Real-time Water Consumption Prediction

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

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

***Index Terms*—water consumption, seq to seq, RNN, probabilistic prediction, DeepAR**

I. Introduction

The urban water supply system is an integral part of urban infrastructure. Water consumption has a particular periodic law. For example, more water is consumed in summer than in winter, and more water is consumed 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 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 water consumption prediction 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 practical impact. So, many researchers have explored some applicable models in terms of the problem. Reference [1] presented a Takagi Sugeno (TS) fuzzy method for predicting future monthly water consumption values from three antecedent water consumption amounts. Reference [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 generally get better accuracy.

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. Reference [3] developed a monthly water consumption prediction model based on Artificial Neural Networks, in combination with data preprocessing techniques (Discrete wavelet transform & multiplicative season algorithm). Reference [4] combined the 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. Reference [5] adapted the Deep Convolutional Neural Networks in the hydrology domain across three water stations and got the best accuracy 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) model could accurately predict future hourly water consumption in advance using just the last 24 hours of data.

Compared to value prediction, probabilistic prediction is more practical in urban water consumption estimation problems. Cutore [12] proposed a probabilistic prediction of urban water consumption using the Shuffled Complex Evolution Metropolis algorithm. Hutton [13] presented a probabilistic methodology for quantifying, diagnosing, and reducing model structural and predictive errors in short-term water demand forecasting. Gagliardi [14] 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 introducing deep learning, which has shown its strong power in the value prediction of water consumption.

III. Dataset

The dataset includes water consumption data in two water stations from 2019 to 2020 in Shanghai, China, and related features, such as temperatures and holidays.

A. Water Consumption Data

The water consumption data in HangYang (HY) and WuJing water (WJ) stations from 2019.01.01 to 2020.12.31 are recorded at a frequency of one per minute. So, each station includes $(365 + 366) \times 24 \times 60 = 1,052,640$ water consumption records. Fig. 1 presents a profile of water consumption trends of the two stations.

B. Related Features

Other possibly related features ranging from 2019.01.01 to 2020.12.31 are collected to increase the accuracy of the water consumption prediction. Four features are used in the prediction of water consumption value. They are the maximum temperature of the day, the minimum temperature of the day, whether the day is a weekend, and whether the day is a holiday. The data include $365 + 366 = 731$ records since the features are recorded at a frequency of one per day.

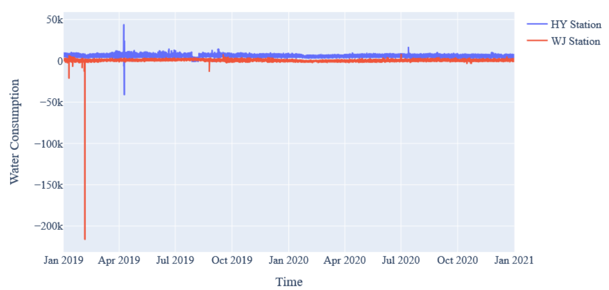

Fig. 1. The water consumption of two stations (2019-2020)

IV. Approach

A. Preprocess

From Fig. 1, the raw data need to be preprocessed before being fed into the water consumption prediction model. Group the water consumption data by day and compute the means and standard deviation of water consumption in a day. Abnormal values are detected using the 3σ criterion (Fig. 2) and replaced by mean values at the corresponding time. Fig. 3 presents the denoised water consumption data of HanYang and WuJing stations.

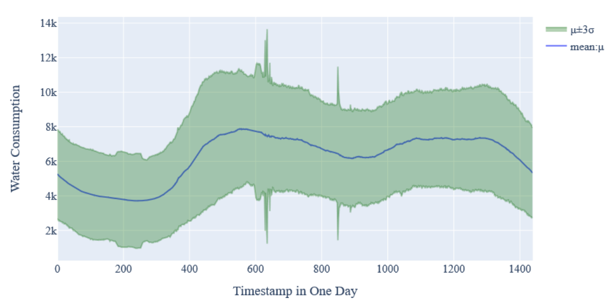

Fig. 2. Remove outliers of HY station data with the 3σ criterion

The data needs to be expanded to match the water consumption records for other related features. Moreover, some categorical variables, such as holidays, are 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. Transfer the holiday variable from the discrete variable into the continuous variable.

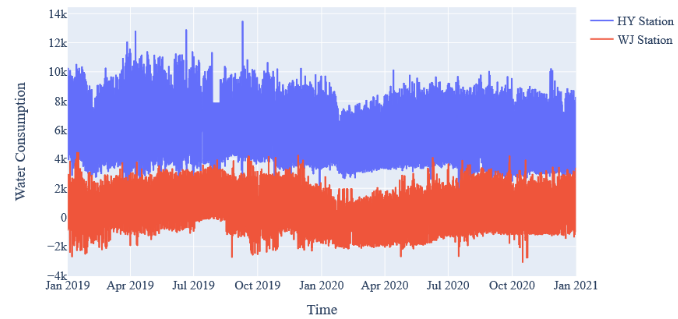

Fig. 3. The denoised water consumption of HY and WJ stations (2019-2020)

In Fig. 4, the impact of holidays on water consumption decays exponentially with the distance from the holiday. Finally, before being fed into models, all data (water consumption series and features) are normalized.

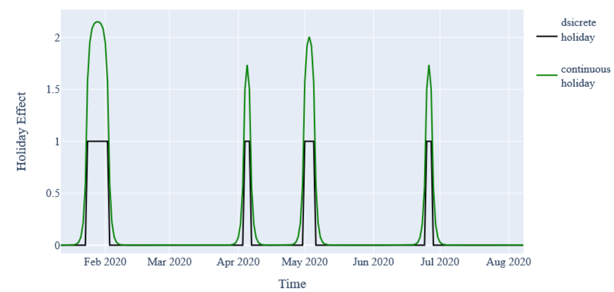

Fig. 4. Transfer the holiday variable

B. Prediction of water consumption value

The prediction model of water consumption value is established first. The water consumption data of HanYang station are used in the model. The daily water consumption data can be represented by a vector with $24 \times 60 = 1440$ elements. According to the pre-experiment, considering the model's performance and training time, outputting the whole water consumption data of one day is not the best choice. Resample the water consumption data of one day at the frequency of 20 minutes. The final output of the value prediction model should be a vector with $1440/20 + 1 = 73$ (including the endpoint) elements. The model's inputs can include water consumption data from several days ago. Here, only the previous day's data are used as inputs to simplify. The same resampling method is used to process the input data.

In order to get as many samples as possible, the sliding window skill is used while extracting samples from the water consumption dataset. Set the size of the sliding window $s = 5$ minutes, which means extracting one sample every 5 minutes. To solve the problem that each sample's starting time is different, the corresponding timestamps of a day are added as another input for each sample. Thus, after adding four related features, the shape of the model's input is (None, 73, 6), and the shape of the model's output is (None, 73, 1). Through the above processing, 209,949 samples are extracted from the dataset. Split all samples into the training set, development set, and test set sequentially in a ratio of 8:1:1.

1) Models: The part refers to TensorFlow's official tutorials of time series forecast [15]. Six representative models and their performances in predicting water consumption value will be discussed as follow.

- Model-1 is the 'Repeat model,' regarded as the base model. The goal of the problem is to predict one day in the future, given the previous day. It is reasonable to assume that tomorrow's water consumptions are similar to today's. A simple approach is to repeat the previous data as the estimation.
- Model-2 is the 'Dense' model. The input matrix is flattened as a vector. Then one dense layer with 'ReLu' activation function and one linear layer, which is equal to the dense layer with linear activation function, are stacked behind.
- Model-3 is the 'CNN' model. Since time-series data has only two dimensions, a one-dimensional convolutional neural network (CNN) is used. Only the last five rows of inputs are kept to match the size of the CNN filter ($filter\_size = 5$). Then one linear layer is stacked for outputs.
- Model-4 is the 'LSTM' model. The model includes one-layer Long Short-Term Memory (LSTM) networks and one linear layer. Here LSTM model is just the most straightforward architecture, which predicts the entire output sequence in a single step.
- Model-5 is the 'GRU' model. The Gated Recurrent Units (GRU) is similar to LSTM. Both are variants of recurrent neural networks and are good at dealing with long-term sequences. The architecture of the 'GRU' model includes one GRU layer and one linear layer, which is similar to the 'LSTM' model.
- Model-6 is the 'BiLSTM model. The Bidirectional LSTM (BiLSTM) model consists of two LSTM networks: one taking the input in a forward direction and the other in a backward direction. The model should be the most complex model with the most parameters.

The above six models only include one special layer to compare their prediction performance when other conditions are as identical as possible. Detailed hyper-parameters of each model can be found in the appendix.

2) Training and evaluation: Since it is a regression problem, the mean square error (1) is chosen as the loss function. In the equation, N is the number of samples, T is the number of time steps for each output vector.

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

Use mini-batch gradient descent to train models, where the minimum batch size is 256. Use Adam optimizer with a 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 epochs when using the GPU accelerator(1 RTX 3070) and converge. From the loss plot, only the loss function of the Dense model will oscillate in a specific range with the increase of training times. The loss functions of other models are monotonically decreasing. Select Root Mean Square Error (RMSE) as the evaluation metric (2):

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

Fig. 5 shows the RMSE of different models. Table I shows the corresponding values. The figure and table show 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, the GRU model performs best. The RMSE of the GRU model is 468, which is only 80% of the base mode – 'Repeat' model.

TABLE I
THE RMSE OF SIX MODELS (HY STATION)

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

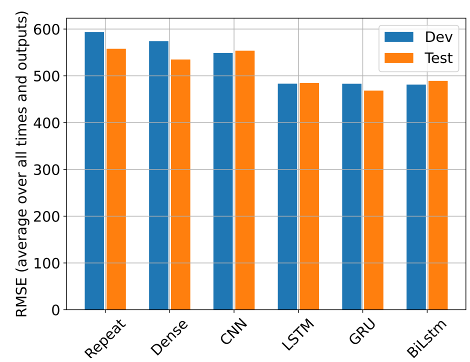

Fig. 5. The RMSE of six models (HY station)

Fig. 6 intuitively shows one random prediction result using different models. The figure implies that the RNN model can catch critical information, such as predicting peaks and waves.

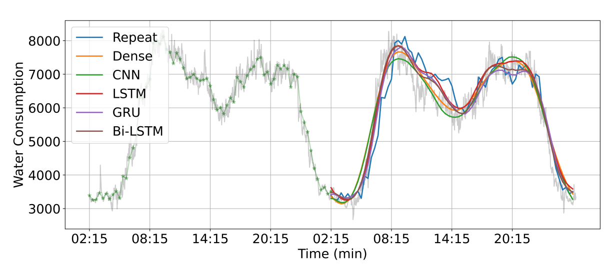

Fig. 6. One prediction result in the test set (HY station)

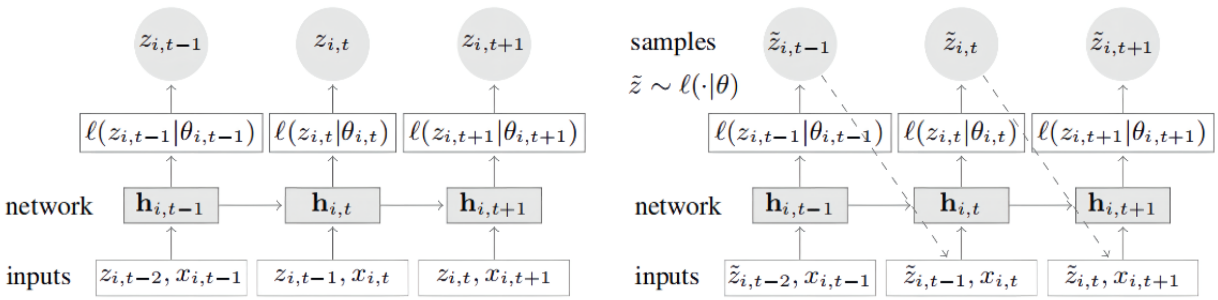

Fig. 7. The architecture of DeepAR model in training (left) and predicting (right)

C. Probability Prediction of Water Consumption

In practical application, probabilistic prediction is more practical than point prediction. For example, probabilistic prediction can give a water consumption range with confidence, which can help detect pipe bursts.

1) Deep Auto-Regressive (DeepAR): Reference [16] proposed DeepAR, a methodology for producing accurate probabilistic forecasts, based on training a deep auto-regressive recurrent network model on related time series. The algorithm has the characteristics of strong learning ability and strong extensibility. It can learn multiple related time series at the same time and 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 reference [16] is used here to introduce the algorithm. Denoting the value (water consumption) of time series i at time t by $z_{i,t}$ and the value of associated covariants (related features) i at time t by $x_{i,t}$, the goal of probabilistic prediction is to model the conditional distribution (3):

$$P(z_{i,t_0:T}|z_{i,1:t_0-1}, x_{i,1:t_0-1}) \tag{3}$$

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

The DeepAR model based on an auto-regressive recurrent network architecture is summarized in Fig.7. DeepAR model assumes that the model distribution Q_Θ consists of a product of likelihood factors (4)

$$Q_\Theta(z_{i,t_0:T}|z_{i,1:t_0-1}, x_{i,1:t_0-1}) = \prod_{t=t_0}^{T} l(z_{i,t}|\theta(h_{i,t}, \Theta)) \tag{4}$$

parameterized by the output $h_{i,t}$ of an auto-regressive recurrent network (5)

$$h_{i,t} = h(h_{i,t-1}, z_{i,t-1}, x_{i,t-1}, \Theta) \tag{5}$$

where h is a function implemented by a multi-layer recurrent neural network with LSTM cells and Θ is the model parameters. The likelihood $l(z_{i,t}|\theta)$ 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,

$$l(z_{i,t}|\theta) = ({2\pi\sigma_{i,t}}^2)^{-\frac{1}{2}} exp(-(z_{i,t} - \mu_{i,t})^2/(2{\sigma_{i,t}}^2)) \tag{6}$$

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 (7)

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{t=t_0}^{T} l(z_{i,t}|\theta(h_{i,t}, \Theta)) \tag{7}$$

where N is the number of time series.

Use Adam 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 $\widetilde{z}_{i,t}\ l\left(\cdot|\theta\left(\widetilde{h}_{i,t}, \Theta\right)\right)$ where $\widetilde{h}_{i,t} = h(h_{i,t-1}, \widetilde{z}_{i,t-1}, x_{i,t-1}, \Theta)$ initialized with $\widetilde{h}_{i,t_0-1} = h_{i,t_0-1}$ and $\widetilde{z}_{i,t_0-1} = z_{i,t_0-1}$.

2) Training the model: Gluon Time Series (GluonTS) is a toolkit developed by Amazon for probabilistic time series modeling [17], focusing on deep learning-based models. GluonTS has implemented the API of the DeepAR model [18]. The paper calls the API in GluonTS directly to compile and train the model. After large-scale random search and manual tuning, the model's parameters are determined. The samples are got from resampling every 10 minutes. In this section, the past five days' data represented by a vector with 720 elements are inputs, and the next day's data represented by a vector with 144 elements are outputs. Split the data with the same ratio (8:1:1) 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 remaining 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 the water consumption series of HanYang Station are considered. 'Weekend' variables (whether the day is a weekend) are not included in the inputs. Because the API in GluonTS has considered the time lag effects, reflecting the impact caused by weekends. The internal 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 epochs using the GPU accelerator(1 RTX 3070).

3) Evaluation: Import additional metrics Mean Absolute Percentage Error (MAPE) and the percentage of the absolute values of residuals more than 1000 (γ). According the background of the problem, 'real value ± 1000' is a value acceptable to the water supply company.

$$MAPE = \frac{\sum_{i=1}^{N}\sum_{t=1}^{T}\left|y_t^{(i)} - \hat{y}_t^{(i)}\right|/y_t^{(i)}}{N \times T} \tag{8}$$

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

TABLE II
THE METRICS OF THE DEEPAR MODEL IN THE TEST SET

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

Fig. 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 $predictions \pm 1000$. A red point means the absolute difference between observation and prediction is more than 1000. The purple area means there were some pipe bursts during this period. From this figure, we can see that the model can do well in alarming. Besides, the source code in the API in GluonTS

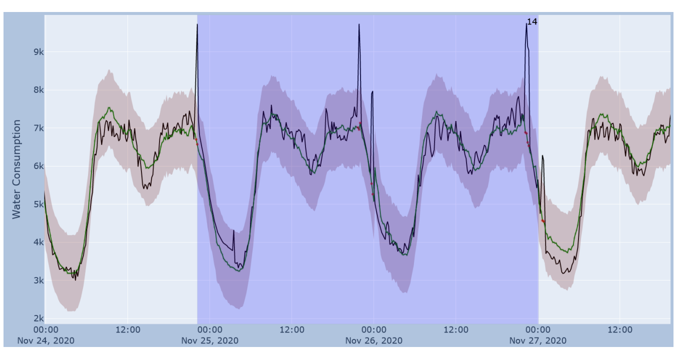

Fig. 8. A period of prediction results in HanYang Station

can be slightly adapted to update the model's parameters in real-time. That is to say, when it is used in reality, the model can make real-time probability predictions of water consumption with the latest data.

V. CONCLUSION

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

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

REFERENCES

[1] A. Altunkaynak, M. Özger, and M. Çakmakci, "Water consumption prediction of istanbul city by using fuzzy logic approach," *Water Resources Management*, vol. 19, no. 5, pp. 641–654, 2005.

[2] H. Du, Z. Zhao, and H. 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*, vol. 12, no. 3, p. 760, 2020.

[3] A. Altunkaynak and T. A. Nigussie, "Monthly water consumption prediction using season algorithm and wavelet transform–based models," *Journal of Water Resources Planning and Management*, vol. 143, no. 6, p. 04017011, 2017.

[4] M. A. Al-Zahrani and A. Abo-Monasar, "Urban residential water demand prediction based on artificial neural networks and time series models," *Water Resources Management*, vol. 29, no. 10, pp. 3651–3662, 2015.

[5] H. Assem, S. Ghariba, G. Makrai, P. Johnston, L. Gill, and F. Pilla, "Urban water flow and water level prediction based on deep learning," in *Joint European conference on machine learning and knowledge discovery in databases*, pp. 317–329, Springer, 2017.

[6] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 240–254, 1994.

[7] Y. Goldberg, "Neural network methods for natural language processing," *Synthesis lectures on human language technologies*, vol. 10, no. 1, pp. 1–309, 2017.

[8] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," in *International Conference on Machine Learning*, pp. 1462–1471, PMLR, 2015.

[9] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio chord recognition with recurrent neural networks.," in *ISMIR*, pp. 335–340, Citeseer, 2013.

[10] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal, "Recurrent neural networks for emotion recognition in video," in *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pp. 467–474, 2015.

[11] G. Bejarano, A. Kulkarni, R. Raushan, A. Seetharam, and A. Ramesh, "Swap: Probabilistic graphical and deep learning models for water consumption prediction," in *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 233–242, 2019.

[12] P. Cutore, A. Campisano, Z. Kapelan, C. Modica, and D. Savic, "Probabilistic prediction of urban water consumption using the scem-ua algorithm," *Urban Water Journal*, vol. 5, no. 2, pp. 125–132, 2008.

[13] C. J. Hutton and Z. Kapelan, “A probabilistic methodology for quantifying, diagnosing and reducing model structural and predictive errors in short term water demand forecasting,” *Environmental Modelling & Software*, vol. 66, pp. 87–97, 2015.
[14] F. Gagliardi, S. Alvisi, Z. Kapelan, and M. Franchini, “A probabilistic short-term water demand forecasting model based on the markov chain,” *Water*, vol. 9, no. 7, p. 507, 2017.
[15] “Time series forecasting :   tensorflow core.”
[16] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, “Deepar: Probabilistic forecasting with autoregressive recurrent networks,” *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.
[17] “Gluonts.model.deepar package.”

VI. Appendix

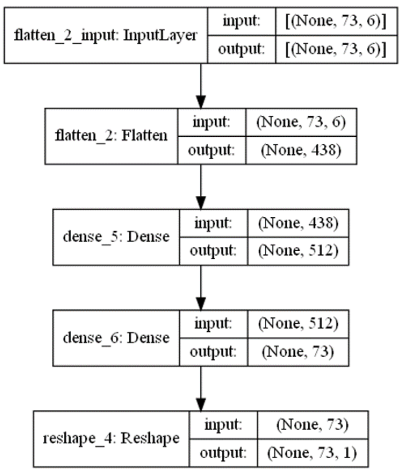

Fig. 9. The architecture of the ’Dense’ model

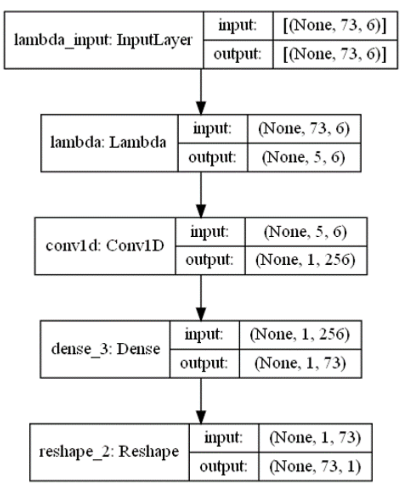

Fig. 10. The architecture of the ’CNN’ model

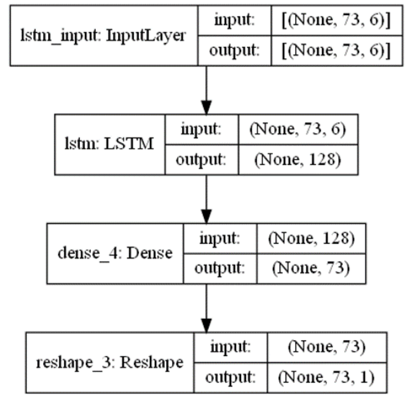

Fig. 11. The architecture of the ’LSTM’ model

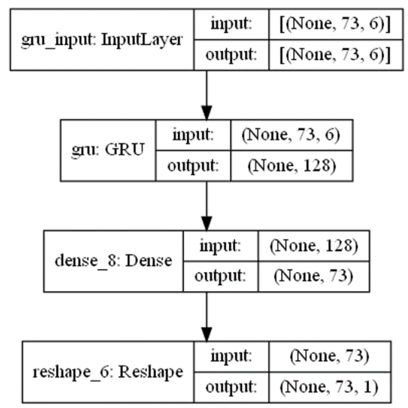

Fig. 12. The architecture of the ’GRU’ model

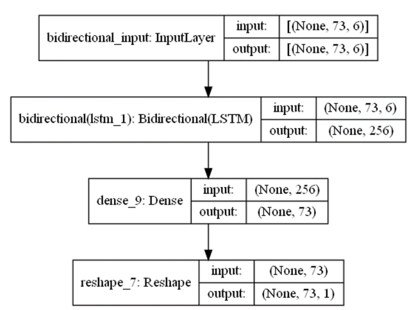

Fig. 13. The architecture of the ’BiLSTM’ model