EasyChair Preprint

№ 10983

A Digital Twin of a Sewage Water System Using Neural Networks

Riccardo de Socio, Chelsea Owen, Ege Turkyener, Emre Davut and Sergio Lopez Dubon

September 28, 2023

A DIGITAL TWIN OF A SEWAGE WATER SYSTEM USING NEURAL NETWORKS

Riccardo, de Socio
desocio@foredata.ai

Chelsea, P.J., Owen
chelseaphilippajudith.owen@studenti.unipd.it

Ege, A., Turkyener
egealp.turkyener@studenti.unipd.it

Emre, Davut
emre.davut@studenti.unipd.it

Sergio, A., Lopez Dubon[1]
Sergio.LDubon@ed.ac.uk

KEYWORDS

Water Management, Forecast, Anomaly Detection, Graph Neuronal Networks, Real-Time

ABSTRACT

Under a climate change scenario, and with environmental pressures and regulations in the European Union, the correct management in a quasi-real-time and the forecast under different scenarios of the sewage water system (SWS) play a crucial role; they significantly impact urban floods and water quality treatments. Under this scenario, we developed a Data-Driven Digital Twin (DT) using different Neuronal Networks (NN) for a small SWS basin in northern Italy. The basin under study consists of a 140 km long sewage network and a total of 22 Doppler sensors that measure every six minutes the water velocity, water pressure (depth) and water temperature and three rain gauges that measure every minute for a total of 1140 days of register. Due to the conditions in which the sensors work, it is typical to have low-quality measures. For this reason, we developed, trained, and included in the DT an NN capable of detecting any anomaly value, assigning a possible cause to the problem (i.e., dirty sensor), and suggesting a correct potential value; this model shows an accuracy >90%. After this quality control, the data passes into the main DT. This research evaluates two approaches: a convolutional layer NN and a Graph NN. Both models mimic the configuration of the SWS and use the same data to be trained. The model was evaluated under scenarios of missing data (i.e., sensor removal). Both models show a general accuracy of>90%. Finally, this project shows DT's successful development and application due to industry, government and academic collaboration.

1. INTRODUCTION

The increasing population of industrialised cities puts more pressure on water systems worldwide. A report from the World Resources Institute (WRI) in 2019 found that a quarter of the world's population lives in extremely high water-stress countries [1]. The UN's Human Development Report from 2006 stated that water scarcity has already affected every continent. The report stated that there is enough freshwater on the planet for seven billion people. Still, it is distributed unevenly and too much is wasted, polluted and unsustainably managed [2]. Countries are increasingly investing in monitoring and assessing water levels in order to better manage the available resources [3-5].

The EU Urban Waste Water Treatment Directive of 1991 was introduced to protect the environment from untreated discharges [6]. More than thirty years later, Italy treats only 56% of its sewage in line with this legislation [7]. The UK has a network of over 416,175 km of water mains and more than 393,460 km of sewers and is a pioneer in treating and recycling wastewater [8]. However, growing public awareness about a sewage discharge legal loophole being aggressively exploited by water companies led to the UK government

announcing the Storm Overflows Discharge Reduction Plan. This aims to reduce the number of untreated sewage overflows into English waterways by 40 per cent before 2040 [9].

In recent years, the need to find better, cost-effective solutions for the future has forced innovation in the water management market. Investment in smart systems has grown, and worldwide, smart water systems (SWS) represent an ever-increasing market size, predicted to double in the next five years [10]. A SWS refers simply to a network of water meters and endpoints, a data collection system and system management infrastructure. They can be applied to utilities of all sizes and locations. The water flow is measured continuously, and the endpoints send flow data to the collection system at regular intervals. This means the network is monitored closely, and issues like leaks or pressure losses can quickly be diagnosed. Customers can also see their usage in real-time, which can encourage more responsibility [11].

Predictive neural networks are an area of machine learning (ML) which can be applied in this context as an essential tool for mitigating negative environmental impact. The use of machine learning in water resources has been extending in the last years (i.e.,[12-13]); in this work, we developed a series of machine learning models able to cleanse the data and model the complex interaction inside a SWS using two different approaches; Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs).

Classification of data plays an essential role in the process. Accurate categorisation will mean more accurate predictions in the later stages of this project, allowing more informed decisions and better analysis for the users. Additionally, as with all ML algorithms, the quality of training data is profoundly important. This work significantly emphasises data engineering procedures to ensure the data is suitable for the classification tasks. We address the issues of quality, bias, noise, missing data and normalisation. This ensures not only a meaningful result now but also for all future clients.

CNNs represent a significant advancement in the field of computational technology. These networks can detect intricate patterns within data, which is essential for various applications. The convolution operation to extract temporal features from input data helps make significant predictions [14]. One must define and train the network architecture using appropriate data to create a neural network with a convolutional layer to recognise the linear and nonlinear correlations between input and output. Additionally, pooling layers are frequently employed within CNNs to reduce input size while preserving spatial relationships [15]. These advantages of the features lead us to adopt CNN with these distinctive attributes.

The utilisation of CNNs also extends to SWS; consider SWS a network of interconnected pipes and stations. CNNs serve as the analytical tool to unravel this complex network. In this paper, we explore the integration of CNNs and GNNs into SWS, showcasing their role in enhancing efficiency and performance. In this work, we model a SWS using GNNs in quasi-real time. An SWS is an intricate network of constant changes and interactions, and the graph theory provides a valuable mathematical framework that allows us to understand the essence of these interactions by representing entities as nodes and their connections as edges within a network structure. For a SWS, the intricate web of sewer pipes, pumping stations, and treatment facilities can be effectively represented as a graph, where nodes represent various system components, and edges depict their connections.

Graph theory traces its origins back to the seminal paper [16]; in this paper, Euler addressed the problem of the Seven Bridges of Königsberg by introducing the concept of a graph and formulating the bridges and land masses as nodes and edges. Euler successfully demonstrated that it was impossible to find a path crossing each bridge exactly once. In recent years, graph theory has intersected with machine learning, leading to the emergence of GNN [17]. Standard NN methods are not suitable for processing graph-structured data. However, GNNs provide an approach to handle such data. Nowadays, graph theory is used in various fields such as computer science, social science, biology, environment, economics, and linguistics, and modelling these interconnected systems has become of the utmost importance.

2. DATA

The data used in this study come from a network of measuring instruments of Gruppo CAP temporarily installed for three years to monitor the entire network of the municipality of Sesto San Giovanni (Milan),

shown in Figure 1. The monitoring was carried out to build a digital twin of the sewer network, consisting of a real-time fluid dynamic model, which can also be run in forecasts, thanks to deep learning regression techniques. The digital twin facilitates all operations of the network operator, as it always provides knowledge of the status of the entire water infrastructure.

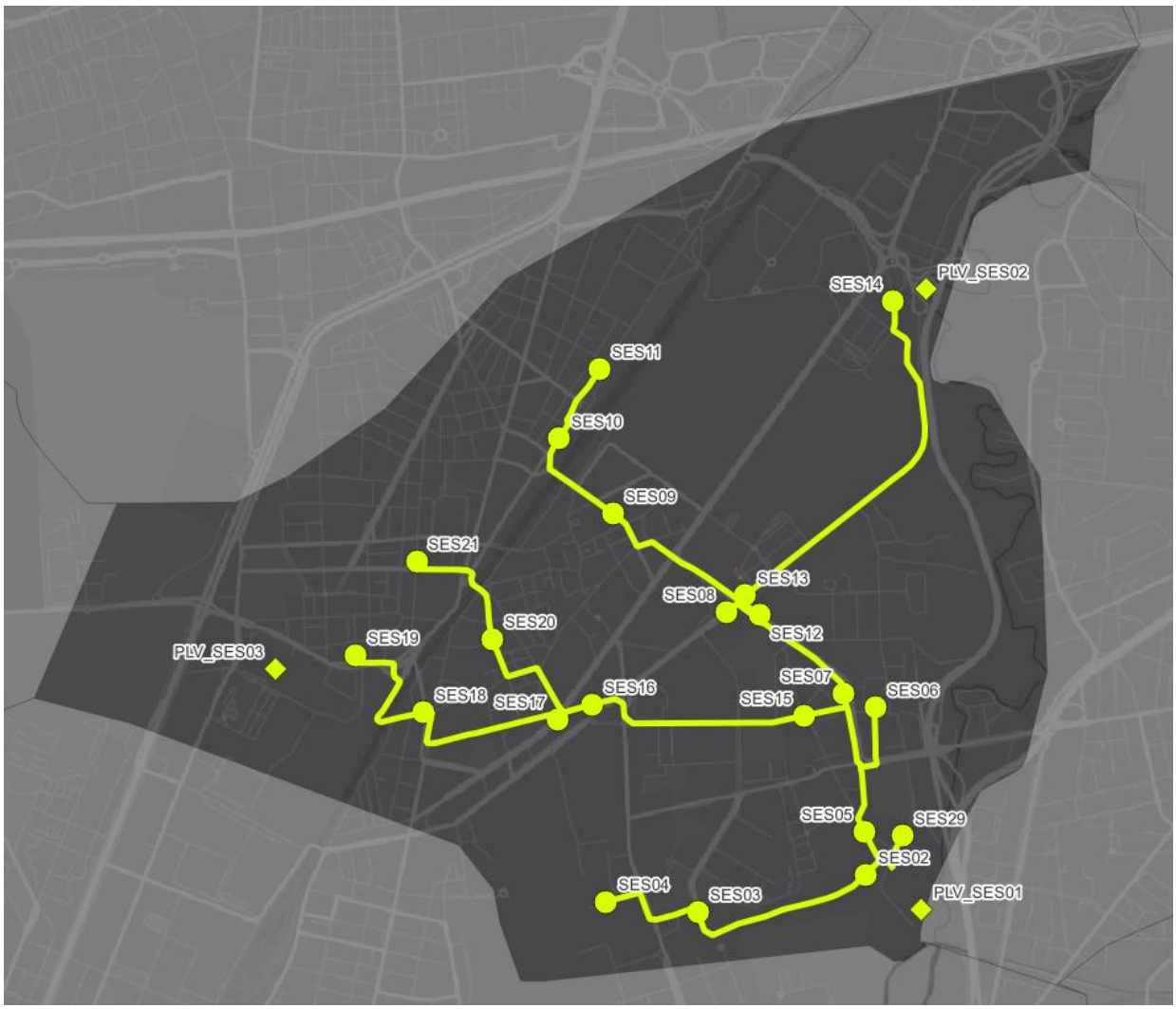

Figure 1. Map of the main pipes of Sesto San Giovanni sewage network.

The water network is 140 km long and is equipped with 21 flow meters (see diagram Figure 2), Doppler ultrasound technology (brand Nivus model KDO), i.e., area-velocity method, and three rain gauges. The sampling time varies from 3 to 6 minutes, and the data is transmitted every 4 hours. Ultrasonic meters with Doppler technology represent the state of the art in sewerage measurements. However, they suffer from several problems, such as sensor fouling (with consequent loss of velocity) and pressure compensator malfunction (with consequent level drift) in the first place, but also too much or too little density of suspended sediment and, like all measuring instruments, a decrease in performance as they approach their limits of level and velocity measurement.

Therefore, this type of hydraulic measurement always requires a post-process, which, at state of the art, can be carried out in real-time thanks to deep learning. Therefore, the data from the various gauges can only be used as boundary conditions for the real-time calibration of a fluid dynamics model after a post-process. In these deep learning models, metadata were also used, mainly concerning the geometry of the network and the type of instrument: shape and size of the cross-section, length and slope of the pipes, mounting position of the sensor, thickness of the sludge, dead band of the velocity measurement, lower and upper measurement limits of the instruments, their resolution and measurement errors.

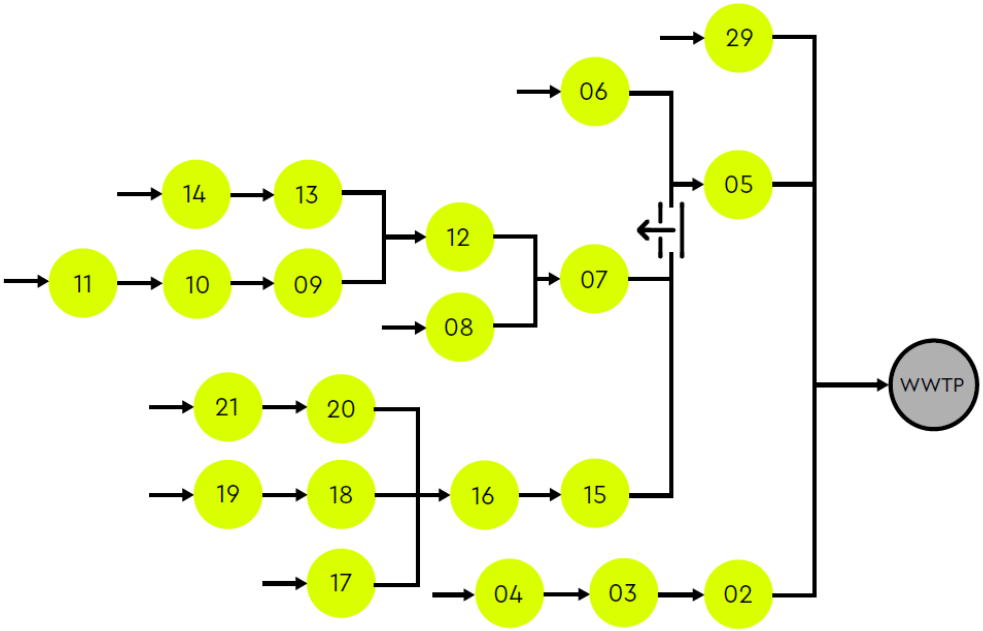

Figure 2: Diagram of the sewage water system measurement points.

3. METHODS

In order to properly construct a digital twin of the sewer, several algorithms, including deep learning, were used in cascade. The work was divided into the following phases:

1) Data preparation,
2) Data classification,
3a) Data cleansing using NN,
3b) Data cleansing using GNN.

3.1 Data preparation

Data preparation is the application of a set of indispensable techniques for the subsequent use of data. The first operation applied to the data was a resampling at 6 minutes from the raw data to ensure consistency within the data. This is because some of the measurement sites which had initially been sampled at 3 minutes were later changed to 6 minutes during the measurement campaign. The units of measurement of the individual parameters were then considered in relation to instrumental accuracy to identify the significant figures for each parameter. Then, inaccuracies in recording the measurements, such as resets and occasional null cumulative values, were corrected for the rainfall measurements. During all these operations, particularly during resampling, handling the initial null values appropriately without confusing them with any nulls that may have been introduced, even temporarily, for data processing is crucial.

Data engineering is the set of data manipulation techniques preliminary to a deep learning model aimed at improving performance. In particular, measured raw parameters can be modified, or new parameters can be created. Data engineering is specific to each model and is therefore dealt with in the sections on the models themselves.

3.2 Data classification

The measured data classification consists of creating an algorithm for automatically labelling the measured data, dividing them into 'valid' and 'non-valid'. It is an essential process for correctly preparing data for subsequent regression algorithms. We use three measurement campaigns, which all use the same measurement instruments. The three measurement campaigns are subdivided into two for “training” and the last for “testing”. Then, a further subdivision of the measurement sites in each of the “training” campaigns splits them into “train”

and "test". The final layer of this is to take the measurement sites for training and split each of them into "train", "test" and "validation" datasets. This is demonstrated in Figure 3.

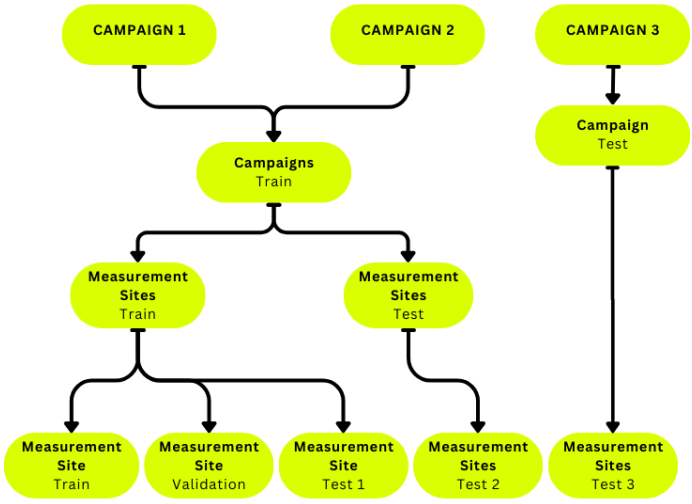

Figure 3. Flowchart to demonstrate the subdivision of training and testing measurement sites.

3.2.1 Data engineering

It is crucial to ascertain the credibility of data before using them to be able to support any conclusion. For the reasons stated in the previous paragraph, apt data cleansing in this setup is an important problem to solve. At each measurement point, information on a range of different features is recorded periodically. The most important are the water flow's velocity and water level and the quality diagnostic parameter (q), which provides insight into that particular measurement's accuracy. There is also the water temperature (t) and the rain intensity (i), which is a sparse series as there is no rain during dry time, and thus, there is no correlation. The rain intensity, recorded using rain gauges, is calculated per every measurement point via a Kriging interpolation. When it does rain, the level and velocity are affected. The temperature may help to see if it is raining, as a drop or rise from a specific temperature is probably explained by rain. Then, there will be a particular concentration time, meaning the time for the rain to reach sewage after the rain starts, before the level and velocity are affected. The concentration-time itself, of course, depends on the local urbanisation, especially the soil sealing.

To have consistent timestamping throughout all the data, it must be resampled up to one-minute intervals. This is because, in some cases, the frequency of data collection changes during the measurement campaign, and it needs to be consistent before training. Scaling data before inputting into neural network systems is a well-known crucial pre-processing step to improve outputs. The inputs are scaled to the same interval as the outputs (which for the classification task are boolean), so [0, +1] in this case.

To scale the velocity, we first exclude any raw negative values. Additionally, we exclude values smaller than the measurement error of the sensor, so smaller than 0.06 m/s (see). Velocity (v) is then scaled to the 95th percentile from all the measured values of that measurement campaign. This is because, whilst it is, in theory, possible to have extremely fast flows, it does not make sense to scale for those values as they are so rare. They will increase the skewness of the distribution, which is known to greatly impact model learning as the degree of skew affects the learning metrics [18]. Therefore, most values will fall within the model's 0-1 scaled range. However, if an anomaly occurs, it will be outside the scaled input interval. The model may learn to recognise such values as abnormalities. This approach is also followed for scaling the temperature.

$$\varepsilon = 95^{th}\ Percentile\ of\ V \tag{1}$$

$$v_{scaled} = \frac{v}{\varepsilon} \tag{2}$$

where V is all of the measured values of velocity larger than 0.06 m/s, and v ach individual value.

3.2.2 Classification model

Data.

The data have been classified manually, i.e. invalid velocities have been selected, and the result of this operation is a boolean series. The model takes data as a time series, where each single measurement point is associated to the previous 45 minutes of data. We take the scaled data and create three-dimensional tensors in the following way: dimension one is the samples containing the previous time steps (chosen to be 44) and the five features (velocity, level, quality, temperature and rain intensity) for each minute. An individual label accompanies each of the timestamps. It is simplest to visualise each sample represented by a 45x5 grid, and then all the grids are stacked on top of each other.

One of the biggest challenges of the model setup is the management of the amounts of data during training, as the data should have comparable amounts of “valid” and "invalid” data to avoid overfitting and maximise generalisability. This will be addressed in the following paragraphs. Time will constitute another dimension for the model. For every input parameter, a certain number of the previous measurements are included, i.e. associated with every timestamp, which allows the model to know the state of all parameters during previous timestamps. This is to aid in the higher-level aim for the model to learn what time of day it is because usually the flow is influenced by anthropic activities.

The highest performance is gained by a model with two one-dimensional convolutional layers and one dense layer before the output. The loss is a custom function, which is a weighted binary cross-entropy, using the class weights for the whole training set. The weights are an average of the distributions across all the measurement points.

$$weighted\ cross\ entropy\ loss(i) = -\Sigma\, w(i) \cdot \big(y(i) \cdot \log\big(p(i)\big) + \big(1 - y(i)\big) \cdot \log\big(1 - p(i)\big)\big) \quad (3)$$

where $w(i)$ is the weight of class $i$and $y(i)$ and $p(i)$ are the true and predicted labels for the class, respectively.

Training.

The training set consists of 70% of data from each measurement site assigned for training. The remaining 30% is split equally into validation and test sets. As explained in the next section, other measurement sites from those same measurement campaigns are unused in training and used only later for testing. The model then trains measurement site by measurement site, and one epoch is once all the training measurement sites have been seen. In between each epoch, the order of the measurement sites is shuffled to avoid all the measurement sites from one campaign being grouped together and the model overfitting to one campaign or the other. The model is trained consecutively on the data from one measurement site and then the next without shuffling the order of the data. This is to avoid the model getting information from later timestamps before training on the current timestamp. A custom function is additionally implemented, which rejects data sets with less than 5% or more than 95% bad data. Additionally, it downsamples the data sets which have more than 70% bad data and upsamples those with fewer than 30% bad data. This is to try to avoid the model becoming biased to extreme label distributions and outputting a single value without learning to recognise the patterns.

3.3a Regression Neural etwork

In this section, the Regression Neural Network through Convolution (CNN) is analysed, and its performance within the context of sewage water system management is evaluated. CNN is a significant machine/deep learning architecture originally designed for visual data processing like image recognition and computer vision. Still, it can also be effectively applied to handle one-dimensional (1D) data, such as time series. Its strength lies in its ability to automatically learn and extract meaningful features from the input data. In the context of 1D data, CNNs utilise convolutional layers with small filters (kernels) that slide over the sequential data, performing convolutions to detect patterns and spatial relationships between adjacent elements. Because of these features, CNNs are well-suited for handling one-dimensional data, like time series, as sequence data, allowing the convolutional operation to extract relevant features.

The proposed model architecture is made of Convolutional Neural Networks (CNNs). The inputs of the model consist of data from multiple sources:

- Rain gauge data: Rain measurements from three rain gauges are captured every minute for a total of 1140 days.
- Doppler sensor data: Doppler sensors collect data on water velocity (v), water level (l), and water temperature every 6 minutes.

3.3a.1 Data engineering

To handle low-quality sensor data, the model takes as inputs all labels predicted by the model described in paragraph 3.2, which is capable of detecting anomalies in velocities. All the data engineering steps used for that model are valid also for this one. The peculiarity of the data engineering of this model lies precisely in handling the outliers reported by the first model. These 'invalid' measurements are, in fact, reconstructed from the 'valid' ones by means of a regression of the type:

$$v = a\big(f(l)\big)^b \tag{4}$$

where l is the level, v is the velocity, and a and b parameters are linearly regressed.

In addition, another parameter was calculated and added as input to the model, which consists of a regression of the level through Fourier transforms. This virtual parameter provides the notion of “time of the day” in terms of anthropic activity in a site-specific way. Later, another algorithm to automatically detect the level drift is applied. Hence, the levels are corrected through the aforementioned regression equation when velocity values are “valid”, and with the calculated periodic level otherwise. These preprocessing steps enable more reliable downstream predictions.

Finally, as done for the classification model, data are split into the train, validation, and test sets, respectively 70, 15, and 15 per cent of the totality of the data.3.3a.2 Regression model. The model is designed as a Data-Driven Digital Twin for the SWS. It utilises Convolutional Neural Networks (CNNs) to process the input data from the sources above and forecast the level, which is a key parameter in SWSs. The model aims to provide quasi-real-time management and predictions under different scenarios.

The CNN component of the model processes data from each measurement point (Doppler sensors’ sites) through convolutional layers with various numbers of filters. These layers perform feature extraction and help capture temporal patterns within the data. The feature maps from all measurement points are then concatenated to represent the system comprehensively.

The model begins with an input layer that takes in data related to measurement points as a 3D tensor, whose dimensions are the same as explained in paragraph 3.2. This data is representative of measurements of sensors, meteorological information, and virtual parameters. For each additional measurement point, a separate input layer is created to process the data specific to that measurement point. The data from each measurement point undergoes a similar set of convolutional layers. The number of nodes determines the number of filters for these convolutional layers. These layers apply a rectified linear unit (ReLU) activation function to introduce non-linearity to the data and help in feature extraction. The causal padding is used, which maintains causality in the temporal data.

After processing the data from all measurement points through their respective convolutional layers, the resulting feature maps are concatenated together. This step helps combine the information from different sources to make more informed predictions. The concatenated feature maps are then passed through additional convolutional layers to extract more complex features. The kernel size, strides, and activation function used in these layers are similar to the previous convolutional layers. Following this, a Global Max Pooling layer is applied to reduce the dimensionality of the data and capture the most important features.

Finally, dense layers with the ReLU activation function are employed to generate predictions for the desired output parameters. After this, the model is trained using labelled data, i.e., the same level values used as input, and the loss for each measurement point is monitored during training. Root Mean Squared Error (RMSE) metric is used to assess the model's accuracy on the test data. The evaluation is likely performed using a validation set to ensure the model's generalisation capabilities.

3.3b. Graph Neural Network

A graph comprises two fundamental elements: nodes, also known as vertices and edges. The Graph Neural Network (GNN) is a robust neural network architecture specifically designed to handle data organised in graph structures, making it an ideal choice for modelling the connected nature of SWS components [19]. A graph (G) contains a pair of sets of vertices (V) and a set of edges (E). In the context of graph representation, the graph can be denoted as

$$G = (V, E) \tag{5}$$

where the total number of nodes is denoted by,

$$|V| = N \tag{6}$$

and the total number of edges denoted by

$$|E| = N^e \tag{7}$$

The graph's structure is represented using a 2-dimensional.

$$R^{NxN} \tag{8}$$

The adjacency matrix, denoted by A. The entries i, j indicate eitherhe presence or absence of an edge between the nodes $i$and j The process of graph representation includes hidden state h_v and output vector o_v to capture and represent the information of each node in the graph. These node representations play an essential role in encoding the node's attributes and interactions with neighbouring nodes.

GNN, as an initial step, involves identifying the graph structure. This process can fall into two main categories such as structural scenarios and non-structural scenarios. In structural scenarios, the graph structure is defined within the application context. This includes physical systems. Non-structural scenario graphs are implicit, necessitating the construction of the graph from the task at hand, such as for image analysis, a scene graph can be constructed. Within the context of the SWS, given its nature as a physical system, it aligns with the structural scenario. After the foundation of the graph, graphs are typically categorised based on their type and scale as homogeneous, heterogeneous, directed, undirected, static, and dynamic. In the context of the SWS, the system's characteristic is static, directed, and heterogeneous.

- **Homogeneous Graphs:** Nodes and edges share the same types, and nodes/edges share the same identity space.
- **Heterogeneous Graphs:** Nodes and edges do not need to share the same type and more commonly share different types, and nodes/edges share different identity spaces.
- **Directed Graphs:** Edges have a specific direction from one node to another.
- **Undirected Graphs:** Edges with no specific direction from one node to another can be treated as two directed edges.

- **Static Graphs:** Graphs that do not change over time. For instance, road networks and citation networks.
- **Dynamic Graphs:** Graphs that change over time. For instance, disease spreads as nodes represent individuals and edges represent potential transmission routes. The topology of the graph changes as individuals heals and stop transmitting.

The GNN model utilizes a graph representation to capture the dependencies between various elements in the SWS. The GNN representation employs nodes to represent various system components in this context. These nodes could correspond to measurement points, sewage treatment units or other relevant elements within the network. In this work, they represent the measurement points. To mimic the physical connections and relationships between these components, GNN utilises edges on the graph. In this context, the edges represent the pipes connecting different SWS nodes. The lengths of these edges correspond to the physical distances between the measurement points or nodes in the sewage network (see Figure 2). The task of predicting the water velocity and level for different SWS nodes falls into the graph regression category. In graph regression, the GNN aims to learn a mapping between the input features and node-edge attributes together with the target output. Nodes corresponding to different elements are added to the graph. Edges are established between the nodes based on the connections observed in the SWS. These edges define relationships between various components.

3.3b.1 Data engineering

All data engineering operations described for the Neural Network Regression model are applicable to this model. In this specific case, the edge values are expressed in distance and are measured in kilometres (km) as the unit of measurement. These values typically fall within a range that is relatively close to the interval of 0 to 1. To evaluate the GNN model's performance, the dataset is divided into three sets: training set, validation set, and test set, respectively 70, 15, and 15 per cent. The training set is used to train the GNN model, the validation set helps in tuning hyperparameters and preventing overfitting, and the test set evaluates the model's generalisation on unseen data. To efficiently process the graph data through the GNN model, the graph is converted into a specific format to be processed by the training.

3.3b.2 Graph model

At the start of the GNN process, initial states are assigned to both nodes and edges. These states capture the specific characteristics of each node and edge in the SWS. On the base of the GNN model lays the graph update layer. These layers perform graph updates iteratively and return a new Graph Tensor with updated features. This update mechanism leverages a message-passing technique, where each node exchanges information with its connected nodes, as depicted in Figure 2. As a result, the GNN learns and captures complex relationships within the SWS nodes and edges. Within each graph update, 1D convolutional layers are activated, playing a crucial role in capturing the patterns in the data. The final output layer of the GNN model utilises an appropriate activation function to ensure that the predictions are within a meaningful range. Once the model is trained on the training dataset, evaluating its performance on the unseen test dataset is crucial. Root Mean Squared Error (RMSE) is used to assess the model’s accuracy on the test data.

3. RESULTS

3.1 Data classification

We visualise the training over 50 epochs in Figure 4, using the loss calculated from the custom weighted binary cross-entropy loss function. The loss, recall and precision over this training period are shown. The training is considered to be complete after this number of epochs, as the recall and precision are both at >0.99, and the loss is <0.01. We note that recall is a particularly important metric. Recall, in the case of binary classification, is calculated as in Equation 9. It gives an insight into how the model is learning, as its value depends on the number of false negatives, so we use it to track those performances directly. Similarly, precision is instead calculated as in Equation 10. This allows tracking of the model is progress to reduce the number of false positives. We then visualise the confusion matrices for each testing dataset to see more closely where the model is making errors in Figure 5

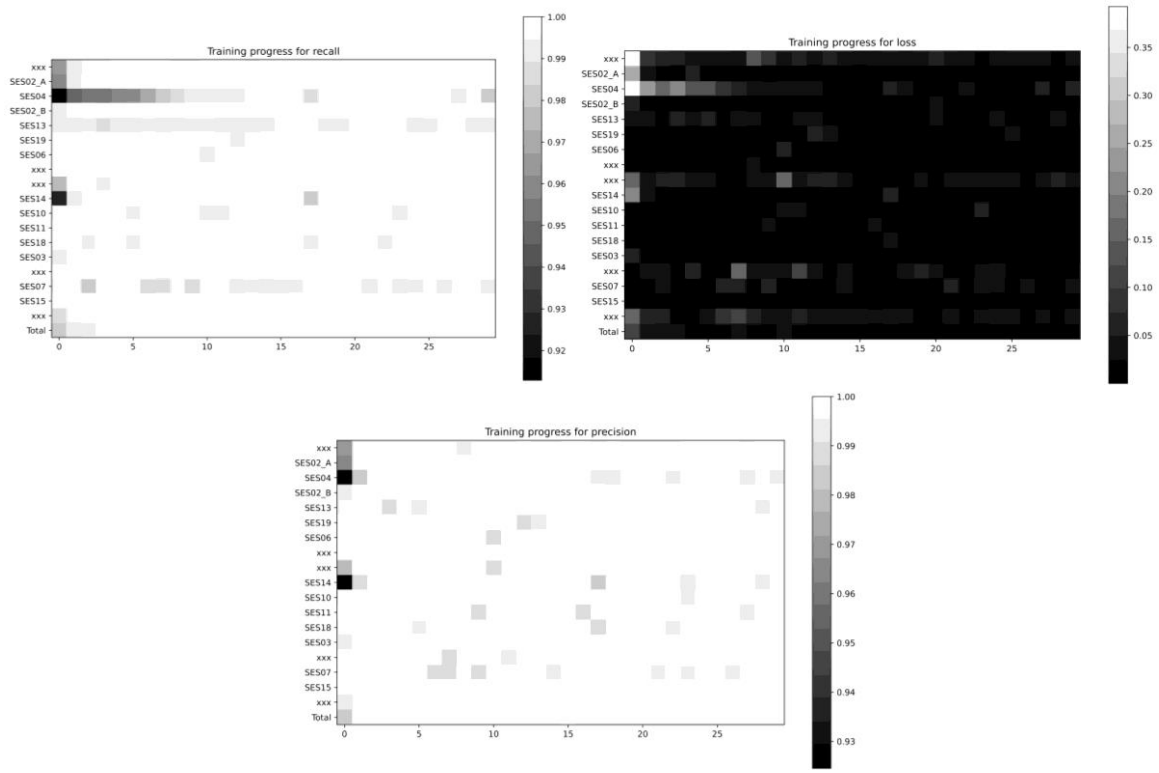

Figure 4. Benchmarks during training over different approaches.

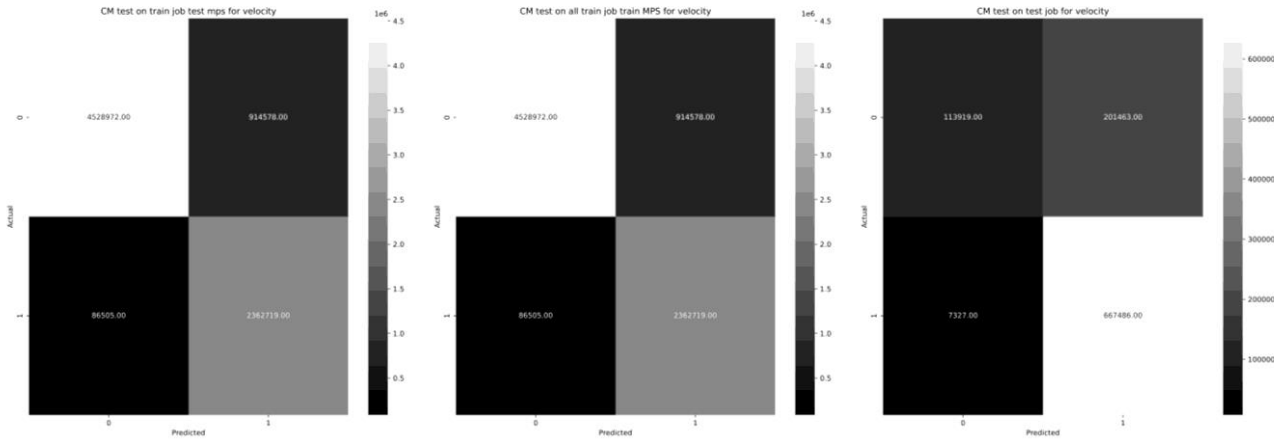

Figure 5. (a) Combined confusion matrix showing the accuracy of the model predictions for test data coming from the same measurement sites as were using in training. (b) Similarly, the accuracy of predictions for the test dataset of measurement sites unseen by the model during training. (c) Finally, the accuracy of predictions on the test dataset from an unseen measurement campaign.

$$R = \frac{Number\ of\ True\ Positives}{Number\ of\ True\ Positives + Number\ of\ False\ Negatives} \tag{9}$$

$$P = \frac{Number\ of\ True\ Positives}{Number\ of\ True\ Positives + Number\ of\ False\ Positives} \quad (10)$$

We utilise a multi-layered testing method to maximise the generalisability of the model. Firstly, the test set from the selected measurement sites is used. Secondly, the model is tested on unseen measurement sites from the same jobs used for training. Thirdly, it is tested on data from a measurement campaign not seen in training. The comparison between testing datasets is visualised in Figure 6. We see that the precision and recall on unseen measurement campaigns are similar to the values on the test values from measurement sites used for training. However, the loss is higher. This is probably due to overfitting one (or both) of the training data campaigns. One of those used for training is significantly larger than the other two. Therefore, the model has probably learned to follow the trends of that measurement campaign more closely.

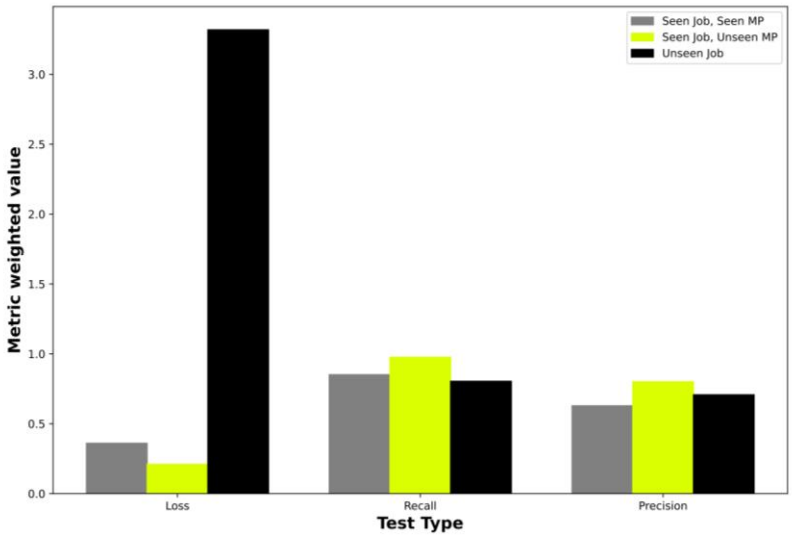

Figure 6. Comparison of metrics for the different types of tests.

Checking more closely, we see that the higher loss from the unseen measurement campaign is due to the model predicting "invalid" for that which is "valid" data. This is a problem to look more in-depth at. However, it would be more problematic if the model was predicting bad for good data as this would cause issues in later processing. Thus, the model performs very well regarding the metric we care the most about – correctly predicting values for bad data.

3.12 Regression neural network

The study's outcomes represent a valuable assessment of CNN's predictive capabilities on the selected sewage network. In the second part of the results, depicted in Figure 7, we delve into RMSE analysis, offering quantitative insights into the model's predictive accuracy. For each measurement point, we obtain a good number of RMSE. This suggests very good predictive performance for each of them. When the RMSE percentage calculation is made, these results are respectively obtained: 0.38% for SES07, 0.71% for SES08, and 0.47% for SES12. RMSE is just one of the possible metrics to evaluate the model performance, and it has a certain level of generalisation as it introduces a mean without considering other aspects. Hence a qualitative plot is also used in order to further check the behaviour of the prediction throughout the 3-time series.

In the level comparison subplots, we displayed firstly is rain intensity in Figure 8 This hyetograph is structured in 1-hour intervals and visually depicts precipitation occurrences, unveiling the temporal rainfall patterns that significantly impact sewage network water levels.

The following section three in this subplot illustrates level comparisons for SES07, SES08, and SES12, respectively. In particular, predicted levels are compared with the raw ones, i.e., those before the data engineering pre-treatment. We notice a strong correlation between actual level data and the model's predictions for SES07. These visuals indicate that the model effectively captures the inherent dynamics of these sewage networks. In particular, for SES07 and SES08, it is noteworthy that several level drifts have been automatically corrected throughout the process.

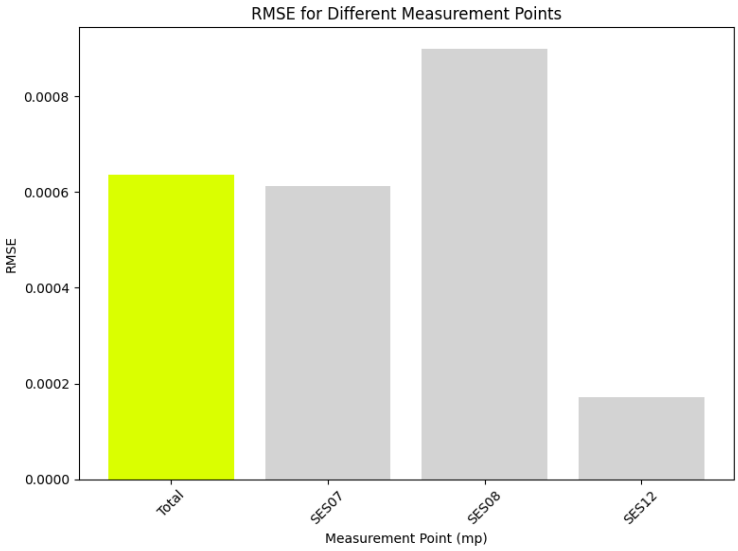

Figure 7. RMSE comparison for individual points and overall value using CNN.

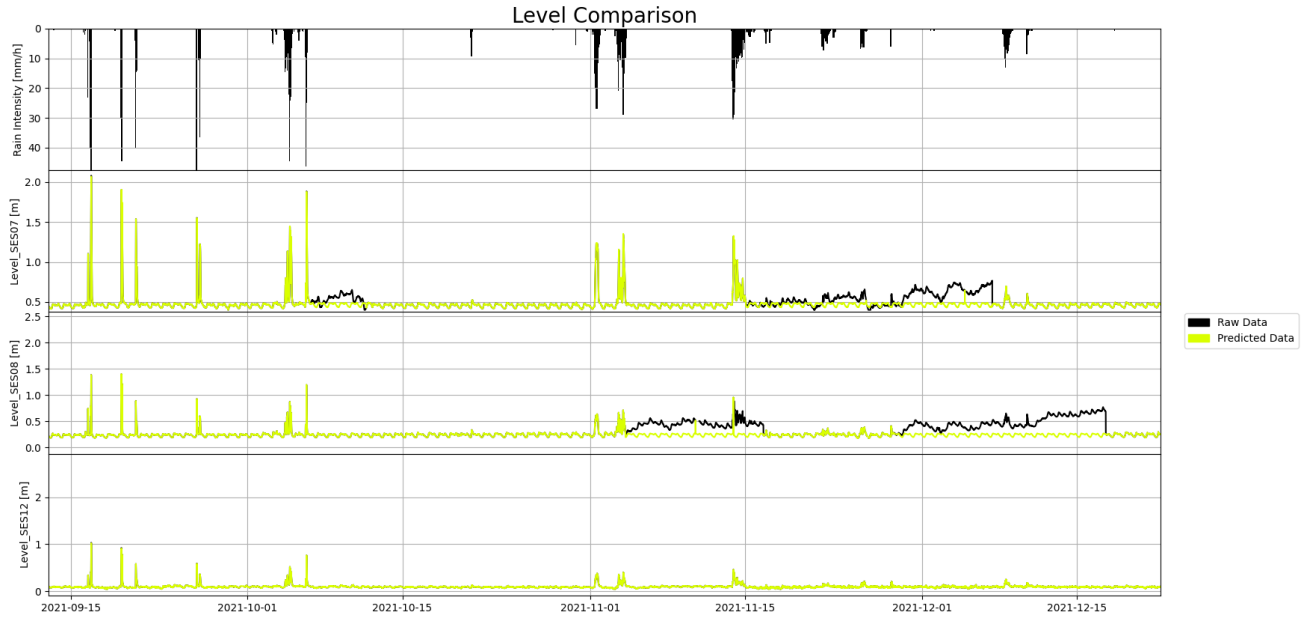

Figure 8. Rainfall and observed and predicted water levels using CNN for different points.

3.3 Graph neural network

The graph neural network has been trained on the same subset of parameters as the Regression Model. These parameters collectively play a crucial role in predicting water levels in the network. This training has been conducted on the same limited subset of measurement points as the Regression Model, using SES07, SES08 and SES12.

The careful choice of this smaller network segment is not arbitrary; rather, it reflects a systematic approach to model development. Such an approach is frequently employed in complex systems, allowing for more focused analysis and controlled experimentation. In this instance, SES07, SES08, and SES12 were selected as

representative nodes within the broader network. The Graph Neural Network model presented for the SWS demonstrates the power of GNNs in capturing complex relationships and dynamics within connected systems.

The results, which will be detailed in both quantitative Figure 9 and qualitative Figure 10 forms, echo the effectiveness of the GNN model. Overall, the outcomes are promising and contribute positively to the network's broader understanding of water level prediction, showcasing the robustness and potential of the GNN approach.

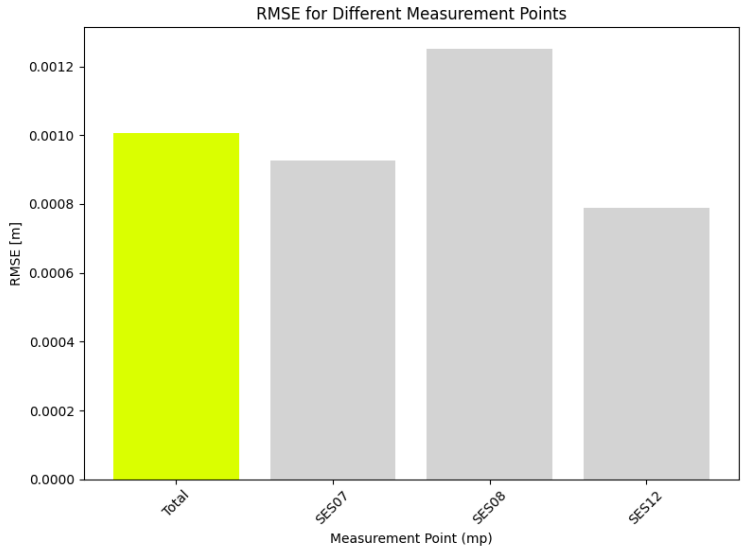

Figure 9. RMSE comparison for individual points and overall value using GNN.

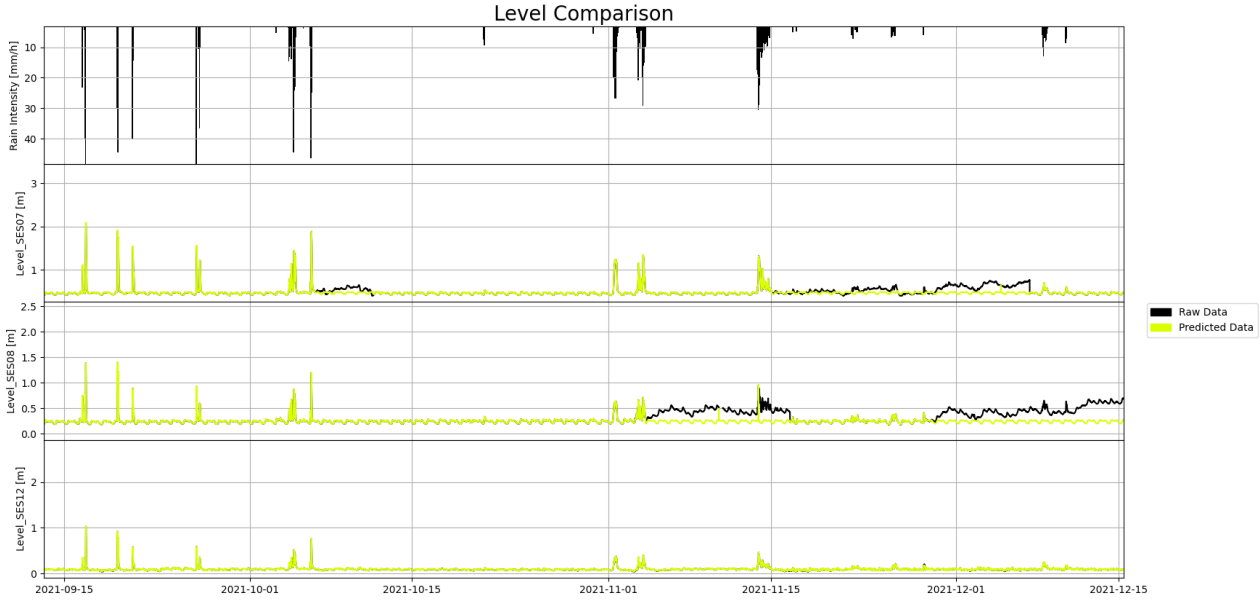

Figure 10. Rainfall and observed and predicted water levels using GNN for different points.

4. CONCLUSIONS AND FUTURE WORK

This study is aimed at building a digital twin of a SWS and provides the data measured by the instruments cleansed of the technical problems that measurement technology has. In this specific case, the digital twin is an ensemble model consisting of a real-time fluid dynamic model on which boundary conditions are imposed, resulting from signal cleaning by means of cascade deep learning techniques.

4.1 Classification algorithm

The Data Classification algorithm at this stage presents already good results but is still overfitting in favour to the jobs shown during training. Surely in the future, retrieving more data will decrease the overfitting phenomenon. Another future development will be to predict the label “invalid level”, instead of just “invalid velocity”. At this stage, the level has, in fact, been left out, as the level class is normally very unbalanced towards “valid levels”, level is a much more stable measure than velocity.

4.2 Regression algorithms

Regression models, with and without the use of graphs, produced excellent results. However, it is emphasised that we only predicted the level parameter, leaving out speed, which, as mentioned, is less stable than level. Speed will therefore also be addressed in the future. Furthermore, a further future development will be to predict future timestamps by adding weather variables to the input. Finally, thanks to the graph neural network, it will be possible to predict nodes that have been measured in the past but are no longer currently measured (so-called virtualisation of measurement points).

NOMENCLATURE

SWS	Sewage Water Systems
GNN	Graph Neural Network
V	Vertices
E	Edges
RMSE	Root Mean Square Error

ACKNOW LEDGEMENTS

The last author: This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801215 and the University of Edinburgh Data-Driven Innovation programme, part of the Edinburgh and South East Scotland City Region Deal.

REFERENCES AND CITATIONS

[1] AQUEDUCT water risk atlas, https://www.wri.org/applications/aqueduct/water-risk-atlas visit on 01/09/2023.

[2] HDR 2006 - Beyond scarcity: Power, poverty and the global water crisis," Human Development Report (1990 to present), Human Development Report Office (HDRO), United Nations Development Programme (UNDP), 2006, December.

[3] The UK government, Multi-billion pound investment as government unveils new long-term plan to tackle flooding, Press release, https://www.gov.uk/government/news/multi-billion-pound-investment-as-government-unveils-new-long-term-plan-to-tackle-flooding, visit on 01/09/2023.

[4] OECD, Financing Water Supply, Sanitation and Flood Protection: Challenges in EU Member States and Policy Options, OECD Studies on Water, OECD Publishing, Paris, 2020, https://doi.org/10.1787/6893cdac-en.

[5] African Development Bank Group, The African Development Bank Group Water Strategy 2021 – 2025, Towards a Water Secure Africa, June 2021

[6] European Commission on Energy, Climate change, Environment; Urban wastewater, https://environment.ec.europa.eu/topics/water/urban-wastewater_en, visit on 01/09/2023.

[7] European Commission, European Environment Agency, European Information Gateway to Watter Issues, Italy, https://water.europa.eu/, visit on 01/09/2023

[8] The UK government, Water and treated water, promotion material, Published 11 May 2015, https://www.gov.uk/government/publications/water-and-treated-water/water-and-treated-water visit on 01/09/2023.

[9] The UK government, Storm overflows discharge reduction plan, Policy paper, https://www.gov.uk/government/publications/storm-overflows-discharge-reduction-plan, visit on 01/09/2023

[10] Rezaei Kalvani S, Celico F. The Water–Energy–Food Nexus in European Countries: A Review and Future Perspectives. Sustainability. 2023; 15(6):4960. https://doi.org/10.3390/su15064960

[11] Li J, Yang X, Sitzenfrei R. Rethinking the Framework of Smart Water System: A Review. Water. 2020; 12(2):412. https://doi.org/10.3390/w12020412

[12] Ruixing Huang, Chengxue Ma, Jun Ma, Xiaoliu Huangfu, Qiang He, Machine learning in natural and engineered water systems, Water Research, Volume 205, 2021, 117666, ISSN 0043-1354, https://doi.org/10.1016/j.watres.2021.117666.

[13] Sheng Miao, Changliang Zhou, Salman Ali AlQahtani, Mubarak Alrashoud, Ahmed Ghoneim, Zhihan Lv, Applying machine learning in intelligent sewage treatment: A case study of chemical plant in sustainable cities, Sustainable Cities and Society, Volume 72, 2021, 103009, ISSN 2210-6707, https://doi.org/10.1016/j.scs.2021.103009.

[14] Song Pham Van, Hoang Minh Le, Dat Vi Thanh, Thanh Duc Dang, Ho Huu Loc, Duong Tran Anh; Deep learning convolutional neural network in rainfall–runoff modelling. Journal of Hydroinformatics 1 May 2020; 22 (3): 541–561. doi: https://doi.org/10.2166/hydro.2020.095

[15] Muhammed Sit, Bekir Z. Demiray, Zhongrun Xiang, Gregory J. Ewing, Yusuf Sermet, Ibrahim Demir; A comprehensive review of deep learning applications in hydrology and water resources. Water Sci Technol 15 December 2020; 82 (12): 2635–2670. doi: https://doi.org/10.2166/wst.2020.369

[16] N. Biggs, E. K. Lloyd, and R. J. Wilson. 1986. Graph Theory, 1736-1936. Clarendon Press, USA.

[17] Wu, L., Cui, P., Pei, J., Zhao, L., & Guo, X. (2022, August). Graph neural networks: foundation, frontiers and applications. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 4840-4841). https://dl.acm.org/doi/10.1145/3534678.3542609

[18]L. A. Jeni, J. F. Cohn and F. De La Torre, "Facing Imbalanced Data--Recommendations for the Use of Performance Metrics," 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, Geneva, Switzerland, 2013, pp. 245-251, doi: 10.1109/ACII.2013.47.

[19] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, Maosong Sun, Graph neural networks: A review of methods and applications, AI Open, Volume 1, 2020, Pages 57-81, ISSN 2666-6510, https://doi.org/10.1016/j.aiopen.2021.01.001.