Forecasting Reservoir Inflow via Recurrent Neural ODEs

Fan Zhou, Liang Li*
University of Electronic Science and Technology of China
fan.zhou@uestc.edu.cn, liliang2333@gmail.com

Abstract

Forecasting reservoir inflow is critical for making many policies, ranging from flood control and agriculture irrigation to water ecology management, hydropower generation, and landslide prevention. Prior studies mainly exploit autoregressive models – e.g., recurrent neural networks (RNN) and its many variants – to model the flow time series's temporal pattern. However, existing approaches rely on regular and accurate inflow observations, which either fail to predict multi-scale inflow (e.g., an hour, a day, or a month ahead prediction) or ignore the uncertainty of observations due to confounding factors such as snowmelt and precipitation. To address the limitations, we propose a novel inflow forecasting model by incorporating the uncertainty of the observations into the RNN model and the continuous-time dynamics of the latent states with neural ordinary differential equations (ODE). Our method, called FlowODE, explicitly encodes the stochasticity of hidden conditions in addition to the temporal dependencies among inflow observations. Moreover, FlowODE explores a continuum of layers instead of discrete RNNs to model the hidden states' dynamics, allowing us to infer the inflow at any time horizon flexibly. We conduct extensive experiments on the real-world datasets collected from two large-scale hydropower dams. The results show that our method consistently outperforms previous inflow forecasting models while providing adaptable predictions and a flexible balance between prediction accuracy and computational cost.

Introduction

Artificial reservoirs are built by constructing dams on rivers, regulating natural streams by storing excess water in the rainy season, and supplying the stored water for future use. Large-scale reservoirs play vital roles in optimizing water resources management, such as water supply, flood/drought regularization, hydropower maximization, aquatic ecosystem balance, sediment transportation, and potential geological hazards (e.g., landslide and fluvial deposit) (Cuena 1983; Yin et al. 2016; Chang and Tsai 2016; Moshe et al. 2020). Most of the dams operate according to the predefined rules depending on climate change and historical observations. Because of numerous influencing factors, both intrinsic (e.g., precipitation and snowmelt) and extrinsic (e.g., downstream water regulation and agriculture irrigation), optimal reservoir operation is difficult (Petrik and Zilberstein 2011). For example, excess water in summer should have been stored for future hydropower generation, but the storage capacity needs to be maintained at a low level to tolerate the possible flood peaks. However, water discharge may result in significant loss of electricity revenue, which may be fundamentally reduced if accurate and reliable inflow forecast can be made in advance (Ahmad and Hossain 2019).

Related work. Traditionally, reservoirs are operated based on the knowledge of experts who usually design mathematical/physical models to simulate the dynamics of inflow/outflow. For example, earlier works (Sigvaldson 1976; Cuena 1983; Georgakakos and Marks 1987) prescribe time-based rules based on which different operations of the reservoir system can be simulated and configured for making optimal policies. However, predefined rule-based models cannot handle sudden events (e.g., flood and dam break), which prevents their applicability in real-time reservoir operation. (Petrik and Zilberstein 2011) propose to optimize hydroelectric systems using the knowledge of inflow and water discharge and introduce linear dynamic programs to solve the partially observable Markov decision processes.

Typical machine learning methods have shown the ability to fit complex multivariate time series data (e.g., inflow and outflow), and extract empirical knowledge and time-varying demands for improving reservoir operation. For instance, AutoRegressive Integrated Moving Average (ARIMA) family models have been used to model hydrological time series (Wang et al. 2015). Bayesian networks, as well as K-means clustering, were applied for predicting annual and monthly inflow (Noorbeh, Roozbahani, and Moghaddam 2020). Other algorithms, such as support vector regression and neural networks, have also been employed for learning nonlinear and nonstationary characteristics of hydrological data in the literature (Aboutalebi, Bozorg Haddad, and Loáiciga 2015; Hadiyan, Moeini, and Ehsanzadeh 2020).

Recent advances in deep learning, especially the recurrent neural networks (RNN), spurs a few of these studies on applying RNNs for modeling the hydrological time series and predicting reservoir inflow (Yang et al. 2019; Babaei, Moeini, and Ehsanzadeh 2019) due to the ability of RNNs on learning intricate dynamic temporal dependencies. (Yang et al. 2019) develops a real-time reservoir operation system

*Corresponding author

and verifies the effectiveness of RNNs in annual flow prediction based on the records of reservoirs in the upper Chao Phraya River. Similarly, (Banihabib, Bandari, and Peralta 2019) studied daily streamflow predictions and presented a model combing RNN and dynamic memory with exogenous inputs to forecast regular reservoir inflow. A recent study (Apaydin et al. 2020) evaluates several deep autoregressive models, including vanilla RNN, long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997), and gated recurrent unit (GRU) (Chung et al. 2014), on inflow prediction using the daily observational flow in Ermenek dam reservoir located in Turkey and found that LSTM performs the best on prediction accuracy.

Motivations. Despite the promising results achieved in prior studies, they are still susceptible to particular challenges. First, the future reservoir inflow is inherently uncertain due to the inaccurate observations and unpredictable factors (e.g., snowmelt and groundwater). Although previous endeavors have combined multiple deterministic results to yield probabilistic prediction or infer the probability distribution of predicted errors (Chang and Tsai 2016), few works pay attention to the capability of modeling observation uncertainty and inferring the density of stochastic variables of the neural networks. Besides, inflow prediction refers to a range of continuous-time series (e.g., discharge water, hydropower generation, and climate change). However, neural networks model the continuous dynamic systems but take discrete-time observations as input. This may not meet the complex control systems' requirements such as reservoir operation, where high-frequency feedback is necessary to maintain system flexibility and stability. Lastly, multi-horizon predictions (e.g., hourly and weekly) that are imperative for robust systems have not been sufficiently studied yet. This attribute would enable the systems to not only respond quickly to emergencies (e.g., flood and landslide) but also persistently optimize long-term strategies (e.g., aquatic ecosystem protection and hydroelectricity revenue).

Present work. To remedy the above limitations of prior studies, we present an alternative view of reservoir operation by modeling the multivariate time series as a continuous dynamic system instead of discrete-time neural networks. Specifically, we present a stochastic RNN to capture the sequential dependencies among the temporal observations, while empowering deterministic and simple parametric models with uncertainty and multimodality. Inspired by recent advances in neural ordinary differential equations (ODE) (Chen et al. 2018; Rubanova, Chen, and Duvenaud 2019), we propose to deal with multivariate flow time series in a continuous dynamic manner, rather than the deep but fixed layers used in previous RNN-based flow prediction models (Yang et al. 2019; Banihabib, Bandari, and Peralta 2019) and stochastic RNNs (Fraccaro et al. 2016; Goyal et al. 2017). By extending the discrete state transitions to continuous transformations, our model can forecast multi-horizon inflow without retraining the model. Moreover, the ability to trace the footprint of solving ODEs enables us to balance computational cost and the prediction accuracy.

In this study, we develop a new model, called FlowODE, to optimize reservoir operation by improving the inflow prediction accuracy. Our main contributions are as follows:

- We present a deep Bayesian RNN-based inflow prediction model to explicitly account for the stochastic variables associated with observations in addition to sequential conditions.
- We initiate the attempt to exploit neural ODEs for modeling multivariate flow data and provide a new perspective of learning continuous time series and inferring different horizons of future inflow.
- We conduct extensive experiments on real-world datasets collected from two large-scale reservoirs. The empirical evaluations demonstrate that FlowODE significantly improves the reservoir inflow prediction while providing explainable results of the model behavior.

Methodology: FlowODE

Problem Definition

We consider a set of multivariate time series observations $\mathbf{X}$, each of which $\mathbf{x} \in \mathbf{X}$ consists of electricity power $\mathbf{v}$ and water flow $\mathbf{w}$. The electricity, defined as $\mathbf{v} = \{v^1, v^2, v^3\}$, contains of three types of power, i.e., total power generation, on-grid power and auxiliary power. The water flow, defined as $\mathbf{w} = \{w^1, w^2, w^3\}$, is composed of water inflow, outflow and generation flow. Specifically, we study the following problem in this work:

Problem 1 ***Multi-horizon Reservoir Inflow Forecasting (MRIF).*** *Given a series of N past water flow observations $\mathbf{W}_t = \{\mathbf{w}_{t-N+1}, \mathbf{w}_{t-N+2}, \cdots, \mathbf{w}_t\} \in \mathbb{R}^{N\times P}$ ($P = 3$ denotes the vector dimensions) and the historical electricity power $\mathbf{V}_t = \{\mathbf{v}_{t-N+1}, \mathbf{v}_{t-N+2}, \cdots, \mathbf{v}_t\} \in \mathbb{R}^{N\times P}$, and external factors $\mathbf{e}$, the goal of MRIF task is to learn a function $\mathcal{F}$ to predict the volume of water inflow $\hat{\mathbf{w}}_\tau$ at future time step $\tau = t + \triangle t$, where $\triangle t$ can be an hour ($\triangle t = 1$), a day ($\triangle t = 24$), or a week ($\triangle t = 7 \times 24$):*

$$\hat{\mathbf{w}}_\tau = \mathcal{F}\left(\mathbf{W}_t, \mathbf{V}_t | \mathbf{e}_\tau; \Theta\right), \tag{1}$$

where Θ denote parameters.

Architecture Overview

FlowODE, as an inference network (encoder-decoder architecture), not only captures the rich temporal correlations among multivariate observations but also has the ability to account for the uncertainty of the time series with the proposed stochastic recurrent neural networks (SRNN). Besides, ODE solvers play the role of extrapolation decoder in this inference network, outputting the future latent variable $\mathbf{z}$ at multi-horizon time steps, which will be used for predicting different horizons of future inflow. Moreover, a hierarchical RNN is designed for modeling high-level temporal dependency to capture latent input variables $\mathbf{Z}_{1:t}$ and hidden states $\mathbf{H}_{1:t}$. The multi-head attention mechanism enables FlowODE to focus on the critical part of inflow observations. Otherwise, we consider social and natural factors into the model and present a gated fusion layer for real impact information selection. Finally, we use multilayer perceptrons (MLPs) for multi-horizon water inflow forecasting.

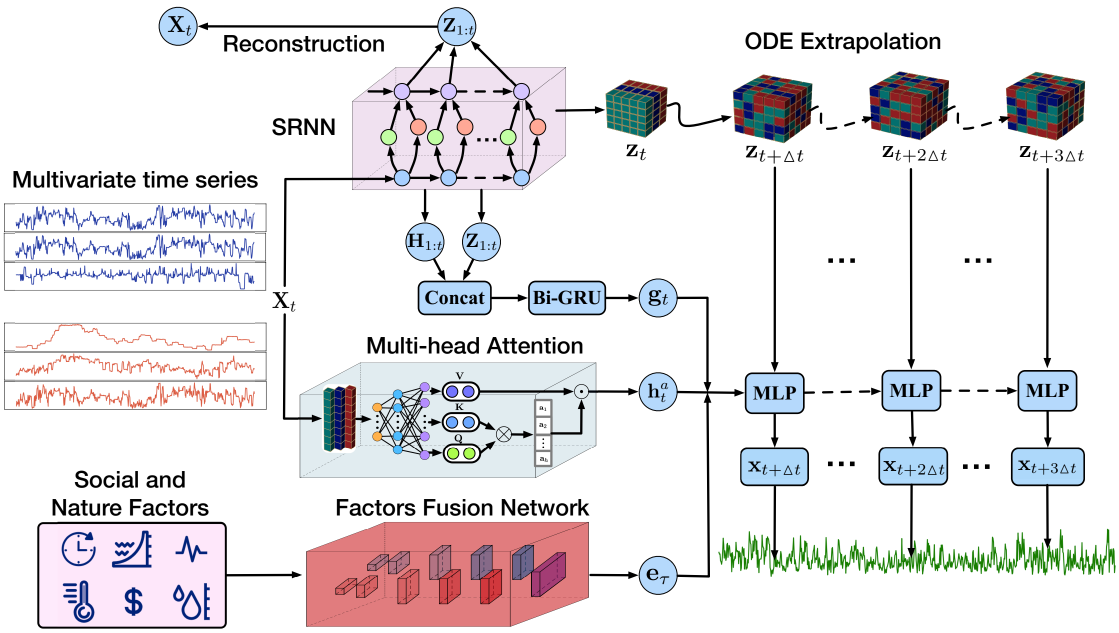


Figure 1: Overall architecture of the proposed FlowODE.

We summarize the overall architecture of FlowODE in Figure 1. In the rest of this section, we will introduce the details of each component.

Stochastic RNN with ODEs

A simple but effective approach is to utilize an RNN for modeling the temporal dependencies for traditional time series tasks. For example, LSTM and GRU are widely used as the basic module for modeling and predicting time series. However, when applying such RNNs for modeling large-scale multivariate time series, the prediction accuracy would be deteriorated by the uncertainty relationship (internal dependency) among various data. Besides, RNNs perform better on capturing fixed-interval dependence and may fail or achieve poor performance on multi-horizon predictions. To address this problem, we introduce a stochastic RNN (SRNN) model incorporating stochastic latent variables $\mathbf{z}$ into RNN to model the uncertainty of water flow. More specifically, with observations $\mathbf{X}$ as input, we first generate the mean μ and variance σ instead of modeling the hidden state $\mathbf{h}$ directly at every time step. Then, we use numerical ODE solver to transform μ and σ in a continuous way, which means we can obtain μ and σ of each time step more accurately and encourage GRU to produce latent variable at any desired time as long as we change the integration time in the ODE solver.

Neural ODE (NODE) (Chen et al. 2018) considered the infinite-steps hidden state update in neural networks to replace the discrete sequence of hidden states transformation. NODE solves the initial value problem with continuous transform and can compute the constant dynamics of hidden states $\mathbf{z}$ via ODEs:

$$\frac{d\mathbf{z}(t)}{dt} = f_\omega(\mathbf{z}(t), t), \quad \text{where} \quad \mathbf{z}(t) = \mathbf{z}_t,$$
$$\mathbf{z}(t_1) = \mathbf{z}(t_0) + \int_{t_0}^{t_1} f_\omega(\mathbf{z}(t), t)dt, \tag{2}$$

where f_ω is parameterized by ω specifying a neural network. By regarding the infinite hidden state update process of the neural ODE block as solving ODEs with numerical methods such as Euler, Runge–Kutta and adjoint method (Dormand and Prince 1980; Ascher, Ruuth, and Spiteri 1997), it allows us to obtain the hidden states $\mathbf{z}(t)$ at *any* desired moment (Chen et al. 2018). NODE proposed to use an adjoint method to simulate a dynamical system. However, the dynamics of either the hidden state or the adjoint might be unstable due to numerical instability of backward ODE solve.

In our FlowODE, we abandon the adjoint method that NODE employed to compute the gradients and alternatively use the primary Euler method as the numerical solver. We summarize the procedure as:

$$
\begin{aligned}
&\mathbf{I} = \left[\mu^{\mathbf{h}}_{t-1}, \sigma^{\mathbf{h}}_{t-1}, \mathbf{x}_t\right], \\
&\mathbf{O} = \left[\mu^{\mathbf{h}}_{t-1}\mathbf{R}\left(\mathbf{I}\right), \sigma^{\mathbf{h}}_{t-1}\mathbf{R}\left(\mathbf{I}\right), \mathbf{x}_t\right], \\
&\hat{\mu}, \hat{\sigma} = \tanh\left(\mathbf{W_o}\mathbf{O} + \mathbf{b_o}\right), \\
&\left[\mu, \sigma\right] = \mathrm{ODESolve}\left(f_\omega, \left[\hat{\mu}, \hat{\sigma}\right], t-1, t\right), \\
&\mu^{\mathbf{h}}_t = \left(1 - \mathbf{U}\left(\mathbf{I}\right)\right)\mu + \mathbf{U}\left(\mathbf{I}\right)\mu^{\mathbf{h}}_{t-1}, \\
&\sigma^{\mathbf{h}}_t = \left|\left(1 - \mathbf{U}\left(\mathbf{I}\right)\right)|\sigma| + \mathbf{U}\left(\mathbf{I}\right)\sigma^{\mathbf{h}}_{t-1}\right|,
\end{aligned} \tag{3}
$$

where f_ω denotes the differentiable network parameterized by ω; $\mathbf{I}$ represents the input of reset gate $\mathbf{R}$ and update gate $\mathbf{U}$; $\mathbf{O}$ denotes the input of new mean and sigma update network in the GRU cell; $\mathbf{W}_o$ indicates the learnable parameters in GRU unit. Now, we can directly sample $\mathbf{z}_t$ from the learned distribution $q_\phi\left(\mathbf{z}_t|\mathbf{X}_t, \mathbf{z}_{t-1}\right)$ using the reparameterization trick (Kingma and Welling 2014):

$$
\mathbf{z}_t = \mathbf{W}_\mu\left[\mu^{\mathbf{h}}_t, \mu^{\mathbf{h}}_{t-1}\right] + \mathbf{W}_\sigma\left[\sigma^{\mathbf{h}}_t, \sigma^{\mathbf{h}}_{t-1}\right] * \epsilon, \tag{4}
$$

where $\mathbf{W}_\mu$ and $\mathbf{W}_\sigma$ are the corresponding parameters. ϵ are samples from a standard Gaussian $\epsilon \sim \mathcal{N}(0, I)$. Therefore, SRNN outputs a sequence of latent variables $\mathbf{Z}_{1:t}$ taking into account the uncertain observations. We can now train the water flow inference network by maximizing the evidence lower bound (ELBO) as follows:

$$
\begin{aligned}
\mathrm{ELBO}\left(\theta, \phi\right) &= \mathbb{E}_{q_\phi}\log\left[\frac{p_\theta\left(\mathbf{X}_t, \mathbf{z}_t\right)}{q_\phi\left(\mathbf{z}_t|\mathbf{X}_t\right)}\right] \\
&= \mathbb{E}_{q_\phi}\log\left[p_\theta\left(\mathbf{X}_t|\mathbf{z}_t\right)\right] + \mathbb{E}_{q_\phi}\log\left[p_\theta\left(\mathbf{z}_t\right)\right] \\
&\quad - \mathbb{E}_{q_\phi}\left[q_\phi\left(\mathbf{z}_t|\mathbf{X}_t\right)\right],
\end{aligned} \tag{5}
$$

where the first item is the reconstruction likelihood function based on the variational posterior distribution, and the last two items are the KL Divergence of the prior distribution and the variational posterior distribution. Parameters θ and ϕ are the inference network $q(\cdot)$ and generate network $p(\cdot)$, respectively.

Multi-horizon Dependency Learning

RNN-based models usually perform well on fix-interval observations but may fail to model irregular samplings. In our case, we would like to model flexible sampling intervals and learn the multi-horizon dependencies. Here we exploit self-attention to learn the conditioned non-uniform dependencies among the flow data. Standard self-attention mechanism (Vaswani et al. 2017), such as additive or dot-product attention, have been widely employed for global long-term temporal dependency extraction. However, multi-horizon water flow prediction relies on a different part of historical observations, which means we need to enable our model to focus on different historical information periods. In FlowODE, we use multi-head attention, as scaled dot-product attention, to enhance the model's ability to judge the importance of information in different periods.

Given observations $\mathbf{X}_t = \{\mathbf{x}_{t-N+1}, \mathbf{x}_{t-N+2}, \cdots, \mathbf{x}_t\} \in \mathbb{R}^{B\times N\times 2P}$, we first utilize the linear function to transform the matrix $\mathbf{X}_t$ into query matrix $\mathbf{Q}$, key matrix $\mathbf{K}$ and value matrix $\mathbf{V}$ ($\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V} \in \mathbb{R}^{B\times N\times M}$, where B represents batch size and M denotes the dimensions of matrix), and split each matrix to h parallel heads:

$$
\begin{aligned}
\mathbf{Q}\mathbf{W}^{\mathbf{Q}} &= \left[\mathbf{Q}_1\mathbf{W}^{\mathbf{Q}}_1, \mathbf{Q}_2\mathbf{W}^{\mathbf{Q}}_2, \ldots, \mathbf{Q}_h\mathbf{W}^{\mathbf{Q}}_h\right], \\
\mathbf{K}\mathbf{W}^{\mathbf{K}} &= \left[\mathbf{K}_1\mathbf{W}^{\mathbf{K}}_1, \mathbf{K}_2\mathbf{W}^{\mathbf{K}}_2, \ldots, \mathbf{K}_h\mathbf{W}^{\mathbf{K}}_h\right], \\
\mathbf{V}\mathbf{W}^{\mathbf{V}} &= \left[\mathbf{V}_1\mathbf{W}^{\mathbf{V}}_1, \mathbf{V}_2\mathbf{W}^{\mathbf{V}}_2, \ldots, \mathbf{V}_h\mathbf{W}^{\mathbf{V}}_h\right],
\end{aligned} \tag{6}
$$

where $\mathbf{W}^{*}_i \in \mathbb{R}^{M/h\times M/h}$ are the i-th head parameter of queries, keys and values, respectively. $\mathbf{Q}_i \in \mathbb{R}^{Bh\times N\times M/h}$ represents the i-th head of the query matrix. Subsequently, we calculate the affinity matrix corresponding to each head to obtain the weight scores:

$$
\mathbf{a}_i = \mathrm{softmax}\left(\frac{\left(\mathbf{Q}_i\mathbf{W}^{\mathbf{Q}}_i\right)^\top\left(\mathbf{K}_i\mathbf{W}^{\mathbf{K}}_i\right)}{\sqrt{M/h}}\right), \text{where} \quad i \in h,
$$

where $\mathbf{a}_i$ is the weight score for i-th head. Then, each feature representation $\mathbf{h}^a_i$ can be calculated by element-wise product operator with i-th value matrix $\mathbf{V}_i$ and its corresponding parameter $\mathbf{W}^{\mathbf{V}}_i$:

$$
\mathbf{h}^a_i = \mathbf{a}_i \odot \left(\mathbf{V}_i\mathbf{W}^{\mathbf{V}}_i\right), \tag{7}
$$

where $\odot$ denotes the Hadamard product. Finally, the attention feature representation $\mathbf{h}^a_t$ at time t can be obtained by concatenating h parallel heads representations:

$$
\mathbf{h}^a_t = \mathrm{concat}\left[\mathbf{h}^a_1, \mathbf{h}^a_2, \cdots, \mathbf{h}^a_h\right]. \tag{8}
$$

Extrapolation & Reconstruction Decoders

There are two decoders in FlowODE, namely the reconstruction decoder and the extrapolation decoder. The former is used to reconstruct the input to obtain the stochastic variables $\mathbf{Z}_t$ in a self-supervised learning manner. The later one takes a series of output of SRNN as the input, i.e., $\mathbf{Z}_t = \{\mathbf{z}_{t-N+1}, \mathbf{z}_{t-N+2}, \cdots, \mathbf{z}_t\}$, and uses a vanilla GRU to reconstruct the original input data $\mathbf{X}_t$.

As for the extrapolation decoder, we use ODE Solver as the continuous autoregressive layers. Taking the stochastic latent variable $\mathbf{z}_t$ at time t as the initial value, we can arbitrarily calculate the latent variables $\mathbf{z}_\tau$ corresponding to different time steps in the future (e.g., an hour, a day, or a week)

$$
\mathbf{z}_\tau = \mathrm{ODESolve}\left(f_\nu, \mathbf{z}_t, \tau\right), \tag{9}
$$

where we use Dopris method (Dormand and Prince 1980) as the numerical solution. The main objective of the extrapolation decoder is to adapt our model to multi-horizon scenarios, which, combined with the multi-horizon dependency learned by the self-attention, can be readily used to predict the future inflow at any scale.

External Factors Fusion

As we mentioned, external factors (e.g., precipitation and flood discharge in the upstream reservoir) are also crucial to predicting water inflow. Furthermore, reservoir inflow is highly seasonal and varies significantly with weather conditions. For example, increased temperature will cause glaciers to melt. Besides, human activities (e.g., irrigation and navigation) will also affect the reservoir inflow. Therefore, in this work, we collectively refer to these effects as external factors and design a factor extraction network for learning the impact of these factors. Specifically, we embed continuous temporal features into low-level dimensions and directly fed the categorical features into the network, appending an ODESolve layer to capture such influence. After that, all external factor information is compressed into a vector $\mathbf{v}_e$, which contains both negative (e.g., noise, outlier, abnormal points, and inaccurate measurement points) and positive information. To screen out the negative factors while maintaining the beneficial knowledge, we introduced a gated fusion layer to generate the influence factor $\mathbf{e}$:

$$\mathbf{e} = \operatorname{sigmoid}\left(\mathbf{W}_e \mathbf{v}_e + \mathbf{b}_e\right) \circ \operatorname{ODESolve}\left(\mathbf{v}_e\right), \quad (10)$$

where sigmoid is used as the activation function, normalizing the factors impact into the range [0,1].

Inflow Prediction

For the inflow prediction task, modeling multi-level dependencies is an indispensable technique for improving the model's performance. Here we utilize a hierarchical RNN to obtain multi-level temporal dependencies $\mathbf{g}_t$ by passing hidden states and latent variables into a Bidirectional-GRU:

$$\mathbf{g}_t = \left[\overrightarrow{\mathrm{GRU}}\left(\left[\mathbf{Z}_t, \mathbf{H}_t\right]\right), \overleftarrow{\mathrm{GRU}}\left(\left[\mathbf{Z}_t, \mathbf{H}_t\right]\right)\right]. \quad (11)$$

With the learned latent variables $\mathbf{z}_\tau$ via extrapolation decoder, attention feature representations $\mathbf{h}_t^a$ by multi-head attention, as well as the external factor $\mathbf{e}$ learned by factors fusion layer, we are ready to forecast the multi-horizon water inflow volume. We concatenate all of these learned features and employ basic MLPs as the predictor, which generates the future water flow volume $\hat{\mathbf{w}}_\tau$ at time step τ:

$$\hat{\mathbf{w}}_\tau = \operatorname{MLP}\left(\left[\mathbf{z}_\tau, \mathbf{h}_t^a, \mathbf{g}_t, \mathbf{e}_\tau\right]\right), \quad (12)$$

Objective: Forecasting multi-horizon water flow volume can be regarded as a linear regression task. Therefore, by simultaneously minimizing the mean squared error between the real water volume ($\mathbf{w}_\tau$) and the predicted value ($\hat{\mathbf{w}}_\tau$), and maximizing the ELBO via Eq.5, the loss function is defined as:

$$\mathcal{L}(\Theta) = \left\|\mathbf{X}_t - \hat{\mathbf{X}}_t\right\|_2^2 + \left\|\mathbf{w}_t - \hat{\mathbf{w}}_t\right\|_2^2 - \operatorname{ELBO}(\theta, \phi), \quad (13)$$

where Θ denotes all parameters in FlowODE.

Experiments

Datasets: We conduct experiments on two real-world datasets collected from two large-scale hydropower dams.

- **PBG:** is an artificial dam built in 2006 that houses a hydroelectric power station with 6×600 MW generators for a total installed capacity of 3,600 MW, and is the largest hydropower station on the Dadu River.
- **SXG:** is a smaller hydropower station located in the downstream of PBG, which is installed with 4×165 MW generators. As a regulation station of PBG, its inflow is significantly affected by the discharge of PBG.

Dataset	**PBG**	**SXG**
Time interval	1 hour	1 hour
power generation (Mwh)	[0.0, 3587.8]	[0.0, 660.0]
Water flow		
Water inflow	[0.0, 7020.0]	[0.0, 5571.0]
Avg. inflow	1549.0	1496.9
Water outflow	[119.0, 5670.0]	[53.8, 6090.0]
Generation flow	[119.0, 2470.0]	[55.1, 2480.0]
External Factors (meteorology, time and sale price)		
temperature/°C	[-24.4 21.7]	[-24.4 21.7]
HourOfDay	[0.0, 24.0]	[0.0, 24.0]
DayOfWeek	[1, 7]	[1, 7]
Max Precipitation	72.7 mm	74.1 mm
Control water level	841 m	665 m
Electricity price	mask	mask

Table 1: Statistics of datasets. Some social/economic information such as sale prices is masked.

Table 1 shows the statistics of PBG and SXG data. Each dataset includes two types of multivariate time series data – hydropower generation and water flow information, and each dataset is split into two different periods (P1 and P2), both spanning one year. Otherwise, the external factors such as rainfall and temporal are also taken into account for enhancing the robustness of the FlowODE.

Preprocessing: Due to the volume of water inflow can not be measured directly, we obtain the inflow using reservoir storage minus the water outflow at a certain time. Meanwhile, the Min-Max normalization is also used to regularize all the value for speeding up the training. We segment the time series into fixed-length sequences with length N (N = $1 \times 24 \times 7$), and all sampling interval is 1 hour. Therefore, each batch of training data has the shape $\mathbf{X}_t \in \mathbb{R}^{B \times N \times 2P}$, where B denotes the batch size.

Baselines: We compare our method with the following approaches that are widely used for time series forecasting: (1) **Historical Average (HA)**: uses the average value of the previous T historical water inflow volume as the forecasting result at $T + \tau$ time. (2) **ARIMA**: combines autoregressive (AR) and moving average (MA) for water inflow prediction, and has been used for inflow prediction (Wang et al. 2015). (3) **SARIMA**: expends additional seasonal terms in the ARIMA models and therefore has more robust than ARIMA and has been widely used for time series forecasting. (4) **SVR**: is a support vector regression method that can be used to predict the reservoir inflow (Aboutalebi, Bozorg Haddad, and Loáiciga 2015). (5) **BN**: a Bayesian network-based method that predicts the reservoir inflow Noorbeh2020 while preserving the uncertain nature of inflow. (6) **LSTM** (Hochreiter and Schmidhuber 1997): is a well-known RNN model that captures the long-short temporal dependencies with gating mechanism, and has been widely used for inflow prediction (Yang et al. 2019; Babaei, Moeini, and Ehsanzadeh 2019). (7) **Bi-GRU** (Chung et al. 2014): takes GRUcell as the basic unit and considers the impact of post-set information on the current hidden state by concatenating the forward and backward hidden states. (8) **GRU-VAE** (Su et al. 2019): reconstruct the input time se-

Datasets	PBG						SXG					
Time span	P1			P2			P1			P2		
Metrics	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
HA	640.9	472.8	0.690	685.0	514.8	0.734	332.7	253.2	1.335	582.6	374.8	0.801
ARIMA	425.2	314.5	0.681	486.7	378.1	0.705	257.4	206.8	0.733	328.2	223.3	0.555
SARIMA	407.6	298.5	0.675	465.7	361.2	0.699	244.4	197.3	0.712	307.6	200.1	0.512
SVR	355.6	277.7	0.659	453.1	355.9	0.684	232.8	186.4	0.636	231.5	160.4	0.344
BN	345.1	271.2	0.655	450.6	352.3	0.676	229.5	177.6	0.614	209.2	144.7	0.321
LSTM	311.8	256.8	0.651	422.9	330.6	0.659	203.8	155.9	0.518	174.6	136.4	0.288
Bi-GRU	308.7	248.2	0.646	416.1	328.9	0.665	200.6	152.8	0.510	177.6	126.8	0.278
GRU-VAE	307.3	247.2	0.632	415.1	326.2	0.663	197.8	150.8	0.501	174.6	124.8	0.261
LatentODE	306.5	246.7	0.626	412.4	324.5	0.659	195.1	150.2	0.498	172.7	124.9	0.258
FlowODE	**296.3**	**236.6**	**0.596**	**401.5**	**312.3**	**0.621**	**184.7**	**141.4**	**0.466**	**166.2**	**114.8**	**0.231**

Table 2: Performance comparisons of algorithms on two datasets over two different periods. A paired t-test was performed for statistical significance of the results ($p < 0.005$).

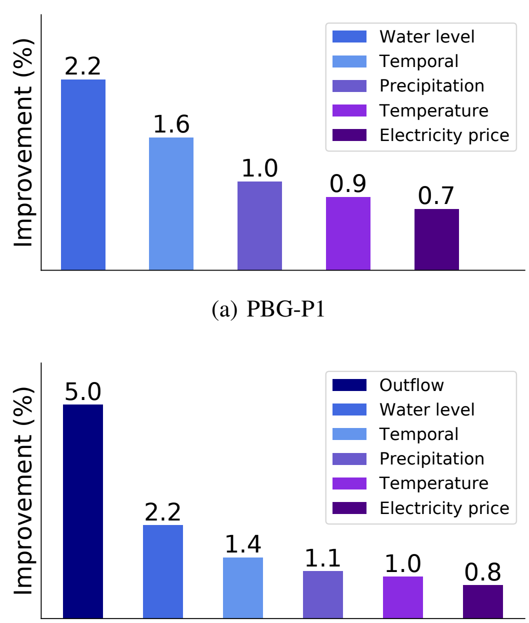
ries from the latent variable of observations by VAE, and employs GRU as the basic module for both the encoder and decoder. (9) **LatentODE** (Rubanova, Chen, and Duvenaud 2019): generalizes RNNs to have continuous-time hidden dynamics defined by ODEs. It explicitly models the irregular sampled time series and can, in theory, handle arbitrary time gaps between observations.

Experimental Setting: We implemented FlowODE based on PyTorch1.1.0 and Python3.6 on Ubuntu 16.04 operating system with a single NVIDIA GeForce GTX 2080ti GPU. The batch size B is default set as 128. The learning rate is set to 0.0001 at the beginning and decays half every 30 epochs. ADAM optimization algorithm (Kingma and Ba 2014) is used to train all baselines model with the setting of β_1 = 0.9, β_2 = 0.999. In Dopris ODE solver, we set the relative tolerance *rtol* as 1e-5 and the absolute tolerance *atol* as 1e-5. We run 200 epochs in total during the training stage, and then verify the model on testing data.

Metrics: We evaluate all methods with three widely used metrics for time series prediction: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

Inflow Prediction Performance: Table 2 shows the performance of different approaches to the inflow prediction of two reservoirs. We can observe that the proposed FlowODE model achieves the best results in terms of three metrics. Classical time series models such as HA, ARIMA, and SVR perform poorly due to their inability to model non-linear dependencies in inflow time series. SVR and BN, which have been widely used in inflow prediction, perform worse than deep learning models such as LSTM and Bi-GRU. This result demonstrates the superiority of RNNs on learning long-short term dependencies in time series. However, the vanilla RNN models can be improved by capturing the stochasticity of time series data, as done by GRU-VAE, LatentODE, and our FlowODE. Besides, LatentODE and FlowODE slightly outperform GRU-VAE, demonstrating the benefit of extending classical RNNs with continuous dynamics. This also implies that ODEs allow us to define a generative process over the inflow time series based on the deterministic evolution of an initial latent state and improve different time interval predictions. Finally, FlowODE continuously outperforms LatentODE because the latter is purposely designed for sporadically-observed time series such as patient measurements (De Brouwer et al. 2019). However, water inflow and other time series (e.g., discharge and electricity generation) are measured systematically at fixed time intervals in our hydro-stations.

Influence of External Factors:

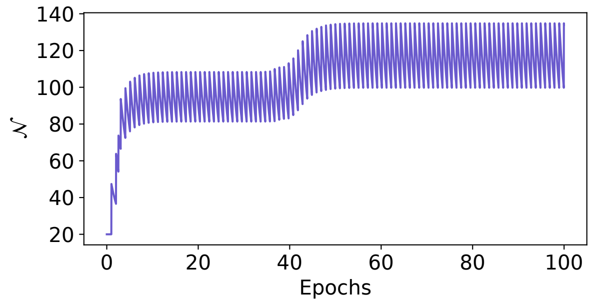


(a) PBG-P1

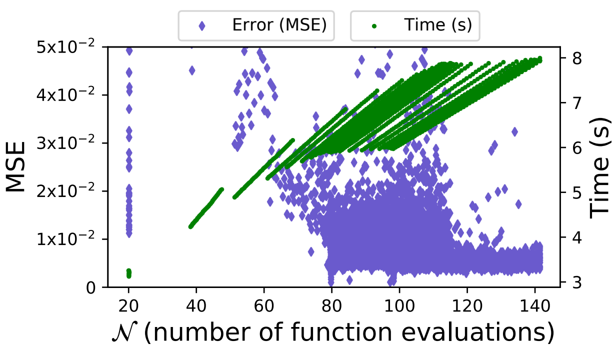
(b) SXG-P1

Figure 2: Influence of different external factors.

In Figure 2, we individually study the influence of each factor on the future inflow forecasting. Among the factors, we find that water level is an essential signal of inflow prediction because it contains implicit but critical information that cannot be explicitly measured, such as snowmelt and underground water. Climatic factors such as temperature and rainfall contribute inflow prediction, but their impact is limited because such information often lags behind the reservoir inflow. We also find that the most critical factor is the outflow of the upstream reservoir, e.g., the discharge of PBG dam contributes significantly for inflow prediction of SXG that is located downstream of PBG. This result suggests that the operation of cascaded reservoirs requires overall coordination and optimization.



(a) The number of function evaluations on PBG-P1.

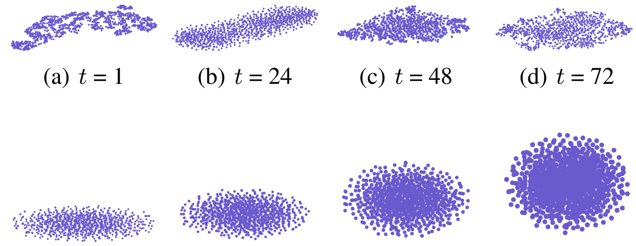


(b) $\mathcal{N}$ vs. MSE (PBG-P1).

Figure 3: Computation overhead vs. Prediction accuracy.

Computation Cost vs. Prediction Accuracy: As mentioned earlier, the ODE solution allows our method to dynamically balance the trade-off between prediction accuracy and the computation cost. As shown in Figure 3(a), the required function evaluations $\mathcal{N}$ increase very intensely at the beginning, but quickly become stable. By studying the results depicted in Figure 3(b), we can see that the more evaluated points (functions), the lower the prediction error our model can achieve. This property is typically useful for inflow prediction in extreme events (e.g., flood). That is, we can trade some accuracy for faster response.

Evolution of Latent Factors: As we know, vanilla RNNs are limited to extracting temporal dependencies. While in multivariate time series, e.g., the water flow and electricity generation, the correlation between multivariate is complex. In SRNN, we learn the uncertainty of time series through modeling the stochastic latent variables, which carry more expressive information and inherent dependencies than deterministic RNNs, as have been observed in previous works (Fraccaro et al. 2016; Goyal et al. 2017). Figure 4 visualizes the evolution of latent representations increasing with the hidden states. As more samples fed into the SRNN, the distribution of latent variables $\mathbf{z}$ is gradually changed from stochastic to a certain Gaussian. Note that we use a Gaussian to regularize the latent space, which is simple but effective in our case. However, more complex and flexible latent space transformations such as normalizing flows (Rezende and Mohamed 2015) and the more effective variants (Papamakarios et al. 2019) can be easily used to replace the diagonal Gaussian.



(a) $t = 1$ (b) $t = 24$ (c) $t = 48$ (d) $t = 72$ (e) $t = 96$ (f) $t = 120$ (g) $t = 148$ (h) $t = 168$

Figure 4: Latent representation learning in SRNN.

Discussion

In this work, we present a multi-horizon prediction inference network FlowODE for reservoir inflow forecasting. FlowODE consists of a newly designed stochastic RNN, which overcomes the issue of modeling uncertain latent variables in classical RNNs. To forecast future inflow volume at any desired time, we introduce neural ODE solvers to evolve the latent representation, which bridges the gap between the nature of continuous flow time series and the underlying discrete neural networks. Experimental results demonstrate the effectiveness and superiority of the proposed framework in solving the multi-horizon water inflow prediction.

There are several limitations of the proposed model that requires further investigation in the future. Although the dams' cascade effect has been observed in our experiments, modeling more dams is non-trivial and its impact is still unclear. Moreover, dynamically adjusting the reservoirs' water level to maximize hydroelectricity generation's profit and optimize our flood control policies and inundation prediction are important ongoing work. Besides, modeling more complex and flexible latent factors using more sophisticated statistical approaches such as normalizing flows may improve the inflow prediction performance of FlowODE. Furthermore, we used Euler method and Dopris method as ODE numerical solvers in SRNN and latent representation, respectively, which can be further improved with more fast and accurate neural ODE training such as Legendre polynomials and higher-order state approximation (Quaglino et al. 2020).

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant No.62072077 and No.61602097).

References

Aboutalebi, M.; Bozorg Haddad, O.; and Loáiciga, H. A. 2015. Optimal monthly reservoir operation rules for hydropower generation derived with SVR-NSGAII. *Journal of Water Res. Plan. and Man.* 141(11): 04015029.

Ahmad, S. K.; and Hossain, F. 2019. A generic data-driven technique for forecasting of reservoir inflow: Application for hydropower maximization. *Environmental Modelling & Software* 119: 147–165.

Apaydin, H.; Feizi, H.; Sattari, M. T.; Colak, M. S.; Shamshirband, S.; and Chau, K.-w. 2020. Comparative Analysis of Recurrent Neural Network Architectures for Reservoir Inflow Forecasting. *Water* 12(5): 1500.

Ascher, U. M.; Ruuth, S. J.; and Spiteri, R. J. 1997. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics* 25(2-3): 151–167.

Babaei, M.; Moeini, R.; and Ehsanzadeh, E. 2019. Artificial Neural Network and Support Vector Machine Models for Inflow Prediction of Dam Reservoir (Case Study: Zayandehroud Dam Reservoir). *Water Res. Man.* 33(6): 2203–2218.

Banihabib, M. E.; Bandari, R.; and Peralta, R. C. 2019. Auto-Regressive Neural-Network Models for Long Lead-Time Forecasting of Daily Flow. *Water Res. Man.* 33(1): 159–172.

Chang, F.-J.; and Tsai, M.-J. 2016. A nonlinear spatio-temporal lumping of radar rainfall for modeling multi-step-ahead inflow forecasts by data-driven techniques. *Journal of Hydrology* 535: 256–269.

Chen, T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. In *NeuIPS*.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555* .

Cuena, J. 1983. The Use of Simulation Models and Human Advice to Build an Expert System for the Defense and Control of River Floods. In *IJCAI*, 246–249.

De Brouwer, E.; Simm, J.; Arany, A.; and Moreau, Y. 2019. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. In *NeurIPS*, 7379–7390.

Dormand, J. R.; and Prince, P. J. 1980. A family of embedded Runge-Kutta formulae. *Journal of computational and applied mathematics* 6(1): 19–26.

Fraccaro, M.; Sønderby, S. K.; Paquet, U.; and Winther, O. 2016. Sequential neural models with stochastic layers. In *NIPS*, 2199–2207.

Georgakakos, A. P.; and Marks, D. H. 1987. A new method for the real-time operation of reservoir systems. *Water Res. Res.* 23(7): 1376–1390.

Goyal, A. G. A. P.; Sordoni, A.; Côté, M.-A.; Ke, N. R.; and Bengio, Y. 2017. Z-forcing: Training stochastic recurrent networks. In *NIPS*, 6713–6723.

Hadiyan, P. P.; Moeini, R.; and Ehsanzadeh, E. 2020. Application of static and dynamic artificial neural networks for forecasting inflow discharges. *Sustainable Computing: Informatics and Systems* 27.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv: 1412.6980* .

Kingma, D. P.; and Welling, M. 2014. Auto-encoding variational bayes. In *ICLR*.

Moshe, Z.; Metzger, A.; Elidan, G.; Kratzert, F.; Nevo, S.; and El-Yaniv, R. 2020. HydroNets: Leveraging River Structure for Hydrologic Modeling. In *AI for Physical Sciences Workshop ICLR*.

Noorbeh, P.; Roozbahani, A.; and Moghaddam, H. K. 2020. Annual and Monthly Dam Inflow Prediction Using Bayesian Networks. *Water Res. Man.* 34(9): 2933–2951.

Papamakarios, G.; Nalisnick, E.; Rezende, D. J.; Mohamed, S.; and Lakshminarayanan, B. 2019. Normalizing Flows for Probabilistic Modeling and Inference. *arXiv preprint arXiv:1912.02762* .

Petrik, M.; and Zilberstein, S. 2011. Linear Dynamic Programs for Resource Management. In *AAAI*.

Quaglino, A.; Gallieri, M.; Masci, J.; and Koutník, J. 2020. SNODE: Spectral Discretization of Neural ODEs for System Identification. In *ICLR*.

Rezende, D.; and Mohamed, S. 2015. Variational Inference with Normalizing Flows. In *ICML*, 1530–1538.

Rubanova, Y.; Chen, T. Q.; and Duvenaud, D. K. 2019. Latent Ordinary Differential Equations for Irregularly-Sampled Time Series. In *NeurIPS*, 5321–5331.

Sigvaldson, O. 1976. A simulation model for operating a multipurpose multireservoir system. *Water Res. Res.* 12(2): 263–278.

Su, Y.; Zhao, Y.; Niu, C.; Liu, R.; Sun, W.; and Pei, D. 2019. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In *KDD*, 2828–2837.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*, 5998–6008.

Wang, W.-c.; Chau, K.-w.; Xu, D.-m.; and Chen, X.-Y. 2015. Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition. *Water Res. Man.* 29(8): 2655–2675.

Yang, S.; Yang, D.; Chen, J.; and Zhao, B. 2019. Real-time reservoir operation using recurrent neural networks and inflow forecast from a distributed hydrological model. *Journal of Hydrology* 579: 124229.

Yin, Y.; Huang, B.; Wang, W.; Wei, Y.; Ma, X.; Ma, F.; and Zhao, C. 2016. Reservoir-induced landslides and risk control in Three Gorges Project on Yangtze River, China. *Journal of Rock Mechanics and Geotechnical Engineering* 8(5): 577–595.