Norwegian University of Life Sciences
Faculty of Science and Technology

Philosophiae Doctor (PhD)
Thesis 2019:55

Sewer Overflow Control using Different Modeling Techniques and Internet of Things

Overløpskontroll i avløpsnett med forskjellige modelleringsteknikker og *internet of things*

Duo Zhang

Sewer Overflow Control using Different Modeling Techniques and Internet of Things

Overløpskontroll i avløpsnett med forskjellige modelleringsteknikker og *internet of things*

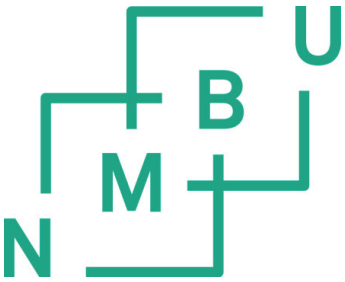
Philosophiae Doctor (PhD) Thesis

Duo Zhang

Norwegian University of Life Sciences

Faculty of Science and Technology

Ås (2019)



Thesis number 2019:55

ISSN 1894-6402

ISBN 978-82-575-1614-7

Tables of Contents

Acknowledgement

This work was carried out at the Norwegian University of Life Sciences. It was supported by the Regnbyge-3M project (grant number 234974), which was funded by the Oslofjord Regional Research Fund.

Many people contributed to the research underlying this thesis. First, I would like to offer my deepest gratitude to my supervisors, Prof. Harsha Ratnaweera, Mr. Geir Lindholm, and Prof. John Wyller. As my main supervisor, Prof. Harsha Ratnaweera provided valuable ideas and was very supportive throughout all phases of my research. I wish to thank him for his guidance, encouragement, and constructive criticism. I sincerely thank Mr. Geir Lindholm and everyone from his company, Rosim AS, for allowing the hydraulic model and providing the data used in this work. I am also grateful to Prof. John Wyller, whose knowledge of mathematics inspired me to explore the field of artificial intelligence.

Special thanks also go to DOSCON AS, Rosim AS, Ecomotive AS, and Silicon Labs AS for partially employing me during my research. Their employment not only funded me but also helped me gain practical work experience, which will be valuable for my long-term career.

I am also deeply grateful to all my colleagues at WESH group, the Faculty of Science and Technology, Norwegian University of Life Sciences, and the city of Drammen.

Finally, I would like to thank my family. To my lovely daughter, Linxi Zhang, my wife, Jingjing Li, and my mother, Chun Xu, thank you for your support!

Duo Zhang

March 19, 2019

List of Figures

List of Abbreviations

AI	Artificial intelligence
ANN	Artificial neural network
BP	Backpropagation
BPTT	Backpropagation through time
CNN	Convolution neural network
CSO	Combined sewer overflow
CSTR	Continuously stirred tank reactor
FFNN	Feedforward neural network
FRC	Fast response component
GRU	Gated recurrent unit
IBWT	Inter-basin water transfer
ICWT	Inter-catchment wastewater transfer
IoT	Internet of Things
LSTM	Long short-term memory
MLP	Multilayer perceptron
MOUSE	Model for Urban Sewers
MPC	Model predictive control
MTL	Multi-task learning
NARX	Nonlinear autoregressive network with exogenous inputs
PCA	Principal component analysis
PE	Population equivalents
PID	Proportional integral derivative
RL	Reinforcement learning
RNN	Recurrent neural network
SRC	Slow response component
SUDS	Sustainable urban drainage system
SVM	Support vector machine

SVR	Support vector regression
WNN	Wavelet neural network
WWTP	Wastewater treatment plant

Papers Included in this Thesis

Paper I

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. doi:10.1016/j.jhydrol.2017.11.018.

Paper II

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.* doi:10.1007/s11269-018-1919-3.

Paper III

Zhang, D., Hølland, E. S., Lindholm, G., & Ratnaweera, H. (2018). Hydraulic modeling and deep learning based flow forecasting for optimizing inter-catchment wastewater transfer. *Journal of Hydrology, 567, 792-802.* doi:10.1016/j.jhydrol.2017.11.029.

Paper IV

Zhang, D., Holland, E. S., Lindholm, G., & Ratnaweera, H. (2018). Enhancing operation of a sewage pumping station for inter catchment wastewater transfer by using deep learning and hydraulic model. *arXiv:1811.06367 [cs.CY]*. (*preprint in arXiv*)

Paper V

Zhang, D., Lindholm, G., & Ratnaweera, H. (2018). DeepCSO: Forecasting of combined sewer overflow at a citywide level using multi-task deep learning. *arXiv:1811.06368 [cs.CY].* (*preprint in arXiv*)

Other publications

The following papers represent various contributions that broadened my knowledge regarding aspects of sewer management and modeling. They were prepared during the Ph.D. study period, but they are not included in this thesis.

Zhang, D., Lindholm, G., Martinez, N., & Ratnaweera, H. (2018). Exploiting capacity of sewer system using unsupervised learning algorithms combined with dimensionality reduction. *arXiv* preprint arXiv:1811.03883.

Zhang D., Teimouri M., Ratnaweera H., Tveite H., Lindholm G., & Popov I. (2016). Web based GIS + GPS tracking application for sewer inspection robot. In *IWA Specialist Conference "Advances in particle science and separation: Meeting tomorrow's challenges."* 22–24 June, 2016. Oslo, Norway.

Zhang D., Ratnaweera H., & Martinsen M. (2016). Performance of wastewater treatment plants in cold climate of Norway. In *EWA Conference "Water management: Challenges in cold climate."* 25–27 June, 2016. Spitsbergen, Norway.

Zhang, D. & Ratnaweera, H. (2014). Optimising particle separation with CFD. In *IWA Specialist Conference "Advances in particle science and separation: from mm to nm scale and beyond."* June 15–18, 2014. Sapporo, Japan.

Abstract

Increased urbanization and extreme rainfall events are causing more frequent instances of sewer overflow, leading to the pollution of water resources and negative environmental, health, and fiscal impacts. At the same time, the treatment capacity of wastewater treatment plants is seriously affected.

The main aim of this Ph.D. thesis is to use the Internet of Things and various modeling techniques to investigate the use of real-time control on existing sewer systems to mitigate overflow. The role of the Internet of Things is to provide continuous monitoring and real-time control of sewer systems. Data collected by the Internet of Things are also useful for model development and calibration. Models are useful for various purposes in real-time control, and they can be distinguished as those suitable for simulation and those suitable for prediction. Models that are suitable for a simulation, which describes the important phenomena of a system in a deterministic way, are useful for developing and analyzing different control strategies. Meanwhile, models suitable for prediction are usually employed to predict future system states. They use measurement information about the system and must have a high computational speed.

To demonstrate how real-time control can be used to manage sewer systems, a case study was conducted for this thesis in Drammen, Norway. In this study, a hydraulic model was used as a model suitable for simulation to test the feasibility of different control strategies. Considering the recent advances in artificial intelligence and the large amount of data collected through the Internet of Things, the study also explored the possibility of using artificial intelligence as a model suitable for prediction.

A summary of the results of this work is presented through five papers. Paper I demonstrates that one mainstream artificial intelligence technique, long short-term memory, can precisely predict the time series data from the Internet of Things. Indeed, the Internet of Things and long short-term memory can be powerful tools for sewer system managers or engineers, who can take advantage of real-time data and predictions to improve decision-making.

In Paper II, a hydraulic model and artificial intelligence are used to investigate an optimal in-line storage control strategy that uses the temporal storage volumes in pipes to reduce overflow. Simulation results indicate that during heavy rainfall events, the response behavior of the sewer system differs with respect to location. Overflows at a wastewater treatment plant under different control scenarios were simulated and compared. The results from the hydraulic model show that overflows were reduced dramatically through the intentional control of pipes with in-line storage capacity. To determine available in-line storage capacity, recurrent neural networks were employed to predict the upcoming flow coming into the pipes that were to be controlled.

Paper III and Paper IV describe a novel inter-catchment wastewater transfer solution. The inter-catchment wastewater transfer method aims at redistributing spatially mismatched sewer flows by transferring wastewater from a wastewater treatment plant to its neighboring catchment. In Paper III, the hydraulic behaviors of the sewer system under different control scenarios are assessed using the hydraulic model. Based on the simulations, inter-catchment wastewater transfer could efficiently reduce total overflow from a sewer system and wastewater treatment plant. Artificial intelligence was used to predict inflow to the wastewater treatment plant to improve inter-catchment wastewater transfer functioning. The results from Paper IV indicate that inter-catchment wastewater transfer might result in an extra burden for a pump station. To enhance the operation of the pump station, long short-term memory was employed to provide multi-step-ahead water level predictions.

Paper V proposes a DeepCSO model based on large and high-resolution sensors and multi-task learning techniques. Experiments demonstrated that the multi-task approach is generally better than single-task approaches. Furthermore, the gated recurrent unit and long short-term memory-based multi-task learning models are especially suitable for capturing the temporal and spatial evolution of combined sewer overflow events and are superior to other methods. The DeepCSO model could help guide the real-time operation of sewer systems at a citywide level.

Sammendrag

Økt urbanisering og ekstreme nedbørsmengder skaper hyppigere forekomst av overløp i kloakkrør, noe som fører til forurensing av vann-ressurser, og skaper negative helse, miljø, og økonomiske konsekvenser. Samtidig blir kapasiteten til renseanleggene kraftig påvirket.

Hovedmålet med denne Ph.D. avhandlingen er å bruke *internet of things* og forskjellige modelleringsteknikker for å undersøke bruken av sanntidsovervåking på eksisterende kloakkrørsystemer for å redusere overløp. *Internet of things* oppgave er å kontinuerlig overvåke og sanntidskontrollere kloakksystemet. Dataene samlet inn fra *internet of things* er også nyttig i bruk til å utvikle modeller og kalibrering. Modellene er nyttige i varierende formål for sanntidskontroll, de kan brukes til simulering av hendelser, eller for å predikere hendelser. Modeller til bruk for simulering av hendelser, som beskriver de viktige fenomenene i et system på en deterministisk måte, er nyttige for å utvikle og analysere de forskjellige strategiene for kontroll. Modeller tilpasset for å predikere er vanligvis satt til å forutse fremtidige stadier av systemet, og må ha en høy beregnings hastighet og bruke målinger og informasjon gitt av systemet.

For å demonstrere hvordan sanntidskontroll kan brukes for å forbedre driften av kloakksystemet ble det gjennomført en casestudie for denne avhandlingen i Drammen, Norge. I denne studien ble det brukt en hydraulisk modell til simulering, for å teste gjennomføringsgraden av forskjellige kontrollstrategier. De nylige fremskrittene innenfor kunstig intelligens kombinert med det store antall data samlet via *internet of things*, viste i studien at det ville være mulig å ytterligere utforske mulighetene til å benytte kunstig intelligens som en modell for prediksjon.

Et sammendrag av resultatet fra dette arbeidet er presentert via fem artikler. Artikkel I demonstrerer at en mainstream kunstig intelligens teknikk, *long short-term memory*, presist kan forutse tidseriedata fra *internet of things*. *Internet of things* og *long short-term memory* kan være kraftfulle verktøy for operatører eller ingeniører av kloakksystemet, som kan dra fordeler av sanntidsdata og prediksjon for å forbedre

beslutningsevnen.

I artikkel II er en hydraulisk modell og kunstig intelligens benyttet for å detektere en optimal in-line lagerkontroll strategi som bruker det temporære lagervolumet i rør for å redusere overløp. Simulerings resultater fra den hydrauliske modellen som indikerte voldsomt regnfall viste at renseanleggets responsatferd varierer med hensyn til lokasjon. Overløp på et renseanlegg under forskjellige kontrollscenarier ble simulert og sammenlignet. Resultatene fra den hydrauliske modellen viser at overløp ble redusert dramatisk med forsettlig bevisst kontroll av rør med in-line lagringskapasitet. For å bestemme tilgjengelig lagringskapasitet på anlegget ble det benyttet recurrent neural networks for å forutsi den kommende strømmen som kommer inn i rørene som skulle kontrolleres.

Artikkel III og IV beskriver en ny løsning for avløpsvannoverføring. Metoden tar sikte på å omfordele blandet kloakkstrøm ved å overføre avløpsvann fra et anlegg til et nærliggende oppsamlingssted. I artikkel III, blir kloakksystemets hydrauliske atferd under forskjellige kontrollscenarier vurdert ved bruk av den hydrauliske modellen. Som antydet av de simuleringene, kan omfordeling av avløpsvann effektivt redusere det totale overløpet fra et kloakksystem og renseanlegg. Kunstig intelligens ble brukt til å forutsi tilsig til renseanlegget for å forbedre avløpsvannfordelingsfunksjonen. Resultatene fra artikkel IV indikerer at avløpsvannfordeling kan bli en byrde for en pumpestasjon. For å forbedre driften av pumpestasjonen ble det benyttet *long short-term memory*, som kunne gi predikasjoner om vannstand i flere trinn fremover.

Artikkel V foreslår en DeepCSO modell basert på basert på sensorer med høy oppløsning og *multi-task learning* teknikker. Eksperimenter demonstrerte at *multi-task learning* tilnærmingen generelt er bedre enn single-task tilnærmingen. Videre, er *gated recurrent unit* og *long short-term memory* baserte *multi-task learning* modeller spesielt egnet for å fange opp den tidsmessige og romlige utviklingen av kombinerte kloakkoverløpshendelser og er overlegne andre metoder. DeepCSO-modellen kan være til hjelp i sanntidsdriften av et kloakksystem på et bynivå.

1. Introduction

The sewer system is an important part of modern cities' infrastructure. The first sewer systems for modern cities were constructed more than 150 years ago. The initial purpose of sewer systems was to transport the wastewater and stormwater out of city centers to ensure public health (Ferriman, 2007). Today, the sewer systems have been connected to wastewater treatment plants (WWTPs) (Mollerup et al., 2015), polluted water flows to the WWTPs for treatment before being discharged to the environment, the focus of the sewer systems and WWTPs not only on public health but also on the environment and resource recovery (Vanrolleghem & Vaneeckhaute, 2014).

The sewer systems can be combined sewer systems, meaning that to transport wastewater and stormwater in the same pipes, or separate sewer systems that convey stormwater and wastewater in distinct pipes. European cities have a high proportion of combined sewer systems. In recent years, many cities have started to shift to separate sewer systems (De Toffol et al., 2007). Therefore, cities often have both types of sewer systems, separate sewer systems can be found in the newly constructed parts of the cities, whereas older parts of the cities are using combined sewer systems. Converting to separate sewer systems in urban centers often complicated due to the need to minimally disrupt the physical environments and daily operations as well as financial constraints.

A sewer system collects domestic sewage, industrial wastewater and/or rainwater runoff. During dry weather and minor rain events, it can transport all the wastewater it collects to a WWTP for treatment. However, heavy rainfall events result in damaging infrastructure with floods and sewer overflows due to the surcharge of the sewer systems. It also causes wastewater volume exceeds the inlet capacity of the WWTPs, leading to unavoidable bypass (Lund et al., 2018). In these cases, the sewer systems and WWTPs discharge untreated wastewater directly into water bodies or the surrounding environment, which can have severe consequences. Pollutants in sewer overflows, such as chemical and microbiological pollution, hazardous substances and other micropollutants, can decrease the water quality

of the receiving waters. Sewer overflows may carry bacteria, viruses, and other substances that can affect human health. Additionally, sewer overflows can have negative aesthetic impacts. Accordingly, sewer overflow has become a priority concern for many cities.

Extreme rainfall events are predicted to occur more frequently and with greater intensity because of climate change (Andersen et al., 2015), for sewer systems, the primary adverse effects are more frequent and severe sewer overflows. A study indicates that a 33%-83% increase in the volume of annual sewer overflows is expected in the future in Oslo, Norway (Nilsen et al., 2011). Another case study in Fredrikstad, Norway reveals that with increased rainfall, the volume of sewer overflows increased 1.5-3 times (Nie et al. 2009).

Besides extreme weather events, rapid urbanization is another important factor for the increased sewer overflows (Nie et al., 2009). Currently, more than 50% of the world's population is living in urban areas, and the number of people living in urban areas is still increasing. Urbanization often leads to more impervious surfaces because of constructing infrastructure to settle more people. The increased impervious surfaces reduce the possibility to infiltrate water and substantially increase the flow in the sewer system (Torgersen, 2017). Urbanization also affecting the water quality, for example, phosphorus (P) from highly populated areas is the main reason for eutrophication (Franco et al., 2018).

In Norway, the sewer systems are experiencing a general deterioration (Bruaset, 2019), thus infiltration and inflow from extraneous water and illicit connections are posing serious challenges for the sewer systems (Beheshti et al., 2015). Infiltration and inflow from unwanted sources, such as direct stormwater inflow, groundwater infiltration, and leakages from the drinking water network, etc., can overload the sewer system, cause sewer overflows, and decreasing overall treatment efficiency of the WWTPs (Beheshti & Sægrov, 2018). A study in Trondheim, Norway indicated that extraneous water via infiltration and inflow accounted for 46% of total water delivered to the WWTP during dry weather conditions (Beheshti & Sægrov, 2019).

Increased urbanization, extreme rainfall events, and deteriorated sewer pipes are inducing more frequent sewer overflows, leading to the pollution of streams, rivers, and fjords in Scandinavia. Most sewer systems in Norway were designed and built before 1980, and it did not anticipate the increase in rainfall intensities and urbanization that some cities have experienced (Torgersen et al., 2014). Additionally, the sewer systems in many Norwegian cities have a lag in rehabilitation, for example, despite the recommend annual rehabilitation rate of 1-2%, the actual rates have been below 0.7% during the last years (Bruaset, 2019). These factors are significant drivers for developing sewer system management strategies to avoid adverse effects of sewer overflows and to maintain the acceptable service level.

Additionally, with increased wastewater volume, some of the wastewater transported to the WWTP via the sewer system may be discharged via bypasses at the entry to the WWTP with partial or no treatment. In some cases in which the WWTP attempts to treat all or most incoming water, capacity limitations of the unit processes may result in reduced treatment efficiency. Meanwhile, pollution management requirements are becoming generally more stringent, further limiting the possibility of treat increased water inflows, while environmental authorities may include overflows in the catchment in the discharge permits, triggering a need to minimize sewer overflows.

1.1. Sewer system management strategies

There are several types of overflow mitigation measures. Structural measures (Lee et al., 2017) entail constructing new hydraulic facilities and increasing sewer network capacity either by physical expansion or by investing in separate sewers. This kind of solutions uses concrete and steel are referred to as gray infrastructure. These measures require major investments in the public waterworks infrastructure. In metropolitan cities, the locations and sizes of storage units are limited by the densely populated urban space, and the disruption to residences and businesses during construction also leads to social and economic costs (Ganora et al., 2017; Ngo et al., 2016).

In Norway, most sewer systems are combined sewer systems and located in central parts of cities (Torgersen et al., 2014). Structural measures, such as replacing current pipes with larger pipes, renewing

pipes to maintain necessary service levels, constructing separate sewer systems, or building extra storage spaces, are unrealistic due to the large investments required and space limitations.

The sustainable urban drainage system (SUDS) is an alternative to accommodate expected increases in sewer overflow. A SUDS aims to reduce the adverse influence of surface water in urban areas using non-piped solutions, such as ponds, open ditches, and green roofs (Torgersen et al., 2014). This kind of measures uses vegetative or sustainability-based practices are referred to as green infrastructure. However, it is unlikely that SUDS alone can meet the needs during extream events. On the other hand, the application of the SUDS could be challenging because of, for example, a lack of experience with SUDS operation and maintenance, a lack of knowledge regarding interaction with other water bodies, and institutional impediments and barriers to SUDS practices (Zhou, 2014).

In parallel to considering SUDS, it is necessary to look for nonstructural approaches (Lee et al., 2017). Recently proposed nonstructural measures primarily emphasize intelligent real-time surveillance and control (Grum et al., 2011), such as real-time control of WWTP inflows (Maxym & Sreekanth, 2015), gates installed in sewer pipelines (Carbone et al., 2014; Garofalo et al., 2017), pump stations (Ganora et al., 2017), and storage tanks (Lee et al., 2017). This type of solutions aims to utilize equalization volumes and available capacity in existing systems with minimal changes to the infrastructure by actively managing water streams.

1.2. The Internet of Things for real-time control

Internet of Things (IoT) technologies can be a useful tool for continuous monitoring and the subsequent real-time control of sewer systems. The role of the IoT in this context is to monitor process variables, such as the water level and flow rate, and to adjust actuators, such as pumps and gates. Additionally, the IoT does not only connect things but also generates enormous amounts of data from those things. In sewer system design and management, where the use of models is a common practice, such models require data about the sewer systems. The data collected by the IoT not only provide more information about the performance of the sewer systems but also aids in model development and calibration.

1.3. Simulation and prediction models for real-time control

The development of the IoT and the demand for better models and more intelligent control invite engineers to derive useful information from the data. With a focus on control, models can be distinguished as those suitable for prediction and those suitable for simulation (Breinholt et al., 2012), while modeling tasks can be separated into online and offline applications (Löwe et al., 2014).

Online applications are relevant for real-time warnings or control of sewer systems and usually require models to have a high computational speed and to use measurement information about the system (Löwe et al., 2014). Predicting future system states is an important part of online applications (Ahm, 2015).

Predictions can enhance the control of a sewer system in several ways. Prediction models can estimate combined sewer overflow (CSO) performance and generate early warnings for spills (Mounce et al., 2014). Predictions can result in sufficient lead-time to trigger control strategies, such as maximizing in-line storage and adjusting set points for moveable gates and pumping stations. Moreover, the prediction is a prerequisite of online optimization strategies that base optimal control settings on predictions and the effects of different actuator settings (Li et al., 2019; Löwe et al., 2014; Wei, 2013).

In contrast, offline applications typically focus on the design or analysis of a system, while models for offline applications should be able to use as much information about the system as possible. Models suitable for simulation are appropriate for offline applications, such as developing and analyzing different control strategies for an existing sewer system, which can describe the important phenomena of the system in a deterministic way (Breinholt et al., 2012).

1.4. Problem statement

There is a growing need for timely and efficient management of increasing wastewater flows in sewers. Despite the technical feasibility of relying on infrastructure expansion to manage increasing wastewater volumes, the financial and social implications of such initiatives are quite challenging in current urban contexts. Recent research indicates the potential value of focusing on nonstructural measures and, more

specifically, of real-time control of existing sewer systems to leverage the hidden capacities of existing water infrastructure.

Recent IoT advances in many areas have greatly improved real-time control by providing timely information required for monitoring and operations. However, whether and how real-time control of an existing sewer system adequately reduces overflow are questions that have not been sufficiently answered. Despite the anticipated potential of such solutions, it is quite challenging—both technically and financially—to evaluate such strategies in practice, as they result in many disturbances to the real system. Therefore, to test the feasibility of the control strategies, models suitable for simulation, such as hydraulic models, are required.

The new wealth of data collected by the IoT also poses many new questions, especially about how to leverage the large volume of data collected by the IoT. In this context, a data-driven modeling technique is a promising candidate due to its ability to extract useful information from data without explicit knowledge of the physical behavior of the system. Data-driven models are incapable of describing the full dynamics of the system being modeled, but they are useful for online applications because of the shorter simulation time and the ability to make predictions.

In recent years, data-driven modeling has been focused on machine learning, which consists of methods that determine the relationship between a system's inputs and/or outputs without human intervention. Deep learning, a paradigm of artificial neural networks (ANNs), is a state-of-the-art branch of machine learning and is also the dominant technology in today's artificial intelligence (AI) research. However, the potential of AI and data collected by the IoT in terms of coping with an important online application involving prediction has not been adequately studied. Whether it is possible to use AI to construct models suitable for prediction to improve real-time control of a sewer system, whether AI can outperform other data-driven modeling techniques, and how to maximize the benefits of AI are questions that still need answers.

Thus, the aim of this research is to provide insight into the use of real-time control when managing sewer

overflows and to test two hypotheses:

1. Proactive management strategies within an existing sewer system, when integrated with the IoT and various modeling techniques focused on real-time control of a sewer system, could use existing infrastructure to minimize sewer overflows and subsequently minimize environmental pollution, health threats, and other socio-economic impacts.
2. Based on data collected by the IoT, the use of AI as models suitable for prediction could enable better decision-making, greater automation, and higher accuracy and efficiency as regards the real-time control of a sewer system.

This thesis is organized as follows: First an introduction to the background and use of the various state-of-the-art tools and approaches, such as the IoT, sewer system models, machine learning, and deep learning, is provided. Then, the methods and materials chosen in this research are presented. Next, the summaries of the papers associated with this thesis are described along with the conclusions. Finally, perspectives on this thesis are discussed.

2. Background of potential techniques for sewer overflow control

2.1. Architecture of the Internet of Things

The IoT comprises technologies that enable "things"—such as sensors, wireless communication protocols, actuators, and web interfaces—to interact and cooperate with each other to achieve common goals. The IoT architecture consists mainly of a sensing layer, a network layer, and an application layer. The application layer, which is at the top of the architecture, enables users to access all the system's functionalities. The sensing layer, which uses sensors to track the status of things, is at the bottom of the IoT architecture. Between the sensing and application layers is the network layer, which transmits data recorded by the sensors through wireless communication protocols, bridging the physical and digital worlds (Atzori et al., 2010). Actuators are also required in many IoT applications to control things.

A few studies have explored improving urban issues using IoT. Kruger et al. (2015) designed a gateway device for communicating between a 6LoWPAN-based intelligent water meter and an IPv6 LAN network in a building. Chang et al. (2010) implemented a citywide IoT in Taipei, Taiwan. The sensor nodes were deployed all around Taipei, and a centralized storage center stored the acquired data. The sensor nodes collected data on humidity, temperature, wind speed, amount of rainfall, and so on. In terms of water and wastewater, there is also a substantial amount of literature available on sewage gas monitoring (Lim et al., 2013), water pipeline monitoring (Almazyad et al., 2014), and leaks and blockage monitoring (Stoianov et al., 2007).

2.2. The necessity of models for real-time control

Models are useful for various purposes in real-time control, and effective models can represent processes clearly and conceptually. Models are easier to work with than experimenting in the real world since engineers can gain insight and understanding of the system through model simulations without disturbing the operation of existing systems.

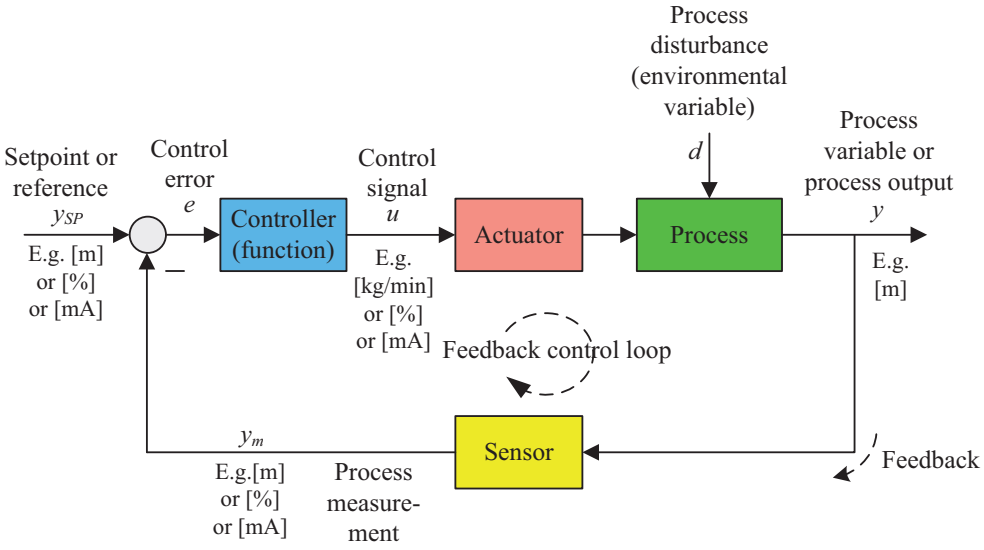


Figure 1. *Classic feedback control loops (source: http://techteach.no/)*

Figure 1 is a block diagram of the most common control method, feedback control. With this approach, a control error is first calculated according to the difference between the setpoint (the desired value) and the sensor-measured value. The control signal is then adjusted by the controller as a function of the control error, and, finally, the actuator tunes the processes according to the signal received from the controller.

Models are useful in the design and tuning of different control rules, such as the proportional integral derivative (PID) control. The PID control is an "error-driven" control, which means that the control is based on the control error. One problem with error-driven control is that the adjustment of the control variable always happens after the control error occurs. A model-based control method can adjust an error before it happens. A typical example of model-based control is model predictive control (MPC).

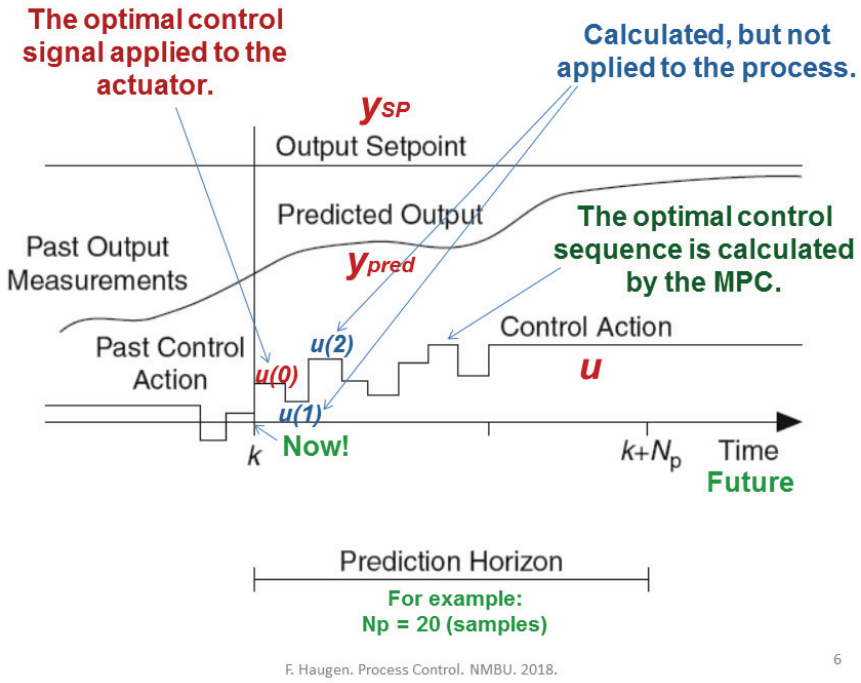


Figure 2. *Model predictive control (source: http://techteach.no/)*

Modeling the dynamics of the underlying physical system is the centerpiece of model-based control (Smarra et al., 2018). As shown in Figure 2, while conventional process control only takes the past and current situations into account, MPC uses a model to predict a system's future output based on predicted values and setpoints and uses an optimizer to find the optimal control signal sequence over the prediction horizon.

Various types of models can improve real-time control via almost all components of the control loop, from the controller (e.g., MPC) to the actuator, process, disturbance, sensor, and setpoint. A model can detect anomalies of the actuators or other assets. For complex processes, a model is often necessary for system identification or state estimation and for estimating the influence of the disturbances on the processes when implementing feedforward control. In terms of sensors, engineers can use soft sensing

models to measure parameters that are difficult to measure directly. Accurately predicting the setpoint using a model can enable early warning events, enhance decision-making, and give operators more response time.

2.3. Classifications of models in sewer system studies

From a mathematical point of view, a model is a set of equations that describe the evolution of a state variable in space and time (Popescu, 2014). Models are classified into two groups—dynamic or static—depending on whether the state variable is time dependent. If the state variable does not depend on the space variable, the model is finite. Finite dynamic models are represented using ordinary differential equations. Otherwise, the model is continuous, and continuous dynamic models are represented by partial differential equations.

In the field of sewer system studies, models can be described from an engineering point of view as falling into three categories: physically based distributed models, lumped conceptual models, and data-driven models.

Physically based distributed models are also called "mechanistic models" because they are based on engineering and scientific mechanisms of the process being modeled. A mechanistic model is usually deterministic because the principle of uncertainty is not considered (otherwise, the model would be a stochastic model). Physically based distributed models are "white-box models," meaning that all necessary information about the system is available, and all the processes are understandable and accounted for.

In many studies about sewer system control (Seggelke et al., 2005; Autixier et al., 2014; Lucas & Sample, 2015), physically based distributed models use Saint Venant equations—a set of partial differential equations that describe the conservation of mass and momentum—to simulate flow routing in sewer pipes. Physically based distributed sewer models require detailed information about a sewer system to describe the explicit hydraulic conditions in each part of the system (Compagni et al., 2019). The partial

differential equations of such models must be solved using advanced numerical methods with a high computational burden, such as the finite difference method, finite element method, and finite volume method. These characteristics make the model's construction, calibration, and computation extremely complex. Although this model provides a solid understanding of a sewer system's hydraulic behavior, its features limit its application for real-time purposes (El-Din & Smith, 2002).

In lumped parameter models or state-space models, the system's dynamic behavior is modeled without providing spatial details. Lumped conceptual models assume that the system properties are uniform over a given volume, which is represented by a continuously stirred tank reactor (CSTR). These kind of models are "gray-box models," meaning that not enough data is available and the model only partially understands the system processes. When using conceptual models for sewer system modeling, flow routing in a sewer system is described using a CSTR in series (Saagi et al., 2016), so the model is able to predict spatial variations. Lumped conceptual models have a faster computational speed than physically based distributed models and are typically used in long-term simulations or to resolve optimization problems.

Unlike physically based distributed models, data-driven models are derived directly from data rather than explicit knowledge of physical behavior. Data-driven models are "black-box models" because no prior knowledge and no understanding of the processes involved exist during the model-building stage. As with mechanistic models, data-driven models are also empirical models because they do not reflect the mechanisms of the system being modeled. When substituting for more complex physically based distributed models when shorter simulation times are required, data-driven models are often called "surrogate models" (Vanrolleghem et al., 2005).

2.4. Machine learning and deep learning

2.4.1. Machine learning

Today, machine learning is a mainstream AI technique. From a theoretical point of view, machine

learning algorithms are divided into two main groups: supervised and unsupervised learning. Supervised learning uses an algorithm to map the input to output to learn from data with both input variables and output variables. Unsupervised learning only requires input data.

There are two major categories of unsupervised learning algorithms: clustering and dimension reduction. Clustering algorithms, such as k-means (MacQueen, 1967), hierarchical cluster analysis (Murtagh, 1983), and self-organizing maps (Kohonen, 1990), identify hidden structures within input data and group similar objects into clusters (Mayer et al., 2014). Principal component analysis (PCA) is the most common dimension-reduction method. Through PCA, the original high-dimensional data are projected onto lower-dimensional vectors (principal components). The principal components are linear combinations of the original variables that determine the dominant patterns and major trends in the data (Gaitan et al., 2016; Wang et al., 2017).

There are two types of supervised learning algorithms: classification and regression. In a classification problem, the output data, or target, is categorized. The aim of classification is to identify which category the object belongs to based on the object's features (Huang et al., 2017; Jaramillo et al., 2018).

Regression algorithms should be used when using machine learning as models suitable for prediction. In a regression model, the outputs are continuous rather than categorized. In the urban water field, regression analysis can involve soft sensing (Wang et al., 2017) and time series prediction (Li et al., 2019).

Typical supervised learning methods include k-nearest neighbors, decision trees, random forest, and support vector machines (SVMs) (Cortes & Vapnik, 1995).

In the past, the SVM was the major competitor of the neural network family. Although ANNs currently dominate AI technology, they were previous almost unnoticed and overshadowed by SVMs. An SVM aims to find a hyperplane to divide into different categories with a gap that is as wide as possible. In the case of nonlinear problems, an SVM uses a kernel function to transfer nonlinearly separable data into a linearly separable space. Support vector regression (SVR) is a sub-category of SVM used for solving

regression problems. Despite being a type of linear model, SVR can solve nonlinear problems by using a kernel to transfer data into a feature space and then using a linear learning mechanism to learn a nonlinear function.

2.4.2. Artificial neural networks

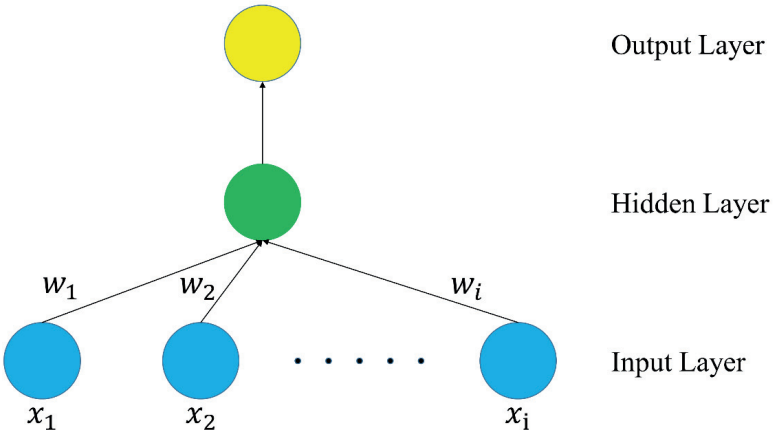


Figure 3. *Feedforward neural network architecture*

ANNs are the precursor of deep learning, and the feedforward neural network (FFNN) is one of the most popular forms. Usually, an FFNN is composed of an input layer, a hidden layer, and an output layer. There are multiple neurons in each layer, and different layers are connected by weights and biases. Figure 3 illustrates the architecture of an FFNN, with the circles denoting neurons and the lines between the circles denoting the weights. The FFNN first computes the weighted sums of the inputs, which are mathematically represented as follows:

$$s = \sum_{i=1}^{n} w_i x_i + b \qquad (1)$$

where w_i represents the weights, x_i is the inputs, and b is the bias. The computed weighted sum s is fed

into the neuron. The neuron consists of an activation function. There are various functions that can be used, the most classic of which is the sigmoid function, which is defined as follows:

$$f(s) = \frac{1}{1 + e^{-s}} \quad (2)$$

When training the FFNN, the goal is to minimize the cost function. The cost function can be defined as follows:

$$C = \frac{1}{2}(f(s) - f(s)_{observed})^2 \quad (3)$$

where C is the cost of the cost function, $f(s)$ is the predicted output from the neuron, and $f(s)_{observed}$ is the observed true value.

Backpropagation (BP) is one of the most commonly used training algorithms. It uses the chain rule of differentiation to calculate the partial derivative or gradient of the cost corresponding to the weights. For a single training example of a neuron, the gradient of cost C corresponding to a weight w_i is represented as follows:

$$\frac{\partial C}{\partial w_i} = \frac{\partial C}{\partial f(s)} \frac{\partial f(s)}{\partial s} \frac{\partial s}{\partial w_i} \quad (4)$$

The partial derivative of the cost function corresponding to the activation function is shown below:

$$\frac{\partial C}{\partial f(s)} = f(s) - f(s)_{observed} \quad (5)$$

The partial derivative of the sigmoid activation function corresponding to the weighted sum input of the neuron s is as follows:

$$\frac{\partial f(s)}{\partial s} = f(s)\,(1 - f(s)) \quad (6)$$

2.4.3. Recurrent neural networks

Currently, two of the most popular topics in the deep learning field are improving computer visioning

using convolution neural networks (CNNs) and improving modeling sequences using recurrent neural networks (RNNs). A major difference between an FFNN and the human brain is that an FFNN does not have a "memory." With connections between hidden neurons, an RNN (Elman, 1990) is more biologically plausible than an FFNN for modeling sequences. Because an RNN can process inputs using its internal memory, such a network is particularly applicable to tasks such as time series forecasting. By facilitating time delay units through feedback connections, RNNs can be trained to learn sequential or time-varying patterns and are thus more suitable for time series prediction than other neural networks and have attracted much attention in recent years (Chang et al., 2014).

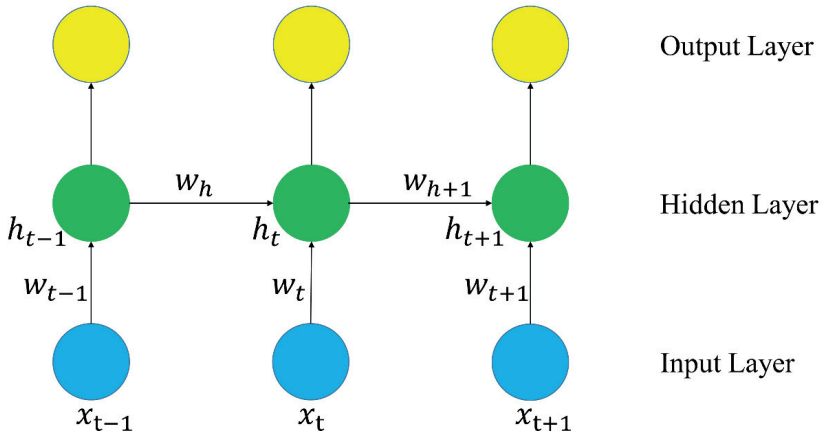


Figure 4. *Recurrent neural network (RNN) schematic*

As illustrated in Figure 4, in addition to the weighted sum of the input values, an RNN also takes the state of the hidden neuron at the previous time steps as an input for the next time step. In this way, the RNN passes messages to a successor. The neuron output of an RNN at time step t is calculated with the following equation:

$$h_t = f(w_h\, h_{t-1} + w_t\, x_t + b) \qquad (7)$$

where h_t is the state of the hidden neuron at time step t; h_{t-1} is the state of the hidden neuron at time step $t-1$; w_{t-1}, w_t, and w_{t+1} are weights between the input values and hidden neurons; w_h and w_{h+1} are weights between hidden neurons; and $f()$ is the activation function.

A variant of the RNN is the nonlinear autoregressive network with exogenous inputs (NARX). A NARX network is a type of dynamic ANN architecture that has exogenous inputs. The NARX can be represented by the following equation:

$$output(t+1) = f\begin{bmatrix} output(t-1), output(t-2), \dots, output(t+1-q); \\ input(t-k), input(t-k-1), \dots, input(t-k-p+1) \end{bmatrix} \qquad (8)$$

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

There are two types of NARX training methods: the series parallel training method and the parallel method. The series parallel method can be mathematically represented by the following equation:

$$output(t+1) = f\begin{bmatrix} actual(t), actual(t-1), \dots, actual(t+1-q); \\ input(t-k), input(t-k-1), \dots, input(t-k-p+1) \end{bmatrix} \quad (9)$$

In the series parallel method, the regressor of the output in the input layer can only use the actual value. When referring to performance multi-step ahead predictions, the *actual(t), actual(t − 1), …, actual(t + 1 − q)* values are the future values that cannot be acquired at the current time step. If the calculated outputs are fed back to the network's input layer as the output's regressor, one can set this mode as the parallel method:

$$output(t+1) = f\begin{bmatrix} \widehat{actual}(t), \widehat{actual}(t-1), \dots, \widehat{actual}(t+1-q); \\ input(t-k), input(t-k-1), \dots, input(t-k-p+1) \end{bmatrix} \quad (10)$$

where the symbol (∧) denotes the estimated values.

An RNN can be trained to use a variant of BP called "backpropagation through time" (BPTT). Unlike BP, BPTT calculates not only the gradient of the cost corresponding to the input weights but also the gradient of the cost corresponding to the hidden weights of the previous time steps.

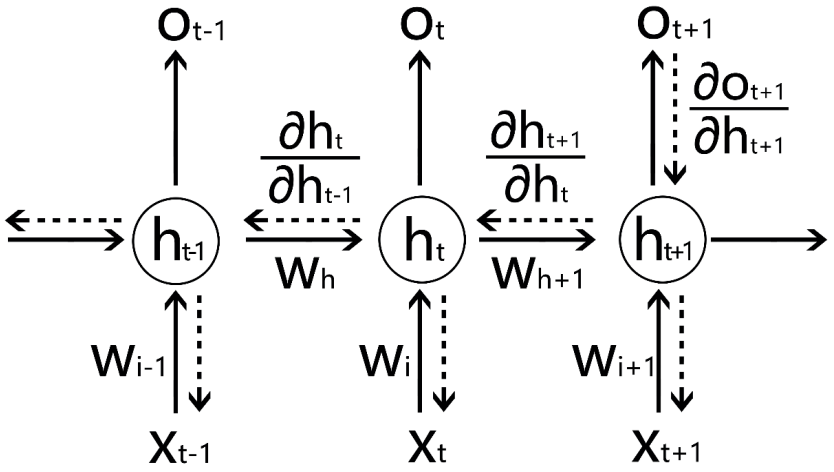


Figure 5. *The architecture of the RNN and schematic diagram of backpropagation through time (BPTT). The solid lines indicate how the RNN inherits previous hidden neuron states. The dashed lines indicate the direction of BPTT.*

In Figure 5, the dashed line with an arrow is the direction of the gradient calculation of BPTT. In backpropagation through time, the gradient of the output at time step *t + 1* ($\boldsymbol{O_{t+1}}$ in Figure 5) corresponding to the state of the hidden neuron at time step *t + 1* ($\boldsymbol{h_{t+1}}$ in Figure 5) is calculated first—in other words, $\frac{\partial O_{t+1}}{\partial h_{t+1}}$ in Figure 5. The gradient of the state of the hidden neuron at time step *t + 1* corresponding to the state of the hidden neuron at the previous time step ($\frac{\partial h_{t+1}}{\partial h_t}$ in Figure 5) is then calculated, and BP is performed on earlier neurons step by step in this way. When using the BPTT method, with gradient calculation, the error of a partial derivative accumulates through the time steps. Hence, it is extremely difficult to learn and tune the parameters of the earlier neurons and learn long-term dependencies because the gradient going through the network either becomes very small and vanishes or becomes very large and explodes. This problem is commonly known as the vanishing/exploding gradient problem.

2.4.4. Long short-term memory

Hochreiter and Schmidhuber (1997) developed a special RNN, known as long short-term memory (LSTM), to combat the vanishing/exploding gradient problem. As a pillar of deep learning, LSTM is the technology behind the latest versions of Google Translate. Unlike a traditional RNN, LSTM replaces ordinary hidden neurons with a series of memory blocks, each of which is composed of a memory cell and three gates.

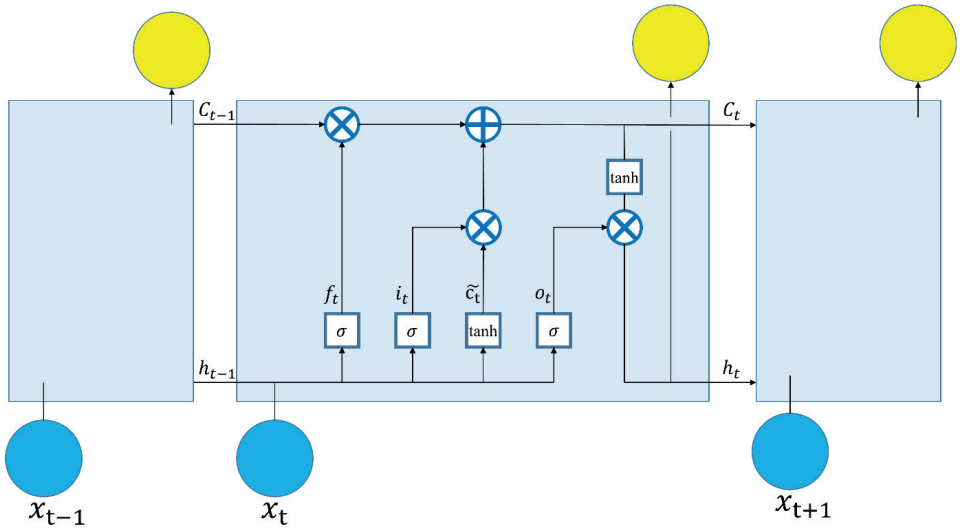


Figure 6. *Schematic of long short-term memory*

Figure 6 gives an example of an LSTM memory block, where x_t is the input vector, W and b are parameters for weights and bias, ° represents the scalar product of two vectors, σ_g is the sigmoid function, and σ_h is the hyperbolic tangent function (denoted as "tanh" in Figure 6). For a given input z, the output of the hyperbolic tangent function is as follows:

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \qquad (11)$$

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

The forget gate is used to reset memory blocks, thereby preventing the cell status from containing redundant information:

$$f_t = \sigma_g(W_f * [x_t , h_{t-1}] + b_f) \quad (12)$$

The forget gate receives the current input x_t and the state of the hidden neuron at the previous time steps h_{t-1} and then uses the weights for the forget gate W_f, the bias for the forget gate b_f, and the sigmoid activation function to generate the outputs, which are between 0 and 1.

The input gate is designed to permit inputs to modify the memory cell state:

$$i_t = \sigma_g(W_i * [x_t , h_{t-1}] + b_i) \quad (13)$$

The input gate receives the current input x_t and the state of the hidden neuron at the previous time steps h_{t-1}. It then uses the sigmoid activation function to generate outputs between 0 and 1.

Subsequently, the memory cell can impede outside interference and remain unchanged from one time step to another, thus allowing the LSTM to learn time series with long spans:

$$c_t = f_t \circ c_{t-1} + i_t \circ \overline{c_t} \quad (14)$$

$$\overline{c_t} = \sigma_h(W_c * [x_t , h_{t-1}] + b_c) \quad (15)$$

In equation (14), the cell state at the current time step, c_t, is determined according to the output values of the forget gate f_t and the input gate i_t. The forget gate determines how much information of the previous cell state c_{t-1} should be dropped (0 means drop all the information from c_{t-1}, and 1 means preserve all the information from c_{t-1}). $\overline{c_t}$ is the temporal cell state, which is calculated by a hyperbolic tangent function using current input $x_t$and the state of the hidden neuron at the previous time steps h_{t-1}. Similar to the forget gate, the input gate determines how the cell state should be updated according to the current information.

The output gate allows the cell state to affect other neurons or blocks it from doing so:

$$o_t = \sigma_g(W_o * [x_t, h_{t-1}] + b_o) \quad (16)$$

The output gate also uses the sigmoid function, the current input x_t, and the state of the hidden neuron at the previous time steps h_{t-1}.

The hidden state to be fed into the next time step is calculated according to the output gate and cell state transferred by a hyperbolic tangent function:

$$h_t = o_t \circ \sigma_h(c_t) \quad (17)$$

2.4.5. Gated recurrent unit

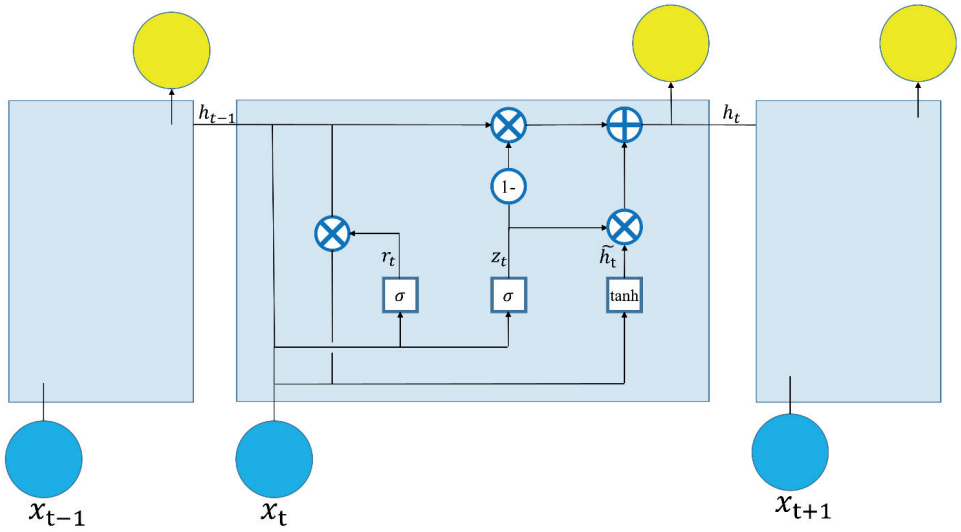


Figure 7. *Schematic of a gated recurrent unit*

The major drawback of LSTM is its complexity. The success of LSTM has resulted in frequent attempts to simplify it. The gated recurrent unit (GRU) is a recent advance in neural networks (Cho et al., 2014). As a variant of LSTM, the GRU also uses a gating mechanism to learn long-term dependencies, but its structure is greatly simplified compared to that of LSTM. Figure 6 shows the gating mechanism of a GRU.

The GRU is formulated as follows:

$$z_t = \sigma_g(W_z * [x_t, h_{t-1}] + b_z) \quad (18)$$

$$r_t = \sigma_g(W_r * [x_t, h_{t-1}] + b_r) \quad (19)$$

$$h_t = z_t \circ \bar{h_t} + (1 - z_t) \circ h_{t-1} \quad (20)$$

$$\bar{h_t} = \sigma_h(W_h * [x_t, (r_t \circ h_{t-1})] + b_h) \quad (21)$$

where x_t is the input vector, h_t is the output vector, z_t is the update gate vector, r_t is the reset gate vector, W and b are parameters for weights and bias, $\circ$ represents the scalar product of two vectors, $\sigma(\,)$ is the activation function, σ_g is the sigmoid activation function, and σ_h is the hyperbolic tangent activation function.

The GRU only has a reset gate and an update gate. Similar to LSTM, the reset gate r_t and update gate z_t of the GRU also receive the current input x_t and the state of the hidden neuron at the previous time steps h_{t-1} and then use the weights, bias, and sigmoid activation function to generate the outputs. Equations (18) and (19) are for the update gate and the reset gate, respectively. The GRU combines the input and forget gates into an update gate to balance between the previous activation and the candidate activation. In equation (20), the activation of *h* at time *t* depends on *h* at the previous time h_{t-1} and the candidate *h* ($\tilde{h}$ in Figure 7). The update gate *z* decides how much of the previous memory to preserve. The GRU unit forgets the previously computed state when the reset gate is off.

2.4.6. Multi-task learning

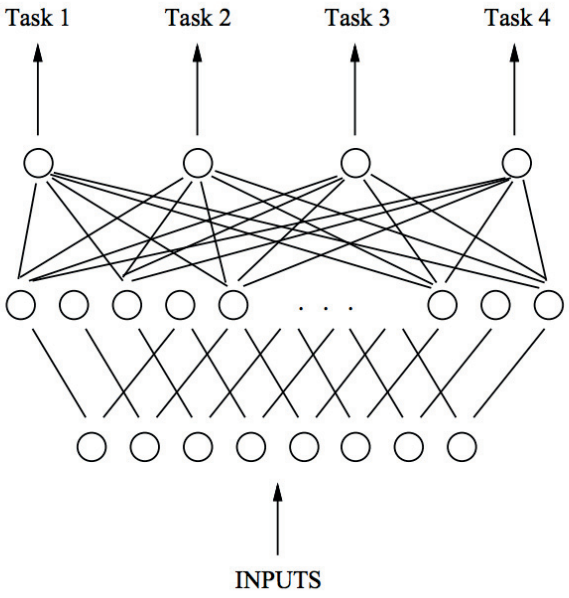


Figure 8. *Multi-task learning (source: Caruana, 1998)*

Multi-task learning (MTL) (Zhang, & Yang, 2017) aims at solving multiple tasks at the same time. If all tasks, or at least a subset of them, are assumed to be related to each other, the MTL approach can generalize better than a single-task model by sharing representations between the related tasks (Ruder, 2017). Deep learning is increasingly popular in MTL. Typically, this approach uses the first several hidden layers to learn common representations for multiple tasks and then to generate outputs for each task.

3. Methods and materials

3.1. Case study area

To demonstrate that sewer systems can be holistically managed through surveillance, modeling, and control based on the IoT, hydraulics models, and AI, a case study was conducted in Drammen, Norway.

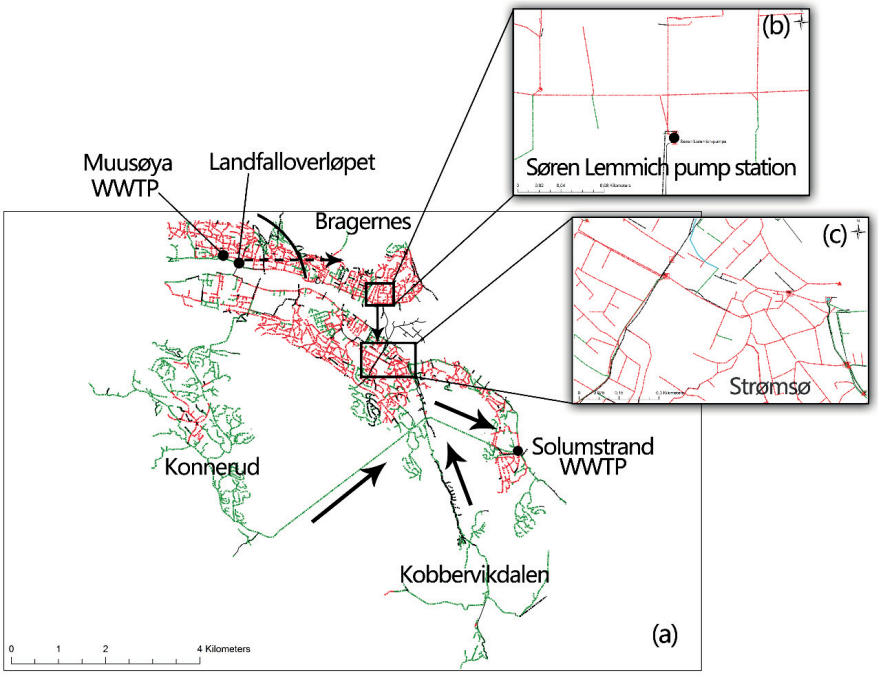


Figure 9. Overview of the sewer system in Drammen

The catchment area of Drammen's sewer system is around 15 km^2. The total length of the sewer system is approximately 500 km. Figure 9 presents an overview of Drammen. The Drammen Fjord flows through Drammen, and there are two catchments north of it, the Muusøya catchment and the Bragernes catchment. The curve in Figure 9a is the boundary of these two catchments. The Muusøya WWTP treats sewage collected in the Muusøya catchment. Wastewater from the Bragernes catchment is transported by the Søren Lemmich pump station (Figure 9b) from the north of the Drammen Fjord to the south (the Strømsø

catchment, Figure 9c). Subsequently, sewage from the Bragernes catchment and the Strømsø catchment merges with wastewater from the Konnerud and Kobbervikdalen catchments and then discharges to the Solumstrand WWTP.

The Muusøya WWTP has a designed treatment capacity of 33,000 population equivalents (PE), a dimensioning flow (Qdim) of 780 m^3/h, and a maximum flow (Qmax) of 1,200 m^3/h. The Solumstrand WWTP has a designed treatment capacity of 130,000 PE; the Qdim and Qmax for the Solumstrand WWTP are 2,000 m^3/h and 4,000 m^3/h, respectively. The combined sewers account for more than 80% and less than 50%, respectively, of the sewer system associated with the Muusøya WWTP and the Solumstrand WWTP. Moreover, the drainage area of the Muusøya WWTP has a higher population density than the rest of the Drammen city due to its location in the traditional city center. The lower WWTP capacity, the higher combined sewer percentage, and the denser population have resulted in a severe overflow problem in the Muusøya area.

3.2. Control strategies investigated

In line with the theme of the Regnbyge-3M project, two control strategies are proposed and investigated in this case study. The purpose of these strategies is to use real-time control on the existing sewer system to minimize overflow.

The first control strategy is in-line storage control. When rain showers occur, parts of the wastewater system experience overload, and other parts experience partially filled sewers. There is a potential in control pipes with excess capacity to maximum the sewer system's in-line storage capacity and prioritize different wastewater streams to minimize damage and inconvenience.

The second control strategy is inter-catchment water transfer (ICWT), which is inspired by inter-basin water transfer, involving the transfer of water from basins having sufficient water (donor basins) to basins facing water shortages (receiving basins). The ICWT method takes both the sewer systems and the WWTP into consideration. The WWTP receives sewage collected from the sewer systems of the

associated catchments. These sewer systems and catchments have different behaviors in response to rainfall events, and WWTPs also differ in treatment capabilities. A catchment with a more overflowed WWTP can be regarded as the donor basin. On the contrary, a catchment with a less overflowed WWTP can be regarded as the receiving basin. The essence of ICWT is to address the spatial mismatch between sewer flow and WWTP treatment capacities in different catchments.

3.3. Internet of Things: Regnbyge.no

In the Regnbyge-3M project, an IoT-based sewer surveillance web portal, Regnbyge.no, has been developed to monitor the water level and velocity of the main pipes of the sewer system as well as rainfall. Data collected by Regnbyge.no can be further used for calibration of the hydraulic model and the development of AI models.

The sensing layer of the Regnbyge.no consists of ultrasonic water level sensors and Doppler velocity sensors as well as rain gauges. The measurements collected by the sensors and rain gauges are transmitted using wireless telemetry to a remote server. A spatial database has been designed to manage the transmitted data, thereby simplifying the process for searching, editing, and sharing real-time information in a user-friendly way. Once data are stored in the database, information can be shared via the web and accessed through desktop or mobile devices. Furthermore, the data in the database are visualized using a web-based geographic information system.

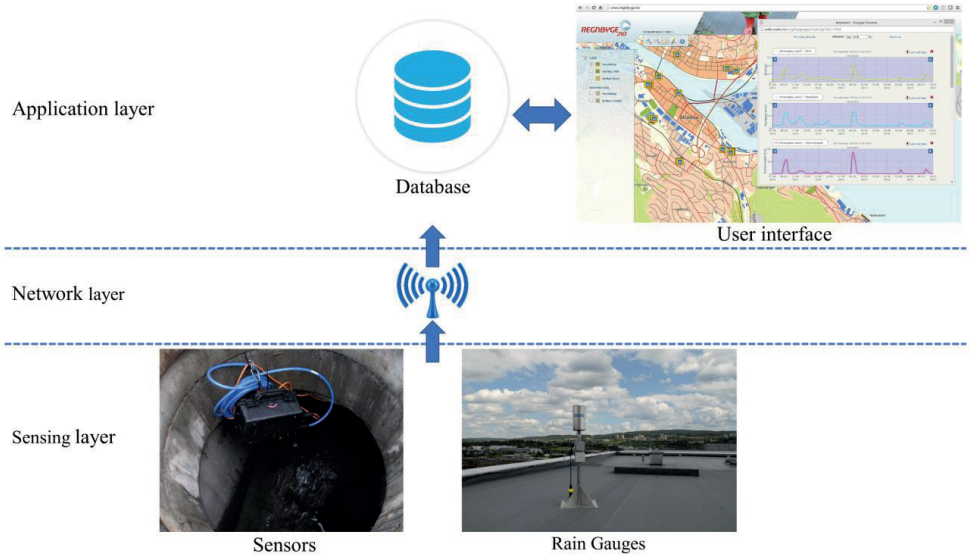


Figure 10. *Schematic of the Regnbyge.no*

Figure 10 displays the schematic of the Regnbyge.no.

3.4. Hydraulic model: Rosie

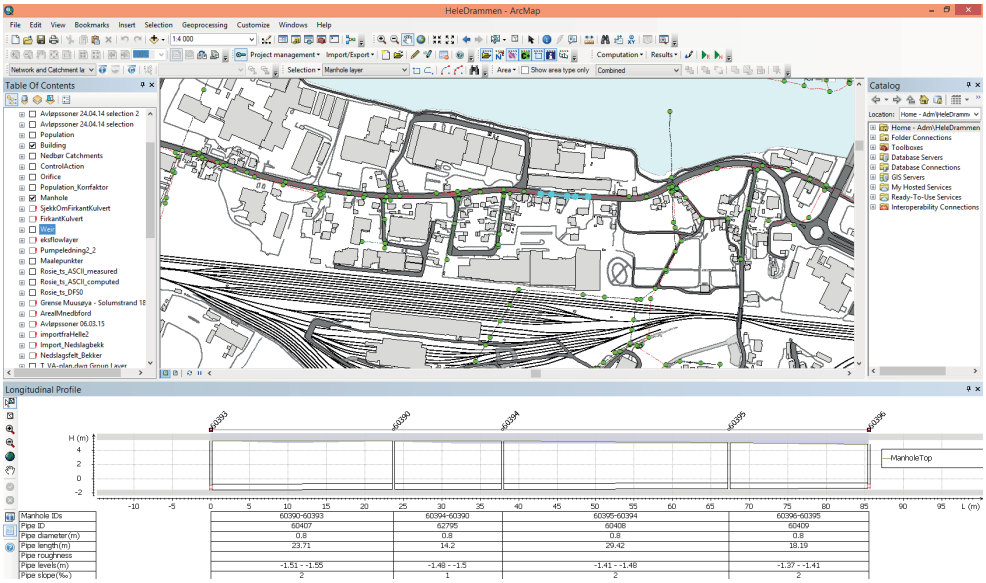


Figure 11. *Interface of the Rosie software*

To test the effectiveness of the proposed control strategies, a sewer system hydraulic model is used as a model suitable for simulation. The specific sewer hydraulic model used in this study is known as Rosie. It is an ArcGIS additional application for planning, sizing, and modeling water distribution and sewerage systems. The Rosie software maintains the interface and all functions of ArcGIS while using Model for Urban Sewers (MOUSE) as the computational engine to simulate the sewer system.

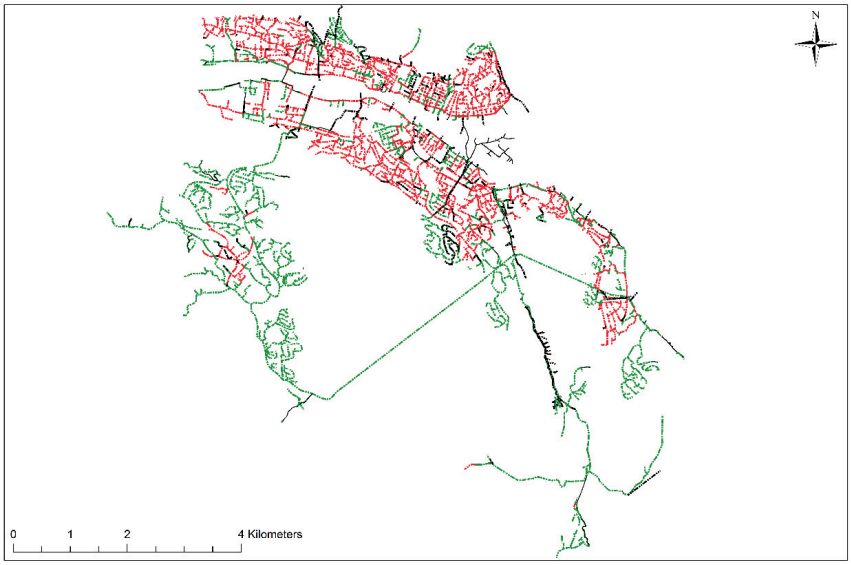


Figure 12. *Hydraulic model of the Drammen sewer system*

Figure 12 provides a fully detailed hydraulic model of the Drammen sewer system, which contains 9,113 pipes, 9,094 manholes, 129 weirs, 78 pumps, and 39 outlets. The runoff consists of two major components: the fast response component (FRC) and the slow response component (SRC). The FRC consists of the direct response to rainfall, which is calculated by the time–area curve method in Rosie. The SRC is the runoff generated cumulatively from the previous hydrological processes and accumulated as interflow and baseflow. The Rosie application uses the rainfall-dependent infiltration/inflow method to calculate the SRC. The pipe hydrodynamic flow is simulated by one-dimensional free surface Saint Venant continuity and momentum equations.

The MOUSE real-time control module is used to simulate sewer control. In this module, the operation of control structures, such as pumps, moveable weirs, or orifices, is designed by a curve that establishes the relationship between the water level and water flow.

3.5. Artificial intelligence development environment

In this study, the Jupyter Notebook was used as the programming interface, and the deep learning models were implemented using TensorFlow. TensorFlow is an open-source deep learning software tool released by Google in 2015. Other Python-based machine learning libraries, including pandas, NumPy, scikit-learn, and Matplotlib, were also used. Specifically, pandas and NumPy were used to load the dataset as the data frame and to prepare the raw data in the format of the desired array. Scikit-learn was used for model selection and preprocessing. Matplotlib was used for visualization.

4. Results and conclusions

A summary of the above-mentioned technologies and control strategies for the Drammen sewer system is presented in terms of the five papers included in this thesis.

4.1. Paper I: Proof of concept

LSTM is a pillar of deep learning but is seldom used in water resource-related fields. Paper I describes how data were collected from an IoT network that monitored a CSO structure to shed light on the performance of LSTM. Different neural networks and deep learning models were built to simulate and predict the water level of the CSO structure.

A comparison of a multilayer perceptron (MLP), wavelet neural networks (WNNs), and two deep learning methods (i.e., LSTM and GRU) demonstrated that only the MLP presented accurate one- and two-step-ahead predictions, while its multi-step-ahead predictions were not accurate. A WNN can improve multi-step-ahead predictions. However, regarding the hydrography of the observed and predicted water levels, the WNN presented several problems, including time delays and strong fluctuations. The LSTM and GRU approaches exhibited superior capabilities in multi-step-ahead time-series prediction. Furthermore, the GRU achieved prediction performances similar to LSTM with a quicker learning curve.

4.2. Paper II: In-line storage control

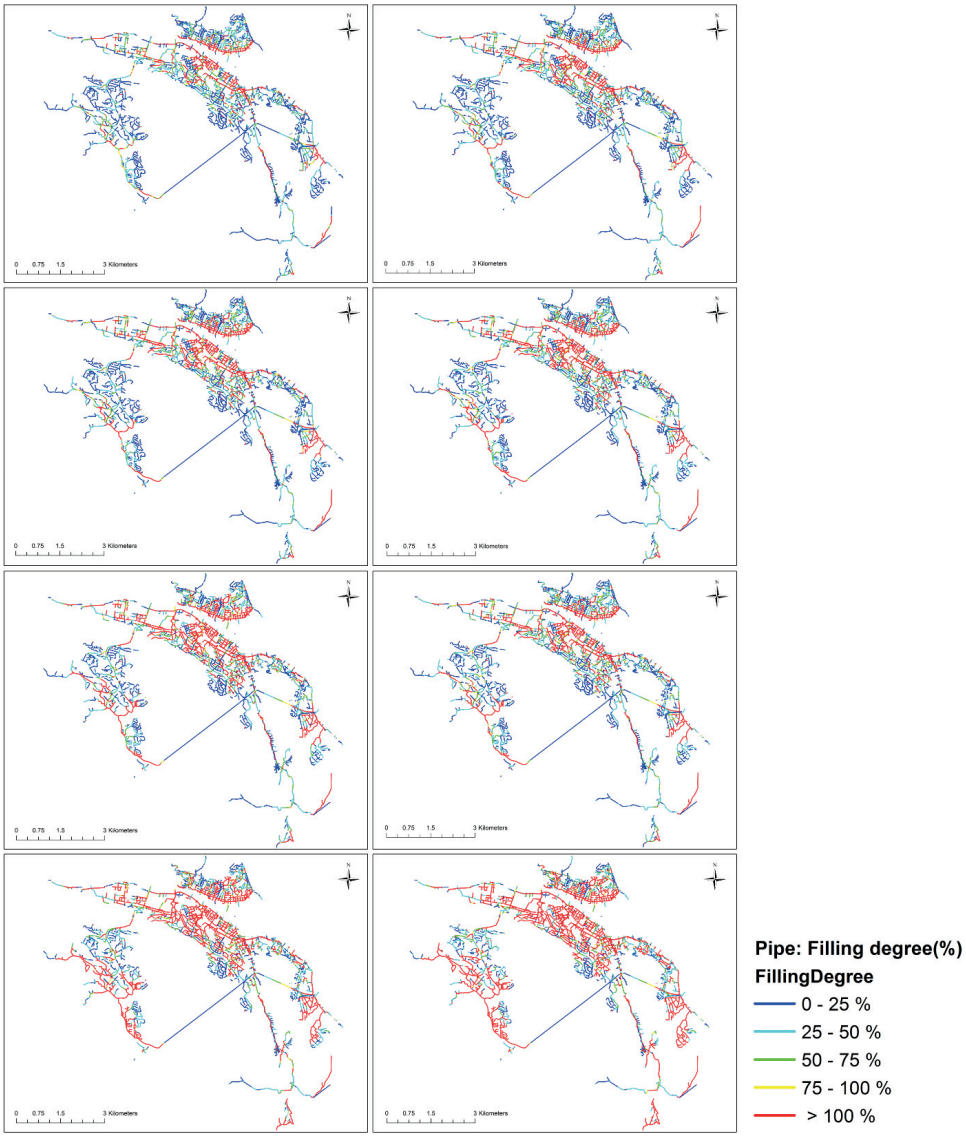


Figure 13. *Simulated filling degree during rainfall events for different return periods*

Paper II discusses the possibility of using sewer in-line storage control to reduce overflow at the WWTP. This paper combines the benefits of both the hydraulic model and RNN. As a model suitable for simulation, the hydraulic model could provide clear insight into the sewer system. Using this model to find spatially distributed free space in the sewer system, the paper evaluates the performance of the sewer system during rainfall events with different return periods. The rainfall scenarios were designed according to the standard intensity–duration–frequency curve. Five scenarios were simulated under the current climate situation with return periods of 2, 5, 10, 20, and 50 years, and three scenarios were simulated that considered climate change effects with intensities 1.5 times heavier than the 2-, 20-, and 50-year return periods, named the 2-plus, 20-plus, and 50-plus scenarios. The duration of each rainfall event was 12 h.

Through simulations based on rainfall events with various return periods, we found that the response behavior of the sewer system varied with respect to location (Figure 13). The in-line storage control strategy was applied to the part of the sewer that had the leftover capacity, and overflows at the WWTP were reduced dramatically. Finally, since in-line storage control depends on the water volume presently stored in the sewer pipes and the volume of the upcoming water, there is a particular need to develop real-time forecasting solutions that can describe the available capacity in the sewer system. To improve decision-making and provide sufficient response time for the proposed control solution, RNN was employed to forecast flow. Three RNN architectures, namely, Elman, NARX, and LSTM, were compared. The LSTM approach exhibited superior capabilities regarding time series prediction.

4.3. Papers III and IV: Inter-catchment wastewater transference

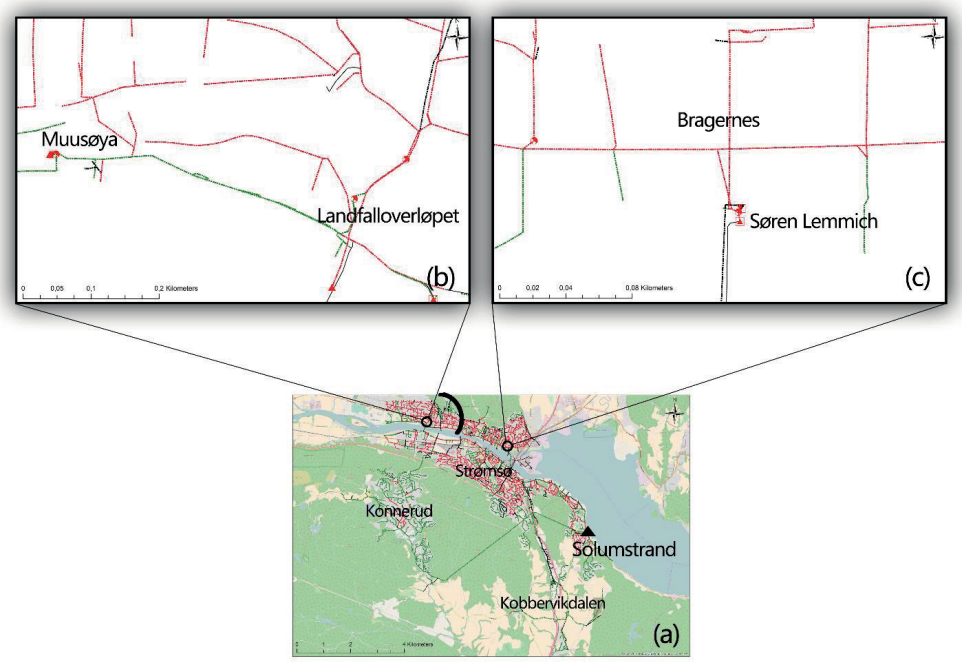


Figure 14. *Overview of catchments and wastewater treatment plants in Drammen*

The other control strategy, ICWT, was investigated in Paper III. In this study, the ICWT method was used to redistribute spatially mismatched sewer flows by transferring wastewater from the Muusøya WWTP to the Solumstrand WWTP. Three tasks were involved in the development of the ICWT system: (1) testing the effectiveness of ICWT, (2) studying how to properly redistribute inflow to the WWTP, and (3) predicting the influent flow at future time horizons to effectively manage ICWT. Since the influent flow to the Muusøya WWTP has a significant impact on ICWT, the associated hydraulic facilities should be adjusted and arranged based on the expected influent flow volume.

A hydraulic model was used for the first and second tasks to assess the hydraulic behaviors of the sewer system under different control scenarios. The observed precipitation in 2014 with a temporal resolution of

1 min was used for the hydraulic simulation. The simulation ran continuously from January 1, 2014 to December 31, 2014 as a baseline scenario to compare against different control scenarios. The simulations showed that ICWT could efficiently reduce total overflow from the sewer system as well as overflow from the Muusøya WWTP. LSTM was employed to predict inflow to the Muusøya WWTP and improve ICWT operations. The experiments demonstrated that LSTM could be of great use in predicting sewer flow.

In Paper IV, the hydraulic model revealed that, concerning the whole system's performance, ICWT might place an additional burden on the Søren Lemmich pump station. In fact, under the ICWT scheme, the Søren Lemmich pump station would become the bottleneck of the whole system. Hence, it would need to operate with high sensitivity. If the Søren Lemmich pump station could not pump wastewater in a timely manner, ICWT would only represent an extra burden for the Bragernes catchment and would not mitigate the overflow.

The operation of a pump station depends on water level information. Pumps are activated when the water level reaches the start level of the pumps. Thus, the operation of a pump station can be enhanced only if accurate water level prediction information can be provided (Chiang et al., 2010). For a pump to operate in a timely manner and for operators to have enough response time, the system must employ a model that can provide multi-step-ahead water level information (Liu et al., 2016; Chang et al., 2014; Chen et al., 2014). AI was employed for this purpose. The experiments demonstrated that LSTM was superior to the GRU, RNN, FFNN, and SVR approaches.

4.4. Paper V: DeepCSO

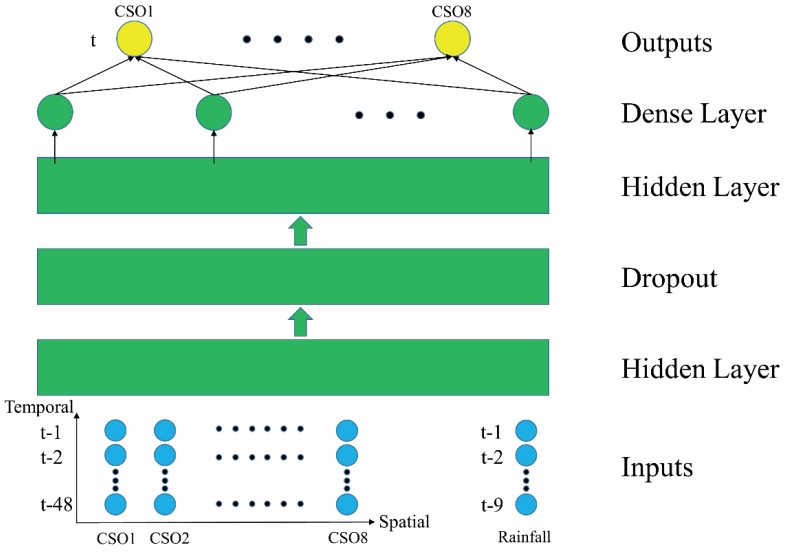


Figure 15. *Architecture of the proposed DeepCSO model*

Paper V builds on Papers I–IV, maintaining that the hydraulic model is less suitable for real-time purposes. Alternatively, data-driven methods have gained substantial interest, but most studies have only focused on modeling a single component of a sewer system and supply information at a very abstract level. Accordingly, Paper V considers the spatiotemporal correlations of the sewer system, in which the hydrological behavior of one part of the sewer system is related to its previous status, rainfall, and upstream—or even downstream—parts and proposes the DeepCSO model, which aims at forecasting CSO events from multiple CSO structures simultaneously in near real time at a citywide level. The DeepCSO model is based on large and high-resolution sensors that can be deployed throughout cities and uses MTL techniques. The DeepCSO model constitutes an intermediate methodology that combines the flexibility of data-driven methods and the rich information contained in deterministic methods while avoiding the drawbacks of these two methods. A comparison of the results demonstrated that the multi-task approach was generally more effective than the single-task approach. Furthermore, the GRU and

LSTM were especially suitable for capturing the temporal and spatial evolution of CSO events, and they were superior to other methods.

4.5. Conclusions

The main objectives of this research were to confirm or discard the hypotheses regarding the feasibility of using proactive sewer management strategies to minimize overflow and the integrated power of the IoT, hydraulic model, and AI in enhancing sewer system control, resulting in fewer overflows. The data collected by the IoT is a prerequisite for developing and calibrating the hydraulic model and for ensuring that AI models make precise predictions. The hydraulic model, as a model suitable for simulation, was applied to assess sewer management strategies. AI models were used as models suitable for prediction.

The Drammen sewer system was selected for a case study. Two strategies, namely, in-line storage control and ICWT, were proposed and investigated. A fully detailed hydraulic model was developed to assess the performance of these sewer management strategies. For in-line storage control, the hydraulic model was used to identify which part of the sewer system had the leftover capacity and to test the efficiency of the proposed in-line storage control strategy. For ICWT, the hydraulic model was used to assess the hydraulic behaviors of the sewer system under different control scenarios. The simulations demonstrated that both in-line storage control and ICWT could provide extra benefits for overflow mitigation.

Additionally, the potential of using AI as a model suitable for prediction was explored. Most IoT data collected by sensors have a temporal dimension and can be modeled as a time series. The case study demonstrated that, of various AI techniques, a type of RNN—LSTM—is able to precisely predict the time series data from the IoT. The IoT and LSTM can be powerful tools for sewer system managers or engineers, who can take advantage of real-time data and predictions to improve their decision-making.

Studies related to sewer systems require the modeling of complex and dynamic urban hydrological processes. Of the available models, two approaches were used in this study, both of which have advantages and disadvantages. In one, the extensively used hydraulic model approach,

hydrological/hydraulic principles are explicitly modeled. However, hydraulic models are less appropriate for real-time purposes. Conversely, the implicit machine learning approach can provide predictions in real time but cannot describe the hydrological/hydraulic behavior of a sewer system in detail. Moreover, machine learning models are highly system-specific and thus are not easily adaptable to new conditions. To solve practical problems, engineers or researchers should consider combining the advantages of both approaches so they can complement each other based on the available data and purposes. Hydraulic models can be used to gain a better understanding of certain control strategies or to assess the performance of the sewer system (offline applications). AI can be used for real-time prediction purposes for sewer systems (online applications). In engineering practice, the two types of models are complementary and are not competitors.

Finally, the use of MTL and the deep learning-based DeepCSO model was demonstrated and evaluated to simultaneously predict overflow at multiple CSO structures. The proposed model provides an intermediate methodology that combines the flexibility of data-driven methods and the rich information contained in hydraulic models while avoiding the drawbacks of these two approaches. A comparison of the results demonstrated that the deep learning-based multi-task model is superior to traditional methods.

5. Perspectives

Ubiquitous sensors monitor infrastructure such as sewer systems and collect large amounts of data. Data analytics can be the sticking point for leveraging intelligent infrastructure management. Since deep learning offers many potential benefits, further studies on adapting deep learning to water resource-related fields would be of interest.

5.1. Water quality data and deep learning

A limitation of this study is that only volume-based real-time control was investigated. In this type of control, the control strategy is designed to minimize the overflow volume, partially because only sensors for rainfall and water quantities are installed. As a result, only rainfall and water quantity data were available for model development in the case study.

Advances in water quality instrumentation have facilitated the real-time collection of water quality data (Leeder et al., 2012), which could enable more comprehensive control strategies, such as pollution-based and immission-based real-time control. Pollution-based real-time control attempts to minimize the total amount of pollutants entering the receiving water body, while immission-based real-time control intends to directly optimize the receiving water quality (Vanrolleghem et al., 2005).

As in this study, AI could be used as a model suitable for prediction in pollution-based and immission-based real-time control, but focusing more on water quality data. In addition to prediction, soft sensors represent another interesting field for the application of AI. The online monitoring of key water quality variables, such as chemical oxygen demand and total phosphorus, limits real-time control of water quality. AI-based soft sensors could provide an online estimation of these variables based on historical measurements of easy-to-measure process variables (Shang et al., 2014). This method could be an effective solution to achieve online monitoring of water quality parameters.

5.2. Convolutional neural networks for image data

This study focused on time series data collected by the IoT. Image data is another major source of data. Currently, CNNs (LeCun & Bengio, 1995) are the dominant method in terms of image processing and computer visioning. In water- and wastewater-related research, CNNs have been used for, among other purposes, the classification of microbeads in urban wastewater (Yurtsever & Yurtsever, 2019) and automatic water impurity detection (Gupta & Ruebush, 2019).

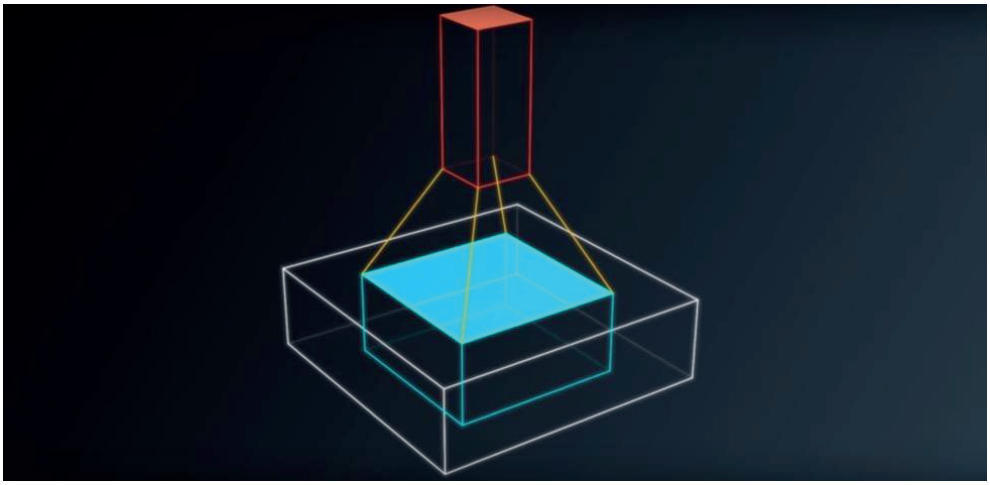


Figure 16. *Convolutional neural networks (source: www.udacity.com)*

An image typically has three dimensions: width, height (the number of pixels of the image), and depth (in RGB, red, green, and blue). A CNN uses filters to extract information at a more abstract level from the original large image. The filter looks at small patches of the image, and the weights and biases for a given output layer are shared across all patches in a given input layer.

AI could allow more image data from water-related fields to be used in an unprecedented way, such as rainfall nowcasting based on satellite images or rainfall radar, sewer pipe damage detection using CNNs and sewer pipe inspection images, improving coagulant dosing control through analyzing images of flocs, analyzing membrane-fouling images for monitoring, and controlling the fouling of membranes.

5.3. Autoencoders for anomaly detection

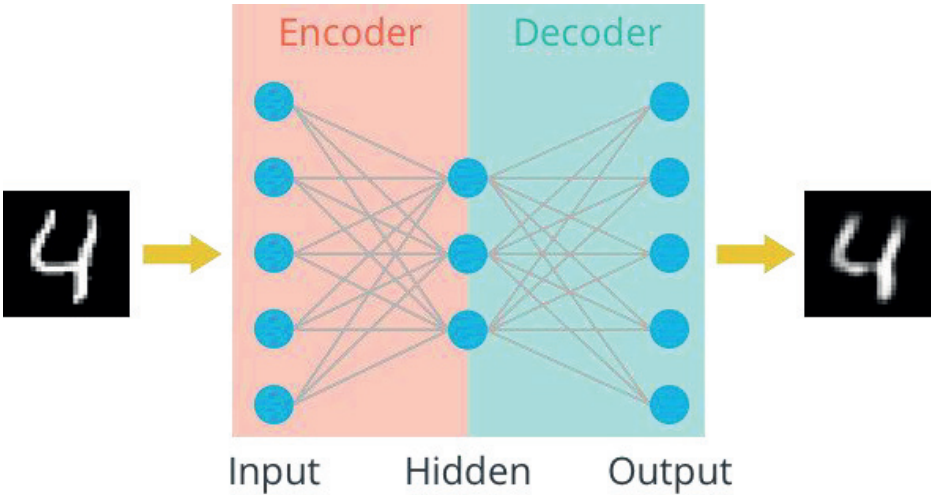


Figure 17. *Architecture of an autoencoder (source: www.udacity.com)*

Anomaly detection techniques can provide increased knowledge of system performance in the form of event identification (Leeder et al., 2012). In the water and wastewater fields, anomaly detection techniques can be used for detecting changes or anomalies regarding water quality time series data (Muharemi et al., 2019), pipe bursts (Wu et al., 2016), equipment failures, and the like.

The autoencoder is a promising AI technique for anomaly detection. An autoencoder consists of two major parts—an encoder and decoder—according to its position before or after the bottleneck layer. The input data first pass through the encoder, which makes a compressed representation of the input data at the bottleneck layer. Then, this representation is passed through a decoder to reconstruct the input data at the output layer. The objective of training the autoencoder is to reduce the difference between the input and output data.

In autoencoder-based anomaly detection, the autoencoder is trained using data collected during normal situations. Subsequently, it computes the reconstruction error, which is the difference between the input and output data. If the reconstruction error exceeds a certain threshold, one can assume that the pattern of

input data is novel and that it therefore has a higher probability of being non-normal.

5.4. Reinforcement learning for control optimization

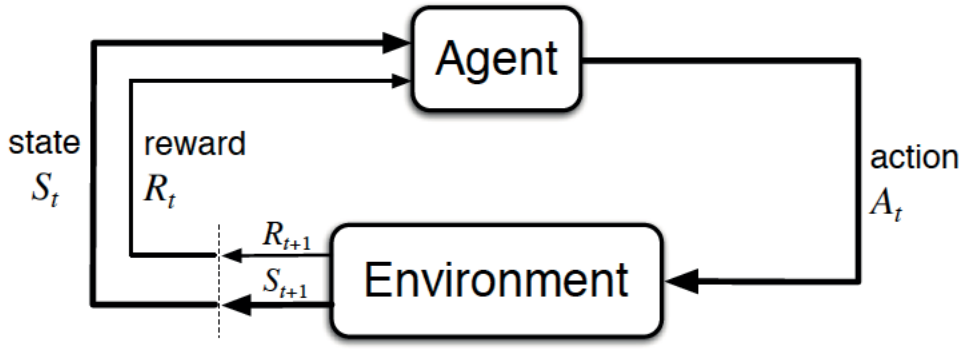


Figure 18. *The agent–environment interaction in reinforcement learning (source: Sutton & Barto, 2017)*

Identifying how to control a complex system such as a sewer system is highly challenging. Conventional control optimization has traditionally relied on common expert knowledge and previous experience (Pang et al., 2019). Reinforcement learning (RL) may be a promising technology for optimizing the control of complex systems.

RL is a generic approach designed to automatically devise a decision policy or control strategy (Blumensaat et al., 2019). The RL framework consists of an agent and its environment. At each time step, the agent receives the environment's state, and, in response, the agent chooses an appropriate action. In the next time step, the agent receives a reward and a new state. The agent's goal is to maximize the expected cumulative reward.

Because RL derives a control strategy by trial and error, a system model is required to avoid direct disturbances to real systems (Blumensaat et al., 2019). In terms of sewer system control, a hydraulic model of the sewer system may be used to validate different control rules and then allow the agent to learn proper control reactions.

Recent developments in combining deep learning and RL have exhibited effective performance. The

latest iteration of AlphaGo, AlphaGo Zero, is based on deep learning and RL. In contrast to AlphaGo, which is trained on amateur and professional games, AlphaGo Zero learns to play by playing games against itself, without any prior knowledge and with only basic Go game rules as input. After training, AlphaGo Zero defeated AlphaGo by a score of 100 games to 0.

References

Ahm, M. K. S. (2015). *Adjustment of rainfall estimates from weather radars using in-situ stormwater drainage sensors.* (Doctoral dissertation, Aalborg University).

Almazyad, A. S., Seddiq, Y. M., Alotaibi, A. M., Al-Nasheri, A. Y., BenSaleh, M. S., Obeid, A. M., & Qasim, S. M. (2014). A proposed scalable design and simulation of wireless sensor network-based long-distance water pipeline leakage monitoring system. *Sensors*, *14*(2), 3557–3577.

Andersen, S. T., Albrechtsen, H. J., & Mark, O. (2015). *Urban flooding and health risk analysis by use of quantitative microbial risk assessment: Limitations and improvements*. (Doctoral dissertation, Technical University of Denmark).

Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer Networks, 54*(15), 2787–2805.

Autixier, L., Mailhot, A., Bolduc, S., Madoux-Humery, A. S., 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. *Science of the Total Environment, 499*, 238–247.

Beheshti, M., & Sægrov, S. (2018). Quantification assessment of extraneous water infiltration and inflow by analysis of the thermal behavior of the sewer network. *Water, 10*(8), 1070.

Beheshti, M., & Sægrov, S. (2019). Detection of extraneous water ingress into the sewer system using tandem methods–a case study in Trondheim city. *Water Science and Technology, 79*(2), 231-239.

Beheshti, M., Sægrov, S., & Ugarelli, R. (2015). Infiltration/inflow assessment and detection in urban sewer system. *Vann. 2015, 1 (1),* 24-34.

Blumensaat, F., Leitão, J. P., Ort, C., Rieckermann, J., Scheidegger, A., Vanrolleghem, P. A., & Villez, K. (2019). How urban storm and wastewater management prepares for emerging opportunities and threats: Digital transformation, ubiquitous sensing, new data sources, and beyond-A horizon scan. *Environmental science & technology, 53(15), 8488-8498.*

Breinholt, A., Mikkelsen, P. S., Madsen, H., & Grum, M. (2012). *Uncertainty in prediction and simulation of flow in sewer systems.* (Doctoral dissertation, Technical University of Denmark).

Bruaset, Stian. (2019). *Long-term sustainable management of the urban water and wastewater pipe networks*. (Doctoral dissertation, Norwegian University of Science and Technology).

Carbone, M., Garofalo, G., & Piro, P. (2014). Decentralized real time control in combined sewer system

by using smart objects. *Procedia Engineering*, *89*, 473–478.

Caruana, R. (1998). A dozen tricks with multitask learning. In *Neural networks: tricks of the trade* (pp. 165–191). Berlin, Heidelberg: Springer.

Chang, B., Wang, H. Y., Peng, T. Y., & Hsu, Y. S. (2010). Development and evaluation of a city-wide wireless weather sensor network. *Educational Technology & Society*, *13*(3), 270–280.

Chang, F. J., Chen, P. A., Lu, Y. R., Huang, E., & Chang, K. Y. (2014). Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. *Journal of Hydrology*, *517*, 836–846.

Chen, J., Ganigué, R., Liu, Y., and Yuan, Z. (2014). Real-time multistep prediction of sewer flow for online chemical dosing control. *Journal of Environmental Engineering*, *140*(11), 04014037.

Chiang, Y. M., Chang, L. C., Tsai, M. J., Wang, Y. F., and Chang, F. J. (2010). Dynamic neural networks for real-time water level predictions of sewerage systems-covering gauged and ungauged sites. *Hydrology and Earth System Sciences*, *14*(7), 1309–1319.

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

Compagni, R. D., Polesel, F., von Borries, K. J., Zhang, Z., Turolla, A., Antonelli, M., & Vezzaro, L. (2019). Modelling micropollutant fate in sewer systems–A new systematic approach to support conceptual model construction based on in-sewer hydraulic retention time. *Journal of Environmental Management*, *246*, 141–149.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

De Toffol, S., Engelhard, C., & Rauch, W. (2007). Combined sewer system versus separate system–a comparison of ecological and economical performance indicators. *Water science and technology, 55*(4), 255-264

El-Din, A. G., & Smith, D. W. (2002). A neural network model to predict the wastewater inflow incorporating rainfall events. *Water Research*, *36*(5), 1115–1126.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.

Ferriman, A. (2007). BMJ readers choose the" sanitary revolution" as greatest medical advance since 1840. *BMJ: British Medical Journal*, *334*(7585), 111.

Franco, S., Gaetano, V., & Gianni, T. (2018). Urbanization and climate change impacts on surface water quality: Enhancing the resilience by reducing impervious surfaces. *Water research, 144*, 491-502.

Gaitan, S., van de Giesen, N. C., & ten Veldhuis, J. A. E. (2016). Can urban pluvial flooding be predicted by open spatial data and weather data? *Environmental Modelling & Software*, *85*, 156–171.

Ganora, D., Isacco, S., & Claps, P. (2017). Framework for enhanced stormwater management by optimization of sewer pumping stations. *Journal of Environmental Engineering, 143(8). 1943-7870.*

Garofalo, G., Giordano, A., Piro, P., Spezzano, G., & Vinci, A. (2017). A distributed realtime approach for mitigating CSO and flooding in urban drainage systems. *Journal of Network and Computer Applications*, *78*, 30–42.

Grum, M., Thornberg, D., Christensen, M. L., Shididi, S. A., & Thirsing, C. (2011, September). 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*.

Gupta, A., & Ruebush, E. (2019). AquaSight: Automatic water impurity detection utilizing convolutional neural networks. arXiv preprint arXiv:1907.07573.

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

Huang, P., Jin, Y., Hou, D., Yu, J., Tu, D., Cao, Y., & Zhang, G. (2017). Online classification of contaminants based on multi-classification support vector machine using conventional water quality sensors. *Sensors*, *17*(3), 581.

Jaramillo, F., Orchard, M., Muñoz, C., Antileo, C., Sáez, D., & Espinoza, P. (2018). On-line estimation of the aerobic phase length for partial nitrification processes in SBR based on features extraction and SVM classification. *Chemical Engineering Journal*, *331,* 114–123.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464–1480. doi:10.1109/5.58325.

Kruger, C. P., Abu-Mahfouz, A. M., & Hancke, G. P. (2015, March). Rapid prototyping of a wireless sensor network gateway for the internet of things using off-the-shelf components. In *2015 IEEE International Conference on Industrial Technolog (ICIT)* (pp. 1926–1931). Seville, Spain: IEEE.

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, *3361*(10), 1995.

Lee, E. H., Lee, Y. S., Joo, J. G., Jung, D., & Kim, J. H. (2017). Investigating the impact of proactive pump operation and capacity expansion on urban drainage system resilience. *Journal of Water Resources Planning and Management, 143(7). 1943-5452.*

Leeder, A., Mounce, S. R., & Boxall, J. B. (2012). Analysis of multi-parameter water quality data using event detection software on laboratory simulated events. In *WDSA 2012: 14th Water Distribution Systems Analysis Conference, 24-27 September 2012 in Adelaide, South Australia* (p. 1018). Engineers Australia.

Li, J., Sharma, K., Liu, Y., Jiang, G., & Yuan, Z. (2019). Real-time prediction of rain-impacted sewage flow for on-line control of chemical dosing in sewers. *Water Research*, *149*, 311–321.

Lim, J. S., Kim, J., Friedman, J., Lee, U., Vieira, L., Rosso, D., Gerla, M., & Srivastava, M. B. (2013). SewerSnort: A drifting sensor for in situ wastewater collection system gas monitoring. *Ad Hoc Networks*, *11*(4), 1456–1471.

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

Löwe, R., Madsen, H., & Mikkelsen, P. S. (2014). *Probabilistic forecasting for on-line operation of urban drainage systems.* (Doctoral dissertation, Technical University of Denmark).

Lucas, W. C., & Sample, D. J. (2015). Reducing combined sewer overflows by using outlet controls for green stormwater infrastructure: Case study in Richmond, Virginia. *Journal of Hydrology*, *520*, 473–488.

Lund, N. S. V., Falk, A. K. V., Borup, M., Madsen, H., & Steen Mikkelsen, P. (2018). Model predictive control of urban drainage systems: A review and perspective towards smart real-time water management. *Critical reviews in environmental science and technology, 48*(3), 279-339.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In L. M. Le Cam, & J. Neyman (Eds.), *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 281–297). Berkeley, CA: University of California Press.

Maxym L. & Sreekanth L. (2015). Methodology to develop optimum control strategies: Controlling wastewater plant inflows. In *ISAWater/Wastewater and Automatic Controls Symposium*, Orlando, FL, USA.

Mayer, A., Winkler, R., & Fry, L. (2014). Classification of watersheds into integrated social and biophysical indicators with clustering analysis. *Ecological Indicators*, *45*, 340–349.

Mollerup, A. L., Sin, G., Mikkelsen, P. S., Johansen, N. B., & Thornberg, D. (2015). A methodological approach to designing sewer system control. (Doctoral dissertation, Technical University of Denmark).

Mounce, S. R., Shepherd, W., Sailor, G., Shucksmith, J., & Saul, A. J. (2014). Predicting combined sewer

overflows chamber depth using artificial neural networks with rainfall radar data. *Water Science and Technology*, *69*(6), 1326–1333.

Muharemi, F., Logofătu, D., & Leon, F. (2019). Machine learning approaches for anomaly detection of water quality on a real-world data set. *Journal of Information and Telecommunication, 16(3), 294-307.*

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, *26*(4), 354–359.

Ngo, T., Yoo, D., Lee, Y., & Kim, J. (2016). Optimization of upstream detention reservoir facilities for downstream flood mitigation in urban areas. *Water*, *8*(7), 290.

Nie, L., Lindholm, O., Lindholm, G., & Syversen, E. (2009). Impacts of climate change on urban drainage systems–a case study in Fredrikstad, Norway. *Urban Water Journal,* 6(4), 323-332.

Nilsen, V., Lier, J. A., Bjerkholt, J. T., & Lindholm, O. G. (2011). Analysing urban floods and combined sewer overflows in a changing climate. *Journal of water and climate change, 2*(4), 260-271.

Pang, J., Yang, S., He, L., Chen, Y., & Ren, N. (2019). Intelligent control/operational strategies in WWTPs through an integrated Q-learning algorithm with ASM2d-ruided Reward. *Water*, *11*(5), 927.

Popescu, I. (2014). *Computational hydraulics*. IWA Publishing.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv* pre-print. arXiv:1706.05098.

Saagi, R., Flores-Alsina, X., Fu, G., Butler, D., Gernaey, K. V., & Jeppsson, U. (2016). Catchment & sewer network simulation model to benchmark control strategies within urban wastewater systems. *Environmental Modelling & Software*, *78*, 16–30.

Seggelke, K., Rosenwinkel, K. H., Vanrolleghem, P. A., & Krebs, P. (2005). Integrated operation of sewer system and WWTP by simulation-based control of the WWTP inflow. *Water Science and Technology*, *52*(5), 195–203.

Shang, C., Yang, F., Huang, D., & Lyu, W. (2014). Data-driven soft sensor development based on deep learning technique. *Journal of Process Control*, *24*(3), 223–233.

Smarra, F., Jain, A., de Rubeis, T., Ambrosini, D., D'Innocenzo, A., & Mangharam, R. (2018). Data-driven model predictive control using random forests for building energy optimization and climate control. *Applied Energy*, *226*, 1252–1272.

Stoianov, I., Nachman, L., Madden, S., & Tokmouline, T. (2007, April). PIPENET: A wireless sensor network for pipeline monitoring. In *6th International Symposium on Information Processing in Sensor Networks* (pp. 264–273). Cambridge, MA: IEEE.

Sutton, R. S., & Barto, A. G. (2017). *Reinforcement learning: An introduction* (2nd ed., in progress). Cambridge: MIT Press.

Torgersen, G. (2017). *Sustainable planning to reduce urban flooding: an interdisciplinary approach.* (Doctoral dissertation, Norwegian University of Life Sciences).

Torgersen, G., Bjerkholt, J., & Lindholm, O. (2014). Addressing flooding and SuDS when improving drainage and sewerage systems—A comparative study of selected Scandinavian cities. *Water*, *6*(4), 839–857.

Vanrolleghem, P. A., & Vaneeckhaute, C. (2014). Resource recovery from wastewater and sludge: Modelling and control challenges. *IWA Specialist conference on Global Challenges: Sustainable Wastewater Treatment and Resource Recovery*. International Water Association (IWA).

Vanrolleghem, P. A., Benedetti, L., & Meirlaen, J. (2005). Modelling and real-time control of the integrated urban wastewater system. *Environmental Modelling & Software*, *20*(4), 427–442.

Wang, X., Kvaal, K., & Ratnaweera, H. (2017). Characterization of influent wastewater with periodic variation and snow melting effect in cold climate area. *Computers & Chemical Engineering*, *106*, 202–211.

Wang, X., Ratnaweera, H., Holm, J. A., & Olsbu, V. (2017). Statistical monitoring and dynamic simulation of a wastewater treatment plant: A combined approach to achieve model predictive control. *Journal of Environmental Management, 193,* 1–7.

Wei, X. (2013). *Modeling and optimization of wastewater treatment process with a data-driven approach* (Doctoral dissertation, The University of Iowa).

Wu, Y., Liu, S., Wu, X., Liu, Y., & Guan, Y. (2016). Burst detection in district metering areas using a data driven clustering algorithm. *Water Research*, *100*, 28–37.

Yurtsever, M., & Yurtsever, U. (2019). Use of a convolutional neural network for the classification of microbeads in urban wastewater. *Chemosphere*, *216*, 271–280.

Zhang, Y., & Yang, Q. (2017). A survey on multi-task learning. *arXiv* pre-print. arXiv:1707.08114.

Zhou, Q. (2014). A review of sustainable urban drainage systems considering the climate change and urbanization impacts. *Water*, *6*(4), 976–992.

Appended papers

Paper I

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

Journal of Hydrology 556 (2018) 409–418

Contents lists available at ScienceDirect

Journal of Hydrology

journal homepage: www.elsevier.com/locate/jhydrol

Research papers

Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring

Duo Zhang [a,*], Geir Lindholm [b], Harsha Ratnaweera [a]

[a] Faculty of Sciences and Technology, Norwegian University of Life Sciences, 1432 Ås, Norway
[b] Rosim AS, Brobekkveien 80, 0582 Oslo, Norway

ARTICLE INFO

Article history:
Received 29 July 2017
Received in revised form 8 November 2017
Accepted 10 November 2017
Available online 14 November 2017
This manuscript was handled by G. Syme, Editor-in-Chief.

Keywords:
Combined sewer overflow
Deep learning
Long short-term memory
Gated recurrent unit
Internet of Things

ABSTRACT

Combined sewer overflow causes severe water pollution, urban flooding and reduced treatment plant efficiency. Understanding the behavior of CSO structures is vital for urban flooding prevention and overflow control. Neural networks have been extensively applied in water resource related fields. In this study, we collect data from an Internet of Things monitoring CSO structure and build different neural network models for simulating and predicting the water level of the CSO structure. Through a comparison of four different neural networks, namely multilayer perceptron (MLP), wavelet neural network (WNN), long short-term memory (LSTM) and gated recurrent unit (GRU), the LSTM and GRU present superior capabilities for multi-step-ahead time series prediction. Furthermore, GRU achieves prediction performances similar to LSTM with a quicker learning curve.

1. Introduction

The sewer system is one of the most important infrastructures in modern cities. Although separate sewer systems have been introduced, decades-old combined sewer system still continues to service many developed cities. Under normal circumstances, these systems transport all wastewater to the wastewater treatment plant. During heavy rainfall events, when wastewater volume exceeds the capacity of the sewer system, the combined sewer system is designed to overflow and discharge wastewater directly to water bodies. This phenomenon is called Combined Sewer Overflow (CSO). Climate change and rapid urbanization in metropolitan areas have resulted in an increase in the CSO over the last decades. Decreasing the influence of overflows is an important part of reducing pollution in water bodies (Garofalo et al., 2017; Autixier et al., 2014; Lucas and Sample 2015). Several cities have employed Internet of Things (IoT) to monitor the performance of sewer systems and to provide useful data to managers and engineers (Montserrat et al., 2015). The basic idea of IoT is to let 'things'-such as sensors, actuators, mobile or desktop devices, etc. -be able to interact and cooperate with each other through wireless communication protocols (Giusto, 2010). IoT is a new catalyst for the data explosion. The world will have 50 billion IoT devices by 2020, these devices will generate massive data, to extract insight from data collected by IoT and convey the extracted insight to stakeholders, the analysis and application of IoT data are as important as data collection. For example, for the CSO structures, in addition to properly monitoring, it is also imperative to construct a model to predict the CSO events by utilizing data from the realm of IoT. The model is expected to provide sufficient response time for making decisions about CSO control, protect downstream hydraulic infrastructures during extreme rainfall events and mitigate the impact of CSO on the receiving waters (Garofalo et al., 2017; Chang et al., 2014; Mounce et al., 2014; Darsono and Labadie, 2007; Grum et al., 2011).

Most IoT data collected by sensors have a temporal dimension and can be modeled as a time series. Characterized by high complexity, dynamism, and non-stationarity, time series forecasting has always presented a challenge to hydrologists (Nourani et al., 2014). Recent years have seen a significant rise in the number of machine learning approaches applied to hydrologic time series forecasting. Among numerous machine learning algorithms such as support vector machine, k-nearest neighbors or linear discriminant analysis, Artificial Neural Network (ANN) had shown superior performance for IoT time series data (Alam et al., 2016). As the most traditional ANN, multilayer perceptron (MLP) is a very popular tool for handling hydrologic time series problems in previous researches. A few studies have explored using MLP and data

* Corresponding author.
E-mail addresses: Duo.Zhang@nmbu.no (D. Zhang), geir@rosim.no (G. Lindholm), Harsha.Ratnaweera@nmbu.no (H. Ratnaweera).

https://doi.org/10.1016/j.jhydrol.2017.11.018
0022-1694/

collected by IoT to predict CSO events (Kurth et al., 2008; Mounce et al., 2014). Another common practice in solving hydrologic time series problems is couple Wavelet Transforms (WT) with different ANN such as the MLP. WT is a kind of technique that can illustrate how the frequency content of a signal changes over time. WT relies on the mother wavelet, which can be changed according to the shape and compactness of the signal (Barzegar et al., 2017). ANN that use mother wavelet as activation functions instead of traditionally used sigmoid function are called Wavelet Neural Network (WNN) (Wang and Ding, 2003). WNN had shown have better fitting in estimating hydrologic time series data (Nourani et al., 2014). The advances of WNN had led to an increase in hydrological studies for different applications such as hydro-climatology time series prediction (Alexandridis and Zapranis, 2013), reservoir inflow modeling (Abghari et al., 2009; Chen et al., 2006) and evaporation prediction (Abghari et al., 2012).

A state-of-the-art branch of ANN is deep learning (Hinton et al., 2006). ANN have been less active during a period called Artificial Intelligence (AI) winter (Marçais and de Dreuzy, 2017). In recent years, renewed interest in ANN surged, in part due to exposure in popular media when the computer program (Google DeepMind's AlphaGo) defeated Go game world champion (Silver et al., 2016). The game changer behind the revival of ANN is deep learning. Deep learning is a topic that is making big waves now, alongside AlphaGo, a fascinating application of deep learning is the latest Google Neural Machine Translation (GNMT) system released in 2016. The GNMT system produces translation quality that is vastly improved compared to previous systems, even for a notoriously difficult language pair: Chinese to English (Google, 2016). The technology utilized by the GNMT system is a kind of Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997).

RNN is a kind of advanced ANN that involves feedback connections in the architecture. Unlike traditional ANN that all the inputs and outputs are independent of each other, the output of RNN depends on previous computations as well as calculations of the current time step. One way to think about RNN is that they have a "memory" that captures information from previous computations. RNN are considered very effective in modeling complex hydrologic time series. However, due to the difficulty of learning long-range dependencies, the training of RNN can be extremely challenging. This problem is commonly known as the vanishing/-exploding gradient problem (Hochreiter and Schmidhuber, 1997). Therefore, certain types of RNN, such as LSTM, were designed to solve these problems. LSTM is today's dominant technology in the field of natural language processing. When processing natural language, the model has to consider not only the current word but also other adjoining words in the sentence or even paragraph. Data with this kind of context information called sequential data. Time series data are the most popular form of sequential data. Stimulated by the success of LSTM on machine translation, a few studies have explored the power of LSTM on time series prediction and obtained promising results (Zaytar and Amrani, 2016; Ma et al., 2015; Xingjian et al., 2015).

One drawback of LSTM is its complexity. Therefore, simplifying LSTM, has become a highly researched topic in the field of computer science. The Gated Recurrent Unit (GRU) (Cho et al., 2014) first proposed in 2014 is one of the most successful LSTM variants. Chung et al., (2014) evaluated the performance of LSTM and GRU on the tasks of polyphonic music modeling and speech signal modeling. The evaluation revealed the GRU to be comparable to LSTM, and better performing than more traditional RNN. To the best of the author's knowledge, no prior studies have employed GRU to analyze time series data. Given the success of the GRU in modeling sequential data, such as speech synthesis (Wu & King, 2016), sentiment classification (Tang et al., 2015), targeted sentiment analysis (Zhang et al., 2016), etc., the effectiveness of GRU on the prediction of hydrologic time series has to be investigated.

The objectives of this research are to investigate the performance of LSTM and GRU on predicting the multi-step ahead hydrologic time series data collected by the IoT, and to compare with the performance of traditional methods such as MLP and WNN. The remainder of this paper is organized as follows: a general description about the study area and different algorithms are provided in the first section. The simulation results are presented in section two, where the prediction efficiencies of the studied algorithms are compared. Conclusions and envisioned future developments are discussed at the end of this paper.

2. Method and data

2.1. Case study area

Drammen is a coastal city with approximately 150, 000 inhabitants located in the southeast of Norway. It is the fourth largest city in Norway, and the capital of Buskerud County. The catchment of Drammen's sewer system is around 15 km^2 and the total length of the sewer system is approximately 500 km. Fig. 1 is an overview of Drammen. In Fig. 1, the rain gauges are denoted by squares and CSO structures are denoted by triangles. The traditional city center distributes along the Drammen Fjord, which flows through Drammen. The sewer system in the city center mainly consists of the combined sewer system. During heavy rainfall events, the combined sewer system in the city center discharges overflows directly into the Drammen Fjord though CSO structures, cause heavy pollution. In order to mitigate pollutions from the combined sewer system, the government of Drammen initialized the Regnbyge 3 M project. The ultimate goal of the Regnbyge 3 M project is integrates intelligent monitoring, modeling and control solutions, manage the sewer system and water bodies in a holistic way.

This study focuses on the CSO structure of Dr_Hansteensgate, which located adjacent to important infrastructures such as the train station, shopping center, and the stadium. The risks of overflow and consequent pollution risks may be decreased if accurate hydrological time series prediction can be provided. Therefore, to test the performance of different ANN models against this objective, Dr_Hansteensgate CSO (highlighted by a circle in Fig. 1) was selected for consideration.

2.2. The Regnbyge.no IoT

In order to monitor the hydraulic behavior of CSO structures, data must be provided for further analysis and model construction. In the first phase of the Regnbyge 3 M project, Rosim AS, Norway developed an IoT, Regnbyge.no, to monitor the sewer system. Typical IoT architecture consists of three layers: sensing layer, network layer, and application layer (Atzori et al., 2010). The sensing layer includes devices that are deployed in the field, such as sensors, actuators, and wireless transmitters. The major function of the sensing layer is to collect data. The network layer provides a bridge between the sensing layer and the application layer. In the network layer, data sensed by sensors transmit through wireless communication protocols to the application layer. On the top of the IoT architecture is the application layer, it enabling users, algorithms, or models to interact with devices in the sensing layer.

The sensing layer of the Regnbyge.no IoT consists of ultrasonic water level sensors produced by NIVUS GmbH, Germany and rain gauges. The measurements collected by the water level sensors and rain gauges transmit using wireless telemetry to a remote server located inside Rosim AS. A spatial database is designed to manage the transmitted data, simplifying the process for searching,

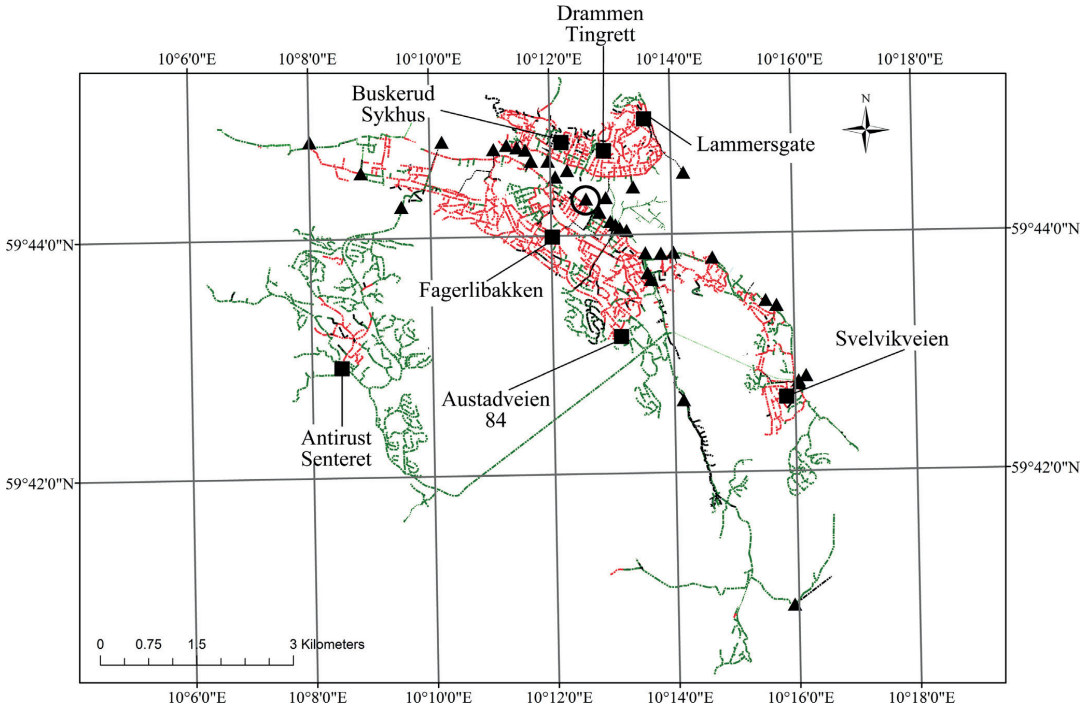


Fig. 1. Overview of Drammen sewer system, Norway, squares denote rain gauges and triangles denote CSO structures.

editing, and giving real-time information in a user-friendly way. Once data is stored in the database, information can be shared via the web and accessed through a desktop or mobile devices. Furthermore, the data in the database is visualized using web-based geographic information system (Web-GIS). Currently, the Web-GIS only provides an interface for displaying real-time and historical data of the water level and rainfall. In the next phase of the Regnbyge 3 M project, forecasting models and actuators will be implemented in the Regnbyge.no IoT. The forecasting model will predict future water levels, according to the predicted water level, managers of the sewer system can operate actuators timely using control commands sent from the server, thus guide the behavior of CSO structures.

Fig. 2 displays major components of the Regnbyge.no IoT platform, Fig. 2 (a) shows an ultrasonic water level sensor installed on the top of a CSO structure, Fig. 2 (b) illustrates how engineers embed a sensor's antenna in the road surface, Fig. 2 (c) is the installation of a transmitter on a lamp post. Fig. 2 (d) demonstrates the user interface of the Regnbyge.no IoT.

2.3. ANN

2.3.1. MLP

MLP is one of the most popular ANN, which is usually comprised of input layer, hidden layer, and output layer. There are some neurons in each layer and different layers are connected by weights and bias. Fig. 3 shows the architecture of a three-layered MLP. In Fig. 3, the circles denote neurons and lines between circles denote weights. The MLP first computes the weighted sum of the inputs, which can be mathematically represented as:

$$s = \sum_{i=1}^{n} w_i x_i + b \tag{1}$$

where w_i represents the weights, x_i is the inputs, b is the bias. Afterwards, the computed weighted sum s is fed into the neuron. The neuron consists of an activation function. There are various functions that can be used and the most classical one is the sigmoid function. The sigmoid function is defined as:

$$f(s) = \frac{1}{1+e^{-s}} \tag{2}$$

When training the MLP, the ultimate goal is to minimize the cost function. The cost function can be defined as:

$$C = \frac{1}{2}(f(s) - f(s)_{observed})^2 \tag{3}$$

where C is the cost of the cost function, $f(s)$ is the predicted output from neuron and $f(s)_{observed}$ is the observed true value.

Back propagation (BP) is the most commonly used training algorithms. BP uses the chain rule of differentiation to calculate the partial derivative or gradient of the cost corresponding to the weights. For a single training example of a neuron, the gradient of cost C corresponding to a weight w_i can be represented as:

$$\frac{\partial C}{\partial w_i} = \frac{\partial C}{\partial f(s)} \frac{\partial f(s)}{\partial s} \frac{\partial s}{\partial w_i} \tag{4}$$

The partial derivative of the cost function corresponding to activation function is:

$$\frac{\partial C}{\partial f(s)} = f(s) - f(s)_{observed} \tag{5}$$

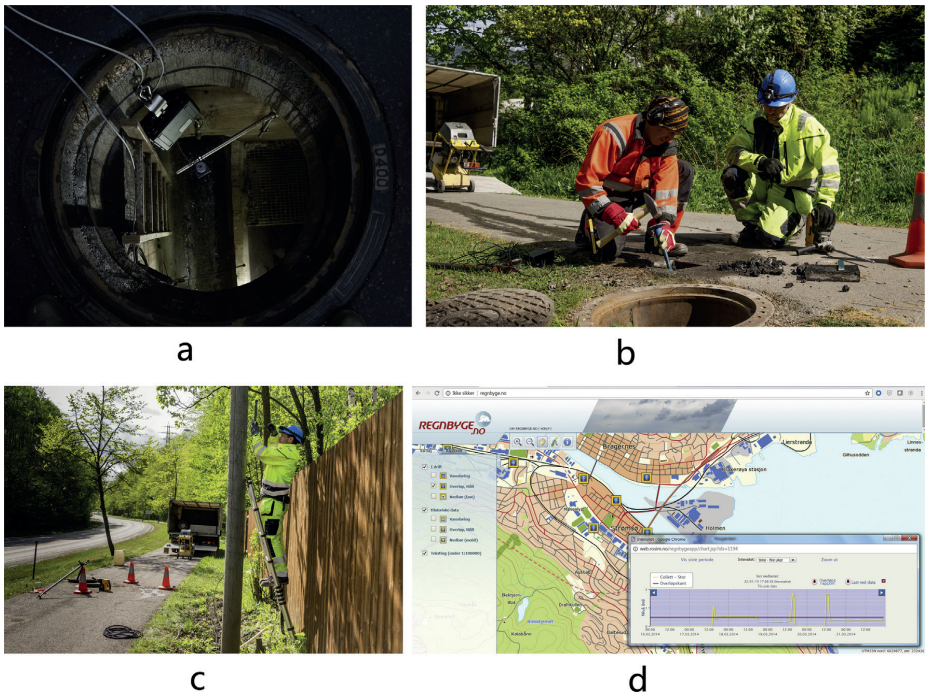


Fig. 2. Major components of the Regnbyge.no IoT; (a) an example of field-deployed water level sensor; (b) engineers are embedding antenna in the road surface; (c) engineer are installing wireless transmitter on a lamp post; (d) the user interface of the Regnbyge.no IoT.

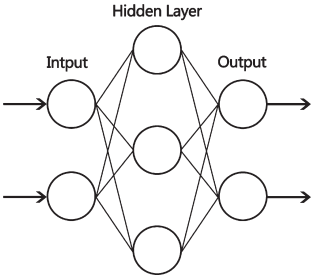


Fig. 3. Architecture of a three-layered MLP.

The partial derivative of the sigmoid activation function corresponding to the weighted sum input of the neuron s is:

$$\frac{\partial f(s)}{\partial s} = f(s)(1 - f(s)) \tag{6}$$

BP calculates the error contribution of each neuron, the cost function, i.e. error between predicted and observed value can be minimized through adjusting the weights of each neuron.

2.3.2. WNN

WT is a multi-resolution analysis in time and frequency domain. For example, for time series $f(t)$, WT is the sum over all time of $f(t)$ multiplied by the scale:

$$W_f(a,b) = |a|^{-1/2} \int_{-\infty}^{+\infty} f(t)\bar{\psi}\left(\frac{t-b}{a}\right)dt \tag{7}$$

where $\psi(t)$ is the mother wavelet, a is the parameter defining the window of analysis, b is the parameter localizing the wavelet function in the time domain, and $f(t)$ is the complex conjugate of the basic wavelet function. $W_f(a,b)$ represents the correlation between the signal $f(t)$ and a scaled version of the function $\psi(t)$.

Similar to MLP, the WNN also consists of an input layer, hidden layer, and output layer. The difference of WNN is its use mother wavelet as an activation function. The neuron of WNN often referred as wavelons, which transfer the input variables to the dilated and translated version of the mother wavelet. The mother wavelet used in this study is the Morlet mother wavelet. Morlet mother wavelet is a complex exponential with a Gaussian envelope that ensures localization. Fig. 4 shows the Morlet mother wavelet. The mathematical representation of the Morlet wavelet activation function is given as:

$$f(x) = \cos(1.75x) * e^{-x^2/2} \tag{8}$$

When training the WNN through BP, the partial derivation of the Morlet wavelet activation function can be written as:

$$\frac{\partial f(x)}{\partial x} = -1.75 * \sin(1.75x) * e^{-x^2/2} - x * \cos(1.75x) * e^{-x^2/2} \tag{9}$$

According to several recent studies (Majeed et al., 2017; Chitsaz et al., 2015; Xu & Liu, 2013; Alexandridis and Zapranis, 2013), Morlet wavelet ANN is a good alternative in cases with difficult nonlinear systems. The Morlet Wavelet ANN is reliable and robust, it can solve the problem with almost the same accuracy as other WNNs with the Mexican hat wavelet or Haar wavelet based activation function.

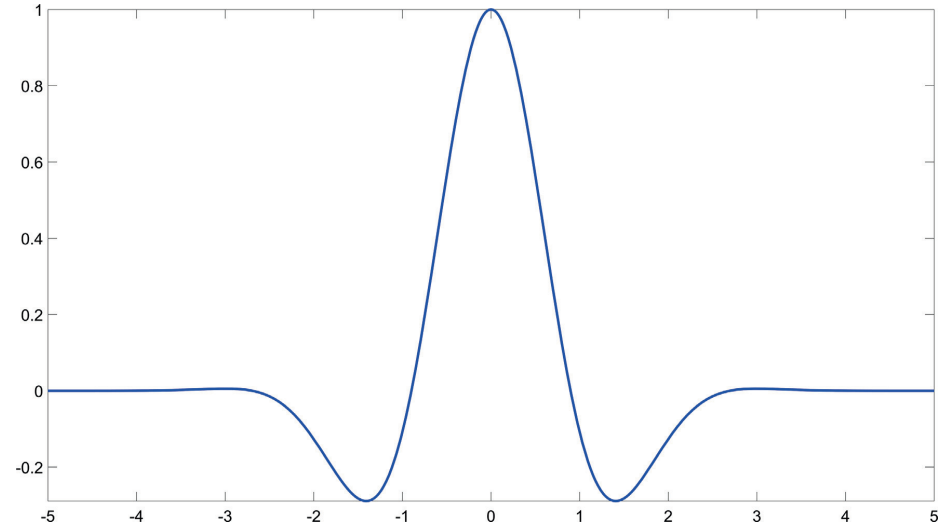


Fig. 4. The Morlet mother wavelet.

2.3.3. LSTM

Different from traditional ANN, RNN (Elman, 1990) takes the state of the hidden neuron at the previous time steps as an additional input for the next time step (Ishak et al., 2003).

As shown in Fig. 5, the neuron state of the present time step h_{t+1} is calculated by the equation:

$$h_{t+1} = w_{h+1}h_t + w_{i+1}x_{t+1} + b \tag{10}$$

Similar to h_{t+1}, the h_t is calculated by:

$$h_t = w_h h_{t-1} + w_i x_t + b \tag{11}$$

where h_{t+1}, h_t, and h_{t-1} are states of the hidden neuron at the time step t + 1, t, and t-1 respectively, w_{i+1}, w_i and w_{h+1}, w_h are weights between input values and hidden neurons (input weights), and between hidden neurons (hidden weights) respectively.

The training of RNN relies on an extended version of BP called backpropagation through time (BPTT). Unlike BP, BPTT not only calculates the gradient of the cost corresponding to the input weights but also the gradient of the cost corresponding to the hidden weights of the previous time steps. As shown in Fig. 5, the dashed line and arrow is the direction of gradient calculation of BPTT, BPTT first calculates the gradient of output at time step t + 1 (O_{t+1} in Fig. 5) corresponding to the state of hidden neuron at time step t + 1 (h_{t+1} in Fig. 5), i.e. $\frac{\partial O_{t+1}}{\partial h_{t+1}}$ in Fig. 5. It then calculates the gradient of the state of hidden neuron at time step t + 1 corresponding to the state of hidden neuron at previous time step ($\frac{\partial h_{t+1}}{\partial h_t}$ in Fig. 5), and backpropagation to earlier neurons step by step in this way. When using the BPTT method, with gradient calculation, the error of partial derivative accumulates through time steps. Hence, it will be extremely hard to learn and tune the parameters of the earlier neurons and learn long-term dependencies. Because the gradient going through the network either gets very small and vanish, or gets very large and explode, this problem is commonly known as the vanishing/exploding gradient problem. In recent years, modern RNN combat vanishing/exploding gradient, such as LSTM, had been proposed (Lipton et al., 2015; Gers 2001; Gers et al., 2000).

The LSTM replaced the ordinary neuron in the hidden layer with a memory cell and three gates: the input gate, forget gate and output gate. LSTM can selectively update the memory cell state based on the new input, forget irrelevant content, or selectively output part of the memory cell state as the new hidden neuron state according to the state of the input, forget and output gate respectively, similar to data in a computer's memory. In this way, the LSTM is able to learn long time span time series.

Fig. 6 shows the neuron in the hidden layer of LSTM, i, f and o represent the input, forget and output gate respectively. c and $\bar{c}$ denote the memory cell and the new memory cell. The principal

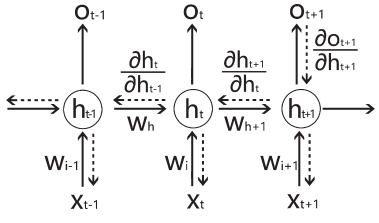


Fig. 5. The architecture of the RNN and schematic diagram of BPTT. The solid lines indicate how RNN inherits previous hidden neuron states. The dashed lines indicate the direction of BPTT.

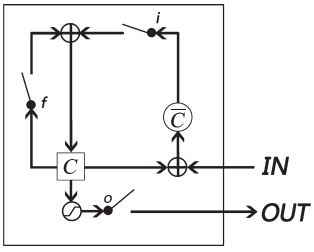


Fig. 6. Neuron in the hidden layer of LSTM.

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

Input gate:

$$i_t = \sigma_g(W_i * x_t + U_i * h_{t-1} + V_i \circ c_{t-1} + b_i) \quad (12)$$

Forget gate

$$f_t = \sigma_g(W_f * x_t + U_f * h_{t-1} + V_f \circ c_{t-1} + b_f) \quad (13)$$

Output gate:

$$o_t = \sigma_g(W_o * x_t + U_o * h_{t-1} + V_o \circ c_{t-1} + b_o) \quad (14)$$

Cell state:

$$c_t = f_t \circ c_{t-1} + i_t \circ c_t \quad (15)$$

$$c_t = \sigma_c(W_c * x_t + U_c * h_{t-1} + b_c) \quad (16)$$

Output vector:

$$h_t = o_t \circ \sigma_h(c_t) \quad (17)$$

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, σ_g is the sigmoid function, σ_h and σ_c are hyperbolic tangent function, for a given input z, the output of hyperbolic tangent function is:

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (18)$$

2.3.4. GRU

The concept of a GRU is quite similar to LSTM. The GRU also use gates to modulate the flow of information inside the neuron in the hidden layer. The difference is the GRU has two gates (reset gate and an update gate), while LSTM has three gates. GRU combines the input and forget gates into an update gate to balance between previous activation and the candidate activation. The activation of h at time t depends on h at the previous time and the candidate h (the $\bar{h}$ in Fig. 7). The update gate z decides how much of the previous memory to keep around. The GRU unit forgets the previously computed state when the reset gate r is off. (Cho et al., 2014; Chung et al., 2014).

The GRU is formulated as:

$$z_t = \sigma_g(W_z * x_t + U_z * h_{t-1} + b_z) \quad (19)$$

$$r_t = \sigma_g(W_r * x_t + U_r * h_{t-1} + b_r) \quad (20)$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \bar{h}_t \quad (21)$$

$$\bar{h}_t = \sigma_h(W_h * x_t + U_t * (r_t \circ h_{t-1}) + b_h) \quad (22)$$

where x_t is the input vector, h_t is the output vector, z_t is the update gate vector, h_t is the reset gate vector. W, U and b are parameters for weights and bias. $\circ$ represents the scalar product of two vectors, $\sigma(.)$ is the sigmoid function. σ_g represent the sigmoid activation function, σ_h represent the hyperbolic tangent activation function.

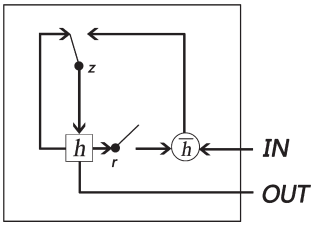


Fig. 7. Neuron in the hidden layer of GRU.

2.3.5. Model implementation

In this paper, the MLP and WNN were implemented using Matlab, R2016a. The LSTM and GRU were programmed using Keras (Chollet, 2015). Keras is a high-level deep learning library. It is written in Python and runs on top of either TensorFlow (Abadi et al., 2016) or Theano backend. TensorFlow backend was employed in this study. TensorFlow is an open-source software for deep learning, released by Google in 2015. Besides Keras and TensorFlow, other Python libraries such as matplotlib, numpy, pandas, sklearn were also used.

2.4. Model performance criteria

The performance of the developed models was evaluated by three criteria, the root mean square error (RMSE), Nash-Sutcliffe Efficiency (NSE) and the coefficient of determination (R^2).

The calculation of RMSE as shown below:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Y_i^{obs} - Y_i^{sim})^2}{n}} \quad (23)$$

RMSE value of 0 means a perfect fit between observed and predicted values.

NSE is a parameter that determines the relative importance of residual variance (noise) compare to the variance in the measured data (information). The NSE is calculated by the following equation:

$$NSE = 1 - \left[\frac{\sum_{i=1}^{n}(Y_i^{obs} - Y_i^{sim})^2}{\sum_{i=1}^{n}(Y_i^{obs} - Y^{mean})^2}\right] \quad (24)$$

NSE varies from $-\infty$ to 1, NSE = 1 indicates a perfect correlation between simulated and observed data, values between 0.0 and 1.0 is generally acceptable.

The equation for the coefficient of determination (R^2) is:

$$R^2 = \left[\frac{(\sum_{i=1}^{n}(Y_i^{sim} - Y_{sim}^{mean})(Y_i^{obs} - Y^{mean}))^2}{\sum_{i=1}^{n}(Y_i^{sim} - Y_{sim}^{mean})^2 \sum_{i=1}^{n}(Y_i^{obs} - Y^{mean})^2}\right] \quad (25)$$

The R^2 values range between 0 and 1, R^2 value of 1 indicates a perfect correlation.

In above-listed equations: nY_i^{obs} = the i-th observed data.Y_i^{sim} = the i-th simulated data.Y^{mean} = mean value of observed data.Y_{sim}^{mean} = mean value of simulated data.

n = number of data.

3. Results and discussion

3.1. Datasets for ANN training

According to CSO water level data of Dr_Hansteensgate and rainfall intensity data from the Fagerlibakken rain gauge recorded by the Regnbyge.no IoT during 2014, 26 CSO events were selected for constructing ANN models in this study. During a CSO event, the water level in the CSO structures rises due to rainfall in the associated catchment. The Fagerlibakken rain gauge was selected using the cross correlation function XCORR function in Matlab, R2016a. The collected time series data have 2352 records with a temporal resolution of 10 min. In consideration of maintaining similar statistical characteristics, 20 events were selected as training sets and the remaining 6 events were allocated as testing sets. The difference between training and testing is regularization mechanisms,

which is used to prevent overfitting, are turned off during the testing mode. Dropout and L2 weight penalty method were used as regularization methods in this study. Dropout is a technique that randomly drops selected neurons and set their associated weights to zero during training. L2 weight penalty method adds an extra squared term to the cost function to constrain the weights. It keeps the weights small unless they have big error derivatives. The summary statistics for water levels and rainfall are presented in Table 1.

3.2. Training of ANN

As the primary objective of this paper, to anticipate the occurrence of CSO events and develop early warning and supply references for necessary CSO reduction measures, different ANN models were developed. The purposes of the developed models are to predict the one- to six-step ahead water level in the CSO structure based on data collected by IoT. To make a fair comparison, the input of the developed ANN models remain the same i.e. data from the previous six steps collected by IoT. After confirming input data and prediction goals, the training of ANNs was implemented through trial-and-error procedures. While the best-performed MLP has one hidden layer with fifteen hidden neurons, the WNN used in this study has one hidden layer with eighteen wavelons. The optimal structure of LSTM and GRU have two hidden layers with ten hidden neurons in each layer. In this study, the Levenberg–Marquardt algorithm was used for training of MLP. The WNN was trained using the standard BP method. For LSTM and GRU, the ADAM (Kingma and Ba, 2014) method was selected as the optimization algorithm.

3.3. Performance of ANN

All the four ANN models performed rather consistently in the training stages while showing different performance during the

Table 1
Summary statistics for water levels (m) and the rainfalls (mm/s).

Event	Model stage	Max level (m)	Average level (m)	Standard deviation level	Max rainfall (mm/s)	Average rainfall (mm/s)	Standard deviation rainfall
1	Training	1.19	0.73	0.31	4.76	1.92	1.22
2		1.18	0.81	0.32	0.44	0.17	0.11
3		0.54	0.30	0.17	0.14	0.07	0.03
4		1.24	0.76	0.30	25.00	3.66	4.45
5		0.50	0.38	0.12	4.55	1.16	0.93
6		0.83	0.61	0.20	5.56	2.00	1.44
7		1.27	0.72	0.34	33.33	11.00	7.32
8		1.29	0.80	0.33	25.00	8.48	6.36
9		1.25	0.87	0.28	50.00	10.38	10.65
10		0.67	0.50	0.17	2.27	0.94	0.55
11		1.11	0.68	0.28	2.22	1.07	0.50
12		1.20	0.76	0.32	9.09	2.13	1.81
13		1.20	0.90	0.34	10.00	1.90	1.79
14		1.20	0.71	0.33	11.11	1.93	1.55
15		1.17	0.93	0.27	2.22	0.88	0.43
16		1.07	0.59	0.29	2.04	0.73	0.48
17		1.17	0.70	0.30	2.86	0.60	0.37
18		0.62	0.52	0.14	0.97	0.52	0.25
19		0.81	0.66	0.18	2.04	0.69	0.35
20		1.20	0.86	0.35	4.00	1.48	0.74
21	Testing	1.18	0.57	0.11	1.31	0.58	0.37
22		1.22	0.83	0.31	25.00	6.73	6.55
23		1.21	0.89	0.31	9.09	2.96	1.60
24		1.18	0.71	0.30	3.33	1.37	0.72
25		0.81	0.48	0.22	1.41	0.39	0.26
26		1.25	0.81	0.31	14.29	4.04	2.98

Table 2
Performance of different models for one- to six-step ahead CSO water level prediction within the testing stage.

Time steps		GRU	LSTM	WNN	MLP
6	RMSE	0.1536	0.1529	0.1694	0.2233
	R^2	0.7960	0.7987	0.7525	0.7660
	NSE	0.7807	0.7828	0.7332	0.5133
5	RMSE	0.1353	0.1398	0.1402	0.2115
	R^2	0.8313	0.8336	0.8041	0.8120
	NSE	0.8138	0.8186	0.8075	0.5616
4	RMSE	0.1188	0.1263	0.1270	0.2011
	R^2	0.8786	0.8653	0.8768	0.8694
	NSE	0.8693	0.8522	0.8506	0.6138
3	RMSE	0.1004	0.1130	0.1030	0.1780
	R^2	0.9133	0.8991	0.9334	0.8861
	NSE	0.8717	0.8627	0.8445	0.7061
2	RMSE	0.0820	0.1021	0.0887	0.1656
	R^2	0.9542	0.9357	0.9397	0.8967
	NSE	0.8967	0.8816	0.8432	0.7648
1	RMSE	0.0641	0.0720	0.0654	0.0918
	R^2	0.9892	0.9765	0.9711	0.9531
	NSE	0.9267	0.9266	0.9319	0.9197

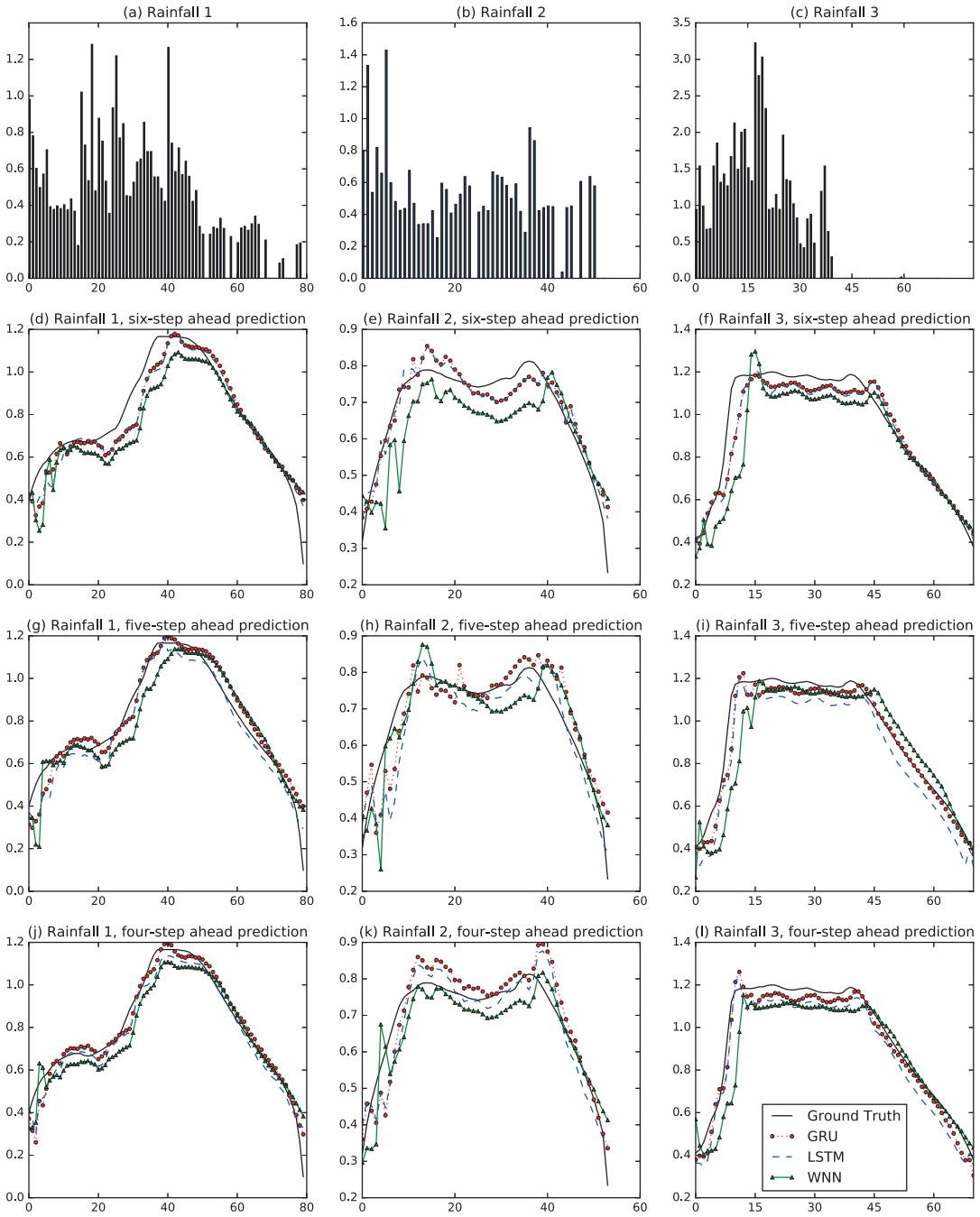


Fig. 8. Four- to six-step ahead forecasts of selected rainfall events with respect to the LSTM, GRU, and WNN.

testing stages. The results for the one- to six-step ahead CSO water level forecasting within the testing stages are provided in Table 2.

As shown in Table 2, the model prediction accuracy reduced as the prediction steps increased. For the one-step-ahead prediction, all the ANN models showed satisfying results, with low RMSE value, high NSE value, and high R^2 value. When turning the prediction time step to two, the performance of MLP deteriorated immediately. Compared to the one-step-ahead prediction, the RMSE

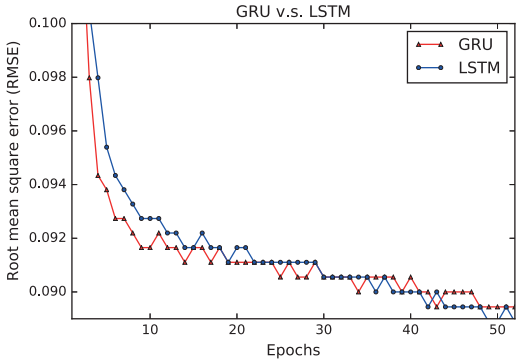


Fig. 9. Learning curves of LSTM and GRU.

value of MLP for the two-step ahead prediction almost doubled, increasing from 0.0918 to 0.1656, the R^2 value and NSE values reduced from 0.9531 and 0.9197 to 0.8967 and 0.7648 respectively. With longer time steps, the MLP performance decreased continually. It reveals that as a static feedforward ANN, the MLP is not suitable for multi-step-ahead forecasting. The WNN presented better performance than MLP. For three-step ahead prediction, the WNN even got the highest R^2 value, 0.9334. However, this fact alone doesn't prove that the performance of WNN is better than LSTM and GRU when comprehensively considering all the performance criteria. Because the remaining values of RMSE, NSE, and R^2 of WNN are very unstable across the different time steps compared to LSTM and GRU.

When the prediction time step exceeds four, LSTM and GRU outperform MLP and WNN in terms of all the performance criteria. LSTM and GRU even produce low RMSE value (around 0.15), high NSE value (around 0.8) and high R^2 value (around 0.8) as the prediction time step reaches six. Besides, LSTM shows a slightly better performance than GRU, but the difference is marginal.

To further illustrate the performance of LSTM, GRU, and WNN in a more intuitive way, hydrographs of the observed versus four-, five- and six-step ahead predicted water level for three rainfall events in the testing sets were drawn in Fig. 8.

For six-step ahead prediction (Fig. 8(d)–(f)), when the CSO water level starts to increase, the LSTM and GRU react quicker than WNN, whereas the WNN presents the time-lag problem. Moreover, it obviously shows that LSTM and GRU better predict the peak water level, while the WNN either underestimates or overestimates the peak value. The inconsistency of WNN indicates its poor performance compared to LSTM and GRU. Similar observations can also be found in the five- and four-step ahead prediction (Fig. 8(g)–(i)). Only in Fig. 8(k) does the WNN look more stable than LSTM and GRU.

As we can see from Fig. 8, in general, the LSTM and GRU can better capture the sudden change of water levels compared to WNN, and significantly alleviate the time-lag problem at the peak value. In terms of peak water level, the WNN either underestimates or undergoes strong fluctuations at the peak value. This is probably due to the insufficient learning capability of WNN. Predictions by both LSTM and GRU showed very similar behavior, especially for the six-step ahead prediction. However, the GRU has several advantages compared to LSTM.

Fig. 9 displays the learning curves of LSTM and GRU. In Fig. 9, the x-axis is epochs (one full training cycle on the training samples) and the y-axis is the RMSE value. The GRU makes faster progress than LSTM during training, the RMSE value of the GRU is lower than LSTM during the first 30 epochs, but the final RMSE value of the GRU is slightly higher than LSTM. The quick learning curve of the GRU is due to its simplified architecture. For large-scale data, the model training can consume a significant amount of CPU time. Hence, the GRU's quicker convergence and comparable performance is a good alternative to LSTM. On the other hand, the GRU has fewer parameters. To make predictions in IoT, the controller has to calculate the future water level based on real-time water level data read from the sensor and the parameters of the prediction model. With fewer parameters, the hardware implementation and the real-time computing process of controllers could be simplified and accelerated.

4. Conclusion

In order to reduce pollutions caused by CSO, it is essential to predict the CSO event in advance. The prediction could enable better decision-making, greater automation, and higher efficiencies. Through a case study in Drammen, Norway, the present work compared the performance of four ANN on the multi-step-ahead prediction of the CSO water level collected by IoT. The preliminary conclusions summarized below.

The MLP only presented accurate predictions for the one- and two-step ahead prediction, but not for multi-step-ahead predictions. WNN can improve the multi-step ahead predictions. However, when checking the hydrography of observed and predicted water level, the WNN presented several problems such as time delay and strong fluctuation. With the gating mechanisms, the LSTM and GRU outperformed WNN and MLP. Both LSTM and GRU achieved satisfying results for multi-step-ahead prediction. The behavior of LSTM and GRU is quite similar. LSTM had marginally better performance than GRU, but the advantages of GRU are the quicker learning curve, fewer parameters, and simpler architecture. For practical problems with very large datasets, the GRU provides a trade-off between accuracy and training time.

IoT will become the backbone of urban success, devices are integrated together in order to enhance key 'smart' sectors such as transport, energy, water supply and waste treatment. On the other hand, advanced AI techniques such as the LSTM have shown their power of optimizing IoT. There are many possible scenarios to extend IoT for purposes of environmental monitoring and modeling applications.

The potential power of deep learning is fascinating, and it is worth to conduct further studies about adapting deep learning into water resource related fields. Studies about the performance of other deep learning methods such as deep belief nets, stacked deep encode and generative adversarial networks in water resource related fields, are other interesting research directions in the future.

Acknowledgements

This work has been supported by the Regnbyge-3M project (grant number 234974), which is granted by the Oslofjord Regional Research Fund. The authors would like to thank the engineers from Rosim AS for their supports.

References

Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., et al., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:160304467.

Abghari H., Nouri M., and Rezaeian Zadeh M., 2009. Reservoir dam inflow modeling using wavelet-base neural network and multilayer perceptron network. International Conference on Water Resources, Shahrood, Iran.

Abghari, H., Ahmadi, H., Besharat, S., Rezaverdinejad, V., 2012. Prediction of daily pan evaporation using wavelet neural networks. Water Resour. Manage. 26 (12), 3639–3652.

Alam, F., Mehmood, R., Katib, I., Albeshri, A., 2016. Analysis of eight data mining algorithms for smarter Internet of Things (IoT). Proc. Comput. Sci. 98, 437–442.
Alexandridis, A.K., Zapranis, A.D., 2013. Wavelet neural networks: a practical guide. Neural Networks 42, 1–27.
Atzori, L., Iera, A., Morabito, G., 2010. The internet of things: a survey. Comput. Networks 54 (15), 2787–2805.
Autixier, L., Mailhot, A., Bolduc, S., Madoux-Humery, A.S., 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.
Barzegar, R., Fijani, E., Moghaddam, A.A., Tziritis, E., 2017. Forecasting of groundwater level fluctuations using ensemble hybrid multi-wavelet neural network-based models. Sci. Total Environ. 599, 20–31.
Chang, F.J., Chen, P.A., Lu, Y.R., Huang, E., Chang, K.Y., 2014. Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. J. Hydrol. 517, 836–846.
Chen, Y., Yang, B., Dong, J., 2006. Time-series prediction using a local linear wavelet neural network. Neurocomputing 69 (4), 449–465.
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.
Chollet F., 2015. Keras: Deep learning library for theano and tensorflow.
Chung, J., Gulcehre, C., Cho, K., and Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.
Darsono, S., Labadie, J.W., 2007. Neural-optimal control algorithm for real-time regulation of in-line storage in combined sewer systems. Environ. Model. Software 22 (9), 1349–1361.
Elman, J.L., 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. Network Comput. Appl. 78, 30–42.
Gers, F., 2001. Long short-term memory in recurrent neural networks, (Dissertation), Universität Hannover.
Gers, F.A., Schmidhuber, J., Cummins, F., 2000. Learning to forget: continual prediction with LSTM. Neural Comput. 12 (10), 2451–2471.
Giusto, D., 2010. A. Lera, G. Morabito, l. Atzori (Eds.) The Internet of Things. Springer, 2010. ISBN: 978-1-4419-1673-0.
Google, 2016. https://research.googleblog.com/2016/09/a-neural-network-for-machine.html. (Accessed 27 April 2017).
Grum, M., Thornberg, D., Christensen, M. L., Shididi, S. A., and 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, G.E., Osindero, S., Teh, Y.W., 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.
Ishak, S., Kotha, P., Alecsandru, C., 2003. Optimization of dynamic neural network performance for short-term traffic prediction. Trans. Res. Rec.: J. Trans. Res. Board 1836, 45–56.
Kingma, D., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
Kurth, A., Saul, A., Mounce, S., Shepherd, W., and Hanson, D., 2008. Application of Artificial Neural Networks (ANNs) for the prediction of CSO discharges. In: 11th International Conference on Urban Drainage.
Lipton, Z. C., Berkowitz, J., and Elkan, C., 2015. A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.
Lucas, W.C., Sample, D.J., 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. Trans. Res. Part C: Emerg. Technol. 54, 187–197.
Majeed, K., Masood, Z., Samar, R., Raja, M.A.Z., 2017. A genetic algorithm optimized morlet wavelet artificial neural network to study the dynamics of nonlinear troech's system. Appl. Soft Comput.
Marçais, J., de Dreuzy, J.R., 2017. Prospective interest of deep learning for hydrological inference. *Groundwater*.
Montserrat, A., Bosch, L., Kiser, M.A., Poch, M., Corominas, L., 2015. Using data from monitoring combined sewer overflows to assess, improve, and maintain combined sewer systems. Sci. Total Environ. 505, 1053–1061.
Mounce, S.R., Shepherd, W., Sailor, G., Shucksmith, J., Saul, A.J., 2014. Predicting combined sewer overflows chamber depth using artificial neural networks with rainfall radar data. Water Sci. Technol. 69 (6), 1326–1333.
Nourani, V., Baghanam, A.H., Adamowski, J., Kisi, O., 2014. Applications of hybrid wavelet–Artificial Intelligence models in hydrology: a review. J. Hydrol. 514, 358–377.
Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Dieleman, S., 2016. Mastering the game of Go with deep neural networks and tree search. Nature 529 (7587), 484–489.
Tang, D., Qin, B., Liu, T., 2015. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In EMNLP, pp. 1422–1432.
Wang, W., Ding, J., 2003. Wavelet network model and its application to the prediction of hydrology. Nat. Sci. 1 (1), 67–71.
Wu, Z., King, S., 2016. Investigating gated recurrent networks for speech synthesis. In: Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, pp. 5140–5144. IEEE.
Xingjian, S.H.I., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C., 2015. Convolutional LSTM network: a machine learning approach for precipitation nowcasting. Adv. Neural Inf. Proc. Syst., 802–810
Zaytar, M.A., Amrani, C.E., 2016. Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. Int. J. Comput. Appl. 143, 7–11.
Zhang, M., Zhang, Y., Vo, D.T., 2016. Gated Neural Networks for Targeted Sentiment Analysis. In AAAI pp. 3087–3093.
Xu, L., Liu, S., 2013. Study of short-term water quality prediction model based on wavelet neural network. Math. Comput. Model. 58 (3), 807–813.
Chitsaz, H., Amjady, N., Zareipour, H., 2015. Wind power forecast using wavelet neural network trained by improved Clonal selection algorithm. Energy Convers. Manage. 89, 588–598.

Paper II

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.* doi:10.1007/s11269-018-1919-3.

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

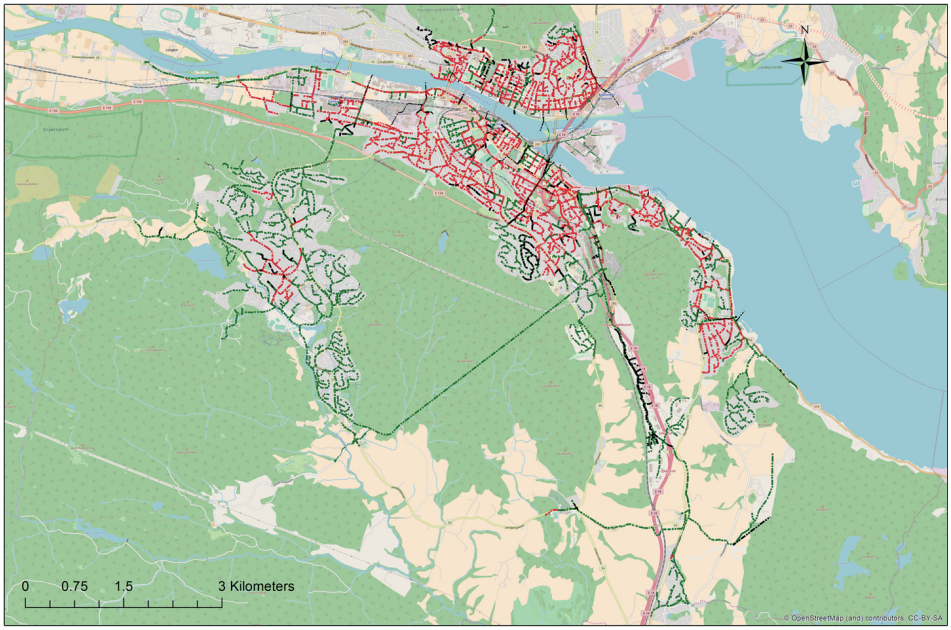


Fig. 1 Overview of Drammen city, Norway

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:

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

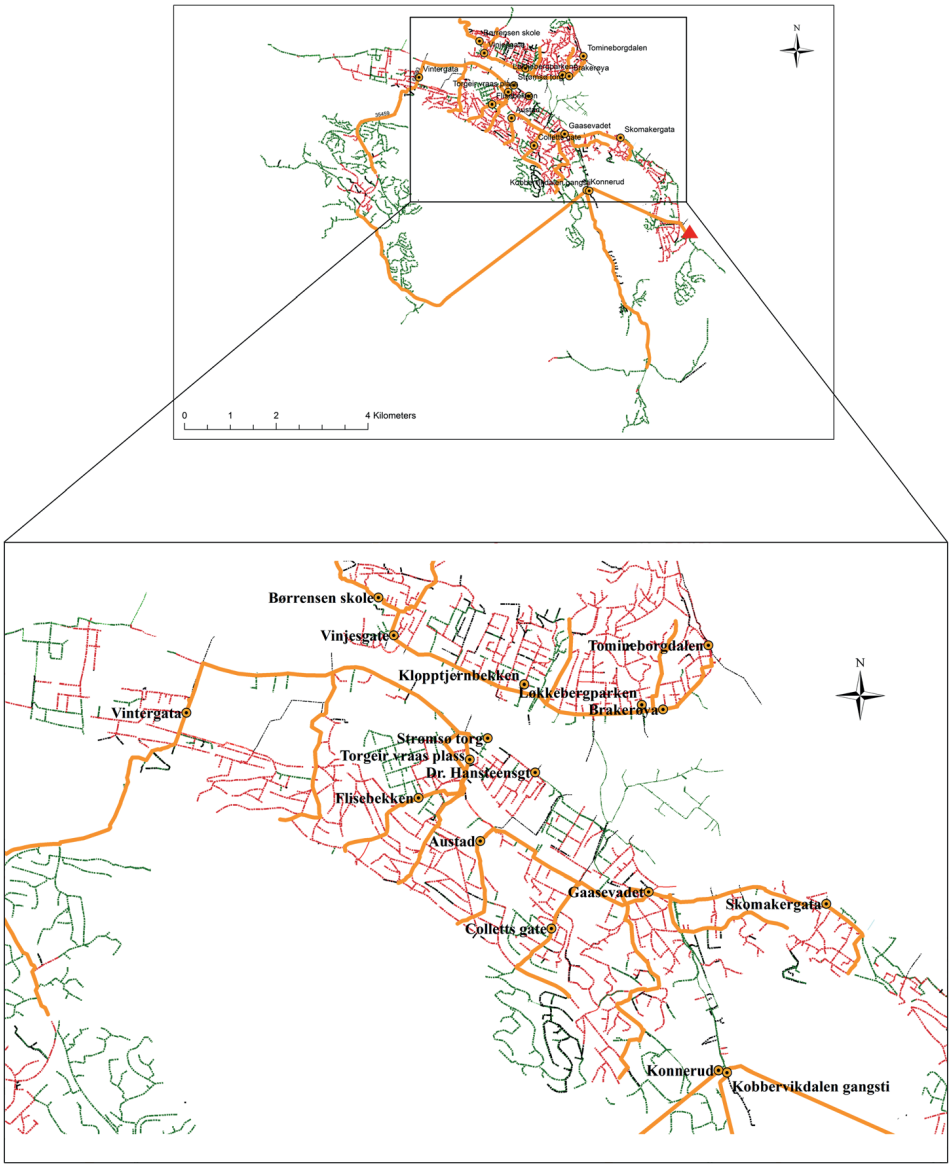


Fig. 2 Large pipes and monitoring sites of the Drammen sewer system

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

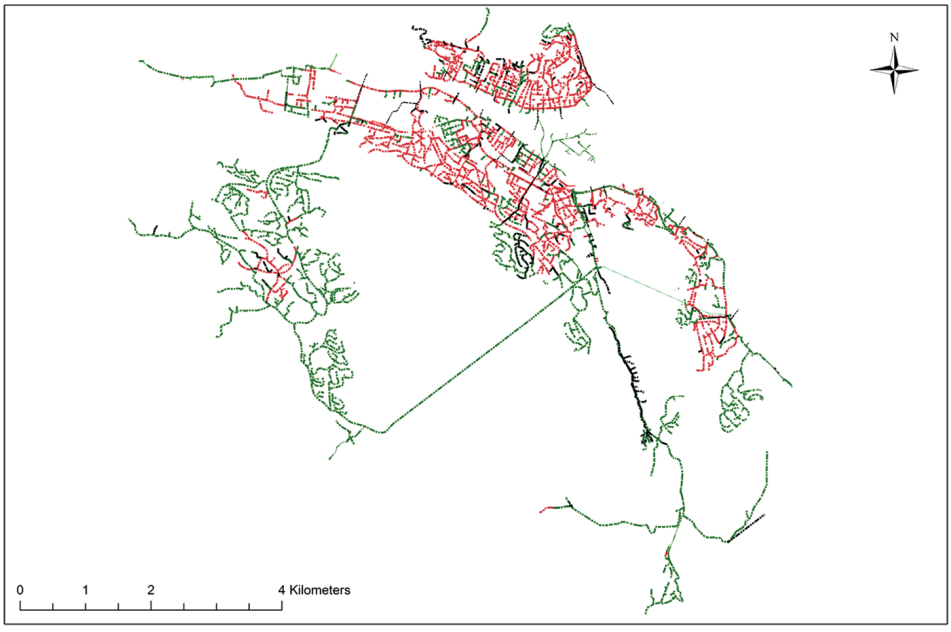


Fig. 3 Hydraulic model for the Drammen sewer system

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:

$$output(t+1) = f\begin{bmatrix} output(t-1), output(t-2), \ldots, output(t+1-q); \\ input(t-k), input(t-k-1), \ldots, input(t-k-p+1) \end{bmatrix}$$

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 ($k \geqslant 0$) 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:

$$output(t+1) = f\begin{bmatrix} actual(t), actual(t-1), \ldots, actual(t+1-q); \\ input(t-k), input(t-k-1), \ldots, input(t-k-p+1) \end{bmatrix}$$

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:

$$output(t+1) = f\begin{bmatrix} \widehat{actual}(t), \widehat{actual}(t-1), \ldots, \widehat{actual}(t+1-q); \\ input(t-k), input(t-k-1), \ldots, input(t-k-p+1) \end{bmatrix}$$

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.

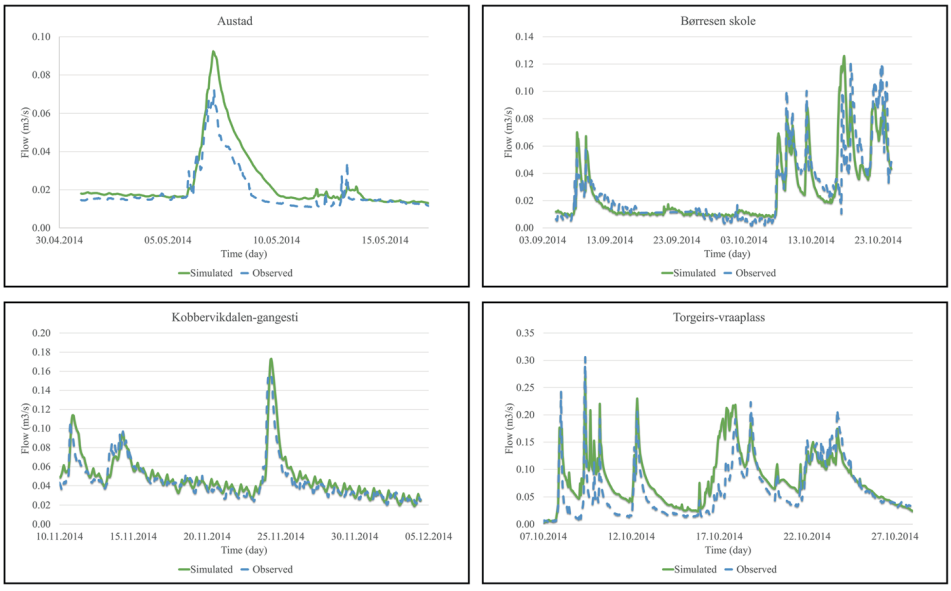


Fig. 4 Model calibration

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,

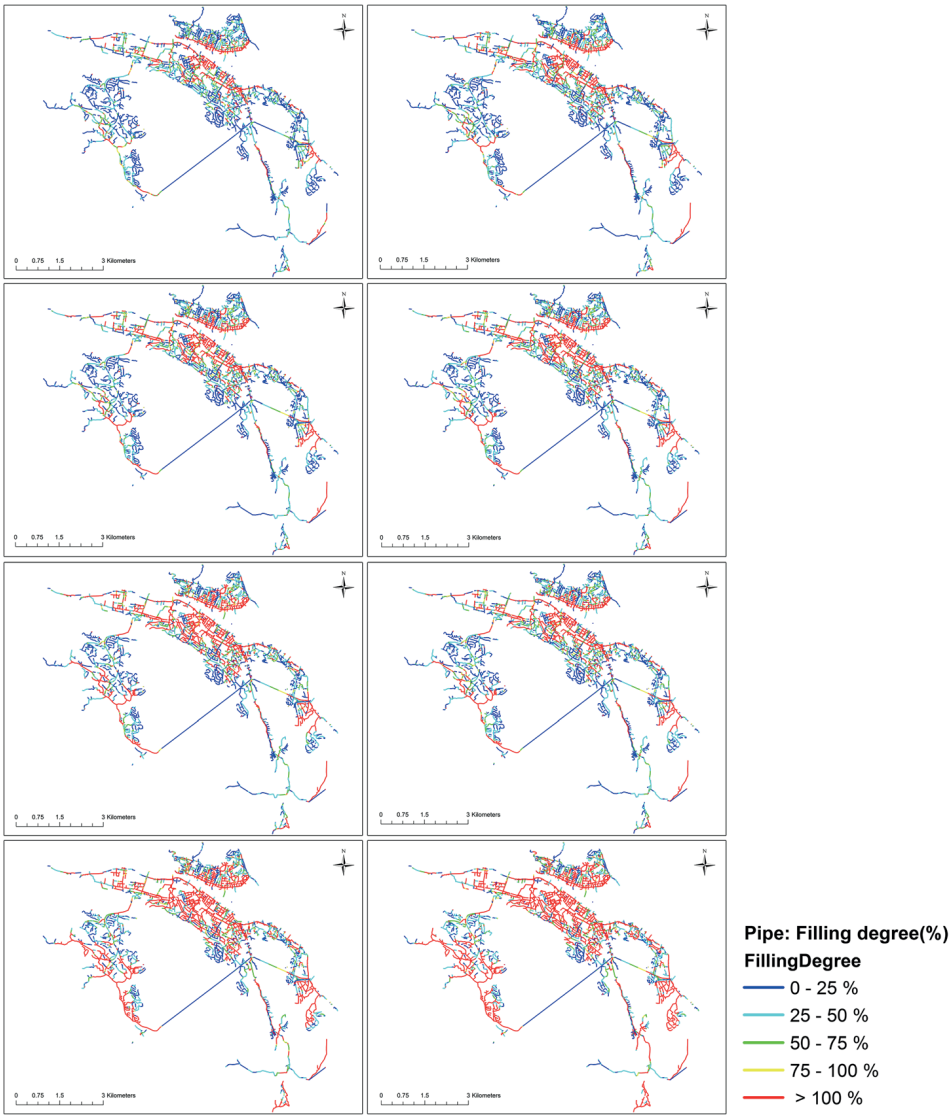


Fig. 5 Simulated filling degree under different return periods

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

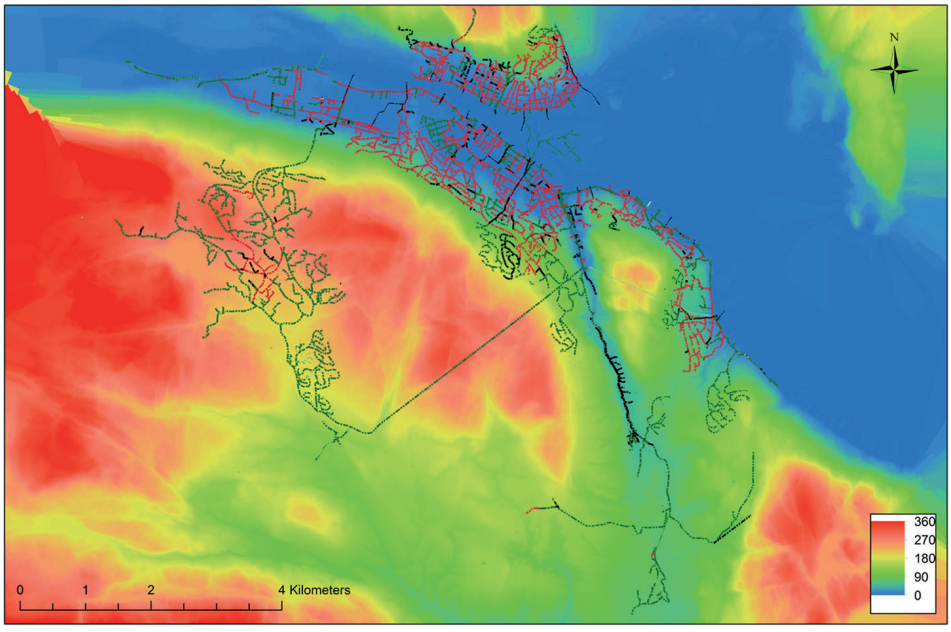


Fig. 6 DEM of Drammen

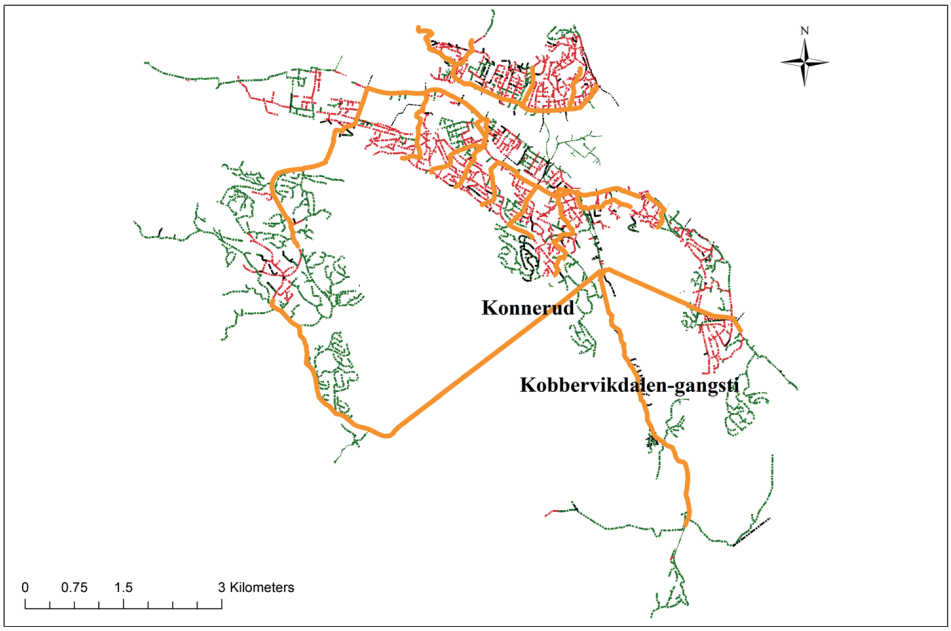


Fig. 7 Locations of Konnerud and Kobbervikdalen-gangsti tunnel

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

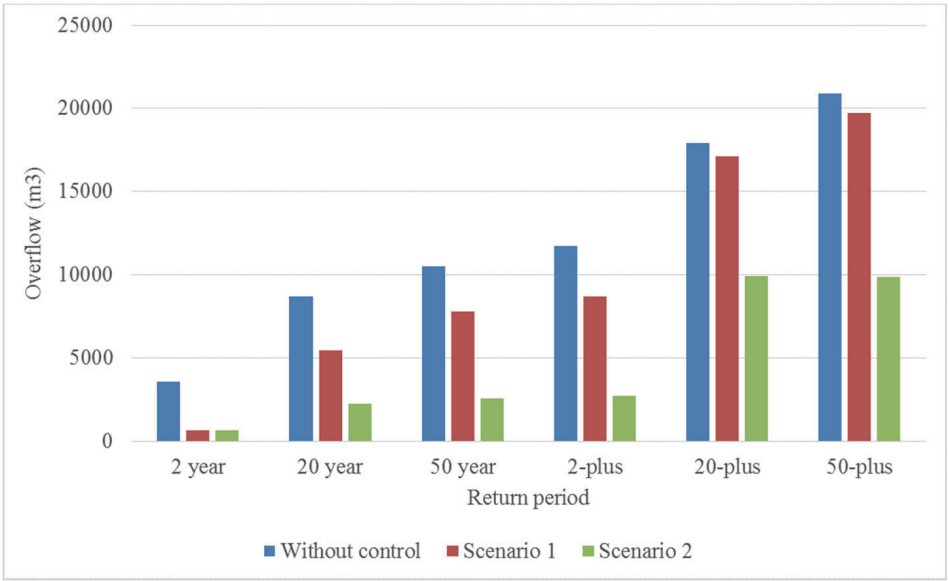


Fig. 8 Overflow at the Solumstrand WWTP under different control scenarios and return periods

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

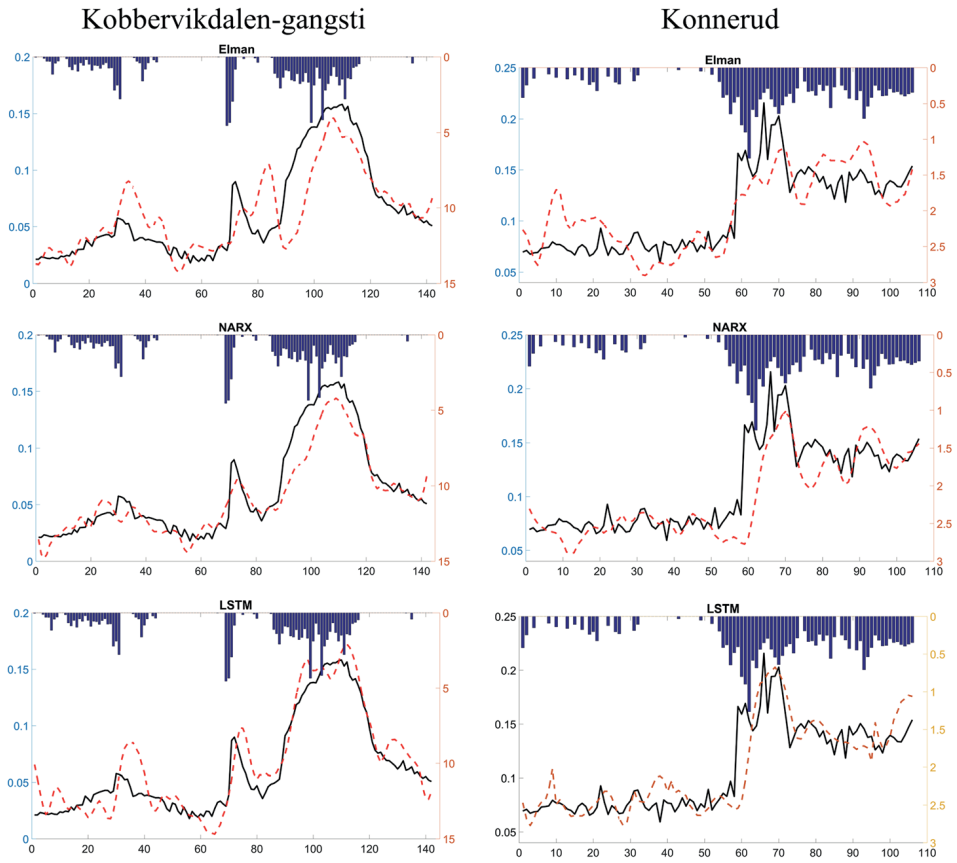


Fig. 9 RNN hydrograph

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

Paper III

Zhang, D., Hølland, E. S., Lindholm, G., & Ratnaweera, H. (2018). Hydraulic modeling and deep learning based flow forecasting for optimizing inter-catchment wastewater transfer. *Journal of Hydrology, 567, 792-802.* doi:10.1016/j.jhydrol.2017.11.029.

Journal of Hydrology 567 (2018) 792–802

Contents lists available at ScienceDirect

Journal of Hydrology

journal homepage: www.elsevier.com/locate/jhydrol

Research papers

Hydraulic modeling and deep learning based flow forecasting for optimizing inter catchment wastewater transfer

Duo Zhang [a,*], Erlend Skullestad Hølland [a], Geir Lindholm [b], Harsha Ratnaweera [a]

[a] Faculty of Sciences and Technology, Norwegian University of Life Sciences, 1432, Ås, Norway
[b] Rosim AS, Brobekkveien 80, 0582 Oslo, Norway

ARTICLE INFO

Article history:
Received 24 July 2017
Received in revised form 16 November 2017
Accepted 17 November 2017
Available online 21 November 2017
This manuscript was handled by Andras Bardossy, Editor-in-Chief

Keywords:
Inter Catchment Wastewater Transfer
Sewer overflow
Long Short-Term Memory
Deep learning

ABSTRACT

Sewer overflow is a priority concern for many cities. Owing to the high density of buildings and populations, constructing new storage tanks in urban areas is getting more and more difficult. Therefore, this paper proposes a novel Inter Catchment Wastewater Transfer (ICWT) method for sewer overflow mitigation. The ICWT method aims at redistributing the spatial mismatched sewer flow according to the treatment capacity of Wastewater Treatment Plant (WWTP). Three tasks are involved in the development of ICWT. First, test the effectiveness of ICWT. Second, study how to properly redistribute inflow to the WWTP. Third, the operation of ICWT highly depends on inflow to the WWTP, to support the management of ICWT, it is imperative to construct a flow rate prediction model. A hydraulic model is used for task 1 and 2, individual and combined effects of the storage tank and ICWT on sewer overflow are investigated. The simulation results show that less overflow is obtained. In considering the deficiencies of hydraulic model, one of the most promising deep learning techniques, Long Short-Term Memory (LSTM), is employed to undertake task 3. Experiments demonstrated that LSTM could be of great use in predicting sewer flow.

1. Introduction

Sewer overflow refers to a condition in which untreated sewage is discharged directly into the environment from sewer outlets or bypass structures of the wastewater treatment plant (WWTP), when the sewage volume exceeds the capacity of the sewer system or WWTP. More frequent sewer overflow can be expected with the increased impermeable surface, extreme rainfall events and urbanization. Consequently, better ways of mitigating sewer overflow will be required.

Storage units such as retention basins or storage tanks are the most straightforward solution to reduce sewer overflow. However, constructing new storage units in metropolitan cities can be a complicated and costly task. The locations and sizes of storage units are limited by densely populated urban space and need to be selected appropriately. Constructing storage units require significant investment. Besides, the disruption to residences and businesses due to construction works cause social and economic cost (Ganora et al., 2017; Ngo et al., 2016). The drawbacks of constructing new storage units have motivated the research for cost-effective methods for mitigating sewer overflow. These alternative approaches mainly focus on aggregating the spatial and temporal dynamic free space of the sewer system, thus maximizing the capacity of the sewer system with minimum construction works.

Several studies have been conducted to explore the in-line storage capacities of sewer systems rather than construct new storage facilities (Darsono et al., 2007; Grum et al., 2011; Garofalo et al., 2017). The in-line storage control solution aims at exploiting the full storage capacity of the sewer system through regulating gates, pumps, and weirs. For example, in a less overloaded conduit, a moveable weir could accumulate storm water temporarily by increasing the weir height during a rainfall event. This approach could efficiently reduce infrastructure investments.

Similarly, sewer overflow can be mitigated by operating existing storage tanks proactively. Lee et al. (2017) investigated a novel pump operating method for the storage tank on urban flood control in Daerim sewer network, Seoul, Korea. The operating method considers both the storage tank and sewer conduit, it implements two monitoring nodes, one in the storage tank and the other one in sewer conduit. The pumps in a storage tank will proactively discharge reserved volume until the water level of the conduit rises up to the maximum level. This method lets the existed storage tank accommodate to unprecedented flow rates and reduce flood successfully.

* Corresponding author.
E-mail addresses: Duo.Zhang@nmbu.no (D. Zhang), geir@rosim.no (G. Lindholm), Harsha.Ratnaweera@nmbu.no (H. Ratnaweera).

https://doi.org/10.1016/j.jhydrol.2017.11.029
0022-1694/

Owing to the high density of urbanized areas in modern cities, it is difficult to construct storage units or separate sewer systems. In some developed cities, deep tunnel systems have been used as an alternative solution. The deep tunnel could temporally transport storm water and sewage during heavy rainfall events, thereby reducing downstream peak flows. In a recent case study in Guangzhou city, China, engineers found that the deep tunnel could improve the flood control and drainage capacities of the basin successfully (Wu et al., 2016).

It can be summarized from the above literature that recently proposed methods for improving sewer management mainly emphasizes on intelligent control of existing sewer devices, maximum utilization of sewer storage capacity or exploits the underground space. Summarizing previous studies, a novel inter catchment wastewater transfer (ICWT) method is proposed in this paper.

The idea of ICWT is inspired by the concept of inter-basin water transfer (IBWT). IBWT refers to transfer water from basins having sufficient water (donor basin) to basins facing water shortages (receiving basin). The purpose of IBWT is to counterpoise the uneven distribution of water resources and imbalanced water demand in different basins. IBWT could create a win-win situation by utilizing the differences of flow regime in different basins (Wang et al., 2015; Yevjevich, 2001). Since the 20 th century, in order to meet growing residential, commercial and agricultural demand, many large-scale IBWT projects have been proposed, such as the South-North Water Transfer Project, which aims to transfer 44.8 billion cubic meters of water annually from the Yangtze River in southern China to the north.

This study generalizes the concept of IBWT to allocate inflow to WWTPs in different catchments. The ICWT method takes both the sewer systems and WWTPs into consideration. Most developed cities have more than one WWTPs (Vrebos et al., 2014), each of WWTPs receives sewage collected from the sewer systems of the associated catchments, these sewer systems and catchments have different behaviors against rainfall events, the WWTPs also have different treatment capabilities. A catchment with a more overflowed WWTP can be regarded as the donor basin. On the contrary, a catchment with a less overflowed WWTP can be regarded as the receiving basin. The essence of ICWT is to address the spatial mismatch between sewer flow and WWTP treatment capacities in different catchments.

The development of the proposed ICWT solution is a complex task. In general, there are three tasks in this study. First, the effectiveness of ICWT method should be tested and compared with traditional solutions such as construct new storage tank (task 1). Second, control rules should be derived to ensure adequate redistribution of the sewage stream (task 2). Third, the operation of ICWT involves a series of sewer control structures and highly depends on the flow information of its associated catchment, it is imperative to forecast inflow to the WWTP, thus providing sufficient response time and enhancing ICWT control (task 3) (Liu et al., 2016; Chen et al., 2014; Duchesne et al., 2001).

In this study, the hydraulic model is used for task 1 and 2. Hydraulic models are the most common tools in terms of design and test sewer control (Autixier et al., 2014; Lucas et al., 2015; Seggelke et al., 2005). Hydraulic models can supply high-quality information about the sewer system's behavior and give insight into the effects of different control strategies (Chiang et al., 2010). So that the output of hydraulic models is suitable for hydraulic planning and design. However, hydraulic models require complete knowledge of sewer systems and rainfall patterns; it can only provide sewer hydraulic information based on current or previous rainfall events. Additionally, the calibration of hydraulic models requires adjusting many parameters. Carrying out simulations by using hydraulic models demands manual operation. For a large sewer system, the hydraulic model also needs long computational time. These features make hydraulic models limited adequate for application in task 3 (El-Din et al., 2002).

To predict sewer flow and enable real-time operation (task 3), deep learning is among the top methods. Unlike hydraulic models, which adopt hydrological and hydraulic principles, deep learning avoids sophisticated theories by making data-driven predictions through learning from data. As a breakthrough of artificial neural network, deep learning has solved problems that have resisted artificial intelligence for decades. One exciting news of deep learning is Google DeepMind's AlphaGo (Silver et al., 2016) beat the world's best human Go game player Ke Jie in 2017. A deep learning technique that has made revolutionary strides in the sequential data modeling domain is the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). The latest Google translation system powered by LSTM has vastly improved the translation quality, brought service nearly to the level of human translators (Google, 2016). The information contained in language is a kind of sequential data, when undertaking translation tasks, the correct word to use depends on the context, and the other words in the sentence or even paragraph. Time series data such as the inflow to the WWTP is also a kind of typical sequential data. LSTM has shown its superior performance for prediction of traffic time series (Ma et al., 2015), and has been adopted by Uber for extreme event ride requests forecasting (Laptev et al., 2017). However, to the best of the author's knowledge, there are very few studies about the application of LSTM in the water recourse related domain, given the success of the LSTM reported by both academia and industry, the effectiveness of LSTM need to be investigated. In this paper, LSTM was employed to construct a model for WWTP inflow prediction to support the management of ICWT.

Combining the advantages of hydraulic models and machine learning/deep learning methods is a trend in nowadays research (Yu et al., 2013; Chiang et al., 2010; Darsono et al., 2007). The present work aims at describing a systematic sewer system management approach using ICWT in order to improve the whole system behavior with minimal construction, and looking at the efficiency of LSTM on flow prediction. The remainder of this paper is organized as follows: the first part is a general description about study area, hydraulic model and LSTM. The second part is simulation results for different control scenarios based on the hydraulic model. The third part presents the prediction efficiency of the LSTM. Conclusion and future envision are discussed at the end of this paper.

2. Method and materials

2.1. Description of case study area and the proposed ICWT scheme

Drammen is located in the southeast of Norway. With more than 150,000 inhabitants, Drammen is the fourth largest city in Norway, and the largest city and capital in Buskerud County. The catchment area of Drammen's sewer system is around 15 km^2 and the total length of the sewer system is approximately 500 km. There are two WWTPs in Drammen, the Muusøya WWTP and the Solumstrand WWTP. Fig. 1 is an overview of the Drammen city.

The traditional city center distributes along the Drammen Fjord. There are two major catchments on the north bank of the Drammen Fjord, the Muusøya catchment and the Bragernes catchment, the curve in Fig.1 (a) indicating the boundary of these two catchments. The Muusøya WWTP treat sewage collected from the Muusøya catchment (Fig.1 (b)). Wastewater aggregated from the Bragernes catchment is transported by the Søren Lemmich pump station (Fig.1 (c)) from the north bank to the Strømsø catchment,

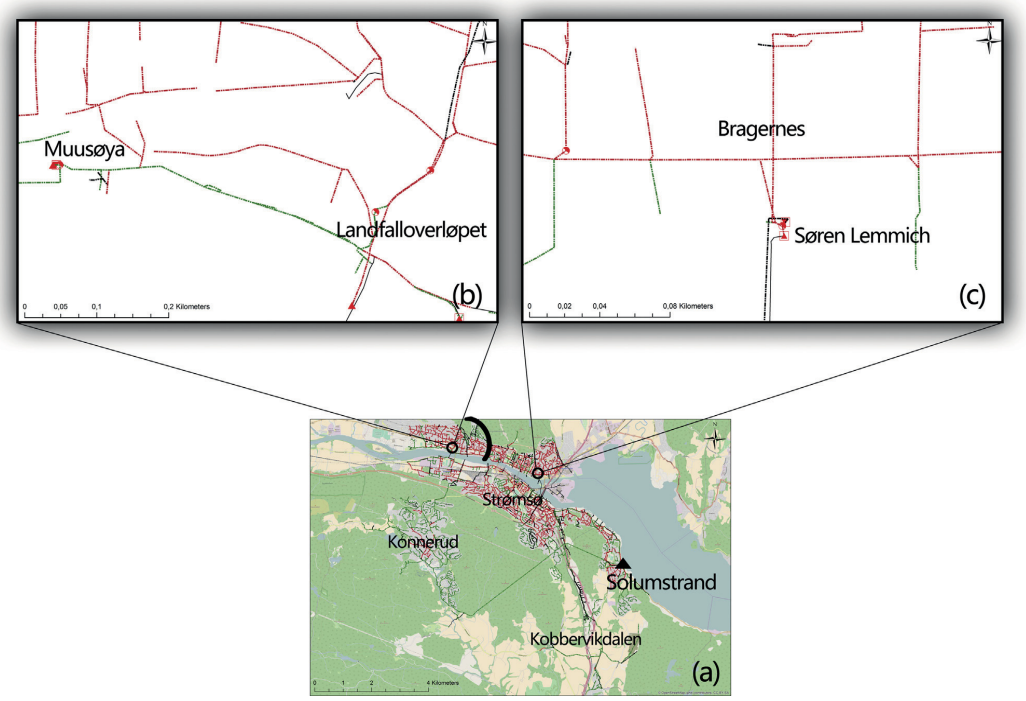


Fig. 1. Overview of Drammen, Muusøya WWTP and Søren Lemmich pump station.

which is on the south bank of Drammen Fjord. Afterward, sewage stream merges with wastewater collected from the Konnerud catchment and the Kobbervikdalen catchment, and finally discharged to the Solumstrand WWTP.

To reduce overflow from the sewer system and the WWTPs, the government of the Drammen city launched an innovative project: Regnbyge 3 M. The ultimate goal of the Regnbyge 3 M project is to integrates intelligent monitoring, modeling and control solutions, manage sewer system and WWTP in a holistic way.

The capacities of the sewer system and WWTPs are unevenly distributed in different catchments of Drammen. More than 80% of sewer system in the Muusøya catchment are combined sewer, while combined sewer in the sewer system associated with Solumstrand WWTP only accounts for less than 50%. For Muusøya WWTP, the designed treatment capacity is 33,000 PE (population equivalents), the dimensioning flow (Qdim) is 780 m^3/h, and the maximum flow (Qmax) is 1200 m^3/h. The Solumstrand WWTP is the major WWTP in Drammen, with a designed treatment capacity of 130,000 PE, the Qdim and Qmax is 2000 m^3/h and 4000 m^3/h respectively. Moreover, as a part of the traditional city center, the Muusøya catchment has a denser population than the rest of catchments. Because of combined sewer, lower WWTP capacity and higher population density, the Muusøya WWTP is suffering from the severe overflow. In the current phase, the Drammen municipality is planning to construct a storage tank to reduce overflow from the Muusøya WWTP. Due to the high density of buildings, there is very limited space for constructing storage tanks in the Muusøya catchment. Suggested by experts, the Landfalloverløpet was selected as the appropriate location for the storage tank, but the maximum dimension of the storage tank is restricted to 20,000 m^3, which is insufficient to deal with the current overflow situation. The ICWT is proposed in order to compensate insufficient capacity of the storage tank. The purpose of the ICWT schemes is to convey part of the WWTP inflow from the Muusøya WWTP to the Solumstrand WWTP.

2.2. The Regnbyge.no sewer monitoring system

To monitor the sewer system in real time, as well as provide data for model construction and calibration. A web based sewer monitoring system, called Regnbyge.no, was developed at the beginning of the Regnbyge 3M project. The Regnbyge.no sewer monitoring system is comprised of water level sensors, velocity sensors (NIVUS GmbH; Germany) and rain gauges deployed all around the Drammen city. The data collected by these sensors are transmitted to the data center of the Rosim AS. A spatial database was designed to manage the collected data, and ease the process of searching and editing data. The collected data is visualized by a web-based Geographic Information System (GIS), which supplies a user-friendly way to check the real-time and historical information. Fig. 2 is the screenshot of the regnbyge.no sewer monitoring system. Please note that the flow shown in the user interface is calculated according to water level, velocity and the pipe shapes.

2.3. Hydraulic model

To test the effectiveness of ICWT and derive proper control rules (task 1 and 2), a full detailed hydraulic model, with 9113 pipes, 9094 manholes, 129 weirs, 78 pumps and 39 outlets, is developed. Fig.3 is the hydraulic model for the Drammen sewer system. The sewer hydraulic model used in this study is Rosie. Rosie is an

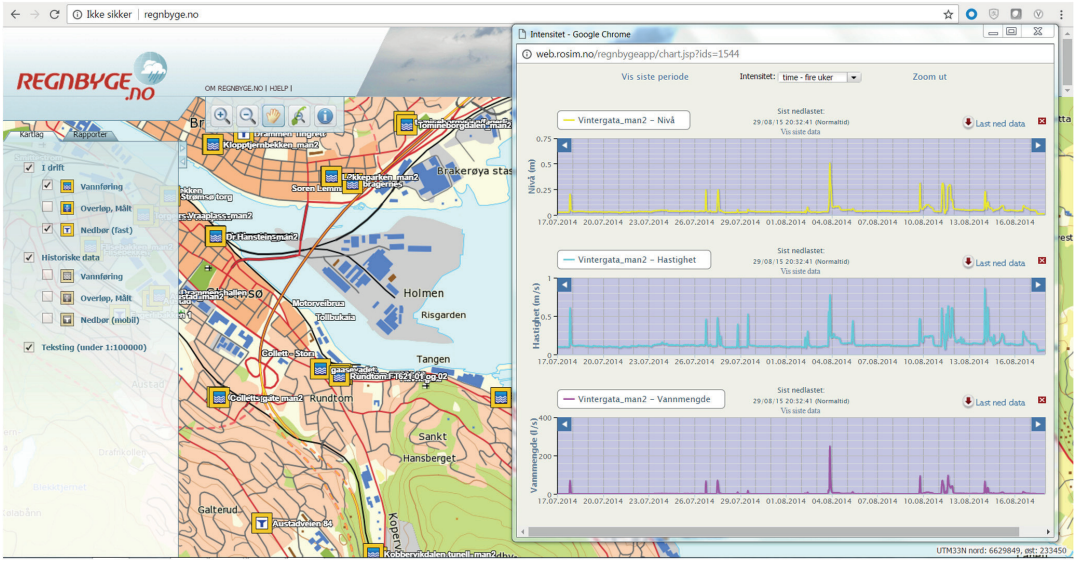


Fig. 2. The user interface of the Regnbyge.no sewer monitoring system.

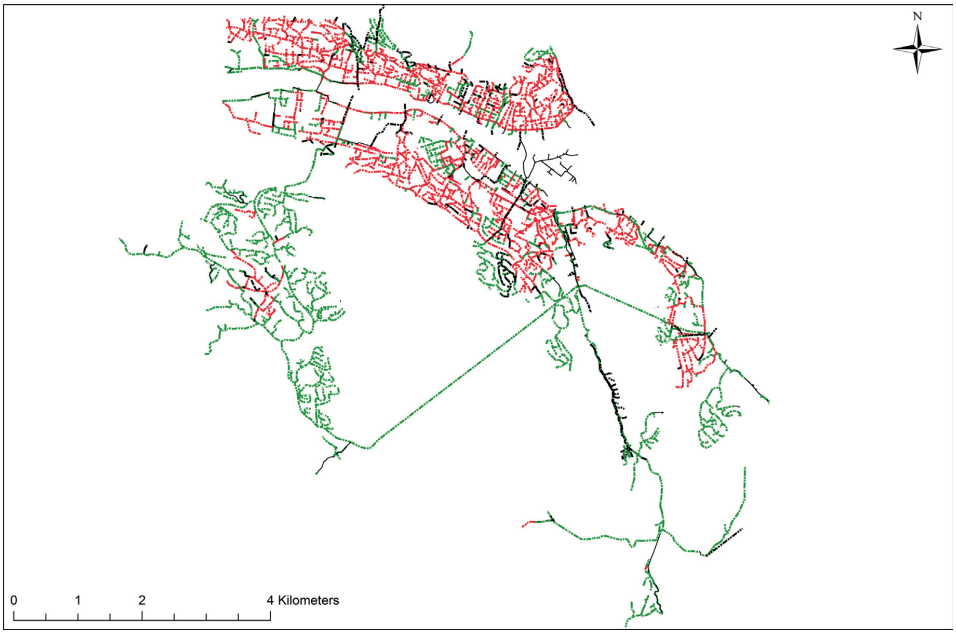


Fig. 3. The Hydraulic model for sewer system of Drammen.

ArcGIS additional application for planning, sizing and modeling of water distribution and sewerage systems developed by Rosim AS. The Rosie software maintains the interface and all the functions of ArcGIS, while using the MOUSE DHI as the computational engine to simulate the sewer system.

The runoff consists of two major components: the fast response component (FRC) and the slow response component (SRC). The FRC is the direct response from rainfall, which is calculated by the time–area (T-A) curve method in Rosie. Runoff model A is selected in this study. The SRC is the runoff generated gradually from the

previous hydrological processes and accumulated as interflow and baseflow. The Rosie use the rainfall dependent infiltration (RDI) method to calculate SRC. The pipe hydrodynamic (HD) flow is simulated by one dimension free surface Saint Venant continuity and momentum equations. The MOUSE real-time control (RTC) module is used to simulate sewer control. In the MOUSE RTC module, the operation of control structures, such as pumps, moveable weirs or orifices, are designed by a curve that provides the relationship between water level (H) and water flow (Q).

There are two types of weirs available in Rosie. One type of weir define it as a structure that connects two sewer nodes and the other type of weir diverts sewage out of the system once the sewage level in the pipe exceeds the weir crest level. Sewer overflow is calculated by summarizing discharges from the latter type of weirs using the HD model.

2.4. LSTM

This study employs a novel neural network architecture, LSTM, to undertake task 3 (inflow prediction for the Muusøya WWTP). The neural network is usually comprised of input layer, hidden layer and output layer. Parameters that connect input values with hidden neurons, as well as link hidden neurons with outputs called weights.

In the most widely used feedforward neural network (FFNN), the FFNN first computes a weighted sum of its input values and weights:

$$s = \sum_{i=1}^{n} w_i x_i + b \tag{1}$$

Then the weighted sum s is fed into the hidden layer. The hidden layer is made up of a number of hidden neurons, which contain an activation function. The hidden neurons transfer weighted sum s into output by using the activation function, such as the sigmoid function:

$$f(s) = \frac{1}{1+e^{-s}} \tag{2}$$

In the above equations, w_i represents the weights, x_i is the input values, b is the bias.

The LSTM is a special type of recurrent neural network (RNN), Fig.4 (a) is a schematic of the Elman RNN (Elman, 1990), which is one of the earliest RNN architecture. RNN feeds not only the weighted sum of inputs and weights, but also the states of the hidden neurons at the previous time steps into hidden neurons at the present time step. As shown in Fig.4 (a), the hidden neuron output at time step t is calculated by the equation:

$$h_t = f(w_h h_{t-1} + w_i x_t + b) \tag{3}$$

Where h_t is state of the hidden neuron at the time step t, h_{t-1} is state of the hidden neuron at the time step t-1, w_i and w_h are weights between input values and hidden neurons, and between hidden neurons respectively, $f()$ is the activation function.

Usually, neural network are trained by using the Back propagation (BP) method, BP use chain rule of differentiation to calculate the gradients of the error corresponding to the weights. In FFNN, BP propagates the error between predicted and observed values backward to the hidden layer, then to the weights. The neural network reduces differences between observed and predicted values through adjusting the weights.

The training of RNN uses an extended version of BP, also known as backpropagation through time (BPTT). BPTT adjust w_i and w_h during training, it means the chain rule of differentiation not only along the direction of hidden layer and input weights w_i, but also along previous time steps. When using the BPTT method, because the error of derivation accumulates through time, it will be extremely hard to learn and tune the parameters of the earlier layers. Because the gradient going through the network either gets very small and vanish, or get very large and explode. This is commonly known as the vanishing and exploding gradient problem.

Different from RNN, the LSTM use a memory cell and three gates to control information in the hidden neuron. The function of forget gate is to reset memory cell. The input gate permits inputs to modify the memory cell state. The output gate allows or obstructs the memory cell state from influencing other neurons. The memory cell can impede outside interference, which further allows the LSTM to learn time series with long spans.

Fig.4 (b) shows the hidden unit of LSTM. The i, f and o in Fig.4 denote the input, forget and output gate. The c and $\bar{c}$ represent the memory cell state and the new memory cell state. The mathematical representation of LSTM is written in the following equations:

Input gate:

$$i_t = \sigma_g(W_i * x_t + U_i * h_{t-1} + V_i^{\circ} c_{t-1} + b_i) \tag{4}$$

Forget gate:

$$f_t = \sigma_g(W_f * x_t + U_f * h_{t-1} + V_f^{\circ} c_{t-1} + b_f) \tag{5}$$

Output gate:

$$o_t = \sigma_g(W_o * x_t + U_o * h_{t-1} + V_o^{\circ} c_{t-1} + b_o) \tag{6}$$

Cell state:

$$c_t = f_t^{\circ} c_{t-1} + i_t^{\circ} \bar{c}_t \tag{7}$$

$$\bar{c}_t = \sigma_c(W_c * x_t + U_c * h_{t-1} + b_c) \tag{8}$$

Output vector:

$$h_t = o_t^{\circ} \sigma_h(c_t) \tag{9}$$

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, σ_g is the sigmoid function, σ_h and σ_c are the hyperbolic tangent function, for a given input z, the output of the hyperbolic tangent function is:

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{10}$$

In this study, the LSTM is implemented using Keras. The Docker Toolbox is used to setup a development environment to let different deep learning software and libraries coexist and function

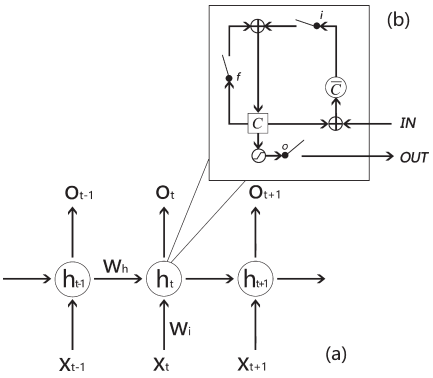


Fig. 4. Schematic of RNN and hidden units of LSTM.

correctly. The Jupyter notebook is used as the programming interface. Keras is a Python based high-level deep learning library. It is running on top of TensorFlow or Theano. TensorFlow is used as the backend of Keras in this study. TensorFlow is an open-source deep learning software released by Google in 2015. Other Python based machine learning libraries, includes Pandas, NumPy, Scikit-learn and Matplotlib are also used. Specifically, Pandas and NumPy are used to load the dataset as the data frame and prepare the raw data in the format of the desired array. Scikit-learn is used for model selection and preprocessing, such as tuning parameters and data normalization. Matplotlib is used for visualization.

2.5. Model performance criteria

Three model performance criteria are used in this study, i.e. root mean square error (RMSE), Nash-Sutcliffe Efficiency (NSE) and the coefficient of determination (R^2). The calculation of RMSE as shown below:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(Y_i^{obs} - Y_i^{sim}\right)^2}{n}} \tag{11}$$

NSE is a parameter that determines the relative importance of residual variance (noise) compare with the variance in the measured data (information). The NSE is calculated by the following equation:

$$NSE = 1 - \left[\frac{\sum_{i=1}^{n}(Y_i^{obs} - Y_i^{sim})^2}{\sum_{i=1}^{n}(Y_i^{obs} - Y^{mean})^2}\right] \tag{12}$$

NSE varies from $-\infty$ to 1, NSE = 1 indicates a perfect correlation between simulated and observed data, values between 0.0 and 1.0 is generally acceptable.

The equation for R^2 is:

$$R^2 = \left[\frac{\left(\sum_{i=1}^{n}\left(Y_i^{sim} - Y_{sim}^{mean}\right)\left(Y_i^{obs} - Y^{mean}\right)\right)^2}{\sum_{i=1}^{n}\left(Y_i^{sim} - Y_{sim}^{mean}\right)^2 \sum_{i=1}^{n}\left(Y_i^{obs} - Y^{mean}\right)^2}\right] \tag{13}$$

In the three above listed equations:

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

3. Results

3.1. Calibration of the hydraulic model

The historical flow data obtained from the Regnbyge.no sewer monitoring system is used to validate the hydraulic model. Fig. 5 shows the hydrographs of the hydraulic model outputs versus the recorded values at five monitoring sites. The recorded values cover both dry weather season and wet weather season. It clearly indicates that the flow rate simulated by the hydraulic model is consistent with the real values. Table 1 is the model performance criteria of the hydraulic model. All the criteria show acceptable values. Results displayed in Fig. 5 and Table 1 confirm the high reliability of the hydraulic model simulations. The calibrated model is then used in the following scenario analyses.

3.2. Scenario simulations of the hydraulic model

The observed precipitation in 2014 with a temporal resolution of 1 min is used for the hydraulic simulation. The simulation runs continuously from Jan. 01, 2014 to Dec. 31, 2014 as a baseline scenario against different control scenarios.

The total overflow volume of the Drammen sewer system for the baseline scenario is 2,096,668 m^3. Table 2 shows the simulated overflow volume through weirs associated to the Muusøya WWTP under the baseline scenario. The 507_Landfall is the major bypass structure of the Muusøya WWTP, which is the weir that takes away most of the peak flows to the Muusøya WWTP. The simulation result illustrates that weirs relate to the Muusøya WWTP contribute 22.7% of the total overflow. As the baseline scenario simulation indicates, implement overflow control solutions towards the Muusøya WWTP could bring the most immediate and significant result. Table 3 is simulated overflow volume in every month of 2014. One can observe that the overflow volume is significantly higher during the rainy season of Drammen from June to August.

In order to test the feasibility of ICWT, and compare its efficiency with the storage tank, seven control scenarios are designed. Table 4 gives a description of different control scenarios. The main goal of the control is to: 1) reduce the total overflow, 2) reduce overflow from the Muusøya WWTP and 3) avoid bringing extra burden to the Solumstrand WWTP. In the control scenarios, the distribution of sewage is determined according to the principal of the greedy algorithm. Greedy algorithm refers to for the solution to a problem builds up a solution step by step, always taking the action that gets the most immediate benefit at the next step (local optimal solution) within the specified constraints step by step rather than considering the global optimum (Cormen, 2009).

According to the principal of the greedy algorithm, the control structures such as pump first maximum inflow to Muusøya WWTP during peak flow events, as it is the easiest way to deal with peak flow. If the influent flow rate exceeds the maximum designed treatment capacity of the Muusøya WWTP, the exceeded volume is drained to the storage tank (except scenario 4, which activate ICWT directly). If the storage tank is still insufficient to dealing with the inflow, for scenario 5–7, the ICWT is activated to divert sewage to the adjacent Brageners catchment.

The impact of the different control scenarios on the total overflow is investigated. Fig. 6 displays the reduction of total overflow volume compare with the baseline scenario. Table 5 lists the percentage of total overflow reduction of control scenarios compare with the baseline scenario.

As expected, no matter only use storage tank or applying the ICWT, all seven control scenarios reduced overflow volume compare with the baseline scenario. There is a clear tradeoff between storage tank sizes and overflow volume, with the total overflow decreasing as the storage tank size increases. With the smallest storage tank dimension, the total overflow of scenario 1 is close to the baseline scenario, the relative percentage is only 1.20%, which indicates that the effect of scenario 1 is not distinct. When the storage tank is built with the maximum available size of 20000 m^3 (scenario 3), the overflow reduction is 258,895 m^3, a reduction of 12.35%.

The storage tank is less efficient compared with ICWT. As shown in Fig. 6, in scenario 1 and scenario 2, using a storage tank with a volume of 1000 m^3 and 5000 m^3, the volume of reduced overflow is less than only implementing the ICWT (scenario 4). Scenario 4 with only ICWT implemented gives an overflow reduction of 120,504 m^3, 5.75% less than the total overflow volume of the baseline scenario. ICWT enhances the capability of the storage tank dramatically. Overflow reduction of scenario 1 and scenario 2 increase from 25,146 m^3 and 95,412 m^3 to 141,741 m^3 (scenario 5) and 201,011 m^3 (scenario 6) respectively. Scenario 7 yields the highest

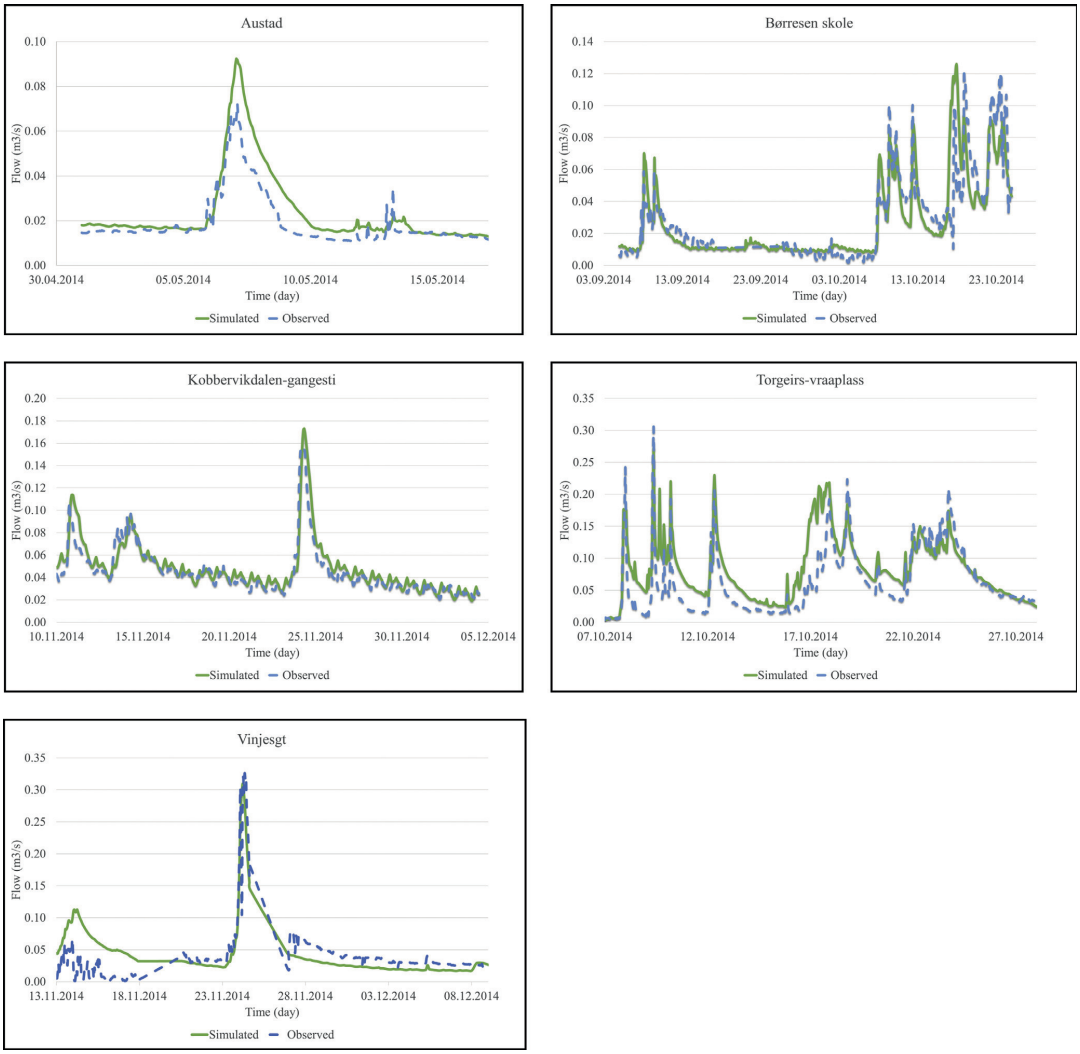


Fig. 5. Hydrographs of observed versus simulated flow at five stations.

Table 1
Values of model calibration criteria at five stations.

Measurement point	NSE	R^2	RMSE
Austad	0.54	0.91	0.008
Børresen skole	0.68	0.71	0.014
Kobbervikdalen-gangsti	0.80	0.85	0.010
Torgeir-vraaplass	0.51	0.67	0.036
Vinjesgt	0.53	0.57	0.029

Table 2
Overflow volume of outlets associated to the Musøya WWTP for the baseline scenario.

Weir ID	Simulated overflow volume (m^3)
50433_w_1	5751
505_Store Landfall	6016
507_Landfall	456,171
508_Landfall	7113

overflow reduction, i.e. a reduction of 15.51%. Nevertheless, it should be noticed that using a storage tank of 5000 m^3 (only one-fourth of the maximum dimension used by scenario 7), scenario 6 achieves over half the effect of scenario 7, i.e. 9.6%.

Although implementing storage tank and ICWT resulted in significant reduction of total overflow, one obvious concern for ICWT is whether it will bring extra burden to the Solumstrand WWTP. The volume of overflow from the bypass structures of the two WWTPs under different scenarios is listed in Table 6. The overflow from the bypass structure of Muusøya WWTP is reduced substantially. For the Solumstrand WWTP, the impact of ICWT is almost negligible.

Table 3
Overflow volume in every month of 2014 for the baseline scenario.

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Overflow volume (m^3)	7929	4216	58,633	422,122	56,382	120,189	116,986	149,685	92,498	737,182	329,652	1195

Table 4
Description of the control scenarios.

Scenario	Description
Scenario 1	Only construct a 1000 m^3 storage tank at Landfalloverløpet
Scenario 2	Only construct a 5000 m^3 storage tank at Landfalloverløpet
Scenario 3	Only construct a 20,000 m^3 storage tank at Landfalloverløpet
Scenario 4	Only implement the ICWT between the Muusøya catchment and the Brageners catchment
Scenario 5	1000 m^3 storage tank + ICWT
Scenario 6	5000 m^3 storage tank + ICWT
Scenario 7	20,000 m^3 storage tank + ICWT

Table 7 is the total overflow volume of different control scenarios from March to November. It can be seen that the overflow reduction is most significant during the rainy season of Drammen.

3.3. LSTM

The hydraulic simulations of the eight scenarios clearly demonstrate the viability of ICWT. However, since these hydraulic simulations are based on perfect foreknowledge of the rainfall events, they cannot be implemented for the actual real-time purpose. To achieve successful control of the storage tank and ICWT, the overflow control facilities should open/close timely in order to redistribute WWTP inflow. For the better operation of the storage tank and ICWT, it is essential to anticipate the future WWTP inflow, thus enhancing decision-making and giving enough response time for the operation.

To overcome the aforementioned disadvantages of hydraulic model and traditional neural network, LSTM is employed to predict inflow to the Muusøya WWTP. The training data for LSTM development were collected from Jan. 1, 2014 to Dec. 31, 2014 with a temporal resolution of 1 min. The raw data have over 500,000 records. To validate the generalization ability of the proposed LSTM algorithm, the raw data are divided into two subsets: training set and testing set. Data from the first 75% is used for training, and the remaining 25% is used for testing. The difference between training and testing is regularization mechanisms, which is used as penalize to limit overfitting, are turned off during the testing stage. Table 8 is summarized statistics of the training data.

Considering technical details of the pump operation, the proposed models aim at predicting flow rate in the next hour based on the flow rate and rainfall in the past hour. During dry weather, the flow rate is stable and the pump can be operated at a lower frequency, while during wet weather, it is essential to predicate flow rate with a higher resolution. To fulfill pump operation requirements, the raw data is resampled to frequencies of 10 min, 15 min, half hour and one hour, models with corresponding frequencies are developed.

There are primarily four modes of LSTM application (Karpathy, 2016). The one to one mode such as image classification, have one input (image) and one output (image label). The one to many mode generates many outputs using one input, e.g. image captioning, which takes an image and outputs a sentence of words. The many to one mode receive sequence input and output one label (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment). The many to many mode (or sequence to sequence, Abbreviated as seq2seq) is the method for machine translation or chat robot, with a sentence in one language as input and a sentence in the target language as output. The one hour frequency model is defined as a one to one problem, i.e. predicting flow rate of the next hour by using the values from

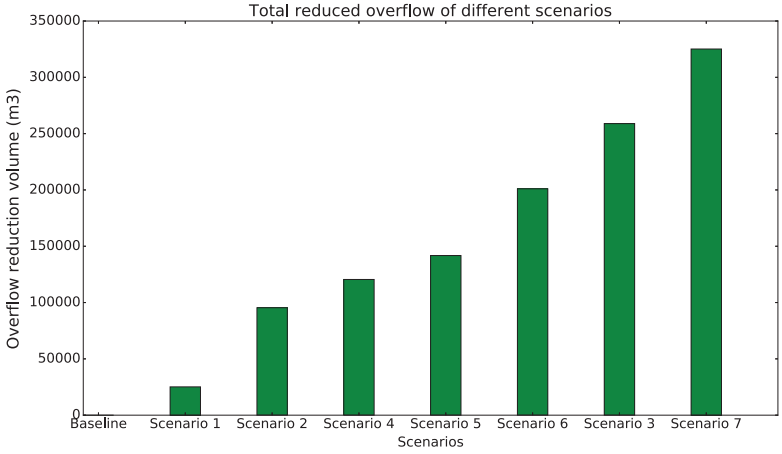


Fig. 6. Total overflow volume reduction of different scenarios.

Table 5
Relative percentage of total overflow reduction of control scenarios compares to the baseline scenario.

Scenario	Relative percentage compare with the baseline scenario (%)
Baseline scenario	0.00%
Scenario 1	1.20%
Scenario 2	4.55%
Scenario 4	5.75%
Scenario 5	6.76%
Scenario 6	9.59%
Scenario 3	12.35%
Scenario 7	15.51%

Table 6
Volume of overflow from the bypass structures of the Muusøya WWTP and the Solumstrand WWTP.

	Muusøya WWTP (m^3)	Solumstrand WWTP (m^3)
Baseline scenario	456,171	6214
Scenario 1	438,595	6174
Scenario 2	372,403	6170
Scenario 3	215,059	6144
Scenario 4	323,404	6182
Scenario 5	311,159	6197
Scenario 6	266,370	6165
Scenario 7	162,722	6193

the previous hour. LSTM models with other three frequencies are defined as a seq2seq problem (Zaytar et al., 2016), e.g. for the 10 min frequency data, the model uses the values of the past 6 steps to predict the next 6 steps.

Several state-of-the-art techniques such as dropout and RMSprop are used to optimize the LSTM. Dropout (Hinton et al., 2012) is a simple but effective way to increase the generalization ability of neural network. Traditional neural network training usually uses the ensemble method to prevent overfitting, but it could significantly increase CPU consumption. The key idea of dropout is temporary discard part of units (either hidden neurons or visible inputs) from the neural network during training. Dropout generates a number of different "thinned" networks during training, and in the testing stage, a single un-thinned network is used (Srivastava et al., 2014). RMSprop is an unpublished, adaptive learning rate method proposed by Geoff Hinton in Lecture 6e of his Coursera course. RMSprop utilizes the magnitude of recent gradients to normalize the gradients, and keeps a moving average over the root mean squared gradients for each weight. RMSprop also divides the learning rate by an exponentially decaying average of squared gradients.

Through trial and error experiments, the final LSTM architecture used in this paper has three hidden layers with 10 LSTM units in each hidden layer, dropout layer is implemented between hidden layers. After ascertaining the optimum structure of LSTM, the model is used to predict the inflow to the Muusøya WWTP. Furthermore, the LSTM is compared with two of the most widely used machine learning models in the water resource field: FFNN and support vector regression (SVR). To make a fair comparison, the FFNN remains the same deep structure with LSTM. The kernels and parameters of SVR are confirmed using the grid search method, the optimal SVR used in this study is an RBF kernel SVR with a gamma value of 0.5 and C value of 5.

Table 9 presents the model performance. The results imply that LSTM outperforms FFNN and SVR. LSTM produces a high NSE value

Table 7
Total overflow volume from March to November under different control scenarios.

Model	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov
Scenario 1	57,784	420,834	54,732	119,065	108,576	146,716	91,541	732,503	326,433
Scenario 2	54,162	414,206	51,236	111,459	97,531	139,422	86,532	716,762	316,609
Scenario 3	52,430	375,023	45,310	91,576	82,197	123,216	73,105	679,867	301,713
Scenario 4	54,199	342,789	51,937	116,375	106,363	143,005	88,476	731,620	328,060
Scenario 5	53,723	341,477	51,350	115,340	100,775	140,273	86,817	726,333	325,481
Scenario 6	52,430	338,330	48,020	106,239	93,774	133,414	82,536	711,645	315,933
Scenario 7	52,430	328,623	45,353	87,710	81,837	119,563	68,936	675,431	298,298

Table 8
Summary statistics of the datasets.

Model stage	Max flow rate (m^3/s)	Average flow rate (m^3/s)	Standard deviation of flow rate	Max rainfall (mm/s)	Average rainfall (mm/s)	Standard deviation of rainfall
Training	1.50	0.18	0.12	33.33	2.41	5.36
Testing	1.79	0.14	0.09	25.29	1.94	4.47

Table 9
Model performance of LSTM, FFNN and SVR with different sampling frequencies.

Model	Performance criteria	Sampling frequency			
		One hour	Half hour	15 min	10 min
LSTM	R^2	0.9677	0.9555	0.9426	0.9396
	RMSE	0.0150	0.0176	0.0203	0.0208
	NSE	0.9668	0.9549	0.9406	0.9378
FFNN	R^2	0.9607	0.9321	0.9330	0.8962
	RMSE	0.0235	0.0236	0.0277	0.0319
	NSE	0.9181	0.9188	0.8798	0.8542
SVR	R^2	0.9660	0.9533	0.9433	0.9413
	RMSE	0.0496	0.0435	0.0410	0.0392
	NSE	0.6357	0.7224	0.7570	0.7792

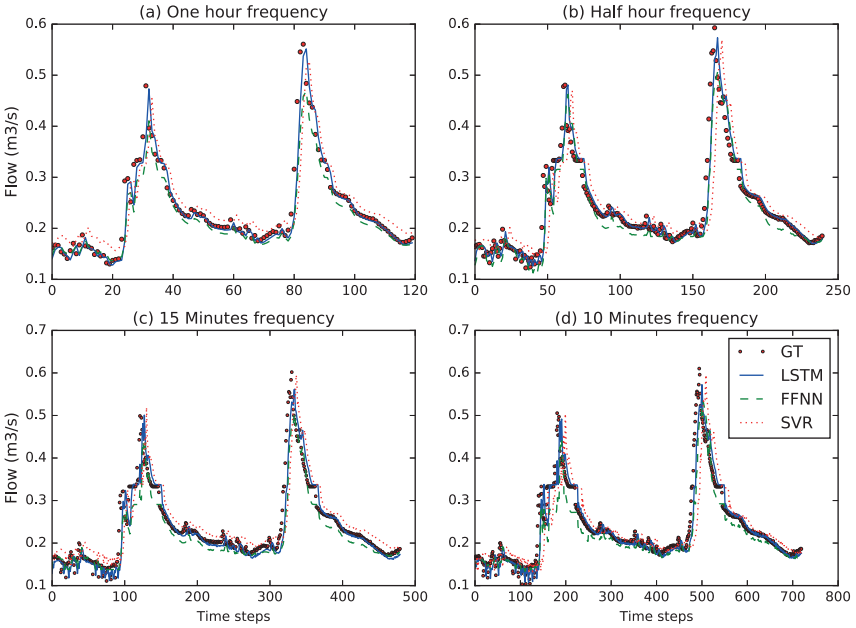


Fig. 7. Hydrographs of LSTM, FFNN and SVR outputs versus the ground truth (GT) values for 10 min, 15 min, half hour and one hour frequency models from Nov. 7, 2014 to Nov. 12, 2014.

(close to 0.95), high R^2 value (higher than 0.93) and low RMSE value (around 0.02). The values of the three criteria indicate that the LSTM has the ability to predict highly nonlinear and variable time series, such as WWTP inflow.

To illustrate the performance of the developed LSTM, FFNN and SVR models in a more intuitive way, hydrographs of ground truth (GT) versus forecasted flow from Nov. 7, 2014 to Nov. 12, 2014 are drawn in Fig. 7. It can be seen from Fig. 7 that LSTM is capable of predicting the inflow to the Muusøya WWTP in spite of significant variations in flow rate during these days. The LSTM is able to capture the major trends and peaks of observations, while the FFNN model underestimates the peak value. It also can be observed that for the periods of rainfall, the LSTM anticipate two increases in flow rate timely, while the SVR presents a time delay problem.

4. Discussion and Conclusion

The increase of urbanization and extreme rainfall events are inducing more frequent sewer overflow. At the same time, the treatment capacity of WWTP also seriously affected. Efficient sewer management measures have to be applied to mitigate sewer overflow. For existing sewer system, although storage tank could supply a straightforward solution for overflow reduction, their locations and dimensions are highly constrained. In densely populated cities, alternative solutions should be investigated.

This paper describes a novel ICWT solution. The effectiveness of ICWT is evaluated by a case study for a real sewer system in Drammen, Norway. The hydraulic behaviors of the sewer system with different control scenarios are assessed using hydraulic model. Annual long simulations of eight scenarios show that the ICWT could provide extra benefits for overflow mitigation.

This study also demonstrates that LSTM is able to provide precise time series predictions for sewer system management. LSTM can be a powerful tool for managers or engineers of the sewer system, who can take advantage of the prediction to improve their decision-making. Intelligent urban infrastructures will become the backbone of future cities. Sensors, actuators and algorithms are integrating together to enhance important infrastructures such as sewer system or WWTP. There are many possible scenarios to complement urban infrastructure control with deep learning.

In recent years, deep learning methods such as LSTM are gaining more and more successes in many industries. However, the sophisticated mathematical knowledge and programming skill behind deep learning limit its application in the water resource field. Even so, the potential power of deep learning still fascinating, and more studies about absorbing deep learning into the water resource field is necessary.

Acknowledgements

This work has been supported by the Regnbyge-3M project (grant number 234974), which is granted by the Oslofjord Regional Research Fund. The authors would like to thank the engineers from Rosim AS for their supports.

References

Autixier, L., Mailhot, A., Bolduc, S., Madoux-Humery, A.S., 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.

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, Y.M., Chang, L.C., Tsai, M.J., Wang, Y.F., Chang, F.J., 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.

Cormen, T.H., 2009. Introduction to algorithms. MIT press.

Darsono, S., Labadie, J.W., 2007. Neural-optimal control algorithm for real-time regulation of in-line storage in combined sewer systems. Environ. Modell. Soft. 22 (9), 1349–1361.

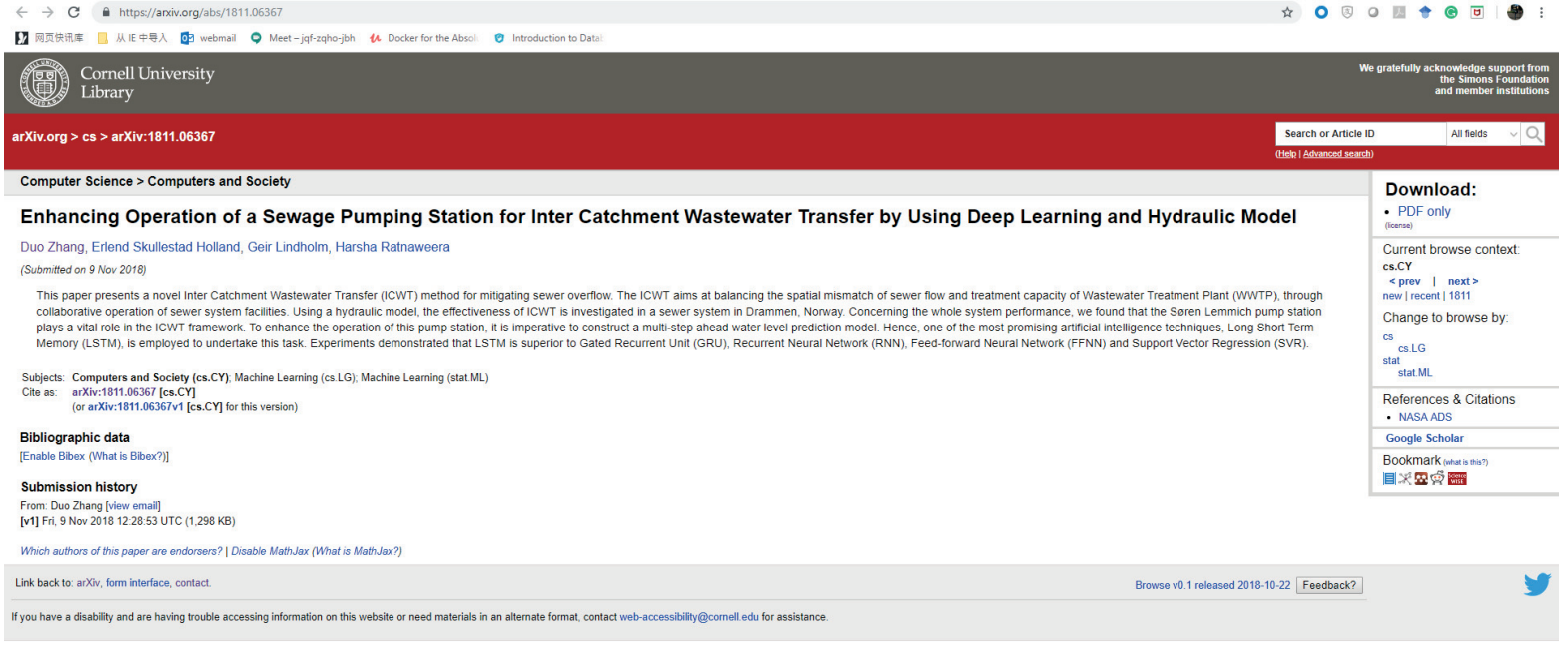
Duchesne, S., Mailhot, A., Dequidt, E., Villeneuve, J.P., 2001. Mathematical modeling of sewers under surcharge for real time control of combined sewer overflows. Urban Water 3 (4), 241–252.
El-Din, A.G., Smith, D.W., 2002. A neural network model to predict the wastewater inflow incorporating rainfall events. Water Res. 36 (5), 1115–1126.
Elman, J.L., 1990. Finding structure in time. Cognit. Sci. 14 (2), 179–211.
Ganora, D., Isacco, S., Claps, P., 2017. Framework for enhanced stormwater management by optimization of sewer pumping stations. J. Environ. Eng. 143 (8), 04017025.
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. Network Comput. Appl. 78, 30–42.
Google. (2016). https://research.googleblog.com/2016/09/a-neural-network-for-machine.html (accessed 27 April 2017)
Grum, M., Thornberg, D., Christensen, M. L., Shididi, S. A., Thirsing, C., 2011. Full-scale real time control demonstration project in Copenhagen's largest urban drainage catchments. Proceedings of the 12th international conference on urban drainage.
Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv, preprint arXiv:1207.0580.ors.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Karpathy, A., 2016. The unreasonable effectiveness of recurrent neural networks (accessed 07 July 2017) http://karpathy.github.io/2015/05/21/rnn-effectiveness, .
Laptev N., Smyl S., Shanmugam S., 2017. https://eng.uber.com/neural-networks/ (accessed 05 October 2017)
Lee, E.H., Lee, Y.S., Joo, J.G., Jung, D., Kim, J.H., 2017. Investigating the impact of proactive pump operation and capacity expansion on urban drainage system resilience. J. Water Resour. Plann. Manage. 143 (7), 04017024.
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, W.C., Sample, D.J., 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. Transp. Res. Part C: Emerg. Technol. 54, 187–197.
Ngo, T.T., Yoo, D.G., Lee, Y.S., Kim, J.H., 2016. Optimization of upstream detention reservoir facilities for downstream flood mitigation in Urban Areas. Water 8 (7), 290.
Seggelke, K., Rosenwinkel, K.H., Vanrolleghem, P.A., 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.
Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., et al., 2016. Mastering the game of Go with deep neural networks and tree search. Nature 529 (7587), 484–489.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Machine Learn. Res. 15 (1), 1929–1958.
Vrebos, D., Vansteenkiste, T., Staes, J., Willems, P., Meire, P., 2014. Water displacement by sewer infrastructure in the Grote Nete catchment, Belgium, and its hydrological regime effects. Hydrol. Earth Syst. Sci. 18 (3), 1119–1136.
Wang, Q., Zhou, H., Liang, G., Xu, H., 2015. Optimal operation of bidirectional inter-basin water transfer-supply system. Water Resour. Manage. 29 (9), 3037–3054.
Wu, H., Huang, G., Meng, Q., Zhang, M., Li, L., 2016. Deep tunnel for regulating combined sewer overflow pollution and flood disaster: a case study in Guangzhou city, China. Water 8 (8), 329.
Yevjevich, V., 2001. Water diversions and interbasin transfers. Water Int. 26 (3), 342–348.
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, M.A., Amrani, C.E., 2016. Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. Int. J. Comput. Appl. 143, 7–11.

Paper IV

Zhang, D., Holland, E. S., Lindholm, G., & Ratnaweera, H. (2018). Enhancing operation of a sewage pumping station for inter catchment wastewater transfer by using deep learning and hydraulic model. *arXiv:1811.06367 [cs.CY]*. (*preprint in arXiv*)

Access to full paper at:

https://arxiv.org/abs/1811.06367



Enhancing Operation of a Sewage Pumping Station for Inter Catchment Wastewater Transfer by Using Deep Learning and Hydraulic Model

Duo Zhang[1]; Erlend Skullestad Hølland[1]; Geir Lindholm[2]; Harsha Ratnaweera[1]

1. Faculty of Sciences and Technology, Norwegian University of Life Sciences, 1432, Ås, Norway
2. Rosim AS, Brobekkveien 80, 0582, Oslo, Norway

Abstract: This paper presents a novel Inter Catchment Wastewater Transfer (ICWT) method for mitigating sewer overflow. The ICWT aims at balancing the spatial mismatch of sewer flow and treatment capacity of Wastewater Treatment Plant (WWTP), through collaborative operation of sewer system facilities. Using a hydraulic model, the effectiveness of ICWT is investigated in a sewer system in Drammen, Norway. Concerning the whole system performance, we found that the Søren Lemmich pump station plays a vital role in the ICWT framework. To enhance the operation of this pump station, it is imperative to construct a multi-step ahead water level prediction model. Hence, one of the most promising artificial intelligence techniques, Long Short Term Memory (LSTM), is employed to undertake this task. Experiments demonstrated that LSTM is superior to Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), Feed-forward Neural Network (FFNN) and Support Vector Regression (SVR).

Keywords: Deep learning; Sewer overflow; Long Short Term Memory; Inter catchment wastewater transfer

Author names and affiliations:

Duo Zhang (corresponding author):

Ph.D. candidate, Faculty of Science and Technology, Norwegian University of Life Sciences

Email: Duo.Zhang@nmbu.no

Erlend Skullestad Hølland

Master student, Faculty of Science and Technology, Norwegian University of Life Sciences

Geir Lindholm:

CEO, Rosim AS, Brobekkveien 80, 0582, Oslo, Norway

Email: geir@rosim.no

Harsha Ratnaweera:

Professor, Faculty of Science and Technology, Norwegian University of Life Sciences

Email: Harsha.Ratnaweera@nmbu.no

1. Introduction

Control overflow from the sewer system and Wastewater Treatment Plant (WWTP) is a crucial and challenging task for many cities in developed countries, such as the Drammen city in Norway. The Drammen city is located in southeastern Norway, it has two wastewater treatment plants: the Muusøya WWTP and the Solumstrand WWTP. The Muusøya WWTP has a designed treatment capacity of 33,000 PE (population equivalents), a dimensioning flow (Qdim) of 780 m^3/h, and a maximum flow (Qmax) of 1,200 m^3/h. The Solumstrand WWTP has a designed treatment capacity of 130,000 PE, the Qdim and Qmax for the Solumstrand WWTP is 2,000 m^3/h and 4,000 m^3/h respectively. Combined sewer accounts for more than 80% and less than 50% respectively in the sewer system associated with the Muusøya WWTP and the Solumstrand WWTP. Moreover, the drainage area of the Muusøya WWTP has a higher population density than the rest of Drammen due to it is located in the traditional city center. The lower WWTP capacity, a higher portion of combined sewer, and denser population have resulted in the severe overflow problem in the Muusøya area. Therefore, the Drammen city launched the Regnbyge 3M project to mitigate overflow from the sewer system and the WWTP of Drammen.

There are two types of overflow mitigation measures: structural measures and nonstructural measures (Lee et al., 2017). Structural measures refer constructing new hydraulic facilities and the rehabilitation of sewer components (e.g., expansion of sewer pipes). Nonstructural measures are methods that maximize the capacity of the sewer system with minimal changes to the infrastructure through intelligent operating strategies. As the most popular structural measures, the storage tank is still servicing in many developed cities. However, due to limited space or high investments, storage tanks cannot be always constructed in densely populated urban context (Ganora et al., 2017; Ngo et al., 2016) such as the Muusøya area. The drawbacks of structural measures have motivated the research for nonstructural methods, such as exploit the sewer in-line storage capacity (Darsono and Labadie, 2007; Grum et al., 2011; Garofalo et al., 2017), intelligent sewer control (Lee et al., 2017) and explore underground space (Wu et al., 2016).

Considering the spatial mismatch of capacity of the sewer system and WWTP between the Muusøya

WWTP and the Solumstrand WWTP, as well as according to the goal of the Regnbyge 3M project, we propose a novel nonstructural method: Inter Catchment Wastewater Transfer (ICWT) for the Drammen city. The idea of ICWT is inspired by the concept of Inter-basin Water Transfer (IBWT). IBWT refers to transfer water from basins having sufficient water (donor basin) to basins facing water shortages (receiving basin) (Wang et al., 2015, Yevjevich 2001). IBWT utilizes the differences of flow regime in different basins to create a win-win situation.

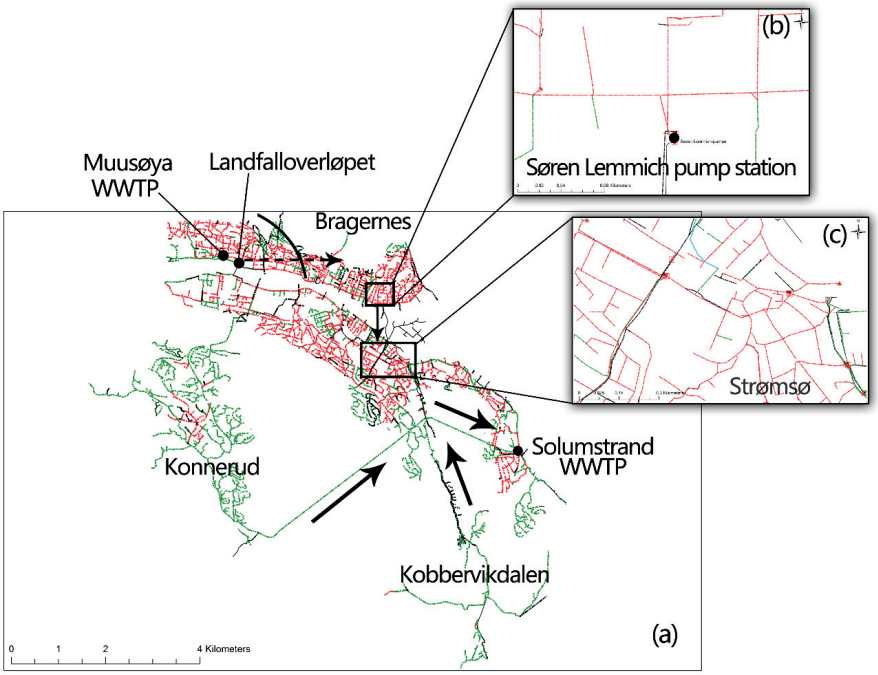


Fig. 1. Overview of the sewer system in Drammen

Fig.1 gives an overview of Drammen. The Drammen Fjord flow through Drammen, there are two catchments on the north of the Drammen Fjord, the Muusøya catchment and the Bragernes catchment, the curve in Fig. 1 (a) is the boundary of these two catchments. The Muusøya WWTP treats sewage collected in the Muusøya catchment. Wastewater from the Bragernes catchment is transported by the Søren Lemmich

pump station (Fig. 1 (b)) from the north of the Drammen Fjord to south (the Strømsø catchment, Fig. 1 (c)). Afterward, sewage from the Bragernes catchment and the Strømsø catchment merges with wastewater from the Konnerud catchment and the Kobbervikdalen catchment, then discharge to the Solumstrand WWTP. We generalize the concept of IBWT to sewer system management, the drainage area of the Muusøya WWTP can be regarded as the 'donor basin', and the drainage area of the Solumstrand WWTP can be regarded as the 'receiving basin'. If flow to the Muusøya WWTP already exceeded its capacity and the Solumstrand WWTP still has leftover treatment capacity, part of wastewater can be conveyed to the Solumstrand WWTP for treatment. The ultimate objective of ICWT is to balance the distribution of sewer flow and uneven WWTP treatment capacities in different catchments.

Therefore, three specific questions are raised: First, whether ICWT could reduce the overflow? Second, what are the individual and combined effects of ICWT and structural measures such as storage tank? Third, it is obvious that under the ICWT scheme, the Søren Lemmich pump station will become the bottleneck of the whole system. The Søren Lemmich pump station will be requested to operate with high sensitivity, if the Søren Lemmich pump station cannot pump wastewater timely, the ICWT will only bring extra burden to the Bragernes catchment rather than mitigate overflow. The operation of a pump station highly depends on the water level information. Pumps will be activated when the water level reaches the start level of pumps. The operation of a pump station can be enhanced if accurate water level prediction information can be provided (Chiang et al., 2010). To timely operate a pump, enhance decision-making or give enough response time for operators, it is imperative to find a model that can provide the multi-step ahead water level information (Liu et al., 2016; Chang et al., 2014; Chen et al., 2014).

In present practice, the assessments of the effectiveness of nonstructural measures count on hydraulic models mainly (Autixier et al., 2014; Lucas and Sample, 2015; Chiang et al., 2010; Seggelke et al., 2005). Hydraulic models allow engineers gain insight into the functioning and effects of nonstructural measures (Chiang et al., 2010). So that the output of hydraulic models is suitable for the first and the second question. However, the implementation of hydraulic models requires perfect foreknowledge of the sewer system.

Besides, calibration, simulation and operation of hydraulic models is a time-consuming manual process. The hydraulic models can only provide information based on previous or current rainfall events. The aforementioned disadvantages of hydraulic models limit its application for question 3 (El-Din et al., 2002).

Indeed, question 3 is a hydrologic time series problem. In recent years, there is a significant rise in the number of machine learning approaches applied to hydrologic modeling and forecasting (Nourani et al., 2014). Unlike hydraulic models that derived from the hydraulic and hydrological process, the machine learning approaches learning from data without human intervention. Moreover, the trained machine learning algorithms can produce future hydrological data by being fed with current and previous data. Abovementioned advantages of machine learning have stimulated researchers to absorb it into studies about the sewer system (Yu et al., 2013; Montserrat et al., 2015; Granata et al., 2016; Zhang et al., 2016; Mounce et al., 2014).

The machine learning methods have been less active in the past decade during a period called artificial intelligence (AI) winter (Marçais & de Dreuzy 2017). In recent years, with the computer program (Google DeepMind's AlphaGo) defeated Go game world champion (Silver et al., 2016), there is renewed interest in machine-learning methods. The breakthrough technology behind AlphaGo is state of the art branch of machine learning - deep learning. The deep learning is a topic that is making big waves now, in addition to AlphaGo, another typical application of deep learning is the latest Google translation system. The new Google translation system vastly improved the translation quality, brought service nearly to the level of human translators (Google, 2016). The game changer behind the latest Google translation system is a kind of Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997).

When performing translation, the model has to consider not only the current word, but also the other words in the sentence or even paragraph. Data with this kind of context information called sequential data. Time series data are the most popular form of sequential data, stimulated by the success of LSTM on machine translation, a few studies have explored the power of LSTM on traffic time series forecasting (Hsu, 2017).

In a case study using speed data from a sensor in Beijing, China, Ma et al., (2015) made a comparison between LSTM and RNN, SVM and traditional time series models, and LSTM outperforms other methods on time forecasting. In Tokyo, Song et al., (2016) developed an LSTM based system for predicting human mobility and transportation mode at a citywide level. Using traffic time series data from a highway around Oslo, Kanestrøm (2017) compared LSTM with recent advances of Stacked Sparse Auto Encoder (SSAE) and Deep Neural Network (DNN) on time series forecasting. The results found that the LSTM model always outperformed other models. Although LSTM has shown its superior performance, to the best of the author's knowledge, there are no prior reports about the application of LSTM in the urban hydrology studies, the effectiveness of LSTM need to be investigated.

The objective of this study has three components: (1) assessing the feasibility of ICWT, (2) studying the individual and combined effects of ICWT and storage tank, and (3) evaluating the performance LSTM on hydrologic time series prediction.

2. Method and materials

2.1 The Regnbyge.no sewer monitoring system

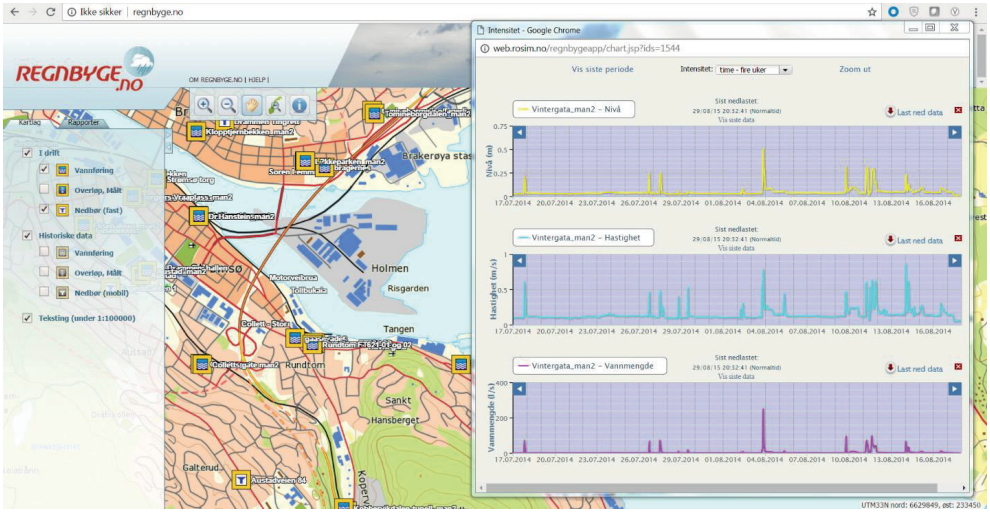


Fig. 2. The user interface of the Regnbyge.no sewer monitoring system

For the purpose of monitoring the sewer system as well as collecting data for model development, the Rosim AS, Norway developed a sewer monitoring system, Regnbyge.no, at the initial phase of the Regnbyge 3M project. The Regnbyge.no system consists of a number of water level sensors, velocity sensors from NIVUS GmbH, Germany and rain gauges deployed in Drammen. These sensors and rain gauges transmit collected data wirelessly to the data center at Rosim AS. A spatial database is employed to ease the process of searching, editing and managing of the collected data. Furthermore, a web Geographic Information System (GIS) is used to visualize data in the user interface of the Regnbyge.no system. Fig. 2 is the screenshot of the Regnbyge.no sewer monitoring system.

2.2 Hydraulic model

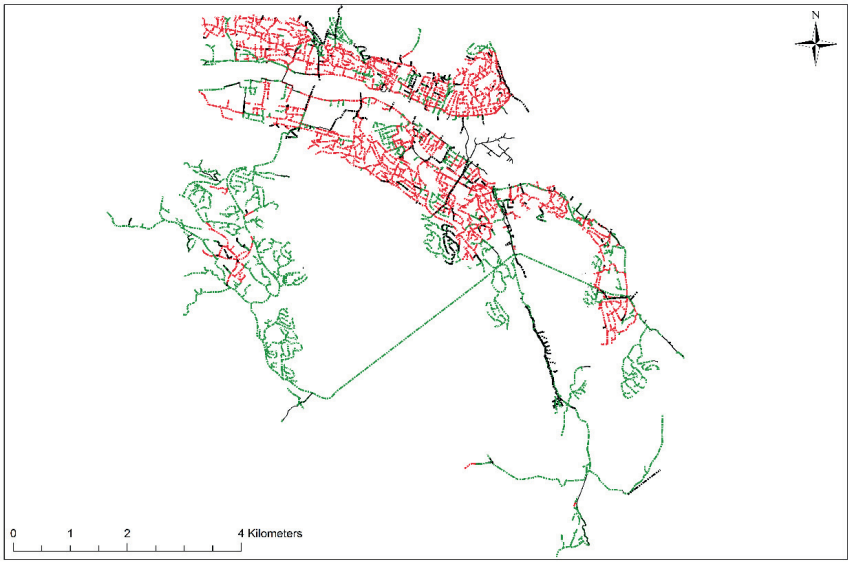


Fig. 3. Hydraulic model for the sewer system of Drammen

To test whether could ICWT reduce overflow, as well as study the individual and combined effects of ICWT and storage tank in considering the whole system behavior (question 1 and 2), a fully detailed hydraulic model was developed. Fig. 3 shows the hydraulic model for the sewer system of Drammen, the model consists of 9113 pipes, 9094 manholes, 129 weirs, 78 pumps and 39 outlets. The software used for hydraulic model development is Rosie, which is an ArcGIS extension developed by Rosim AS. The Rosie software maintains the interface and all the functions of ArcGIS, while using the MOUSE DHI as the computational engine. Interest readers may refer to the website of Rosie (http://web.rosim.no/index.php/tjenester/modellering-av-vann-og-avlopsnett/, in Norwegian) for more details.

In the present study, the direct response from the rainfall is calculated by the time–area (T-A) curve method A. The Rainfall Dependent Inflow/Infiltration (RDII) model is used to calculate the runoff generated from the previous hydrological processes. The pipe hydrodynamic computation is based on Saint-Venant continuity and momentum equations. The MOUSE RTC (Real Time Control) module is used to simulate different control strategies.

2.3 Machine learning

To provide the multi-step ahead water level prediction for managers to make decisions about pump operation. The performance of different machine learning methods, e.g. traditional algorithms such as Support Vector Regression (SVR), Feed-forward Neural Network (FFNN), traditional RNN and recent advances in deep learning (LSTM and Gated Recurrent Unit (GRU)) are evaluated.

Support Vector Machine (SVM) methods such as SVR was the major competitor of the neural network family. Although neural networks such as deep learning are nowadays dominating artificial intelligence technology, however, once upon a time, neural networks were almost unnoticed as they were overshadowed by the SVM (Cortes & Vapnik, 1995). SVR is a subcategory of SVM designed for regression problems. SVR is a kind of linear model, but it could solve nonlinear problems by using a kernel to transfer data into

a feature space, and then use a linear learning mechanism to learn a nonlinear function.

FFNN is one of the most classical neural network architectures, which is comprised of input layer, hidden layer, and output layer. There are some neurons in each layer and different layers are connected by weights and bias. The FFNN first computes the weighted sum of the inputs, which can be mathematically represented as:

$$s = \sum_{i=1}^{n} w_i x_i + b \tag{1}$$

Where w_i represents the weights, x_i is the inputs, b is the bias. Afterwards, the computed weighted sum s is fed into the neuron. The neuron uses an activation function to transfer the weighted sum s into the output. Usually, the FFNN are trained by using Back propagation (BP) method, BP defines the relative importance of weights for input to a neuron use chain rule of differentiation, through adjusting the weights, the FFNN reduces differences between observed and predicted values.

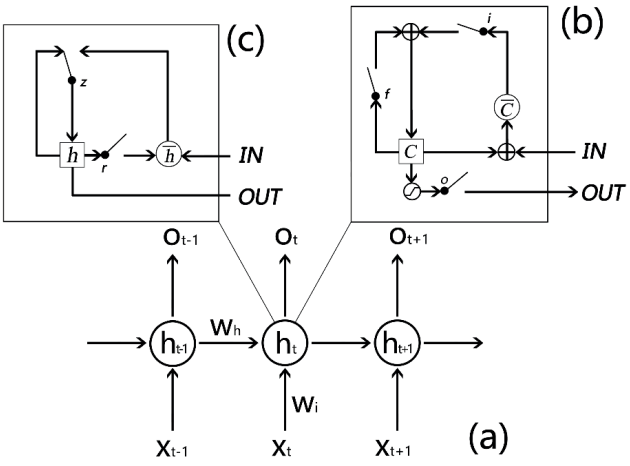


Fig. 4. Schematic of RNN, LSTM and GRU

The memory function of human brain inspired the concept of RNN (Elman, 1990). In addition to the

weighted sum of input values, RNN also takes the state of the hidden neuron at the previous time steps as input for the next time step. As shown in Fig. 4 (a), the hidden neuron output at time step t is calculated by the equation:

$$h_t = f(w_h\, h_{t-1} + w_i\, x_t + b) \tag{3}$$

Where h_t is state of the hidden neuron at the time step t, h_{t-1} is state of the hidden neuron at the time step t-1, w_i and w_h are weights between input values and hidden neurons, and between hidden neurons respectively, $f()$ is the activation function.

Vanishing and exploding gradients, i.e. the partial derivative calculated by the chain rule of differentiation going through the network either get very small and vanish, or get very large and explode, are common problems in the training of the RNN. When train the RNN use BP (also known as backpropagation through time (BPTT)), the chain rule of differentiation not only along the direction of hidden layer and weights, but also along each time steps. Because the error of derivation accumulates through time steps, it will be extremely hard to learn and tune the parameters of the earlier layers.

LSTM was invented to combat with vanishing and exploding gradients problem. Different from traditional RNN, the LSTM uses a memory cell and three gates to control information in the hidden neuron, with on/off of the gates, information can get into, stays in or read from the cell.

Fig. 4 (b) shows the neuron in the hidden layer of LSTM, i, f and o represent the input, forget and output gate respectively. c and $\bar{c}$ denote the memory cell and the new memory cell. The principal of the memory cell in LSTM can be mathematically represented by the following equations:

Input gate:

$$i_t = \sigma_g(W_i * x_t + U_i * h_{t-1} + V_i \circ c_{t-1} + b_i) \tag{12}$$

Forget gate:

$$f_t = \sigma_g(W_f * x_t + U_f * h_{t-1} + V_f \circ c_{t-1} + b_f) \tag{13}$$

Output gate:

$$o_t = \sigma_g(W_o * x_t + U_o * h_{t-1} + V_o \circ c_{t-1} + b_o) \tag{14}$$

Cell state:

$$c_t = f_t \circ c_{t-1} + i_t \circ \overline{c_t} \tag{15}$$

$$\overline{c_t} = \sigma_c(W_c * x_t + U_c * h_{t-1} + b_c) \tag{16}$$

Output vector:

$$h_t = o_t \circ \sigma_h(c_t) \tag{17}$$

Where x_t is the input vector. W, U, V, and b are parameters for weights and bias. ° represents the scalar product of two vectors, σ_g is the sigmoid function, σ_h and σ_c are the hyperbolic tangent function, for a given input z, the output of the hyperbolic tangent function is:

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{18}$$

The GRU is a recent advance in neural networks (Cho et al., 2014; Chung et al., 2014). As a variant of LSTM, the GRU also use a gating mechanism to learn long-term dependencies but its structure is much more simplified compare with LSTM. Fig. 4 (c) shows the gating mechanism of GRU. GRU has only a reset gate and an update gate. The GRU combines the input and forget gates into an update gate to balance between previous activation and the candidate activation. The activation of h at time t depends on h at the previous time and the candidate h (the $\overline{h}$ in Fig. 4 (c)). The update gate z decides how much of the previous memory to keep around. The GRU unit forgets the previously computed state when the reset gate r off.

The GRU is formulated as:

$$z_t = \sigma_g(W_z * x_t + U_z * h_{t-1} + b_z) \tag{19}$$

$$r_t = \sigma_g(W_r * x_t + U_r * h_{t-1} + b_r) \tag{20}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \bar{h_t} \tag{21}$$

$$\bar{h_t} = \sigma_h(W_h * x_t + U_t * (r_t \circ h_{t-1}) + b_h) \tag{22}$$

Where x_t is the input vector, h_t is the output vector, z_t is the update gate vector, h_t is the reset gate vector. W, U and b are parameters for weights and bias. ° represents the scalar product of two vectors, $\sigma(.)$ is the sigmoid function. σ_g represent the sigmoid activation function, σ_h represent the hyperbolic tangent activation function.

In this study, the LSTM, GRU, RNN and FFNN is implemented using Keras. Keras is a Python based high-level deep learning library. It is running on top of TensorFlow or Theano. TensorFlow is used as the backend of Keras in this study. TensorFlow is an open-source deep learning software released by Google in 2015. The SVR is implemented using Python machine learning library Scikit-learn.

2.4 Model performance criteria

The performance of different models is evaluated by three criteria, the root mean square error (RMSE), Nash-Sutcliffe Efficiency (NSE) and the R^2. The calculation of RMSE as shown below:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(Y_i^{obs} - Y_i^{sim}\right)^2}{n}} \tag{11}$$

The NSE is calculated by the following equation:

$$NSE = 1 - \left[\frac{\sum_{i=1}^{n}\left(Y_i^{obs} - Y_i^{sim}\right)^2}{\sum_{i=1}^{n}\left(Y_i^{obs} - Y^{mean}\right)^2}\right] \tag{12}$$

The equation for R^2 is:

$$R^2 = \left[\frac{(\sum_{i=1}^{n}(Y_i^{sim} - Y_{sim}^{mean})\,(Y_i^{obs} - Y^{mean}))^2}{\sum_{i=1}^{n}\left(Y_i^{sim} - Y_{sim}^{mean}\right)^2 \sum_{i=1}^{n}\left(Y_i^{obs} - Y^{mean}\right)^2}\right] \tag{13}$$

In the three above listed equations:

Y_i^{obs} = the i-th observed data.

Y_i^{sim} = the i-th simulated data.

Y^{mean} = mean value of observed data.

Y_{sim}^{mean} = mean value of simualted data.

n = number of observed data.

3. Results

3.1 Calibration of the hydraulic model

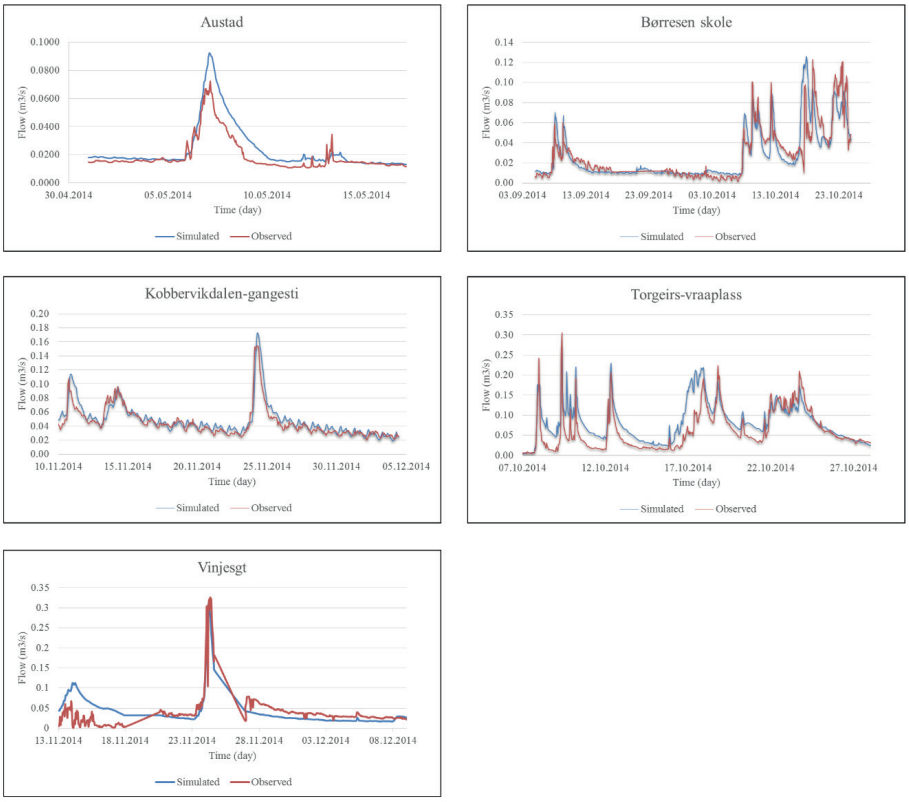


Fig.5. Hydrographs of observed versus hydraulic model simulated flow

Table 1. Calibration results of hydraulic model

Monitoring site	NSE	R^2	RMSE
Austad	0.54	0.91	0.008
Børresen skole	0.68	0.71	0.014
Kobbervikdalen-gangsti	0.80	0.85	0.010
Torgeir-vraaplass	0.51	0.67	0.036

Vinjesgt	0.53	0.57	0.029

Flow data recorded by the regnbyge.no sewer monitoring system is used to calibrate the hydraulic model. Fig. 5 shows the hydrographs of the hydraulic model outputs versus the recorded values at five monitoring sites. It clearly indicates that the simulated values are consistent with the recorded values. Table 1 lists the model performance criteria. All the criteria show acceptable values. Results display in Fig. 5 and Table 1 confirmed a high reliability of the hydraulic. The calibrated model is used in the following scenario simulations.

3.2 Scenario simulations

In the current phase of the Regnbyge 3M project, in order to mitigate overflow from the Muusøya WWTP, the Drammen municipality is planning to construct a storage tank at Landfalloverløpet, however, due to dense buildings and population, the maximum size of the storage tank is restricted to 20,000 m^3, which is insufficient to deal with the current overflow situation. The proposed ICWT solution is expected to compensate insufficient capacity of the storage tank.

Eight scenarios are designed to study individual and combined effects of the storage tank and ICWT on overflow mitigation. The operation of the storage tank and ICWT is simulated using the RTC module in Rosie. When the inflow to the Muusøya WWTP exceeds its maximum capacity, the wastewater is diverted to the storage tank, if the storage tank is full, then the ICWT is activated to convey wastewater to the Brageners catchment. Table 2 gives descriptions of the eight scenarios. The annual long simulation was run continuously from January 01, 2014 to December 31, 2014 to simulate sewer system behaviors.

Table 2. Descriptions of designed scenarios for hydraulic simulation

Scenario	Descriptions
Scenario 1	Baseline scenario, without any overflow control measures
Scenario 2	Only construct a 1,000 m^3 storage tank at Landfalloverløpet
Scenario 3	Only construct a 5,000 m^3 storage tank at Landfalloverløpet
Scenario 4	Only construct a 20,000 m^3 storage tank at Landfalloverløpet

Scenario 5	Only implement the ICWT between the Muusøya catchment and the Brageners catchment
Scenario 6	1,000 m^3 storage tank + ICWT
Scenario 7	5,000 m^3 storage tank + ICWT
Scenario 8	20,000 m^3 storage tank + ICWT

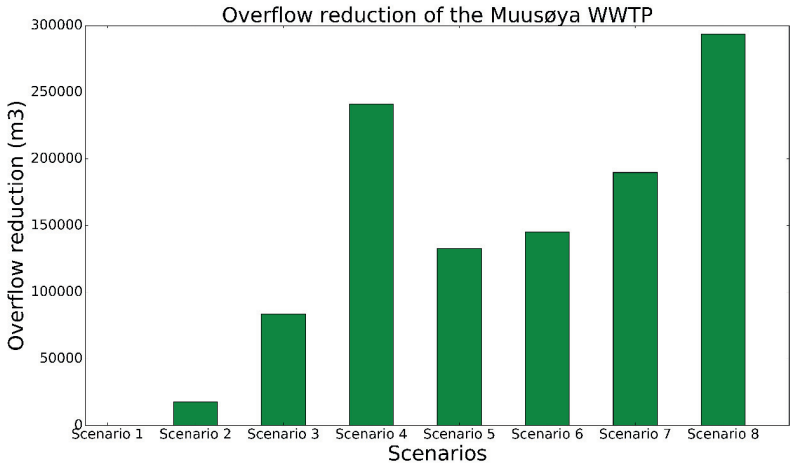


Fig 6. Volume of reduced overflow from the Muusøya WWTP

Fig. 6 displays overflow reduction from the Muusøya WWTP under different scenarios. The overflow from the Muusøya WWTP is 456,171 m^3 for scenario 1. There is a clear tradeoff between storage tank sizes and overflow reduction. For scenario 2 with the smallest storage tank size, the overflow reduction is only 17,576 m^3. A 5000 m^3 storage tank (scenario 3) reduces 83,768 m^3 overflow. The largest storage tank (scenario 4) decreased the overflow more than 50% compare to scenario 1. Compare to ICWT, the storage tank seems less efficient. Scenario 5 with only ICWT reduces more overflow than scenario 3. With same storage tank size, the application of ICWT substantially enhances overflow reduction. The volume of overflow reduction for scenario 2, scenario 3 and scenario 4 are increased from 17,576 m^3, 83,768 m^3, 241,112 m^3 to 145,012 m^3, 189,801 m^3, 293,449 m^3, respectively for scenario 6, scenario 7 and scenario 8.

Table 3. Total overflow of the sewer system

Scenario	Total overflow volume (m^3)
Scenario 1	2,096,668
Scenario 2	2,071,522
Scenario 3	2,001,256
Scenario 4	1,837,773
Scenario 5	1,976,164
Scenario 6	1,954,927
Scenario 7	1,895,657
Scenario 8	1,771,521

Alongside with the overflow from the Muusøya WWTP, the total overflow are also analyzed. Total overflow volume for scenario 1 is approximately 2,096,668 m^3. Scenario 2 only decreases the overflow volume by 1.20%, which is almost not distinct to scenario 1. With the maximum available storage tank dimension (20,000 m^3), the total overflow reduced to 1,837,773 m^3, a reduction of 12.35%. Only implement ICWT reduces total overflow to 1,976,164 m^3, which is more efficient than a 5,000 m^3 storage tank. Compare to scenario 3 and scenario 4, the ICWT further decrease total overflow volume from 2,071,522 m^3 and 2,001,256 m^3 to 1,954,927 m^3 and 1,895,657 m^3. Scenario 8 has the lowest overflow volume with both maximum storage tank size and ICWT.

Table 4. Volume of overflow from the Søren Lemmich pump station

Scenario	Overflow from the Søren Lemmich pump station (m^3)
Scenario 1	32,940
Scenario 5	52,076
Scenario 6	45,079
Scenario 7	34,862
Scenario 8	18,650

One obvious concern about implementing ICWT is whether it will bring extra burden to the Søren Lemmich pump station. Volume of overflow from the Søren Lemmich pump station are investigated. For scenario 5, the total amount of overflow is greater than scenario 1, it means only implement ICWT bring extra overflow to the Søren Lemmich pump station, although the total overflow reduced. Applying the storage tank resulted in a reduction in overflow from the Søren Lemmich pump station. For scenario 6, a 1,000 m^3 storage tank

reduces overflow compare to scenario 5 but the overflow still higher than scenario 1. Overflow volume for scenario 7 is similar to scenario 1, it means a 5,000 m^3 storage tank can relieve the burden of the Søren Lemmich pump station brought by ICWT. For scenario 8, one can observe that total overflow, overflow from the Muusøya WWTP and overflow from the Søren Lemmich pump station are reduced. It can be concluded that ICWT is an efficient overflow mitigating measure, however, in considering the whole system behavior, a storage tank with the size range from 5,000 m^3 to 20,000 m^3 is suggested to implement in conjunction with ICWT.

3.3 Machine learning

The hydraulic simulations clearly demonstrate the viability of ICWT (question 1 and 2). The purpose of this section is to explore the potentiality of machine learning, particularly deep learning in hydrological time series forecasting (question 3). To train the forecasting models, water level data of the Søren Lemmich pump station and corresponding rainfall data collected by the Regnbyge.no system is used. The data are collected from March 20, 2014 to December 1, 2014, there are 73,597 records with a temporal resolution of 5 min. To validate the generalization of the machine learning algorithm, the data are divided into two subsets: training set and testing set. Data from the first 75% were used for training, and the remaining 25% were used for testing. Table 5 gives the summary statistics of the datasets. In the present study, data are scaled to the range [0, 1] before training. After developing the models, the scaled values are rescaled to real values.

Table 5. Summary statistics of the water level data

Model stage	Max water level (m)	Average water level (m)	Standard deviation of water level	Max rainfall (mm/s)	Average rainfall (mm/s)	Standard deviation of rainfall
Training	9.51	2.77	1.44	10.15	2.62	2.11
Testing	9.18	3.17	2.15	9.07	2.68	2.08

Considering technical details of the pump operation, the outputs of the models are a lead-time up to 24 steps (2 hours). The inputs of the models are selected by applying cross-correlation and autocorrelation to the datasets, using the XCORR and AUTOCORR function in MATLAB R2016a. Afterward, the LSTM is implemented through trial and error experiments, different hyper-parameters such as the number of hidden layer, number of hidden neuron, and different optimizers (RMSprop, Adadelta, Adam, Adamax) are tried. The optimal structure of LSTM has two hidden layers with 128 hidden neurons in each layer, the Adam is chosen as the optimizer. Additionally, Dropout is used to prevent overfitting. During training, Dropout (Hinton et al., 2012) temporary discard part of neurons from the neural network. This procedure can be regarded as generating a number of "thinned" neural networks during training but use a single un-thinned neural network in testing (Srivastava et al., 2014). A dropout ratio of 0.35 is selected in this study. The LSTM is trained for 200 epochs with a batch size of 128. After ascertaining the structure of LSTM, the performance of LSTM is compared with other models. To make a fair comparison, the GRU, FFNN and RNN remain the same structure with LSTM. The performance of SVR is subject to the kernel function and the other parameters such as gamma, C and epsilon, the grid search method in the Python Scikit-learn library is used to find the optimal kernel and parameters for SVR. The optimized SVR used in this study is an RBF kernel SVR with a gamma value of 0.5, a C value of 5 and an epsilon value of 0.01.

Table 6. Summary of model performance

Lead time	Performance criteria	Deep learning methods		Traditional methods		SVM
		GRU	LSTM	RNN	FFNN	SVR
20 minutes (4 steps)	R^2	0.8969	0.9014	0.9011	0.9006	0.8674
	RMSE	0.6987	0.6790	0.6774	0.6799	0.7886
	NSE	0.8943	0.9002	0.9006	0.8999	0.8653
40 minutes (8 steps)	R^2	0.8942	0.8965	0.8738	0.8631	0.8573
	RMSE	0.7040	0.6929	0.7438	0.7175	0.8178
	NSE	0.8927	0.8961	0.8628	0.8586	0.8552
1 hour	R^2	0.8900	0.8926	0.8376	0.8169	0.7902

(12 steps)	RMSE	0.7186	0.7091	0.7991	0.8377	0.8959
	NSE	0.8882	0.8912	0.8417	0.8022	0.8174
	R^2	0.8819	0.8832	0.8298	0.8072	0.7942
1 hour 20 minutes (16 steps)	RMSE	0.7435	0.7553	0.8021	0.8518	0.9042
	NSE	0.8803	0.8765	0.8143	0.8143	0.8038
	R^2	0.8721	0.8707	0.7903	0.7643	0.7750
1 hour and 40 minutes (20 steps)	RMSE	0.7856	0.7740	0.8572	0.9258	0.9199
	NSE	0.8664	0.8703	0.7993	0.7524	0.7669
	R^2	0.8620	0.8670	0.7756	0.7538	0.7390
2 hours (24 steps)	RMSE	0.8107	0.7961	0.8972	0.9232	0.9541
	NSE	0.8578	0.8628	0.7854	0.7155	0.7330

The results for the multi-step-ahead water level forecasting within the test period are provided in Table 6. Overall, the LSTM outperforms other models based on R^2, RMSE and NSE criteria. The performance of GRU is slightly worse than the LSTM but the difference is marginal. LSTM and GRU have higher R^2 and NSE values than the other three models for long term predictions. As expected, longer time step caused less accuracy. The models perform consistently well for 4-step and 8-step ahead forecasting, whereas as the forecasting lead time exceeds 1 hour (12 steps), significant differences appear among their performances. The RMSE of the models increases, while the R^2 and NSE decrease, as the forecasting step increases.

The value of the three criteria indicates that the LSTM has the ability to predict water level with long lead time. The performance statistics of the LSTM yields an R^2 of 0.8707 and 0.8670 respectively, an RMSE of 0.7740 and 0.7961 respectively, and an NSE of 0.8703 and 0.8628 respectively for 20-step and 24-step ahead water level prediction. The results prove that with the gating mechanism and memory cell, the LSTM can substantially improve the accuracy of multi-step-ahead water level forecasts.

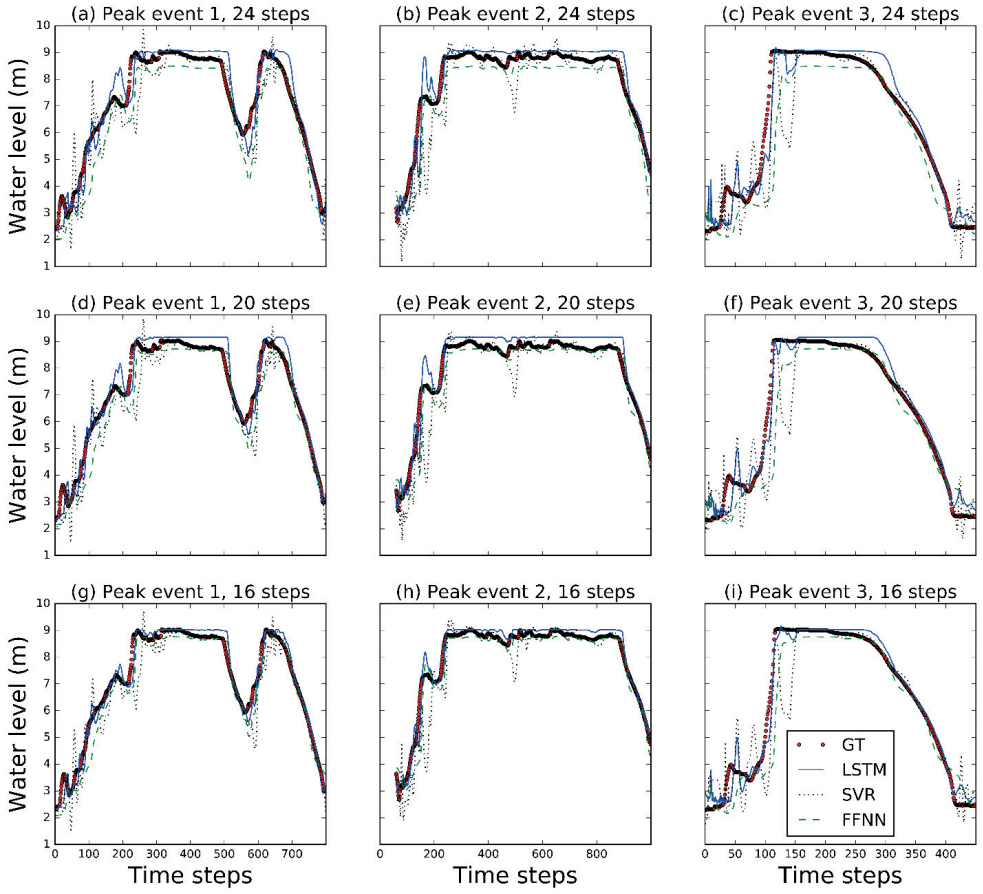


Fig. 7. Hydrographs of observed versus 24, 20 and 16-step ahead forecasted water levels of the LSTM, FFNN and SVR from three peak events

To intuitively illustration the model performance, the hydrographs of Ground Truth (GT) value, LSTM, FFNN and SVR are presented in Fig.7. Fig.7 is observed versus 24, 20 and 16-step ahead forecasted water levels of the three selected models from three peak events. In general, LSTM is still superior to others. Fig.7 shows that the LSTM predicted data are generally consistent with the observed data. It can be seen from Fig.7 that the developed LSTM model is able to predict the water level despite significant variations in water levels during rainfall events. In Fig.7 (a), (c), (d) and (f), one can observe that FFNN have significant

time-lag phenomena, whereas LSTM significantly mitigates this problem, it can be observed that water level increases are anticipated timely by LSTM. Besides, the LSTM is able to capture the major trends and peaks of observations. While FFNN often under-estimates the peak value, as shown in Fig.7 (a)-(c). Compare to LSTM and FFNN, SVR presents strong fluctuations at the rising limbs and has sudden drops at some peaks.

4. Conclusion

This paper delineates a novel ICWT solution for mitigating sewer overflow with minimal construction works. Hydraulic model is developed to test the effectiveness of ICWT and study extra burden received by the Søren Lemmich pump station. To further enhance the operation of the Søren Lemmich pump station, a representative deep learning technology, LSTM, is employed to provide multi-step ahead water level predictions. Several useful findings can be concluded from this study.

1) Most previous studies about sewer overflow control only focus on a single component of the sewer system. To control the overflow in a systematical way, the sewer system should be reconsidered as a whole system to let individual sewer components cooperate in a holistic way. As indicated by hydraulic simulations of eight scenarios, the ICWT could efficiently reduce total overflow from the sewer system and overflow from the Muusøya WWTP, however, the ICWT may bring extra burden to the Søren Lemmich pump station. In considering the whole system behavior, the ICWT is suggested to act in concert with the storage tank.
2) Nonstructural overflow mitigating solutions usually request key facilities to be operated with high sensitivity. The risks of overflow may be increased if accurate hydrological time series prediction information cannot be provided. Five different machine learning models, including deep learning methods (LSTM and GRU), the traditional RNN and FFNN, and the SVR, are compared in this study. Experiments demonstrated that the LSTM is superior to other methods. The LSTM model is capable of forecasting multi-step ahead hydrological time series, and therefore can be a great tool for sewer system managers.

3) Studies relative to sewer systems require the modeling of complex and dynamical urban hydrological processes. There are two approaches of models existed, in the extensively used hydraulic model approach, hydrological/hydraulic principals are explicitly modeled. However, the complicated model construction, calibration and computation make hydraulic models less adequate for real time purpose. On the other hand, the implicit machine learning approach could provide predictions in real time, but it cannot describe the hydrological/hydraulic behavior of sewer system in detail. To solve practical problems, engineers or researchers should consider combining the advantages of both approaches to let them complement each other.
4) The potential power of deep learning is fascinating, ubiquitous sensors are monitoring infrastructures such as sewer system, and collecting a large amount of data. The fusion of sensors, actuators and algorithms will become the backbone of future cities. Data analytics is the sticking point for leveraging intelligent infrastructure management. Given the revolutionary strides made by deep learning in recent years, there are many prospective interests of deep learning for hydrological studies.

Acknowledgements

This work has been supported by the Regnbyge-3M project (grant number 234974), which is granted by the Oslofjord Regional Research Fund. The authors would like to thank the engineers from Rosim Company for their supports.

Conflict of Interest

None

References

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

Chang, F. J., Chen, P. A., Lu, Y. R., Huang, E., & Chang, K. Y. (2014). Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. *Journal of Hydrology*, *517*, 836-846.

Chen, J., Ganigué, R., Liu, Y., and Yuan, Z. (2014). Real-time multistep prediction of sewer flow for online chemical dosing control. *Journal of Environmental Engineering*, *140*(11), 04014037.

Chiang, Y. M., Chang, L. C., Tsai, M. J., Wang, Y. F., and Chang, F. J. (2010). Dynamic neural networks for real-time water level predictions of sewerage systems-covering gauged and ungauged sites. *Hydrology and Earth System Sciences*, *14*(7), 1309-1319.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.

Darsono, S., and Labadie, J. W. (2007). Neural-optimal control algorithm for real-time regulation of in-line storage in combined sewer systems. *Environmental modelling and software*, *22*(9), 1349-1361.

Dettmers T. (2015). Deep Learning in a Nutshell: History and Training. https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-history-training/ Accessed 15 Oct. 2017.

El-Din, A. G., and Smith, D. W. (2002). A neural network model to predict the wastewater inflow incorporating rainfall events. *Water research*, *36*(5), 1115-1126.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179-211.

Ganora, D., Isacco, S., & Claps, P. (2017). Framework for Enhanced Stormwater Management by Optimization of Sewer Pumping Stations. *Journal of Environmental Engineering*, 143(8), 04017025.

Garofalo, G., Giordano, A., Piro, P., Spezzano, G., and Vinci, A. (2017). A distributed real-time approach for mitigating CSO and flooding in urban drainage systems. *Journal of Network and Computer Applications, 78*, 30-42.

Google. (2016). https://research.googleblog.com/2016/09/a-neural-network-for-machine.html Accessed 27 Apr. 2017.

Granata, F., Gargano, R., & de Marinis, G. (2016). Support vector regression for rainfall-runoff modeling in urban drainage: A comparison with the EPA's storm water management model. *Water*, 8(3), 69.

Grum, M., Thornberg, D., Christensen, M. L., Shididi, S. A., and Thirsing, C. (2011). "Full-scale real time control demonstration project in Copenhagen's largest urban drainage catchments." *Proceedings of the 12th international conference on urban drainage*.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580.ors*

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

Hsu, D. (2017). Time Series Forecasting Based on Augmented Long Short-Term Memory. *arXiv preprint arXiv:1707.00666*.

Kanestrøm, P. Ø. (2017). *Traffic flow forecasting with deep learning* (Master's thesis, NTNU).

Lee, E. H., Lee, Y. S., Joo, J. G., Jung, D., & Kim, J. H. (2017). Investigating the Impact of Proactive Pump Operation and Capacity Expansion on Urban Drainage System Resilience. *Journal of Water Resources Planning and Management*, 143(7), 04017024.

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

Lucas, W. C., and Sample, D. J. (2015). Reducing combined sewer overflows by using outlet controls for Green Stormwater Infrastructure: Case study in Richmond, Virginia. *Journal of Hydrology*, *520*, 473-488.

Ma, X., Tao, Z., Wang, Y., Yu, H., and 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.

Marçais, J., & de Dreuzy, J. R. (2017). Prospective interest of deep learning for hydrological inference. *Groundwater*.

Montserrat, A., Bosch, L., Kiser, M. A., Poch, M., and Corominas, L. (2015). Using data from monitoring combined sewer overflows to assess, improve, and maintain combined sewer systems. *Science of the Total Environment*, *505*, 1053-1061.

Mounce, S. R., Shepherd, W., Sailor, G., Shucksmith, J., and Saul, A. J. (2014). Predicting combined sew er overflows chamber depth using artificial neural networks with rainfall radar data. *Water Science and Te chnology*, 69(6), 1326-1333.

Ngo, T. T., Yoo, D. G., Lee, Y. S., & Kim, J. H. (2016). Optimization of upstream detention reservoir facilities for downstream flood mitigation in Urban Areas. *Water*, 8(7), 290.

Nourani, V., Baghanam, A. H., Adamowski, J., and Kisi, O. (2014). "Applications of hybrid wavelet–Artificial Intelligence models in hydrology: A review." *Journal of Hydrology*, 514, 358-377.

Seggelke, K., Rosenwinkel, K. H., Vanrolleghem, P. A., and Krebs, P. (2005). Integrated operation of sewer system and WWTP by simulation-based control of the WWTP inflow. *Water science and technology*, *52*(5), 195-203.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484-489.

Song, X., Kanasugi, H., & Shibasaki, R. (2016, July). DeepTransport: Prediction and Simulation of Human Mobility and Transportation Mode at a Citywide Level. In *IJCAI* (pp. 2618-2624).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.

Wang, Q., Zhou, H., Liang, G., & Xu, H. (2015). Optimal operation of bidirectional inter-basin water transfer-supply system. *Water Resources Management*, 29(9), 3037-3054.

Wu, H., Huang, G., Meng, Q., Zhang, M., & Li, L. (2016). Deep Tunnel for Regulating Combined Sewer Overflow Pollution and Flood Disaster: A Case Study in Guangzhou City, China. *Water, 8(8),* 329.

Yevjevich, V. (2001). Water diversions and interbasin transfers. *Water International*, 26(3), 342-348.

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

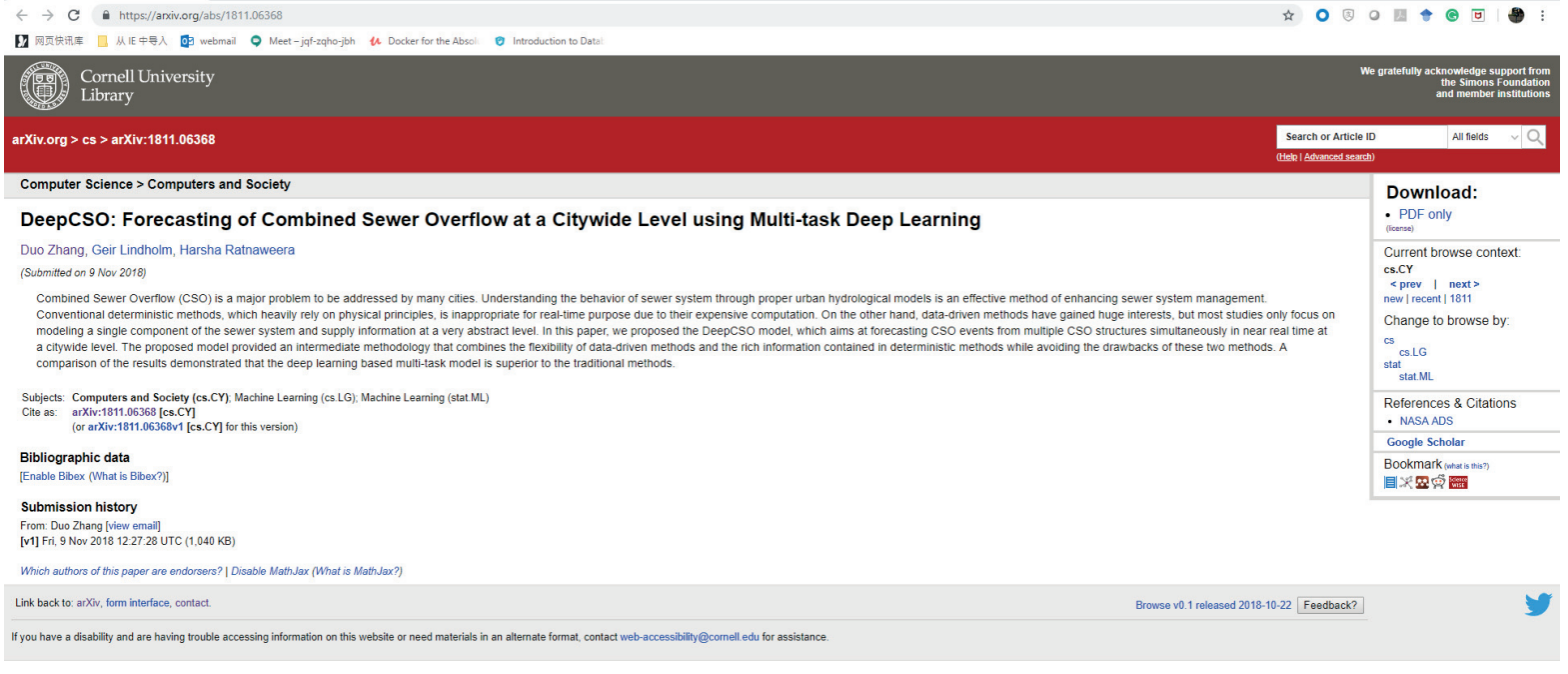
Zhang, Z., Kusiak, A., Zeng, Y., & Wei, X. (2016). Modeling and optimization of a wastewater pumping system with data-mining methods. *Applied Energy*, *164*, 303-311.

Paper V

Zhang, D., Lindholm, G., & Ratnaweera, H. (2018). DeepCSO: Forecasting of combined sewer overflow at a citywide level using multi-task deep learning. *arXiv:1811.06368 [cs.CY]*. (*preprint in arXiv*)

Access to full paper at:

https://arxiv.org/abs/1811.06368



https://arxiv.org/abs/1811.06368

网页快讯库 从IE中导入 webmail Meet – jqf-zqho-jbh Docker for the Absol Introduction to Data

Cornell University Library

We gratefully acknowledge support from the Simons Foundation and member institutions

arXiv.org > cs > arXiv:1811.06368

Search or Article ID | All fields

(Help | Advanced search)

Computer Science > Computers and Society

DeepCSO: Forecasting of Combined Sewer Overflow at a Citywide Level using Multi-task Deep Learning

Duo Zhang, Geir Lindholm, Harsha Ratnaweera

(Submitted on 9 Nov 2018)

Combined Sewer Overflow (CSO) is a major problem to be addressed by many cities. Understanding the behavior of sewer system through proper urban hydrological models is an effective method of enhancing sewer system management. Conventional deterministic methods, which heavily rely on physical principles, is inappropriate for real-time purpose due to their expensive computation. On the other hand, data-driven methods have gained huge interests, but most studies only focus on modeling a single component of the sewer system and supply information at a very abstract level. In this paper, we proposed the DeepCSO model, which aims at forecasting CSO events from multiple CSO structures simultaneously in near real time at a citywide level. The proposed model provided an intermediate methodology that combines the flexibility of data-driven methods and the rich information contained in deterministic methods while avoiding the drawbacks of these two methods. A comparison of the results demonstrated that the deep learning based multi-task model is superior to the traditional methods.

Subjects: **Computers and Society (cs.CY)**; Machine Learning (cs.LG); Machine Learning (stat.ML)
Cite as: **arXiv:1811.06368 [cs.CY]**
(or **arXiv:1811.06368v1 [cs.CY]** for this version)

Bibliographic data

[Enable Bibex (What is Bibex?)]

Submission history

From: Duo Zhang [view email]
[v1] Fri, 9 Nov 2018 12:27:28 UTC (1,040 KB)

Which authors of this paper are endorsers? | Disable MathJax (What is MathJax?)

Download:

- PDF only (license)

Current browse context:
cs.CY
< prev | next >
new | recent | 1811

Change to browse by:
cs
cs.LG
stat
stat.ML

References & Citations
- NASA ADS

Google Scholar

Bookmark (what is this?)

Link back to: arXiv, form interface, contact.

Browse v0.1 released 2018-10-22 Feedback?

If you have a disability and are having trouble accessing information on this website or need materials in an alternate format, contact web-accessibility@cornell.edu for assistance.

DeepCSO: Forecasting of Combined Sewer Overflow at a Citywide Level using Multi-task Deep Learning

Duo Zhang[1]; Geir Lindholm[2]; Harsha Ratnaweera[1]

1. Faculty of Sciences and Technology, Norwegian University of Life Sciences, 1432, Ås, Norway
2. Rosim AS, Brobekkveien 80, 0582, Oslo, Norway

Abstract: Combined Sewer Overflow (CSO) is a major problem to be addressed by many cities. Understanding the behavior of sewer system through proper urban hydrological models is an effective method of enhancing sewer system management. Conventional deterministic methods, which heavily rely on physical principles, is inappropriate for real-time purpose due to their expensive computation. On the other hand, data-driven methods have gained huge interests, but most studies only focus on modeling a single component of the sewer system and supply information at a very abstract level. In this paper, we proposed the DeepCSO model, which aims at forecasting CSO events from multiple CSO structures simultaneously in near real time at a citywide level. The proposed model provided an intermediate methodology that combines the flexibility of data-driven methods and the rich information contained in deterministic methods while avoiding the drawbacks of these two methods. A comparison of the results demonstrated that the deep learning based multi-task model is superior to the traditional methods.

Keywords: Combined sewer overflow; Long short-term memory; Deep learning; Urban hydrological model; Multi-task learning

Author names and affiliations:

Duo Zhang (corresponding author):

Ph.D. candidate, Faculty of Science and Technology, Norwegian University of Life Sciences, 1432, Ås, Norway.

Email: Duo.Zhang@nmbu.no

Geir Lindholm:

CEO, Rosim AS, Brobekkveien 80, 0582, Oslo, Norway.

Email: geir@rosim.no

Harsha Ratnaweera:

Professor, Faculty of Science and Technology, Norwegian University of Life Sciences, 1432, Ås, Norway.

Email: Harsha.Ratnaweera@nmbu.no

1. Introduction

In recent years, increased impermeable surface, extreme rainfall event and urbanization have resulted in more frequent Combined Sewer Overflow (CSO). Owing to the demand for on-time information, a lot of cities have developed the surveillance system to offer insights into the performance of CSO structures (Montserrat et al. 2015; Power 2016; Ayyeka 2017). Intelligent urban infrastructures such as smart sewer system will become the backbone of future cities (Jaokar 2015). To give sewer surveillance system 'intelligence', both data acquisition and extract useful information from the collected data are indispensable. In this context, developing a versatile urban hydrological model is imperative to capture useful information from a large amount of collected data and to enhance various tasks. Indeed, how to effectively leverage the data collected by ubiquitous infrastructure sensors through proper modeling techniques has become a sticking point for future intelligent sewer system management (Wu & Rahman 2017).

In general, methods involved in urban hydrological modeling can be classified into two major categories: deterministic and data-driven methods (Nourani et al. 2014). Admittedly, deterministic methods can provide fully detailed information for sewer systems. However, deterministic methods require sophisticated foreknowledge about the sewer system, incorporate a huge number of parameters and the simulation are based on numerical methods. These characteristics make the model construction, calibration and computation of deterministic methods extremely complex. Therefore, deterministic methods are inappropriate for application in real time purpose (El-Din & Smith 2002). Another disadvantage of deterministic methods is that the computation of deterministic methods is based on given rainfall, it cannot provide future hydrological information (Chiang et al. 2010). Accurate hydrological time series forecasting could support engineers' decision-making, pinpoint the vulnerable part of the sewer system in advance, warm up sewer control facilities or early warning peak events. Hence, hydrological time series forecasting is often a prerequisite for successful sewer system control.

By contrast, data-driven methods are flexible in model development, it avoids complicated

hydraulic/hydrological theories by learning from data without human intervention. Moreover, Data-driven methods can produce future hydrological data by being fed with current and previous data. Many research efforts have been done to enrich data-driven approaches for hydrological time series forecasting. Due to capable of handling non-linear and non-stationary problems, the Artificial Intelligence (AI) methods have shown promise among numerous data-driven approaches. A particularly popular sub-set of AI used for hydrological time series forecasting is the machine learning. Typical machine learning algorithms include Support Vector Regression (SVR) and various artificial neural network (ANN) structures. Unlike shallow ANN structures, deep learning models extract high-level abstractions in data through processing data by the internal layers, thus, deep learning is able to provide efficient high-dimensional interpolators that cope with multiple scales and heterogeneous information (Marçais & de Dreuzy 2017). Deep learning has made revolutionary strides in recent years, typical examples of deep learning include AlphaGo (Silver et al. 2016) and the latest Google translation system (Google 2016). Deep learning method has also shown its superior performance compare to traditional methods on traffic time series forecasting (Hsu 2017; Ma et al. 2015; Kanestrøm 2017), and hence employed by Uber (Laptev et al. 2017) for their ride request forecasting system.

Although as the most promising data-driven methods, machine learning/deep learning has presented its power in many studies, we could still find two major deficiencies by summarizing previous researches. First, the success of deep learning in both academia and industry suggests a natural prospective interest for the use of deep learning for hydrological time series forecasting, but there are very few reports studied the performance of deep learning on hydrological data. Second, in order to forecast urban hydrological time series in near real time, data-driven methods seem a good alternative to deterministic methods, although the latter method could provide fully detailed information. However, in most urban hydrological studies, researchers only focus on predicting hydrological time series for a single component of the sewer system. This kind of model can only provide information at a very abstract level. One may develop models separately for individual parts of the sewer system, but this approach neglecting the existed physical correlation of sewer components. Moreover, a system with many independent models is less efficient due

to redundant information contained in these models (Bezuglov et al. 2016), maintain such a system also requires more works to adjust hyper-parameters of individual neural networks.

Therefore, the purpose of this study is to find an intermediate methodology that combines the flexibility of data-driven methods and the rich information contained in deterministic methods, while avoiding the drawbacks of these two methods. To overcome aforementioned shortages of different models, we consult the principal of Multi-Task Learning (MTL, Zhang & Yang 2017). MTL aims at solving multiple tasks at the same time. If all the tasks or at least a subset of these tasks is assumed to be related to each other, the MTL approach usually could generalize better than single task model by sharing representations between related tasks (Ruder 2017). Deep learning becomes more and more popular in MTL. Usually, this approach uses the first several hidden layers to learn common representations for multiple tasks and then generate outputs for each task. In considering the spatiotemporal correlations of the sewer system in which the hydrological behavior of one part of the sewer system is related to its previous status, rainfall and those upstream or even downstream parts, we propose the DeepCSO model, which aims at forecasting the hydrological time series of multiple CSO structures simultaneously using deep learning. The main characteristics of deterministic methods, data-driven methods and the proposed DeepCSO model are summarized in Table 1. The methodology is demonstrated with a case study of a sewer system in Drammen, Norway.

Table 1. Pros and cons of deterministic methods, data-driven methods and DeepCSO

	Deterministic methods	Data-driven methods	DeepCSO (this study)
Data required	• Detailed information about the studied sewer system and catchment for model development • Rainfall data for model computation • Sewer hydrological data such as flow or water level for model calibration	Usually only requires rainfall data or sewer hydrological data	Similar to the data-driven methods, the only difference is DeepCSO requires data from multiple CSO structures

Principals adopted	Hydraulic/hydrological principals, e.g: • Time–Area (T-A) method • Rainfall Dependent Infiltration/Inflow (RDII) • Saint-Venant continuity and momentum equations	Different statistical principals according to different algorithms	Use the state of the art branch of data-driven methods, deep learning
Model construction	Very complex and time-consuming, must specify properties of every sub-catchments, pipelines and sewer nodes	Relatively easy, the model could learn from data without human intervention	Need more data preprocessing and preparation compare with traditional data-driven methods
Model calibration	Very complex and time-consuming, there are numerous parameters for different sub-catchments and sewer components must be adjusted manually	Relatively easy, the model has much fewer parameters compare with the deterministic methods	Relatively easy, the model has much fewer parameters compare with the deterministic methods
Model computation	Slow, for large sewer systems, require hours or even days to run	Fast, in near real-time	Fast, in near real-time
Model Output	Detailed current or previous hydrological information about the sewer system, suitable to perform scenario analyzes for hydraulic planning and design.	Could forecast in near real time, but only for a single sewer component	Could forecast hydrological information for several CSO structures. It balances the pros and cons of deterministic methods and data-driven methods

2. Methods and materials

2.1 Case study area

The Drammen city is a coastal city in the Buskerud County, southeast Norway. Drammen has a predominantly cold climate. The average annual precipitation of Drammen is approximately 731 mm, and the precipitation mainly occurs between June and October.

The sewer system of Drammen serves around 150,000 inhabitants, the drainage area of the sewer system is about 15 km^2, and the total length of the sewer system is approximately 500 km. The sewer system of

Drammen roughly consists of 65% combined sewer system and 35 % separate sewer system. Most of the combined sewer system distributes along the Drammen Fjord. The downtown area of Drammen has a denser population and most of the important infrastructures such as the train station, shopping center, and the stadium are located in this area. During heavy rainfall events, the combined sewer system in the downtown area discharges overflows directly into the Drammen Fjord though CSO structures, cause heavy pollution. In order to mitigate impacts of CSO, the Drammen city initialized the Regnbyge 3M project. The ultimate goal of this project is to manage the sewer system with intelligent monitoring, modeling and control solutions. Developing an accurate CSO forecasting model is a vital part of the Regnbyge 3M project.

2.2 Data description

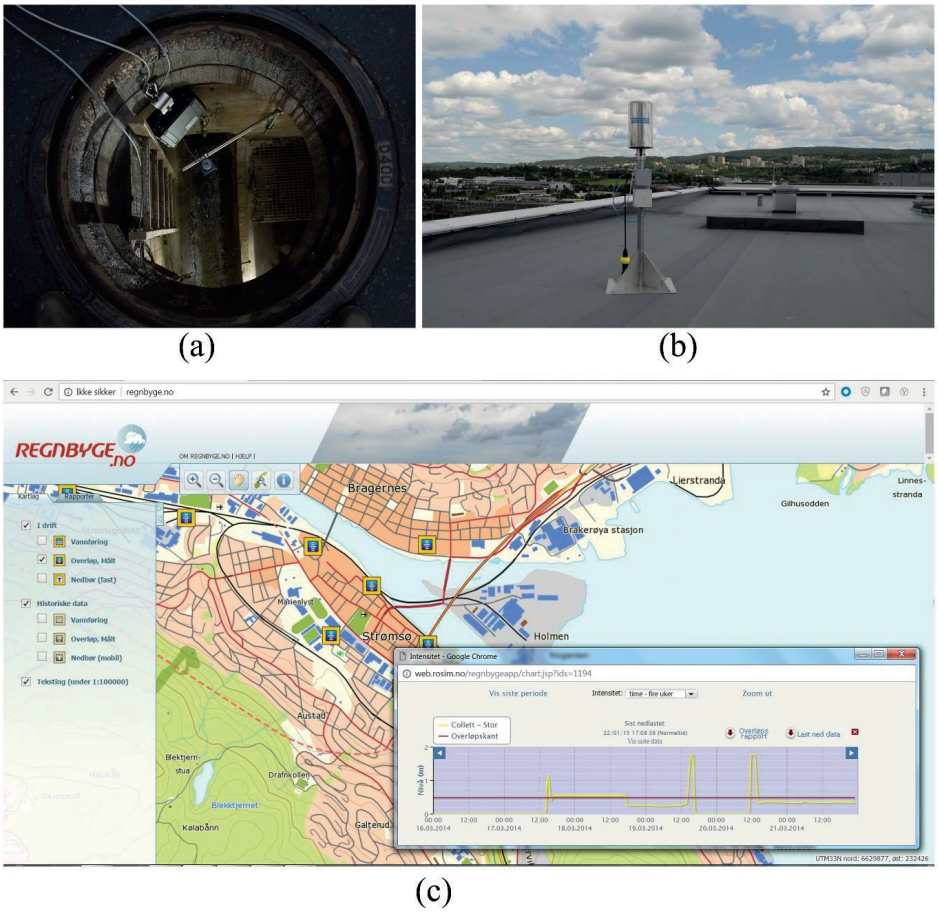


Fig.1. The Regnbyge.no IoT. Fig. 1 (a) shows an ultrasonic water level sensor mounted on the top of a CSO structure, Fig. 1 (b) is a rain gauge. Fig. 1 (c) demonstrates the user interface of the Regnbyge.no IoT

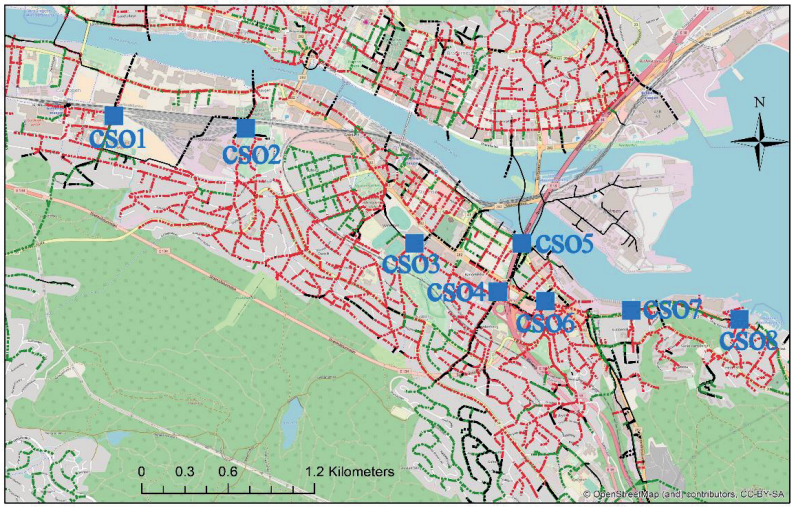


Fig.2. Overview of the studied CSO structures in Drammen, Norway

In the first phase of the Regnbyge 3M project, we implemented an IoT system, called Regnbyge.no, to monitor the CSO structures and collect data for further model development. The Regnbyge.no consists of ultrasonic water level sensors produced by NIVUS GmbH, Germany and rain gauges. The collected water level and rainfall data are transmitted to Rosim AS, Norway. A spatial database and a web-based geographic information system (Web-GIS) is designed to manage the collected data and provide a user interface. Fig. 1 displays major components of the Regnbyge.no IoT.

Water level data and rainfall data from 8 CSO structures located in the downtown of Drammen, which collected from March/19/2014 to September/27/2014 from the Regnbyge.no IoT, is used for model development. Fig 2. shows the distribution of the studied CSO structures (denoted by squares). The collected data contained 27756 records with a temporal resolution of 10 min for each CSO structure. Table 2 is summary statistics of the water level data from 8 CSO structures. To avoid too many Norwegian characters, all the CSO structures will refer to their CSO ID hereafter in this paper.

Table 2. Summary statistics of water level data from the studied CSO structures

CSO ID	CSO name in Norwegian	Max water level (m)	Mean water level (m)	Standard deviation (m)
CSO 1	Vintergata	1.66	0.28	0.36
CSO 2	Smithestrøm	1.14	0.56	0.38
CSO 3	Drammenshallen	1.14	0.12	0.13
CSO 4	Collet	1.77	0.2	0.11
CSO 5	Motorveibrua	3.3	1.46	1.04
CSO 6	Gåsevadet	1.15	0.23	0.27
CSO 7	Havnegata	0.75	0.06	0.07
CSO 8	Skomakergata	0.9	0.18	0.17

2.3 Deep learning

Analogous to a human brain, ANN uses hierarchically organized networks that are consisted of weighted connected neurons to perform complex tasks such as prediction. Feed Forward Neural Network (FFNN) is one of the most traditional ANN architecture.

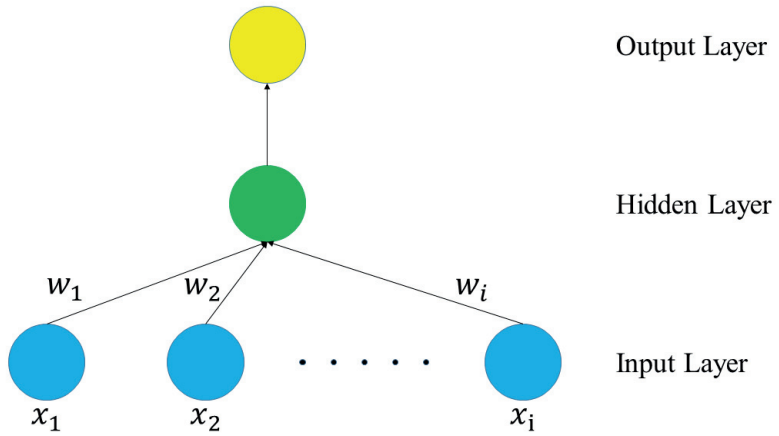


Fig. 3. Schematic of FFNN

Fig. 3 is an example of a three-layer FFNN, which is comprised of input layer, hidden layer, and output layer. Neurons in the input layer receive input values. Afterward, neurons in the hidden layer link the input

neurons and the output neurons, as well as provide nonlinearity to the network. Outputs from neurons are multiplied by the connection weights and bias before fed into the neurons in the next layer. The connection weights determine the strength of the relationship between connected neurons. Neurons in the hidden layers and output layer sum all the inputs and convert the summed inputs into output value according to the activation function. This process can be mathematically represented as:

$$s = f(\sum_{i=1}^{n} w_i x_i + b) \quad (1)$$

Where w_i represents the weights, x_i is the inputs, b is the bias and $f()$ is the activation function.

FFNN usually trained by using Backpropagation (BP) method, BP defines how the input data patterns are related to output data. The algorithm uses the chain rule of differentiation to determine how the network should adjust the weights, thus reduces errors between observed and predicted values.

A major difference between FFNN and the human brain is that the FFNN doesn't have 'memory'. With connections between hidden neurons, RNN is biologically more plausible than FFNN. Because RNN can process inputs use their internal memory, hence it is particularly applicable to tasks such as time series forecasting.

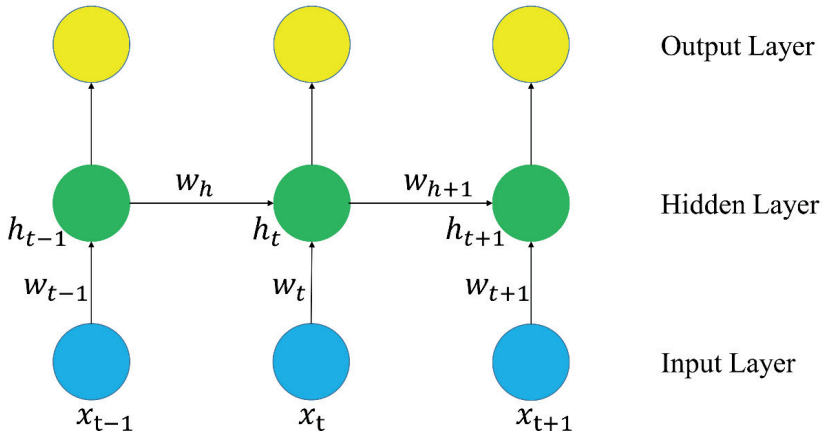


Fig.4. Schematic of RNN

As shown in Fig. 4, in addition to the weighted sum of input values, RNN also takes the state of the hidden neuron at the previous time steps as input for the next time step. In this way, RNN passing message to a successor. The neuron output of RNN at time step t is calculated by the equation:

$$h_t = f(w_h\, h_{t-1} + w_t\, x_t + b) \tag{2}$$

Where h_t is state of the hidden neuron at the time step t, h_{t-1} is state of the hidden neuron at the time step t-1, w_{t-1}, w_t and w_{t+1} are weights between input values and hidden neurons, w_h and w_{h+1} are weights between hidden neurons, $f()$ is the activation function.

The training of RNN use a variant of BP called backpropagation through time (BPTT), it means the algorithm calculates not only the partial derivative along the direction of the hidden layer but also along each time step. Because the error of derivation accumulates through time steps, the partial derivative going through the network either get very small and vanish, or get very large and explode. In this case, it will be extremely hard to learn and tune the parameters of the earlier layers. This problem is known as vanishing

and exploding gradients problem.

To address these drawbacks, Hochreiter & Schmidhuber (1997) developed a special RNN, Long Short-Term Memory. Different from traditional RNN, the LSTM replace ordinary hidden neurons with a series of memory blocks. Each memory block is composed of a memory cell and three gates.

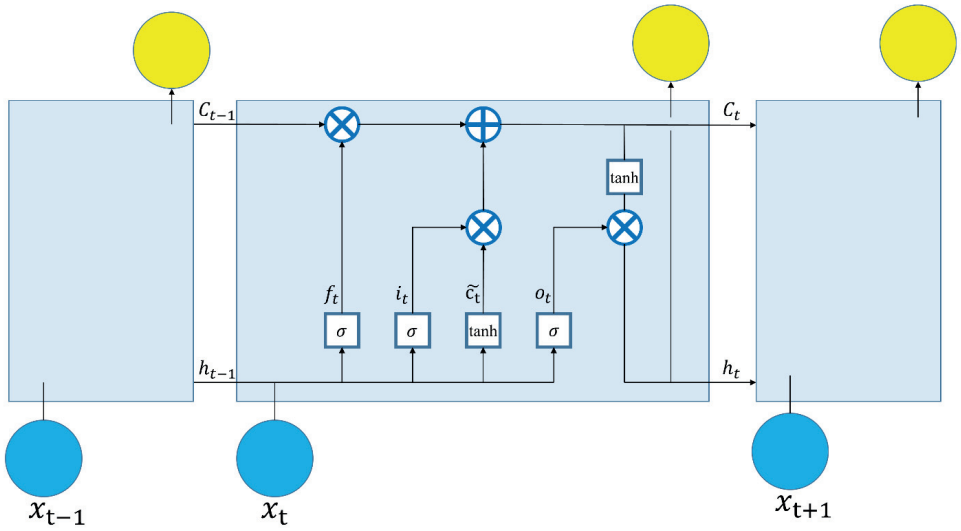


Fig.5. Schematic of LSTM

Fig. 5 gives an example of an LSTM memory block. The principal of the memory cell in LSTM can be mathematically represented by the following equations:

The input gate is designed for permits inputs to modify the memory cell state:

$$i_t = \sigma_g(W_i * [x_t, h_{t-1}] + b_i) \tag{3}$$

The forget gate is used to reset memory blocks, thereby preventing the cell status from containing redundant information:

$$f_t = \sigma_g(W_f * [x_t, h_{t-1}] + b_f) \tag{4}$$

The output gate allows or obstructs the cell state from affecting other neurons:

$$o_t = \sigma_g(W_o * [x_t , h_{t-1}] + b_o) \tag{5}$$

The memory cell can impede outside interference and remain unchanged from one-time step to another, thus allows the LSTM to learn time series with long spans:

$$c_t = f_t \circ c_{t-1} + i_t \circ \bar{c_t} \tag{6}$$

$$\bar{c_t} = \sigma_c(W_c * [x_t , h_{t-1}] + b_c) \tag{7}$$

Output vector:

$$h_t = o_t \circ \sigma_h(c_t) \tag{8}$$

Where x_t is the input vector. W and b are parameters for weights and bias. ° represents the scalar product of two vectors, σ_g is the sigmoid function, σ_h and σ_c are the hyperbolic tangent function (denoted as 'tanh' in Fig.5), for a given input z, the output of the hyperbolic tangent function is:

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{9}$$

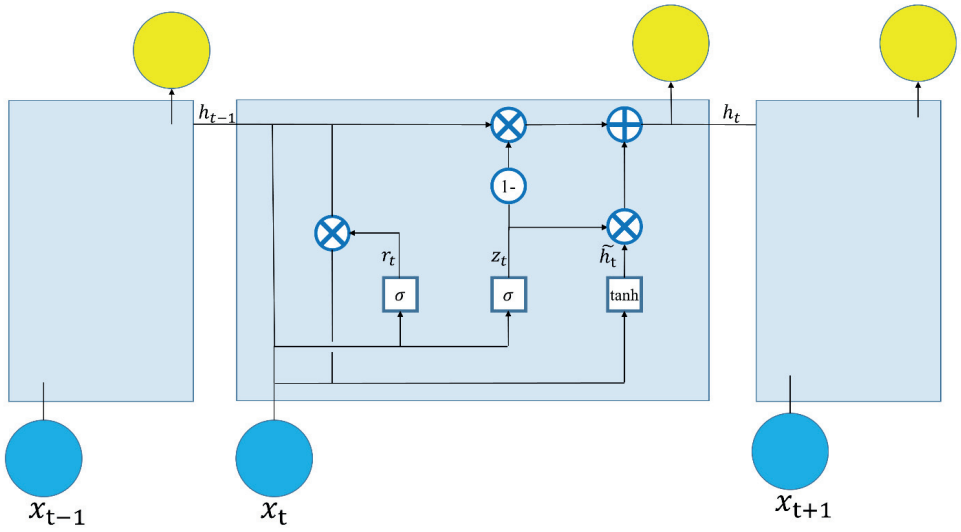


Fig. 6. Schematic of GRU

The major drawback of LSTM is its complexity. Stimulated by the success of LSTM, how to simplify LSTM thereby become a highly researched topic in the field of computer science. The GRU is a recent advance in neural networks (Cho et al. 2014). As a variant of LSTM, the GRU also uses a gating mechanism to learn long-term dependencies but its structure is much more simplified compare with LSTM. Fig. 6 shows the gating mechanism of GRU. GRU has only a reset gate and an update gate. The GRU combines the input and forget gates into an update gate to balance between previous activation and the candidate activation. The activation of h at time t depends on h at the previous time and the candidate h (the $\bar{h}$ in Fig. 6). The update gate z decides how much of the previous memory to keep around. The GRU unit forgets the previously computed state when the reset gate is off.

The GRU is formulated as:

$$z_t = \sigma_g(W_z * [x_t , h_{t-1}] + b_z) \tag{10}$$

$$r_t = \sigma_g(W_r * [x_t , h_{t-1}] + b_r) \tag{11}$$

$$h_t = z_t \circ \bar{h_t} + (1 - z_t) \circ h_{t-1} \tag{12}$$

$$\bar{h_t} = \sigma_h(W_h * [x_t , (r_t \circ h_{t-1})] + b_h) \tag{13}$$

Where x_t is the input vector, h_t is the output vector, z_t is the update gate vector, h_t is the reset gate vector. W and b are parameters for weights and bias. $\circ$ represents the scalar product of two vectors, $\sigma(.)$ is the sigmoid function. σ_g represent the sigmoid activation function, σ_h represent the hyperbolic tangent activation function.

In this study, the LSTM, GRU, RNN and FFNN are implemented using Keras. Keras is a Python-based high-level deep learning library. It is running on top of TensorFlow or Theano. TensorFlow is used as the backend of Keras in this study. TensorFlow is an open-source deep learning software released by Google in 2015. Other Python-based machine learning libraries, includes Pandas, NumPy, Scikit-learn and

Matplotlib are also used. Specifically, Pandas and NumPy are used to load the dataset as the data frame and prepare the raw data in the format of the desired array. Scikit-learn is used for model selection and preprocessing, such as tuning parameters and data normalization. Matplotlib is used for visualization.

2.4 Model performance metrics

In this study, we use three metrics, Coefficient of Correlation (CC), Root Mean Squared Error (RMSE) and Nash–Sutcliffe Efficiency (NSE) to evaluate the performance of different models.

CC calculates the combined dispersion against the single dispersion of the observed and predicted values. The equation for the CC is:

$$CC = \frac{\sum_{i=1}^{n}(Y_i^{sim} - Y_{sim}^{mean})(Y_i^{obs} - Y^{mean})}{\sqrt{\sum_{i=1}^{n}\left(Y_i^{sim} - Y_{sim}^{mean}\right)^2}\sqrt{\sum_{i=1}^{n}\left(Y_i^{obs} - Y^{mean}\right)^2}} \tag{14}$$

The CC values range between -1 and 1, which describes how much of the observed dispersion is explained by the prediction. CC value higher than 0.7 indicates variables are highly correlated.

Root mean squared error (RMSE) is one of the most common metrics used to measure accuracy for continuous variables such as time series. The calculation of RMSE as shown below:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(Y_i^{obs} - Y_i^{sim}\right)^2}{n}} \tag{15}$$

RMSE value of 0 means a perfect fit between observed and predicted values.

NSE is a parameter that determines the relative importance of residual variance (noise) compare to the variance in the measured data (information). The range of NSE lies between $-\infty$ and 1. An NSE value of lower than zero indicates that the mean value of the observed time series would have been a better predictor than the model, values between 0.0 and 1.0 is generally acceptable, higher than 0.5 is considered to be a good value for NSE. The NSE is calculated by the following equation:

$$NSE = 1 - \left[\frac{\sum_{i=1}^{n}\left(Y_i^{obs} - Y_i^{sim}\right)^2}{\sum_{i=1}^{n}\left(Y_i^{obs} - Y^{mean}\right)^2}\right] \quad (16)$$

In above-listed equations:

Y_i^{obs} = the i-th observed data.

Y_i^{sim} = the i-th simulated data.

Y^{mean} = mean value of observed data.

Y_{sim}^{mean} = mean value of simulated data.

n = number of data

3. Results and discussion

According to the definition of MTL, the DeepCSO model has 8 outputs with one for each CSO structure. Autocorrelation and cross-correlation (Mounce et al. 2014) analysis are performed to select input CSO water level data and rainfall respectively for the model. The hidden layer is particularly important as it transforms original representation into common features of tasks (Zhang & Yang 2017). There are no direct experiences about sewer system, but according to several comparative MTL studies about traffic forecasting (Song et al. 2016), air pollutants prediction (Li et al. 2017) and storm surge prediction (Bezuglov et al. 2016), the hidden layer is designed as two hidden layers and a dense layer (fully connected layer of neurons). The first two hidden layers are initially used to extract representative features from CSO water level data. Next, dense layer is used to generate the prediction outputs. The structure of the proposed DeepCSO model is shown in Fig.7:

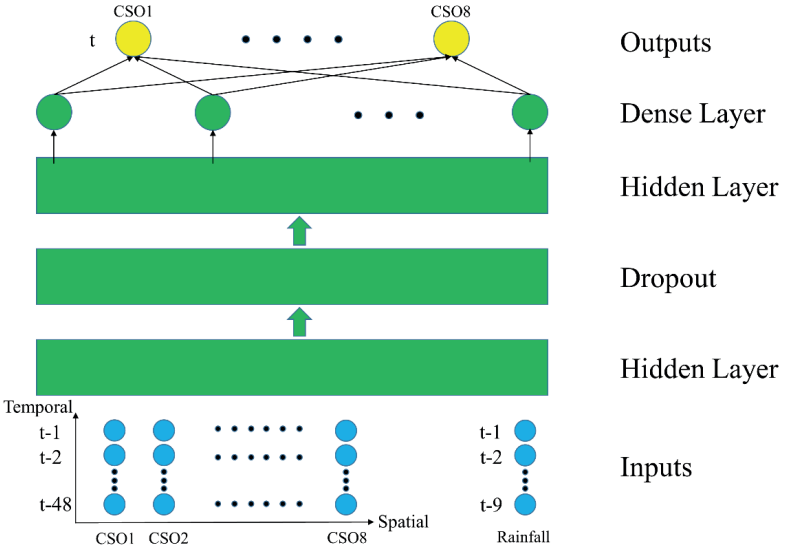


Fig.7. Architecture of the proposed DeepCSO model

We selected 80 percent of the data as the training set, and the remaining 20 percent was used as the test set. Data are scaled to the range [0, 1] before training. Because previous studies suggested that LSTM usually perform better than other methods, so that we first designed the DeepCSO model with two LSTM layers and a dense layer. Several hyperparameters should be preset before building the model, including the number of nodes in each LSTM layer, batch size, optimizer and drop out ratio. We investigated the effect of each parameter while keeping the other parameters fixed, to find the optimum hyperparameters. Table 3 gives an overview of the investigated hyperparameters and the optimal values. For simplicity, the number of nodes in each LSTM layer was set to same value.

Table 3. Studied hyper parameters

Hyper parameter	Candidate values	Optimal value
Number of hidden neurons	32, 64, 128, 256, 512, 1024	512
Batch size	128, 256, 512, 1024, 2048	1024
Optimizer	RMSprop, Adadelta, Adagrad, Adam, Adamax, Nadam	Adam
Drop out ratio	0.5, 0.35, 0.2, 0 (no drop out)	0.2

Afterward, the performance of LSTM is compared with another deep learning method (GRU), traditional RNN, traditional neural network (FFNN) and the most common single task time series method (SVR). To make a fair comparison, the GRU, FFNN and RNN remain the same structure with LSTM. The SVR model is developed for each station separately using input data from a single CSO.

Table 4. Performance of single step ahead predictions of different methods

Performance metrics	CSO ID	Models				
		RNN	GRU	LSTM	FFNN	SVR
CC	CSO 1	**0.9775**	0.9762	0.977	0.9668	0.8177
	CSO 2	0.9935	0.994	**0.9942**	0.9758	0.9584
	CSO 3	**0.9778**	0.9701	0.9774	0.9151	0.8561
	CSO 4	0.9321	0.922	**0.9328**	0.7513	0.9201
	CSO 5	0.9922	**0.9944**	0.9915	0.9838	0.8144
	CSO 6	0.9968	0.997	**0.9974**	0.9848	0.9744
	CSO 7	0.9791	**0.9795**	0.9794	0.9502	0.9778
	CSO 8	0.9803	0.9794	**0.9805**	0.9408	0.9324
RMSE	CSO 1	**0.0812**	0.0831	0.082	0.109	0.2275
	CSO 2	0.0273	0.0288	**0.0243**	0.0704	0.0648
	CSO 3	0.021	0.0208	**0.0202**	0.0369	0.0532
	CSO 4	**0.0251**	0.0257	0.0285	0.0524	0.0281
	CSO 5	0.1738	**0.0842**	0.1019	0.1698	0.6815
	CSO 6	0.0264	**0.0247**	0.0264	0.0742	0.066
	CSO 7	0.0145	0.0143	**0.0141**	0.0242	0.015
	CSO 8	0.033	0.0314	**0.0301**	0.0544	0.0544
NSE	CSO 1	**0.955**	0.9529	0.9541	0.9188	0.6465
	CSO 2	0.9845	0.9828	**0.9878**	0.8971	0.9127
	CSO 3	0.9387	0.9397	**0.9432**	0.8107	0.6067
	CSO 4	**0.8553**	0.8489	0.8138	0.3715	0.8189
	CSO 5	0.935	**0.9847**	0.9777	0.9379	0.9141
	CSO 6	0.9914	**0.9924**	0.9913	0.9318	0.946
	CSO 7	0.9546	0.9559	**0.9573**	0.8737	0.9517
	CSO 8	0.9514	0.956	**0.9595**	0.8682	0.868

Table 4 illustrates the results of the three performance metrics of different models for the 8 CSO structures. The highest CC and NSE values and lowest RMSE values are marked in bold. It clearly indicates that for the single step ahead forecasting, all the five models could achieve good accuracy, but LSTM could get relatively better performances, the GRU and RNN also perform well.

Table 5. Performance of three-step ahead predictions of different methods

Performance metrics	CSO ID	Models				
		RNN	GRU	LSTM	FFNN	SVR
CC	CSO 1	**0.9637**	0.9543	0.9538	0.9412	0.7667
	CSO 2	0.9749	**0.9782**	0.9731	0.9623	0.934
	CSO 3	0.8946	**0.9069**	0.904	0.8528	0.7638
	CSO 4	0.8334	0.8488	**0.8491**	0.7161	0.8253
	CSO 5	0.9867	**0.9908**	0.9878	0.9799	0.9686
	CSO 6	0.9874	**0.9892**	0.988	0.9762	0.9643
	CSO 7	0.904	0.9254	0.9186	0.9027	**0.9273**
	CSO 8	0.9324	**0.9384**	0.9348	0.9073	0.8662
RMSE	CSO 1	0.1164	0.1168	**0.1152**	0.1453	0.2445
	CSO 2	0.0537	**0.0504**	0.0618	0.0864	0.0785
	CSO 3	0.0388	0.0413	**0.0385**	0.0444	0.0567
	CSO 4	0.0368	**0.0353**	0.0389	0.047	0.3841
	CSO 5	0.158	**0.1383**	0.1563	0.1576	0.5321
	CSO 6	0.0534	**0.0447**	0.0497	0.0843	0.0785
	CSO 7	0.031	**0.0273**	0.0282	0.0333	0.057
	CSO 8	0.0552	**0.0536**	0.0537	0.0778	0.0918
NSE	CSO 1	0.9076	0.9069	**0.9095**	0.8559	0.592
	CSO 2	0.9402	**0.9473**	0.9208	0.8452	0.8722
	CSO 3	0.791	0.7628	**0.7941**	0.7259	0.5526
	CSO 4	0.6899	**0.7147**	0.6532	0.4934	0.6622
	CSO 5	0.9461	**0.9587**	0.9473	0.9463	0.3889
	CSO 6	0.9647	**0.9753**	0.9694	0.9119	0.9236
	CSO 7	0.7932	0.8396	0.8285	0.761	**0.8576**
	CSO 8	0.8644	**0.8721**	0.8714	0.7303	0.6243

Table 6. Performance of six-step ahead predictions of different methods

Performance metrics	CSO ID	Models				
		RNN	GRU	LSTM	FFNN	SVR
CC	CSO 1	0.9089	**0.9104**	0.9045	0.9009	0.7212
	CSO 2	0.9457	**0.9588**	0.9533	0.9464	0.9164
	CSO 3	0.7415	**0.8036**	0.7586	0.7501	0.6089
	CSO 4	0.7431	0.762	0.7617	0.6774	**0.7767**
	CSO 5	0.9706	0.9774	**0.9775**	0.9709	0.9499
	CSO 6	0.9648	**0.9758**	0.9682	0.9616	0.9456
	CSO 7	0.7568	**0.8127**	0.7913	0.8021	0.8047
	CSO 8	0.877	**0.8856**	0.8708	0.8609	0.7912
RMSE	CSO 1	0.1652	**0.1592**	0.1643	0.1773	0.2689
	CSO 2	0.0783	**0.0712**	0.0766	0.0777	0.0906
	CSO 3	0.0586	**0.0514**	0.063	0.0573	0.0716
	CSO 4	0.0461	0.0433	**0.0431**	0.0522	0.0691
	CSO 5	0.1719	0.1951	**0.1442**	0.2526	0.7042
	CSO 6	0.0814	**0.0686**	0.0765	0.1012	0.1041
	CSO 7	0.0458	**0.041**	0.0421	0.0433	0.0468

NSE	CSO 8	0.0727	**0.0698**	0.0742	0.0776	0.1006
	CSO 1	0.8138	**0.8271**	0.8158	0.7856	0.5066
	CSO 2	0.8729	**0.8947**	0.8784	0.8746	0.8295
	CSO 3	0.5229	**0.633**	0.4475	0.543	0.2871
	CSO 4	0.5127	0.5703	**0.5745**	0.3762	0.2948
	CSO 5	0.9359	0.9174	**0.9549**	0.8616	0.7507
	CSO 6	0.9179	**0.9416**	0.9275	0.873	0.8657
	CSO 7	0.5483	**0.6394**	0.6198	0.5964	0.6093
	CSO 8	0.7649	**0.7828**	0.755	0.7316	0.5496

Next, multi-step ahead forecasts are investigated. As known, multi-step ahead forecasting is much more complex to deal with than single step ahead forecasting. Tables 5 and 6 are performances of three-step ahead and six-step ahead forecasting respectively. The prediction performance is deteriorated with longer time steps. The error is becoming more pronounced for the six-step ahead forecasting.

Three useful findings can be extracted from Tables 5 and 6:

1) First, compared with the single task SVR model, the multi-task models exhibited better and stable prediction performance, it means by leveraging information in multiple related tasks, the multi-task model can improve the generalization performance of the tasks. For the sewer system, single task models only extract the temporal relation of a single sewer component, it neglects the spatiotemporal correlations between sewer components.
2) Second, compare to traditional methods such as FFNN and RNN, LSTM and GRU can more efficiently capture spatiotemporal correlations and therefore presents better performance.
3) In most cases, GRU shows a slightly better performance than LSTM, but the difference is marginal. This finding is consistent with the research of Chung et al (2014), but they evaluated the performance of LSTM and GRU on the tasks of polyphonic music modeling and speech signal modeling, we extended their conclusions to time series forecasting.

Table 7. Performance of eight-step ahead prediction of the GRU based model

CSO name	CC	RMSE	NSE
CSO 1	0.8822	0.1824	0.7731

CSO 2	0.9471	0.0763	0.8794
CSO 3	0.7389	0.0595	0.5079
CSO 4	0.7327	0.0451	0.6333
CSO 5	0.9775	0.1571	0.9463
CSO 6	0.965	0.083	0.9146
CSO 7	0.7158	0.0483	**0.499**
CSO 8	0.8521	0.0786	0.7253

We further extend the time step for the GRU based DeepCSO model, we found that even for eight-step ahead forecasting, most of metrics are still in the range of good (CC higher than 0.7 and NSE higher than 0.5). The only exception is marked in bold.

4. Conclusion

Studies relative to sewer systems require the modeling of complex and dynamical urban hydrological processes. The complicated model construction, calibration and computation make the extensively used deterministic methods less adequate for real-time purpose. On the other hand, the implicit data-driven methods could provide predictions in real time, but it can only provide information at a very abstract level.

With large and high-resolution sensors that are now being deployed throughout cities. To guide the real-time operation of the sewer system at a citywide level, develop data-driven models that could characterize the spatiotemporal variability in sewer systems is very necessary. In this context, develop forecasting models separately for individual targets will become less efficient, because this kind of individual models should be uniquely calibrated and re-calibrated for each site, moreover, this approach ignores interconnected nature of sewer system. For instance, the behavior of water level in one CSO structure may influence by both adjacent CSO structures and rainfall intensity. Due to this kind of spatial-temporal nature, MTL approach is employed in this study to develop our proposed DeepCSO model.

Five different models, including deep learning methods (LSTM and GRU), the traditional RNN and FFNN, and the SVR, are compared in this study. Experiments demonstrated that the multi-task approach is

generalized better than single task approach, furthermore, the GRU and LSTM are especially suitable to capture the temporal and spatial evolution of CSO event and superior to other methods.

The deep learning based MTL model developed in this study, called DeepCSO, reflect dynamics of CSO water levels accurately not only across time, but also across sites. As indicated by the results, the DeepCSO model could be a powerful tool by which to predict CSO water levels. The proposed DeepCSO model has the potential to serve as an operational tool for sewer system control. On the other hand, the ability of deep learning to model highly non-linear and nuanced relationships between input-output data sets will motivate more research in the application of deep learning methods to the water management domain.

Acknowledgements

This work has been supported by the Regnbyge-3M project (grant number 234974), which is granted by the Oslofjord Regional Research Fund. The authors would like to thank the engineers from Rosim AS for their supports.

References

Ayyeka. (2017). Ayyeka Sigfox IoT sensors monitor sewage deep underground San Francisco. https://www.networkworld.com/article/3171072/internet-of-things/ayyeka-sigfox-iot-sensors-monitor-sewage-deep-underground-san-francisco.html Accessed 07 November 2017

Bezuglov, A., Blanton, B., & Santiago, R. (2016). Multi-Output Artificial Neural Network for Storm Surge Prediction in North Carolina. *arXiv preprint arXiv:1609.07378*.

Chiang, Y. M., Chang, L. C., Tsai, M. J., Wang, Y. F., & Chang, F. J. (2010). Dynamic neural networks for real-time water level predictions of sewerage systems-covering gauged and ungauged sites. *Hydrology and Earth System Sciences*, 14(7), 1309-1319.

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

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

El-Din, A. G., & Smith, D. W. (2002). A neural network model to predict the wastewater inflow incorporating rainfall events. *Water Research*, 36(5), 1115-1126.

Google. (2016). https://research.googleblog.com/2016/09/a-neural-network-for-machine.html Accessed 27 April 2017

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.

Hsu, D. (2017). Time Series Forecasting Based on Augmented Long Short-Term Memory. *arXiv preprint arXiv:1707.00666*.

Jaokar A. (2015). An Introduction to Deep Learning and it's role for IoT/ future cities. http://www.opengardensblog.futuretext.com/archives/2015/05/an-introduction-to-deep-learning-and-its-role-for-iot-future-cities.html Accessed 07 November 2017

Kanestrøm, P. Ø. (2017). *Traffic flow forecasting with deep learning* (Master's thesis, NTNU).

Laptev N., Smyl S., & Shanmugam S. (2017). Engineering Extreme Event Forecasting at Uber with Recurrent Neural Networks. https://eng.uber.com/neural-networks/ Accessed 05 October 2017

Li, X., Peng, L., Yao, X., Cui, S., Hu, Y., You, C., & Chi, T. (2017). Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. *Environmental Pollution*, 231, 997-1004.

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.

Marçais, J., & de Dreuzy, J. R. (2017). Prospective interest of deep learning for hydrological inference. *Groundwater*, *55*(5), 688-692.

Montserrat, A., Bosch, L., Kiser, M. A., Poch, M., & Corominas, L. (2015). Using data from monitoring combined sewer overflows to assess, improve, and maintain combined sewer systems. *Science of the Total Environment*, 505, 1053-1061.

Mounce, S. R., Shepherd, W., Sailor, G., Shucksmith, J., & Saul, A. J. (2014). Predicting combined sewer overflows chamber depth using artificial neural networks with rainfall radar data. *Water Science and Technology*, 69(6), 1326-1333.

Nourani, V., Baghanam, A. H., Adamowski, J., & Kisi, O. (2014). Applications of hybrid wavelet–Artificial Intelligence models in hydrology: A review. *Journal of Hydrology*, 514, 358-377.

Power D. (2016). IoT gets down and dirty in Australian smart sewers trial. https://readwrite.com/2016/04/22/iot-gets-dirty-australian-smart-sewer-it4/ Accessed 07 November 2017

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489.

Song, X., Kanasugi, H., & Shibasaki, R. (2016, July). DeepTransport: Prediction and Simulation of Human Mobility and Transportation Mode at a Citywide Level. In IJCAI (pp. 2618-2624).

Wu, Z. Y., & Rahman, A. (2017). Optimized Deep Learning Framework for Water Distribution Data-Driven Modeling. *Procedia Engineering*, 186, 261-268.

Zhang, Y., & Yang, Q. (2017). A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*.

Errata list

PhD candidate: Duo Zhang
Thesis: Sewer Overflow Control using Different Modeling Techniques and Internet of Things
Date: 11 November 2019

Side	Line	Original text	Corrected text
15	19	It also causes wastewater volume exceeds the inlet capacity of the WWTPs … …	It also causes wastewater volume to exceed the inlet capacity of the WWTPs
17	19	This kind of solutions uses concrete and steel are referred to as gray infrastructure.	This kind of solution uses concrete and steel is referred to as gray infrastructure.
18	6	This kind of measures uses vegetative or sustainability-based practices ……	This kind of measure uses vegetative or sustainability-based practices
18	15	This type of solutions aims to utilize equalization volumes	This type of solution aims to utilize equalization volumes
26	8	These kind of models are “gray-box models,” … …	This kind of model is “gray-box models,” … …
34	4	… … LSTM replaces ordinary hidden neurons with a series of memory blocks, each of which is composed of a memory cell and three gates.	… … LSTM replaces ordinary hidden neurons with a series of memory blocks, each of memory block is composed of a memory cell and three gates.
49	4	… studying how to properly redistribute inflow to the WWTP …	… studying how to redistribute inflow to the WWTP properly …
49	5	…… predicting the influent flow at future time horizons to effectively manage ICWT.	…… predicting the influent flow at future time horizons to manage ICWT effectively.
51	8	The DeepCSO model is based on large and high-resolution sensors	The DeepCSO model is based on large scale and high-resolution sensors .....
52	19	…… a type of RNN—LSTM—is able to precisely predict the time series data from the IoT.	… … a type of RNN—LSTM—is able to predict the time series data from the IoT precisely.
53	12	…… was demonstrated and evaluated to simultaneously predict overflow at multiple CSO structures.	…… was demonstrated and evaluated to predict overflow at multiple CSO structures simultaneously.
54	21	This method could be an effective solution to achieve online monitoring of water quality parameters.	This method could be an effective solution to achieve the online monitoring of water quality parameters.

ISBN: 978-82-575-1614-7
ISSN: 1894-6402

Norwegian University
of Life Sciences

Postboks 5003
NO-1432 Ås, Norway
+47 67 23 00 00
www.nmbu.no