Real-time Water Consumption Prediction

Value and Probabilistic Prediction

Cao Jinpu
Civil and Environmental Engineering Department
jinpu@stanford.edu

Abstract—Urban water supply service is one of the key functions of urban infrastructure. Accurate prediction of water consumption in the future is helpful to detect the abnormalities of water supply systems including pipe bursts in real-time, and effectively improve the economy and stability of the water supply system. Based on two water consumption dataset, the paper finds that Recurrent Neural Networks model represented by GRU model outperforms other artificial neural networks models when conducting the value prediction of water consumption. The root mean square error of GRU model is only 80% of the basic model's error. Besides, the paper develops a probabilistic prediction model of water consumption based on Deep Autoregression (DeepAR) model whose mean absolute percentage error on test set is only 6%. It can be used in the pipe bursts alarming.

Keywords—water consumption, seq to seq, RNN, probabilistic prediction, DeepAR

I. Introduction

The urban water supply system is an important part of urban infrastructure. Water consumption has a certain periodic law. For example, more water is used in summer than in winter, and more water is used during the day than at night; however, it will also be affected by some other factors, such as holidays or the immigration of population. The water supply system needs to predict the change of water consumption in advance in order to provide users with stable water supply services. On the other hand, the water supply systems in many cities are old and prone to abnormal situations such as pipe bursts. If the reasonable range of water consumption in a period can be predicted in advance, once the tube burst occurs, it can be repaired as soon as possible. Therefore, accurate prediction of water consumption can improve the operation efficiency of urban water supply systems and reduce the operation cost.

II. Literature review

The urban water consumption prediction is a critical problem and has a really practical impact. So, many researchers have explored some useful models in terms of the problem. Abdüsselam [1] presented a Takagi Sugeno (TS) fuzzy method for predicting future monthly water consumption values from three antecedent water consumption amounts. Hongyan [2] proposed a Markov modified autoregressive moving average (ARIMA) model to predict the future daily water consumption data according to the periodicity and randomness nature of the daily water consumption data. Compared to the traditional time series prediction model, these models can get better accuracy in general.

With the development of artificial intelligence theory and computer hardware, more and more machine learning or deep learning techniques have been adapted into water consumption prediction. Altunkaynak [3] developed a monthly water consumption prediction model based on artificial neural networks (ANNs), in combination with data preprocessing techniques (Discrete wavelet transform & multiplicative season algorithm). Al-Zahrani [4] constructed the combined technique of Artificial Neural Networks and time series models based on the available daily water consumption and climate data for predicting the future daily water demand for AI-Khobar city in the Kingdom of Saudi Arabia. Haytham [5] adapted the Deep Learning technique (deep convolutional neural network DeepCNNs) in the hydrology domain and better accuracy across three water stations compared to state-of-the-art models (Artificial Neural Networks, Support Vector Machines, Wavelet-ANNs, Wavelet-SVMs) used in the hydrology applications.

Using recurrent neural networks to build sequence models has got great success in practical applications, such as time series prediction [6], natural language processing [7], image generation [8], audio recognition [9], video model [10]. Bejarano [11] designed a smart water prediction system that predicts future hourly water consumption based on historical data and demonstrated that the Long Short Term Memory (LSTM) based deep Recurrent Neural Network (RNN) model is able to accurately predict future hourly water consumption in advance using just the last 24 hours of data at test time. The paper of Said [12] implied that LSTM can be implemented on a univariate water consumption time series prediction problem and can make decent predictions for water consumption time series.

Compared to value prediction, probabilistic prediction is more practical in urban water consumption estimation problems. Cutore [13] proposed a probabilistic prediction of urban water consumption using the Shuffled Complex Evolution Metropolis algorithm. Hutton [14] presented a probabilistic methodology for quantifying, diagnosing, and reducing model structural and predictive errors in short-term water demand forecasting. Gagliardi [15] proposed a probabilistic short-term water demand forecasting model based on the Markov Chain. In the field of probabilistic prediction of water consumption, there is little research on the method of introducing deep learning, which has shown its strong power in the value prediction of water consumption.

III. DATASET

I worked as a research assistant in the past summer quarter and got the water consumption in two water stations in Shanghai, China. And also to consider more possible features, I crawled some data about climate, holidays, and environment from the website.

A. Water Consumption Data

There are two water consumption datasets. One is for Hanyang Water Station (HY), and another is for Wujing Water station (WJ). Figure 1 gives an intuition of what the data looks like. The data records two years' water consumption, and the frequency of the data for each station is 1 minute. So, for each dataset includes $(365+366)\times 24\times 60=1{,}052{,}640$ records for each station.

Figure 1. The water consumption of two stations (2019-2020)

B. Other features

Other possibly related features are collected to increase the accuracy of water consumption prediction. Four features are used in the paper. They are maximum temperature of the day, minimum temperature of the day, whether the day is weekend and whether the day is holiday. There are $365+366=731$ records in this dataset since only one record for every day.

IV. APPROACH

A. Preprocessing

From Figure 1, we can see some preprocessing needs to be done. To clean the outliners, the water consumption is regrouped by day. The mean and standard are computed. Abnormal values are detected using the 3σcriterion (Fig.2.) and mean values are used instead. Fig.3. shows the results of data cleaning.

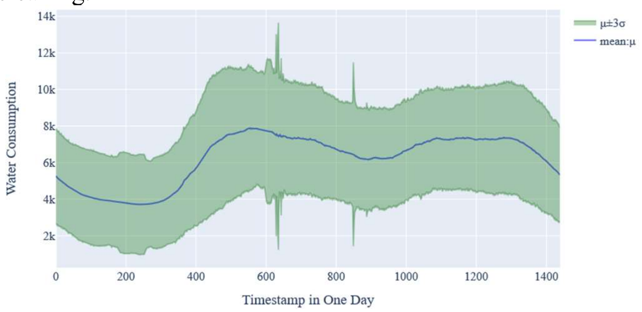


Figure 2. Clean HY station data with the 3σcriterion

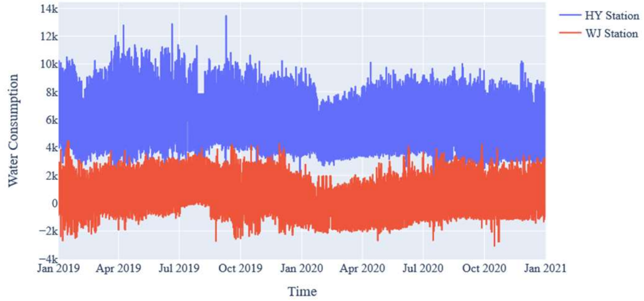


Figure 3. The denoised water consumption of HY and WJ (2019-2020)

For other features, the data needs to be expanded to match to the water consumption records. And for some special categorical variables, such as holidays, it is not informative enough to represent them as 0-1 variables. Because generally, the water consumption will increase before and after the festival, rather than only on the festival day. We use some special methods to make holidays be the continuous variable. (Fig.4.). In this figure, the holiday effect decays exponentially with the distance from the holiday.

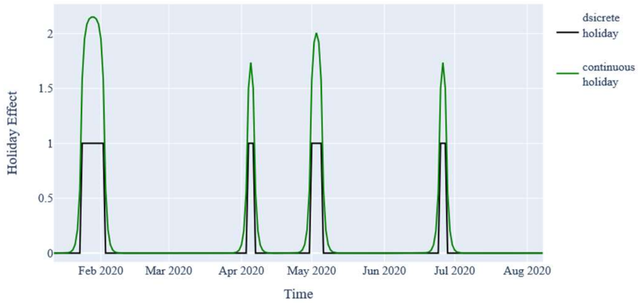


Figure 4. Continuous holiday variable

B. Water Consumption Prediction (value)

Considering the practicability of model, we hope that the model can predict the water consumption in the future. The daily water consumption is a vector of $24\times 60=1440$ dimensions. According to the pre-test, considering the model accuracy and the time required for training, directly outputting the complete water consumption of one day is not the best choice. Here we resample the water consumption of one day with sampling frequency $\Delta_{output}=20$ minutes. This means the output of the model is a vector with $1440/20\ +1=73$ (including the end point) elements.

The same resampling method can be used in the inputs. The same sampling frequency $\Delta_{input}=20$ minutes is used for resampling. Actually, inputs can include water consumption data from several days ago. Here, to simplify, as the base model, only previous day's data is used.

In order to get as many samples as possible, sliding window skill is used when extracting samples from dataset. Here, set the size of sliding window $s=5$ minutes, which means we extract one sample every 5 minutes. After using the sliding window skills to augment, we have enough data for training a deep learning model. But the start time of each sample is different. To solve this problem, the timestamps in every day of

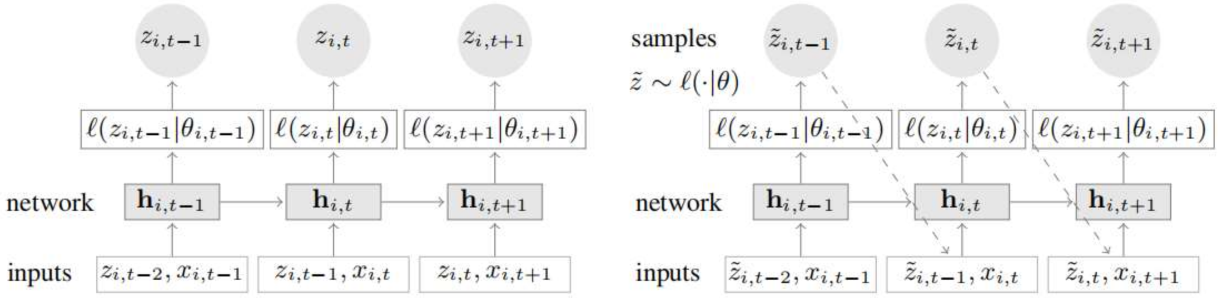


Figure 7 The architecture of DeepAR model in training (left) and predicting (right)

each sample are added as another inputs. Thus, after adding four other related features, the shape of the water consumption value prediction model's input is (None, 73, 6), and the shape of the model's output is (None, 73, 1).

With the above method, 209,949 samples are extracted from the dataset (in this part, only the data in HanYang Station is used for example). Split all samples into training set, development set and test set in a ratio of 9:1:1. It should be noted that here we need to split sequentially rather than randomly. Because this is a time series problem, samples are not independent between each other.

Models

The part refers to the official tutorials of time series forecast in TensorFlow [16]. Many models have been tried to do the regression work. The article only shows the six representative models and their prediction performance:

• Model-1 is 'Repeat model', which is regarded as the base model. Since this task is to predict 1 day into the future, given 1 day of the past, a simple approach is to repeat the previous day, assuming tomorrow will be similar.

• Model-2 is 'Dense' model. The input matrix is flattened as a vector. Then 1 dense layer with 'relu' activation function and 1 linear layer, which is equal to the dense layer with linear activation function, are stacked behind.

• Model-3 is 'CNN' model. Here we use 1 dimensional convolutional neural network since time series data has only two dimensions. To match the size of CNN filter (filter_size = 5 in the paper), only last 5 rows of input are left. Then one linear layer is added for outputs.

• Model-4 is 'LSTM' model. Only one layer LSTM is used here. One linear layer is added behind for outputs. Here LSTM model is just the simplest architecture, which predicts the entire out sequence in a single step.

• Model-5 is 'GRU' model. GRU model is similar to the LSTM model. Both of them are advanced Recurrent Neural Network. GRU model here is also the simplest architecture.

• Model-6 is 'BiLSTM model, which is a variate of LSTM. The model consists two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. The model should be the most complex model with most parameters.

The above six models only include one special layer in order to compare their prediction ability under the same other conditions as far as possible. For the detailed architecture of each model, such as the hidden units of each layer, please refer to the appendix.

Train models

Since it is the regression problem, mean square error is choosed as the loss function. In the equation, N is the number of samples, T is the number of time steps each output vector.

$$Loss = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T}\left(y_t^{(i)}-\hat{y}_t^{(i)}\right)^2}{N}$$

Use mini batch gradient descent to train models, where the minimum batch size is 256. Use Adam optimizer with 0.01 learning rate. Compile models in Keras and fit them. It will take no more than 30 minutes for each model to finish 100 epoches when use GPU accelerator(1 RTX 3070). From the loss plot, only the loss function of Dense model will oscillate in a certain range with the increase of training times. The loss functions of other models are monotonically decreasing.

Evaluation

Select Root Mean Square Error (RMSE) as the evaluation metric. Fig.5. shows the RMSE of different models. Table 1 shows the corresponding values. From the figure and table we can see that recurrent neural networks models (GRU, LSTM, BiLSTM) perform better than others, which is reasonable since RNN is designed for sequence problems. During all RNN models, GRU model performs best. The RMSE of the GRU model is only 468, which is 80% of the base mode – Repeat model.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\sum_{t=1}^{T}\left(y_t^{(i)}-\hat{y}_t^{(i)}\right)^2}{N}}$$

Table 5. The RMSE of 6 models (HanYang Station)

Model	Repeated	Dense	CNN
train	703	818	742
dev	594	574	549
test	558	535	554
Model	**LSTM**	**GRU**	**BILSTM**
train	636	656	628
dev	483	483	481
test	485	468	489

Figure 6 shows one random prediction result using different models intuitively. From the figure we can see that the RNN model can catch the key information, such as the peaks and waves when predicting.

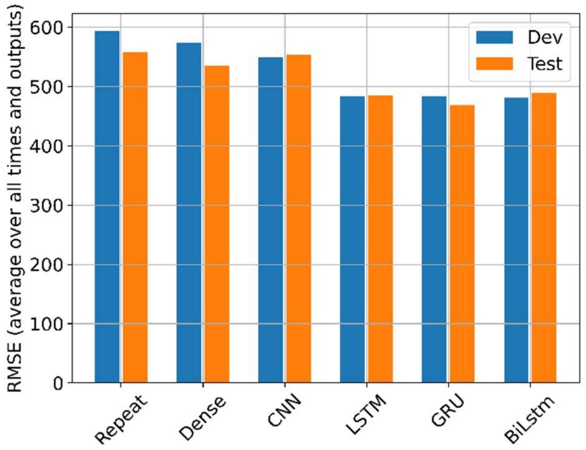


Figure 5. The RMSE of 6 models (HanYang Station)

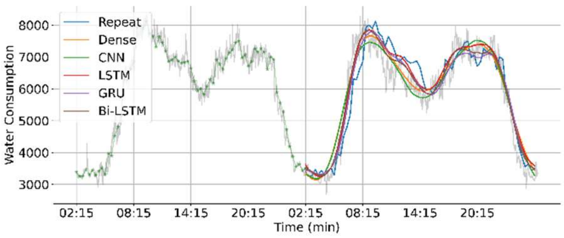


Figure 6. One prediction result in test dataset

C. *Water Consumption Prediction (probabilistic)*

In practical application, probabilistic prediction is more practical than point prediction. For example, as I mentioned at the beginning of the paper, the probabilistic prediction can give a water consumption range with confidence, which can help detect pipe bursts.

Theory of DeepAR

Salinas [17] proposed DeepAR, a methodology for producing accurate probabilistic forecasts, based on training an auto-regressive recurrent network model on related time series. The algorithm has the characteristics of strong learning ability and strong expansibility. It can not only learn multiple related time series at the same time but also consider different types of related variables. So, the state-of-the-art DeepAR algorithm was deployed here to build the probabilistic prediction model of water consumption.

The same notation as in the paper [17] is used here to introduce the model. Denoting the value of time series i at time t by $z_{i,t}$ and the value of associated covariateds i at time t by $x_{i,t}$, the goal of probabilistic prediction is to model the conditional distribution

$$P(z_{i,t_0:T}|z_{i,1:t_0-1}, x_{i,1:t_0-1})$$

Of the future of each time series $\left[z_{i,t_0}, z_{i,t_0+1}, \cdots, z_{i,T}\right] := z_{i,t_0:T}$ given its past $\left[z_{i,1}, \cdots, z_{i,t_0-2}, z_{i,t_0-1}\right] := z_{i,1:t_0-1}$, where t_0 denotes the time point from which we known for all time points.

The DeepAR model based on an autoregressive recurrent network architecture is summarized in Fig 7. DeepAR model assumes that the model distribution Q_Θ consists of a product of likelihood factors

$$Q_\Theta(z_{i,t_0:T}|z_{i,1:t_0-1}, x_{i,1:t_0-1}) = \prod_{t=t_0}^{T} l(z_{i,t}|\theta(h_{i,t}, \Theta))$$

parameterized by the output $h_{i,t}$ of an autogressive recurrent network

$$h_{i,t} = h(h_{i,t-1}, z_{i,t-1}, x_{i,t-1}, \Theta) \ (1)$$

where h is a function implemented by a multi-layer recurrent neural network with LSTM cells and Θ is the model parameters. The likelihood $l(z_{i,t_0}|\theta(h_{i,t}, \Theta))$ is a fixed distribution whose parameters are given by $\theta(h_{i,t}, \Theta)$. Since the water consumption is the real-valued data, Gaussian likelihood is used here. That is to say,

$$l(z_{i,t}|\theta(h_{i,t}, \Theta)) = l\left(z_{i,t}\middle|\mu_{i,t}, \sigma_{i,t}\right)$$
$$= (2\pi{\sigma_{i,t}}^2)^{-\frac{1}{2}} exp(-(z_{i,t} - \mu_{i,t})^2/(2{\sigma_{i,t}}^2))$$

where $\mu_{i,t} = w_\mu^T h_{i,t} + b_\mu$ and $\sigma_{i,t} = \log(1 + \exp(w_\sigma^T h_{i,t} + b_\sigma))$. Thus, The model consists of the parameters of the RNN $h(\cdot)$ as well as the parameters of $\theta(\cdot)$, which can be learned by maximizing the log-likelihood

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{t=t_0}^{T} l(z_{i,t}|\theta(h_{i,t}, \Theta))$$

where N is the number of time series.

Use Adma optimizer with learning rate = 0.001 to learn the parameters of the model. When it comes to prediction, h_{i,t_0-1} can be obtained by computing for $t = 1, \cdots, t_0 - 1$ first. Then for $t = t_0, t_0 + 1, \cdots, T$, sample $\tilde{z}_{i,t} \sim l\left(\cdot|\theta\left(\tilde{h}_{i,t}, \Theta\right)\right)$ where $\tilde{h}_{i,t} = h(h_{i,t-1}, \tilde{z}_{i,t-1}, x_{i,t-1}, \Theta)$ initialized with $\tilde{h}_{i,t_0-1} = h_{i,t_0-1}$ and $\tilde{z}_{i,t_0-1} = z_{i,t_0-1}$.

Model

Gluon Time Series (GluonTS) is the Gluon toolkit for probabilistic time series modeling, focusing on deep learning-based models.[18] GluonTS has implemented the API of DeepAR model.The paper calls the API in GluonTS directly to compile and train the model.

After large-scale random search and manual tuning, the parameters of the model are determined. The samples are got from resampling every 10 minutes. In this part, we use 5 days (720) in the past as inputs and 1 days (144) in the future as outputs. Split the data with the same ratio (9:1:1) used in the value prediction model. So, the training data is from 2019-01-01 to 2020-08-07. The development data is from 2020-08-07 to 2020-10-19. The left data belongs to the test data.

When predicting the water consumption of HanYang Station, the inputs include the water consumption series of HanYang Station, daily mean temperature and holiday information. When predicting the water consumption of WuJin Station, the performance will be better when we consider the water consumption series of HanYang Station. It should be noted that the weekends information are not included in the inputs. This is because the API has considered the time lag effects, which can reflect the weekends information from my perspective. Before fed into the model, all data (water consumption series and features) should be normalized.

The inner network of the DeepAR model is a two layers LSTM network with 50 hidden units in each layer. All remaining parameters remain at their default values. Compile the model and Train it. It will take no more than 1.5 hours for each model to finish 100 epoches when use GPU accelerator(1 RTX 3070).

Evaluation

Import extra metrics Mean Absolute Percentage Error (MAPE) and the percentage of the residuals more than 1000 (γ). The reason I introduced γ here is because according the background of the problem, ± 1000 is a value acceptable to the water supply company.

$$MAPE = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T}\left(\left|y_t^{(i)} - \hat{y}_t^{(i)}\right| / y_t^{(i)}\right)}{N*T}$$

Table 2 shows the metrics of DeepAR model in the test set. We can see that the DeepAR model performs well according to the three metrics. The model performs better in the WuJin Station data than HanYang Station. It can be found by studying the original data (Fig.3.) that from Jan 2020 to March 2020, the water consumption has been greatly reduced (might because of the Covid-19) of HanYang Station. So, it might affect the performance of model in the test set of HanYang Station.

Table 2. The metrics of DeepAR model in the test set

	RMSE	MAPE	% more than ±1000
HanYang Station	430	5.30%	3.60%
WuJin Station	331	6.00%	1.32%

Figure 8 shows a period of prediction results in the test set of HanYang Station. The black line is the observations, the green line is the predictions (median), and the red area is predictions ± 1000. A red point means the absolute difference between observation and prediction is more than 1000. The purple area means there are some pipe bursts during this period. From this figure we can see that the model can do well in alarming.

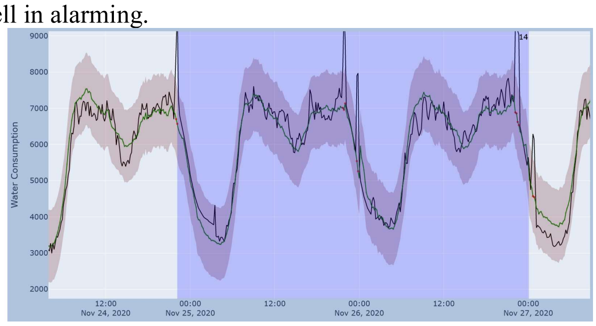


Figure 8. A period of prediction results in HanYang Station

Besides, I revised some source code in the API in order to make real-time updating of the model. That is to say, when it is used in reality, the model can update the parameters of the model with the lastes data.

V. Conclusion

According to the experimental results of this paper, it can be found that the revised recurrent neural network model represented by GRU can well deal with the problem of time series data prediction. In terms of water consumption prediction (value) in this paper, the water consumption sequence of the previous day is used to predict the water consumption sequence of the next day. Using GRU model, the root mean square error (RMSE) of the test set is only 80% of the model where previous day's water consumption is seen as equal as the next day's water consumption.

The Deep Autoregressive (DeepAR) model based on the recurrent neural networks shows satisfactory performance in probabilistic prediction. According to the experimental results of this paper, based on the DeepAR model, using the water consumption five days ago to predict the water consumption interval after one day, the mean absolute percentage error (MAPE) in test set can be controlled within 6%, which can be used for some pipe bursts alarming.

VI. Future Work

In the paper, two types of models can not be compared directly since the probabilistic predition model uses five days' data, however, the value prediction model only uses one day's data. More work needs be done to make the two kinds of models comparable.

References

[1] Altunkaynak, Abdüsselam, Mehmet Özger, and Mehmet Çakmakci. "Water consumption prediction of Istanbul city by using fuzzy logic approach." Water Resources Management 19.5 (2005): 641-654.

[2] Du, Hongyan, Zhihua Zhao, and Huifeng Xue. "ARIMA-M: A New Model for Daily Water Consumption Prediction Based on the Autoregressive Integrated Moving Average Model and the Markov Chain Error Correction." Water 12.3 (2020): 760.

[3] Altunkaynak, Abdusselam, and Tewodros Assefa Nigussie. "Monthly water consumption prediction using season algorithm and wavelet transform–based models." Journal of Water Resources Planning and Management 143.6 (2017): 04017011.

[4] Al-Zahrani, Muhammad A., and Amin Abo-Monasar. "Urban residential water demand prediction based on artificial neural networks and time series models." Water Resources Management 29.10 (2015): 3651-3662.

[5] Assem, Haytham, et al. "Urban water flow and water level prediction based on deep learning." Joint European conference on machine learning and knowledge discovery in databases. Springer, Cham, 2017.

[6] Connor, Jerome T., R. Douglas Martin, and Les E. Atlas. "Recurrent neural networks and robust time series prediction." IEEE transactions on neural networks 5.2 (1994): 240-254.

[7] Goldberg, Yoav. "Neural network methods for natural language processing." Synthesis lectures on human language technologies 10.1 (2017): 1-309.

[8] Gregor, Karol, et al. "Draw: A recurrent neural network for image generation." International Conference on Machine Learning. PMLR, 2015.

[9] Boulanger-Lewandowski, Nicolas, Yoshua Bengio, and Pascal Vincent. "Audio Chord Recognition with Recurrent Neural Networks." ISMIR. 2013.

[10] Ebrahimi Kahou, Samira, et al. "Recurrent neural networks for emotion recognition in video." Proceedings of the 2015 ACM on international conference on multimodal interaction. 2015.

[11] Bejarano, Gissella, et al. "Swap: Probabilistic graphical and deep learning models for water consumption prediction." Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation. 2019.

[12] Said, Norzanah Md, et al. "Univariate water consumption time series prediction using deep learning in neural network (DLNN)." International Journal of Advanced Technology and Engineering Exploration 8.76 (2021): 473.

[13] Cutore, P., et al. "Probabilistic prediction of urban water consumption using the SCEM-UA algorithm." Urban Water Journal 5.2 (2008): 125-132.

[14] Hutton, Christopher J., and Zoran Kapelan. "A probabilistic methodology for quantifying, diagnosing and reducing model structural and predictive errors in short term water demand forecasting." Environmental Modelling & Software 66 (2015): 87-97.

[15] Gagliardi, Francesca, et al. "A probabilistic short-term water demand forecasting model based on the Markov Chain." Water 9.7 (2017): 507.

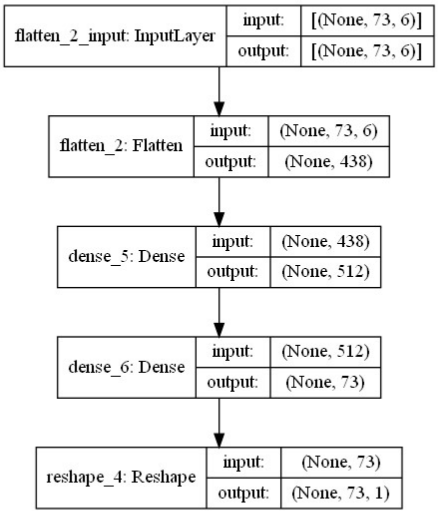
[16] https://www.tensorflow.org/tutorials/structured_data/time_series#advanced_autoregressive_model

[17] Salinas, David, et al. "DeepAR: Probabilistic forecasting with autoregressive recurrent networks." International Journal of Forecasting 36.3 (2020): 1181-119

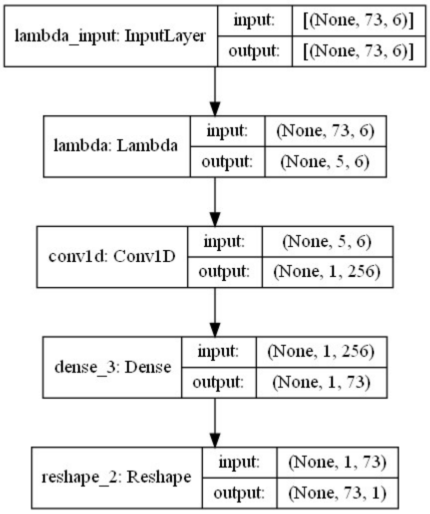
[18] https://ts.gluon.ai/api/gluonts/gluonts.model.deepar.html?highlight=deepar#module-gluonts.model.deepar

APPENDIX

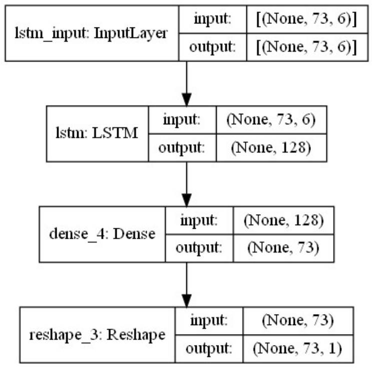
Dense



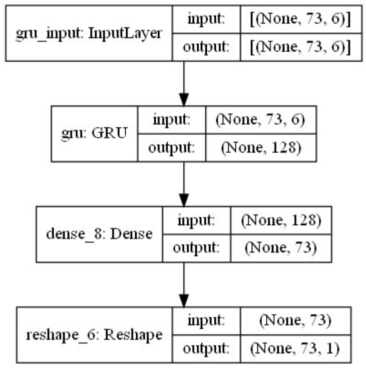
Dense



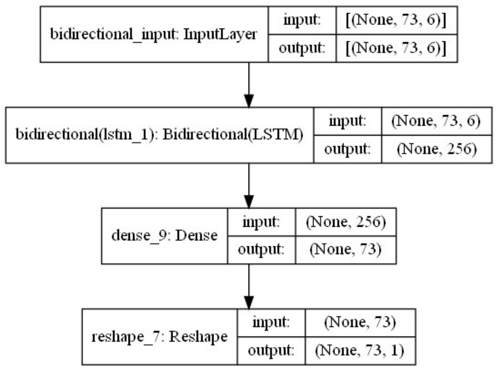
CNN



LSTM



GRU



BiLSTM