

Learning Interpretable Deep State Space Model for Probabilistic Time Series Forecasting

Longyuan Li^{1,2}, Junchi Yan^{2,3*}, Xiaokang Yang^{2,3} and Yaohui Jin^{1,2*}

¹State Key Lab of Advanced Optical Communication System and Network

²MoE Key Lab of Artificial Intelligence, AI Institute

³Department of Computer Science and Engineering

Shanghai Jiao Tong University

{jefli, yanjunchi, xkyang, jinyh}@sjtu.edu.cn

Abstract

Probabilistic time series forecasting involves estimating the distribution of future based on its history, which is essential for risk management in downstream decision-making. We propose a deep state space model for probabilistic time series forecasting whereby the non-linear emission model and transition model are parameterized by networks and the dependency is modeled by recurrent neural nets. We take the automatic relevance determination (ARD) view and devise a network to exploit the exogenous variables in addition to time series. In particular, our ARD network can incorporate the uncertainty of the exogenous variables and eventually helps identify useful exogenous variables and suppress those irrelevant for forecasting. The distribution of multi-step ahead forecasts are approximated by Monte Carlo simulation. We show in experiments that our model produces accurate and sharp probabilistic forecasts. The estimated uncertainty of our forecasting also realistically increases over time, in a spontaneous manner.

1 Introduction and Related Works

1.1 Background and Motivation

Time series forecasting is a long-standing problem in literature, which has attracted extensive attention over the decades. For downstream decision-making applications, one important feature for a forecasting model is to issue long-term forecasts with effective uncertainty estimation. Meanwhile the relevant factors, in the form of exogenous variables in addition to the time series data, can be robustly identified, to improve the interpretation of the model and results.

To address these challenges, in this paper, we develop an approach comprising a state space based generative model and a filtering based inference model, whereby the non-linearity of both emission and transition models is achieved

*Yaohui Jin and Junchi Yan are corresponding authors. This work is supported by National Key Research and Development Program of China (2018YFC0830400), (2016YFB1001003), STCSM(18DZ1112300).

by deep networks and so for the dependency over time by recurrent neural nets. Such a network based parameterization provides the flexibility to fit with arbitrary data distribution. The model is also tailored to effectively exploit the exogenous variables along with the time series data, and the multi-step forward forecasting is fulfilled by Monte Carlo simulation. In a nut shell the highlights of our work are:

1) We present a novel deep state space model for probabilistic time series forecasting, which can i) handle arbitrary data distribution via nonlinear network parameterization based on the proposed deep state space model; ii) provide interpretable forecasting by incorporating extra exogenous variable information using the devised ARD network; iii) uncertainty modeling of the exogenous variables to improve the robustness of variable selection. We believe these techniques as a whole are indispensable for a practical forecasting engine and each of them may also be reused in other pipelines.

2) In particular, by taking the automatic relevance determination (ARD) view [Wipf and Nagarajan, 2008], we devise an ARD network which can estimate exogenous variables' relevance to the forecasting task. We further consider and devise the Monte Carlo sampling method to model the uncertainty of the exogenous variables. It is expected (and empirically verified) to capture interpretable structures from time series which can also benefit long-term forecast accuracy.

3) We conduct experiments on a public real-world benchmark, showing the superiority of our approach against state-of-the-art methods. In particular, we find our approach can spontaneously issue forecasting results with reasonable growing uncertainty over time steps and help uncover the key exogenous variables relevant to the forecasting task.

1.2 Related Works

Time Series Forecasting

Classical models such as auto-regressive (AR) and exponential smoothing [Brockwell *et al.*, 2002] have a long history for forecasting. They incorporate human priors about time series structure such as trend, seasonality explicitly, and thus have difficulty with diverse dependency and complex structure. With recent development of deep learning, RNNs have been investigated in the context of time series *point* forecasting, the work [Långkvist *et al.*, 2014] reviews deep learning methods for time series modelling in various fields. With their

capability to learn non-linear dependencies, deep nets can extract high order features from raw time series. Although the results are seemingly encouraging, deterministic RNNs lack the ability to make probabilistic forecasts. [Flunkert *et al.*, 2017] proposes DeepAR, which employs an auto-regressive RNN with mean and standard deviation as output for probabilistic forecasts.

Variational Auto-Encoders (VAEs)

Deep generative models are powerful tools for learning representations from complex data [Bengio *et al.*, 2013]. However, posterior inference of non-linear non-Gaussian generative models is commonly recognized as intractable. The idea of variational auto-encoder [Kingma and Welling, 2013] is to use neural networks as powerful functional approximators to approximate the true posterior, combined with reparameterization trick, learning of deep generative models can be tractable.

State Space Models (SSMs)

SSMs provide a unified framework for modelling time series data. Classical AR models e.g. ARIMA can be represented in state space form [Durbin and Koopman, 2012]. To leverage advances in deep learning, [Chung *et al.*, 2015] and [Fraccaro *et al.*, 2016] draw connections between SSMs and VAEs using an RNN. Deep Kalman filters (DKFs) [Krishnan *et al.*, 2015; Krishnan *et al.*, 2017] further allow exogenous input in the state space models. For forecasts, deep state space models (DSSM) [Rangapuram *et al.*, 2018] use an RNN to generate parameters of a linear-Gaussian state space model (LGSSM) at each time step. Another line of work for building non-linear SSMs are Gaussian process state space models (GP-SSMs) [Eleftheriadis *et al.*, 2017]. However the joint Gaussian assumption of GP-SSMs may be restrictive to non-Gaussian data. Moreover, inference at each time step has at least $O(T)$ complexity for the number of past observations even with sparse GP approximations.

Perhaps the most closely related works are DSSM [Rangapuram *et al.*, 2018] and DeepAR [Flunkert *et al.*, 2017]. Compared with DSSM, the emission model and transition model are non-linear, and our model supports non-Gaussian likelihood. Compared with DeepAR, target values are not used as inputs directly in our approach, making it more robust to noise. Also forecasting samples can be generated computationally more efficiently as the RNN only need to be unrolled once for the entire forecast, whereas DeepAR has to unroll the RNN with entire time series for each sample. Moreover, our model supports exogenous variables with uncertainty as input in the forecasting period. Our model learns relevance of exogenous variables automatically, leading to improvement in interpretability and enhanced forecasting performance.

1.3 Preliminaries

Problem Formulation

Consider we have an M -dimensional multi-variate series having T time steps: $\mathbf{x}_{1:T} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T\}$ where each $\mathbf{x}_t \in \mathbb{R}^M$, and $\mathbf{x}_{1:T} \in \mathbb{R}^{M \times T}$. Let $\mathbf{u}_{1:T+\tau}$ be a set of synchronized time-varying exogenous variables associated with $\mathbf{x}_{1:T}$, where each $\mathbf{u}_t \in \mathbb{R}^D$. Our goal is to estimate the distribution of future window $\mathbf{x}_{T+1:T+\tau}$ given its history $\mathbf{x}_{1:T}$ and

extended exogenous variables $\mathbf{u}_{1:T+\tau}$ until the future:

$$p_\theta(\mathbf{x}_{T+1:T+\tau} | \mathbf{x}_{1:T}, \mathbf{u}_{1:T+\tau}) \quad (1)$$

where θ denotes the model parameters. To prevent confusion for *past* and *future*, in the rest of this paper we refer time steps $\{1, 2, \dots, T\}$ as *training period*, and $\{T+1, T+2, \dots, T+\tau\}$ as *forecasting period*. The time step $T+1$ is referred to as *forecast start time*, and $\tau \in \mathbb{N}^+$ is the *forecast horizon*. Note that the exogenous variables $\mathbf{u}_{1:T+\tau}$ are assumed to be known in both training period and forecasting period, and we allow it to be uncertain and follow some probability distribution $\mathbf{u}_t \sim p(\mathbf{u}_t)$ in the forecasting period.

State Space Models (SSMs)

Given observation sequence $\mathbf{x}_{1:T}$ and its conditioned exogenous variables $\mathbf{u}_{1:T}$, a probabilistic latent variable model is:

$$p(\mathbf{x}_{1:T} | \mathbf{u}_{1:T}) = \int p(\mathbf{x}_{1:T} | \mathbf{z}_{1:T}, \mathbf{u}_{1:T}) p(\mathbf{z}_{1:T} | \mathbf{u}_{1:T}) d\mathbf{z}_{1:T} \quad (2)$$

where $\mathbf{z}_{1:T}$ for $\mathbf{z}_t \in \mathbb{R}^{n_z}$ denotes sequence of latent stochastic states. The generative model is composed of an *emission model* $p_\theta(\mathbf{x}_{1:T} | \mathbf{z}_{1:T}, \mathbf{u}_{1:T})$ and a *transition model* $p_\theta(\mathbf{z}_{1:T} | \mathbf{u}_{1:T})$. To obtain state space models, first-order Markov assumption is imposed on the latent states, and then the emission model and transition model can be factorized:

$$p(\mathbf{x}_{1:T} | \mathbf{z}_{1:T}, \mathbf{u}_{1:T}) = \prod_{t=1}^T p(\mathbf{x}_t | \mathbf{z}_t, \mathbf{u}_t) \quad (3)$$

$$p(\mathbf{z}_{1:T} | \mathbf{u}_{1:T}) = \prod_{t=1}^T p(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{u}_t)$$

where the initial state w is assumed to be zero.

2 Proposed Model

In this section, we describe i) the deep state space model architecture including both the emission and transition models; ii) the proposed Automatic Relevance Determination (ARD) network to more effectively incorporate the exogenous variable information; iii) the techniques for model training; iv) the devised Monte Carlo simulation for multi-step probabilistic forecasting that can generate seemingly realistic uncertainty estimation over time. We present our approach using a single multi-variate time series sample for notational simplicity. While in fact batches of sequence samples are used for training.

2.1 Generative Model

We propose a deep state space model, for the term *deep* it is reflected in two aspects: i) We use neural network to parameterize both the non-linear emission model and transition model; ii) We use recurrent neural networks (RNNs) to capture long-term dependencies.

State space models e.g. linear-Gaussian models and hidden Markov models are neither suitable for modelling long-term dependencies nor non-linear relationships. We relax the first-order Markov assumption in Eq. 3 by including a sequence

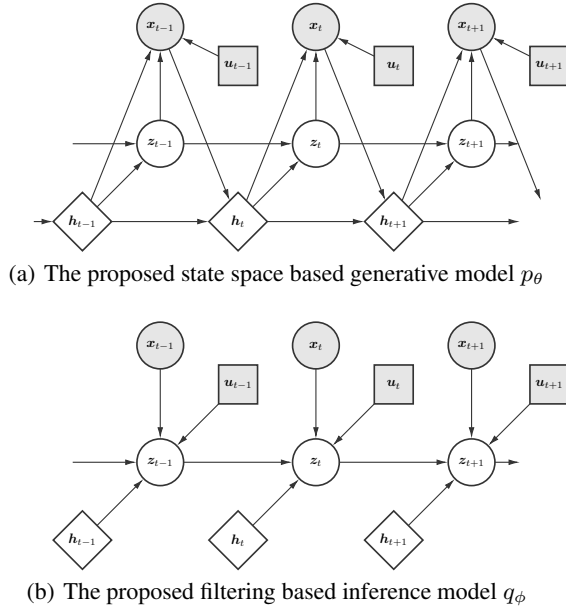


Figure 1: Proposed generative model and inference model. Shaded and white nodes are observations and hidden variables respectively. Circle and diamond denote stochastic states and deterministic states respectively, and shaded square nodes denote exogenous variables.

of deterministic states $\mathbf{h}_{1:T}$ recursively computed by RNNs to capture the long-term dependencies. Here we simplify the model by assuming the state transition involves no exogenous variable¹. The generative model is shown in Fig. 1(a), and the joint probability distribution can be factorized as:

$$p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}, \mathbf{h}_{1:T}) = \prod_{t=1}^T p_{\theta_x}(\mathbf{x}_t | \mathbf{z}_t, \mathbf{h}_t, \mathbf{u}_t) p_{\theta_z}(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{h}_t) p_{\theta_h}(\mathbf{h}_t | \mathbf{h}_{t-1}, \mathbf{x}_{t-1}) \quad (4)$$

We omit initial states $\mathbf{z}_0 = 0$ and $\mathbf{h}_0 = 0$ for brevity, and let $\mathbf{x}_0 = \mathbf{x}_1$ for cold starting. In state space form, our model can be represented as:

$$\mathbf{h}_t \sim \delta(\mathbf{h}_t - \mathbf{f}_{\theta_d}(\mathbf{h}_{t-1}, \mathbf{x}_{t-1})) \quad (5)$$

$$\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{\theta_z}(\mathbf{z}_{t-1}, \mathbf{h}_t), \boldsymbol{\Sigma}_{\theta_z}(\mathbf{z}_{t-1}, \mathbf{h}_t)) \quad (6)$$

$$\mathbf{x}_t \sim \mathcal{P}\mathcal{D}(\mathbf{T}_{\theta_x}(\mathbf{z}_t, \mathbf{h}_t, \mathbf{u}_t)) \quad (7)$$

where $\mathcal{P}\mathcal{D}$ is an arbitrary probability distribution, and its sufficient statistic \mathbf{T}_{θ_x} is parameterized by neural networks θ_x . For stochastic latent states, $\boldsymbol{\mu}_{\theta_z}(\cdot)$ and $\boldsymbol{\Sigma}_{\theta_z}(\cdot)$ mean and covariance functions for Gaussian distribution of state transitions, which are also parameterized by neural networks. For deterministic RNN states, $\mathbf{f}_{\theta_d}(\cdot)$ is the RNN transition function, and $\delta(\cdot)$ is a delta distribution, and \mathbf{h}_t can be viewed as following a delta distribution centered at $\mathbf{f}_{\theta_d}(\mathbf{h}_{t-1}, \mathbf{x}_{t-1})$.

The deep state space model in Eq. 5,6,7 is parameterized by $\theta = \{\theta_x, \theta_z, \theta_h\}$, in our implementation, we use gated

¹In experiments we empirically find using exogenous variables to parameterize the transition model can weaken the performance, which may be due to overfitting and we leave it for future work.

recurrent unit (GRU)[Cho *et al.*, 2014] as transition function $\mathbf{f}_{\theta_d}(\cdot)$ to capture temporal dependency. For stochastic transition, we take the following parameterization:

$$\begin{aligned} \boldsymbol{\mu}_{\theta_z}(t) &= \text{NN}_1(\mathbf{z}_{t-1}, \mathbf{h}_t, \mathbf{u}_t), \\ \boldsymbol{\sigma}_{\theta_z}(t) &= \text{SoftPlus}[\text{NN}_2(\mathbf{z}_{t-1}, \mathbf{h}_t, \mathbf{u}_t)] \end{aligned} \quad (8)$$

where NN_1 and NN_2 denotes two neural networks parameterized by θ_z , and $\text{SoftPlus}[x] = \log(1 + \exp(x))$.

For $\mathbf{T}_{\theta_x}(\mathbf{z}_t, \mathbf{h}_t, \mathbf{u}_t)$, we also use networks to parameterize the mapping. In order to constrain real-valued output $\tilde{y} = \text{NN}(\mathbf{z}_t, \mathbf{h}_t, \mathbf{u}_t)$ to the parameter domain, we use the following transformations:

- real-valued parameters : no transformation.
- positive parameters: the SoftPlus function.
- bounded parameters $[a, b]$: scale and shifted Sigmoid function $y = (b - a) \frac{1}{1 + \exp(-\tilde{y})} + a$

2.2 Structure Inference Network

We are interested in maximizing the log marginal likelihood, or evidence $\mathcal{L}(\theta) = \log p_\theta(\mathbf{x}_{1:T} | \mathbf{u}_{1:T})$, where the latent states and RNN states are integrated out. Integration of deterministic RNN states can be computed by simply substituting the deterministic values. However the stochastic latent states can not be analytically integrated out, because the generative model is non-linear, and moreover the emission distribution may not be conjugate to latent state distribution. We resort to variational inference for approximating intractable posterior distribution [Jordan *et al.*, 1999]. Instead of maximizing $\mathcal{L}(\theta)$, we build a structure inference network q_ϕ parameterized by ϕ , with the following factorization in Fig. 1(b):

$$q_\phi(\mathbf{z}_{1:T}, \mathbf{h}_{1:T} | \mathbf{x}_{1:T}, \mathbf{u}_{1:T}) = \prod_{t=1}^T q_\phi(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{x}_t, \mathbf{h}_t, \mathbf{u}_t) p_{\theta_h}(\mathbf{h}_t | \mathbf{h}_{t-1}, \mathbf{x}_{t-1}) \quad (9)$$

where the same RNN transition network structure with the generative model is used. Note that we let stochastic latent states follow a isotropic Gaussian distribution, where covariance matrix $\boldsymbol{\Sigma}$ is diagonal:

$$\begin{aligned} \boldsymbol{\mu}_\phi(t) &= \text{NN}_1(\mathbf{z}_{t-1}, \mathbf{x}_t, \mathbf{h}_t, \mathbf{u}_t), \\ \boldsymbol{\sigma}_\phi(t) &= \text{SoftPlus}[\text{NN}_2(\mathbf{z}_{t-1}, \mathbf{x}_t, \mathbf{h}_t, \mathbf{u}_t)] \end{aligned} \quad (10)$$

Then we maximize the *variational evidence lower bound (ELBO)* $\mathcal{L}(\theta, \phi) \leq \mathcal{L}(\theta)$ [Jordan *et al.*, 1999] with respect to both θ and ϕ , which is given by:

$$\begin{aligned} \mathcal{L}(\theta, \phi) &= \iint q_\phi \log \frac{p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}, \mathbf{h}_{1:T} | \mathbf{u}_{1:T})}{q_\phi(\mathbf{z}_{1:T}, \mathbf{h}_{1:T} | \mathbf{x}_{1:T}, \mathbf{u}_{1:T})} d\mathbf{z}_{1:T} d\mathbf{h}_{1:T} \\ &= \sum_{t=1}^T \mathbb{E}_{q_\phi(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{x}_t, \mathbf{h}_t, \mathbf{u}_t)} [\log p_{\theta_x}(\mathbf{x}_t | \mathbf{z}_t, \mathbf{h}_t, \mathbf{u}_t)] \\ &\quad - \text{KL}(q_\phi(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{x}_t, \mathbf{h}_t, \mathbf{u}_t) || p_{\theta_z}(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{h}_t)) \end{aligned} \quad (11)$$

where $q_\phi(\mathbf{z}_{1:T}, \mathbf{h}_{1:T} | \mathbf{x}_{1:T}, \mathbf{u}_{1:T})$ is simplified by notation q_ϕ , and KL denotes Kullback-Leibler (KL) divergence. Our

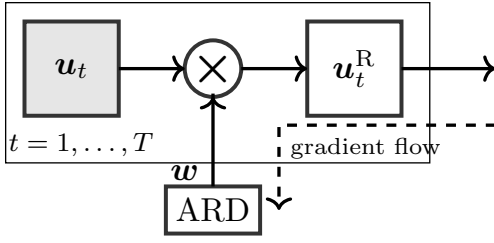


Figure 2: Sketch of the devised automatic relevance determination (ARD) network for modeling associated exogenous variables \mathbf{u} .

model jointly learns parameters $\{\theta, \phi\}$ of the generative model p_θ and inference model q_ϕ by maximizing ELBO in Eq. 11. However, latent states $\mathbf{z}_{1:T}$ are stochastic thus the objective is not differentiable. To obtain differentiable estimation of the ELBO, we resort to Stochastic Gradient Variational Bayes (SGVB): $\mathcal{L}_{SGVB}(\theta, \phi) \simeq \mathcal{L}(\theta, \phi)$, since it provides efficient estimation of ELBO [Kingma and Welling, 2013]. At each time step, instead of directly sampling from $q_\phi(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{x}_t, \mathbf{h}_t, \mathbf{u}_t)$, our model samples from an auxiliary random variable $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and re-parameterizes $\mathbf{z}_t = \boldsymbol{\mu}_\phi(t) + \epsilon \odot \boldsymbol{\sigma}_\phi(t)$. As such, the gradient of the objective with respect to both θ and ϕ can be back-propagated through the sampled \mathbf{z}_t . Formally we have:

$$\begin{aligned} \mathcal{L}_{SGVB} = & \sum_{t=1}^T \log p_{\theta_x}(\mathbf{x}_t | \mathbf{z}_t, \mathbf{h}_t, \mathbf{u}_t) \\ & - \text{KL}(q_\phi(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{x}_t, \mathbf{h}_t, \mathbf{u}_t) || p_{\theta_z}(\mathbf{z}_t | \mathbf{z}_{t-1}, \mathbf{h}_t)) \end{aligned} \quad (12)$$

The SGVB estimation can be viewed as a single-sample Monte Carlo estimation of the ELBO. To reduce ELBO variance and seek a tighter bound, we propose to use multi-sampling variational objective [Mnih and Rezende, 2016; Burda *et al.*, 2015]. It can be achieved by sampling K independent ELBOs and taking the average as objective:

$$\mathcal{L}_{SGVB}^K = \frac{1}{K} \sum_{k=1}^K \mathcal{L}_{SGVB}^{(k)}$$

2.3 Automatic Relevance Determination Network

In real-world, in addition to time series it-self, time series data are often correlated to a variety of exogenous variables depending on specific applications, such as: i) time-features: absolute time (trend), hour-of-day, day-of-week, month-of-year (which can carry daily, weekly, yearly seasonality information); ii) flagged periods: such as national holidays, business hours, weekends; iii) measurements: weather, sensor readings. Those features can be incorporated in the form of exogenous variables. In particular, it is often the case that only a few of such variables are relevant to the forecasting task at hand. Existing methods [Wen *et al.*, 2017; Rangapuram *et al.*, 2018; Flunkert *et al.*, 2017] select variables based on expert prior. To prune away redundant variables, we take a data-driven sparse Bayesian learning (SPL) approach. Consider given variables $\mathbf{u}_{1:T+\tau}$, we let:

$$\boldsymbol{\mu}_t^R = \mathbf{w} \odot \mathbf{u}_t \quad (13)$$

where \mathbf{u}_t^R are relevant variables, \mathbf{w} is weights of input variables that has the same shape with \mathbf{u}_t . To impose sparsity

over variables, we take the so-called automatic relevance determination (ARD) [Neal, 2012] view, where the basic idea is to learn an individual weight parameter \mathbf{w} for each input feature to determine relevance. Techniques based on ARD are applied in many classical models, such as SVMs, Ridge regression, Gaussian Processes [Bishop and Tipping, 2000]. Classical ARD approach is to treat \mathbf{w} as a latent variable placed by some sparsity inducing prior $p(\mathbf{w})$, to avoid complex posterior inference of weights \mathbf{w} , we devise an automatic relevant determination (ARD) network whereby \mathbf{w} is parameterized by a neural network with constant input:

$$\mathbf{w} = \text{SoftMax}(\text{NN}_{\text{ARD}}(\mathbf{I})) \quad (14)$$

where $\text{SoftMax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$, and \mathbf{I} is a constant input vector. We illustrate ARD network in Fig. 2. Note that \mathbf{w} is a global variable, and multiply every \mathbf{u}_t for $t = \{1, \dots, T\}$. The differentiable SoftMax function is used as output layer for multi-class classification as its all outputs sum up to 1. Let $\mathbf{u}_{1:T}^R$ be the input of both generative model and inference model. The gradients of parameters in ARD network can be easily back-propagated within the training process. As such, the ARD network is easy to implement, and it avoids intractable posterior inference of traditional Bayesian approach. We show in experiments that the proposed ARD network not only can learn interpretable structure over time series, but also can improve the accuracy of probabilistic forecasts.

2.4 Forecasting with Uncertainty Modeling for Exogenous Variables

Once the parameters of the generative model θ is learned, we can use the model to address the forecasting problem in Eq. 1. Since the transition model is non-linear, we cannot analytically compute the distribution of future values, so we resort to a Monte Carlo approach.

One shall note that for real-world forecasting, some of the exogenous variables themselves may also be the output by a forecasting model e.g. weather forecast of temperature and rainfall probability. Hence from the forecasting time point, the forward part also contain inherent uncertainty and may not be accurate. Here we adopt a Monte Carlo approach to incorporate such uncertainty into the forecasting process. Let $u_{d,t} \sim p(u_{d,t})$, $d = 1, \dots, D$ be the distribution of each individual exogenous variable $\mathbf{u}_t \in \mathbb{R}^D$ for $t = T + 1 \dots, T + \tau$ (while those variables with no uncertainty e.g. national holiday can be expressed by a delta distribution), and let $\mathbf{u}_t \sim p(\mathbf{u}_t)$ denote the distribution of a collection of exogenous variables at time t .

Before forecasting, we first evaluate the whole time series $\mathbf{x}_{1:T}$ over the model, and compute the distribution of the latent states $p_{\theta_z}(\mathbf{z}_T | \mathbf{z}_{T-1}, \mathbf{h}_T)$ and \mathbf{h}_T for the last time step T in the training period. Then starting from $\mathbf{z}_T \sim p_{\theta_z}(\mathbf{z}_T | \mathbf{z}_{T-1}, \mathbf{h}_T)$, we recursively compute:

$$\begin{aligned} \mathbf{h}_{T+t} &= \mathbf{f}_{\theta_d}(\mathbf{h}_{T+t-1}, \mathbf{x}_{T+t-1}) \\ \mathbf{z}_{T+t} &\sim \mathcal{N}(\boldsymbol{\mu}_{\theta_z}(\mathbf{z}_{T+t-1}, \mathbf{h}_{T+t}), \boldsymbol{\Sigma}_{\theta_z}(\mathbf{z}_{T+t-1}, \mathbf{h}_{T+t})) \\ \mathbf{u}_{T+t} &\sim p(\mathbf{u}_{T+t}) \\ \mathbf{x}_{T+t} &\sim \mathcal{PD}(\mathbf{T}_{\theta_x}(\mathbf{z}_{T+t}, \mathbf{h}_{T+t}, \mathbf{u}_{T+t})) \end{aligned} \quad (15)$$

for $t = 1, \dots, \tau$. Our model generates forecasting samples by applying x . The sampling procedure can be parallelized easily as sampling recursions are independent.

2.5 Data Preprocessing

Given time series $\{\mathbf{x}_1, \dots, \mathbf{x}_T\}$, to generate multiple training samples $\{\mathbf{x}_{1:W}^i\}_{i=1}^N$, we follow the protocol used in DeepAR [Flunkert *et al.*, 2017]: the *shingling* technique [Leskovec *et al.*, 2014] is used to convert a long time series into many short time series chunks. Specifically, at each training iteration, we sample a batch of windows with width W at random start point $[1, T - W - 1]$ from the dataset and feed into the model. In some cases we have a dataset \mathcal{X} of many independent time series with varying length that may share similar features, such as demand forecasting for large number of items. We can model them with a single deep state space model, we build a pool of time series chunks to be randomly sample from, and forecasting is conducted independently for each single time series.

3 Experiments

We first compare our proposed model with baselines in public benchmarks, whereby only time series data is provided without additional exogenous variables. In this case, our devised ARD component is switched off and our model degenerates to traditional time series forecasting model. Then we test our model on datasets with rich exogenous variables to show how the proposed ARD network selects relevant exogenous variables and benefit to make accurate forecasts. Note that we refer to our model as DSSMF.

We implement our model by Pytorch on a single RTX 2080Ti GPU. Let dimension of hidden layer of all NNs in generative model and inference model be 100, and dimension of stochastic latent state \mathbf{z} be 10. We set window size to be one week of time steps for all datasets, and set sample size $K = 5$ of SGVB. We use Adam optimizer [Kingma and Ba, 2014] with learning rate 0.001. Forecasts distribution are estimated by 1000 trials of Monte Carlo sampling. We use Gaussian emission model where mean and standard deviations are parameterized by two-layer NNs following protocol described in Sec. 2.1.

3.1 Forecasting using Exogenous Variables without Uncertainty

Dataset

We use two public datasets *electricity* and *traffic* [Yu *et al.*, 2016]. The *electricity* contains time series of the electricity usage in kW recorded hourly for 370 clients. The *traffic* contains hourly occupancy rate of 963 car lanes of San Francisco bay area freeways. We held-out the last four weeks for forecasting, and other data for training. For both datasets, only time-feature are added as exogenous variables $\mathbf{u}_{1:T+\tau}$ without any uncertainty, including absolute time, hour-of-day, day-of-week, is-workday, is-business-hour.

Baselines

We use a classical method SARIMAX [Box *et al.*, 2015], extending ARIMA that supports modeling seasonal components

datasets	SARIMAX	DeepAR	DSSMF
electricity	1.13(0.00)	0.60(0.01)	0.58(0.01)
traffic	1.58(0.00)	0.89(0.01)	0.86(0.01)

Table 1: Mean (S.D.) CRPS for rolling-week forecast over 4 weeks.

and exogenous variable. Optimal parameters are obtained from grid search using R’s *forecast* package [Hyndman *et al.*, 2007]. We also use a recently proposed RNN-based autoregressive model DeepAR [Flunkert *et al.*, 2017], where we use SDK from Amazon Sagemaker platform² to obtain results. Results of all models are computed using a rolling window of forecasting as described in [Yu *et al.*, 2016], where the window is one week (168 time steps).

Evaluation Metric

Different from deterministic forecasts that predict a single value, probabilistic forecasts estimate a probability distribution. Accuracy metrics such as MAE, RMSE are not applicable to probabilistic forecasts. The *Continuous Ranked Probability Score* (CRPS) generalizes the MAE to evaluate probabilistic forecasts, and is a proper scoring rule [Gneiting and Katzfuss, 2014]. Given the true observation value x and the cumulative distribution function (CDF) of its forecasts distribution F , the CRPS score is given by:

$$CRPS(F, x) = \int_{-\infty}^{\infty} (F(y) - \mathbb{1}(y - x))^2 dy \quad (16)$$

where $\mathbb{1}$ is the Heaviside step function. In order to summarize the CRPS score for all time series with different scales, we compute CRPS score on a series-wise standardized dataset.

The results shown in Table 3.1 show that our model outperforms DeepAR and SARIMAX on two public benchmarks.

3.2 Forecasting using Exogenous Variables with Uncertainty

In the following, we further show the interpretability of our model by incorporating the exogenous information. The hope is that the relevant exogenous variables will be uncovered automatically by our ARD network which suggests their importance to the forecasting task at hand.

We refer to our model as DSSMF, and we also include two variants for ablation study: 1) DSSMF-NX, DSSMF without using exogenous variable. 2) DSSMF-NA, DSSMF without the proposed ARD network, and the exogenous variable are input directly. To validate if the devised ARD network works effectively, we add three time series of Gaussian white noise as extra exogenous variables ($\{noise\ 1, noise\ 2, noise\ 3\}$). Our expectation is that these three noise variable will be automatically down-weighted by the ARD module. All exogenous variables are standardized to unit Gaussian. To mimic the real situation for forecasting where exogenous variable may be uncertain, we let $u_{d,t} \sim \mathcal{N}(\tilde{u}_{d,t}, \sigma_t)$ for all exogenous variables except for time-features i.e. exogenous variables without uncertainty in the forecasting period, where $\tilde{u}_{d,t}$ is the true exogenous variable, and σ_t increases from 0 to 1 linearly in the forecasting period.

²<https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html>

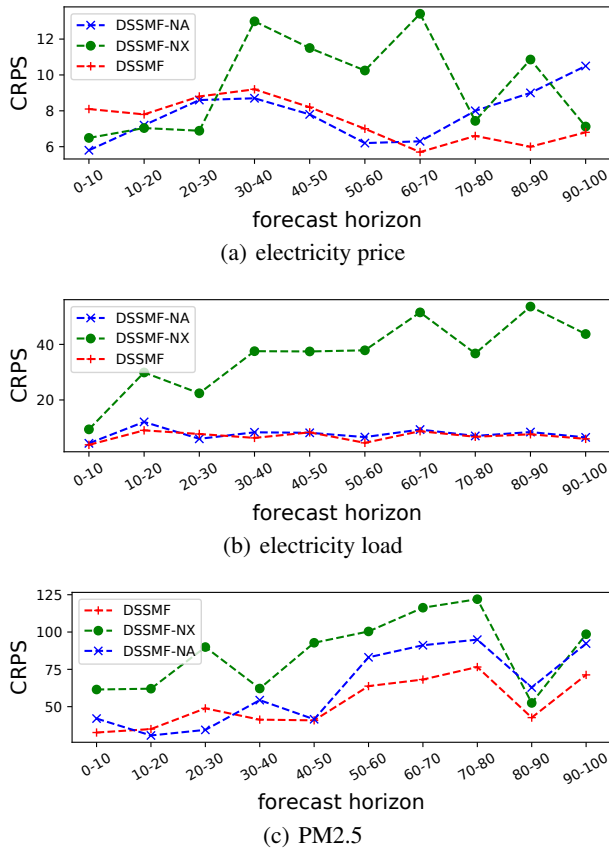


Figure 3: Forecast accuracy w.r.t forecast horizon.

Electricity Price and Load Forecasting

We apply our model to two probabilistic forecasting problems using datasets published in GEFCom2014 [Hong *et al.*, 2016]. We choose this benchmark as it contains rich external information to explore by using the devised ARD network. The first is electricity price forecasting, which involves univariate time series of hourly zonal electricity price. In addition, zonal load forecasts and total system forecast are also available over the forecast horizon. The second task is electricity load forecasting, where the hourly electricity load and air temperature data from 25 weather stations are provided.

Air Quality Forecasting

We apply our model to the Beijing PM2.5 dataset [Liang *et al.*, 2015] for the city suffering heavy air pollution at that time. It contains hourly PM2.5 records together with meteorological data, including dew point, temperature, atmospheric pressure, wind direction, window speed, hours of snow, and hours of rain. We aim to forecast PM2.5 given past PM2.5, accurate meteorological data in the training period, and inaccurate meteorological data in forecasting period.

Figure 3 shows the forecast accuracy with respect to forecast horizon produced by our main model DSSMF and its two variants. One can see that DSSMF-NX without using exogenous variables performs worst, suggesting that exogenous variables are crucial for making accurate forecasts. Compared with DSSMF-NA which is not equipped with the ARD

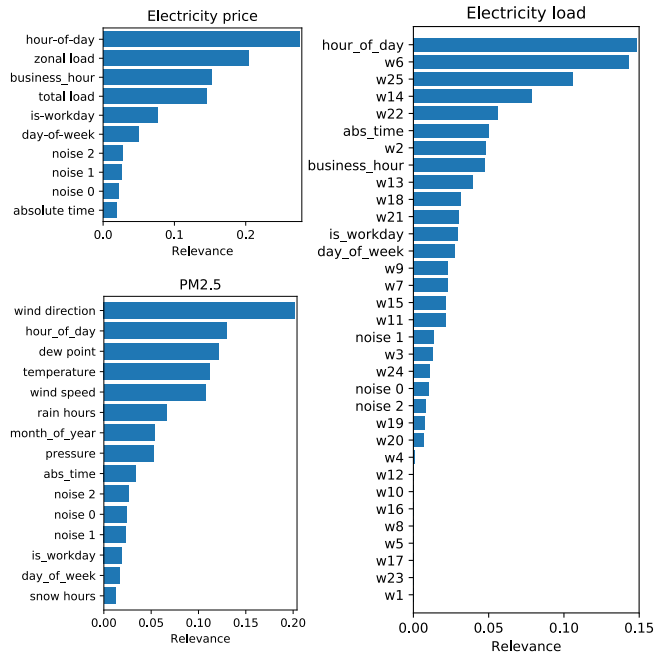


Figure 4: Learned relevance (i.e. w in Eq. 14) of exogenous variables by the proposed ARD network on three different tasks.

component, DSSMF performs better in most cases, showing that the ARD network is beneficial for making forecasts.

Figure 4 shows the learned relevance vector w for the three datasets. It can be found that ARD model has a low relevance weight estimation to the three intentionally added noise ($\{noise 1, noise 2, noise 3\}$) which are extra exogenous variables. This indicates the effectiveness of our ARD model on identifying irrelevant external factors.

For electricity price forecasting, among its top ranked variables, hour-of-day and business-hour indicate daily seasonality, besides zonal load and total load indicating the demand.

For electricity load forecasting, we use $\{w_1, \dots, w_{25}\}$ to denote 25 temperature records from 25 weather stations. Apart from that, it can be found that hour-of-day, absolute time, and business-hour are three most relevant exogenous variables, which also indicate the daily seasonality. For weather data, temperature from the specific station w_6 , w_{25} , w_{14} and w_{22} have the most impact.

For PM2.5 air quality forecasting, we find hour-of-day and month-of-year are two important exogenous variables, which indicates daily and yearly seasonality of PM2.5. Interestingly, wind direction and wind speed are also relevant variables, which indicates that polluted air in Beijing might be blown from other area. Also, this also suggests that wind direction forecast is crucial for making accurate PM2.5 forecasting which in fact has been widely recognized by the public. Besides, meteorological data such as dew point, temperature, and rain are also relevant to PM2.5 forecasting.

Figure 5 shows examples for each of the above tasks, we can see that DSSMF makes accurate and sharp probabilistic forecast. For PM2.5 (the third row), the forecast uncertainty (realistically) grows over time (without any extra manipula-

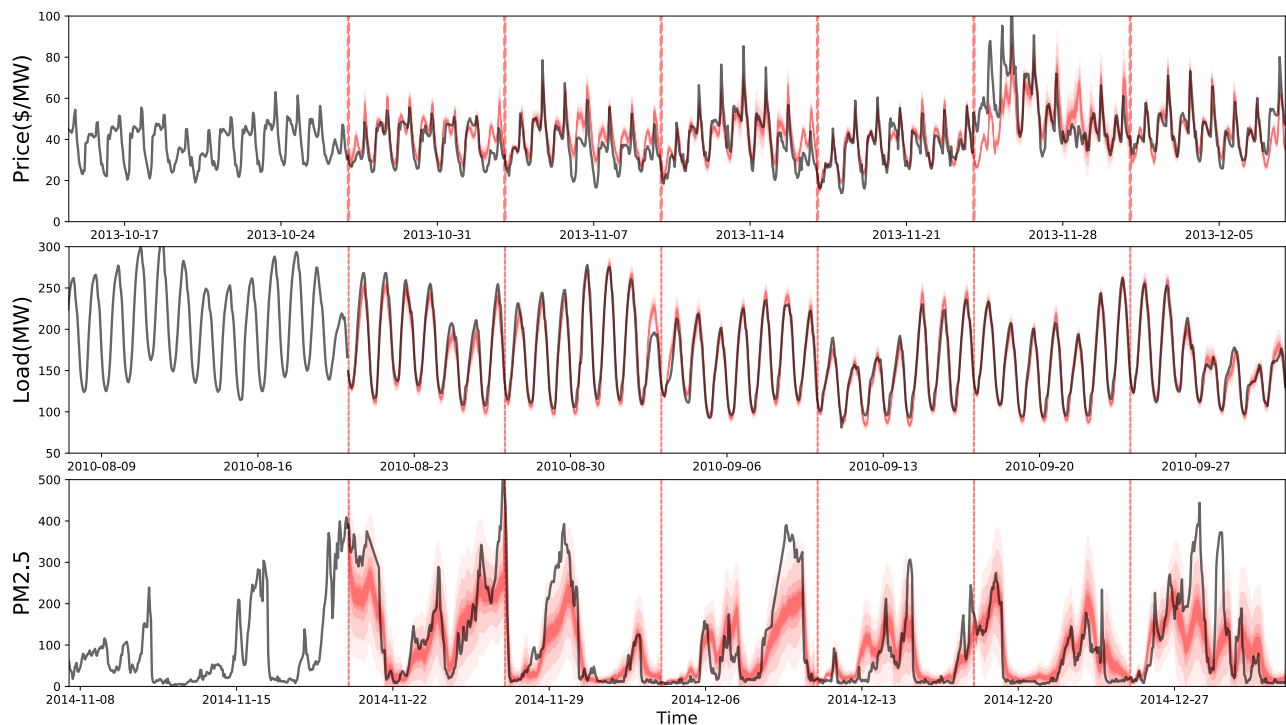


Figure 5: Probabilistic rolling-week multi-step forecasts by our model. Red vertical dashed lines indicate the start time of forecast. Probability densities are plot by confidence intervals [20%, 30%,50%, 80%, 95%] in transparency. Past series is not shown in full length. Best viewed in color and zoom in for better view.

tion on our model). The model spontaneously places higher uncertainty when forecast may not be accurate, which are desired features for downstream decision making process.

4 Conclusion

We have presented a probabilistic time series forecasting model, where the future uncertainty is modeled with a generative model. Our approach involves a deep network based embodiment of the state space model, to allow for non-linear emission and transition models design, which is flexible to deal with arbitrary data distribution. Extra exogenous variables in addition to time series data are exploited by our devised automatic relevance determination network, and their uncertainty is considered and modeled by our sampling approach. These techniques help improve the model robustness and interpretability by effectively identifying the key factors, which are verified by extensive experiments.

References

- [Bengio *et al.*, 2013] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- [Bishop and Tipping, 2000] Christopher M Bishop and Michael E Tipping. Variational relevance vector machines. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 46–53. Morgan Kaufmann Publishers Inc., 2000.
- [Box *et al.*, 2015] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [Brockwell *et al.*, 2002] Peter J Brockwell, Richard A Davis, and Matthew V Calder. *Introduction to time series and forecasting*, volume 2. Springer, 2002.
- [Burda *et al.*, 2015] Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. In *arXiv:1509.00519*, 2015.
- [Cho *et al.*, 2014] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *arXiv:1406.1078*, 2014.
- [Chung *et al.*, 2015] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. In *NIPS*, 2015.
- [Durbin and Koopman, 2012] James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*, volume 38. Oxford University Press, 2012.
- [Eleftheriadis *et al.*, 2017] Stefanos Eleftheriadis, Tom Nicholson, Marc Deisenroth, and James Hensman. Identification of gaussian process state space models. In *Advances in neural information processing systems*, pages 5309–5319, 2017.
- [Flunkert *et al.*, 2017] Valentin Flunkert, David Salinas, and Jan Gasthaus. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *arXiv preprint arXiv:1704.04110*, 2017.
- [Fraccaro *et al.*, 2016] M. Fraccaro, S. Sønderby, U. Paquet, and O. Winther. Sequential neural models with stochastic layers. In *NIPS*, 2016.
- [Gneiting and Katzfuss, 2014] Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151, 2014.
- [Hong *et al.*, 2016] Tao Hong, Pierre Pinson, Shu Fan, Hamidreza Zareipour, Alberto Troccoli, and Rob J Hyndman. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond, 2016.
- [Hyndman *et al.*, 2007] Rob J Hyndman, Yeasmin Khandaqar, et al. *Automatic time series for forecasting: the forecast package for R*. Number 6/07. Monash University, Department of Econometrics and Business Statistics ..., 2007.
- [Jordan *et al.*, 1999] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.
- [Kingma and Ba, 2014] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *arXiv:1412.6980*, 2014.
- [Kingma and Welling, 2013] D. Kingma and M. Welling. Auto-encoding variational bayes. In *arXiv:1312.6114*, 2013.
- [Krishnan *et al.*, 2015] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. In *arXiv:1511.05121*, 2015.
- [Krishnan *et al.*, 2017] Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [Längkvist *et al.*, 2014] Martin Längkvist, Lars Karlsson, and Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.
- [Leskovec *et al.*, 2014] J. Leskovec, A. Rajaraman, and Jeffrey D. Ullman. *Mining of massive datasets*. Cambridge university press, 2014.
- [Liang *et al.*, 2015] Xuan Liang, Tao Zou, Bin Guo, Shuo Li, Haozhe Zhang, Shuyi Zhang, Hui Huang, and Song Xi Chen. Assessing beijing’s pm2. 5 pollution: severity, weather impact, apec and winter heating. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2182):20150257, 2015.
- [Mnih and Rezende, 2016] A. Mnih and D. Rezende. Variational inference for monte carlo objectives. In *arXiv:1602.06725*, 2016.
- [Neal, 2012] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.
- [Rangapuram *et al.*, 2018] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In *Advances in Neural Information Processing Systems*, pages 7796–7805, 2018.
- [Wen *et al.*, 2017] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.
- [Wipf and Nagarajan, 2008] David P Wipf and Srikantan S Nagarajan. A new view of automatic relevance determination. In *Advances in neural information processing systems*, pages 1625–1632, 2008.
- [Yu *et al.*, 2016] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in neural information processing systems*, pages 847–855, 2016.