Data-driven emulation of computationally expensive urban drainage simulators using deep learning and automatic hyperparameter optimization

M. Mahmoodian[1,2,*], U. Leopold[1]

[1]*Luxembourg Institute of Science and Technology (LIST), Esch-sur-Alzette, 4362, Luxembourg*

[2]*Delft University of Technology, Delft, 2628 CD, the Netherlands*

**Corresponding author email: mahmood.mahmoodian@list.lu*

Highlights

- Application of deep learning for data-driven emulation in urban drainage domain.
- Automatic hyperparameter optimization and model selection for deep learning architecture.
- Acceleration of a computationally expensive urban drainage simulator with low accuracy cost.

Introduction

Employing computationally expensive and highly detailed simulators in applications such as model-based real-time control (RTC), uncertainty analysis, calibration, optimization or sensitivity analysis can be a challenge; since in such applications numerous, frequent and fast simulations are required. Data-driven emulation is an approach to tackle this challenge. An 'emulator' or 'surrogate model' can be developed based on data derived from running the computationally expensive simulator. To the best of our knowledge and according to the literature, application of Deep Learning (DL) techniques in data-driven emulation of urban drainage simulators has been absent (or rare, if any). Traditional recurrent neural networks (RNNs) have been among the popular techniques for sequence learning and time series prediction. However, they suffer from 'short-term memory' problem. This is due to the 'vanishing gradient' issue which can occur during back-propagation towards the initial hidden layers of the neural network. Long-Short-Term-Memory (LSTM) (Hochreiter & Schmidhuber, 1997) and its more recent version Gate Recurrent Unit (GRU) (Cho et al., 2014), are two varieties of RNNs which overcome this issue by employing some additional gates in internal structure of nodes. Therefore, LSTM and GRU networks were selected as the candidate DL techniques in this research. Hyperparameter tuning of deep neural networks is normally a time-consuming effort which requires machine learning (ML) expertise as well. Manual trade-off, grid search and random search are still among the most common approaches in practice. In comparison, automatic hyperparameter optimization methods can facilitate ML accessible to layman users as well as professionals. Sequential model-based optimization is one of the most promising approaches in this regard which was selected for hyperparameters optimization and model selection in this study (Snoek et al., 2012).

Methodology

Case Study

The case study area in this research is a small part of Haute-Sûre urban drainage catchment located in north-west of Luxembourg. A detailed simulator (model) was developed for this catchment using a commercial software. The software enables both wastewater quantity and quality modelling by considering numerous physical and chemical processes as well as network details, which can result in a computationally expensive simulator for some ad-hoc applications. Seven years of observed rainfall time series, with 10 minutes time resolution, was used to run the detailed simulator and extract the relevant input-output time series data to develop the emulators for wastewater quantity and quality. The focus of emulation in this study was specifically on a combined sewer overflow (CSO) structure in the case study area. Three time series indicating rainfall (mm/hr), daily dry weather flow pattern, and pump outflow from the retention tank (m3/s) were used to construct the quantity emulator to predict total volume in the retention tank (m3). Two additional time series indicating storage tank volume (m3) and daily patter of water quality indicator were used to construct the quality emulator to predict the total NH4 concentration (kg/m3) in the retention tank. 80% of the post-processed data was used to train the emulators and 20% for validation purpose.

Deep Learning Techniques
LSTM and GRU networks have shown promising results in different fields water science and engineering (Guo et al., 2018; Hu et al., 2018; Widiasari et al., 2018; Qin et al., 2019) and more specifically urban drainage simulation domain (Zhang et al., 2018a,b; Yaqub et al., 2020). The fundamental difference between LSTM and GRU neural networks and traditional RNNs is their internal 'gated' mechanisms which solves the vanishing gradient issue mentioned earlier. The gates help to 'forget', 'remember' or 'update' the information according to the past seen data in the sequence. Hence, LSTM and GRU networks were selected as the candidate deep learning techniques in this research for data-driven emulation. Traditional Multilayer Perceptron (MLP) neural network was used as the baseline technique for comparison purpose.

Hyperparameter Optimization and Model Selection
Hyperparameter optimization in this research was implemented via sequential model-based optimization by taking advantage of the 'scikit-optimize' library developed in Python programming language (Scikit-optimize, 2019). Based on different model performance values obtained from different combinations of hyperparameters, scikit-optimize develops an internal surrogate model (e.g. a Gaussian Process or Random Forest model). This surrogate model, which is computationally cheap, is used to estimate the next combination of hyperparameters for evaluation. A common approach in selecting the next promising combination of hyperparameter values is to optimize a measure such as expected improvement in model performance. Three main hyperparameters were considered in this study for optimization, including: the number of neural network layers (1, 2 or 3); the number of nodes in each layer (between 10 and 300), and the learning rate (between 0.001 and 0.1). We also considered the RNN type (LSTM or GRU) as another hyperparameter. This way, it was possible to implement hyperparameter optimization and model selection simultaneously.

Results and discussion

After performing hyperparameter optimization a GRU network with one deep layer, 192 nodes and a learning rate of 0.006 was selected for the quantity emulator; while an LSTM architecture with one deep layer, 180 nodes and a learning rate of 0.004 was preferred for quality emulator. R2 Score, Nash–Sutcliffe efficiency coefficient (NSE) and Volumetric Efficiency (VE) were used for quantification of emulation error using the unseen data (the closer to 1 the better the results). Figure 2 illustrates validation results for long-term emulation using the unseen data for both emulators. Figure 3 shows the distribution of emulation error metrics for short-term emulation using both emulators. The acceleration factor in this specific case study was ≈ 1250 (i.e. the emulators were approximately 1250 times faster than the detailed original simulator in this case).

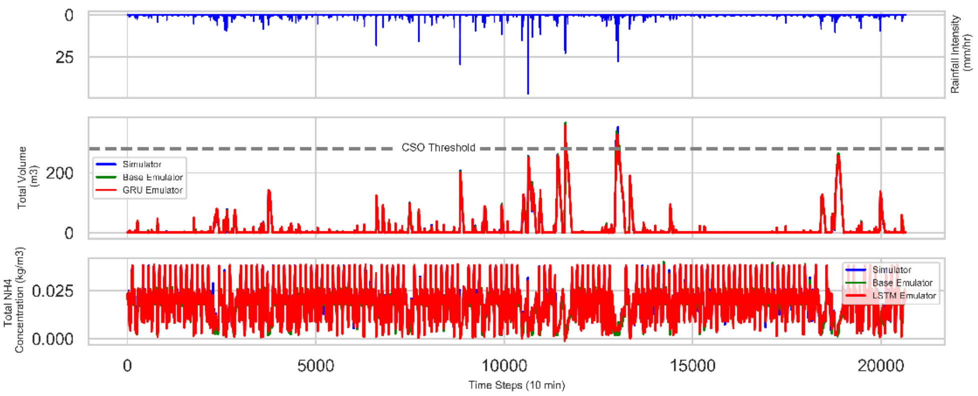

Figure 2. Validation results for the quantity and quality emulators for the test (unseen) dataset. Quantity emulator performance metrics: (VE_DL = 0.922, VE_MLP = 0.902, R2_DL= 0.995, R2_MLP = 0.986); quality emulator performance metrics (NSE_DL = 0.984, NSE_MLP = 0.941, R2_DL = 0.985, R2_MLP= 0.944). DL refers to GRU or LSTM emulators. MLP refers to the baseline emulator.

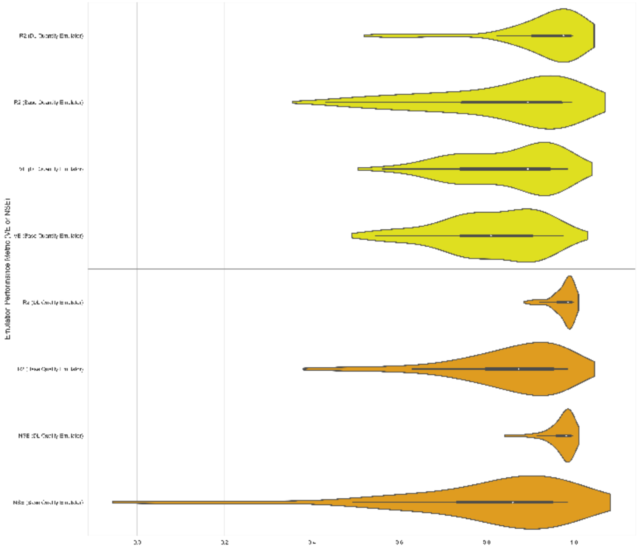

Figure 3. Violin plots showing the distribution of emulation error metrics (R2, NSE, VE) for short-term daily predictions with quantity (yellow) and quality (orange) emulators.

Conclusions and future work

The results showed a promising potential of the introduced approach for data-driven emulation in urban drainage domain. The presented approach is generic and fully data-driven; rather easier to implement in comparison to other data-driven emulation techniques (e.g. Gaussian Processes); independent from type of emulator (quantity or quality); and resulted in a considerable simulation acceleration gain together with a low accuracy cost. Developing multi-output emulators for larger case studies and their applications in computationally demanding tasks (e.g. RTC, uncertainty propagation, sensitivity analysis, and model calibration) can be considered valuable future research directions.

References

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. *ArXiv Preprint ArXiv:1409.1259*, 103–111. https://doi.org/10.3115/v1/w14-4012

Guo, G., Liu, S., Wu, Y., Li, J., Zhou, R., & Zhu, X. (2018). Short-Term Water Demand Forecast Based on Deep Learning Method. *JOURNAL OF WATER RESOURCES PLANNING AND MANAGEMENT*, *144*(12). https://doi.org/10.1061/(ASCE)WR.1943-5452.0000992

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 1735–1780. https://doi.org/https://doi.org/10.1162/neco.1997.9.8.1735

Hu, C., Wu, Q., Li, H., Jian, S., Li, N., & Lou, Z. (2018). Deep Learning with a Long Short-Term Memory Networks Approach for Rainfall-Runoff Simulation. *WATER*, *10*(11). https://doi.org/10.3390/w10111543

Qin, J., Liang, J., Chen, T., Lei, X., & Kang, A. (2019). Simulating and Predicting of Hydrological Time Series Based on TensorFlow Deep Learning. *POLISH JOURNAL OF ENVIRONMENTAL STUDIES*, *28*(2), 795–802. https://doi.org/10.15244/pjoes/81557

Scikit-optimize. (2019). scikit-optimize: sequential model-based optimization in Python. Retrieved from https://github.com/scikit-optimize/scikit-optimize

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 2951–2959.

Widiasari, I. R., Nugoho, L. E., Widyawan, & Efendi, R. (2018). Context-based Hydrology Time Series Data for A Flood Prediction Model Using LSTM. In Facta, M and Riyadi, MA and Widianto, ED and Arfan, M (Ed.), *2018 5TH INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY, COMPUTER, AND ELECTRICAL ENGINEERING (ICITACEE)* (pp. 385–390). 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE.

Yaqub, M., Asif, H., Kim, S., & Lee, W. (2020). Modeling of a full-scale sewage treatment plant to predict the nutrient removal efficiency using a long short-term memory (LSTM) neural network. *Journal of Water Process Engineering*, *37*(May), 101388. https://doi.org/10.1016/j.jwpe.2020.101388

Zhang, D., Lindholm, G., & Ratnaweera, H. (2018). Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring. *JOURNAL OF HYDROLOGY*, *556*, 409–418. https://doi.org/10.1016/j.jhydrol.2017.11.018

Zhang, D., Martinez, N., Lindholm, G., & Ratnaweera, H. (2018). Manage Sewer In-Line Storage Control Using Hydraulic Model and Recurrent Neural Network. *WATER RESOURCES MANAGEMENT*, *32*(6), 2079–2098. https://doi.org/10.1007/s11269-018.