Nonparametric Multivariate Probability Density Forecast in Smart Grids With Deep Learning

Zichao Meng, Graduate Student Member, IEEE, Ye Guo, Senior Member, IEEE, Wenjun Tang, Member, IEEE, and Hongbin Sun, Fellow, IEEE

Abstract—This paper proposes a nonparametric multivariate density forecast model based on deep learning. It not only offers the whole marginal distribution of each random variable in forecasting targets, but also reveals the future correlation between them. Differing from existing multivariate density forecast models, the proposed method requires no a priori hypotheses on the forecasted joint probability distribution of forecasting targets. In addition, based on the universal approximation capability of neural networks, the real joint cumulative distribution functions of forecasting targets are well-approximated by a special positive-weighted deep neural network in the proposed method. Numerical tests from different scenarios were implemented under a comprehensive verification framework for evaluation, including the very short-term forecast of the wind speed, wind power, and the day-ahead forecast of the aggregated electricity load. Testing results corroborate the superiority of the proposed method over current multivariate density forecast models considering the accordance with reality, prediction interval width, and correlations between different random variables.

Index Terms—Deep learning, multivariate monotone neural network (NN), NNs, multivariate probability density forecast.

I. INTRODUCTION

A. Background

RENEWABLE energies, e.g., wind and solar, and flexible loads, e.g., electric vehicles, play key roles in our goal of carbon neutrality. Until 2019, the total worldwide installed capacity of renewable energies has reached over 2532 GW [1], and penetrations of wind and solar power generation in Denmark, Germany, and the United Kingdom have exceeded 62%, 33%, and 30% in 2020, respectively [2]. Besides, electric vehicles have been more and more welcome by the market with the development of energy storage technologies, which decreases carbon emissions. In this way, renewable energies and flexible loads are commonly witnessed in centralized or distributed forms in power systems, and higher penetration levels of them are expected for carbon neutrality all over the world.

However, with the extensive integration of renewable generations and rapid popularization of electric vehicles, power systems are facing increasing uncertainties in both generation and demand sides, and renewable energy dispatch has become a more and more critical component of energy management systems in smart grids. In practice, look-ahead dispatch, which is recursively scheduled by dispatch centers based on the forecast of renewable generation outputs and demands, is typically used to improve the utilization of renewable energy resources and guide better charging strategies for electric vehicles. Effective forecast techniques are thus imperative under uncertainties, which can help to optimize decision-making with different lead times, e.g., from several minutes to years ahead.

B. Literature Review

According to how much information a forecast model can offer, existing forecast models can be categorized into three classes: point forecast [3], [4], quantile and prediction interval (PI) forecast [5], [6], and probability density function (PDF) forecast [7], [8]. Among all these forecast models, PDF forecast provides us with the most information (full probability distribution). In particular, multivariate density forecast in PDF forecast offers correlation information between different random variables through the joint probability distribution, which is of crucial importance for the power system operator to maintain the balance between power generation and consumption with extensive penetration of renewable energies and flexible loads.

The key to multivariate density forecast is to effectively model the full joint probability distribution considering the dependence structure. Thus, the design of the multivariate density forecast is much more complicated than that of the univariate one [9], [10], [24] owing to the complex correlation between random variables [37]. Existing multivariate density forecast models can be classified into parametric models or nonparametric ones. For parametric models, the joint probability distribution of forecasting targets is formed under a priori hypotheses of the distribution type, or the copula used in the joint probability distribution comes from parametric families, e.g., Gaussian and
Archimedean copulas. Thus, parametric models need to estimate statistics (usually expectations and covariance matrices) of random variables or determine the optimal copula family for the suitable dependence structure. Examples of parametric multivariate density forecast models include vector autoregressive (AR) model with exogenous variables (VARX) [11], t-copula with vector autoregressive integrated moving average (VARIMA) and component generalized AR conditional heteroscedasticity (C-GARCH) [12], regular vine (R-vine) copula with marginal distributions from kernel density estimation (KDE) [13] or distance weighted KDE (WKDE) [14], and deep learning-based model under Gaussian distribution [16]. Although these parametric models worked fine in some applications, nonnegligible limitations exist here. In [11], [12], and [16], forecasted joint probability distributions are assumed to follow Gaussian distribution, but these assumptions may not hold because distribution types of random variables (e.g., wind speed, power, and electricity load) may change with time. In [13] and [14], parametric copulas were adopted to build the dependence structure, whereas the complicated correlation between random variables may not be effectively described since it is difficult to identify the optimal parametric form of copulas via goodness-of-fit tests [21], [38].

For nonparametric models, neither a priori assumptions of probability distribution type nor parametric copulas are needed. Existing nonparametric multivariate density forecast models were mainly developed based on kernel density estimators [39] and examples of them include multivariate KDE (MKDE) [17], MKDE with fast sum updating method [40], and asymptotic mean integrated square error MKDE (AMISE-MKDE) [18], [19]. However, in these MKDE-based models, selections of appropriate kernel functions (e.g., Gaussian and Epanechnikov kernels) and bandwidths are tricky in the multivariate scenario. Besides, the performance of the model deteriorates as the dimensionality of the forecasting target increases (also called the “curse of dimensionality”), which limits applications to the high-dimensional density forecast [22]. In addition, when we consider other forms of probabilistic forecast (quantile or PI forecast) but not only density forecast, quantile regression (QR) [15], [41] can be applied to forming nonparametric multivariate probabilistic forecast models. Examples of them include multivariate QR [20] (developed with linear programming) and empirical copula with QR [16] (developed with deep learning) and drawbacks still exist in these nonparametric approaches. The first drawback is the weakness in correlation modeling between random variables, since quantiles for different random variables in [20] were actually modeled independently and the empirical copulas in [16] were piecewise linear and time-invariant, which may fail to capture the complicated time-variant correlation between different random variables [23]. The second drawback is that these forecast models only offer contours of the probability distribution, so theoretically much less information can be provided compared with the density forecast, which fails to offer the highest flexibility for decision-makers.

Another field highly related to our topic is multivariate time series forecasting, in which various deep learning-based end-to-end frameworks have been proposed for jointly temporal representation learning of multivariate time series data [42], [43], [44]. For instance, the long short-term memory (LSTM) network demonstrated prominent performance on large-scale dengue incidence time series forecasting of different cities [45]. Attention mechanisms were used in the recurrent neural network (RNN) for time-invariant patterns across multiple time steps [46] or in the encoder-decoder structure with Bidirectional LSTM (BiLSTM) for adaptively learning the temporal dependency features of multivariate time series data [47], which showed improvements in various multivariate forecasting tasks. The state space model with a jointly-learned deep RNN was proposed for the forecast of different electricity and traffic data, which makes the model data efficient and interpretable [48]. The stacked autoencoder was applied to learning heterogeneous features of data and achieving high-quality multi-step short-term wind power predictions [49]. However, these sequence-to-sequence models can only implicitly learn the joint probability distribution of their forecasting targets and only offer deterministic forecasting results. This is essentially different from the multivariate density forecast since joint probability distributions are established explicitly for the latter.

Therefore, various limitations exist in current multivariate density forecast approaches and other related methods. Nonetheless, for the density forecast, nonparametric approaches generally offer more reasonable results in real-world applications [8], [12], [16], [24], since the real probability distribution may not be able to describe via certain distribution types and parametric families of copulas. Thus, for a better approximation of the unknown joint probability distribution, a nonparametric multivariate density forecast model with powerful approximation capability may be a good fit. This concern has motivated us to build the nonparametric multivariate density forecast with deep learning considering the universal approximation capability of deep neural networks (NNs) and this is a research gap in the multivariate density forecast. Recently, a nonparametric univariate density forecast model, the distribution approximation network–network forecast network (DAN-NFN) [24], has been proposed. In DAN-NFN, continuous cumulative distribution functions (CDFs) of real probability distributions are approximated by a positive-weighted single-in-single-out (SISO) NN with large capacity. It has outstanding performance in various scenarios, such as the wind speed, power, and electricity price forecast. This idea is in line with our previous concern, whereas it is difficult to build the nonparametric multivariate density forecast model through NNs since the much more complicated nature of the joint probability distribution compared with the univariate one, e.g., we should consider the time-variant correlation between random variables in joint probability distributions and ensure that the joint PDF is always nonnegative.

C. Contributions

In this paper, a deep learning-based nonparametric multivariate density forecast model is proposed for the forecast of renewable energy resources and loads. Specifically, the proposed model consists of two deep NNs: a joint distribution approximation network (JDAN) and a network forecast network (NFN).
JDAN is a generic framework for approximating real continuous joint CDFs through a deep NN with four parts involved (i.e., parallel units, normalization layers, a correlation layer, and a fusion layer), and NFN outputs all parameters of JDAN. The proposed model is evaluated under a comprehensive verification framework considering the accordance with reality, PI width as well as correlations between random variables.

Overall, the proposed model has the following advantages:
1) To the best of our knowledge, this is the first time that deep learning has been applied to building a nonparametric multivariate density forecast model in power and energy systems. In our model, the real joint CDFs of forecasting targets are approximated by a large-capacity NN, and no more limitation is introduced on the forecast of joint probability distributions. Therefore, no a priori hypotheses are required on the forecasted joint probability distribution, i.e., the proposed approach is nonparametric. Such a combination of deep learning and nonparametric multivariate density forecast takes both the advantage of deep NNs (universal approximation capability) and existing nonparametric methods (no requirement for a priori hypotheses) [17], [18], [19], [40].

2) The full forecasted joint probability distribution is provided by our model in a nonparametric way. Whereas, in other powerful nonparametric approaches based on QR [16], [20], only contours of the probability distribution can be offered, which may not be enough for the decision-making of system operators.

3) The time-variant correlation is modeled with maximum likelihood estimation (MLE) via gradient descent. The time-variant temporal or spatial correlation between random variables in forecasting targets is represented by the correlation layer of our proposal. The suitable parameters in the correlation layer are automatically learned by maximizing MLE via gradient descent. This is more adaptive than previous parametric multivariate density forecast models, where the correlation between random variables was determined via predesigned parametric copulas [13], [14], or other related nonparametric multivariate probabilistic forecast models, where the dependence structure was built through time-invariant empirical copula [16] or just ignored [20].

4) A one-step end-to-end framework is designed for joint probability distribution modeling. Our model takes a global view in approximating the real joint probability distribution in an end-to-end manner, i.e., the marginal distribution and correlation between random variables are modeled simultaneously. This is potentially superior to the greedy method used in existing copula-based approaches [12], [13], [14], in which the joint probability distribution was constructed following a two-step process: estimate the best marginal distributions for each random variable first, and then choose the best copula to fit the joint probability distribution. Besides, the marginal of forecasted joint probability distributions can be easily obtained from the proposed framework, since the marginal CDF is the input-output function of the parallel unit with the corresponding normalization layer. Whereas in previous parametric methods with multivariate distribution types [11] or nonparametric ones with MKDE [17], [18], [19], [40], marginal distributions were computed from complicated multiple integrals.

The rest of this paper is organized as follows. Section II presents preliminaries about multivariate density forecast and JDAN-NFN. Section III introduces the model design for multivariate density forecast in smart grids. Section IV details the evaluation method. Section V presents numerical simulations results, and conclusions are drawn in Section VI.

II. PRELIMINARIES

A. Multivariate Density Forecast

Multivariate density forecast aims to obtain the joint probability distribution of multiple future random variables (forecasting targets) based on information available. Namely, multivariate density forecast with lead step \( \tau \) aims to estimate the real joint probability distribution of the forecasting target \( \hat{y}_{t+\tau} \) at time spot \( t+\tau \) (denoted as \( \Phi_{t+\tau} \)), i.e., \( y_{t+\tau} \sim \Phi_{t+\tau} \). Denoting the available information set up to time \( t \) as \( X_t \) (\( X_t \) is a vector concatenated with different available time-series features), a multivariate density forecast model can be regarded as a function \( \Phi_{t+\tau}|X_t = \Theta(X_t; \theta) \), where \( \Theta(\cdot; \theta) \) is the density forecast model and \( \theta \) includes parameters of \( \Theta \). \( \Phi_{t+\tau}|X_t \) is the forecasted joint probability distribution, which is an estimate of \( \Phi_{t+\tau} \). Different \( \Phi_{t+\tau}|X_t \) can be determined by selecting different \( \Theta \) and changing \( \theta \). Given an appropriate \( \Theta \), the training of \( \Theta(\cdot; \theta) \) is to find the best \( \theta \), so that the forecasted \( \Phi_{t+\tau}|X_t \) is as close to \( \Phi_{t+\tau} \) as possible. The real joint probability distribution \( \Phi_{t+\tau} \) can take the form of joint PDF \( f(\cdot) \) or joint CDF \( F(\cdot) \). Let \( y_{t+\tau} \) be a D-dimensional random vector, then \( f(\cdot) \) should satisfy the following conditions:

\[
\begin{align*}
\begin{cases}
  f(\cdot) \text{ is nonnegative and bounded}, \\
  f_{-\infty}^{+\infty} \cdots f_{-\infty}^{+\infty} f(y)dy^1 \cdots dy^D = 1.
\end{cases}
\end{align*}
\]

For \( F(\cdot) \), it can be written as

\[
F(y_{t+\tau}) = \int_{-\infty}^{y_{t+\tau}^1} \cdots \int_{-\infty}^{y_{t+\tau}^D} f(y)dy^1 \cdots dy^D.
\]

The joint PDF \( f(\cdot) \) being bounded leads to \( F(\cdot) \) being continuously differentiable. The limit of \( F(y_{t+\tau}) \) is equal to 1 as all \( y_{t+\tau} \) approach \( +\infty \) and is equal to 0 as any variable in \( y_{t+\tau} \) approaches \( -\infty \). Thus, from the perspective of joint CDF, \( F(\cdot) \) should satisfy the following conditions:

\[
\begin{align*}
\begin{cases}
  (i) F(\cdot) \text{ is continuously differentiable}, \\
  (ii) \frac{\partial^D F(y)}{\partial y^1 \cdots \partial y^D}|_{y=y_{t+\tau}} \geq 0, \\
  (iii) \lim_{y_{t+\tau} \to -\infty} F(y_{t+\tau}) = 1, \\
  (iv) \lim_{y_{t+\tau} \to +\infty} F(y_{t+\tau}) = 0, \forall d \in [1, D].
\end{cases}
\end{align*}
\]

The possible value domain of \( y_{t+\tau} \) varies in different forecasting scenarios and we assume that real joint CDFs of forecasting targets are continuously differentiable in their corresponding
value domains. For instance, \(y_{t+\tau}\) ranges from 0 to \(+\infty\) in the wind speed and aggregated electricity load forecasting; ranges from 0 to the total capacity of wind turbines in the wind farm in the wind power forecasting. Equations (1) to (3) can be applied to different forecasting scenarios by changing the upper and lower bounds of the integral, and it should be noticed that we only consider the continuous joint probability distribution here but not the discrete form, i.e., the joint probability mass.

B. Basic Structure of DAN-NFN

Our method is developed based on DAN-NFN whose basic structure is shown in Fig. 1. There are two deep NNs. DAN is a SISO positive-weighted NN and is built for approximating the real CDF of the forecasting target. For NFN, it takes available information as input and outputs all parameters of DAN. Based on the nonlinear approximating capability of NNs [25] and the theorem that a SISO positive-weighted NN can approximate any univariate continuous monotonically nondecreasing function [26], DAN-NFN has shown the state-of-the-art performance in the univariate forecasting task in power systems. However, the application of deep learning approaches to nonparametric multivariate density forecast in smart grids is still an open question.

III. MODEL DESIGN FOR MULTIVARIATE DENSITY FORECAST IN SMART GRIDS

A. Basic Structure of JDAN-NFN

We first present the basic structure of JDAN-NFN. As shown at the bottom of Fig. 2, NFN is used to generate parameters of JDAN and JDAN is used to approximate the real joint CDF. To be specific, NFN outputs Tensors \(W^+, B,\) and \(C\) with different activation functions. Tensor \(W^+\) denotes the weight of JDAN and comes from SoftPlus function \(s_{\theta}(x)=\ln(1+e^x),\) thus \(W^+\in\mathbb{R}^+\). Tensor \(B\) denotes the bias of JDAN and comes from linear function, thus \(B\in\mathbb{R}\). Tensor \(C\) (a vector with dimension \(\binom{D}{2}\)) is the coefficient related to correlations between random variables and comes from tanh function \(\tanh(x) = \frac{e^x-e^{-x}}{e^x+e^{-x}},\) thus \(C\in(-1,1)\). In short, NFN maps input \(X_t\) to \([W^+, B, C]\). JDAN has four parts, i.e., parallel units, normalization layers, a correlation layer, and a fusion layer. It picks \([W^+, B, C]\) as its parameters, and the input-output mapping of JDAN can be regarded as a deterministic function \(\Psi_f(\cdot; W^+, B, C),\) i.e., the estimate of the joint CDF. In our framework, we only train parameters in the NFN and it offers parameters of JDAN. The reasons why we do not directly train the JDAN are: 1) We may need to force the weight of the JDAN to be positive very frequently during the gradient descent process, which makes the training process unstable. 2) The trained JDAN can not represent different joint CDFs since parameters of the JDAN are fixed after the training process. So, we control the weight of JDAN by the output of another NN, i.e., the NFN. We will introduce the detailed structure of NFN and JDAN respectively in the following two subsections.

B. The NFN Part

NNs are able to approximate arbitrary complex mappings by a end-to-end manner and their approximation capability can be increased more effectively with deeper structures [25], thus deep NNs are adopted in NFN. However, one challenge in training deep NNs is that parameters of deep layers may be updated very slowly. Thus, residual structure [27], which is realized through residual connections (identical mappings between nonadjacent layers), is implemented in NFN. Since the inputs of NFN are time series \(X_t,\) LSTM [28] is used in this paper for the modeling of temporal relations. Specifically, the LSTMs adopted here share the same deep residual structure as the one demonstrated in [24]. For example, given a lag interval \(\delta,\) the input to LSTM can be written as \(X_t = [x_{t-\delta+1}, \ldots, x_t],\) then from time spot \(\kappa = t-\delta+1\) to \(\kappa = t\) the recurrent calculation steps in LSTM are as follows:

\[
i_k = \sigma(w_{ix}\cdot x_k + w_{ih}\cdot h_{k-1} + w_{ic}\cdot c_{k-1} + b_i),
\]

\[
f_k = \sigma(w_{fx}\cdot x_k + w_{fh}\cdot h_{k-1} + w_{fc}\cdot c_{k-1} + b_f),
\]

\[
c_k = f_k \odot c_{k-1} + i_k \odot \tanh(w_{cx}\cdot x_k + w_{ch}\cdot h_{k-1} + b_c),
\]

\[
o_k = \sigma(w_{ox}\cdot x_k + w_{oh}\cdot h_{k-1} + w_{oc}\cdot c_{k-1} + b_o),
\]
where \( i \), \( f \), and \( o \) represent input gate, forget gate and output gate, respectively. Vectors \( h \) and \( c \) represent cell state and hidden state, respectively. The sizes (number of elements) of \( h \) and \( c \) are the same. Tensors \( w \) with different subscripts are weights, and \( b \) with different subscripts are biases. Notation \( \odot \) represents entrywise product. We define that the recurrent calculation steps (4)–(7) as one LSTM layer, and the size of \( h \) or \( c \) as its width.

As shown on the upper left of Fig. 2, NFN is built by many LSTM layers with residual connections and three fully connected (FC) layers. Batch normalization (BN) [29] is also implemented to accelerate the training of NFN. Intuitively, the main part of NFN is composed of many blocks, which contain several LSTM layers and residual connections. We denote the number of blocks in NFN as \( N \). As we focus on the depth of NNs, the layers in NFN are built with the same widths for simplicity, i.e., the widths of LSTM layers and FC layers are \( W \) in NFN.

### C. The JDAN Part

The design of JDAN relies on one approximation theorem for NN: a positive-weighted multiple-in-single-out (MISO) feed-forward neural network with a specified structure (described in Fig. 1 in [26]) can arbitrarily approximate any continuous multivariate monotonically nondecreasing function. As the real joint CDF is multivariate monotonically nondecreasing, a MISO positive-weighted NN thus has the potential for approximating it based on the theorem introduced before. From Section II-A, we know that a reasonable estimate of the real joint CDF \( F(\cdot) \) [denoted as \( \hat{F}(\cdot|X) \)] should satisfy all the conditions in (3). It can be proven that a MISO positive-weighted NN is multivariate monotonically nondecreasing. However, one can not extend this property to the higher-order-derivative form that meets condition (ii) in (3) via an easy extension of DAN-NFN to the multivariate density forecast, because the output of the higher-order-derivative of activation functions is not always nonnegative (for sigmoid and tanh) or always zero (for linear and ReLU). In other words, the joint probability distribution can not be approximated by just changing the DAN part in the original DAN-NFN framework from a SISO positive-weighted NN to a MISO one, which is detailed in Appendix A.

To address this problem, JDAN is proposed for the approximation of the real joint probability distribution and its structure consists of the following four parts as shown on the lower right of Fig. 2.

1) **Parallel Units:** Each variable in \( y_{i} \) is sent into a SISO positive-weighted NN (can be regarded as a parallel unit as well), respectively. Sigmoid function \( \sigma(x) = \frac{1}{1 + e^{-x}} \) is chosen as the activation function of hidden layers and linear function is chosen as the activation function of the output layer in each parallel unit, i.e.,

\[
\Psi^{d}(y^{d}_{t}, W^{d}, B^{d}) = W^{d+}_{K+1} \cdots \sigma(W^{d}_{2} \cdot \sigma(W^{d}_{1} \cdot y^{d}_{t+\tau} + B^{d}_{1}) + B^{d}_{2}) \cdots + B^{d}_{K+1}, \tag{8}
\]

where \( y^{d}_{t+\tau}, d \in [1, D] \), is the \( d \)-th variable in \( y_{t+\tau} \). Tensors \( W^{d+} \in W^{d+} \) and \( B^{d} \in B \) represent all the weights and biases in the \( d \)-th parallel unit, respectively, which are determined by the outputs of NFN. \( K \) is the number of hidden layers in the \( d \)-th parallel unit. Tensors \( W^{d+}_{K+1} \in W^{d+} \) and \( B^{d}_{K+1} \in B^{d}, k \in [1, K+1] \), are weights and biases in the \( k \)-th layer of the \( d \)-th parallel unit, respectively. \( \Psi^{d}(-; W^{d+}, B^{d}) \) is the input-output mapping of the \( d \)-th parallel unit, and the number of parallel units is equal to the dimension of \( y^{d}_{t+\tau} \). One can prove that (8) is a monotonically increasing function in the same way following (28)–(34) demonstrated in Appendix A.

Specifically, the detailed information of parallel units are shown on the upper right of Fig. 2. Parallel units are constructed by many FC layers with residual connections for great approximation capability. They can also be viewed as the composition of many blocks similar with the main part of NFN. We denote the number of blocks in each parallel unit as \( N_{j} \), and take the widths of all FC layers as \( W_{j} \) for simplicity.

2) **Normalization Layers:** The output of each parallel unit will be normalized by a normalization layer, denoted as

\[
\overline{\Psi}^{d} = \frac{\Psi^{d}(y^{d}_{t+\tau}; W^{d+}, B^{d}) - \Psi^{d}(U_{d}; W^{d+}, B^{d})}{\Psi^{d}(U_{d}; W^{d+}, B^{d}) - \Psi^{d}(L_{d}; W^{d+}, B^{d})} \tag{9}
\]

where \( U_{d} \) and \( L_{d} \) are the upper and lower bounds of the value domain of \( y^{d}_{t+\tau} \), respectively. One can verify that \( \overline{\Psi}^{d} \) is also monotonously nondecreasing and \( \lim_{y^{d}_{t+\tau} \rightarrow -\infty} \overline{\Psi}^{d} = 0 \).

3) **Correlation Layer:** Correlation layer is used to construct the time-variant correlation between input variables of JDAN based on the output of the normalization layer. In this paper, the correlation layer is represented as

\[
\Psi^{C} = \frac{1}{\binom{D}{2}} \sum_{i>d}^{D} \sum_{i=d}^{D-1} [C_{di} \cdot (1 - \overline{\Psi}^{d}) \cdot (1 - \overline{\Psi}^{d}) + 1], \tag{10}
\]

where \( C_{di} \) is the element in \( C \) with index \( \sum_{k=1}^{d-1}(D-k) + i-d \) if \( d \geq 2 \) or \( i-d \) if \( d = 1 \), which is utilized to demonstrate the time-variant correlation between \( y^{d}_{t+\tau} \) and \( y^{d+}_{t+\tau} \). We can verify that (10) is multivariate nondecreasing as well, and \( \lim_{y^{d}_{t+\tau} \rightarrow +\infty} \Psi^{C} = 1 \).

4) **Fusion Layer:** Fusion layer receives the results of all normalization layers and the correlation layer. The input-output mapping of JDAN is finally constructed as

\[
\Psi_{\mathcal{F}}(y_{t+\tau}; W^{+}, B, C) = \Psi^{C} \prod_{d=1}^{D} \overline{\Psi}^{d}. \tag{11}
\]

Next, we explain why the proposed structure is able to meet all the conditions in (3). Based on the monotonicity of (8), (9), and (10), as analyzed in Appendix B, we verify that

\[
\frac{\partial^{2}\Psi_{\mathcal{F}}(y; W^{+}, B, C)}{\partial y^{1} \cdots \partial y^{D}}|_{y=y_{t+\tau}} \geq 0. \tag{12}
\]
From the product law of limitations, we have the following limits as
\[
\lim_{y_{t+r} \to +\infty} \Psi_{\mathbf{J}}(y_{t+r}; \mathbf{W}^+, \mathbf{B}, \mathbf{C}) = \lim_{y_{t+r} \to -\infty} \Psi_{\mathbf{J}}(y_{t+r}; \mathbf{W}^+, \mathbf{B}, \mathbf{C}) \cdot \lim_{y_{t+r} \to +\infty} \Psi_{\mathbf{J}}(y_{t+r}; \mathbf{W}^+, \mathbf{B}, \mathbf{C}) = 1,
\]
(13)
\[
\lim_{y_{t+r} \to +\infty} \Psi_{\mathbf{J}}(y_{t+r}; \mathbf{W}^+, \mathbf{B}, \mathbf{C}) = 0, \forall d \in [1, D].
\]
(14)

Note that, \(\Psi_{\mathbf{J}}(y_{t+r}; \mathbf{W}^+, \mathbf{B}, \mathbf{C})\) is continuously differentiable as activation functions of JDAN are sigmoid and linear, thus meets condition (i). According to (12)–(14), conditions (ii)–(iv) are also satisfied. Eventually, the forecasted joint probability distribution \(\hat{F}(\mathbf{y}_{t+r}|\mathbf{X}_t)\) with joint CDF \(\hat{F}(|\mathbf{X}_t)\) is formulated as
\[
\hat{F}(\mathbf{y}_{t+r}|\mathbf{X}_t) = \Psi_{\mathbf{J}}(y_{t+r}; \mathbf{W}^+, \mathbf{B}, \mathbf{C}).
\]
(15)

The forecasted joint PDF \(\hat{f}(\mathbf{y}_{t+r}|\mathbf{X}_t)\) can be derived from (15) as
\[
\hat{f}(\mathbf{y}_{t+r}|\mathbf{X}_t) = \frac{\partial^D \Psi_{\mathbf{J}}(y; \mathbf{W}^+, \mathbf{B}, \mathbf{C})}{\partial y^1 \cdots \partial y^D} |_{y=y_{t+r}}.
\]
(16)

In summary, to estimate real joint CDFs, a special MISO positive-weighted NN, i.e., JDAN, is established, which includes parallel units, normalization layers, a correlation layer, and a fusion layer, as shown in Fig. 2. As the forecasted marginal CDF of \(y_{t+r}^d\) can be derived as \(\hat{F}_{Y|_{t+r}}(y_{t+r}^d|\mathbf{X}_t) = \lim_{y_{t+r}^d \to +\infty} \hat{F}(y_{t+r}^d|\mathbf{X}_t) = \Psi^d\), the input-output function denoted by (8) and (9) actually represents the forecasted marginal distribution function of the random variable sent into the corresponding parallel unit.

D. Loss Function and Gradients Propagation

1) Loss Function: According to (15)–(16), JDAN-NFN can offer the forecasted joint CDF \(\hat{F}(|\mathbf{X}_t)\) of forecasting targets and the corresponding joint PDF \(\hat{f}(|\mathbf{X}_t)\), then the loss function for maximum likelihood estimation is constructed based on \(\hat{f}(|\mathbf{X}_t)\). Denoting the observation (real measurement) of \(y_{t+r}\) as \(\mathbf{y}_{t+r}\), the loss function \(L\) used in the training process is formulated as
\[
L(\theta_N, \mathbf{X}_t, \mathbf{y}_{t+r}) = -\ln[\hat{f}(\mathbf{y}_{t+r}|\mathbf{X}_t)]],
\]
(17)
where \(\theta_N\) represents parameters of NFN.

2) Back Propagation of Gradients and Updating of Parameters: The back propagation of gradients follows the chain rule of derivative. First, the gradients of \(L\) with respect to parameters of JDAN are formed as \(\nabla_{\mathbf{W}^+}(L), \nabla_{\mathbf{B}}(L), \) and \(\nabla_{\mathbf{C}}(L)\), respectively. Second, the gradients of \(L\) with respect to parameters of NFN are calculated as
\[
\nabla_{\theta_N}(L) = \nabla_{\theta_N} \mathbf{W}^+ \cdot \nabla_{\mathbf{W}^+} (L) \quad + \nabla_{\theta_N} \mathbf{B} \cdot \nabla_{\mathbf{B}} (L) \quad + \nabla_{\theta_N} \mathbf{C} \cdot \nabla_{\mathbf{C}} (L).
\]
(18)

Finally, the NFN is updated according to \(\nabla_{\theta_N}(L)\), thus we pass gradients to NFN and update the parameters of NFN and JDAN.

IV. PERFORMANCE EVALUATION

The joint probability distribution is difficult not only in modeling but also in evaluation [37]. Here, we propose a comprehensive evaluation method considering both marginal distributions and correlations between random variables in forecasting targets.

A. Evaluation of Marginal Distributions

The forecasted marginal distribution is evaluated within the probabilistic forecast evaluation framework in [30], and average evaluation indices of all forecasted marginal distributions are developed.

1) Reliability: Reliability is the foremost concern for probabilistic forecast model evaluation, which measures deviations between the nominal proportion and the observed frequency of the data below the quantile forecasting. Define the forecasted marginal distributions as \(\hat{F}_d, d \in [1, D]\), and \(q_{t+r|\tau}^a, d\) as the quantile of \(\hat{F}_d\) with nominal proportion \(\alpha_j, d\) \((j \in [1, J], J = 99)\), from \(\alpha_1, d = 1%\) to \(\alpha_J, d = 99%\) with steps 1%, the deviation with different \(\alpha_j, d\) can be obtained as
\[
b_{\tau}^{\alpha_j, d} = \alpha_j, d - \frac{1}{N} \sum_{i=1}^{N} H(q_{t+t|\tau}^a - q_{t+t|\tau}^d),
\]
(19)
where \(N\) is the number of samples in the testing data set, \(b_{\tau}^{\alpha_j, d}\) is the deviation for \(\hat{F}_d\) corresponding to \(\alpha_j, d\), and \(H\) is the unit step function
\[
H(x) = \begin{cases} 
1, & x \geq 0 \\
0, & x < 0.
\end{cases}
\]
(20)

To evaluate all forecasted marginal distributions, the absolute average deviation for the reliability evaluation is defined as \(\bar{b}_{\tau} = \frac{1}{D} \sum_{d=1}^{D} \sum_{j=1}^{J} |b_{\tau}^{\alpha_j, d}|\).

2) Sharpness: Probabilistic model with more satisfied sharpness should be chosen. Sharpness measures the average width of PIs with different nominal proportion \((1-\alpha_j, d)\) as
\[
\hat{\alpha}_{t+r|\tau} = \frac{1}{N} \sum_{i=1}^{N} (q_{t+t|\tau}^{1-\alpha_j, d} - q_{t+t|\tau}^{\alpha_j, d})^2,
\]
(21)
where \(\hat{\alpha}_{t+r|\tau}\) is the average width of PIs for \(\hat{F}_d\) corresponding to \((1-\alpha_j, d)\), and we denote the average width for the sharpness evaluation as \(\bar{\alpha}_{t+r|\tau} = \frac{1}{D} \sum_{d=1}^{D} \sum_{j=1}^{J} \hat{\alpha}_{t+r|\tau}^{\alpha_j, d}\).

3) Skill Score: Skill score takes both the reliability and sharpness into consideration, which is generally a negative value. The higher the skill score the better, and the skill score is 0 for a perfect forecast model. The skill score on a single time spot is defined as
\[
S_t^{d} = \frac{1}{N} \sum_{j=1}^{J} \left\{ H(q_{t+t|\tau}^a - q_{t+t|\tau}^d) - \alpha_j, d (q_{t+t|\tau}^d - q_{t+t|\tau}^a) \right\},
\]
(22)
where \(S_t^{d}\) denotes the skill score for \(\hat{F}_d\) at time spot \(t\). An average skill score is defined as \(\bar{S}_t = \frac{1}{D} \sum_{d=1}^{D} \sum_{i=1}^{N} S_t^{d}\).
TABLE I

| Scenario | Case A | Case B | Case C |
|----------|--------|--------|--------|
| Location | 34°12′23.04″N, 102°44′32.64″W Texas, USA | Brown Hill Wind Farm, \( P_c = 94.5 \) MW | Queensland, Australia |
| Covering period | 2019/12/31 to 2020/12/31 | 2019/12/31 to 2020/12/31 | 2009/12/31 to 2020/12/31 |
| Resolution | 5 minutes | 5 minutes | 2 hours |
| \( X_t \) | historical wind speed series in different directions, historical temperature series, historical relative humidity series, time of the day | historical wind power series of different wind farms, time of the day | historical electricity price series, day of the week, time of the day |
| \( y_{t+\tau} \) | speed vector of next 5 minutes | power vector of next 5 minutes | load vector of the next day |
| \( D \) | 2 | 3 | 12 |

B. Evaluation of Correlations Between Random Variables

The variogram score \( VS \) [31] is applied to evaluate pairwise differences of the components of the multivariate quantity of interest. It is sensitive to misspecifications of the correlation between random variables in the forecasting target. The variogram score is calculated as

\[
VS_{t+\tau|m} = \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} \bigg( \bigg| y_{1:t+\tau}^{(i)} - y_{1:t+\tau}^{(j)} \bigg|^p - E_{\hat{\Phi}}|Y_1 - Y_j|^p \bigg)^2
\]

\[
\approx \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} \bigg( \bigg| y_{1:t+\tau}^{(i)} - y_{1:t+\tau}^{(j)} \bigg|^p - \frac{1}{m} \sum_{k=1}^{m} y_{(k)} - y_{(j)} \bigg)^2,
\]

where \( Y_1 \) and \( Y_j \) \((i, j \in D)\) are the \( i \)th and the \( j \)th entry of a random vector \( Y \) that is distributed according to the forecasted joint probability distribution \( \hat{\Phi}_{t+\tau} |X_t, Y^{(k)}_{1:t} \) and \( Y_{(k)} \) \((k \in m)\) are sampling results of \( Y_1 \) and \( Y_j \) and \( m \) is the number of generating samples. Notation \( \big| \cdot \big|^p \) represents the \( p \)th order norm, and \( p \) is set as 1 here. The weight \( w_{ij} \) for different entry pairs is taken as 1 as we assume that the random variables in forecasting targets are of equal importance. The lower the variogram score the better and an average \( VS \) is defined as \( \frac{1}{m} \sum_{n=1}^{m} V S_{n+\tau|m} \).

To calculate the variogram score in (23), \( m \) forecasting samples need to be generated from the forecasted joint probability distribution at one time spot first. In this paper, we take \( m = 500 \), and forecasting samples are generated with the following steps:

1) Step 1: Obtain forecasted conditional CDFs. Denoting the random variable vector \( y_1^{1:d}_{t+\tau} = [y_1^{d:t+\tau}, \ldots, y_d^{d:t+\tau}] \), \( d \in [1, D] \), we define that

\[
\hat{F}_{1:d}(y_1^{d:t+\tau} | X_t) = \int_{-\infty}^{y_1^{d:t+\tau}} \cdots \int_{-\infty}^{y_d^{d:t+\tau}} \hat{f}(y | X_t) dy_1^{d+1} \cdots dy_D^{D).
\]

The forecasted conditional CDF \( \hat{F}_{d:1:d-1}(y_1^{d:t+\tau} | y_1^{d-1:t+\tau}) \), \( i \in [1, D] \), can be expressed as

\[
\hat{F}_{d:1:d-1}(y_1^{d:t+\tau} | y_1^{d-1:t+\tau}) = \int_{-\infty}^{y_i^{d:t+\tau}} \int_{-\infty}^{y_i^{d-1:t+\tau}} \hat{f}_{d:i-1:d-1}(y_1^{d-1:t+\tau} | y_1^{d-1:t+\tau}, y_i^{d-1:t+\tau}) dy_1^{d+1} \cdots dy_{d-1}^{d-1}.
\]

2) Step 2: Generate the \( D \)-dimensional random vector \( U = [U_1, U_1, \ldots, U_D] \) from the uniform distribution \( U(0, 1)^D \), and obtain the correlated rank random vector \( Y \) using the inverse transform sampling as

\[
\begin{align*}
Y_1 & := \hat{F}_{1}^{-1}(U_1), \\
Y_d & := \hat{F}_{d:1:d-1}^{-1}(U_d | Y_{d-1}, Y_{d-2}, \ldots, Y_1), \quad d \in [2, D],
\end{align*}
\]

where \( \hat{F}_i \) denotes the forecasted marginal distribution of \( y_i^{1:t+\tau} \).

Then, forecasting samples are given from

\[
Y = [Y_1, Y_2, \ldots, Y_D].
\]

V. CASE STUDIES

A. Setting of Numerical Simulations

1) Description of Data Sets: Three real-world data-based case studies were implemented, which are detailed in Table I. The data set of case A is from the West Texas Mesonet issued by National Wind Institute [32], and both the data sets of case B and case C are from the Australian National Electricity Market [33], [34]. Case A is one-step-ahead forecast concerning the wind speed in latitudinal and longitudinal directions. Case B is a one-step-ahead forecast concerning the wind power output of three wind farms. Case C is a twelve-step-ahead (day-ahead) forecast concerning the aggregated load of one city. Features involved in \( X_t \) are also listed in Table I. All these three cases take time of the day as an input feature to consider the diurnal effect. For one-step-ahead forecasts, denote \( t_s \) as the cumulative number of seconds counted from 00:00 to the time stamp of \( y_1^{1:t+\tau} \), then time of the day can be expressed as a data pair \( \{\sin(\frac{2\pi t_s}{24\times3600}), \cos(\frac{2\pi t_s}{24\times3600})\} \). For the twelve-step-ahead forecast, twelve data pairs as demonstrated in one-step-ahead scenarios are generated corresponding to different time stamps of random variables in \( y_1^{1:t+\tau} \). In addition, day of the week is also taken as the input feature in case C to consider the weekly effect. The corresponding data pair is obtained by replacing \( t_s \) with the cumulative number of the day in a week and \( 24 \times 3600 \) with 7. All features in \( X_t \) as well as \( y_1^{1:t+\tau} \) have been normalized before the forecast.
## Training of JDAN-NFN and Other Alternatives

For each data set, 40% samples were used for training, 20% for validation, and 40% for testing. If the average loss on the training data set has been lower than that on the validation data set over 20 successive epochs, the training process will be stopped to avoid overfitting.

A grid-search method was adopted to find the optimal structure of JDAN-NFN (associated with $N_s$, $N_j$, $W_N$, and $W_j$), the learning rate $l_r$, and the lag interval $\delta$. The optimal combination of all hyperparameters \{ $N_s$, $N_j$, $W_N$, $W_j$, $l_r$, $\delta$ \} is determined when the sum of the average score and the negative average variogram score on the validation set is the highest during the grid-search process for all the three cases. Specifically, $N_s$ was chosen from \{2, 4, 8, 16\}, $N_j$ from \{2, 4, 8, 16\}, $W_N$ from \{16, 32, 64, 128\}, $W_j$ from \{16, 32, 64, 128\}, $l_r$ from \{0.0001, 0.001, 0.01\}, $\delta$ from \{15 min, 20 min, 25 min, 30 min, 35 min\} for case A and B and from \{30 h, 60 h, 90 h, 120 h\} for case C. The training of JDAN-NFN was implemented on CentOS 7.6 with 8 TITAN V GPUs and the optimal \{ $N_s$, $N_j$, $W_N$, $W_j$, $l_r$, $\delta$ \} are listed in Table II. For other hyperparameters of the model, we used default settings from the torch package.

Several state-of-the-art multivariate and univariate density forecast models were implemented for comparison, which are as follows.

- **Multivariate models:**
  1. A generalized persistence method that takes the form of multivariate Gaussian distribution. Its expectations are the last measurements and the covariance matrix is from the past residuals based on the basic exponential smoothing.
  2. VARX [11] with covariance matrix under multivariate Gaussian distribution through least squares estimation [see Eqs.(2.3.2) and (2.3.4) in [11]].
  3. The t-copula with VARIMA($\nu_p$, $\nu_q$, $\nu_d$)-C-GARCH($g_p$, $g_q$) (VCG-t-copula) under multivariate Gaussian distribution [12], where $\nu_p$, $\nu_q$, and $\nu_d$ are autoregressive order, moving average order, and differencing degree of VARIMA model respectively, and $g_p$, $g_q$ are orders of C-GARCH model. The coefficients $\nu_p$, $\nu_q$, $\nu_d$, $g_p$, and $g_q$ were determined by maximizing the sum of the average score and the negative average variogram score on the validation set for each data set.
  4. R-vine copula with WKDE (WKDE-R-copula) [14], in which the pair-copulas of the R-vine copula were chosen from Gaussian, t, Frank, Clayton, Gumbel, and Joe copulas with R-vine tree and independence test [35].
  5. Empirical copula with marginal distributions under Gaussian distribution (Gaussian-E-copula), a parametric model proposed in [16], in which the mean and standard deviation for marginal distributions were predicted by BiLSTM.
  6. AMISE-MKDE [18], in which the bandwidth was determined by minimizing the asymptotic mean integrated square error.
  7. Multivariate QR (MQR) [20], a nonparametric model adapted from the univariate QR model in [36] with the formulation of a constrained linear program.
  8. Empirical copula with QR (QR-E-copula), a nonparametric model also proposed in [16], in which marginal quantiles were predicted by BiLSTM and the quantile regression was implemented based on the pinball loss.

All these multivariate alternatives take the same $X_t$ with JDAN-NFN (as listed in Table I), and $\delta$ for $X_t$ was determined by maximizing the sum of the average skill score and the negative average variogram score on the validation set for each data set. Among these models, persistence, VARX, VCG-t-copula, WKDE-R-copula, and Gaussian-E-copula are parametric, while AMISE-MKDE, QR-E-copula, MQR, and JDAN-NFN are nonparametric. All NN-based models (e.g., Gaussian-E-copula, QR-E-copula, and JDAN-NFN) were trained with Adam optimizer, where $\alpha = 0.0005$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and batch size was set as 32.

- **Univariate models:**
  1. Ensemble convolutional neural network (CNN) [10], a deep learning-based parametric model based on the ensemble approach under Gaussian distribution.
  2. Bayesian LSTM [9], a deep learning-based parametric model based on Monte Carlo dropout under Gaussian distribution.
  3. Deep AR recurrent networks (DeepAR) [51], another deep learning-based parametric model obtained via Gaussian likelihood.
  4. QR neural network (QRNN) [50], a deep learning-based nonparametric model with skip connections.
  5. DAN-NFN [24], another deep learning-based nonparametric model built from LSTM.

We built the univariate density forecast model separately for each marginal distribution based on these univariate alternatives. All these methods also take the same $X_t$ with JDAN-NFN, and $\delta$ for $X_t$ was determined by maximizing the average skill score over all marginal distributions on the validation set for each data set. All of these methods are deep learning-based ones and they were trained with Adam optimizer in the multivariate models.

### Performance Evaluation and Comparison for Multivariate Approaches

The evaluation framework provided in Section IV is applied, and only forecasting results on the testing data set were used for evaluation. For each model in each case, average indices $\overline{\nu}_r$ (for reliability), $\overline{\nu}_s$ (for sharpness), $\overline{\nu}_t$ (for skill score), and $\overline{\nu}_v$ (for variogram score) are presented in Table III. In addition, evaluations for each forecasted marginal distribution were implemented. Specifically, for each forecasted marginal

| Case | $N_s$ | $N_j$ | $W_N$ | $W_j$ | $l_r$ | $\delta$ |
|------|-------|-------|-------|-------|------|--------|
| A    | 8     | 4     | 32    | 64    | 0.001| 25 min |
| B    | 4     | 8     | 64    | 64    | 0.001| 30 min |
| C    | 8     | 8     | 64    | 128   | 0.001| 120 h  |

**Table II: Optimal Hyperparameters**
As shown in Table III, on the whole, nonparametric models perform better than parametric ones in the reliability evaluation: $b_\tau$ of two parametric models are over 4%, while $\overline{\delta}_\tau$ of all the nonparametric models are below 3%. Fig 3(e) shows more details about the reliability evaluation for each forecasted marginal distribution, and JDAN-NFN has achieved the least $\overline{\delta}_\tau$ over almost all forecasted marginal distributions. As for the sharpness evaluation presented in Fig 3(e) and Table III, $\overline{\delta}_\delta$ curves$^2$ of KDE-based models and QR-based models show similar patterns respectively, and QR-E-copula and JDAN-NFN perform much better than others. For the skill score, JDAN-NFN showcases the best performance in this case ($\overline{\delta}_\Sigma = -0.8979$). For the correlation evaluation, $\overline{\delta}_\tau$ were also calculated, and that is 0.0453 for MQR and 0.0302 for QR-E-copula. JDAN-NFN is still the best associated with $\overline{\delta}_\Sigma$ (0.0269) from Table III.

3) Analyses for Case C: As shown in Table III and Fig 3(g), AMISE-MKDE performs much worse than WKDE-R-copula in the reliability evaluation, while they have demonstrated similar performances in previous cases, which is consistent with the “curse of dimensionality” of MKDE mentioned in [22]. Different up and down trends are observed in $\overline{\delta}_\tau$ curves for different models, and JDAN-NFN offers the best $\overline{\delta}_\tau$ (2.12%). As for the sharpness evaluation in Fig 3(h), $\overline{\delta}_\delta$ curves show a “n” shape for each model and higher $\overline{\delta}_\delta$ are observed at the middle on the horizontal axis, thus models tend to have worse sharpness on these corresponding forecasted marginal distributions evaluated with greater $\overline{\delta}_\delta$. Table III tells that nonparametric models, namely, QR-E-copula and JDAN-NFN perform better than others on $\overline{\delta}_\Sigma$. As shown in Fig 3(i), $\overline{\delta}_\Sigma$ curves (with similar definition of $\overline{\delta}_\delta$ curves) show a “u” shape in each model. Thus the same conclusion can be drawn as in sharpness evaluation: models perform worse on $\overline{\delta}_\Sigma$ at the middle on the horizontal axis. From Table III, the highest $\overline{\delta}_\Sigma$ ($-1.365$) comes to JDAN-NFN, indicating its prominent superiority over other models. Overall for “n” curves of $\overline{\delta}_\Sigma$ and “u” curves of $\overline{\delta}_\delta$, evaluation results tell us that density forecast models may better learn the uncertainty information of the forecasting target if the lead time is small or close to the load variation cycle, i.e., one day. For the correlation evaluation, $\overline{\delta}_\tau$

1QR-based models only offer quantiles, so the forecasted PDF or CDF of forecasting targets need to be approximated with fitting techniques through the inverse transform of quantile functions, which introduces nonnegligible biases in conditional probability distribution computation.

2The curve assembled by all $\overline{\delta}_\delta$. 

| Case A | Case B | Case C |
|--------|--------|--------|
| $b_\tau \%$ | $\overline{\delta}_\tau$ | $\overline{\delta}_\Sigma$ | $V S_\tau$ | $b_\tau \%$ | $\overline{\delta}_\tau$ | $\overline{\delta}_\Sigma$ | $V S_\tau$ | $b_\tau \%$ | $\overline{\delta}_\tau$ | $\overline{\delta}_\Sigma$ | $V S_\tau$ |
| Persistence | $\overline{\delta}_\tau$ | $\overline{\delta}_\Sigma$ | $V S_\tau$ | $\overline{\delta}_\tau$ | $\overline{\delta}_\Sigma$ | $V S_\tau$ | $\overline{\delta}_\tau$ | $\overline{\delta}_\Sigma$ | $V S_\tau$ |
| 4.64 | 0.0592 | -1.034 | 0.0529 | 4.78 | 0.0676 | -1.556 | 0.1228 | 3.92 | 0.0885 | -1.938 | 0.9732 |
| VARX | 2.40 | 0.0468 | -0.860 | 0.0395 | 4.32 | 0.0772 | -1.390 | 0.0827 | 3.67 | 0.1106 | -1.887 | 0.7650 |
| VCG-t-copula | 2.89 | 0.0463 | -0.846 | 0.0227 | 2.39 | 0.0683 | -1.387 | 0.0789 | 3.06 | 0.0911 | -1.884 | 0.6473 |
| WKDE-R-copula | 2.04 | 0.0483 | -0.822 | 0.0121 | 2.61 | 0.0905 | -1.370 | 0.0469 | 2.26 | 0.1087 | -1.746 | 0.5407 |
| Gaussian-E-copula | 1.95 | 0.0350 | -0.817 | 0.0961 | 2.15 | 0.0642 | -1.338 | 0.0356 | 2.92 | 0.0734 | -1.458 | 0.4145 |
| AMISE-MKDE | 1.99 | 0.0559 | -0.833 | 0.0084 | 2.60 | 0.0915 | -1.383 | 0.0454 | 3.26 | 0.1073 | -1.826 | 0.5405 |
| MQR | 2.59 | 0.0413 | -0.810 | / | 2.55 | 0.0627 | -1.212 | / | 3.48 | 0.0705 | -1.453 | / |
| QR-E-copula | 2.51 | 0.0339 | -0.645 | / | 2.72 | 0.0596 | -1.336 | / | 3.12 | 0.0706 | -1.448 | / |
| JDAN-NFN | 0.79 | 0.0280 | -0.511 | 0.0050 | 1.40 | 0.0521 | -0.8979 | 0.0269 | 2.12 | 0.0787 | -1.365 | 0.3167 |

Table III: Performance Demonstration

\[
\begin{align*}
\text{Persistence} & : \overline{\delta}_\tau = \frac{1}{J} \sum_{j=1}^{J} |\hat{b}_\tau^{j, d}|, \quad \text{average width of PIs across all nominal proportions} \\
\overline{\delta}_\Sigma & : \overline{\delta}_\Sigma = \frac{1}{J} \sum_{j=1}^{J} \overline{\delta}_\Sigma^{j, d}, \quad \text{average corresponding skill score} \\
V S_\tau & : V S_\tau = \frac{1}{J} \sum_{j=1}^{J} V S_\tau^{j, d}, \quad \text{variogram scores}
\end{align*}
\]
for MQR and QR-E-copula are 0.3462 and 0.5420, respectively. JDAN-NFN provides the lowest $\tau$ (0.3167), showing the best performance under the variogram-based scoring rule.

Summarizing analyses in case $A$, $B$, and $C$, for the overall evaluation of marginal distributions, nonparametric models generally perform well than parametric ones. They have made a better trade-off between the reliability and sharpness. On the other hand, deep learning-based methods, i.e., Gaussian-E-copula, QR-E-copula, and JDAN-NFN, offer better comprehensive evaluation results overall compared with those without deep learning, which benefits from the powerful capability of NNs for approximating probability distributions of forecasting targets. From these results, we see advantages from both modeling the probability distribution based on nonparametric methods and deep learning. Thus, it is reasonable to introduce deep learning to nonparametric multivariate density forecast, which is consistent with our proposed algorithm. JDAN-NFN takes both the advantage of the nonparametric method and deep learning, so it has exhibited very competitive performance on marginal distributions compared with the existing counterparts.

In terms of correlations, our proposed end-to-end framework, i.e., JDAN-NFN, shows much more preferable results in dependence structure modeling measured by variogram scores than existing multivariate alternatives in all cases. Here, the dependence structure was determined by time-variant parameters in the correlation layer, and these parameters were determined with MLE via gradient descent. Thus, JDAN-NFN provides much more flexibility for finding a better likelihood through this one-step end-to-end manner compared with the try-and-error in two-step copula-based methods, which is a more powerful way to model the dependence structure. To show the efficacy of our method more directly, we display the improvement of JDAN-NFN in terms of the skill score $S_{\tau}$ and variogram score $\tau$ compared to the existing worst and best multivariate methods here in Table IV, which elucidates significant enhancements JDAN-NFN brings in the ability to model marginal distributions and dependence structures. Moreover, to further improve the performance of JDAN-NFN, one can extend $X_t$ with extra

| Case | Reliability evaluation | Sharpness evaluation | Skill score evaluation |
|------|------------------------|----------------------|------------------------|
| $A$  | ![Reliability](image1)  | ![Sharpness](image2) | ![Skill score](image3) |
| $B$  | ![Reliability](image4)  | ![Sharpness](image5) | ![Skill score](image6) |
| $C$  | ![Reliability](image7)  | ![Sharpness](image8) | ![Skill score](image9) |

Fig. 3. Reliability, sharpness, and skill score evaluations for forecasted marginal distributions. (a), (b), and (c) are reliability, sharpness, and skill score evaluations for case $A$. (d), (e), and (f) are reliability, sharpness, and skill score evaluations for case $B$. (g), (h), and (i) are reliability, sharpness, and skill score evaluations for case $C$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
TABLE IV

| Case | $+\tau S_r$ | $+\tau S_r$ | $+\tau S_r$ | $+\tau S_r$ | $+\tau S_r$ | $+\tau S_r$ |
|------|-------------|-------------|-------------|-------------|-------------|-------------|
| A    | 50.6%       | 50.5%       | 42.3%       | 42.3%       | 78.1%       | 78.1%       |
| B    | 90.5%       | 90.5%       | 52.3%       | 78.1%       | 29.6%       | 24.4%       |
| C    | 42.3%       | 78.1%       | 67.5%       | 67.5%       | 57.6%       | 7.0%        |

$+\tau S_r$ denotes the ratio of the result of the skill score of the worst or best alternative minus that of JDAN-NFN to the skill score of the corresponding alternative, and the same way for $+\tau S_r$ about the variogram score.

useful information in our framework conveniently. Meanwhile, the trained JDAN-NFN generates joint CDFs very quickly, since only the feedforward calculation is needed. Specifically, average computation times of the joint CDF at each time spot in three cases are only 0.016 s, 0.018 s, and 0.049 s, respectively, which are significantly shorter than their lead times (5 min, 5 min, and 1 d, respectively). So it is suitable for online applications.

C. Comparing JDAN-NFN With Univariate Approaches

We further compared our proposal with state-of-the-art deep learning-based univariate benchmarks for a more comprehensive evaluation. Here, univariate forecast models for each marginal distribution were trained separately in each case study, and evaluation of these marginals was implemented under average indices $\bar{b}_r$, $\bar{\tau}_r$, and $\bar{S}_r$ as we did in the evaluation of multivariate models. Since the marginal distributions predicted by the univariate models are independent, the average variogram score $\bar{V}_r S_r$ for them was directly calculated following the two steps shown in (24)–(27) based on the marginal distribution as opposed to the conditional one.

The above evaluation results are presented in Table V, and we do not go into very detail in analyses of these results in each case due to the space limitation. As a whole, similar to the evaluation of multivariate approaches, nonparametric univariate methods generally perform well than parametric ones in terms of evaluation indices in all cases. For two of deep learning-based parametric models, i.e., Ensemble CNN and Bayesian LSTM, the $\bar{S}_r$ of them are much higher than those of other deep learning-based univariate methods and offer larger deviations than several multivariate approaches in Table III without deep learning, e.g., VARX, VCG-t-copula, and WKDE-R-copula, in almost all case A, B, and C. From this, we see that deep learning not always contributes to the performance of density forecast models. Moreover, we notice that probability distributions were formulated in different ways in these univariate approaches: 1) The deterministic forecast model was established first, then the expectation and variance of the forecasted probability distribution were obtained via ensemble (Ensemble CNN) or dropout (Bayesian LSTM) of the deterministic forecast model under Gaussian distribution. 2) The expectation and variance of the probability distribution were directly predicted by the probabilistic forecast model under Gaussian distribution (DeepAR). 3) The whole probability distribution (or multiple quantiles) was directly approximated in a nonparametric way (QRNN and DAN-NFN). Therefore, in deep learning-based density forecast models, directly approximating the whole probability distribution in a nonparametric way more fully exploits the power of deep learning than directly predicting parametric statistics or obtaining these statistics based on deterministic forecast models, which substantiates our proposal for introducing deep learning into the nonparametric multivariate density forecast.

Our proposed multivariate density forecast model, i.e., JDAN-NFN, exhibits a greater variogram score $\bar{V}_r S_r$ in all cases since the dependence structure between random variables in the forecasting target was captured in the proposed framework but not in the univariate approaches. This was materialized via the correlation layer. Besides, JDAN-NFN still demonstrates competitive performance compared with univariate ones in terms of the evaluation of marginal distributions overall. Interestingly we see that evaluation indices associated with marginal distributions, i.e., $b_r$, $\tau_r$, and $S_r$, for JDAN-NFN are significantly better than those of other univariate approaches in case A. This lights us that capturing the dependence structure between random variables may help improve the quality of their marginal distributions in the multivariate density forecast. In cases B and C, JDAN-NFN shows a slight performance degradation compared with the best univariate approach by evaluating their marginal distributions. The reason for the degraded performance of JDAN-NFN may be that the high complexity of the high-dimensional joint probability distribution makes it difficult for the model to approximate the real joint probability distribution. Nonetheless, we may still benefit from our model by dealing with the problem in two ways: 1) Due to the significant improvement of our framework for modeling low-dimensional joint probability distributions, we can transform the modeling of a single high-dimensional joint probability distribution into the modeling of multiple two-dimensional joint probability distributions since we mainly focus on the correlation between two random variables. 2) The design of the correlation layer in our framework may not be unique, and we can investigate another way for the design of the correlation layer to fulfill the better modeling of the dependence structure. Results from these state-of-the-art univariate approaches calibrate the superiority of JDAN-NFN.

D. Prediction Results of the Proposed Method

Due to the space limitation and that high dimensional probability distribution can not be visualized, only the forecasted
two-dimensional joint PDFs of case A, i.e., wind speeds joint PDF forecast, are presented here.

The wind speeds in latitudinal and longitudinal directions form the wind speed vector, which is demonstrated in Fig. 4. Several samples were randomly selected in the testing data set to verify the effectiveness of JDAN-NFN as shown in subfigures of Fig. 5. These forecasted joint PDFs are in the form of heat maps, in which the green line represents the observed wind speed vector and the darker color indicates higher probability density. As shown in Fig. 5, most of the observations lie on areas with high probability density, thus JDAN-NFN has satisfactorily estimated the real joint PDF of forecasting targets. Distinct correlations are also demonstrated between wind speeds in different directions as the forecasted joint PDFs in Fig. 5 are asymmetric along both latitudinal and longitudinal directions. Moreover, Some of the forecasted joint PDFs demonstrate a multi-modal characteristic obviously, e.g., in Fig. 5(e), (f), (g), and (h). This phenomenon means that even equipped with the knowledge learned from the whole training set, the JDAN-NFN is still unable to provide an unambiguous direction on the future wind speed vector. Instead, more than one area with high densities are offered by JDAN-NFN. This can be quite favorable for making optimal decisions.

Table V: Performance Demonstration

| Method          | Case A | Case B | Case C |
|-----------------|--------|--------|--------|
|                 | $\tilde{\kappa}$ | $\tilde{\sigma}_x$ | $\tilde{\sigma}_y$ | $\tilde{\kappa}$ | $\tilde{\sigma}_x$ | $\tilde{\sigma}_y$ | $\tilde{\kappa}$ | $\tilde{\sigma}_x$ | $\tilde{\sigma}_y$ |
| Ensemble CNN    | 4.98   | 0.0538 | -0.851 | 0.0084 | 4.58   | 0.0889 | -1.367 | 0.0512 | 5.34   | 0.0808 | -1.377 | 0.4451 |
| Bayesian LSTM   | 3.26   | 0.0481 | -0.813 | 0.0081 | 4.24   | 0.0820 | -1.283 | 0.0501 | 5.16   | 0.0909 | -1.389 | 0.4325 |
| DeepAR          | 1.56   | 0.0384 | -0.794 | 0.0067 | 1.68   | 0.0558 | -1.179 | 0.0479 | 2.70   | 0.0672 | -1.223 | 0.3784 |
| QRNN            | 1.40   | 0.0364 | -0.699 | 0.0064 | 1.57   | 0.0515 | -1.042 | 0.0364 | 2.59   | 0.0557 | -1.141 | 0.3744 |
| DAN-NFN         | 1.29   | 0.0401 | -0.755 | 0.0056 | 1.26   | 0.0318 | -0.8483| 0.0298 | 1.79   | 0.0592 | -1.132 | 0.3789 |
| JDAN-NFN        | 0.79   | 0.0280 | -0.511 | 0.0050 | 1.40   | 0.0521 | -0.8979| 0.0269 | 2.12   | 0.0787 | -1.365 | 0.3167 |

* denotes parametric approaches;  denotes nonparametric approaches

Fig. 5: Heat maps of forecasted joint PDFs and observations in case A.

Fig. 6: Forecasted marginal PDFs in case A.
based on the forecasted joint PDFs, because more than one plan can be made according to this multi-modal characteristic, and these plans serve as backups to each other.

Furthermore, forecasted marginal PDFs of wind speeds along latitudinal and longitudinal directions over 1500 successive time spots from the testing data set are presented in Fig. 6(a) and (b), respectively. The colormap on the right denotes the probability density, where darker color means greater density while lighter color means smaller density. The green solid lines in each subfigure denote corresponding observations of wind speeds in different directions. It is observed that when the wind speeds fluctuate dramatically, e.g., the time spots counted from 900 to 1350, the forecasted marginal PDF may possess long tails (along longitudinal direction) or show multi-modal characteristics (along latitudinal direction), which shows similar phenomena with the scenarios presented in Fig. 5(e), (f), (g), and (h). On the whole, in all the subfigures, most of the observation lies on areas with high probability densities, which illustrates that these forecasted marginal PDFs are very consistent with the reality and verifies the effectiveness of JDAN-NFN as well.

VI. CONCLUSION

In this paper, an effective JDAN-NFN has been developed for multivariate density forecast, which is the first time to our best knowledge to apply deep learning to building a nonparametric multivariate density forecast model in power and energy systems. In JDAN-NFN, the input-output mapping of JDAN approximates joint CDFs of real joint probability distributions and NFN offers the whole parameters of JDAN. The major advantages of JDAN-NFN, compared with previous works, include: 1) the full real joint CDFs of forecasting targets are approximated by a larger-capacity NN, and no more limitation is introduced on the forecast of joint probability distributions, 2) the forecasted joint probability distribution presented by JDAN-NFN considers the dependence structure and time-variant correlations between random variables are learned automatically via gradient descent, and 3) the marginal distribution and correlation between random variables are modeled by a one-step end-to-end framework simultaneously. Three real-world data-based case studies have corroborated the superiority of JDAN-NFN in different forecasting scenarios considering the reliability, sharpness, skill score, and variogram score.

Our future work will focus on the investigation of effective and efficient adaptive learning algorithms since the nonstationary forecasting target may make the performance of the trained model degenerate with time. Viable approaches include incremental learning methods to make the model itself nonstationary or domain adaptation methods to enhance the model’s robustness for nonstationary data.

APPENDIX A

FAILURE OF DAN-NFN IN MODELING THE JOINT PROBABILITY DISTRIBUTION

We change the DAN part in DAN-NFN from a SISO positive-weighted NN to a MISO one to reformulate DAN-NFN in a multivariate setting preparing for the multivariate density forecast. For conciseness, denote \( y = [y_1, y_2, \cdots, y_D] \) as the input to DAN, which represents a vector with \( D \) variables. We construct the mapping of the reformulated DAN (a MISO positive-weighted NN) as

\[
\Psi(y; w^+, b) = z\{w^+_{K+1} \cdots z[w^+_2 \cdot z(w^+_1 \cdot y + b_1) + b_2] \cdots + b_{K+1}\},
\]

where \( K \) is the number of hidden layers. \( z \) is the activation function (sigmoid, tanh, linear, or ReLU), and can be different in different layers. Tensors \( w^+ \) and \( b \) represent all the weights and biases, respectively, which are determined by the outputs of NFN. Tensors \( w^+_k \in w^+ \) and \( b_k \in b, k \in [1, K+1] \), are weights and biases in \( k \)th layer, respectively.

For \( k = 1, \cdots, K \), define the input-output mapping of layer \( k \) in (28) as

\[
Y_{k} = z(w^+_k \cdot Y_{k-1} + b_k)
\]

\[
= \begin{bmatrix}
\vdots \\
\vdots \\
\vdots \\
\end{bmatrix} = \begin{bmatrix}
z_{k_1} \\
z_{k_2} \\
\vdots \\
\end{bmatrix},
\]

\[
Y_{K+1} = z(w^+_{K+1} \cdot Y_{K} + b_{K+1}),
\]

where \( Y_0 = y \). Tensors \( w^+_{k,l} \) and \( b_{k,l} \) represent the \( l \)th row in \( w^+_{K+1} \) and \( b_{K+1} \), respectively. \( z_{k,l} \) is short for \( z(w^+_k \cdot Y_{k-1,l} + b_k) \). Note that \( w^+_k \) and \( b_{K+1} \) in the output layer are vector and scalar, respectively. Then, for \( k=1, \cdots, K \):

\[
\frac{dY_{k}}{dY_{k-1}} = \begin{bmatrix}
z'_{k_1}(w^+_{k,1})^T \\
z'_{k_2}(w^+_{k,2})^T \\
\vdots \\
\end{bmatrix},
\]

\[
\frac{dY_{K+1}}{dY_{K}} = z'(w^+_{K+1} \cdot Y_{K} + b_{K+1})(w^+_{K+1})^T.
\]

Thus,

\[
\frac{d\Psi(y; w^+, b)}{dy} = \prod_{k=1}^{K+1} \frac{dY_{k}}{dY_{k-1}}.
\]

The first derivatives of different activation function \( z \) are

\[
z' = \begin{cases}
\sigma \cdot (1 - \sigma), & \text{if } z \text{ is sigmoid} \\
2\sigma \cdot (1 - \sigma), & \text{if } z \text{ is tanh} \\
1, & \text{if } z \text{ is linear} \\
0 \text{ or } 1, & \text{if } z \text{ is ReLU}
\end{cases}
\]

where \( \sigma \) denotes the sigmoid function. Since \( \sigma \in (0, 1) \), \( z' \) is nonnegative for all kinds of activation functions considered in this paper. Considering that every entry in \( w^+ \) is positive and combining (31)–(34), one can verify that \( d\Psi(y; w^+, b)/dy \) is a \( d \)-dimensional vector, and any entry in it is positive. Therefore, \( \Psi(y; w^+, b) \) is multivariate monotonically nondecreasing.

Now, we illustrate that why such multivariate monotonically nondecreasing property can not be extended to the higher-order-derivative form that meets condition (ii). For conciseness, an example is taken when there are two layers in the positive-weighted

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
NN, which can be denoted as

$$
\Gamma = z [w_{2,1} \cdot z (w_{1} \cdot y + b_{1}) + b_{2}].
$$

(35)

Defining $y_{p}$ and $y_{q}$ are arbitrary two entries in $y$, based on the analyses about (31)–(34), the second-order partial derivative of $\Gamma$ with respect to them can be derived as

$$
\frac{\partial^{2} \Gamma}{\partial y_{p} \partial y_{q}} = \left[ z_{1,1}^{p} \cdot w_{1,1}^{q} \cdot w_{1,1}^{q} \cdot \cdots \cdot z_{1}^{q} \cdot w_{1,1}^{q} \cdot w_{1,1}^{q} \cdots \right] \cdot z^{p} \left( Y_{2} + b_{2} \right) \cdot (w_{2}^{q})^{T} + \left[ z_{1,1}^{p} \cdot w_{1,1}^{q} \cdot \cdots \cdot z_{1,1}^{p} \cdot w_{1,1}^{q} \cdots \right] \cdot (w_{2}^{q})^{T}
$$

(36)

where $w_{1,1}(w_{1,1,1})$ is the element at $i$th row and $p$th ($q$th) column in $w_{1}^{q}$. The second derivatives of different activation function $z$ are

$$
z'' = \begin{cases} 
\sigma \cdot (1 - \sigma) \cdot (1 - 2\sigma), & \text{if } z \text{ is sigmoid} \\
2\sigma \cdot (1 - \sigma) \cdot (1 - 2\sigma), & \text{if } z \text{ is tanh} \\
0, & \text{if } z \text{ is linear or ReLU}
\end{cases}
$$

(37)

It shows that the nonnegative property does not always hold for $z''$. For sigmoid or tanh activation function, $z''$ will be negative if the intermittent computing result is greater than zero when doing forward or backward propagation in the network, which is observed very commonly. Although $z'$ can be nonnegative all the time for ReLU or linear activation function, the NN still could learn nothing because the gradients are always zero. Therefore, combining (36) and (37), condition (ii) can not be guaranteed.

A simple idea is to find a very special activation function so that

$$
z' \geq 0, z'' \geq 0, \cdots, z^{(D)} \geq 0,
$$

(38)

which ensures condition (ii). One activation function satisfying (38) is the exponential function ($e^{x}$). However, it is rarely used in NNs as exponential function suffers from vanishing/exploding gradient problems easily.

Based on the analyses above, simply replacing DAN in the original DAN-NFN framework as a MISO positive-weighted NN can not be used to represent the joint probability distribution.

APPENDIX B
PROOF OF THE NONNEGATIVITY OF (12)

The proof is derived by induction on the value of $D$, i.e., the number of random variables in forecasting targets. For conciseness, denoting $y = [y_{1}, y_{2}, \cdots, y_{D}]$ as the input to JDAN, the output of $d$th parallel unit after the corresponding normalization layer can be reformulated from (9) to

$$
\Psi^{d} = \frac{\Psi^{d}(y^{d}; W^{d+}, B^{d}) - \Psi^{d}(L_{d}; W^{d+}, B^{d})}{\Psi^{d}(U_{d}; W^{d+}, B^{d}) - \Psi^{d}(L_{d}; W^{d+}, B^{d})}.
$$

(39)

In a base case of $D = 2$ and $y = [y^{1}, y^{2}]$, the input-output mapping of JDAN, denoted as $\Psi^{1}^{J}$, can be represented as

$$
\Psi^{1,2} = \frac{\Psi^{1} \cdot \Psi^{2}}{[C_{12} \cdot (1 - \Psi^{1}) \cdot (1 - \Psi^{2}) + 1]}.
$$

(40)

Then, the second-order partial derivative of $\Psi^{1,2}$ with respect to $[y^{1}, y^{2}]$ can be derived as

$$
\frac{\partial^{2}(\Psi^{1,2})}{\partial y^{1} \partial y^{2}} = \frac{\partial \Psi^{1}}{\partial y^{1}} \cdot \frac{\partial \Psi^{2}}{\partial y^{2}} \cdot [C_{12} \cdot (1 - 2\Psi^{1}) \cdot (1 - 2\Psi^{2}) + 1].
$$

(41)

Since $\Psi^{1}, \Psi^{2} \in [0, 1]$, $C_{12} \in (-1, 1)$, and $\frac{\partial \Psi^{1}}{\partial y^{1}}, \frac{\partial \Psi^{2}}{\partial y^{2}} \geq 0$, we have $\frac{\partial^{2}(\Psi^{1,2})}{\partial y^{1} \partial y^{2}} \geq 0$. Note that this nonnegative property holds for any two random variables.

Next, for $D = m$, the mapping of JDAN, denoted as $\Psi^{J}_{m}$, is given as

$$
\Psi^{J}_{m} = \prod_{d=1}^{m} \frac{\Psi^{d} \cdot \prod_{i > d} \sum_{i=1}^{m-1} [C_{di} \cdot (1 - \Psi^{i}) \cdot (1 - \Psi^{i}) + 1]}{\prod_{d=1}^{m} \prod_{i > d} \sum_{i=1}^{m-1} [C_{di} \cdot (1 - \Psi^{i}) \cdot (1 - \Psi^{i}) + 1],}
$$

(42)

where $g(\Psi) = \prod_{d=1}^{m} \frac{\Psi^{d}}{\Psi^{d}}, h(\Psi) = \sum_{d=1}^{m} \sum_{i=1}^{m-1} [C_{di} \cdot (1 - \Psi^{d}) \cdot (1 - \Psi^{d}) + 1]$, and we omit the constant coefficient $\frac{1}{(m-2)}$ for brevity.

Assuming that the nonnegative property holds for $D = m$, $m > 2$, we have

$$
\frac{\partial^{m}(\Psi^{J}_{m})}{\partial y^{1} \cdots \partial y^{m}} \geq 0
$$

(43)

Now, for $D = m + 1$, the mapping of JDAN can be inferred as

$$
\Psi^{J}_{m+1} = g(\Psi) \cdot \Psi^{m+1}.
$$

$$
\begin{aligned}
\Psi^{m+1} = & \Psi^{m} + \frac{\sum_{d=1}^{m} g(\Psi^{d}) \cdot \Psi^{d} \cdot \Psi^{m+1}}{\sum_{d=1}^{m} g(\Psi^{d}) \cdot \Psi^{d} \cdot \Psi^{m+1}}. \\
= & \Psi^{m} + \sum_{d=1}^{m} \Psi^{d} \cdot \Psi^{d} \cdot \Psi^{m+1}.
\end{aligned}
$$

(44)

The higher-order partial derivative of $\Psi^{J}_{m+1}$ with respect to $m + 1$ random variables can be derived as

$$
\frac{\partial^{m+1}(\Psi^{J}_{m+1})}{\partial y^{1} \cdots \partial y^{m} \partial y^{m+1}} = \frac{\partial^{m}(\Psi^{J}_{m})}{\partial y^{1} \cdots \partial y^{m}} \cdot \frac{\partial \Psi^{m+1}}{\partial y^{m+1}} + \sum_{d=1}^{m} \sum_{i \neq d} \frac{\partial \Psi^{i, m+1}}{\partial y^{i}} \cdot \frac{\partial^{2} \Psi^{i, m+1}}{\partial y^{i} \partial y^{m+1}}.
$$

(45)
Combining (41) and (43), we have
\[ \frac{\partial}{\partial y_m} \left( \frac{\partial}{\partial y_{m+1}} \right) \geq 0, \]
and we can conclude that the nonnegative property still holds for $D = m + 1$.

Therefore, we have proved the nonnegativity of (12) for $D \geq 2$.

REFERENCES

[1] IRENA, “Renewable energy statistics 2020,” The International Renewable Energy Agency, Abu Dhabi, 2020. Accessed: Dec. 21, 2021. [Online]. Available: https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2020/Jul/IRENA_Renewable_energy_statistics_2020.pdf

[2] Agora Energiewende and Ember, “The European power sector in 2020: Up-to-date analysis on the electricity transition,” 2021. Accessed: Dec. 21, 2021. [Online]. Available: https://static.agora-energiewende.de/fileadmin/Projekte/2021/2020_01_EU-Annual-Review_2020/A-EW_202_Report_European-Power-Sector-2020.pdf

[3] J. Wang, X. Chen, F. Zhang, F. Chen, and Y. Xin, “Building load forecasting using deep neural network with efficient feature fusion,” J. Modern Power Syst. Clean Energy, vol. 9, no. 1, pp. 160–169, Jan. 2021.

[4] H. Jahangir, H. Tayarani, S. S. Gougheri, M. A. Golkar, A. Ahmadian, and A. Elkamel, “Deep learning-based forecasting approach in smart grids with microclustering and bidirectional LSTM network,” IEEE Trans. Ind. Electron., vol. 68, no. 9, pp. 8298–8309, Sep. 2021.

[5] W. Zhang, H. Quan, and D. Srinivasan, “An improved quantile regression neural network for probabilistic load forecasting,” IEEE Trans. Smart Grid, vol. 10, no. 4, pp. 4425–4434, Jul. 2019.

[6] H. Wang et al., “Deep learning based ensemble approach for probabilistic power forecasting,” IEEE Trans. Ind. Appl., vol. 55, no. 6, pp. 5671–5681, Nov.-Dec. 2018.

[7] M. Afrasiabi, M. Mohammadi, M. Rastegar, L. Stankovic, S. Afrasiabi, and A. Elkamel, “Deep learning-based forecasting approach in smart grids and A. Elkamel, “Deep learning-based forecasting approach in smart grids with copulas,” in Proc. Winter Simul. Conf., 2020, pp. 488–496.

[8] T. Hu, Q. Guo, Z. Li, X. Sun, “Distribution-free probability density forecast through deep neural networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 2, pp. 612–625, Feb. 2020.

[9] G. F. Montufar et al., “On the number of linear regions of deep neural networks,” in Adv. Neural Inf. Process. Syst., 2014, pp. 2924–2932.

[10] H. Daniels and M. Velikova, “Monotone and partially monotone neural networks,” IEEE Trans. Neural Netw., vol. 21, no. 6, pp. 906–917, Jun. 2010.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognition, 2016, pp. 770–778.

[12] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, “LSTM: A search space Odyssey,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[13] S. Ioffe et al., “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. Int. Conf. Mach. Learn., 2015, pp. 448–456.

[14] P. Pinson et al., “Non-parametric probabilistic forecasts of wind power: Required properties and evaluation,” Wind Energy, vol. 10, no. 6, pp. 597–516, Nov./Dec. 2007.

[15] M. Scheuerer et al., “Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities,” Monthly Weather Rev., vol. 143, no. 4, pp. 1321–1334, Apr. 2015.

[16] West Texas MesoNet,” Accessed: Jun. 1, 2021. [Online]. Available: http://meso-file1.tisol.ttu.edu/tech-l/output/mesonet.php

[17] “Actual generation and load data,” Accessed: Dec. 21, 2021. [Online]. Available:https://www.aemo.com.au/energy-systems/electricity/national-electricity-market-nem/data-nem/market-management-systems/data/generation-and-load

[18] “Aggregated price and demand data,” Accessed: Dec. 21, 2021. [Online]. Available: https://www.aemo.com.au/energy-systems/electricity/national-electricity-market-nem/data-nem/aggregated-data

[19] C. Genest et al., “Everything you always wanted to know about copula modeling but were afraid to ask,” J. Hydrologic Eng., vol. 12, no. 4, pp. 347–368, Jul. 2007.

[20] J. K. Møller et al., “Time-adaptive quantile regression,” Comput. Statist. Data Anal., vol. 52, no. 3, pp. 1292–1303, Jan. 2008.

[21] Y. Zhang et al., “Review on probabilistic forecasting of wind power generation and saved energy,” Renew. Sustain. Energy Rev., vol. 32, pp. 255–270, Apr. 2014.

[22] W. Gregor, “Are copula-GoF-tests of any practical use? Empirical evidence for stocks, commodities and FX futures,” Quart. Rev. Econ. Finance, vol. 51, no. 2, pp. 173–188, May 2011.

[23] T. Duong et al., “Plug-in-boundary methods for bivariate kernel density estimation,” J. Nonparametric Statist., vol. 15, no. 1, pp. 17–30, Oct. 2010.

[24] N. Langrené et al., “Fast and stable multivariate kernel density estimation,” J. Comput. Graphical Statist., vol. 28, no. 3, pp. 596–608, Nov. 2018.

[25] S. B. Taieb, R. Huser, R. J. Hyndman, and M. G. Genton, “Forecasting uncertainty in electricity smart meter data by boosting additive quantile regression,” IEEE Trans. Smart Grid, vol. 7, no. 5, pp. 2448–2455, Sep. 2016.

[26] J. Gamboa, “Deep learning for time-series analysis,” 2017, arXiv: 1701.01887v1. [Online]. Available: https://arxiv.org/abs/1701.01887

[27] K. Tsumoto et al., “Time series forecasting using a deep belief network with restricted Boltzmann machines,” Neurocomputing, vol. 137, pp. 47–56, 2014.

[28] Y. Tian et al., “LSTM-based traffic flow prediction with missing data,” Neurocomputing, vol. 318, pp. 297–305, 2018.

[29] E. Musumeci et al., “Large-scale multivariate forecasting models for Dengue-LSTM versus random forest regression,” Spatial Spatio-Temporal Epidemiol., vol. 35, Nov. 2020, Art. no. 100372.

[30] S. Shih et al., “Temporal pattern attention for multivariate time series forecasting,” Mach. Learn., vol. 108, pp. 1421–1441, Jun. 2019.

[31] S. DU et al., “Multivariate time series forecasting via attention-based encoder-decoder framework,” Neurocomputing, vol. 388, pp. 269–279, May 2020.
Zichao Meng (Graduate Student Member, IEEE) received his B.E. and M.S. degrees in electrical engineering from North China Electric Power University, Beijing, China, in 2017 and 2020, respectively. He is currently working towards the Ph.D. degree of electrical engineering in Tsinghua-Berkeley Shenzhen Institute, Tsinghua University. His research interests include probabilistic forecast and meta-learning-based optimization in power and energy systems.

Ye Guo (Senior Member, IEEE) received his B.E. and Ph.D. degrees from the Department of Electrical Engineering, Tsinghua University, Beijing, China, in 2008 and 2013, respectively. He is currently an Associate Professor with Tsinghua-Berkeley Shenzhen Institute, Tsinghua University. From 2014 to 2018, he was a Postdoctoral Associate with Cornell University. His research interests include distributed optimization, game and market theory, state estimation, and their applications in power and energy systems. Between 2019 and 2021, he won the IEEE PES General Meeting Best Paper Award four times in three years in a row, with one co-authored paper being selected as the ‘Best-of-the-Best Papers’. He has also received the Best Poster Award at PSERC LAB Meeting 2018, and the Best Paper Award at the 4th IEEE EI2 Conference.

Wenjun Tang (Member, IEEE) received the M.S. degree and the Ph.D. degree in electrical engineering from National Cheng-Kung University (NCKU), Tainan, Taiwan, in 2015 and 2019, respectively. She was the Postdoctoral Associate with Tsinghua-Berkeley Shenzhen Institute from 2019 to 2021. She is currently working for China Southern Power Grid. Her interests include electricity markets, demand-size management, and the application of deep learning in smart grids and virtual power plants. She is a member of the Phi Tau Phi, NGN-CIGRE, and the IEEE Power and Energy Society.

Hongbin Sun (Fellow, IEEE) received the B.S. and Ph.D. degrees from the Department of Electrical Engineering, Tsinghua University, Beijing, China, in 1992 and 1996, respectively. He is currently the ChangJiang Scholar Chair Professor with the Department of Electrical Engineering and the Director of Energy Management and Control Research Center, Tsinghua University. He has authored or coauthored more than 60 are IEEE and IET journal papers, and four books. He has been granted five U.S. Patents of Invention and more than 100 Chinese patents of invention. His research interests include electric power system operation and control and energy system integration. He is also the Editor of the IEEE TRANSACTIONS ON SMART GRID, an Associate Editor for the IET Renewable Power Generation, and the Member of the Editorial Board of four international journals and several Chinese journals. He is the IET Fellow.