outcomes based on CRPS and pinball loss metrics. Furthermore, upon comparing the disparities between CRPS and CRPS

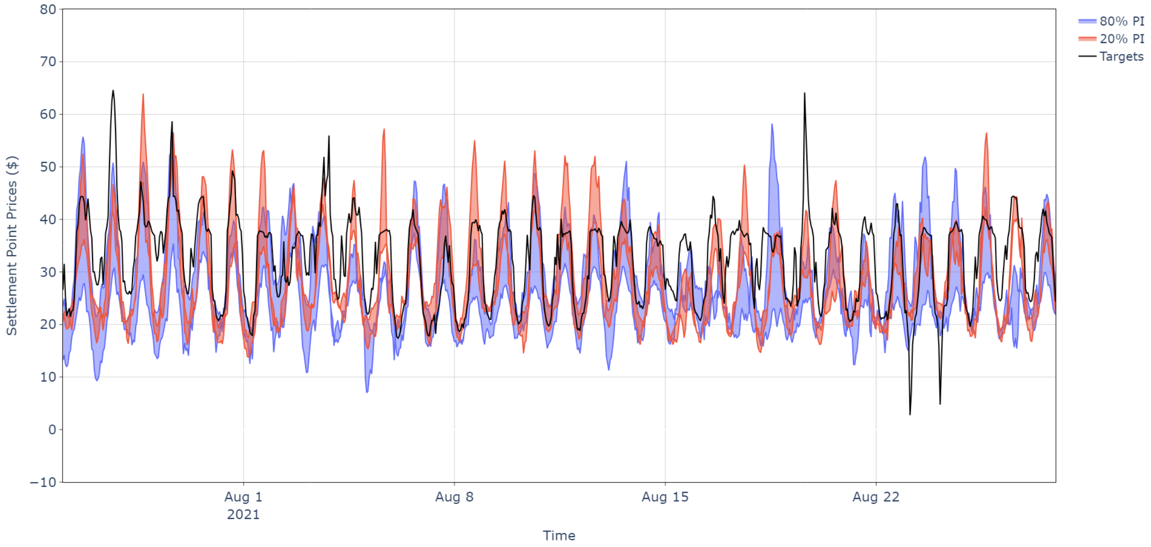


Figure 4.11. Prediction Intervals Comparison Between QR-RNN Predictions and Targets for 80% and 20% Intervals.

sorted, as well as pinball loss and pinball loss sorted, it is evident that the pinball regloss estimated quantiles exhibit fewer cross-over issues, as indicated by smaller differences.

The model performances were additionally evaluated during local daytime hours, specifically from 1 PM to the next day's 3 AM UTC time for Amity, 11 PM to the next day's 0 AM UTC time for Donalsonville, 12 AM to the next day's 1 AM for San Antonio, and 5 PM to the next day's 5 AM for Waianae. The composite quantile regression estimated with pinball regloss continued to demonstrate superior performance compared to the model estimated with pinball loss.

## 4.7 Conclusions

In this chapter, a comparison is made between probabilistic forecasting methods based on quantile regression and Gaussian processes. Building upon the deterministic multivariate time series forecasting structure proposed in Chapter 3, the second step

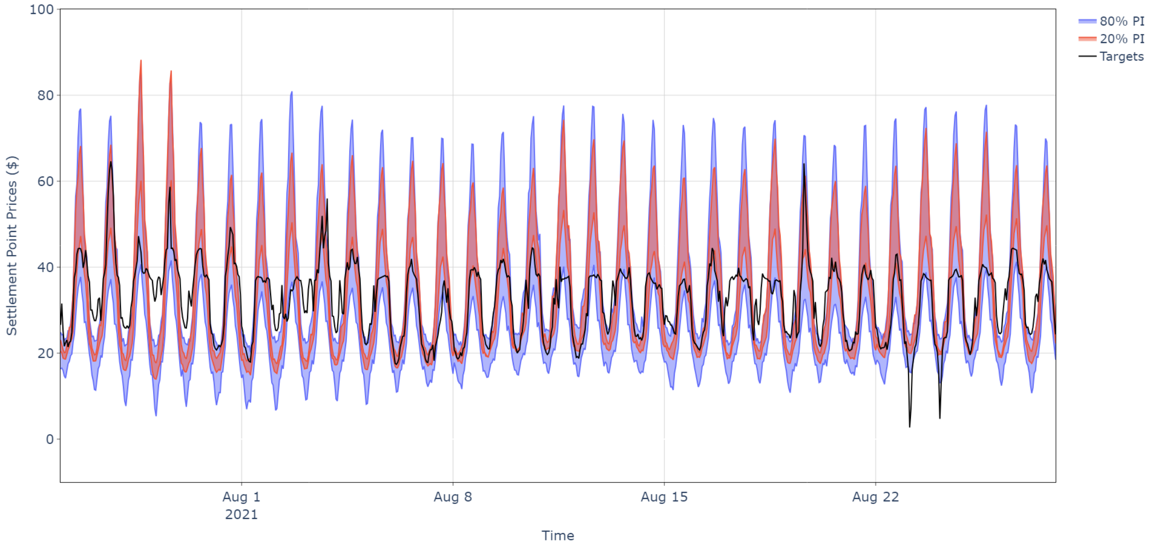


Figure 4.12. Prediction Intervals Comparison Between QR-Transformer Predictions and Targets for 80% and 20% Intervals.

of the two-step training structure was adjusted to predict different quantiles, replacing the mean squared error (mse) loss with the pinball loss for electricity prices.

Through comparisons of prediction intervals from QR-Transformer, QR-GRU, QR-LSTM, and QR-RNN models, QR-Transformer consistently demonstrates stable and accurate performance. The potential overfitting issue observed in RNN-type models persists in probabilistic forecasting, evidenced by narrow deviated intervals.

GP-Transformer model achieved better peak prediction in terms of Winkler-Peak score. However, its uncertainty quantification is less accurate and reliable with a very large bandwidth. Although GP-Transformer effectively covers a majority of the targets, the reliability of probabilistic predictions is compromised by wider bands compared to quantile regression models.

The prediction intervals generated by the composite structure of quantile regression were compared with intervals composed of multiple quantile regression models. The Composite QR-Transformer, which considers the loss of all quantiles simultane-

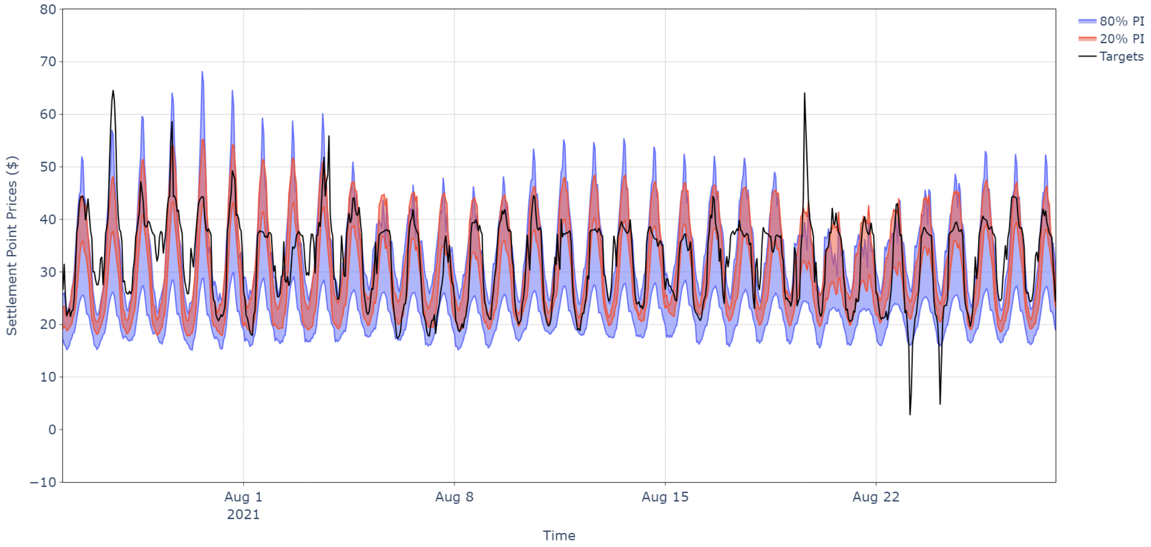


Figure 4.13. Prediction Intervals Comparison Between Composite QR-Transformer Predictions and Targets with Pinball Loss for 80% and 20% Intervals.

ously, demonstrated superior probabilistic prediction performance in CRPS compared to all other methods.

The Winkler-peak regularized pinball loss proved effective in directing model attention to the timing of time series peaks, thereby enhancing peak prediction performance. The MSE-regularized pinball loss was found effective in controlling prediction intervals of quantile regression. Optimizing the regularization parameters during training, monitored by validation CRPS, holds the potential to further enhance the probabilistic forecasting performance of the proposed Composite QR-Transformer method.

Analysis of PICP and PIAW for prediction intervals reveals that existing methodologies face challenges in providing precise uncertainty quantifications for target prediction intervals with specified probabilities (e.g., 80%). This underscores the clear need for improved methods that can deliver more accurate uncertainty quantification.

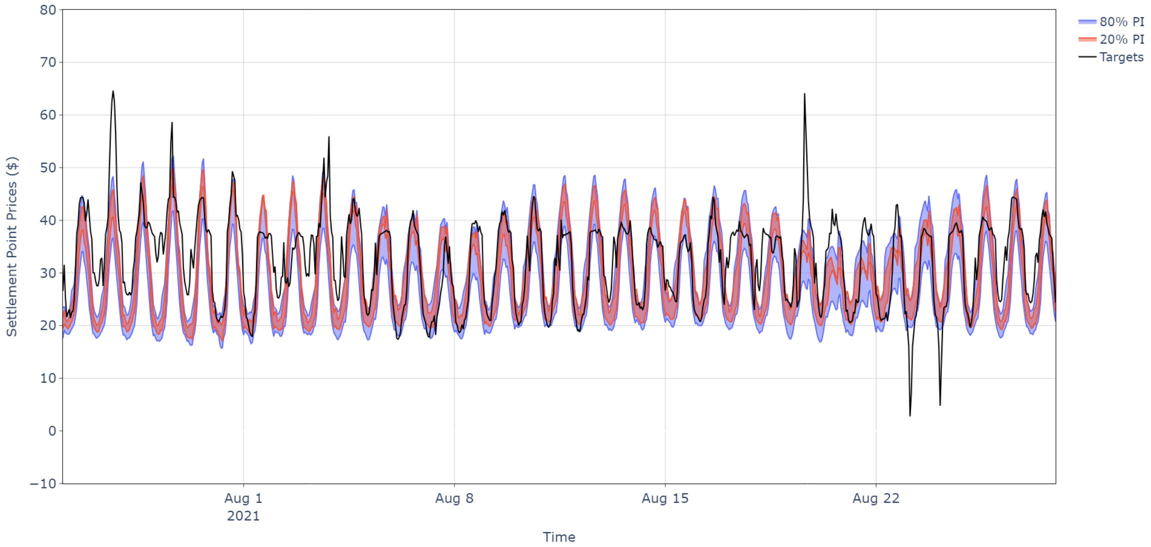


Figure 4.14. Prediction Intervals Comparison Between Composite QR-Transformer with Pinball Regloss Loss Predictions and Targets for 80% and 20% Intervals.

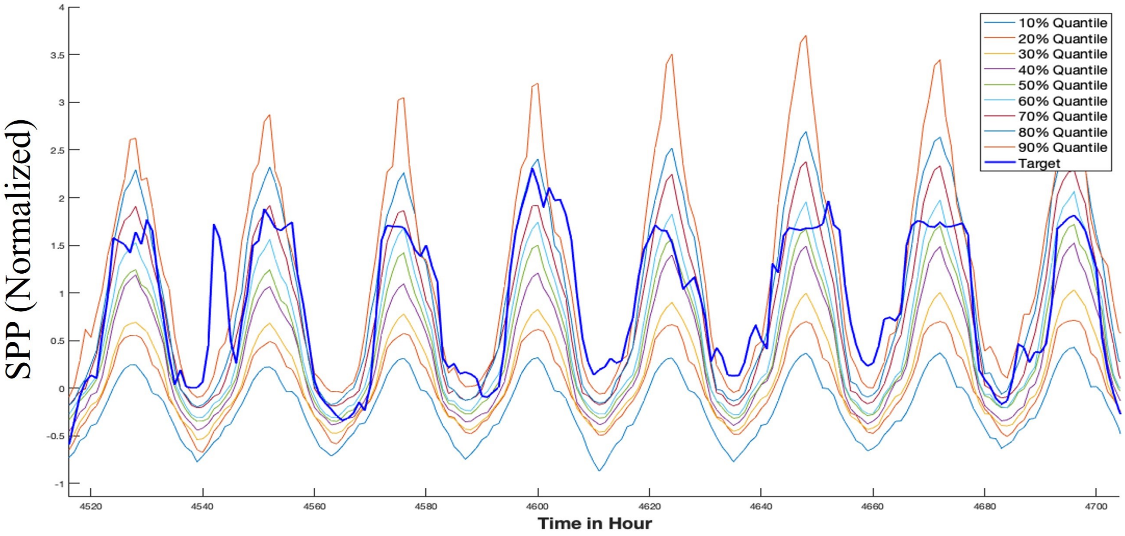


Figure 4.15. Day-ahead Quantile Prediction on Settlement Point Prices With No Spikes.

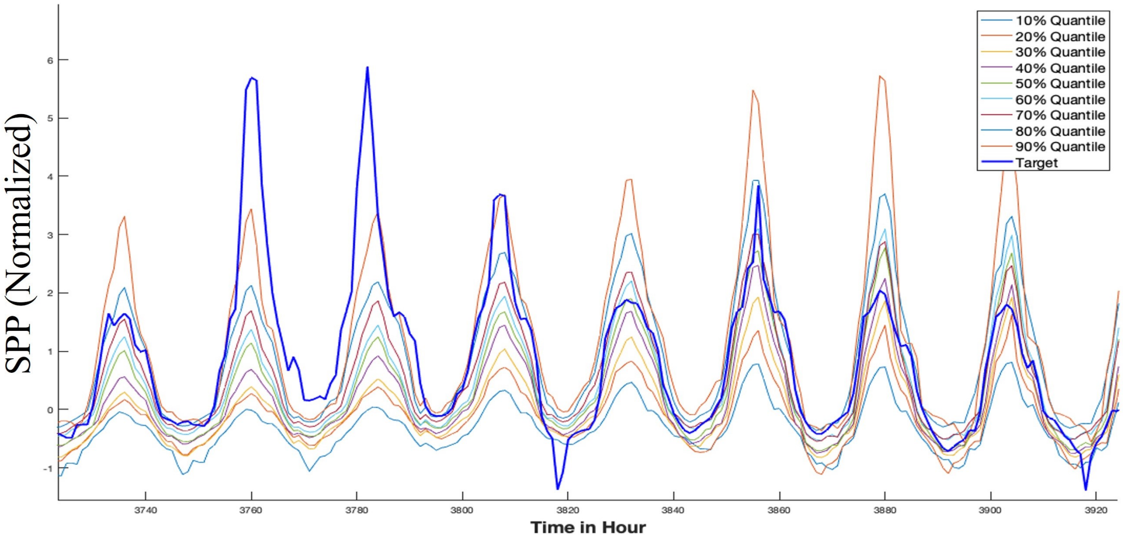


Figure 4.16. Day-ahead Quantile Prediction on Settlement Point Prices With Spikes.

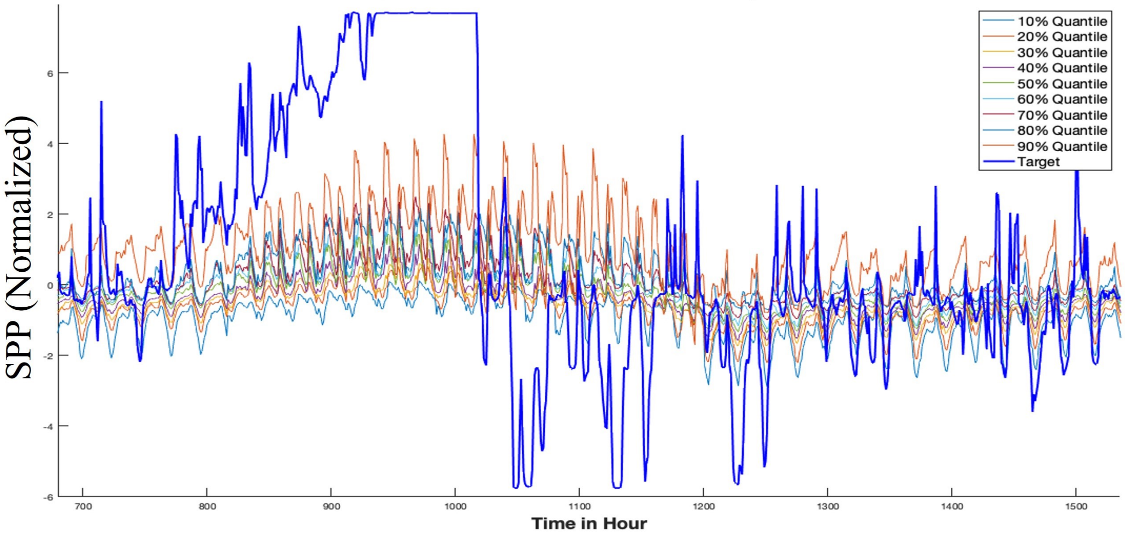


Figure 4.17. Day-ahead Quantile Prediction on Settlement Point Prices With Hikes.

Table 4.6. Winkler-peak Score of Composite Quantile Regression Estimated by Winkler-peak Score Regularized Pinball Loss

| **Quantiles** | **Composite QR-transformer** | **regweight=7** | **regweight=1** |
|---|---|---|---|
| Q10 | 3.267 | 3.592 | 3.509 |
| Q20 | 3.722 | 3.621 | 3.617 |
| Q30 | 3.747 | 3.437 | 3.661 |
| Q40 | 3.650 | 3.433 | 3.560 |
| Q50 | 3.711 | 3.426 | 3.119 |
| Q60 | 3.588 | 3.397 | 3.361 |
| Q70 | 3.415 | 3.440 | 3.487 |
| Q80 | 3.422 | 3.455 | 3.556 |
| Q90 | 3.596 | 3.390 | 3.404 |
| Average | 3.569 | 3.466 | 3.475 |

Table 4.7. Evaluation of Load Forecasting With Composite QR-Transformer

| **Site** | **Objective function** | **CRPS** | **CRPS (sorted)** | **Pinball Loss** | **Pinball Loss (sorted)** |
|---|---|---|---|---|---|
| Amity | Pinball loss | 0.114 | 0.087 | 0.02 | 0.018 |
| Amity | Pinball regloss | 0.042 | 0.043 | 0.022 | 0.018 |
| Donalsonville | Pinball loss | 0.063 | 0.059 | 0.023 | 0.022 |
| Donalsonville | Pinball regloss | 0.045 | 0.044 | 0.021 | 0.02 |
| San Antonio | Pinball loss | 0.071 | 0.053 | 0.019 | 0.016 |
| San Antonio | Pinball regloss | 0.038 | 0.038 | 0.02 | 0.017 |
| Waianae | Pinball loss | 0.152 | 0.145 | 0.035 | 0.034 |
| Waianae | Pinball regloss | 0.083 | 0.082 | 0.036 | 0.036 |

Table 4.8. Evaluation of Daytime Load Forecasting With Composite QR-Transformer

| **Site** | **Objective function** | **CRPS of Daytime** | **CRPS of Daytime (sorted)** |
|---|---|---|---|
| Amity | Pinball loss | 0.116 | 0.099 |
| Amity | Pinball regloss | 0.048 | 0.05 |
| Donalsonville | Pinball loss | 0.072 | 0.066 |
| Donalsonville | Pinball regloss | 0.051 | 0.05 |
| San Antonio | Pinball loss | 0.073 | 0.052 |
| San Antonio | Pinball regloss | 0.039 | 0.04 |
| Waianae | Pinball loss | 0.213 | 0.207 |
| Waianae | Pinball regloss | 0.124 | 0.124 |

# CHAPTER 5

# IMPROVING UNCERTAINTY QUANTIFICATION of PROBABILISTIC MULTIVARIATE TIME SERIES FORECASTING WITH CONFORMAL PREDICTION

## 5.1 Introduction

In applications of probabilistic modeling and forecasting, it is important not only to predict accurately but also to quantify prediction uncertainty accurately. This is especially important in situations involving high-stakes decision making, such demand response decisions, healthcare decisions, financial risk management, natural disaster prediction and response planning, and autonomous vehicle navigation. In many real-world decision-making scenarios, understanding the range of potential outcomes and their probabilities with accurate uncertainty quantification can be crucial for making informed, safe, and effective decisions. Quantifying prediction uncertainty helps in assessing risks, preparing for various scenarios, and choosing actions that are robust against a wide range of possible futures.

For major probabilistic forecasting methods, such as quantile regression and Gaussian Process based methods, the prediction uncertainty can be quantified using a prediction interval, giving lower and upper bounds between which, the response variable lies with a target probability. The quantile regression methods estimate the prediction interval directly using the pinball loss at different quantiles. The GP based methods estimate the prediction intervals of a desired probability by assuming a Gaussian distribution with the predicted mean and standard deviation of the response variable. However, it is noted that most of the probabilistic forecasting methods have limitations on accurate uncertainty quantification in practical applications.

For quantile regression methods, while it is valuable for estimating different quantiles in the distribution of a response variable, often face challenges in accurately predicting quantiles in practice. The accuracy of quantile regression in predicting quantiles can be seriously impacted by several factors:

- Model Specification Issues: If the quantile regression model is not correctly specified — for example, if important predictors are omitted or nonlinear relationships are not properly accounted for — it may not capture the true underlying relationship between the variables. This can lead to inaccurate quantile predictions.
- Data Limitations: The quality and quantity of data available for training can significantly impact the model's accuracy. Insufficient data, especially for extreme quantiles, can lead to unreliable estimates. Also, if the training data is not representative of the population or the situation where the model is applied, it can lead to inaccuracies.
- High Variability in Extreme Quantiles: Estimating extreme quantiles (e.g., 5th or 95th percentile) can be challenging because these are often based on less data (i.e., the tails of the distribution), leading to higher variability and potentially less accurate predictions.
- Impact of Outliers: Quantile regression can be sensitive to outliers, especially for the extreme quantiles. Outliers can disproportionately influence the model estimates, leading to skewed results.
- Data Distribution Drift in Non-Stationary Data: If there is a drift or change in the data distribution between the training phase and testing/application phase, the quantile estimates may become inaccurate. This drift can happen

due to various factors like changing market conditions, evolving patient profiles in healthcare, or environmental changes. Such distribution drifts mean that the patterns learned during training may no longer apply in the same way, leading to less reliable quantile predictions in practice.

On the other hand, GP methods, while powerful in many respects, also face challenges in accurately predicting quantiles in practice due to several reasons:

- Assumption of Gaussian Distribution: GP methods generally assume that data follow a Gaussian distribution. However, real-world data can exhibit non-Gaussian characteristics like skewness, heavy tails, kurtosis, or multimodality. When the true distribution of data deviates significantly from the Gaussian assumption, GP methods may struggle to accurately capture the underlying distribution, leading to inaccuracies in quantile prediction.
- Computational Intensity and Scalability: GP methods are known for their computational intensity, especially as the size of the dataset grows. For very large datasets, the computational requirements can become prohibitive, potentially leading to compromises in model complexity or precision. This can affect the model's ability to accurately estimate quantiles.
- Sensitivity to Hyperparameters and Kernel Choice: The performance of GP methods is highly dependent on the choice of kernels and hyperparameters. Incorrect choices can lead to poor model fit and inaccurate quantile predictions. In practice, finding the optimal configuration can be challenging and time-consuming.
- Difficulty in Handling Non-Stationary Data: GP methods assume stationary processes. In real-world scenarios, where data might exhibit non-stationary behaviors due to trends, seasonality, or regime shifts, GPs might not perform well, affecting their ability to predict quantiles accurately.

Due to these limitations, there is a motivation to investigate uncertainty quantification accuracy and develop new methods to calibrate and improve the performance of uncertainty quantification for probabilistic forecasting models, which can be critical in practical decision-making processes. In recent years, conformal prediction methods have emerged as a powerful tool for uncertainty quantification in machine learning and statistical predictions [24]. Its reliability in quantifying prediction uncertainties has led to widespread applications in diverse domains such as decision support systems [156], safe navigation for self-driving cars [157], and drug discovery [158].

Conformal prediction provides a layer on top of existing models (like quantile regression) to offer a more reliable measure of prediction uncertainty. It does so by using past data to determine how well predictions would have worked in practice, thus providing empirically valid confidence intervals or prediction intervals. This approach can be particularly useful in high-stakes decision-making scenarios where understanding and quantifying uncertainty is crucial. Conformal prediction helps in addressing some of the limitations of quantile regression by providing a framework for assessing and guaranteeing the accuracy of prediction intervals. In particular, the concept of conformal prediction was introduced to address several key aspects:

- Need for Reliable Uncertainty Estimation: traditional probabilistic forecasting models face challenges to quantify prediction uncertainties accurately due to various model and practical factors. Conformal prediction methods is aimed to quantify prediction uncertainty in a reliable and theoretically sound manner.
- Coverage Guarantee: one of the most appealing aspects of conformal prediction is its coverage guarantee. This means that the prediction intervals generated by conformal methods are guaranteed to contain the true outcome with a specified

probability (e.g., 95%). This guarantee holds under very general conditions, making conformal methods widely applicable.

- Non-Parametric and Distribution-Free: Conformal prediction methods do not rely on specific assumptions about the underlying data distribution. This non-parametric and distribution-free nature makes them robust and applicable to a wide range of problems, including those with complex or unknown data distributions.
- Adaptability to Different Models: Conformal prediction methods can be applied on top of any existing predictive model, whether it's a simple linear regression or a complex deep neural network. This adaptability allows for the integration of conformal prediction into existing prediction pipelines with minimal changes.

With the rise of complex machine learning models and the availability of large and diverse datasets, the need for robust uncertainty quantification has become more pronounced. However, despite its importance, robust uncertainty quantification using conformal prediction concept is still an underdeveloped research direction and has not been adopted by many machine learning studies in practice. In Chapter 4, a transformer-based composite quantile regression (TCQR) model was proposed for multi-step multivariate time series forecasting. Compared to existing methods, the proposed method achieved promising performance of prediction accuracy in terms of RMSE, MAE, MAPE, and achieved the promising performance in probabilistic prediction metrics in terms of pinball loss and CRPS. However, we notice that the quality of uncertainty quantification can not be guaranteed at different quantiles. As shown in Table 5.1, the target prediction bands of 80%, 60%, 40%, 20% were defined by the predicted quantiles of (10% - 90%), (20% - 80%), (30% - 70%), (40% - 60%), respectively, by the TCQR model on the testing dataset (the last one year of energy prices). The actual coverage rates were 63.9%, 47.6%, 31.5%, 13.9%, respectively, which were significantly deviated from the target coverage rates. The big difference between target and actual coverage rates shows that the existing models have issues to provide accurate and reliable uncertainty quantification. There is an pressing need to develop new probabilistic forecasting methods with more accurate and robust uncertainty quantification, which can be critical in real-world decision-making applications.

Table 5.1. Coverage Rate and Bandwidth of Composite QR-Transformer Before Calibration

| | **80%** | **60%** | **40%** | **20%** |
|---|---|---|---|---|
| Bandwidth | 14.927 | 8.821 | 5.125 | 2.388 |
| Coverage Rate | 0.639 | 0.476 | 0.315 | 0.139 |

In Chapter 5, we specifically investigated the problem of model uncertainty quantification. In particular, we develop a new adaptive conformal prediction method to improve uncertainty quantification for probabilistic forecasting models. The developed new method is promising to generate accurate prediction intervals with guaranteed coverage and is highly valuable to be used for various data-driven decision-making applications.

### 5.1.1 Motivation and Contributions

In this chapter, conformal prediction was implemented to enhance probabilistic prediction results and offer a statistically robust uncertainty quantification. Existing methods, including conformalized quantile regression and conformal residual fitting methods, were found to overlook the application in multivariate time series and multi-step ahead forecasting. Moreover, these methods do not adapt to the latest data, suggesting that existing methods cannot be applied to datasets with the data shift issue. Therefore, contributions are made in the following aspects.

(1) Proposed the integration of adaptive online updates with quantile random forest-based conformal prediction (AQRF) to achieve narrow bandwidth and guarantee the coverage rate at the same time.

(2) Proposed the combination of adaptive online updates with the conformal residual fitting method (ACRF) to update calibration using the latest available data.

(3) Adjusted AQRF and ACRF methods to multivariate time series and multi-step ahead forecasting application.

(4) Conducted a comparison between the proposed AQRF and ACRF methods and the original QRF and CRF methods.

## 5.2 Related Work

Conformal prediction utilizes historical data to establish accurate confidence levels for new predictions [159]. As a general method to generate prediction intervals without distribution assumption, conformal prediction has a general procedure as follows [24].

(1) Establish a heuristic measure of uncertainty based on the pre-trained model.

(2) Specify the score function, where larger scores indicate poorer agreement between prediction and target.

(3) Determine the quantile of the calibration score.

(4) Utilize this quantile to construct prediction sets for new instances.

The majority of research is concentrated on either defining score functions, determining quantiles, or establishing methods for constructing prediction sets for new instances.

**Conformal Prediction** There are different ways to define the score functions in the second step. The most straightforward way is using nonconformity score which is the absolute error as defined in Equation 5.1 [160]. $\hat{\mu}(X_i)$ is the predicted point value of $i$th sample $(X_i, Y_i)$, and $\mathbb{D}_{cal}$ denotes the calibration dataset.

$$E_i = |Y_i - \hat{\mu}(X_i)|, \text{ where } X_i, Y_i \in \mathbb{D}_{cal} \tag{5.1}$$

The prediction interval at $100(1-\alpha)\%$ confidence level for conformal prediction using the nonconformity score for a new data point is represented by Equation 5.2 [24].

$$C(X_{t+1}) = [\hat{\mu}(X_{t+1}) - Q(1-\alpha, E \in \mathbb{I}_{cal}), \hat{\mu}(X_{t+1}) + Q(1-\alpha, E \in \mathbb{I}_{cal})] \tag{5.2}$$

where $Q(1-\alpha, E_i \in \mathbb{I}_{cal})$ is an empirical $100(1-\alpha)$-th quantile of errors of calibration set$E \in \mathbb{I}_{cal}$.

**Conformalized Quantile Regression (CQR)** However, the effectiveness of conformal calibration using the mentioned nonconformity score diminishes when applied to heteroskedastic errors. To address this issue, conformalized quantile regression (CQR) [26] was introduced as a solution to the heteroskedasticity problem. Assuming the boundary of the prediction interval at the $100(1-\alpha)\%$ confidence level derived from the calibration dataset of probabilistic forecasting is given by $\{\hat{q}_{\alpha_{low}}, \hat{q}_{\alpha_{hi}}\}$, the conformity score for CQR is defined by Equation 5.3. The coverage rate would be guaranteed by this maximum function.

$$\begin{aligned} E_i = max\{\hat{q}_{\alpha_{low}}(X_i) - Y_i, Y_i - \hat{q}_{\alpha_{hi}}(X_i)\} \\ X_i, Y_i \in \mathbb{D}_{cal} \end{aligned} \tag{5.3}$$

The boundaries of the prediction interval at $\alpha$ level, $C(X_{t+1})$, is denoted as Equation 5.4 [26].

$$\begin{aligned} C(X_{t+1}) = [\hat{q}_{\alpha_{low}}(X_{t+1}) - Q(1-\alpha, E \in \mathbb{I}_{cal}), \hat{q}_{\alpha_{hi}}(X_{t+1}) + Q(1-\alpha, E \in \mathbb{I}_{cal})] \\ \mathbb{I}_{cal} = \{E_t, E_{t-1}, ..., E_{t-n}\} \end{aligned} \tag{5.4}$$

where $Q_{1-\alpha}(E \in \mathbb{I}_{cal}) = (1-\alpha)(1+1/|\mathbb{I}_{cal}|)$-th empirical quantile of $\mathbb{I}_{cal}$. $|\mathbb{I}_{cal}|$ is the number of samples in the calibration datasets.

Conformal predictions can be categorized into two types, namely Inductive Conformal Prediction (ICP) and Transductive Conformal Prediction (TCP), based on how they construct prediction intervals [161].

**TCP vs ICP** In TCP, the initial step of creating a general ”rule” is omitted. Consequently, there is no preprocessing applied to the training set beforehand, and all computations rely on each individual test example. The predictions are derived using the actual training set for each specific test example. The process of ICP involves using a training set to formulate a general rule, model, or theory about the data. This generated rule is then applied to individual test patterns to make predictions. The information from the training set is integrated into the general rule, and there is no direct reliance on the training examples during the prediction process [161]. This study will primarily concentrate on the ICP method, which is designed to create a rule using training data.

Given that the research emphasis is on ICP methods, and conformal prediction is employed to calibrate probabilistic forecasting in this chapter, the model utilized for predicting the quantiles should be lightweight and computationally efficient. Therefore, opting for a quantile random forest would be a suitable choice[162].

**Conformal Residual Fitting (CRF)** Nevertheless, according to [163], ICP was found to be inefficient in capturing the local patterns of individual samples.

Consequently, CRF was introduced to create locally adaptive intervals by employing a normalized nonconformity score, as illustrated in Equation 5.5. In this equation, $\sigma$ represents a conformal normalization model.

$$\gamma(x) = \frac{|y - f(x)|}{\sigma(x)} \tag{5.5}$$

**Adaptive Conformal Inference (ACI)** Considering the problem of data shift in the real world, Gibbs and Candes (2021) [164] introduced Adaptive Conformal Inference, which accommodates an unspecified number of shifts in the joint distribution. This approach involves retraining the predictive model and dynamically adjusting the quantile level through an online update mechanism, utilizing a learning rate parameter.

Recognizing the sequential characteristics present in time series data, a framework known as Sequential Predictive Conformal Inference (SPCI) was introduced by Xu and Xie (2023) [165], incorporating autoregressive updates. In this section, a model akin to SPCI was developed, employing a quantile random forest to predict the quantiles of the nonconformity score for each new data point. Additionally, adaptive conformal inferences were integrated to dynamically adjust and update the prediction intervals.

## 5.3 Proposed Methods

### 5.3.1 Multi-Step Multivariate Time Series Conformal Prediction

**Hourly Conformal Prediction** For day-ahead electricity price forecasting, where the prediction spans multiple time steps and the residual distribution may vary for each hour, the hourly nonconformity scores were formulated as detailed in Equation 5.6. Here, $\hat{\mu}(X_i)$ represents the 50th quantile prediction from probabilistic forecasting. $Y_i$ denotes the $i$th observation. $\mathbb{D}_h$ constitutes the calibration dataset specific to hour $h$. $\mathbb{I}_h$ is used to denote nonconformity scores specific to hour $h$.

$$\begin{gathered} E_i = |Y_i - \hat{\mu}(X_i)|, \text{ where } X_i, Y_i \in \mathbb{D}_h \\ \mathbb{I}_{cal} = \{\mathbb{I}_1, \mathbb{I}_2, ..., \mathbb{I}_{24}\},\ E_i \in I_h \end{gathered} \tag{5.6}$$

Thus, the boundary for predictions at the hour $h$ is as Equation 5.7.

$$C(X_{t+1}) = [\hat{\mu}(X_{t+1}) - Q(1-\alpha, E \in \mathbb{I}_h), \hat{\mu}(X_{t+1}) + Q(1-\alpha, E \in \mathbb{I}_h)] \tag{5.7}$$

**CQR Hourly** As discussed in Section 5.2, CQR does not necessitate homoskedastic data for the conformity score function (refer to Equation 5.3), considering the maximum distance between targets and the two boundaries. Consequently, the prediction interval of CQR is better suited to fit distributions of targets compared to conformal prediction. The prediction interval at the level $100(1-\alpha)\%$ is expressed in Equation 5.4.

In the context of day-ahead forecasting, CQR is designed to calibrate 24-hour predictions simultaneously. Consequently, the construction of CQR hourly mirrors the process of hourly conformal prediction. A distinct conformity score set is computed for each hour, utilizing historical conformity scores from the same hour. The prediction interval is independently constructed for each hour.

### 5.3.2 Adaptive Quantile Random Forest (AQRF)

According to SPCI framework, the calibration score is defined as Equation 5.8.

$$E_i = Y_i - \hat{\mu}(X_i), \text{ where } X_i, Y_i \in \mathbb{I}_h \tag{5.8}$$

Then the $100(1-\alpha)\%$ prediction interval can be expressed as Equation 5.9.

$$C(X_{t+1}) = [\hat{\mu}(X_{t+1}) + Q(\alpha/2, E \in \mathbb{I}_h), \hat{\mu}(X_{t+1}) + Q(1-\alpha/2, E \in \mathbb{I}_h)] \tag{5.9}$$

Utilizing the definition of hourly nonconformity scores as outlined in section **??**, the errors were tailored to each respective hour $h$.

**Quantile Random Forest (QRF)** The quantiles of calibration errors are calculated by the QRF algorithm. Input features of the QRF can be:

(1) historical errors $E_t, E_{t-1}, ....$

(2) historical errors $E_t, E_{t-1}, ...$ and PCA transformed features $F_{t+1}$.

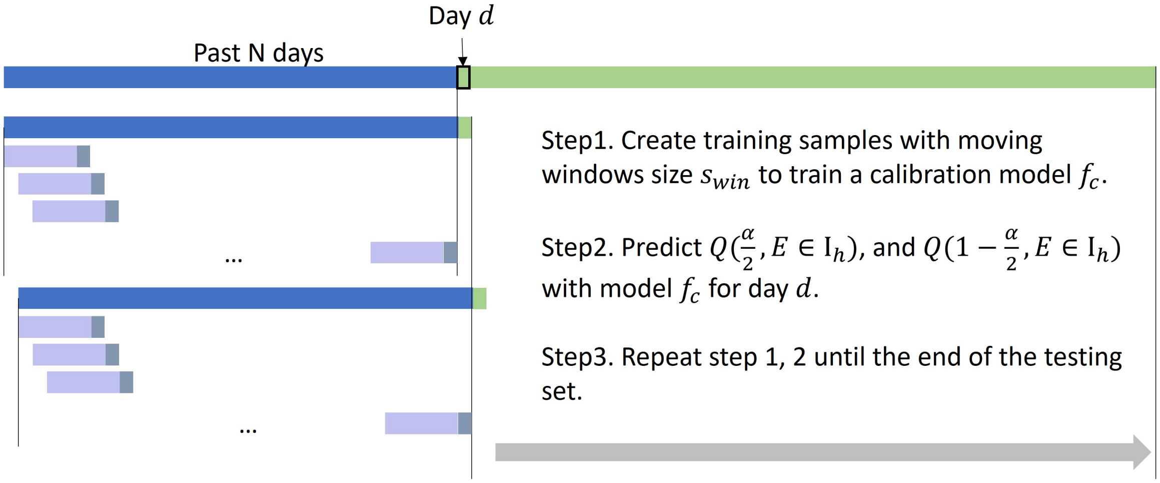
(3) historical errors $E_t, E_{t-1}, ...$ and encoded hour and weekday $F_{t+1}$.

(4) historical errors $E_t, E_{t-1}, ...$, PCA transformed features $F_{t+1}$ and encoded hour and weekday $H_{t+1}$.

(5) PCA transformed features $F_{t+1}$.

(6) encoded hour and weekday $H_{t+1}$.

To build a QRF model for forecasting error quantiles on a daily basis using historical errors, the sampling procedure is illustrated in Figure 5.1.



Step1. Create training samples with moving windows size $s_{win}$ to train a calibration model $f_c$.

Step2. Predict $Q(\frac{\alpha}{2}, E \in \mathrm{I}_h)$, and $Q(1-\frac{\alpha}{2}, E \in \mathrm{I}_h)$ with model $f_c$ for day $d$.

Step3. Repeat step 1, 2 until the end of the testing set.

Figure 5.1. Error Sampling Process for QRF-based Model.

In this chapter, $X_{t+1}$ is the input for predicting $y_t + 1$. $X_{t+1}$ in the probabilistic forecasting is in the dimension of $R^{144\times22}$ which is transformed to $X'_{t+1}$ by PCA to reduce the dimension of features. $H_{t+1}$ is the sin-cos encoded hour and weekday information of $y_t + 1$.

**Adaptively Adjust Uncertainty Estimation** In the QRF-based calibration model, $\alpha$ is set as a constant value for each prediction interval. However, a proposal by Gibbs [164] suggested adjusting $\alpha$ based on the most recent prediction to enhance the results of conformal prediction. Consequently, the adaptive conformal inference was incorporated into both models. For the prediction of $100(1-\alpha)\%$ prediction interval bounds, $\alpha_{t+1}$ will be modified in accordance with the value of $\alpha_t$ using Equation 5.10. The indicator function **1** is utilized, being equal to 1 if $y_t$ is not within the prediction interval $C(X_t)$, and 0 otherwise.

$$\alpha_{t+1} = \alpha_t + \gamma(\alpha - \mathbf{1}_{\{y_t \notin C(X_t)\}}) \tag{5.10}$$

The adaptive quantile random forest method introduced in this chapter will be denoted as AQRF.

### 5.3.3 Adaptive Conformal Residual Fitting (ACRF)

Split the calibration data set $\mathbb{D}_{cal}$ into two parts, a half as $\mathbb{D}_{cal1}$ and another half as $\mathbb{D}_{cal2}$. Then the ratio of actual errors to the predicted errors $r$ can be derived in the process described in Figure 5.2.

Utilizing calibration data $\mathbb{D}_{cal1}$, a random forest model $f_{err}$ is constructed, and this model $f_{err}$ is employed to forecast the errors of $\mathbb{D}_{cal2}$ as $\hat{E}_{cal2}$. The ratio of the true error $E_{cal2}$ to the predicted error $\hat{E}_{cal2}$ is represented as $r$. Subsequently, the $q$-th quantile of $r$, denoted as $e$, is utilized to compute the calibrated boundaries as specified in Equation 5.11.

As illustrated in Figure 5.2, the model $f_{err}$ can undergo training with various inputs to anticipate errors. In this context, features such as PCA-transformed features

$F$, historical errors $E$, encoded hour and weekday $H$, and their combinations were taken into account, as indicated in Table 5.5.3.

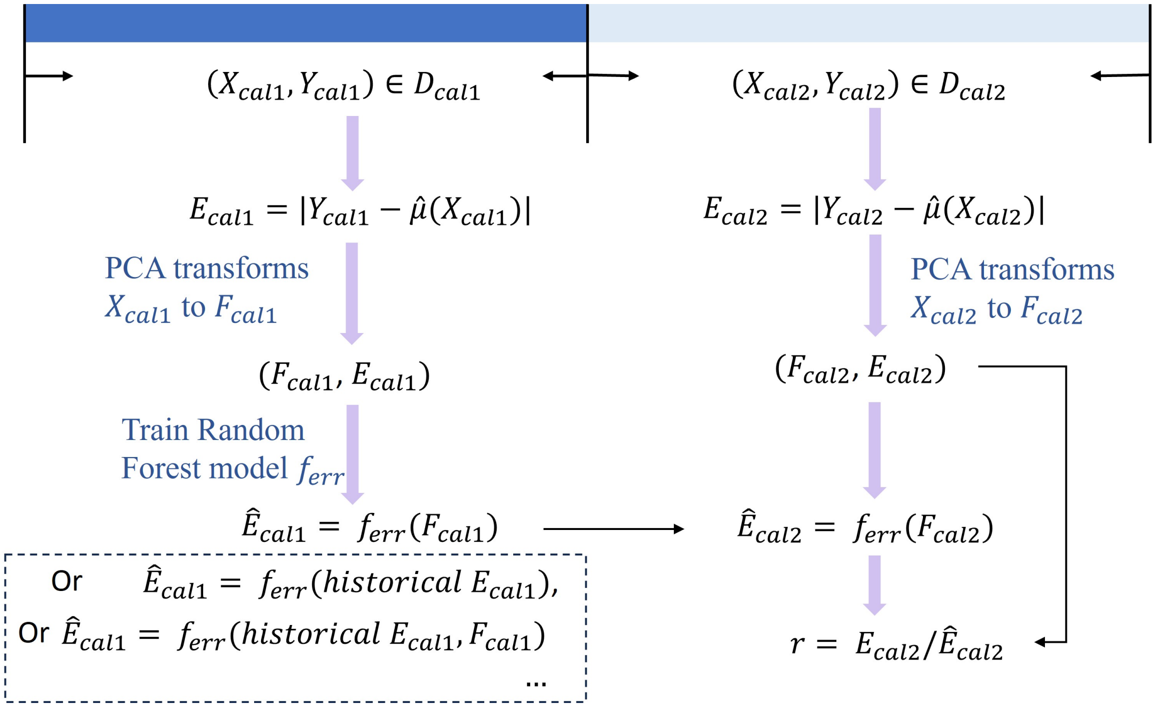

$(X_{cal1}, Y_{cal1}) \in D_{cal1}$
$(X_{cal2}, Y_{cal2}) \in D_{cal2}$
$E_{cal1} = |Y_{cal1} - \hat{\mu}(X_{cal1})|$
$E_{cal2} = |Y_{cal2} - \hat{\mu}(X_{cal2})|$
PCA transforms $X_{cal1}$ to $F_{cal1}$
PCA transforms $X_{cal2}$ to $F_{cal2}$
$(F_{cal1}, E_{cal1})$
$(F_{cal2}, E_{cal2})$
Train Random Forest model $f_{err}$
$\hat{E}_{cal1} = f_{err}(F_{cal1})$
$\hat{E}_{cal2} = f_{err}(F_{cal2})$
Or $\hat{E}_{cal1} = f_{err}(historical\ E_{cal1})$,
Or $\hat{E}_{cal1} = f_{err}(historical\ E_{cal1}, F_{cal1})$
...
$r = E_{cal2}/\hat{E}_{cal2}$

Figure 5.2. Flowchart of Calculating Conformal Residual Fitting Parameter $r$ For Each New Data Point.

Then the prediction interval for the data point $X_{t+1}$ is as in Equation 5.11, where $e$ is the empirical $q$th quantile of $r$.

$$C(X_{t+1}) = [\hat{\mu}(X_{t+1}) - e \cdot f_{err}(F_{t+1}), \hat{\mu}(X_{t+1}) + e \cdot f_{err}(F_{t+1})] \tag{5.11}$$

The $\alpha$ value in CRF updated with adaptive inference in Equation 5.10 will be abbreviated as ACRF.

## 5.4 Performance Metrics

**Reliability** Prediction Interval Coverage Probability (PICP) is employed to assess the likelihood that the actual target values fall within the prediction intervals provided by the predictions. As shown in the Equation 5.12, n denotes the number of samples, $\theta_i^\alpha$ is an indicator function and it equals 1 if the target $i$ is covered by the prediction interval, otherwise 0. Thus, the higher the PICP is, the more targets are covered by the prediction interval which means the prediction is more reliable.

$$PICP = \frac{1}{n}\sum_{i=1}^{n}\theta_i^\alpha \tag{5.12}$$

**Sharpness** Prediction Interval Average Width (PIAW) measures the average width of the prediction interval. Equation 5.13 shows the average of $n$ samples' bandwidth, where $U_i^\alpha$ and $L_i^\alpha$ represents the upper and lower bounds of $i$th prediction at $100(1-\alpha)\%$ level. When employing sharpness to assess prediction intervals, the goal is to minimize the value of PIAW. However, in the context of probabilistic forecasts, it is crucial to ensure that the uncertainties of the forecasts are adequately quantified. Therefore, it is essential to meet reliability requirements, and efforts should be made to minimize the width between the upper and lower bounds of the prediction interval.

$$PIAW = \frac{1}{n}\sum_{i=1}^{n}(U_i^\alpha - L_i^\alpha) \tag{5.13}$$

## 5.5 Experiments

Conformal predictions were employed to calibrate the probabilistic forecasting outcomes produced by the composite QR-Transformer model in Chapter 4.

### 5.5.1 Dataset

As this chapter builds upon the findings of Chapter 4, the same dataset is utilized. This dataset encompasses predictors presented as multivariate time series, covering information on weather, time, load, and electricity prices, while the labels correspond to day-ahead electricity prices. The probabilistic model underwent training using data from the years 2017 to 2020. The testing set encompasses the last 245 days of 2021, while the initial approximately 100 days are employed for constructing conformal prediction models.

### 5.5.2 Prediction Performance of Benchmark Conformal Prediction Models

As illustrated in Table 5.5.2 and 5.5.2, the 80% prediction interval of original probabilistic forecasting from the composite QR-Transformer only covered 63.9% of observations and the bandwidth is 14.927. Similar low coverage rates occur in other prediction intervals as well. The conformal prediction rows in Table 5.5.2 and 5.5.2 refer to conformal prediction with nonconformity scores. Hourly conformal prediction refers to the conformal prediction with nonconformity scores with hourly calibration data in Section **??**. CQR denotes conformal quantile regression. N refers to the length of the calibration dataset.

According to the results shown in Table 5.5.2 and 5.5.2, calibration with errors collected in the past 7 days are long enough to achieve decent coverage rates and bandwidth for each PI which are similar to using errors in the past 35 or 112 days. So, the following results are based on the moving window size 7. The benchmark methods have already achieved coverage rates close to the expected confidence level.

### 5.5.3 Prediction Performance of AQRF Prediction Method

Various input configurations were experimented with for the quantile random forest-based method, and the corresponding outcomes are presented in Table 5.5.3. As indicated in the table, the coverage rates, especially for 80% prediction intervals, do not reach the levels observed with benchmark methods, although they surpass

Table 5.2. Benchmark: PICP for Prediction Intervals Through Conformal Prediction With Non-Conformity Score and Conformal Quantile Regression

| **Caliberation Method** | **N** | **80% PI** | **60% PI** | **40% PI** | **20% PI** |
|---|---|---|---|---|---|
| Conformal prediction | 7 | 0.782 | 0.597 | 0.422 | 0.226 |
| Hourly conformal prediction | 7 | **0.800** | 0.638 | 0.482 | 0.334 |
| CQR | 7 | 0.791 | 0.610 | 0.425 | 0.226 |
| CQR hourly | 7 | 0.801 | 0.634 | 0.483 | 0.335 |
| Conformal prediction | 35 | 0.788 | 0.586 | 0.397 | 0.204 |
| Hourly conformal prediction | 35 | 0.786 | **0.600** | 0.414 | 0.226 |
| CQR | 35 | 0.790 | **0.600** | **0.402** | **0.199** |
| CQR hourly | 35 | 0.798 | 0.599 | 0.411 | 0.220 |
| Conformal prediction | 112 | 0.757 | 0.549 | 0.359 | 0.182 |
| Hourly conformal prediction | 112 | 0.752 | 0.556 | 0.370 | 0.190 |
| CQR | 112 | 0.785 | 0.590 | 0.382 | 0.180 |
| CQR hourly | 112 | 0.793 | 0.576 | 0.389 | 0.183 |
| Before calibration |  | **0.639** | **0.476** | **0.315** | **0.139** |

the coverage rate before calibration. When incorporating error terms $E(t), E(t-1), \ldots, E(t-s)$ and transformed features $Ft$, the QRF method exhibits the highest PICP. Furthermore, the PICP with only $Ft$ as inputs is comparable to the best-performing model, suggesting that the transformed features alone adequately capture the variance in errors.

Here are two demonstrations displaying the 80% prediction interval bounds calibrated by QRF for two periods in 2021. As depicted in Figure 5.3, the QRF-calibrated bounds effectively encompass the majority of targets in the absence of spikes. In the second demonstration Figure 5.4, while the upper bound of the prediction interval increases following spikes, the adjustment occurs several days later and is not prompt.

Upon integration with adaptive conformal inference, the QRF method attains coverage rates that are comparable to benchmark methods, albeit with a narrower

Table 5.3. Benchmark: PIAW for Prediction Intervals Through Conformal Prediction With Non-Conformity Score and Conformal Quantile Regression

| **Caliberation Method** | **N** | **80% PI** | **60% PI** | **40% PI** | **20% PI** |
|---|---|---|---|---|---|
| Conformal prediction | 7 | 21.943 | 15.197 | 10.644 | 6.284 |
| Hourly conformal prediction | 7 | **24.934** | 17.312 | 12.717 | 8.744 |
| CQR | 7 | 22.453 | 15.141 | 10.224 | 5.864 |
| CQR hourly | 7 | 24.318 | 16.625 | 12.305 | 8.423 |
| Conformal prediction | 35 | 22.446 | 15.150 | 10.148 | 5.537 |
| Hourly conformal prediction | 35 | 23.047 | **16.048** | 10.945 | 6.198 |
| CQR | 35 | 22.525 | **14.715** | **9.746** | **5.282** |
| CQR hourly | 35 | 22.355 | 15.468 | 10.467 | 5.934 |
| Conformal prediction | 112 | 23.647 | 13.809 | 8.286 | 4.000 |
| Hourly conformal prediction | 112 | 23.796 | 14.328 | 8.683 | 4.235 |
| CQR | 112 | 23.325 | 13.872 | 8.690 | 4.158 |
| CQR hourly | 112 | 23.271 | 14.302 | 8.950 | 4.280 |
| Before calibration | | **14.927** | **8.821** | **5.125** | **2.388** |

bandwidth, as illustrated in Table 5.5. This implies that the AQRF method more precisely enhances the coverage rate of probabilistic forecasting.

As the calibration of AQRF was performed on an hourly basis, the alpha values were adjusted hourly in response. The alpha updates for 6 AM, 12 PM, 18 PM, and 0 AM over the prediction days are illustrated in Figure 5.5 below. In accordance with Equation 5.10, the alpha values decrease when the previous target lies outside the prediction interval and increase when the previous target falls within the prediction interval.

Figure 5.6 illustrates the contrast between the boundaries of QRF and AQRF. As depicted in the figure, when the targets extend beyond the QRF boundaries (indicated by the red dashed line and green dashed line), the adjusted AQRF boundaries (represented by the red solid line and green solid line) expand. The AQRF boundaries effectively adapt to cover a higher percentage of targets.

Table 5.4. Enhancing Precision: PICP and Bandwidth For 80% Prediction Interval Through Quantile Random Forest Based Method

| **Inputs** | **Settings** | **PICP** | **PIAW** |
|---|---|---|---|
| $E_t,\ E_{t-1},\ \ldots\ ,\ E_{t-s}$ | 24 models for 24 hours | 0.696 | 17.527 |
| $E_t,\ E_{t-1},\ \ldots\ ,\ E_{t-s},\ H_{t+1}$ | 1 model for 24 hours | 0.737 | 17.899 |
| $\boldsymbol{E_t,\ E_{t-1},\ \ldots,\ E_{t-s},\ F_{t+1}}$ | 1 model for 24 hours | **0.755** | **16.280** |
| $E_t,\ E_{t-1},\ \ldots\ ,\ E_{t-s}, H_{t+1},\ F_{t+1}$ | 1 model for 24 hours | 0.753 | 16.946 |
| $H_{t+1},\ F_{t+1}$ | 1 model for 24 hours | 0.725 | 15.594 |
| $H_{t+1}$ | 1 model for 24 hours | 0.729 | 19.740 |
| $F_{t+1}$ | 1 model for 24 hours | **0.747** | **15.614** |

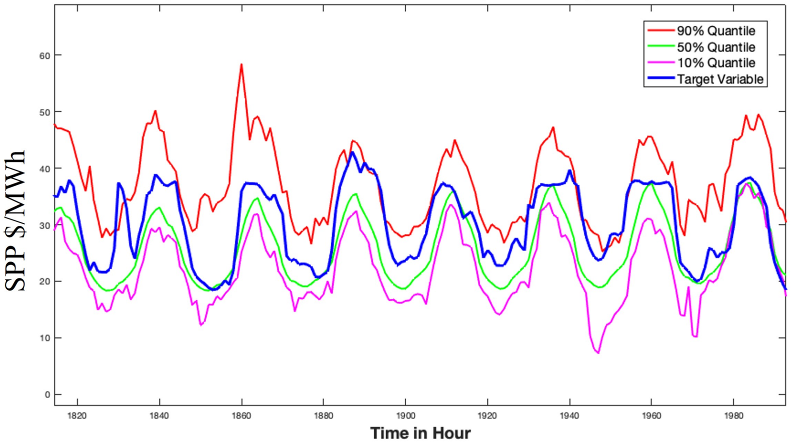


Figure 5.3. Day-ahead Quantile Prediction of Electricity Prices Using QRF Method Demo 1.

### 5.5.4 Prediction Performance of ACRF Prediction Method

Conformal Residual Fitting (CRF) involves several decisions in terms of settings, such as the size of two datasets $\mathbb{D}_{cal1}$ and $\mathbb{D}_{cal2}$, and specifying the number of components in PCA transformation. Preliminary experiments were conducted to identify the optimal settings for CRF, as outlined in Table 5.6. The table demonstrates that the best settings, based on PICP and PIAW, involve 70 samples in $\mathbb{D}_{cal1}$, 30 samples in $\mathbb{D}_{cal1}$, and 20 components (explaining 90.5% variance) in PCA transformation. Subsequent CRF experiments utilized these settings accordingly.

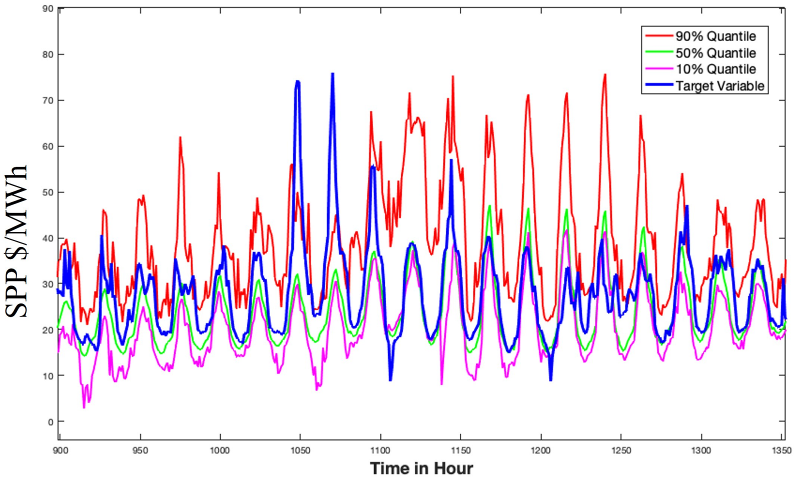


Figure 5.4. Day-ahead Quantile Prediction of Electricity Prices Using QRF Method Demo 2.

Table 5.7 presents the outcomes of conformal residual fitting, while Table 5.8 displays the outcomes of adaptive conformal residual fitting. Upon comparing the two tables, it becomes evident that the results are remarkably similar, and there are no substantial improvements observed after integrating adaptive conformal inference. One plausible explanation for this observation is that the quantiles of the CRF method were defined based on the empirical quantiles of the parameter $r$. Consequently, updating the $\alpha$ may not exert a significant influence on the prediction interval, as is the case in the QRF method where quantiles are directly modeled with $\alpha$.

Table 5.5. PICP and PIAW of AQRF Calibration

| **Inputs** | $\gamma$ | **PICP** | **PIAW** |
|---|---|---|---|
| $E_t$, $E_{t-1}$, …, $E_{t-s}$, $F_{t+1}$ | 0.001 | 0.719 | 16.110 |
| $E_t$, $E_{t-1}$, …, $E_{t-s}$, $F_{t+1}$ | 0.005 | 0.747 | 17.298 |
| $E_t$, $E_{t-1}$, …, $E_{t-s}$, $F_{t+1}$ | 0.010 | 0.764 | 18.446 |
| $E_t$, $E_{t-1}$, …, $E_{t-s}$, $F_{t+1}$ | 0.020 | 0.781 | 19.525 |
| $E_t$, $E_{t-1}$, …, $E_{t-s}$, $F_{t+1}$ | 0.030 | 0.788 | 19.785 |
| $E_t$, $E_{t-1}$, …, $E_{t-s}$, $F_{t+1}$ | **0.040** | **0.791** | **19.809** |
| $E_t$, $E_{t-1}$, …, $E_{t-s}$, $F_{t+1}$ | 0.050 | 0.789 | 19.631 |
| $F_{t+1}$ | 0.030 | 0.788 | 19.597 |
| $F_{t+1}$ | **0.040** | **0.790** | **19.724** |
| $F_{t+1}$ | 0.050 | 0.789 | 19.549 |

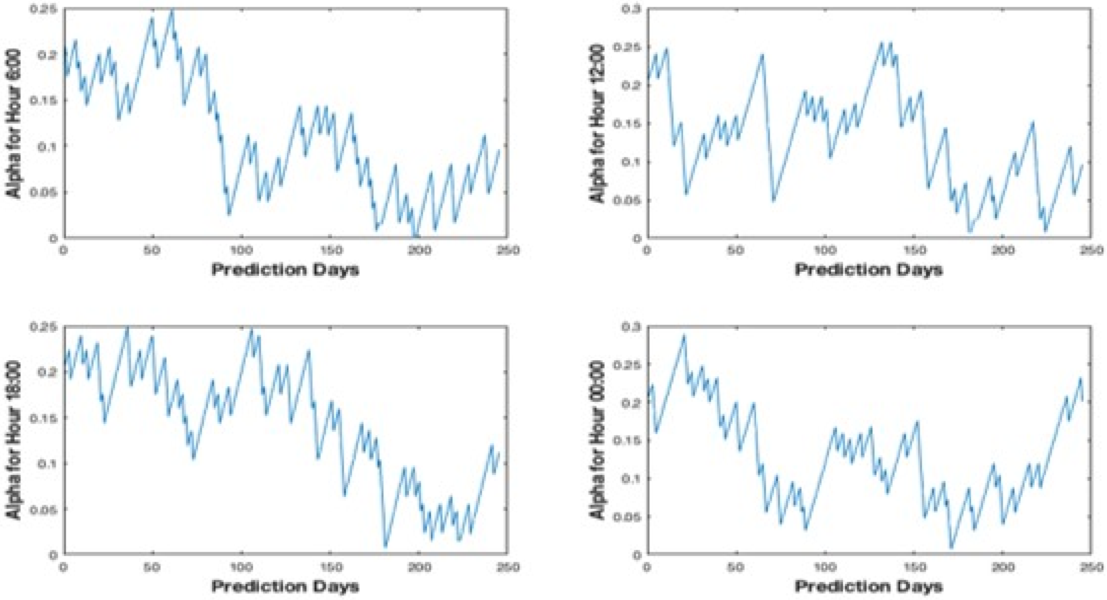


Figure 5.5. Adaptive $\alpha$ Values of AQRF.

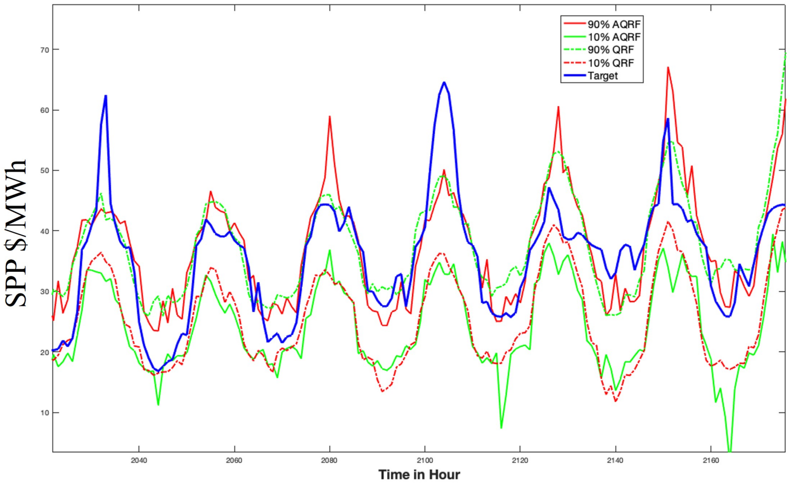


Figure 5.6. Comparisons of AQRF and QRF Bounds..

Table 5.6. PICP and PIAW of Conformal Residual Fitting under Different Settings

| **PI** | **Samples in Dcal1** | **Samples in Dcal2** | **PCA n_components** | **Variance Explanation** | **PICP** | **PIAW** |
|---|---|---|---|---|---|---|
| 80% | 50 | 50 | 20 | 0.922 | 0.809 | 25.436 |
| 60% | 50 | 50 | 20 | 0.922 | 0.614 | 17.082 |
| 40% | 50 | 50 | 20 | 0.922 | 0.430 | 11.422 |
| 20% | 50 | 50 | 20 | 0.922 | 0.226 | 6.221 |
| 80% | 50 | 50 | 30 | 0.960 | 0.807 | 25.119 |
| 60% | 50 | 50 | 30 | 0.960 | 0.616 | 16.993 |
| 40% | 50 | 50 | 30 | 0.960 | 0.431 | 11.396 |
| 20% | 50 | 50 | 30 | 0.960 | 0.231 | 6.233 |
| 80% | 50 | 50 | 40 | 0.985 | 0.757 | 25.294 |
| 60% | 50 | 50 | 40 | 0.985 | 0.549 | 13.787 |
| 40% | 50 | 50 | 40 | 0.985 | 0.346 | 7.429 |
| 20% | 50 | 50 | 40 | 0.985 | 0.163 | 3.445 |
| 80% | 70 | 30 | 20 | 0.905 | 0.792 | 24.212 |
| 60% | 70 | 30 | 20 | 0.905 | 0.609 | 16.616 |
| 40% | 70 | 30 | 20 | 0.905 | 0.425 | 11.306 |
| 20% | 70 | 30 | 20 | 0.905 | 0.229 | 6.458 |

Table 5.7. PICP and PIAW of Conformal Residual Fitting

| **PI** | **PICP** | **PIAW** |
|---|---|---|
| 80% | 0.792 | 24.212 |
| 60% | 0.609 | 16.616 |
| 40% | 0.425 | 11.306 |
| 20% | 0.229 | 6.458 |

Table 5.8. PICP and PIAW of Adaptive Conformal Residual Fitting

| **Quantiles** | **PICP** | **PIAW** |
|---|---|---|
| 80% | 0.802 | 25.355 |
| 60% | 0.608 | 16.680 |
| 40% | 0.410 | 11.138 |
| 20% | 0.207 | 6.140 |

## 5.6 Conclusion

In this chapter, the effectiveness of conformal prediction in enhancing the coverage rate of probabilistic forecasts generated by deep learning methods is demonstrated. The proposed Adaptive CRF (ACRF) conformal prediction method is shown to achieve the target coverage rate, even though the uncertainty bandwidth is not minimized.

The utilization of PCA-transformed multivariate time series features proves effective in modeling prediction errors and quantifying prediction uncertainties. Notably, historical prediction errors are found to be insignificant in predicting future errors. Additionally, the proposed AQRF conformal prediction method ensures the target coverage rate with the smallest bandwidth compared to existing methods.

The chapter introduces AQRF as a means of calibrating the boundaries of prediction intervals derived from probabilistic forecasting. Through comparative analysis with benchmark methods, specifically conformal prediction with nonconformity scores and conformal quantile regression, AQRF demonstrates a coverage rate comparable to that of benchmark models, but with reduced bandwidth. This method serves as a general approach applicable on top of any existing predictive models, providing robust and accurate uncertainty quantifications with desired prediction intervals at specified probabilities.

# CHAPTER 6

# CONCLUSIONS AND FUTURE WORK

In addressing the challenges associated with sudden spikes in electricity prices and the complexities of modeling multivariate time series data, a transformer model employing a two-step training process was applied in this manuscript. This design effectively manages the learning and forecasting of patterns in multivariate time series data, with the inclusion of a pretraining step enhancing model convergence and mitigating overfitting concerns.

The case study, utilizing ERCOT market data, showcased the superiority of the TDEPF model over other widely used models such as LSTM, RNN, and GRU in day-ahead electricity price forecasting. Experimental findings underscored the efficacy of the proposed spike transformation approach in mitigating price spikes, while also demonstrating the model's proficiency in capturing predictive patterns.

The forecasting model holds significant potential to assist market participants in making more informed decisions regarding bidding on day-ahead electricity prices, contributing to a more efficient and stable deregulated market.

Additionally, a comparison was conducted between probabilistic forecasting methods based on quantile regression and Gaussian processes. Adapting the two-step training structure proposed in Chapter 3, the second step was adjusted to predict different quantiles, replacing mean squared error (mse) loss with pinball loss for electricity prices.

Comparisons of prediction intervals from various models revealed that QR-Transformer consistently demonstrated stable and accurate performance. However, potential overfitting issues observed in RNN-type models persisted in probabilistic forecasting, evident in narrow deviated intervals.

The effectiveness of GP-Transformer in covering a majority of targets was noted, but the reliability of probabilistic predictions was compromised by wider bands compared to quantile regression models. The composite structure of quantile regression models, considering the loss of all quantiles simultaneously, resulted in milder peaks compared to separate QR-Transformer models. Although modifications to the Pinball loss narrowed prediction bands, balancing the coverage rate and bandwidth of prediction intervals remains a promising challenge.

Concluding this study, the introduction of an adaptive QRF-based method for calibrating prediction interval boundaries was presented. Comparative analysis with benchmark methods, specifically conformal prediction with nonconformity scores and conformal quantile regression, confirmed that the adaptive QRF-based method ensures a comparable coverage rate with reduced bandwidth. Additionally, it was demonstrated that PCA-transformed features, without past errors, effectively account for prediction errors.

In the future, with the availability of more extensive electricity and weather data, there is potential for enhancing forecasting accuracy through the implementation of a cluster-based model. This approach becomes particularly beneficial due to the seasonal variations in the pattern of electricity prices. It is worth noting that in this study, the consideration was limited to weather and load information. In reality, electricity prices are significantly influenced by the transmission of electricity, and disruptions in the operation of a generator can lead to spikes in prices. Integrating transmission-related data or conducting simulations to incorporate these factors into the forecasting process has the potential to further improve prediction results. A similar forecasting framework could also be extended to load forecasting. If predictions of day-ahead load are included as factors in the regression, there is an opportunity to enhance the forecasting results. This study showed promising experimental results of the proposed AQRF method. Further theoretical analysis will be performed to prove the convergence and conditional coverage of the proposed AQRF method.

# REFERENCES

[1] ERCOT, Ercot maps, `https://www.ercot.com/news/mediakit/maps`, accessed: 2023-02-08 (2023).

[2] U.S. Energy Information Administration, State profiles and energy estimates, `https://www.eia.gov/state/`, accessed: 2022-04-18 (2019).

[3] U.S. Energy Information Administration, Profile analysis, `https://www.eia.gov/state/analysis.php?sid=TX#5`, accessed: 2022-04-18 (2021).

[4] S. Shayegh, D. L. Sanchez, Impact of market design on cost-effectiveness of renewable portfolio standards, Renewable and Sustainable Energy Reviews 136 (2021) 110397.

[5] A. Mahmoud, A. Mohammed, A survey on deep learning for time-series forecasting, Machine learning and big data analytics paradigms: analysis, applications and challenges (2021) 365–392.

[6] G. C. Reinsel, Elements of multivariate time series analysis, Springer Science & Business Media, 2003.

[7] J. DeVilbiss and M. T. Brown, Wind was second-largest source of u.s. electricity generation on march 29, `https://www.eia.gov/todayinenergy/detail.php?id=52038`, accessed: 2022-04-13 (2022).

[8] R. G. Kavasseri, K. Seetharaman, Day-ahead wind speed forecasting using f-arima models, Renewable Energy 34 (5) (2009) 1388–1393.

[9] J. Zhang, Z. Tan, S. Yang, Day-ahead electricity price forecasting by a new hybrid method, Computers & Industrial Engineering 63 (3) (2012) 695–701.

[10] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, A. Lendasse, Methodology for long-term prediction of time series, Neurocomputing 70 (16-18) (2007) 2861–2869.

[11] S. B. Taieb, G. Bontempi, A. F. Atiya, A. Sorjamaa, A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition, Expert systems with applications 39 (8) (2012) 7067–7083.

[12] J. G. De Gooijer, R. J. Hyndman, 25 years of time series forecasting, International journal of forecasting 22 (3) (2006) 443–473.

[13] R. Weron, Electricity price forecasting: A review of the state-of-the-art with a look into the future, International journal of forecasting 30 (4) (2014) 1030–1081.

[14] J. Nowotarski, R. Weron, Recent advances in electricity price forecasting: A review of probabilistic forecasting, Renewable and Sustainable Energy Reviews 81 (2018) 1548–1568.

[15] R. K. Agrawal, F. Muchahary, M. M. Tripathi, Ensemble of relevance vector machines and boosted trees for electricity price forecasting, Applied Energy 250 (2019) 540–548.

[16] G. Díaz, J. Coto, J. Gómez-Aleixandre, Prediction and explanation of the formation of the spanish day-ahead electricity price through machine learning regression, Applied Energy 239 (2019) 610–625.

[17] M. Zahid, F. Ahmed, N. Javaid, R. A. Abbasi, H. S. Zainab Kazmi, A. Javaid, M. Bilal, M. Akbar, M. Ilahi, Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids, Electronics 8 (2) (2019) 122.

[18] H. Jahangir, H. Tayarani, S. Baghali, A. Ahmadian, A. Elkamel, M. A. Golkar, M. Castilla, A novel electricity price forecasting approach based on dimension

reduction strategy and rough artificial neural networks, IEEE Transactions on Industrial Informatics 16 (4) (2019) 2369–2381.

[19] J. Xu, R. Baldick, Day-ahead price forecasting in ercot market using neural network approaches, in: Proceedings of the Tenth ACM International Conference on Future Energy Systems, 2019, pp. 486–491.

[20] Z. Chang, Y. Zhang, W. Chen, Electricity price prediction based on hybrid model of adam optimized lstm neural network and wavelet transform, Energy 187 (2019) 115804.

[21] S. Voronin, J. Partanen, Price forecasting in the day-ahead energy market by an iterative method with separate normal price and price spike frameworks, Energies 6 (11) (2013) 5897–5920.

[22] H. Jiang, B. Kim, M. Guan, M. Gupta, To trust or not to trust a classifier, Advances in neural information processing systems 31.

[23] T. J. Sullivan, Introduction to uncertainty quantification, Vol. 63, Springer, 2015.

[24] A. N. Angelopoulos, S. Bates, A gentle introduction to conformal prediction and distribution-free uncertainty quantification, arXiv preprint arXiv:2107.07511.

[25] M. Fontana, G. Zeni, S. Vantini, Conformal prediction: a unified review of theory and new challenges, arXiv preprint arXiv:2005.07972.

[26] Y. Romano, E. Patterson, E. Candes, Conformalized quantile regression, Advances in neural information processing systems 32.

[27] M. Zaffran, O. Féron, Y. Goude, J. Josse, A. Dieuleveut, Adaptive conformal predictions for time series, in: International Conference on Machine Learning, PMLR, 2022, pp. 25834–25866.

[28] C. Deb, F. Zhang, J. Yang, S. E. Lee, K. W. Shah, A review on time series forecasting techniques for building energy consumption, Renewable and Sustainable Energy Reviews 74 (2017) 902–924.

[29] R. S. Tsay, Multivariate time series analysis: with R and financial applications, John Wiley & Sons, 2013.

[30] M. B. Shrestha, G. R. Bhatta, Selecting appropriate methodological framework for time series data analysis, The Journal of Finance and Data Science 4 (2) (2018) 71–89.

[31] E. Zivot, J. Wang, Vector autoregressive models for multivariate time series, Modeling financial time series with S-PLUS® (2006) 385–429.

[32] C. Chatfield, Time-series forecasting, Chapman and Hall/CRC, 2000.

[33] J. D. Cryer, K.-S. Chan, Time series analysis: with applications in R, Vol. 2, Springer, 2008.

[34] G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, Time series analysis: forecasting and control, John Wiley & Sons, 2015.

[35] L.-M. Liu, Identification of seasonal arima models using a filtering method, Communications in Statistics-Theory and Methods 18 (6) (1989) 2279–2288.

[36] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B (Methodological) 58 (1) (1996) 267–288.

[37] N.-J. Hsu, H.-L. Hung, Y.-M. Chang, Subset selection for vector autoregressive processes using lasso, Computational Statistics & Data Analysis 52 (7) (2008) 3645–3657.

[38] A. Shojaie, G. Michailidis, Discovering graphical granger causality using the truncating lasso penalty, Bioinformatics 26 (18) (2010) i517–i523.

[39] R. A. Davis, P. Zang, T. Zheng, Sparse vector autoregressive modeling, Journal of Computational and Graphical Statistics 25 (4) (2016) 1077–1096.

[40] T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, Journal of econometrics 31 (3) (1986) 307–327.

[41] L. Bauwens, S. Laurent, J. V. Rombouts, Multivariate garch models: a survey, Journal of applied econometrics 21 (1) (2006) 79–109.

[42] A. Silvennoinen, T. Teräsvirta, Multivariate garch models, in: Handbook of financial time series, Springer, 2009, pp. 201–229.

[43] G. R. Newsham, B. J. Birt, Building-level occupancy data to improve arima-based electricity use forecasts, in: Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building, 2010, pp. 13–18.

[44] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, V. Vapnik, Predicting time series with support vector machines, in: International conference on artificial neural networks, Springer, 1997, pp. 999–1004.

[45] Z. Han, J. Zhao, H. Leung, K. F. Ma, W. Wang, A review of deep learning models for time series prediction, IEEE Sensors Journal 21 (6) (2019) 7833–7848.

[46] D. C. Park, M. El-Sharkawi, R. Marks, L. Atlas, M. Damborg, Electric load forecasting using an artificial neural network, IEEE transactions on Power Systems 6 (2) (1991) 442–449.

[47] H. S. Hippert, C. E. Pedreira, R. C. Souza, Neural networks for short-term load forecasting: A review and evaluation, IEEE Transactions on power systems 16 (1) (2001) 44–55.

[48] A. Tealab, Time series forecasting using artificial neural networks methodologies: A systematic review, Future Computing and Informatics Journal 3 (2) (2018) 334–340.

[49] J. Wang, J. Wang, W. Fang, H. Niu, Financial time series prediction using elman recurrent random neural networks, Computational intelligence and neuroscience 2016.

[50] M. Hüsken, P. Stagge, Recurrent neural networks for time series classification, Neurocomputing 50 (2003) 223–235.

[51] J. Brownlee, Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python, Machine Learning Mastery, 2018.

[52] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: International conference on machine learning, PMLR, 2013, pp. 1310–1318.

[53] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[54] H. Sak, A. Senior, F. Beaufays, Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, arXiv preprint arXiv:1402.1128.

[55] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional lstm and other neural network architectures, Neural networks 18 (5-6) (2005) 602–610.

[56] A. Sagheer, M. Kotb, Time series forecasting of petroleum production using deep lstm recurrent networks, Neurocomputing 323 (2019) 203–213.

[57] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, arXiv preprint arXiv:1409.1259.

[58] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555.

[59] H. Y. Kim, C. H. Won, Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models, Expert Systems with Applications 103 (2018) 25–37.

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

[61] T. Hollis, A. Viscardi, S. E. Yi, A comparison of lstms and attention mechanisms for forecasting financial time series, arXiv preprint arXiv:1812.07699.

[62] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473.

[63] S. Du, T. Li, Y. Yang, S.-J. Horng, Multivariate time series forecasting via attention-based encoder–decoder framework, Neurocomputing 388 (2020) 269–279.

[64] C. Fan, Y. Zhang, Y. Pan, X. Li, C. Zhang, R. Yuan, D. Wu, W. Wang, J. Pei, H. Huang, Multi-horizon time series forecasting with temporal attention learning, in: Proceedings of the 25th ACM SIGKDD International conference on knowledge discovery & data mining, 2019, pp. 2527–2535.

[65] S. Shun-Yao, S. Fan-Keng, L. Hung-yi, Temporal pattern attention for multivariate time series forecasting [j], Machine Learning 108 (8-9).

[66] S. Huang, D. Wang, X. Wu, A. Tang, Dsanet: Dual self-attention network for multivariate time series forecasting, in: Proceedings of the 28th ACM international conference on information and knowledge management, 2019, pp. 2129–2132.

[67] R. Mohammadi Farsani, E. Pazouki, A transformer self-attention model for time series forecasting, Journal of Electrical and Computer Engineering Innovations (JECEI) 9 (1) (2020) 1–10.

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

[69] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 35, 2021, pp. 11106–11115.

[70] R. He, A. Ravula, B. Kanagal, J. Ainslie, Realformer: Transformer likes residual attention, arXiv preprint arXiv:2012.11747.

[71] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, Advances in Neural Information Processing Systems 34 (2021) 22419–22430.

[72] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, S. Dustdar, Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting, in: International conference on learning representations, 2021.

[73] A. Khosravi, S. Nahavandi, D. Creighton, A. F. Atiya, Comprehensive review of neural network-based prediction intervals and new advances, IEEE Transactions on neural networks 22 (9) (2011) 1341–1356.

[74] J. Nowotarski, R. Weron, Computing electricity spot price prediction intervals using quantile regression and forecast averaging, Computational Statistics 30 (2015) 791–803.

[75] R. Weron, A. Misiorek, Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models, International journal of forecasting 24 (4) (2008) 744–763.

[76] R. Weron, Modeling and forecasting electricity loads and prices: A statistical approach, John Wiley & Sons, 2007.

[77] A. Misiorek, S. Trueck, R. Weron, Point and interval forecasting of spot electricity prices: Linear vs. non-linear time series models, Studies in Nonlinear Dynamics & Econometrics 10 (3).

[78] M. Bozorg, A. Bracale, P. Caramia, G. Carpinelli, M. Carpita, P. De Falco, Bayesian bootstrap quantile regression for probabilistic photovoltaic power forecasting, Protection and Control of Modern Power Systems 5 (1) (2020) 1–12.

[79] Q. Li, N. Lin, R. Xi, Bayesian regularized quantile regression.

[80] A. Panagiotelis, M. Smith, Bayesian density forecasting of intraday electricity prices using multivariate skew t distributions, International Journal of Forecasting 24 (4) (2008) 710–727.

[81] R. A. Stine, Bootstrap prediction intervals for regression, Journal of the American Statistical Association 80 (392) (1985) 1026–1031.

[82] M. P. Clements, J. H. Kim, Bootstrap prediction intervals for autoregressive time series, Computational statistics & data analysis 51 (7) (2007) 3580–3594.

[83] B. Efron, R. J. Tibshirani, An introduction to the bootstrap, CRC press, 1994.

[84] F. Jareño, R. Ferrer, S. Miroslavova, Us stock market sensitivity to interest and inflation rates: a quantile regression approach, Applied Economics 48 (26) (2016) 2469–2481.

[85] B. Fitzenberger, R. Koenker, J. Machado, B. Melly, Economic applications of quantile regression 2.0, Empirical Economics 62 (1) (2022) 1–6.

[86] J. Gasthaus, K. Benidis, Y. Wang, S. S. Rangapuram, D. Salinas, V. Flunkert, T. Januschowski, Probabilistic forecasting with spline quantile function rnns, in: The 22nd international conference on artificial intelligence and statistics, PMLR, 2019, pp. 1901–1910.

[87] C. A. Sims, Macroeconomics and reality, Econometrica: journal of the Econometric Society (1980) 1–48.

[88] J. C. Chan, Large Bayesian vector autoregressions, Springer, 2020.

[89] J. Cimadomo, D. Giannone, M. Lenza, F. Monti, A. Sokol, Nowcasting with large bayesian vector autoregressions, Journal of Econometrics 231 (2) (2022) 500–519.

[90] J. Morley, B. Wong, Estimating and accounting for the output gap with large bayesian vector autoregressions, Journal of Applied Econometrics 35 (1) (2020) 1–18.

[91] J. Durbin, S. J. Koopman, Time series analysis of non-gaussian observations based on state space models from both classical and bayesian perspectives, Journal of the Royal Statistical Society Series B: Statistical Methodology 62 (1) (2000) 3–56.

[92] M. C. Fung, G. W. Peters, P. V. Shevchenko, Cohort effects in mortality modelling: A bayesian state-space approach, Annals of Actuarial Science 13 (1) (2019) 109–144.

[93] Y. Yamashita, Y. Iwasaki, T. Matsubara, K. Suzuki, Y. Kanzawa, T. Okuda, K. Nishina, C. A. Strüssmann, Comparison of survival rates between domesticated and semi-native char using bayesian multi-variate state-space model, Fisheries Research 221 (2020) 105380.

[94] G. Pleiss, A Scalable and Flexible Framework for Gaussian Processes via Matrix-Vector Multiplication, Cornell University, 2020.

[95] J. Wang, An intuitive tutorial to gaussian processes regression, arXiv preprint arXiv:2009.10862.

[96] H. Wang, Y.-M. Zhang, J.-X. Mao, Sparse gaussian process regression for multistep ahead forecasting of wind gusts combining numerical weather predictions

and on-site measurements, Journal of Wind Engineering and Industrial Aerodynamics 220 (2022) 104873.

[97] A. Zeng, H. Ho, Y. Yu, Prediction of building electricity usage using gaussian process regression, Journal of Building Engineering 28 (2020) 101054.

[98] J. Hensman, A. Matthews, Z. Ghahramani, Scalable variational gaussian process classification, in: Artificial Intelligence and Statistics, PMLR, 2015, pp. 351–360.

[99] V. Adam, S. Eleftheriadis, A. Artemev, N. Durrande, J. Hensman, Doubly sparse variational gaussian processes, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2874–2884.

[100] D. Tran, R. Ranganath, D. M. Blei, The variational gaussian process, arXiv preprint arXiv:1511.06499.

[101] A. Patton, Copula methods for forecasting multivariate time series, Handbook of economic forecasting 2 (2013) 899–960.

[102] T. Nagler, D. Krüger, A. Min, Stationary vine copula models for multivariate time series, Journal of Econometrics 227 (2) (2022) 305–324.

[103] N. Nguyen, B. Quanz, Temporal latent auto-encoder: A method for probabilistic multivariate time series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 9117–9125.

[104] D. Kaur, S. N. Islam, M. A. Mahmud, A vae-based bayesian bidirectional lstm for renewable energy forecasting, arXiv preprint arXiv:2103.12969.

[105] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, R. J. Hyndman, Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond (2016).

[106] F. Ziel, B. Liu, Lasso estimation for gefcom2014 probabilistic electric load forecasting, International Journal of Forecasting 32 (3) (2016) 1029–1037.

[107] R. Wen, K. Torkkola, B. Narayanaswamy, D. Madeka, A multi-horizon quantile recurrent forecaster, arXiv preprint arXiv:1711.11053.

[108] R. L. Winkler, A decision-theoretic approach to interval estimation, Journal of the American Statistical Association 67 (337) (1972) 187–191.

[109] A. Brusaferri, M. Matteucci, P. Portolani, A. Vitali, Bayesian deep learning based method for probabilistic forecast of day-ahead electricity prices, Applied Energy 250 (2019) 1158–1175.

[110] J. E. Matheson, R. L. Winkler, Scoring rules for continuous probability distributions, Management science 22 (10) (1976) 1087–1096.

[111] M. B. Bjerregård, J. K. Møller, H. Madsen, An introduction to multivariate probabilistic forecast evaluation, Energy and AI 4 (2021) 100058.

[112] T. Gneiting, F. Balabdaoui, A. E. Raftery, Probabilistic forecasts, calibration and sharpness, Journal of the Royal Statistical Society Series B: Statistical Methodology 69 (2) (2007) 243–268.

[113] A. Khosravi, S. Nahavandi, D. Creighton, Construction of optimal prediction intervals for load forecasting problems, IEEE Transactions on Power Systems 25 (3) (2010) 1496–1503.

[114] H. Quan, D. Srinivasan, A. Khosravi, Particle swarm optimization for construction of neural network-based prediction intervals, Neurocomputing 127 (2014) 172–180.

[115] J. C. Cuaresma, J. Hlouskova, S. Kossmeier, M. Obersteiner, Forecasting electricity spot-prices using linear univariate time-series models, Applied Energy 77 (1) (2004) 87–106.

[116] S. K. Aggarwal, L. M. Saini, A. Kumar, Electricity price forecasting in deregulated markets: A review and evaluation, International Journal of Electrical Power & Energy Systems 31 (1) (2009) 13–22.

[117] S. Anbazhagan, N. Kumarappan, Day-ahead deregulated electricity market price forecasting using recurrent neural network, IEEE Systems Journal 7 (4) (2012) 866–872.

[118] F. Ziel, R. Weron, Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks, Energy Economics 70 (2018) 396–420.

[119] J. Lago, G. Marcjasz, B. De Schutter, R. Weron, Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark, Applied Energy 293 (2021) 116983.

[120] U.S. Energy Information Administration, Total electric power industry summary statistics, 2021 and 2020, `https://www.eia.gov/electricity/annual/html/epa_01_01.html`, accessed: 2022-12-05 (2022).

[121] U.S. Energy Information Administration, Ercot electricity prices vary more with changes in wind power than with electricity demand, `https://www.eia.gov/todayinenergy/detail.php?id=54159`, accessed: 2022-12-20 (2022).

[122] U. Ugurlu, I. Oksuz, O. Tas, Electricity price forecasting using recurrent neural networks, Energies 11 (5) (2018) 1255.

[123] T. Ulgen, G. Poyrazoglu, Predictor analysis for electricity price forecasting by multiple linear regression, in: 2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), IEEE, 2020, pp. 618–622.

[124] A. Mohamed, M. E. El-Hawary, Mid-term electricity price forecasting using svm, in: 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE, 2016, pp. 1–6.

[125] X. Zhang, J. Wang, Y. Gao, A hybrid short-term electricity price forecasting framework: Cuckoo search-based feature selection with singular spectrum analysis and svm, Energy Economics 81 (2019) 899–913.

[126] S. M. Lakew, M. Cettolo, M. Federico, A comparison of transformer and recurrent neural networks on multilingual neural machine translation, arXiv preprint arXiv:1806.06957.

[127] S. Liao, Z. Wang, Y. Luo, H. Liang, Locational marginal price forecasting using transformer-based deep learning network, in: 2021 40th Chinese Control Conference (CCC), IEEE, 2021, pp. 8457–8462.

[128] G. Zhang, C. Wei, C. Jing, Y. Wang, Short-term electrical load forecasting based on time augmented transformer, International Journal of Computational Intelligence Systems 15 (1) (2022) 1–11.

[129] J. Bottieau, Y. Wang, Z. De Grève, F. Vallée, J.-F. Toubeau, Interpretable transformer model for capturing regime switching effects of real-time electricity prices, IEEE Transactions on Power Systems.

[130] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, C. Eickhoff, A transformer-based framework for multivariate time series representation learning, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2114–2124.

[131] Z. Ma, H. Zhong, L. Xie, Q. Xia, C. Kang, Month ahead average daily electricity price profile forecasting based on a hybrid nonlinear regression and svm model: an ercot case study, Journal of Modern Power Systems and Clean Energy 6 (2) (2018) 281–291.

[132] K. Yamada, H. Mori, A deep learning technique for electricity price forecasting in consideration of spikes, in: TENCON 2021-2021 IEEE Region 10 Conference (TENCON), IEEE, 2021, pp. 744–749.

[133] K. Iwabuchi, K. Kato, D. Watari, I. Taniguchi, F. Catthoor, E. Shirazi, T. Onoye, Flexible electricity price forecasting by switching mother wavelets based on wavelet transform and long short-term memory, Energy and AI 10 (2022) 100192.

[134] Y. Zhou, F.-J. Chang, L.-C. Chang, I.-F. Kao, Y.-S. Wang, C.-C. Kang, Multi-output support vector machine for regional multi-step-ahead pm2. 5 forecasting, Science of the Total Environment 651 (2019) 230–240.

[135] W. Yang, S. Sun, Y. Hao, S. Wang, A novel machine learning-based electricity price forecasting model based on optimal model selection strategy, Energy 238 (2022) 121989.

[136] C.-J. Huang, Y. Shen, Y.-H. Chen, H.-C. Chen, A novel hybrid deep neural network model for short-term electricity price forecasting, International Journal of Energy Research 45 (2) (2021) 2511–2532.

[137] J. Zhang, Z. Tan, Y. Wei, An adaptive hybrid model for short term electricity price forecasting, Applied Energy 258 (2020) 114087.

[138] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805.

[139] C. Seger, An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing (2018).

[140] S. Falkner, A. Klein, F. Hutter, Bohb: Robust and efficient hyperparameter optimization at scale, in: International Conference on Machine Learning, PMLR, 2018, pp. 1437–1446.

[141] K. Wang, H. Wang, S. Li, Renewable quantile regression for streaming datasets, Knowledge-Based Systems 235 (2022) 107675.

[142] F. Liu, M. Umair, J. Gao, Assessing oil price volatility co-movement with stock market volatility through quantile regression approach, Resources Policy 81 (2023) 103375.

[143] M. Mohsin, H. Ullah, N. Iqbal, W. Iqbal, F. Taghizadeh-Hesary, How external debt led to economic growth in south asia: A policy perspective analysis from quantile regression, Economic Analysis and Policy 72 (2021) 423–437.

[144] A. Demir, V. Pesqué-Cela, Y. Altunbas, V. Murinde, Fintech, financial inclusion and income inequality: a quantile regression approach, The European Journal of Finance 28 (1) (2022) 86–107.

[145] J. Á. G. Ordiano, L. Gröll, R. Mikut, V. Hagenmeyer, Probabilistic energy forecasting using the nearest neighbors quantile filter and quantile regression, International journal of forecasting 36 (2) (2020) 310–323.

[146] R. Nepal, H. O. Musibau, T. Jamasb, Energy consumption as an indicator of energy efficiency and emissions in the european union: A gmm based quantile regression approach, Energy Policy 158 (2021) 112572.

[147] P. Orbanz, Y. W. Teh, Bayesian nonparametric models., Encyclopedia of machine learning 1 (2010) 81–89.

[148] S. J. Gershman, D. M. Blei, A tutorial on bayesian nonparametric models, Journal of Mathematical Psychology 56 (1) (2012) 1–12.

[149] H. Zou, M. Yuan, Composite quantile regression and the oracle model selection theory.

[150] Q. Xu, K. Deng, C. Jiang, F. Sun, X. Huang, Composite quantile regression neural network with applications, Expert Systems with Applications 76 (2017) 129–139.

[151] K. Hatalis, A. J. Lamadrid, K. Scheinberg, S. Kishore, A novel smoothed loss and penalty function for noncrossing composite quantile estimation via deep neural networks, arXiv preprint arXiv:1909.12122.

[152] S. Lu, Q. Xu, C. Jiang, Y. Liu, A. Kusiak, Probabilistic load forecasting with a non-crossing sparse-group lasso-quantile regression deep neural network, Energy 242 (2022) 122955.

[153] T. Narayan, S. Wang, K. Canini, M. Gupta, Regularization strategies for quantile regression, arXiv preprint arXiv:2102.05135.

[154] K. Maciejowska, J. Nowotarski, R. Weron, Probabilistic forecasting of electricity spot prices using factor quantile regression averaging, International Journal of Forecasting 32 (3) (2016) 957–965.

[155] J. Bracher, E. L. Ray, T. Gneiting, N. G. Reich, Evaluating epidemic forecasts in an interval format, PLoS computational biology 17 (2) (2021) e1008618.

[156] E. Straitouri, L. Wang, N. Okati, M. G. Rodriguez, Improving expert predictions with conformal prediction, in: International Conference on Machine Learning, PMLR, 2023, pp. 32633–32653.

[157] L. Lindemann, M. Cleaveland, G. Shim, G. J. Pappas, Safe planning in dynamic environments using conformal prediction, IEEE Robotics and Automation Letters.

[158] I. Cortés-Ciriano, A. Bender, Concepts and applications of conformal prediction in computational drug discovery, arXiv preprint arXiv:1908.03569.

[159] G. Shafer, V. Vovk, A tutorial on conformal prediction., Journal of Machine Learning Research 9 (3).

[160] Y. Kato, D. M. Tax, M. Loog, A review of nonconformity measures for conformal prediction in regression, Conformal and Probabilistic Prediction with Applications (2023) 369–383.

[161] H. Papadopoulos, Inductive conformal prediction: Theory and application to neural networks, in: Tools in artificial intelligence, Citeseer, 2008.

[162] N. Meinshausen, G. Ridgeway, Quantile regression forests., Journal of machine learning research 7 (6).

[163] N. Seedat, A. Jeffares, F. Imrie, M. van der Schaar, Improving adaptive conformal prediction using self-supervised learning, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2023, pp. 10160–10177.

[164] I. Gibbs, E. Candes, Adaptive conformal inference under distribution shift, Advances in Neural Information Processing Systems 34 (2021) 1660–1672.

[165] C. Xu, Y. Xie, Sequential predictive conformal inference for time series, in: International Conference on Machine Learning, PMLR, 2023, pp. 38707–38727.

# BIOGRAPHICAL STATEMENT

Jie Han earned her B.S. degree in Geographic Information Science from Nanjing Normal University, China, in 2015, followed by her M.S. degree in Photogrammetry and Remote Sensing from Wuhan University, China, in 2018. She completed her Ph.D. in Industrial Engineering at The University of Texas at Arlington in 2023. From 2020 to 2023, she served as a research graduate assistant in a smart grid project supported by the National Science Foundation award ECCS-1938895, focusing on research in electricity price forecasting using deep learning models. During the summer of 2023, she undertook an internship at the National Renewable Energy Laboratory (NREL), conducting research on load forecasting. Her current research interests revolve around probabilistic multivariate time series forecasting using deep learning models. She is an active member of INFORMS and IISE.