Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-6 (2022) 661–666

Graph-Based Learning for Leak Detection and Localisation in Water Distribution Networks ⋆

Garðar Örn Garðarsson * Francesca Boem * Laura Toni *

* EEE Department, University College London
gardar.gardarsson.20@alumni.ucl.ac.uk, {f.boem, l.toni}@ucl.ac.uk

Abstract: We propose the application of geometric deep learning techniques to the challenging leak detection and isolation problem in water distribution networks (WDNs). Specifically, we train two Chebyshev polynomial kernel Graph Convolutional Networks for the task of prediction, and reconstruction of nodal pressures in a WDN. Comparing the two network outputs (a predicted healthy model state with a reconstructed observation) a residual signal is obtained and analysed to detect leakages. By exploiting topological properties in the proposed approach, leakage isolation is also performed. We benchmark our method on the BattLeDIM 2020 dataset.

Keywords: Fault detection and diagnosis, water distribution systems, geometric deep learning.

1. INTRODUCTION

Water is commonly supplied to households through a Water Distribution Network (WDN). Kingdom et al. (2006) found that globally on average, around 35% of produced water is lost during distribution, with strong societal and economic implications. It is therefore fundamental to develop efficient techniques for the prompt detection and isolation of leakages in WDNs. However, major obstacles are represented by the scarcity of measurements and the uncertainty in demand, making the leakage isolation problem a very challenging one.

Methods for detecting the presence of leaks in WDNs can be classified as *passive* and *active*. *Passive* leak detection systems are those that require maintenance personnel to survey a particular site in search of leaks (Chan et al. (2018)). These will thus not be helpful in reducing *awareness* times of hidden leakages with any significance, although they remain very valid for locating faults once detected. *Active* systems are those which use continuous monitoring of signals from the network to detect leakages in an automated manner. These may be further classified into *model-based* and *data-driven* approaches as described by Chan et al. (2018). *Model-based* approaches for leak detection, rely on the creation of hydraulic mathematical models that simulate the water distribution network, and are used for estimating expected conditions at a given point in the network, e.g., Wu et al. (2018); Soldevila et al. (2016). This model can be learned exploiting the topological structure of the network, as done by Vrachimis et al. (2021), where a node's hydraulic state is determined by modelling the WDN as a graph with nodes representing junctions, water demand locations, reservoirs and tanks, and edges representing pipes. This method obtained strong detection performance and good localisation for significant leaks, but it relies on historical demand data, which may not be accessible everywhere, as some utilities bill customers from their property's size, rather than by meter.

On the other hand, *data-driven* approaches learn to detect leaks given historical data from the WDN. WDN sensor measurements are commonly collected in supervisory control and data acquisition (SCADA) systems and have historically been constrained to few, sparse measurements points. These methods however require large amounts of data Chan et al. (2018). Their development is further hampered by leak data being a minority in the data collected Mounce et al. (2010), and measurement noise. Romero et al. (2020) introduces a data-driven method that exploits pressure measurements. The complete network state estimation is achieved by linear interpolation by solving an optimisation problem. However the method is, *i*) only applied to densely monitored regions of the WDN, *ii*) topology-agnostic, which might result in limited prediction especially in sparsely monitored regions or in highly data-hungry learning models. Thus, the importance of developing methodologies that are topology-(graph-)aware is apparent.

A model-transient-based approach that utilises deep-learning for leak identification was proposed by Kang et al. (2018), where graph-based search was used for leak localisation. Specifically, the proposed method detects leaks as transient oscillations in the vibration signals, using a convolutional neural network (CNN). After leak detection, a localisation algorithm is proposed using the graph-based wavelet method introduced in Srirangarajan et al. (2013), showing good performance. However, the latter has been carried out with engineered tests at night, resulting in unrealistic test conditions Wu and Liu (2017). Despite this, their graph-based method was accurate in localisation.

In this paper, we propose a novel active method for leakage detection and isolation that combines data-driven

⋆ The work of G.Ö.Garðarsson has been supported by the Chevening Scholarship, the UK government's global scholarship programme, funded by the Foreign, Commonwealth and Development Office (FCDO) and partner organisations

10.1016/j.ifacol.2022.07.203

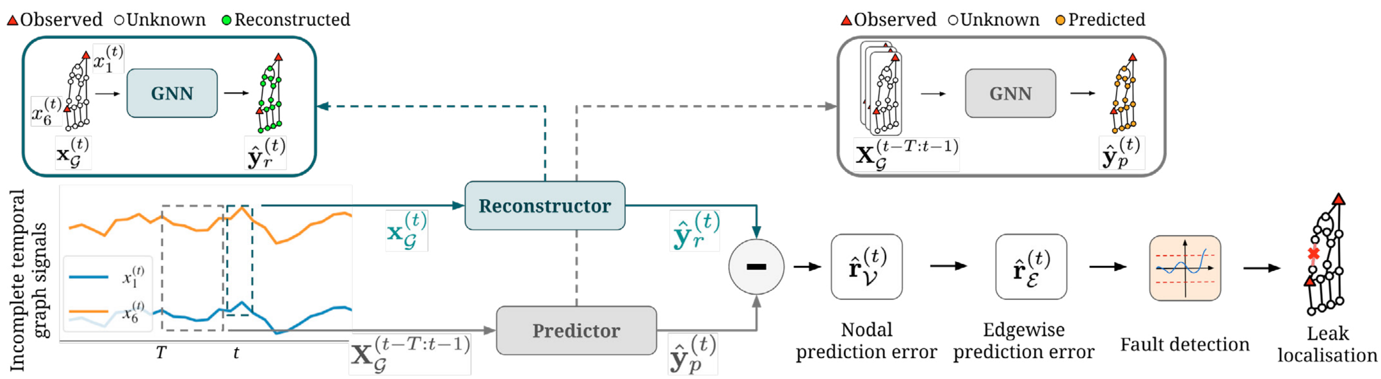

Fig. 1. Proposed architecture, two GNNs reconstruct and predict complete pressure scenes, $\hat{\mathbf{y}}_r^{(t)}$ and $\hat{\mathbf{y}}_p^{(t)}$, from a sparse graph signal, $\mathbf{x}_{\mathcal{G}}^{(t)}$, and a set thereof, $\mathbf{X}_{\mathcal{G}}^{(t-T:t-1)} = [\mathbf{x}_{\mathcal{G}}^{(t-T)}, \mathbf{x}_{\mathcal{G}}^{(t-T+1)}, ..., \mathbf{x}_{\mathcal{G}}^{(t-1)}]$, respectively. The difference between the two, $\hat{\mathbf{r}}_{\mathcal{V}}^{(t)}$, represents a per-node prediction error, which is translated to a per-edge basis $\hat{\mathbf{r}}_{\mathcal{E}}^{(t)}$, such that fault detection methods applied to the signal directly returns a leaky pipe candidate, in line with benchmark objectives.

techniques to learn a WDN model, with a model-based approach for detection. Similarly to Romero et al. (2020), we aim to estimate the complete network state from data, but we consider a topology-aware approach. Namely, where Romero et al. (2020) use linear-graph interpolation to obtain the complete network state based on the sparse sensor measurements, we adopt a *Graph Neural Network* (GNN) with a Chebyshev kernel to predict the complete state of the network. In more detail, we propose a novel method based on GNNs, that exploit the topology of the WDN to reconstruct and predict nodal pressures where measurements are not available. The GNN regressors are trained on a hydraulic simulation of the WDN in a healthy state before being applied to sparse pressure measurement data. From the reconstructed (observed) and predicted (expected) pressure states, a discrepancy is calculated, on which a statistical fault detection method is applied to locate anomalies such as leaks.

This work thereby addresses the leakage localisation problem with relatively few priors, and is not dependant on demand or flow data. Beyond being topology-aware, our method differs from existing ones in the residual generation. While Romero et al. (2020) analyse least square distances to the best line fitted through the point cloud formed by comparing the inferred network state with the nominal state, we convert nodal prediction errors to residual signals on the edges (corresponding to the pipes of a WDN), and evaluate their moving averages against a statistical threshold, to identify leak candidates. For the graph-based inference, we build on contributions by Hajgató et al. (2021) for nodal pressure reconstruction, by 1) extending the learning objective for nodal pressure prediction; 2) analysing different edge-weight assignment of the graph, and 3) introducing a new graph constructions from the WDN based on self-loops to sensor nodes, which led to better detection.

Looking at works that adopted GNNs, Hajgató et al. (2021) recently published their findings on their application for nodal pressure signal reconstruction. Their inputs are nodal observations, i.e., hydraulic states of the monitored nodes, and their outputs are the pressures of all nodes. Values proportional to the hydraulic loss in the network's pipes, and logarithmic variants thereof, were assigned to the adjacency matrix but not found to yield improvements over a binary one. Our paper exploits instead the nodal reconstruction logic proposed in Hajgató et al. (2021) and extends it for the purpose of leakage detection and isolation. While GNNs have been applied to model WDNs in the work of Garzón et al. (2021), to the best of our knowledge this is the first work exploiting GNNs for leakage detection and localisation.

2. PROPOSED METHODOLOGY

We now introduce our proposed solution to the problem of leakage detection and localisation, aimed at identifying leakages from a partly observed WDN. To address it, the hydraulic state of the WDN is estimated at every intersection by processing physical time-series signals from the network in conjunction with topological information about the piping system. As depicted in Fig. 1, we design two deep learning models trained under non-leaky conditions, i) a *reconstructor*, that infers nodal pressure for the current time step given sparse sensor measurements, and ii) a *predictor*, that predicts nodal pressure for the next time step, given a window of previous measurements.

We denote the data observed at time t on node v of the WDN by $x_v^{(t)}$. The signal observed at time t at all nodes is denoted by $\mathbf{x}_{\mathcal{G}}^{(t)} = [x_1^{(t)}, x_2^{(t)}, \ldots, x_N^{(t)}]$, with $x_v^{(t)}$ =NaN if the data is not observed at that location at time t. We assume a sparse signal such that the time-varying data is observed only at a few locations, meaning that the majority of $x_v^{(t)}$ will be NaN. Given $\mathbf{x}_{\mathcal{G}}^{(t)}$ as input, the reconstructor infers the pressure for all missing nodes; which we denote by $\hat{y}_r^{(t)}$. The predictor instead predicts the complete nodal pressures given historical data observed over a time-widow of T instants and denoted by $\mathbf{X}_{\mathcal{G}}^{(t-T:t-1)} = [\mathbf{x}_{\mathcal{G}}^{(t-T)}, \mathbf{x}_{\mathcal{G}}^{(t-T+1)}, \ldots \mathbf{x}_{\mathcal{G}}^{(t-1)}]$. The output of the predictor, namely $\hat{y}_p^{(t)}$, is compared with $\hat{y}_r^{(t)}$ to generate a residual. The node residual is then converted into edge-residual and analysed to localise leakages. Following is a description of how we construct the graph from the WDN topology, and how we identify the problem of inference as a graph signal processing (GSP) method. We then describe, i) the architecture that we propose for the

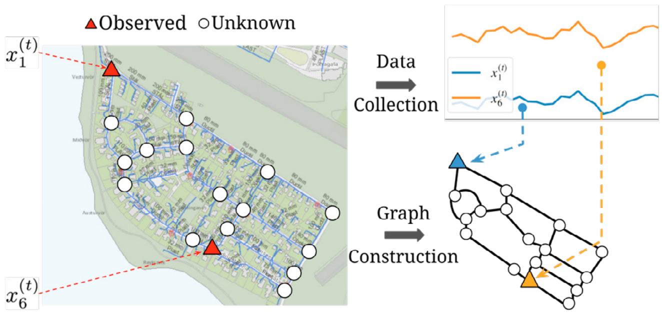

Fig. 2. Undirected graph construction from topological information about the WDN. Data collected over time at sensor location represents the signal on graph.

two networks (reconstructor and predictor), and *ii*) the conversion and analysis of the residual.

2.1 Topological Graph Construction

We start with modeling a WDN (topological objects of junctions connected by pipe segments) using a graph, as shown in Fig. 2.

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V}$ is a set of nodes where each $v \in \mathcal{V}$ denotes a node on the graph (junctions of the WDN) and we consider N nodes (i.e., $|\mathcal{V}| = N$); $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges with $e_{ij} \in \mathcal{E}$ connecting node i and node j. It is worth noting that water in WDNs is characterised by a certain direction of flow, calling for directed graphs; instead in this paper we consider an undirected graph. Using undirected graphs was shown to yield satisfactory results for pressure reconstruction in Hajgató et al. (2021) and will thus be adapted here as it has more applicability for real-life situations. As novel contribution in the graph construction, we introduce self-loops for the node sensors. This increases the node degree of pressure sensor nodes in the WDN, amplifying the influence of the sensor nodes during reconstruction. Experimentally, higher node degrees graph yield to lower detection errors than graphs without self-loops.

While $\mathcal{G}$ captures the topological structure of the network, the relation of a given junction to its neighbours is described by the *adjacency matrix*, $\mathbf{A}$. The latter reveals the connectivity between the nodes of a graph, which can reflect multiple factors, such as the length of the pipe separating them, its diameter, the pipe coefficient and the direction of flow. Three primal weight assignments are investigated, namely, *i*) unweighted edges, *ii*) pipe length-based weights and variants thereof, and *iii*) hydraulic loss-based weights for the pipe segments.

Unweighted edges is the most elementary approach, which assumes binary weight value assignment as $\mathbf{A}_{vu} = 1$ if $e_{vu} \in E$, 0 otherwise. For a *pipe length weighted* adjacency matrix, $\mathbf{A}_{vu} = l_{vu}$, if $e_{vu} \in E$, 0 otherwise, with l_{vu} being the length of the pipe between node u and node v. Finally, a *hydraulic loss weighted* graph assumes the edge weight as an estimated head, or energy loss for the pipe segment calculated empirically using the Hazen-Williams equation, $\mathbf{A}_{vu} = 4.727 \cdot C_{vu}^{1.852} \cdot d_{vu}^{4.871} / l_{vu}$, if $e_{vu} \in E$, 0 otherwise(Hajgató et al. (2021)). The different modes of weight assignment are treated as a hyperparameter when training the GNNs, with pipe-length ones found to yield lowest reconstruction- and prediction errors.

The last step is to define the signal on the graph. Each node holds a T-dimensional *feature vector*, which represents the hydraulic state of the node v over time. The signal observed at time t at the different nodes on the graph is denoted by $\mathbf{X}_{\mathcal{G}}^{(t)} = [x_1^{(t)}, x_2^{(t)}, \ldots, x_N^{(t)}]$, with $x_v^{(t)}$ =NaN if the data is not observed at that location at time t. Hence, reconstructing and predicting the pressure measurements in the WDN is cast as a problem of graph signal processing. Given a partly observed WDN, its hydraulic state is estimated at every intersection of the network by processing the physical time-series signals in conjunction with topological information about the piping system. This effectively describes a graph-signal reconstruction problem and can be addressed with the application of GNNs. In the following section, we first introduce the field of GNNs, motivating our architectural choices. We then conclude the section with the proposed methodology.

2.2 Graph Signal Reconstruction and Prediction

Graph Neural Networks GNNs are neural network architectures that can process geometrical data (non-Euclidean) with the goal of extracting node features combined with graph topology to solve tasks such as prediction, forecasting and classification (Ortega et al. (2018)). In the literature, their architectures can be classified into spatial- and spectral-based GNNs. The former, implement a message-passing process along the graph at each graph-convolution operation, which corresponds to the multiplication of a convolution kernel with the corresponding node feature vectors, followed by a sum or a mean rule. These methodologies are very simple and adapt to different network topologies, however they suffer from a well known oversmoothing problem. In the case of deep networks with many hidden layers, the extracted features assimilate across the graph. The latter, spectral-based architectures, build on multiple hidden layers, each one performing spectral convolutions, defined from a GSP point of view. GSP provides a notion of frequency and a graph Fourier transform, allowing for filtering operation in the spectral (frequency) domain. As a consequence, a graph convolutional layer, can be written by a sum of filtered signals followed by an activation function, where each filter is defined in the graph spectral domain (using an eigendecomposition of a graph Laplacian). These networks alleviate the oversmoothing effect, while suffering from a large computational burden induced by the forward/inverse graph Fourier transform. To simplify the computational complexity, some approaches based on parameterisation using Chebyshev polynomials Defferrard et al. (2017) have been proposed, known as the *ChebNet* model.

Our Architecture For a very sparsely monitored WDN, as the one in our work, a *deeper* network, consisting of more hidden layers, might be needed. At the same time, to ensure a localisation of the leakage, we need to preserve as much as *local* information as possible. Oversmoothing would not allow us to preserve high-frequency (hence local) information, thus we adopt spectral-based architectures. Our initial ChebNet model is based on the one provided by Hajgató et al. (2021) for the WDN of Richmond, which

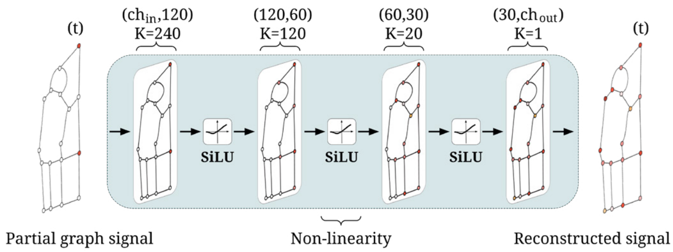

Fig. 3. ChebNet model for graph signal reconstruction. The predictor (Fig. 1) has the same structure but the input channel ch_{in}, has a depth of 3 time periods.

has a similar number of junctions and pipes as the WDN we benchmark our model on. The model consists solely of Chebyshev convolutional layers, which are regularised by the weight decay of the gradient descent optimiser. Layer weights are initialised by the Xavier normal distribution (Glorot and Bengio (2010)) and biases set to zero. The hidden layers are activated with a *SiLU* sigmoid linear unit (Hendrycks and Gimpel (2016)) and the output layer is passed through a sigmoid activation. The ADAM optimiser (Kingma and Ba (2015)) is used due to its properties of having an adaptive learning rate with momentum, that adjusts to the function to be optimised.

An exhaustive ablation study to optimise hyper parameters was provided by Hajgató et al. (2021) leading to the following architecture for the WDN of Richmond: 4 Chebyshev spectral graph convolutional layers with the degree of the polynomial in the hidden layers set to $[K_1, K_2, K_3,] = [240, 120, 20]$, and filter sizes $[F_1, F_2, F_3] = [120, 60, 30]$. This forms the basis of our architecture, depicted in Fig. 3. The input channel size, ch_{in}, here refers to the depth of the input; in the case of the reconstructor, this will then equal one, as a single input graph is used to generate the output, whereas for the predictor, this will be equivalent to the T number of time step used to generate the output prediction. From this basis, we optimised the graph construction methodologies (weights assignment, signal scaling methodologies, and with or without self-loop for observed nodes) as motivated in Sec. 2.1, and the window size for the predictor.

Detection and Localisation Methodology We now formalise the proposed method with Algorithm 1 and focus on *i*) the estimation error, *ii*) the conversion from node to edge error (or residual), *iii*) fault detection and leakage localisation.

Node Estimation Error Evaluation The outputs of the predictor and reconstructor are compared, by evaluating at each iteration the difference between the reconstructed pressure at time t with the predicted one, which was generated at $t-1$ using past information. The difference is the *prediction error* or *residual signal*, namely $\boldsymbol{r}(t) = [r_1(t), r_2(t), \ldots, r_N(t)] = \hat{\boldsymbol{y}}_p^{(t)} - \hat{\boldsymbol{y}}_r^{(t)}$. Under nominal non-faulty conditions, the residual error should have stationary behavior and should be caused mainly by the presence of random noise. When this residual does not display a stationary behavior anymore, this implies a change in the underlying model describing the behaviour of the WDN. In this paper, we identify this change as the presence of a leakage.

Algorithm 1 Leak detection pseudocode

```
for every_timestep : t do
    readings ← readPressureSensors()
    if first_run then
        next_prediction ← GNN_1.predict(readings)
    else
        // Calculate prediction error
        last_prediction ← next_prediction
        reconstruction ← GNN_2.predict(readings)
        node_err ← reconstruction − last_prediction
        // Convert nodal error to edge error
        for connected_nodes : i, j in node_error do
            edge_err ← node_err(i) − node_err(j)
        end for
        // Assess fault condition
        for each_edge : v in edge_err do
            thresh ← α × edge_err(v).rolling(m).std()
            mv_avg ← |edge_err(v).rolling(m).mean()|
            if mv_avg > thresh then
                Leak detected, log edge, v.
            end if
        end for
        // Filter duplicate alarms
        for faulty_edges : v do
            for neighbour in k-hop distance of v do
                if neighbour has fault in the past n steps then
                    Suppress fault alarms from edge
                end if
            end for
        end for
        next_prediction ← GNN_1.predict(readings)
    end if
    Raise alarm for edges v that are in alarm state
end for
```

Edge error evaluation Instead of considering the residual signal at each node, a novel way to process the residuals is proposed, whereby the signals are considered on a per-edge basis, rather than a node one. This stems from the notion that leaks commonly happen along a pipe rather than at a junction, and for it being directly applicable to the benchmark dataset. In the case of an edge e_{uv}, connecting the two neighbouring nodes v and u, we would consider its residual as $r_{uv}^{(E)}(t)$ as $r_{uv}^{(E)}(t) = r_v(t) - r_u(t)$.

Assess fault condition. Once the edge error is evaluated, we need to detect possible faulty conditions. The strategy consists of calculating the moving average of the residual signal $r_{uv}^{(E)}(t)$ of a given edge uv in the network, for a window of size m at observation time t,[1] as per $\bar{r}_{uv}^{(E)}(t) = 1/m \sum_{j=0}^{m-1} r_{uv}^{(E)}(t-j)$ and in defining a detection logic as:

$$\mathcal{FD} = \begin{cases} 0 & \text{if } |\bar{r}_{uv}^{(E)}(t)| < \bar{\rho}_{uv}(t) \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

where, $\bar{\rho}_{uv}(t) = \alpha\bar{\sigma}_{uv}(t)$, and α is a parameter that allows to tune the acceptable false alarm rate based on Chebyshev inequalities, with $\bar{\sigma}_{uv}^{(t)}$ being the rolling standard deviation of the residual signal for a window of size m. This test is then individually applied on each residual signal at every time step to obtain a classification of whether the signal is deemed indicative of a leak being present in its neighbourhood or not.

[1] For the sake of notation simplicity, we omit the dependence on m.

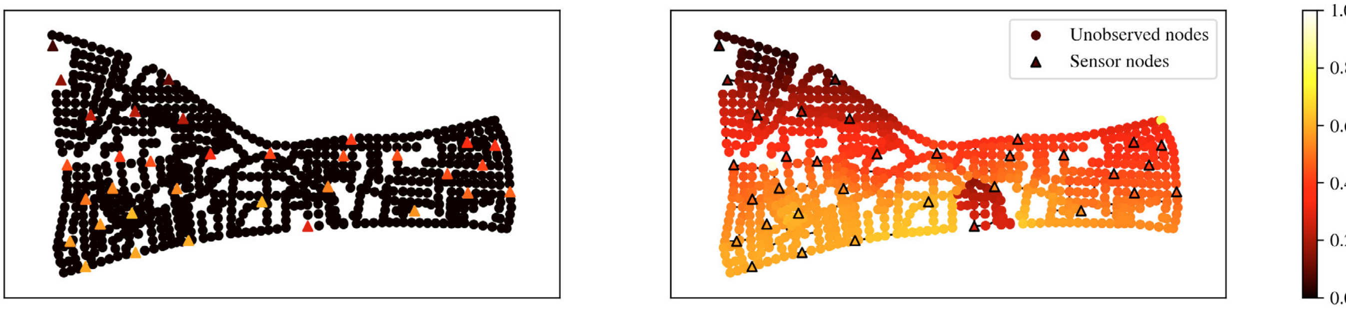

Fig. 4. Sparse, unseen pressure measurement (left) and its reconstructed pressure scene (right). The measurements are normalised to the interval $[0, 1]$, but operating pressure in L-Town is normally between $\sim 20 - 70\ m$.

3. SIMULATION RESULTS

We validate our method on the *BattLeDIM*, Battle of the Leakage Detection and Isolation Methods, challenge (Vrachimis et al. (2020)). Its dataset comes from the hypothetical *L-Town*, with a WDN spanning a total pipe length of 42, 6 km. The network is segmented into three areas and composed of 782 nodes. The WDN is further instrumented with, one tank level sensor, three flow sensors, and 33 pressure sensors. The competition objective is to detect as many of the leaks that occurred in the 2019 dataset as possible. For this, researchers are provided with a nominal model of the WDN, contained in an *EPANET* hydraulic simulation input file, which has $\pm 10\%$ inherent uncertainties, due to demand and topological ambiguity. This nominal model is used for the training of the GNN reconstructor and predictor, as per Sec. 2.2. Competitors are further given sensor measurements from the WDN's control system related to the year of 2018, along with a documented leakage dataset for 2018, of when a leak occurred in the network, when it was fixed and its estimated size. The considered leakages include *background leaks* measuring $1-5\%$ of the network inflow, *medium pipe bursts* representing $5 - 10\%$ of the average system inflow, and *large pipe bursts* of entity $> 10\%$. The average inflow of the system is $\sim 180 m^3/h$. Both *abrupt* and *incipient* leakages occur in the dataset. A scoring function is further supplied in the challenge to evaluate the accuracy of the methods, converting the scores into *Euros* saved from non-revenue water and reparation costs, by the assessed detection method. The sooner a leak is detected and the closer it is to the actual source, the higher the economic reward. *True positives* are defined as detection alarms raised during the lifetime of a leakage and within a distance of 300 m from the leakage location.

Equipped with this competition dataset and model, the proposed GNN reconstructor has been trained. For what concerns the predictor GNN, a window of past partially observed measurements is needed for the inference of nodal pressures at the next time step. The size of the window T has been selected during the training phase, considering values of $T = [1, 2, 3, 6, 12, 24]$, corresponding to windows of 5 min up to 2 hr being used as the input to the GNN, since the data is sampled at 5 min intervals. A value of $T = 3$ was then selected.

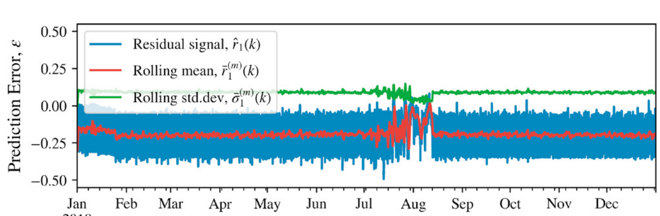

Fig. 5. Residual signal of node 1 for the year 2018.

The model is then tested on unseen data from the historical pressure measurements of 2019. Fig. 4 shows one inferred pressure scene for the 782 nodes, from a single observation of 33 measurements. We then analysed node residual signals $r_v(t)$ at each node v, which are computed at every timestep. An example can be seen for node 1 in Fig. 5. The edge-wise residual signals of pipes in the neighbourhood of a leak in pipe 31 is shown in Fig. 6.

As evident from Fig. 6, the signal mean values are not equal to zero. We deduct the mean value computed at each single pipe only from observations when the pipe is known to be in a non-faulty state, and its neighbors too. Edge-wise residual analysis is then performed on the moving averages of the signals for leak detection. The rolling means $\bar{r}_v(t)$ are calculated described in the previous section and the detection depends on the condition in Eq. 1. There are two parameters that need to be set in the detection method: α the false alarm rate parameter that scales the detection threshold, and the size of the window for calculating the moving average and threshold.

A window size of $m = 7$ days and a false alarm rate parameter $\alpha = 1.0$ are chosen. The leakage candidates are filtered, so that no more than a single detection may be returned on a given day in a neighborhood of 6 hops. The reasoning behind this approach is the intuition that information of the leakage propagates the network from the leakage origin, thus causing false-alarms in areas non-related to the leakage. The simulation results obtained with this configuration led to an Economic score of €326.521, 19 True positives, 138 False positives, and 4 False negatives, as shown in Table 1. It is worth noting that with respect to the teams in the BattleDIM challenge, our method only uses pressure measurements, while other entries also used flow measurements and smart meter measurements. We recognise that the proposed method suffers from false positives, with 138 false alarms given over the year. By an initial analysis, we have observed a strong dependency of the rate from both the topology and the

Team Name	Score	TPR	FP
Proposed method	**€326.521**	**82,61%**	*138*
Tongji-Team	€264.873	56,52%	3
Under Pressure	€260.562	65,22%	4
IRI Romero et al. (2020)	€210.772	43,47%	**1**
Leakbusters	€195.490	47,83%	7
Tsinghua	€167.981	47,83%	5
UNIFE	€127.626	43,47%	4

Table 1. Performance comparison in terms of economic score, true positive rate and false positives.

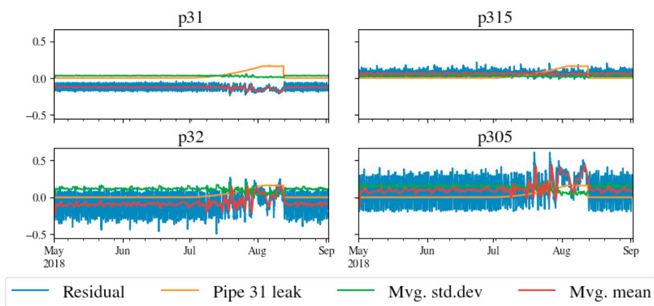

Fig. 6. Edge-wise residual signal of 1-hop neighbours to the leak in pipe 31.

parameters, leading to propose as future works a graph-based processing for reducing the false alarms.

4. CONCLUSIONS

In this work, a novel approach for leakage detection and localisation in WDNs has been proposed, combining data-driven techniques and model-based logic, where GNNs are used to reconstruct and predict pressure values. The discrepancy in the two models' outputs describes a residual signal, which is statistically analysed for changes for leakage detection, and the embedded topological information in the graph neural networks models is exploited for leakage localisation. The approach is evaluated on the BattLeDIM benchmark and obtains the highest economic score among the contestants that competed in the challenge in 2020. The method however suffers from false positives, resulting by the leakage effect propagating the network. Still, the impressive economic score suggests that the method is sensibly detecting leakages, and thus future effort will be devoted to improve the analysis of the residual signals, to overcome this limitation. To this end, other statistical methods for change detection, as well as neural networks classifiers will be investigated. A strategy will then be developed to translate the node-wise detections into edges.

REFERENCES

Chan, T.K., Chin, C.S., and Zhong, X. (2018). Review of Current Technologies and Proposed Intelligent Methodologies for Water Distributed Network Leakage Detection. *IEEE Access*, 6, 78846–78867.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2017). Convolutional neural networks on graphs with fast localized spectral filtering.

Garzón, A., Bentivoglio, R., Isufi, E., Kapelan, Z., and Taormina, R. (2021). Modeling water distribution systems with graph neural networks. In *EGU General Assembly Conf. Abstracts*.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proc. of 13th Int. Conf. on Artificial Intelligence and Statistics*, volume 9, 249–256.

Hajgató, G., Gyires-Tóth, B., and Paál, G. (2021). Reconstructing nodal pressures in water distribution systems with graph neural networks. *CoRR*, abs/2104.13619. URL `https://arxiv.org/abs/2104.13619`.

Hendrycks, D. and Gimpel, K. (2016). Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415. URL `http://arxiv.org/abs/1606.08415`.

Kang, J., Park, Y.J., Lee, J., Wang, S.H., and Eom, D.S. (2018). Novel leakage detection by ensemble cnn-svm and graph-based localization in water distribution systems. *IEEE Trans. on Industrial Electronics*, 65(5), 4279–4289.

Kingdom, B., Liemberger, R., and Marin, P. (2006). The Challenge of Reducing Non-Revenue Water (NRW) in Developing Countries. Water Supply and Sanitation Board Discussion 8, The World Bank.

Kingma, D.P. and Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Mounce, S.R., Mounce, R.B., and Boxall, J.B. (2010). Novelty detection for time series data analysis in water distribution systems using support vector machines. *J. of Hydroinformatics*, 13(4), 672–686.

Ortega, A., Frossard, P., Kovačević, J., Moura, J.M.F., and Vandergheynst, P. (2018). Graph signal processing: Overview, challenges, and applications. *Proc. of the IEEE*, 106(5), 808–828.

Romero, L., Blesa, J., Alves, D., Cembrano, G., Puig, V., and Duviella, E. (2020). Leak Localization In Water Distribution Networks Using Data-Driven And Model-Based Approaches.

Soldevila, A., Blesa, J., Tornil-Sin, S., Duviella, E., Fernandez-Canti, R.M., and Puig, V. (2016). Leak localization in water distribution networks using a mixed model-based/data-driven approach. *Control Eng. Practice*, 55, 162–173.

Srirangarajan, S., Allen, M., Preis, A., Iqbal, M., Lim, H., and Whittle, A. (2013). Wavelet-based burst event detection and localization in water distribution systems. *J. of Signal Proc. Systems*, 72, 1–16.

Vrachimis, S.G., Eliades, D.G., Taormina, R., Ostfeld, A., Kapelan, Z., Liu, S., Kyriakou, M.S., Pavlou, P., Qiu, M., and Polycarpou, M. (2020). Dataset of BattLeDIM: Battle of the Leakage Detection and Isolation Methods. doi:10.5281/zenodo.4017659.

Vrachimis, S.G., Timotheou, S., Eliades, D.G., and Polycarpou, M.M. (2021). Leakage detection and localization in water distribution systems: A model invalidation approach. *Control Eng. Practice*, 110, 104755.

Wu, Y. and Liu, S. (2017). A review of data-driven approaches for burst detection in water distribution systems. *Urban Water Journal*, 14(9), 972–983.

Wu, Y., Liu, S., Smith, K., and Wang, X. (2018). Using correlation between data from multiple monitoring sensors to detect bursts in water distribution systems. *J. of Water Resources Planning and Management*, 144(2).