Water Resour Manage (2018) 32:2079–2098
https://doi.org/10.1007/s11269-018-1919-3

Manage Sewer In-Line Storage Control Using Hydraulic Model and Recurrent Neural Network

Duo Zhang[1] · **Nicolas Martinez**[1] · **Geir Lindholm**[2] ·
Harsha Ratnaweera[1]

Received: 27 March 2017 / Accepted: 29 January 2018 /
Published online: 15 February 2018

Abstract This paper described manage sewer in-line storage control for the city of Drammen, Norway. The purpose of the control is to use the free space of the pipes to reduce overflow at the wastewater treatment plant (WWTP). This study combined the powerful sides of the hydraulic model and neural networks. A detailed hydraulic model was developed to identify which part of the sewer system have more free space. Subsequently, the effectiveness of the proposed control solution was tested. Simulation results showed that intentionally control sewer with free space could significantly reduce overflow at the WWTP. At last, in order to enhance better decision making and give enough response time for the proposed control solution, Recurrent Neural Network (RNN) was employed to forecast flow. Three RNN architectures, namely Elman, NARX (nonlinear autoregressive network with exogenous inputs) and a novel architecture of neural networks, LSTM (Long Short-Term Memory), were compared. The LSTM exhibits the superior capability for time series prediction.

Keywords Sewer system · Hydraulic model · Recurrent neural network · LSTM

1 Introduction

Most people agree that climate change has caused, and will cause, more intensive and often extreme weather. In this context, in order to reduce overflow at the WWTP, the sewer system needs increased capacity, either by physical expansions or by investing in separate sewers. This will require a huge amount of investments. Taking Norway as an example, the estimated value of sewer networks and WWTPs in Norway is around 500 billion Norwegian Kroner (NOK). At least 150 billion NOK investment by 2030 will be required to maintain nowadays service level (Ødegård et al. 2013).

✉ Duo Zhang
duo.zhang@nmbu.no

[1] Faculty of Sciences and Technology, Norwegian University of Life Sciences, 1432 Ås, Norway

[2] Rosim AS, Brobekkveien 80, 0582 Oslo, Norway

During heavy rainfall events, parts of the sewer system will experience overload but other parts may only partially filled. As an alternative solution, there is a potential in utilizing left over capacities in the sewer system to reduce the hydraulic load of the WWTP. Maximum sewer's spatially distributed in-line storage capacity is a cost-effective method of reducing the overflow at the WWTP compare to capital construction (Darsono and Labadie 2007). Properly manage the sewer system over time and space could aggregate in-line storage capacity in a sewer system, reduce pollution from untreated WWTP overflows. In addition, incorporation of in-line storage control into plans for constructing additional storm control facilities such as detention basin may reduce size and investment of these facilities (Darsono and Labadie 2007). This approach could efficiently reduce infrastructure investments, but require adequate software and modeling capabilities. (Grum et al. 2011; Garofalo et al. 2017).

The successfulness of sewer in-line storage control relies on high-quality information about the sewer system. There are two critical tasks in the present study. First, we should identify which part of the sewer system is suitable for control, i.e. have free space during rainfalls. Second, successful sewer system control requires not only the current but also the future flow information (Liu et al. 2016; Chen et al. 2014; Duchesne et al. 2001). So that we need to forecast flow in the sewer system to enhance sewer in-line control structure operations in real time.

With complete knowledge of sewer system and rainfall pattern, the hydraulic model is suitable for task 1. Hydraulic models are the most common tools in most of the studies about sewer system (Autixier et al. 2014; Lucas and Sample 2015; Seggelke et al. 2005). Simulation results from hydraulic models can supply insight into their functioning and show the effects of different control strategies after a rainfall event (Chiang et al. 2010). However, hydraulic models require detailed information of the sewer system, manually operation, a large number of parameters and longer computational time. Although hydraulic models provide a solid understanding of the hydraulic behavior, their features make these models limited adequate for application in task 2 (El-Din and Smith 2002).

To overcome the limitations of the hydraulic model, enabling control system make quick and intelligent decisions, machine learning is among the top methods. Alam et al. (2016) examined the efficiency of eight mainstream machine learning algorithms for the Internet of Things (IoT) data. Includes Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), and the deep learning artificial neural networks (DLANNs). The preliminary results on three real IoT datasets showed that ANNs and DLANNs could provide highly accurate predictions.

Two of the hottest topics in the deep learning field are improving computer visioning using convolution neural networks (CNN) and modeling sequential data using the recurrent neural network (RNN). Flow time series is a kind of typical sequential data. Traditional time series prediction mainly relies on memoryless models. Such as the autoregressive model, which predict the next step in a time series from a fixed number of previous steps. Facilitate time delay units through feedback connections, RNNs can be trained to learn sequential or time-varying patterns (Chang et al. 2014a, b). In the context of giving a precise and timely prediction of flow in the sewer system, the RNN is particularly suitable for task 2.

The earliest research on RNN took place in the 1980s. The Hopfield networks introduced by Hopfield in 1982 (Hopfield 1982) initialized the concept of RNN. Jordan in 1997 introduced one of the earliest architecture for supervised learning on sequences (Jordan 1997). The Jordan RNN is a feedforward network with a hidden layer equipped with special units. Output values feed these values to the hidden nodes at the following time step, according to the state of the special units. If the output values are actions, the special units allow the network to remember actions taken at previous time steps (Lipton et al. 2015). The Elman RNN (Elman 1990.) was introduced in 1990, which takes the state of the hidden node at the previous time step as input for the current time step. This architecture is equivalent to a simple RNN in which each hidden node has a single self-connected recurrent edge.

More and more modern RNN architectures were proposed since the late 1990s. Based on the Hopfield networks and Restricted Boltzmann Machine, Hinton et al. (Hinton et al. 2006) showed how a many-layered neural network, namely deep belief nets, could be pre-trained one layer at a time. This led to one of the first effective deep learning algorithms. Because of Hinton's achievement, the term "deep learning" begins to gain popularity. Another typical modern RNN architecture is the nonlinear autoregressive exogenous model (NARX) (Siegelmann et al. 1997). In water resource field, typical applications of NARX include monthly groundwater levels prediction (Chang et al. 2016), flood inundation nowcast (Chang et al. 2014a, b), water quality modeling (Chang et al. 2015), groundwater arsenic concentrations forecast (Chang et al. 2013) and water level prediction for pump stations (Chang et al. 2014a, b).

One of the most successful RNN architecture is the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997). This architecture replaced ordinary hidden layer node with a memory cell. The memory cell could store, write, or read data, via gates that open and close, just like data in a computer's memory. Using LSTM, Ma et al. (2015) developed an intelligent transportation system. This study shows that LSTM is able to capture nonlinear time series dynamic in an effective manner, compare to memoryless models and traditional RNNs. Recently, a few studies have explored the application of LSTM in the water resource related fields. Remesan and Mathew (2014) introduced LSTM as a machine learning and artificial intelligence based approach for hydrological time series modeling. Xingjian et al. (2015) built a precipitation nowcasting model. Experiments show that LSTM network captures spatiotemporal correlations better and consistently outperforms other models for precipitation nowcasting. Zaytar and Amrani (2016) used LSTM with previous 24 h' values as input, to forecast weather data in the next 24 and 72 h.

Consider the advantages and disadvantages of hydraulic models and machine learning algorithms, a trend of nowadays research is combining the powerful side of hydraulic models and machine learning algorithms. Use storm water management model (SWMM, Rossman 2010) and Jordan neural network, Darsono and Labadie (2007) studied the real-time regulation of combined sewer overflows. Based on synthetic data generated from SWMM based on the data from nearby gauging stations, Chiang et al. (2010) trained a NARX network and built a relationship between rainfalls and water level patterns of an ungauged sewerage system. Yu et al. (2013) studied sewer system in Tokyo use both hydraulic model and machine learning. Hydraulic model is first used to simulate the sewer pipe. Then clustering analysis was applied to simulated data for categorizing rainfalls and CSOs.

Summarize above literature reviews, its possible to conclude that hydraulic model could supply critical information about which part of the sewer still have left over capacity, but it is too slow to make the real-time response. On the contrast, although the machine learning algorithms such as RNNs provide real-time forecasting, it cannot give us an insight of the sewer system. Besides, to the best of our knowledge, there are rare applications of LSTM in the water recourse related domain, as state of the art RNN architecture, the effectiveness of LSTM need to be investigated.

In the present study, we first using the hydraulic model to identify relatively dry pipes (control target), and test the proposed in-line control strategy. Then use RNNs realize flow prediction for the target pipe. The remainder of this paper is organized as follows: a general description about study area, hydraulic model and three RNN algorithms, namely Elman, NARX (nonlinear autoregressive network with exogenous inputs) and LSTM (Long Short-Term Memory), is provided in the first section. Then simulation based on the hydraulic model for different return periods and control scenarios were presented in section two. In the third section, the prediction efficiency of the three RNN algorithms was compared. Conclusion and future envision were discussed at the end of this paper.

2 Method and Data

2.1 Case Study Area

Based on the concept of sewer in-line storage control, the Drammen government initialized the Regnbyge 3 M project. The ultimate goal of this project is integrate intelligent monitoring, modeling and control solutions, manage sewer system and WWTP in a holistic way, thus reducing overflow at the WWTP during extreme rainfall through efficiently utilizing the in-line storage capacities of the sewer system.

Figure 1 is an overview of the case study area, Drammen, Norway (59.44 N, 10.12 E). This city locates in the southeast of Norway. It is the largest city and the capital of the county of Buskerud with more than 150,000 inhabitants. In Drammen, the traditional city center distributed along the Drammen Fjord, important infrastructures, such as train station, shopping center and stadium are located at the southern bank of the fjord. The Drammen sewer system is

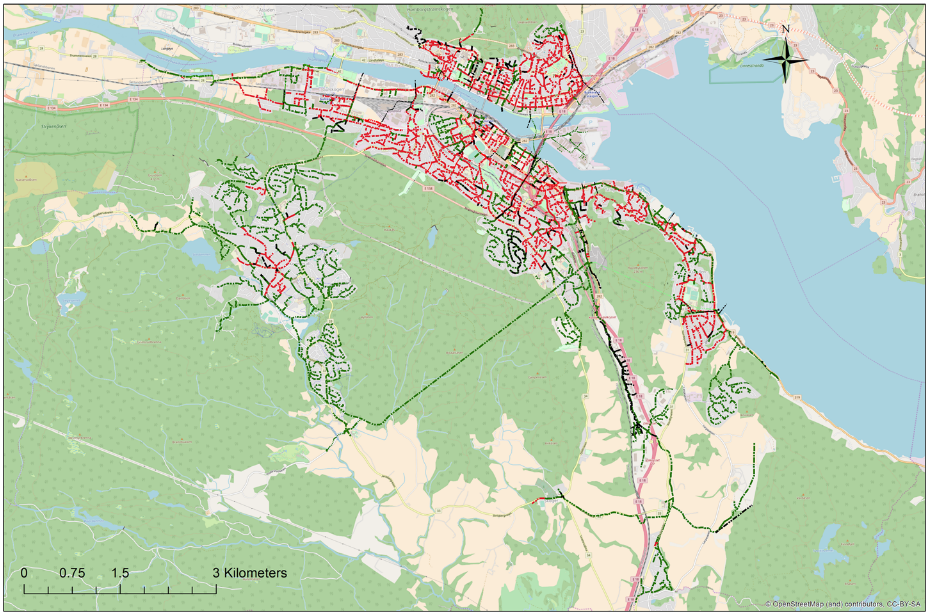

Fig. 1 Overview of Drammen city, Norway

a gravity system. Most of the sewer system in the central area are combined sewer. The southwest part of Drammen is the major residential areas, mostly use separate sewer.

The drainage area for the Drammen sewer system is around 15 Km^2, the total length of the sewer is approximately 500 km. The Solumstrand WWTP is the major WWTP in Drammen, with the designed treatment capacity of 130,000 PE (population equivalents). Overflow is the main problem today for Solumstrand WWTP. Physical expansion or construct separate sewer for the current network will take a long time and a lot of money.

2.2 Monitoring of the Sewer System

The sewer system in Drammen follows a tree structure, in which a series of sub-catchments converges into trunk conduits. At the end of the sewer system, all the trunk conduits link to one final collector pipe, deliver the wastewater to the WWTP. The outlet pipes, which connect sub-catchments to main collector sewer, and the collector sewer is the main large pipes in Drammen sewer system. Figure 2 displayed large pipes of the Drammen sewer system marked by dark yellow color (Table 1).

As the first step of the Regnbyge 3 M project, in order to collect data for further analysis and design the control strategy, flow sensors and water level sensors (NIVUS GmbH; Germany) were installed inside the main large pipes, and the locations of these sensors were marked with a brown point in Fig. 2. The flow was calculated based on water level, velocity and sharp of the pipes. The rain gauge was used to record the rainfall data.

2.3 Hydraulic Model

In order to identify the spatially distributed free space, a full detailed hydraulic model for the Drammen sewer system was developed (Fig. 3). This sewer hydraulic model was developed using Rosie. Rosie is an ArcGIS additional application based on MOUSE DHI (DHI group 2014) for planning, sizing and modeling of water distribution and sewerage systems, developed by Rosim AS, Norway. The direct response from the rainfall is calculated by the time–area (T-A) curve method (runoff model A). The runoff generated gradually from the previous hydrological processes accumulated as interflow and base flow is calculated by Rainfall Dependent Infiltration Module (RDII) model. The hydraulic dynamic pipe flow computation is based on an implicit finite difference method of Saint Venant continuity and momentum equations.

2.4 Model Calibration

In this paper, hydraulic model and RNNs performance were evaluated by the coefficient of determination (R^2) and Nash-Sutcliffe Efficiency (NSE). NSE is a parameter that determines the relative importance of residual variance (noise) compared to the variance in the measured data (information). The NSE is calculated by the following equation:

$$NSE = 1-\left[\frac{\sum_{i=1}^{n}\left(Y_i^{obs}-Y_i^{sim}\right)^2}{\sum_{i=1}^{n}\left(Y_i^{obs}-Y^{mean}\right)^2}\right]$$

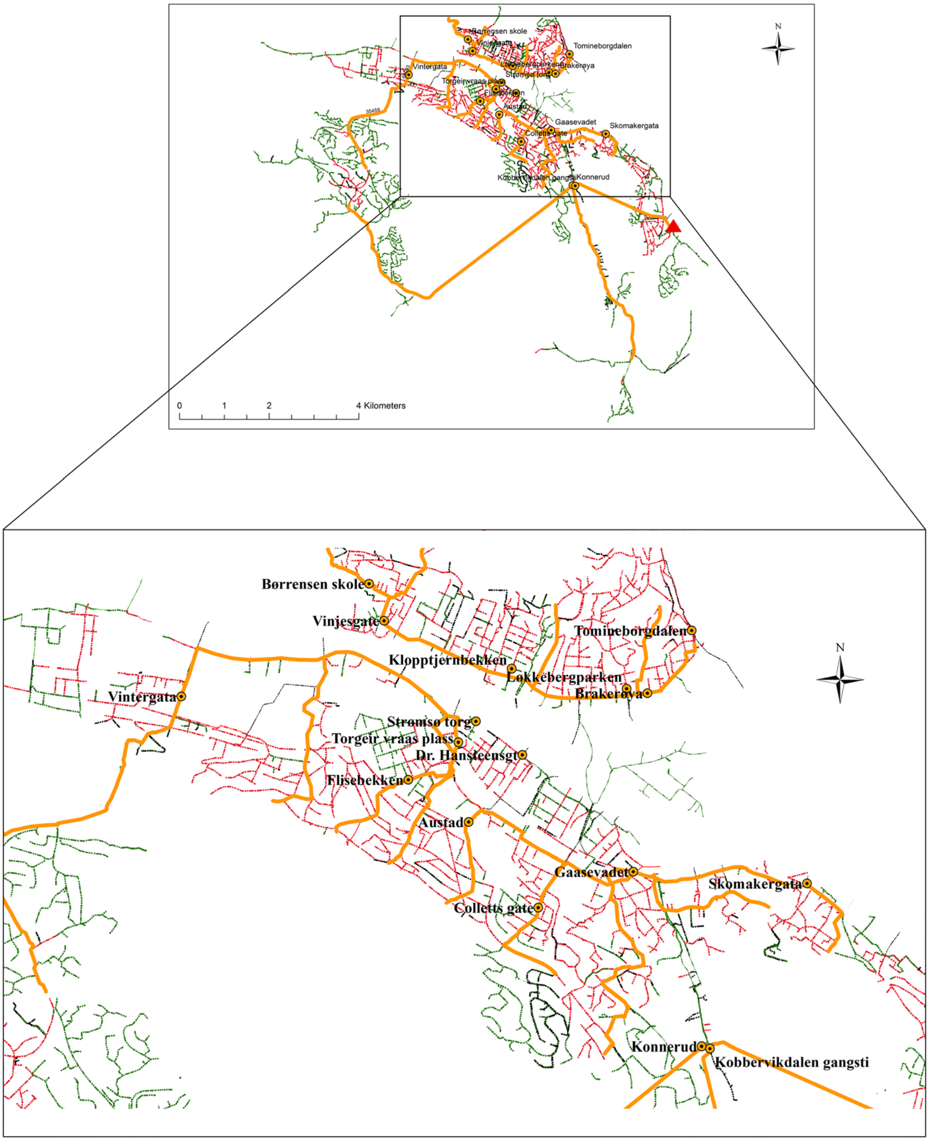

Fig. 2 Large pipes and monitoring sites of the Drammen sewer system

Where:

Y_i^{obs} the *i*-th observed data.
Y_i^{sim} the *i*-th simulated data.
Y^{mean} mean value of observed data.
n number of observed data

NSE varies from -∞ to 1, NSE = 1 indicates a perfect correlation between simulated and observed data, values between 0.0 and 1.0 is generally acceptable. In the present research, the criteria for successful calibration and validation of the hydraulic model is both NSE and R^2 should over 0.5.

Table 1 Name of monitoring sites

	Name of monitoring sites
1	Børresen skole
2	Vinjesgate
3	Klopptjernbekken
4	Løkkebergparken
5	Brakerøya
6	Tomineborgdalen
7	strømsø torg
8	Torgeir vraas plass
9	Flisebekken
10	Dr.Hansteensgt
11	Austad
12	Colletts gate
13	Vintergata
14	Gaasevadet
15	Kobbervikdalen gangsti
16	Konnerud
17	Skomakergata

2.5 Recurrent Neural Networks

2.5.1 Elman Neural Network

The Elman Neural Network (Elman 1990.) is an RNN with internal time-delay feedback connections in the hidden layer. It is a three layer (input layer, hidden layer and output layer) neural network. The input neurons are connected to the hidden neurons, and hidden neurons link to the output layer. In the hidden layer, a time-delay

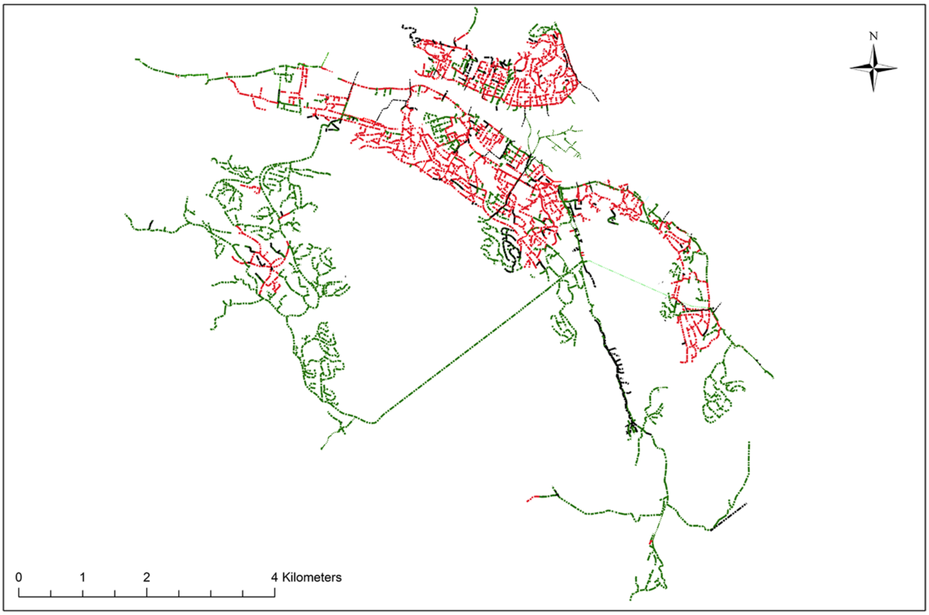

Fig. 3 Hydraulic model for the Drammen sewer system

unit is implemented, which stores the information of the previous set of hidden unit activations, and feeds back into the network as an additional input to all hidden neurons at the next time-step (Ishak et al. 2003). This enables the network inherent memory.

Training Elman neural network can be especially challenging due to the difficulty of learning long-range dependencies. Parameters from input neurons to hidden neurons, from hidden neurons to output neurons and between hidden neurons called weights. When training the Elman neural network, the ultimate goal is to calculate the gradients of the error corresponding to weights. Gradients sum up at each time step for one training example. To calculate these gradients we use the chain rule of differentiation. The problems of vanishing and exploding gradients occur when back-propagating errors across many time steps. Modern dynamic RNN architectures combat vanishing and exploding gradients had been proposed in recent years (Lipton et al. 2015; Gers 2001; Gers et al. 2000).

2.5.2 NARX

The NARX network is a kind of typical dynamic RNN. The N step ahead NARX can be represented by the following equation:

$$output(t+1) = f\left[\begin{array}{c} output(t-1), output(t-2), \ldots, output(t+1-q); \\ input(t-k), input(t-k-1), \ldots, input(t-k-p+1) \end{array}\right]$$

Where input (t) and output(t) is the input and output value at the time step t respectively, the parameters p and q are the time delay lag, $p \geq 1$ and $q \geq 1$, $p \leq q$. The process dead-time parameter k (k⩾0) is a delay term (Menezes and Barreto 2008). f [] is the nonlinear function. Inputs from output (t-1) to output (t + 1-q) function as an autoregressive model, input (t-k) to input (t-k-p + 1) plays the role of an exogenous variable.

There are two types of NARX training methods, the Series Parallel (SP) training method and the Parallel (P) method. The SP method can be mathematically represented by the following equation:

$$output(t+1) = f\left[\begin{array}{c} actual(t), actual(t-1), \ldots, actual(t+1-q); \\ input(t-k), input(t-k-1), \ldots, input(t-k-p+1) \end{array}\right]$$

In the SP method, regressor of the output in the input layer only use the actual value. When performance multi-step ahead predictions, the actual(t), actual(t-1), …, actual(t + 1-q) values are the future value that cannot acquire at the current time step. If the calculated outputs are feedback to the network's input layer as output's regressor, we can this mode as P method:

$$output(t+1) = f\left[\begin{array}{c} \widehat{actual}(t), \widehat{actual(t}-1), \ldots, actual\widehat{(t+}1-q); \\ input(t-k), input(t-k-1), \ldots, input(t-k-p+1) \end{array}\right]$$

Where the symbol (∧) is used to denote estimated values.

2.5.3 LSTM

The Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) make the RNN out of little modules that are designed to remember values for a long time. It is the most successful RNN architectures for sequence learning. The LSTM consist of input layer, recurrent hidden layer and output layer. Different from other RNNs, the LSTM embedded with a memory cell using logistic and linear units with multiplicative interactions. Information gets into, stays in the cell or read from the cell if the corresponding "write", "keep" and "read" gate is on. So that the LSTM is able to learn the time series with long time spans (Ma et al. 2015). With these memory cells, networks are also able to overcome vanishing and exploding gradients problems encountered by earlier recurrent networks.

The principal of the memory cell in LSTM can be mathematically represented by the following equations:

Input gate:

$$i_t = \sigma_g(W_i * x_t + U_i * h_{t-1} + V_i{}^\circ c_{t-1} + b_i)$$

Forget gate:

$$f_t = \sigma_g\left(W_f * x_t + U_f * h_{t-1} + V_f{}^\circ c_{t-1} + b_f\right)$$

Output gate:

$$o_t = \sigma_g(W_o * x_t + U_o * h_{t-1} + V_o{}^\circ c_{t-1} + b_o)$$

Cell state:

$$c_t = f_t{}^\circ c_{t-1} + i_t{}^\circ \sigma_c(W_c * x_t + U_c * h_{t-1} + b_c)$$

Output vector:

$$h_t = o_t{}^\circ \sigma_h(c_t)$$

Where x_t is the input vector. W, U, V and b are parameters for weights and bias. $^\circ$ represents the scalar product of two vectors, $\sigma(.)$ is the logistics sigmoid function.

2.5.4 Model Implementation

In this paper, the Elman and NARX were implemented using the Neural Network Toolbox of Matlab, R2016a. To modify the neural network training, the script generated by the Neural Network Toolbox was exported and customized using command line function. The LSTM was implemented using Keras. Keras is a high-level deep learning library supports recurrent networks. It is written in Python and running on top of either TensorFlow or Theano. TensorFlow backend is employed in this paper. TensorFlow is an open-source software for deep learning, released by Google in 2015. In this study, the Keras LSTM code was adopted after several open source codes. Interesting readers can find the major part of the code in Brownlee (2017) and Schmidt (2016).

The development environment for LSTM was set up using Docker Toolbox. With Docker, developers can download an image file that contains required packages and tools. The image file used in this study was pulled from docker hub: https://hub.docker.com/r/cannin/jupyter-keras-tensorflow-tools/. The source repository for this docker hub can be found in: https://github.com/windj007/docker-jupyter-keras-tools. After set up the development environment in Windows system, the Keras LSTM code was further modified in Jupyter notebooks (formerly IPython).

3 Results and Discussion

3.1 Model Calibration

Figure 4 and Table 2 is the calibration results from four of the monitoring sites. The observed data were retrieved from the regnbyge.no platform. The calibration results listed in Fig. 4 includes sites from north part, downtown and south part of Drammen, covers both dry weather season and wet weather season. The simulated curve in Fig. 4 fitting measured value very well, the NSE and R^2 in Table 2 also indicate a good calibration result. The calibrated model was then used as the baseline in the following scenario analyses.

3.2 Scenario Simulation

In order to find spatially distributed free space of the sewer system, the performance of sewer system was evaluated under rainfall events with different return periods. The rainfall scenarios were designed according to standard Intensity-Duration-Frequency (IDF) curve. Five scenarios under nowadays climate situation with a return period of 2, 5, 10, 20 and 50 years, and three scenarios consider the climate change effect, with intensity 1.5 times heavier than 2, 20 and 50 years return periods, named 2-plus, 20-plus and 50-plus scenarios, were simulated. Duration of all the rainfall events is 12 h.

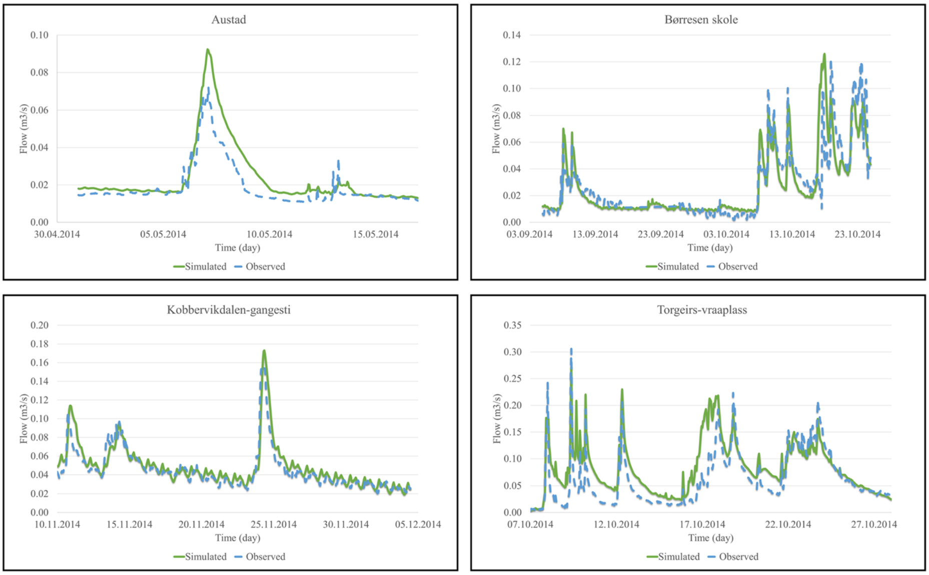

Fig. 4 Model calibration

Table 2 Model calibration result.

Measurement point:	NSE	R^2
Austad	0.54	0.91
Børresen skole	0.68	0.71
Kobbervikdalen-gangsti	0.80	0.85
Torgeir Vraa Plass	0.51	0.67

The distribution characteristics of the maximum filling degree during rainfall events under the different return periods were displayed in Fig. 5. From left up corner to right bottom corner of Fig. 5 are 2, 5, 10, 20, 50, 2-plus, 20-plus and 50-plus scenarios. As we can see from Fig. 5,

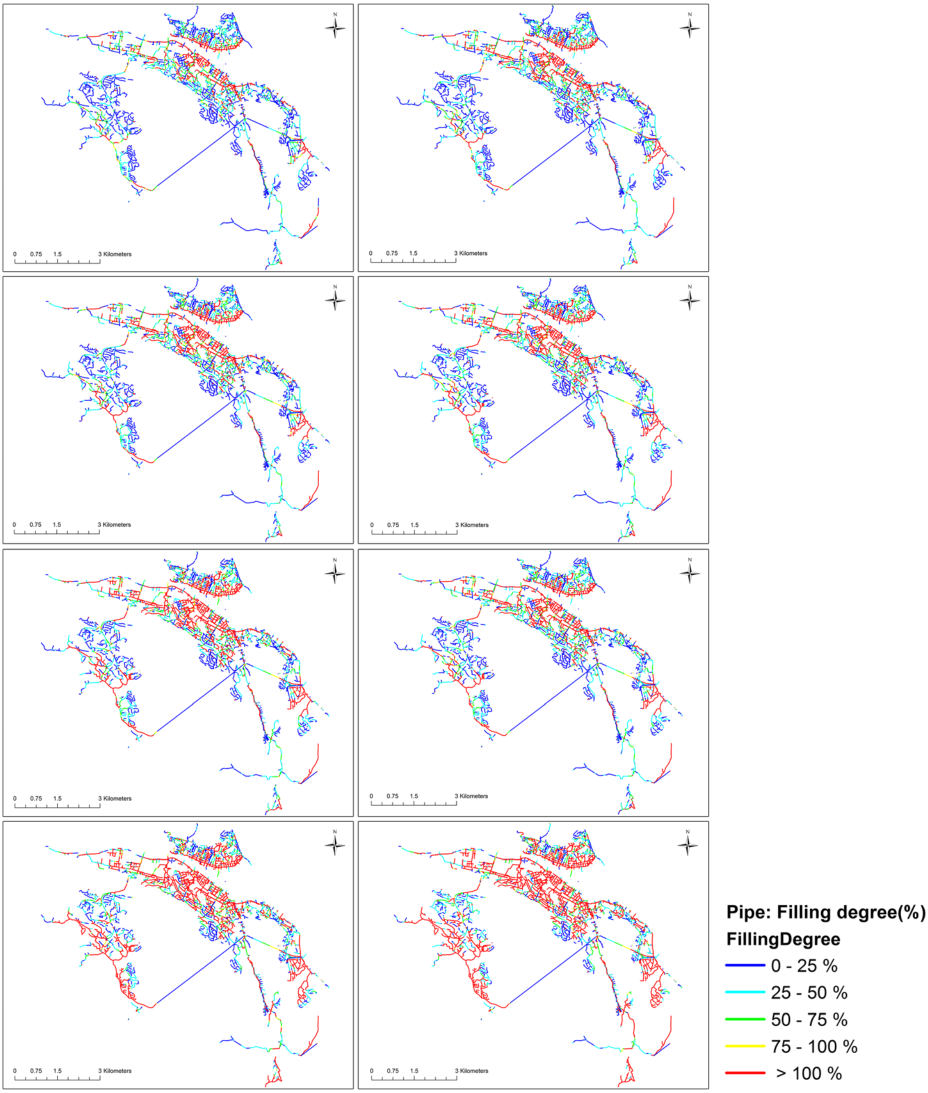

Fig. 5 Simulated filling degree under different return periods

increases in the rainfall return period corresponded to increases in the higher filling degree areas. Under 2 year and 5 year return periods, the pipes that distributed along the Drammen fjord was firstly influenced. With increased rainfall return period, under 10 year, 20 year and 50 year scenarios, pipes in the city center of Drammen reached maximum capacity. With extreme rainfall events (2-plus, 20-plus and 50-plus scenarios), we can observe that most parts of the Drammen sewer system were inundated, but still some part of the sewer have the left-over capacity.

Figure 6 is the Digital Elevation Map (DEM) of Drammen. The scenario simulation revealed the spatial variability of sewer system performance. It indicated that for some part of the sewer system, especially the traditional city center of Drammen, due to its lower elevation and combined sewer system, the flooding risk is very high even for lower return period. Nevertheless, for the southern part of Drammen with higher elevation and relatively new separate sewer system. Even under extreme rainfall events, it still has left over capacity. The sewer in-line control measures could be implemented in these parts of the sewer system.

Based on the scenarios simulations, two large pipes, namely the Konnerud tunnel and the Kobbervikdalen-gangsti tunnel, were selected as the control target. Figure 7 shows the location of the two pipes.

3.3 In-line Storage Control

As a common strategy, regulate the left over capacities of the sewer usually achieved by install control measures inside main pipes to the WWTP. The Solumstrand tunnel is the tunnel leading wastewater from large parts of Drammen finally to the Solumstrand WWTP. Currently, eight pumps and an overflow structure, which are located in the pipe immediately after the tunnel, are controlling the flow to the Solumstrand WWTP (Martinez 2016). The pumps were

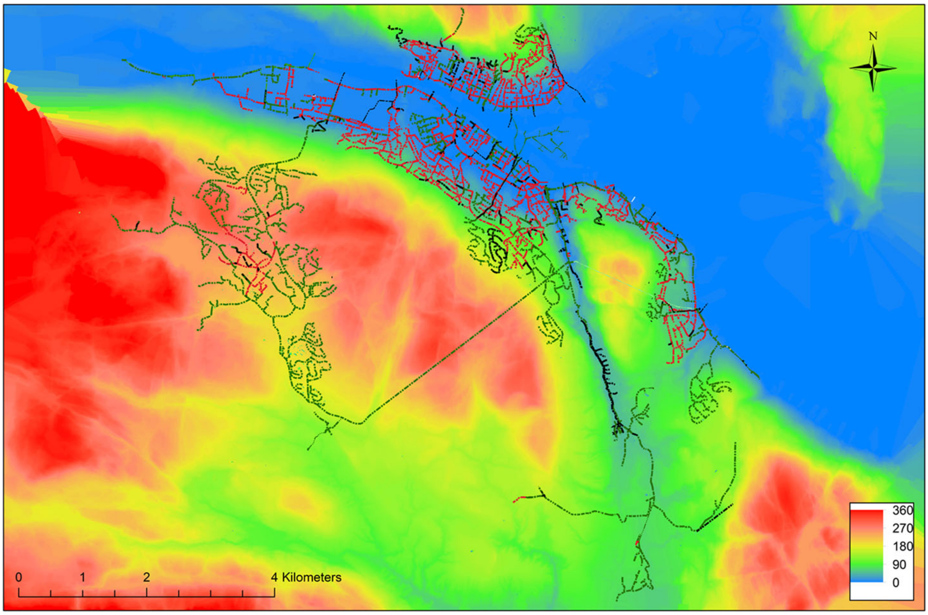

Fig. 6 DEM of Drammen

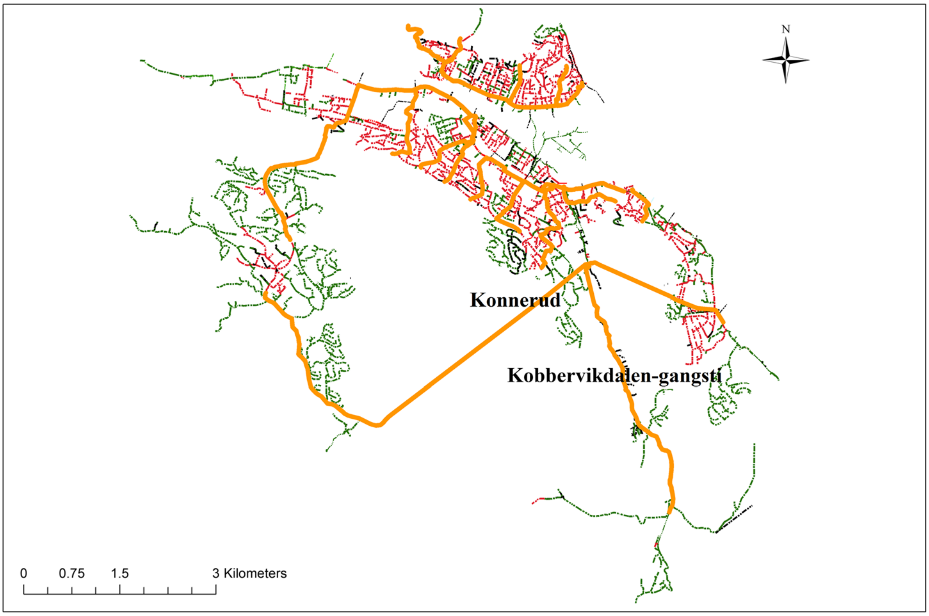

Fig. 7 Locations of Konnerud and Kobbervikdalen-gangsti tunnel

programmed to pump wastewater at different start-stop flow rates, through closing or opening the pumps, maximum flow to the WWTP while keeping the level below the plant bypass.

Similar to the controllable devices function in the RTC module of MOUSE DHI (DHI group 2014), the Rosie software have a "regulator" function to define user specify functional relations for control purpose. To test the effectiveness of the proposed in-line storage control, flow regulators were implemented in the hydraulic model for the Konnerud tunnel use the "regulator" function (Martinez 2016). The Konnerud tunnel has been confirmed have left over capacity from the scenario analysis. The regulators use a Q (flow)-H (head) relation to define the control logic. The purpose of control is, accumulate or release wastewater according to the free space of the large pipe, maximum its capacity but avoid overflow, thus retarding wastewater flow to the WWTP.

In order to compare the effects of without control, control only the Solumstrand tunnel (named scenario 1 hereafter), and control both the Solumstrand tunnel and Konnerud tunnel (named scenario 2 hereafter). Different control measures were simulated under current climate scenario with 2, 20 and 50 years return periods, and three rainfall events represent climate change scenario (2-plus, 20-plus and 50-plus scenarios).

Figure 8 shows the amount of overflow at the Solumstrand WWTP under different control scenarios and rainfall events. It shows that for the 2 year return period scenario, control the Solumstrand tunnel could reduce the overflow by up to 82%. However, with stronger rainfalls, only a small reduction was observed for scenario 1. This suggests that the Solumstrand tunnel alone cannot deal with heavier rainfalls. In scenario 2, we can see a dramatical reduction of overflow for all the return periods except the 2-year scenario, it's because the flow rate under 2-year return period simulation did not trigger the control action in Konnerud tunnel. For the rest of return periods, especially for extreme heavy rainfalls, scenario 2 led to an apparent

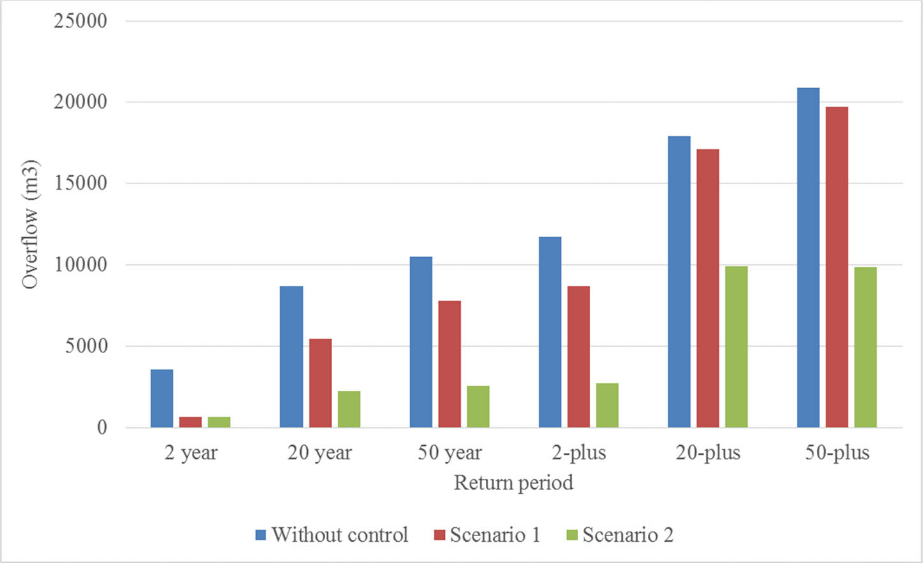

Fig. 8 Overflow at the Solumstrand WWTP under different control scenarios and return periods

reduction of overflow at the WWTP. Control rules also developed for Kobbervikdalen-gangsti tunnel with a similar procedure (Table 3).

3.4 RNNs

In the above parts of this paper, through simulations using a full detailed hydraulic model, we concluded two large pipes with left over capacity as potential control targets, which are Konnerud tunnel and Kobbervikdalen-gangsti tunnel. Then we concluded that the current implemented control strategy is insufficient to deal with extreme rainfall, but when further control the tunnel with left over capacity, we found that the overflow at the WWTP efficiently reduced. It testified the efficiency of the proposed in-line storage control strategy.

Unlike standard rainfall with designed return periods, the flow in the sewer in reality are varied in time following a stochastic dynamic pattern. To achieve successful in-line storage control, the control structures should accumulate or release wastewater timely. Such kind of timely control dependent on not only the present flow but also future flow. For example, if the sewer already full but the flow is coming, the control structures should discharge wastewater to downstream to prevent overflow. If the sewer still has left over capacity, operators can

Table 3 Overflow at the Solumstrand WWTP under different control scenarios and return periods

		Without control	Scenario 1	Scenario 2
Current climate	2 year	3586	642	642
	20 year	8713	5441	2285
	50 year	10,544	7793	2566
Climate change scenario	2-plus	11,751	8687	2720
	20-plus	17,896	17,132	9938
	50-plus	20,911	19,736	9888

confidently let the control structure accumulate wastewater. Detect suddenly change of flow is also important to keep safety operation. It is essential to construct a model that can forecast the flow. The forecasting model should be able to anticipate the future flow, enhance decision-making and give enough response time for control structures' operation.

In this context, the hydraulic models that require detailed knowledge of the drainage area, a large number of parameters and time consuming manually simulation, are inadequate for application in real time. In order to design an algorithm for real-time control, RNN were employed in this section. In this part of the paper, the performance of three types of RNN, namely Elman, NARX and LSTM, were compared. The objective of proposed RNNs is to predict the flow of a sewage stream 30 min ahead based on data measurements over the past 30 min.

In this study, the flow data and rainfall data for the RNNs were retrieved from the regnbyge.no platform. For the flow data of Konnerud tunnel and Kobbervikdalen-gangsti tunnel, the period with strongest flow fluctuation was used as training data for the neural networks. The most effective rain gauge was selected using XCORR (cross-correlation) function in Matlab (Mounce et al. 2014). The recorded flow and rainfall data have a 5 min interval. For the flow data from the Konnerud tunnel, there are a number of 9618 records with 98 missing or invalid data. For the Kobbervikdalen-gangsti tunnel, the data size and missing or invalid data are 9792 and 112 respectively. Missing and erroneous records were remedied using temporally adjacent records. Table 4 shows general statistics of the training datasets. Data were normalized to the range of 0-to-1 before training.

The Elman and NARX neural networks were implemented in the Neural Network Toolbox of Matlab. The Neural Network Toolbox divide the datasets into three subsets, training set, validation set and test set. For the Elman and NARX, 70%, 15% and 15% datasets were used as training set, validation set and testing set respectively. The LSTM was implemented in Keras, Keras has two modes for the datasets (Keras Documentation 2015): training and testing. 20% of the data was selected as the test and 80% to train. The difference between training and testing is regularization mechanisms, which is used as penalize to prevent overfitting, are turned off during the testing.

The training of RNNs was implemented through trial-and-error procedures. Different RNN architectures, i.e. number of hidden layers and hidden neurons in each layer, were tested. The suitable architecture for Elman and NARX was selected according to the performance of the validation set. Due to Keras only have training and testing mode, the optimal architecture for LSTM was chosen based on testing mode performance. The optimal structure of Elman has one hidden layer with ten hidden neurons. The selected NARX have one hidden layer with five neurons. The architecture of LSTM used to have two hidden layers with four LSTM cells in each layer. For regularization in an effort to limit overfitting and improve the model's

Table 4 General statistics of the training datasets

Tunnel name	Time span	Number of data	Max value (m^3/s)	Mean value (m^3/s)	Standard deviation (m^3/s)
Konnerud	28 March 2014–30 April 2014	9618	0.36	0.096	0.086
Kobbervikdalen-gangsti	16 October 2014–18 November 2014	9792	0.22	0.064	0.032

generalization. Large weights were penalized using L2 weight penalty method. The L2 weight penalty method adding an extra squared term to the cost function to constrain the weights. It could keep the weights small unless they have big error derivatives. Use L2 weight penalty method on the recurrent weights can help with exploding gradients.

There are many variations of training algorithm available in Matlab. These algorithms adjust the weights according to the derivation of the objective function, to reduce error. This procedure also called back propagation. Back propagation propagates the error between predicted and observed value backward to the hidden layer, then to the weights. In Matlab, the default algorithm for training is the Levenberg–Marquardt algorithm (‘trainlm’), it is the most commonly used training algorithm. In this study, different training algorithms were tested. The default Levenberg–Marquardt algorithm was selected for the training of Elman. The ‘traincgb’ as the best suited training algorithm was selected for NARX model.

In this work, the Stochastic Gradient Descent (SGD) with the best tradeoff between model performance and training speed was selected as optimization algorithm for the LSTM network. The SGD updates weights use the gradient on the first half dataset, then get the gradient for the new weights on the second half.

Parameters such as learning rate and momentum were tuned to further improve network performance. Learning rate controls how much to update the weight. The momentum, as the physical meaning of momentum, controls how much to let the previous update influence the current weight update. In Matlab and Keras, these fine tunings were done by “net.trainParam” and “keras.optimizers” function respectively.

Summarized results of the trained RNNs were presented in Table 5. For both Konnerud and Kobbervikdalen-gangsti tunnel, its possible to see that the three models perform comparatively well in the training stages. While in the testing stages, LSTM NN outperforms other models in terms of NSE. On the other hand, NARX neural network performs the second best in terms of NSE, and have the highest R^2 value for the Konnerud datasets. As modern dynamic RNNs, both NARX and LSTM got performance far beyond traditional RNN (Elman).

To illustrate the performance of different neural networks in a clearer way, for the Konnerud and the Kobbervikdalen-gangsti datasets, hydrographs of the observed and predicted flow corresponding to a complete rainfall event in the training stage is displayed in Fig. 9. To keep the drawing style in a uniform way, all the visualization was done in Matlab.

As we can see from Fig. 9. Elman has a tendency of overestimating low flow and underestimating peak flow. The memory of Elman rely on hidden neurons with predetermined time lags, it suffers from several issues due to the insufficient learning capability of past events, and thus may not be a suitable model for flow prediction. NARX outperform Elman since

Table 5 RNN results

Model stage	Model type	Konnerud		Kobbervikdalen-gangsti	
		R^2	NSE	R^2	NSE
Training	Elman	0.88	0.81	0.91	0.84
	NARX	0.92	0.89	0.90	0.88
	LSTM	0.94	0.91	0.95	0.87
Testing	Elman	0.67	0.54	0.71	0.60
	NARX	0.81	0.63	0.76	0.68
	LSTM	0.77	0.66	0.79	0.72

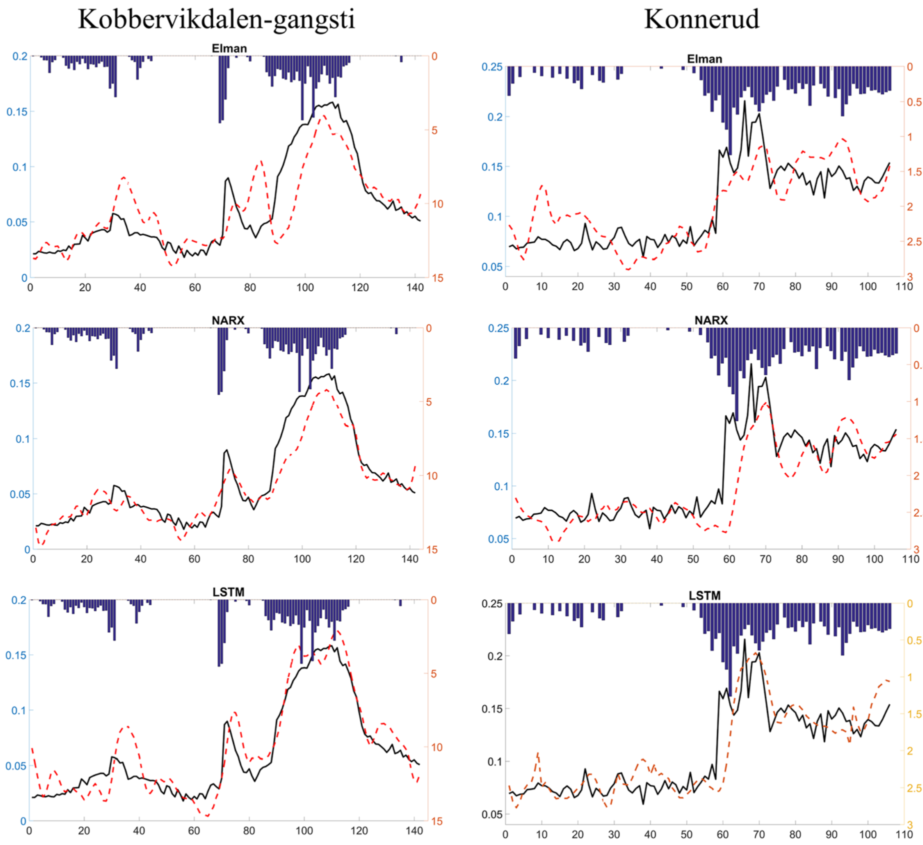

Fig. 9 RNN hydrograph

NARX can incorporate both its previous inputs and exogenous outputs. With the feedbacks of imperfect outputs, the NARX network can effectively make accuracy and reliability forecasts, while comparing to LSTM, the NARX seem have a time-lag phenomenon for the peak flow event. Enhance by the memory cells in the hidden layer. The LSTM can much effectively discover the long-term dependencies. As we can see in the hydrography, LSTM can better capture flow dynamic change of the flow, and mitigate the time-lag problem.

4 Conclusion

Combine the powerful side of both hydraulic model and RNN, an optimal in-line storage control strategy was designed for the city of Drammen, Norway. Several conclusions and perspectives of this study were summarized.

4.1 Hydraulic Model

In order to identify which part of the sewer system have left over capacity, and test the efficiency of proposed in-line storage control strategy, a full detailed hydraulic model was developed. The hydraulic model could give a clear insight into the sewer system. Through simulation based on rainfalls with various return periods, we found that the response behavior of sewer system is different with respect to location. Two large pipes from the sewer system, namely the Konnerud tunnel and Kobbervikdalen-gangsti tunnel, have higher free space even under extreme rainfall events. Subsequently, overflows at the WWTP under different control scenarios were compared. Simulated results showed that the current implemented control measures is insufficient to deal with overflow, when additionally control large pipes with in-line storage capacity, the overflow reduced dramatically. It testified the effectiveness of the proposed in-line storage control solution.

4.2 Recurrent Neural Network

In control system, the data collected by the sensors need to be analyzed to understand complex processes. For the present study, it is essential to establish a model that could forecast future flow, thus enabling better decision making. However, the computationally expensive hydraulic model is inadequate for the real-time forecast purpose. The recurrent neural networks were employed to undertake the real-time forecast task. The performance of three types of neural networks (Elman, NARX and LSTM) were compared. As state of the art technology, the LSTM got the best performance. Another dynamic neural network, NARX, also showed satisfying results. Moreover, the black box features of RNN makes it ideal for real-time forecast purpose.

4.3 Perspective

With deep learning gaining more and more attention in recent years, advanced artificial intelligence techniques such as the LSTM have shown their power. Nevertheless, the complexity of LSTM limited its application. Currently, most of the deep learning libraries are not Windows-friendly. The implementation and training of LSTM also require advanced mathematical knowledge and strong programming skill. On the contrast, with an easy to use Matlab toolbox, the NARX seems more suitable for practical engineering. However, we should notice that deep learning techniques such as LSTM are nowadays mainstream of artificial intelligence. Studies about the improvement of LSTM, e.g. the Gated Recurrent Unit (GRU) (Cho et al. 2014), will accelerate its spreading. We also expect user-friendly software or toolbox, which could easier programming work. Actually, in studies about time series prediction for traffic control, we can observe an inflection point that popular technologies are migrating from traditional RNNs to LSTM. Studies about adapt deep learning into water resource related fields is an interesting research direction in the future.

References

Alam F, Mehmood R, Katib I, Albeshri A (2016) Analysis of eight data mining algorithms for smarter internet of things (IoT). Procedia Comput Sci 98:437–442

Autixier L, Mailhot A, Bolduc S, Madoux-Humery AS, Galarneau M, Prévost M, Dorner S (2014) Evaluating rain gardens as a method to reduce the impact of sewer overflows in sources of drinking water. Sci Total Environ 499:238–247

Brownlee J (2017) Chapter 25: Time series prediction with LSTM recurrent neural networks. In Deep Learning with Python [e-book]. https://machinelearningmastery.com/deep-learning-with-python2/. Accessed 24 Mar 2017

Chang FJ, Chen PA, Liu CW, Liao VHC, Liao CM (2013) Regional estimation of groundwater arsenic concentrations through systematical dynamic-neural modeling. J Hydrol 499:265–274

Chang FJ, Chen PA, Lu YR, Huang E, Chang KY (2014a) Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. J Hydrol 517:836–846

Chang LC, Shen HY, Chang FJ (2014b) Regional flood inundation nowcast using hybrid SOM and dynamic neural networks. J Hydrol 519:476–489

Chang FJ, Tsai YH, Chen PA, Coynel A, Vachaud G (2015) Modeling water quality in an urban river using hydrological factors–Data driven approaches. J Environ Manage 151:87–96

Chang FJ, Chang LC, Huang CW, Kao IF (2016) Prediction of monthly regional groundwater levels through hybrid soft-computing techniques. J Hydrol 541:965–976

Chen J, Ganigué R, Liu Y, Yuan Z (2014) Real-time multistep prediction of sewer flow for online chemical dosing control. J Environ Eng 140(11):04014037

Chiang YM, Chang LC, Tsai MJ, Wang YF, Chang FJ (2010) Dynamic neural networks for real-time water level predictions of sewerage systems-covering gauged and ungauged sites. Hydrol Earth Syst Sci 14(7):1309–1319

Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

Darsono S, Labadie JW (2007) Neural-optimal control algorithm for real-time regulation of in-line storage in combined sewer systems. Environ Model Softw 22(9):1349–1361

DHI group (2014) https://www.mikepoweredbydhi.com/products/mike-urban Accessed 24 Mar 2017

Duchesne S, Mailhot A, Dequidt E, Villeneuve JP (2001) Mathematical modeling of sewers under surcharge for real time control of combined sewer overflows. Urban Water 3(4):241–252

El-Din AG, Smith DW (2002) A neural network model to predict the wastewater inflow incorporating rainfall events. Water Res 36(5):1115–1126

Elman JL (1990) Finding structure in time. Cogn Sci 14(2):179–211

Garofalo G, Giordano A, Piro P, Spezzano G, Vinci A (2017) A distributed real-time approach for mitigating CSO and flooding in urban drainage systems. J Netw Comput Appl 78:30–42

Gers F (2001) Long short-term memory in recurrent neural networks. Dissertation, Universität Hannover

Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: Continual prediction with LSTM. Neural Comput 12(10):2451–2471

Grum M, Thornberg D, Christensen ML, Shididi SA, Thirsing C (2011) Full-scale real time control demonstration project in Copenhagen’s largest urban drainage catchments. In: Proceedings of the 12th international conference on urban drainage, Porto Alegre.

Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7): 1527–1554

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

Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci 79(8):2554–2558

Ishak S, Kotha P, Alecsandru C (2003) Optimization of dynamic neural network performance for short-term traffic prediction. Transportation Research Record: J Transp Res Board 1836:45–56

Jordan MI (1997) Serial order: A parallel distributed processing approach. Adv Psychol 121:471–495

Keras Documentation (2015) https://keras.io/getting-started/faq/. Accessed 24 Mar 2017.

Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.

Liu Y, Ganigué R, Sharma K, Yuan Z (2016) Event-driven model predictive control of sewage pumping stations for sulfide mitigation in sewer networks. Water Res 98:376–383

Lucas WC, Sample DJ (2015) Reducing combined sewer overflows by using outlet controls for Green Stormwater Infrastructure: Case study in Richmond, Virginia. J Hydrol 520:473–488

Ma X, Tao Z, Wang Y, Yu H, Wang Y (2015) Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies 54: 187–197

Martinez N (2016) Analyse av. fordrøyningstiltak på eksisterende avløpsnett i Solumstrand rensedistrikt. Dissertation, Norwegian University of Life Sciences. In Norwegian.

Menezes JMP, Barreto GA (2008) Long-term time series prediction with the NARX network: An empirical evaluation. Neurocomputing 71(16):3335–3343

Mounce SR, Shepherd W, Sailor G, Shucksmith J, Saul AJ (2014) Predicting combined sewer overflows chamber depth using artificial neural networks with rainfall radar data. Water Sci Technol 69(6):1326–1333

Ødegård J, Persson M, Mathiesen TB (2013) Investeringsbehov i vann og avløpssektoren. Norsk Vann. In Norwegian.

Remesan R, Mathew J (2014) Hydrological data driven modelling: a case study approach. Springer, Switzerland, pp 71–110

Rossman, L. A. (2010). Storm water management model user's manual, version 5.0. Cincinnati: National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency.

Schmidt V (2016) Keras recurrent tutorial. https://github.com/Vict0rSch/deep_learning/tree/master/keras/recurrent. Accessed 24 Mar 2017.

Seggelke K, Rosenwinkel KH, Vanrolleghem PA, Krebs P (2005) Integrated operation of sewer system and WWTP by simulation-based control of the WWTP inflow. Water Sci Technol 52(5):195–203

Siegelmann HT, Horne BG, Giles CL (1997) Computational capabilities of recurrent NARX neural networks. IEEE Trans Syst Man Cybern Part B (Cybernetics) 27(2):208–215

Xingjian SHI, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems, pp. 802–810.

Yu Y, Kojima K, An K, Furumai H (2013) Cluster analysis for characterization of rainfalls and CSO behaviours in an urban drainage area of Tokyo. Water Sci Technol 68(3):544–551

Zaytar MA, Amrani CE (2016) Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. Int J Comput Appl 143:7–11