Liquor-HGNN: A heterogeneous graph neural network for leakage detection in water distribution networks

Melanie Schaller[1,*], Michael Steininger[1], Andrzej Dulny[1], Daniel Schlör[1] and Andreas Hotho[1]

[1]Julius-Maximilians Universität Würzburg (JMU), Center for Artificial Intelligence (CAIDAS), Chair for Informatics X (Data Science), Am Hubland, Würzburg, 97074, Germany

Abstract

In this paper, we introduce the Liquor-HGNN model, a novel approach for detecting and localizing leaks in drinking water distribution networks (DWDNs) through the utilization of heterogeneous graph learning. By leveraging a preprocessing model, our approach mounts the challenges posed by data sparsity and sensor heterogeneity limitations. Liquor-HGNN outperforms all other approaches on the same dataset in terms of Economic score. Here, the Economic Score function iterates over the detected leakages, finds the closest pipes to each detected leakage, and calculates the score contribution for each true detection based on the detected distance as well as on the starting time of the leakages. To the best of our knowledge, Liquor-HGNN represents the first-ever application of a heterogeneous Graph Neural Network (GNN) specifically tailored for leak detection in DWDNs.

Keywords

Deep learning and representation learning, Temporal and spatio-temporal data analytics, Applications of machine learning in engineering, Leakage detection and localization

CEUR-WS.org/Vol-3630/LWDA2023-paper40.pdf

1. Introduction

Water pipe systems are crucial for providing clean drinking water to communities, but they are prone to leaks and ruptures, leading to resource waste and damage [1]. Thus, detecting and locating leaks in drinking water distribution networks (DWDNs) is essential for maintaining the integrity and sustainability of these systems [2]. This paper introduces the Liquor-HGNN model, a novel approach that utilizes heterogeneous graph learning to accurately identify the start time of leaks and to pinpoint leaky pipes in DWDNs.

1.1. Background and Motivation

Leakage in the water distribution network is a major source of non-revenue water. Non-revenue water refers to the discrepancy between the volume of water supplied to the distribution network and the amount of water actually billed to customers. On average, non-revenue water accounts

LWDA'23: Lernen, Wissen, Daten, Analysen. October 09–11, 2023, Marburg, Germany

*Corresponding author.

schaller@informatik.uni-wuerzburg.de (M. Schaller); steiniger@informatik.uni-wuerzburg.de (M. Steininger); dulny@informatik.uni-wuerzburg.de (A. Dulny); schloer@informatik.uni-wuerzburg.de (D. Schlör); hotho@informatik.uni-wuerzburg.de (A. Hotho)

https://github.com/MilanShao/ (M. Schaller)

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

for approximately 35 percent of the total water supply [3]. This highlights the pressing need for efficient leak detection methods in order to prevent wastage of resources and maintain the overall quality and reliability of the water system.

1.2. Problem Statement and Relevance

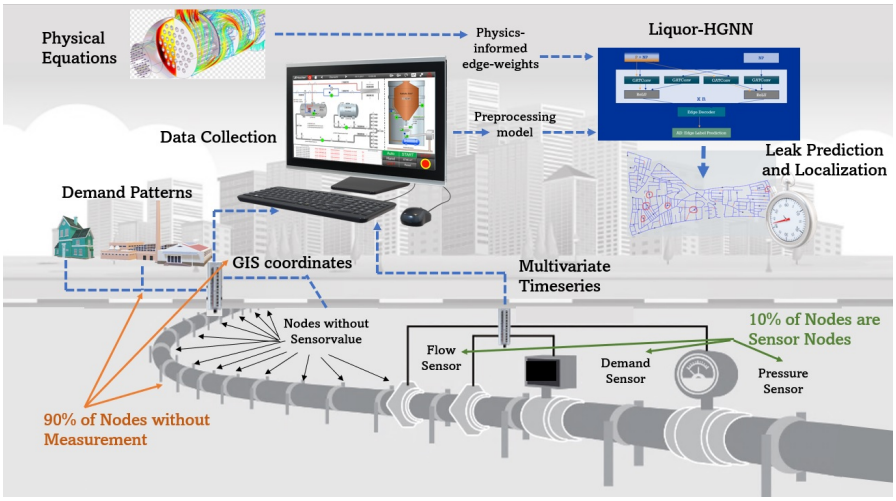

Figure 1: Visualization of the underground pipelines with different sensor types (demand, pressure and flow), demand patterns (here dividing between residential and commercial demand), the data collection of GIS (Geo Information System) Coordinates and the Liquor-HGNN pipeline for leak detection and localization.

Traditional visual inspection methods are impractical for underground pipelines, leading to the installation of sensors in water distribution networks [4]. However, due to financial and spatial limitations, these sensors are sparsely distributed across pipeline sections [5]. The complexity and diversity of sensor data (e.g. the different sorts of sensors, that are used to measure DWDNs like flow, demand or pressure sensors), along with the large distances between sensors (see pipe sections in Fig. 1) and different types of leaks, pose challenges in accurately determining the leaky pipe section and starting time of leaks. These different types of leaks include background leaks, incipient leakages, as well as abrupt leakages such as medium pipe bursts and large pipe bursts [6]. Additionally, the collected time-series data differ according to the different measuring principles of sensors (see Fig. 1). Furthermore, nodes in the network differ in their usage profiles [7], adding further complexity (see demand patterns on the left side in Fig. 1). Despite the high complexity of the task, it is a worthwhile goal to adequately meet the challenge of leakage detection and to satisfy the needs of the stakeholders ranging from water utility companies, water management agencies until the general public [8].

1.3. Objectives

The primary objective is to develop an effective model for detecting leaks in water pipe systems. The model should learn the relationships between different system components, including

sensor types (e.g. flow sensors, pressure sensors, demand measurements, smart meters [5]), network information (i.e. pipe-attributes corresponding to edge-attributes like diameters, length, roughness coefficients, as well as geospatial data like coordinates or elevation and physical information), and user patterns (i.e. residential and commercial demand patterns, etc. [7]), to accurately detect and locate leaks.

Expressed from a more technical perspective, the objective of the model is to predict the edge-labels for leaky or non-leaky pipes at each time step based on the discrete spatio-temporal graph snapshot to localize anomalies. The specific objectives and quantifiable criteria for the success of the Liquor-HGNN approach are to localize leaks for all pipe sections (corresponding to edges of a graph) using measurement information from only around 10 percent of the nodes, to accurately forecast the start time of leaks and to propose an imputation method for handling missing data through prior forecasts.

1.4. Method

To address the complex task of leakage detection, we propose the Liquor-HGNN model[1]. The model selects the most suitable threshold based the Economic Score Metric, that is built upon the metric of KIOS research center [6]. The Economic Score function iterates over the detected leakages, finds close pipes to each detected leakage, and calculates the score contribution for each true detection based on the detected distance and the number of true detections. Our metric also applies a penalty if a leakage is not detected. The Liquor-HGNN model takes into account the physical positions of nodes, their connections via edges as well as the user patterns within the hydraulic demand distribution [7]. It leverages sparsely distributed sensor data and incorporates a graph attention mechanism to focus on the most relevant parts of the graph.

Methodologically, our approach involves a preprocessing procedure to interpolate demand values for every node. The demand over time is divided into different components, including a global trend, seasonal fluctuations, and weekly and daily variations. These components are derived from the predictions of the demand measurements and are separated according to their demand patterns. The predictions are then multiplied with the steady-state demand or base demand of each node [7].

To handle the heterogeneous graph structure, our model employs heterogeneous message-passing layers. This enables the model to capture and utilize the diverse sensor types present in the water pipe system as different node types in the network. The heterogenity allows to learn from the various types of relationships present in the graph. To introduce different edge-weights in heterogeneous graph neural networks, we use the HeteroConv wrapper [9]. This is described partly in the appendix.

2. Related Work

In this section, we first review the 18 submissions made to the BattLeDIM challenge, which serve as the main benchmarking results for the used dataset. The first submission is the ensemble multivariate changepoint detection (EMCPD) by Cheng et al. [10]. This method

[1] https://github.com/MilanShao/Liquor-HGNN/blob/main/README.md

combines six algorithms, including changepoint detection, non-parametric multiple changepoint analysis, divisive hierarchical estimation, kernel changepoint analysis, and Bayesian estimation of abrupt changes and trends. Huang et al. [11] propose a method consisting of five stages: model decomposition, data partitioning, nodal demand calibration, calibration residual-based leakage detection, and an improved vectorial angle method for leakage localization. The Leakbusters team, represented by Daniel et al. [12], developed a method with two algorithm components. The first component analyzes pressure differences between pairs of nodes using SCADA data to identify leakage events. The model is trained on normal time periods, and the reconstruction error is used to detect leakage events. Saldarriaga et al. [13] utilize a genetic algorithm (GA) to identify leaks, while Wang et al. [14] from the Tsinghua Team employ a multistage approach involving empirical mode decomposition, extraction of daily and weekly seasonalities, and emitter representation. The Under Pressure team, represented by Steffelbauer et al. [15], introduces a hierarchical approach for leak diagnosis. It involves demand calibration using AMR data and measured flows, mathematical optimization for calibrating pipe roughness, and the use of a dual network for leak start time detection and localization. Zhang et al. [16] employ a fuzzy similarity priority ratio for leakage localization, while Romero et al. [17] from the IRI team combine a model-based and data-driven methodology. Their approach determines the usage of the two approaches based on the characteristics of the network's different areas. Min et al. [18] and Blocher et al. [19] both use clustering algorithms to solve the task, and Liu et al. [20] employ an LSTM. Dopazo et al. [21] and Tan et al. [22] utilize machine learning approaches.

The winning Tongji Team, represented by Li et al. [23], developed the Multiple Leaks Detection and Isolation Framework (MLDIF). It employs a gradient iteration algorithm with variable steps to calibrate model parameters for each zone and predicts water consumption using AMR measurements.

Wu et al. [24] and Bhowmick et al. [25] both use time series data decomposition, while Marzola et al. [26] spatially localize anomalies through an enumerative procedure. Barros et al. [27] introduce a signal processing approach.

None of these approaches modeled the task using graph neural networks except for Gardarsson et al. [28], who trained two Chebyshev polynomial kernel Graph Convolutional Networks. Although they stated to gain a better economic score, their results have not been reproducable according to the delivered code and have thus been excluded from the conducted experiments.

On the other hand, there is an increasing number of papers that use Heterogeneous Graph Neural Networks (HGNN) for link prediction like HetGNN [29] or MTHetGNN [30].

3. Dataset Description

The BattLeDIM 2020 dataset [6] was created using a real water distribution network in Cyprus, covering a pipe length of 42.6 km. It contains information on pipe breaks, water losses, and two consumer types: residential and commercial, each with distinct demand patterns. The dataset consists of 782 nodes, 33 of them equipped with pressure sensors providing measurements every 5 minutes. The network consists of 905 pipe segments of steel pipes with a roughness coefficient between 120-140 of approximately 50 meters length, that are used as edges of the

Graph Neural Network. All measurements for five-minute timesteps for 80 percent of the year 2018 are considered as training data, 20 percent as validation data and the full year of 2019 as test dataset. Each node has a unique demand pattern for each consumer type, based on the statistical characteristics. Additionally, 82 Automated Metered Readings (AMRs) offer aggregated demand data (see Appendix). The dataset includes the physical network structure and coordinates of the pipes, making it suitable for water network simulations with the Water Network Tool for Resilience (WNTR), a Python package which supports pressure-driven demand simulations and leakage modelling [7]. We predict leakages in the ISO 8601 time format YYYY-MM-DD hh:mm, while the predicted location of the leakage is specified by the link ID.

4. Methodology

The methodology consists of the simulated-based preprocessing model for data interpolation and the the heterogeneous graph neural network for leakage.

4.1. Data Input and Preprocessing for missing data

Our proposed approach uses a heterogeneous graph neural network (HGNN) [29] to incorporate nodes with different node-types. As the water in water distribution networks is characterized by a certain flow direction, we use a directed graph. A heterogeneous graph can be defined as a tuple $G_h = (G, \Phi, \Psi)$, where $G = (V, E)$ is a graph object with given nodes V and edges E, $\Phi : V \to \mathscr{A}_V$ is a node type mapping function and $\Psi : E \to \mathscr{A}_E$ is an edge type mapping function. In the preprocessing step, missing demand values are predicted using the Neural Prophet model [31]. We separate the nodes measuring the signals of the whole distribution network into two sets of nodes, $P \subset V$ containing all nodes with pressure measurements and nodes with demand measurements $D \subset V$. Accordingly we define a signal at time t of the pressure sensor of node v as $x_v^P(t)$ and $x_v^D(t)$ as the signal of the demand sensor.

Note that demand of a node is dependent of its usage pattern. In this dataset usage is distinguished between residential demand and commercial demand which differ in volume and periodicity. In our experiment, nodes are either considered commercial or residential.

Due to the sparsity of data, ninety percent of the nodes lack any measurements, however, their usage pattern (commercial or residential) is given. To address the sparsity of the data and fill in missing demand values, the Neural Prophet model [31] is employed to learn and predict demands for each given usage pattern. Neural Prophet decomposes the time series into several components according to the following equation in order to find the best shaping functions.

For each pattern type c for commercial and r for residential, we use all nodes $v \in D$ to learn a Neural Prophet Model and predict the demand $\hat{x}_{\mathrm{c}}^D(t)$ and $\hat{x}_{\mathrm{r}}^D(t)$ at a given time t as:

$$\hat{x}_{\mathrm{c}}^D(t) = T_{\mathrm{c}}(t) + S_{\mathrm{c}}(t) + E_{\mathrm{c}}(t) + F_{\mathrm{c}}(t) + A_{\mathrm{c}}(t) + L_{\mathrm{c}}(t) \tag{1}$$

$$\hat{x}_{\mathrm{r}}^D(t) = T_{\mathrm{r}}(t) + S_{\mathrm{r}}(t) + E_{\mathrm{r}}(t) + F_{\mathrm{r}}(t) + A_{\mathrm{r}}(t) + L_{\mathrm{r}}(t) \tag{2}$$

where, $T_\bullet(t)$ - Trend at time t, $S_\bullet(t)$ - Seasonal effects at time t, $E_\bullet(t)$ - Event and holiday effects at time t, $F_\bullet(t)$ - Regression effects at time t for future-known exogenous variables, $A_\bullet(t)$ - Auto-regression effects at time t based on past observations, $L_\bullet(t)$ - Regression effects at time t

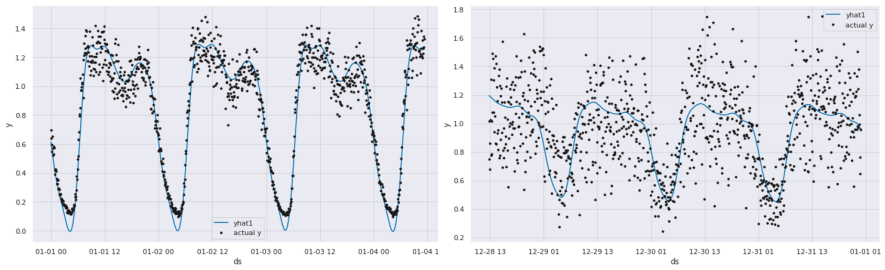

Figure 2: Results of the predicted values $\hat{x}_{\bullet}^{D}(t)$ (blue fit) in comparison to the black doted measured values $x_{\bullet}^{D}(t)$ for residential and commercial patterns.

for lagged observations of exogenous variable for each usage pattern $\bullet \in \{c, r\}$. Neural-Prophet is a special type of generalized additive models (GAM), that decomposes the time series into the above mentioned six types of components. The trend component uses an automatic changepoint detection, while seasonality makes use of Fourier term decomposition. For events the automatic given holidays of each country are taken. For the calculation of regression effects a real-valued regressor is used. The auto-regression effects are modelled by the so called AR-Net, which is a fully connected neural network [31]. This adds non-linear effects to the additive model. For the loss function the Huber loss is used, while the learning rate is optimized within a range test. Adam is used as optimizer.

In the next step, we adopt the base demand x_v^{base} calculation by Klise et al. [7] and multiply it with the predicted value to get overall demand $\tilde{x}_v^D$ for a node v:

$$\tilde{x}_v^D(t) = \begin{cases} x_v^{\text{base}} \cdot \hat{x}_{\text{c}}^D(t) & \text{if } v \text{ has commercial pattern} \\ x_v^{\text{base}} \cdot \hat{x}_{\text{r}}^D(t) & \text{if } v \text{ has residential pattern} \end{cases} \tag{3}$$

In Fig. 2 the fit of the predicted values $\hat{x}_{\bullet}^{D}(t)$, represented by the blue line, is plotted against the actual measured values $x_{\bullet}^{D}(t)$. We predict the demand values of the entire network to obtain a representative distribution, while the pressure values are still only taken at the nodes with pressure measurements without interpolation.

4.2. Heterogeneous Graph Learning

We separate between the two node-types $\mathcal{A}_V = \{\text{NP}, \text{P}\}$, where nodes of type NP only have the demand values (from demand preprocessing or given demands) and nodes of the node-type P additionally have the pressure measurements $x_v^P(t)$. Thus the feature vector $\mathbf{x_j}(t)$ for a node j is given as:

$$\mathbf{x_j}(t) = \begin{cases} x_j^P(t) \,\|\, x_j^D(t) & \text{if } j \in P \text{ and } j \in D \\ x_j^P(t) \,\|\, \tilde{x}_j^D(t) & \text{if } j \in P \text{ and } j \notin D \\ \tilde{x}_j^D(t) & \text{if } j \notin P \text{ and } j \notin D \\ x_j^D(t) & \text{if } j \notin P \text{ and } j \in D \end{cases} \tag{4}$$

Here, || denotes the concatenation operator. Additionally we denote four edge types, each representing a connection between two node types. Thus we have $\mathscr{A}_E = \{(\mathrm{P},\mathrm{P}), (\mathrm{NP},\mathrm{NP}), (\mathrm{NP},\mathrm{P}), (\mathrm{P},\mathrm{NP})\}$. We then compute the heterogeneous graph convolution by using the GAT-Conv [32] at a given time t.

In the following, we omit the time parameter for brevity as we predict leakages for each point in time individually.

$$\hat{\mathbf{x}}_\mathbf{i}^{(\mathbf{e})} = \sum_{j \in \mathscr{N}_e^+(i)} \alpha_{i,j}^{(e)} \mathbf{\Theta}^{(e)} \mathbf{x_j} \tag{5}$$

Here $\mathscr{N}_e^+(i)$ again denotes the neighbors of node i along edges of type $e \in \mathscr{A}_E$, potentially including the node i itself, || denotes the concatenation operator and $\mathbf{x_j}$ is the recorded feature at node j. Here $\mathbf{\Theta}^{(e)}$ is the weight matrix associated with the attention mechanism for each edge type e. Please note that, automatically generated node or edge tensors are created upon initial access and are indexed using string keys. Node types are represented by unique string identifiers, while edge types are defined using a triplet format, signifying the edge type and the two node types it connects. This design allows for varying feature dimensionalities for each type within the data object.

The attention coefficients $\alpha_{i,j}^{(e)}$ [32] are computed as

$$\alpha_{i,j}^{(e)} = \frac{\exp(\mathrm{LeakyReLU}({\mathbf{a}^{(e)}}^\top [\mathbf{\Theta}^{(e)} \mathbf{x_i} \| \mathbf{\Theta}^{(e)} \mathbf{x_j}]))}{\sum_{k \in \mathscr{N}_e^+(i)} \exp(\mathrm{LeakyReLU}({\mathbf{a}^{(e)}}^\top [\mathbf{\Theta}^{(e)} \mathbf{x_i} \| \mathbf{\Theta}^{(e)} \mathbf{x_k}]))} \tag{6}$$

where $\mathbf{\Theta}^{(e)}$ and $\mathbf{a}^{(e)}$ are learnable parameters. In the next step we aggregate the features from each edge type in the following way:

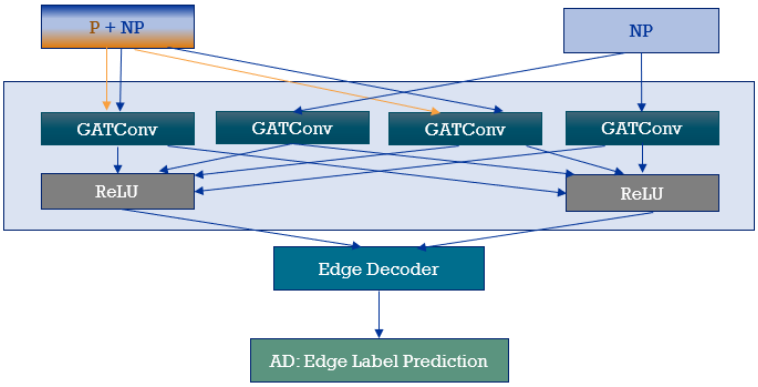

Figure 3: Liquor-HGNN Network Architecture for one layer. Signals from the two types of nodes $\mathscr{A}_V = \{\mathrm{P}, \mathrm{NP}\}$ are processed separately through four different edge types $\mathscr{A}_E = \{(\mathrm{P},\mathrm{P}), (\mathrm{NP},\mathrm{NP}), (\mathrm{NP},\mathrm{P}), (\mathrm{P},\mathrm{NP})\}$. Features from different nodes are merged during the aggregation step of the message passing.

$$\hat{\mathbf{x}}_\mathbf{i} = \text{ReLU}(\sum_{e \in \mathscr{A}_E} \hat{\mathbf{x}}_\mathbf{i}^{(\mathbf{e})}) \tag{7}$$

We stack consecutive heterogeneous message-passing layers (see Fig. 3) of this type in our model, using the predicted features of the previous layer $\hat{\mathbf{x}}_\mathbf{i}$ as input for the next layer. After applying the last layer, we calculate embeddings for each edge using the output node embeddings of the last message-passing layer in the graph decoder by applying a single fully-connected layer. For a given edge k connecting nodes i and j with node embeddings $\hat{\mathbf{x}}_\mathbf{i}$ and $\hat{\mathbf{x}}_\mathbf{j}$ respectively, we have:

$$\text{logits}_k = \sigma(\mathbf{W}[\hat{\mathbf{x}}_\mathbf{i} \| \hat{\mathbf{x}}_\mathbf{j}] + \mathbf{b}) \tag{8}$$

where $\mathbf{W}, \mathbf{b}$ are learnable parameter matrices.

Different threshold values are then tested over the logits to convert them into binary predictions. For each threshold value, a custom economic score is calculated based on the detections made by the model. The economic score takes into account both the true positive detections and their distances from the actual leak location. The higher the economic score, the better the model performs. In order to set the threshold for separating leaky from non-leaky edges on the validation split, we used our own implementation of the Economic Score metric[2] based on the KIOS research metric of the challenge. This KIOS Economic Score metric adopts a purely economic perspective. In this perspective, the water utility assesses its gains based on the money saved from successfully detecting water leaks. Additionally, the utility takes into account the expenses associated with dispatching a repair crew each time they need to search for a leak. A valid detection is one that identifies a link ID situated within a predefined distance from the actual leak site, and the reported start time of the leakage incident falls within the duration of the same leak. The predefined distance corresponds to the operational capability of the close-range equipment employed by the repair crew [6]. As the Economic Score metric consists of more than 1200 lines of code and two pages of formula in the original version, we refer to the source for further information.

The edge embeddings calculated by $\mathbf{W}[\hat{\mathbf{x}}_\mathbf{i} \| \hat{\mathbf{x}}_\mathbf{j}] + \mathbf{b}$ are used as input for the Sigmoid layer in eq. (8) with the binary cross entropy (BCE) loss with logits as loss function, which gives a score indicating how normal (0) or anomalous (1) an edge is. The decision if an edge k is considered leaky is then made according to a threshold τ.

$$\text{leaky}_k = \begin{cases} 0 & \text{logits}_k < \tau \\ 1 & \text{logits}_k \geq \tau \end{cases} \tag{9}$$

5. Experiments

We conducted experiments on the BATTleDIM dataset under challenge conditions in order to maintain strict comparability [6].

For the training phase, we used 80 percent of the 2018 data, while the remaining 20 percent served as the validation split to set the threshold. To evaluate the final model performance, the complete 2019 dataset was held back as an independent test set.

[2] https://github.com/MilanShao/Liquor-HGNN/blob/main/models/liquor_gnn.py

Table 1
Results on the BattleDIM dataset

Team Name	True Positive Rate (%)	False Positives	Economic Score
Liquor-HGNN	47.83	46	**271,584**
Tongji-Team	56.52	3	264,873
Under Pressure	65.22	4	260,562
IRI	43.47	1	210,772
Leakbusters	47.83	7	195,490
Tsinghua	47.83	5	167,981
UNIFE	43.47	4	127,626

To ensure the Liquor-HGNN model's optimal configuration, we conducted hyperparameter optimization (HPO). The selected hyperparameters for Liquor-HGNN include ten layers, 50 epochs, 64 hidden channels, a tuned learning rate of 0.00109, and a batch size of 512. We have been testing different convolutions in preliminary experiments including GATConv, SAGEConv, GraphConv and LEConv. The best results have been gained by GATConv. Therefore we use it in the results section.

6. Results

The following section presents the results of benchmarking experiments conducted on the BattleDIM dataset, comparing against other submissions in the BattleDIM 2020 challenge. Table 1 ranks the participants based on their Economic Score [6].

The best economic score was reached by Liquor-HGNN with a Score of 271,584. Following closely behind was the first place team of the challenge, with an Economic Score of 264,873. In summary, the benchmarking results showcase the effectiveness of Liquor-HGNN.

7. Discussion

The BattleDIM Challenge 2020 introduced a metric [6] to evaluate leakage detection algorithms in water supply networks. This metric considers spatial and temporal characteristics of predicted pipe leakages, aiming to capture their economic impact. However in our experiments, optimizing for this metric seems to introduce unintended and non-intuitive artefacts as for example a model with lower TPR and higher FPs can nevertheless result in a better economic score. The stringent penalties for undetected or late-identified leakages may inadvertently increase false predictions without proportional consequences. This can impose additional costs on water suppliers. Moreover, the metric only evaluates the initial prediction and does not account for potential improvements over subsequent time steps. This is further distorted by the hierarchical order of evaluation, in which the first detected time-step is taken more into account than the ones that follow in chronological order. The optimization of the model towards a high economic score comes along with an increasing number of false values, which are not ranked high for the

penalty calculation. When we use different metrics like F1, Precision, Recall, Balanced Accuracy or AUC-ROC to set the threshold on the validation split the model is optimized towards a higher True Positive rate and focuses more on eliminating false predictions. The highest True Positive Rate we gained was 82,61 percent. If we manually tune the threshold on the validation set, we have been able to gain an Economic Score of 326,521 as highest value but at the cost of 138 False Positives. This means, that if the model is optimized to predict as much as possible as true value the Economic Score rises, but the False values also go up. Nevertheless, the metric's integration of spatial and temporal failures is commendable, catering to the specific needs of water distribution networks. To ensure fair evaluations, future enhancements should strike a balanced approach, considering both false predictions and missed leakages while incorporating iterative performance improvements. By addressing these concerns, the metric could be further refined to effectively capture the real-world challenges faced by leakage detection algorithms.

8. Conclusion

We have gained the highest Economic Score for the Liquor-HGNN approach in comparison to the other challenge submissions. We demonstrated the effectiveness of the localization of leakages in sparsely measured water distribution networks. We further integrated Liquor-HGNN with a demand prediction pre-processing model that leverages the underlying physical information of the hydraulic system. This combination enabled us to overcome the challenges posed by sparse data in complex sensor distribution networks.

References

[1] R. Dighade, M. Kadu, A. Pande, Challenges in water loss management of water distribution systems in developing countries, International Journal of Innovative Research in Science, Engineering and Technology 3 (2014) 13838–13846.

[2] C. Montoya-Pachongo, I. Douterelo, C. Noakes, M. A. Camargo-Valero, A. Sleigh, J.-C. Escobar-Rivera, P. Torres-Lozada, Field assessment of bacterial communities and total trihalomethanes: Implications for drinking water networks, Science of the Total Environment 616 (2018) 345–354.

[3] R. Frauendorfer, R. Liemberger, The issues and challenges of reducing non-revenue water (2010).

[4] N. A. M. Yussof, H. W. Ho, Review of water leak detection methods in smart building applications, Buildings 12 (2022) 1535.

[5] C. Hu, M. Li, D. Zeng, S. Guo, A survey on sensor placement for contamination detection in water distribution systems, Wireless Networks 24 (2018) 647–661.

[6] S. G. Vrachimis, D. G. Eliades, R. Taormina, A. Ostfeld, Z. Kapelan, S. Liu, M. Kyriakou, P. Pavlou, M. Qiu, M. M. Polycarpou, Battledim: Battle of the leakage detection and isolation methods, in: Proc., 2nd Int. CCWI/WDSA Joint Conf, 2020.

[7] K. A. Klise, R. Murray, T. Haxton, An overview of the water network tool for resilience (wntr). (2018).

[8] H. E. Mutikanga, S. K. Sharma, K. Vairavamoorthy, Methods and tools for managing losses

in water distribution systems, Journal of Water Resources Planning and Management 139 (2013) 166–174.

[9] A. TehraniJamsaz, H. Chen, A. Jannesari, Graphbinmatch: Graph-based similarity learning for cross-language binary and source code matching, arXiv preprint arXiv:2304.04658 (2023).

[10] T. Cheng, Y. Li, F. Harrou, Y. Sun, J. Gao, T. Leiknes, A Hybrid Leakage Detection and Isolation Approach Based on Ensemble Multivariate Changepoint Detection Methods, 2020. URL: https://doi.org/10.5281/zenodo.3964167. doi:10.5281/zenodo.3964167, This publication is based upon work supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No: OSR-2019-CRG7-3800.

[11] L. Huang, K. Du, M. Guan, Q. Wang, The combined usage of the hydraulic model calibration residual and an improved vectorial angle method for solving the BattLeDIM problem, 2020. URL: https://doi.org/10.5281/zenodo.3925507. doi:10.5281/zenodo.3925507.

[12] I. Daniel, J. Pesantez, S. Letzgus, M. A. Khaksar Fasaee, F. Alghamdi, K. Mahinthakumar, E. Berglund, A. Cominola, A high-resolution pressure-driven method for leakage identification and localization in water distribution networks, 2020. URL: https://doi.org/10.5281/zenodo.3924632. doi:10.5281/zenodo.3924632.

[13] J. Saldarriaga, L. Solarte, C. Salcedo, C. Montes, L. Martinez, M. Gozalez, M. Cuello, A. Ariza, C. Galindo, N. Ortiz, C. Gomez, S. Vanegas, Battle of the Leakage Detection and Isolation Methods: An Energy Method Analysis using Genetic Algorithms, 2020. URL: https://doi.org/10.5281/zenodo.3923227. doi:10.5281/zenodo.3923227.

[14] X. Wang, J. Li, X. Yu, Z. Ma, Y. Huang, A multistage approach to detect and isolate multiple leakages in district metering areas in water distribution systems, 2020. URL: https://doi.org/10.5281/zenodo.3924109. doi:10.5281/zenodo.3924109.

[15] D. B. Steffelbauer, J. Deuerlein, D. Gilbert, O. Piller, E. Abraham, A dual model for leak detection and localization, 2020. URL: https://doi.org/10.5281/zenodo.3923907. doi:10.5281/zenodo.3923907.

[16] W. Zhang, J. Liu, L. Han, Y. Li, X. Li, Z. Shi, J. Yu, J. Wang, A Real Time Method to Detect the Burst Location of Urban Water Supply Network, 2020. URL: https://doi.org/10.5281/zenodo.3923929. doi:10.5281/zenodo.3923929.

[17] L. Romero, J. Blesa, D. Alves, G. Cembrano, V. Puig, E. Duviella, Leak Localization In Water Distribution Networks Using Data-Driven And Model-Based Approaches, 2020. URL: https://doi.org/10.5281/zenodo.3923501. doi:10.5281/zenodo.3923501.

[18] K. W. Min, T. Kim, Y. H. Choi, D. Jung, J. H. Kim, A Two-Phase Model to Detect and Localize Water Distribution System Leakages, 2020. URL: https://doi.org/10.5281/zenodo.3922019. doi:10.5281/zenodo.3922019.

[19] C. Blocher, F. Pecci, C. Jara Arriagada, I. Stoianov, Detecting and Localizing Leakage Hotspots in Water Distribution Networks via Regularization of an Inverse Problem: An Application to the Battle of Leakage Detection and Isolation Methods 2020 Competition, 2020. URL: https://doi.org/10.5281/zenodo.3921800. doi:10.5281/zenodo.3921800, the project was co-funded by Cla-Val UK Ltd.

[20] R. Liu, Z. Zhang, D. Zhang, Leakage Detection and Isolation in Water Distribution Network Based on Data Mining and Genetic Optimized Hydraulic Simulation, 2020. URL: https:

//doi.org/10.5281/zenodo.3911523. doi:10.5281/zenodo.3911523.
[21] D. Adanza Dopazo, A leakage detection system extracting the most meaningful features with decision trees., 2020. URL: https://doi.org/10.5281/zenodo.3906668. doi:10.5281/zenodo.3906668.
[22] C. A. Tan, V. Phipatanasuphorn, C. H. A. Lai, Deep Learning of Complex Pipe Leakages Events in Drinking Water Distribution Networks for Effective Spatiotemporal Pre-Detections and Isolations of Leak Conditions, 2020. URL: https://doi.org/10.5281/zenodo.3902945. doi:10.5281/zenodo.3902945.
[23] Z. Li, K. Xin, Fast Localization of Multiple Leaks in Water Distribution Network Jointly Driven By Simulation and Machine Learning, 2020. URL: https://doi.org/10.5281/zenodo.3911045. doi:10.5281/zenodo.3911045.
[24] Z. Y. Wu, Y. He, Decomposition-based data analysis with hydraulic model calibration for leakage detection and isolation, 2020. URL: https://doi.org/10.5281/zenodo.3908525. doi:10.5281/zenodo.3908525.
[25] S. Bhomwick, K. Seifert, Water Leakage Detection and Localization: Anomaly Matrix - A Deterministic Approach, 2020. URL: https://doi.org/10.5281/zenodo.3906850. doi:10.5281/zenodo.3906850.
[26] I. Marzola, F. Mazzoni, S. Alvisi, M. Franchini, A Pragmatic Approach for Leakage Detection Based on the Analysis of Observed Data and Hydraulic Simulations, 2020. URL: https://doi.org/10.5281/zenodo.3900989. doi:10.5281/zenodo.3900989.
[27] Brentan, Fluing abstract, 2020. URL: https://doi.org/10.5281/zenodo.4011894. doi:10.5281/zenodo.4011894.
[28] G. Örn Garðarsson, F. Boem, L. Toni, Graph-based learning for leak detection and localisation in water distribution networks*, IFAC-PapersOnLine 55 (2022) 661–666. URL: https://www.sciencedirect.com/science/article/pii/S2405896322005882. doi:https://doi.org/10.1016/j.ifacol.2022.07.203, 11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2022.
[29] C. Zhang, D. Song, C. Huang, A. Swami, N. V. Chawla, Heterogeneous graph neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 793–803. URL: https://doi.org/10.1145/3292500.3330961. doi:10.1145/3292500.3330961.
[30] Y. Wang, Z. Duan, Y. Huang, H. Xu, J. Feng, A. Ren, Mthetgnn: A heterogeneous graph embedding framework for multivariate time series forecasting, 2020. URL: https://arxiv.org/abs/2008.08617. doi:10.48550/ARXIV.2008.08617.
[31] O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, R. Rajagopal, Neuralprophet: Explainable forecasting at scale, 2021. URL: https://arxiv.org/abs/2111.15397. doi:10.48550/ARXIV.2111.15397.
[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, 2017. URL: https://arxiv.org/abs/1710.10903. doi:10.48550/ARXIV.1710.10903.
[33] S. Menon, Piping Calculations Manual, McGraw Hill LLC, 2004. URL: https://books.google.de/books?id=jC5PltQbLjwC.

9. Appendix

9.1. Edge Weight Assignment

In the case of the used LEConv, edge-weights can be assigned to the edges. But unfortunately, this has not been possible for heterogeneous graphs, as they have different edge-types and dimensions. We therefore use the generic HeteroConv wrapper [9] to perform the message passing from node $\mathbf{x_i}(t)$ to node $\mathbf{x_j}(t)$ for the different edge types. For the binary weight assignment, a convention for an unweighted graph is adopted where the adjacency matrix A of two nodes v and u A_{vu} equals 1 if the edge e_{vu} exists in the graph, and 0 otherwise.

To assign weight values based on hydraulic loss, various equations are employed. The pipe-length is defined as l_{vu} and the diameter of the pipe is defined as d_{vu} as well as the slope of the pipe as s_{vu}.

For a hydraulic loss weighted graph, the edge weight is determined using different equations. Thus, 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. For example if we denote the Hazen-Williams equation, we have:

$$headloss_{hw} = 10.67 \left(\frac{\text{velocity}}{C} \right)^{1.852} \frac{\mathrm{l}_{vu}}{\mathrm{d}_{vu}{}^{4.87}} \tag{10}$$

where the velocity is taken from the WNTR tool [7]. The same applies for the following equations.

Darcy-Weisbach equation:

$$headloss_{dw} = \text{friction factor} \frac{l_{vu}}{d_{vu}} \frac{\text{velocity}^2}{2 \cdot 9.81} \tag{11}$$

Prony equation:

$$headloss_{prony} = A \cdot \text{velocity}^2 + B \cdot \text{velocity} \tag{12}$$

Manning's equation:

$$headloss_{man} = \frac{1}{n^2} \left(\frac{\text{velocity}}{r \sqrt{\mathrm{s}_{vu}}} \right)^{\frac{2}{3}} \tag{13}$$

Hagen-Poiseuille equation:

$$headloss_{hp} = \frac{128 \cdot \text{viscosity} \cdot \mathrm{l}_{vu} \cdot \text{velocity}}{\pi \cdot \mathrm{d}_{vu}^4} \tag{14}$$

Energy equation:

$$headloss_{E} = \text{elevation difference} + \text{friction headloss} \tag{15}$$

The friction factor in the Darcy-Weisbach equation is calculated iteratively based on the Colebrook-White equation. The initial friction factor is set to 0.2083 as a starting point in literature [33], because it is close to the friction factor value for a smooth pipe. According to

the Economic Score metric the best headloss equation for edge-weight assignment is taken automatically. The headloss value for each link is then stored in the equivalent tensor of each equation. The edge weights for each relation according to the edge types are then computed as the exponential function of the negative of the headloss values.

The edge-weights associated with the best performing hydraulic head-loss equation for the four edge types at layer l are then denoted as weights $\mathscr{W}_{\mathbf{x_i}(t),connects,\mathbf{x_j}(t)}$, where *connects* denotes the relation and the node features $\mathbf{x_i}(t)$ and $\mathbf{x_j}(t)$ as well as GATConv are introduced earlier. In the edge-weight dictionary the edge-weights are stored according to the edge-types. These edge-weights are then used during message passing in the HeteroConv layers, where the LEConv layers are applied to compute the updated node features.

9.2. Acknowledgement

We thank FlowChief GmbH for financing this research project.

9.3. BattleDIM Water Network Visualization

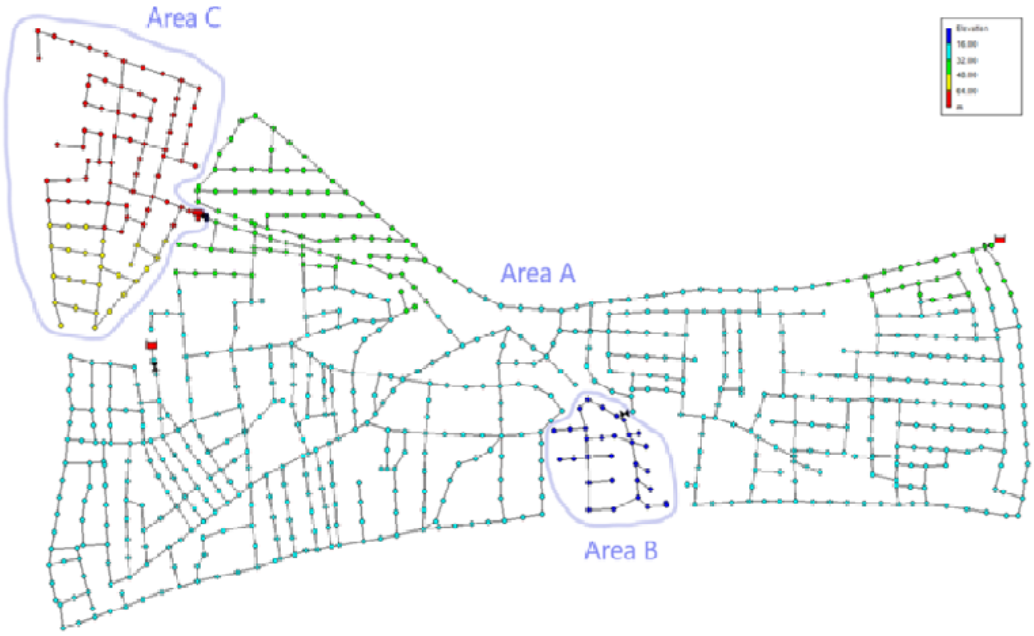

Figure 4: BattleDIM Challenge Water Distribution network with demand measurements in Area C, a valve layer without measurements in Area B and sparsely distributed pressure measurements in Area A, colours represent the elevation of the edges [6].

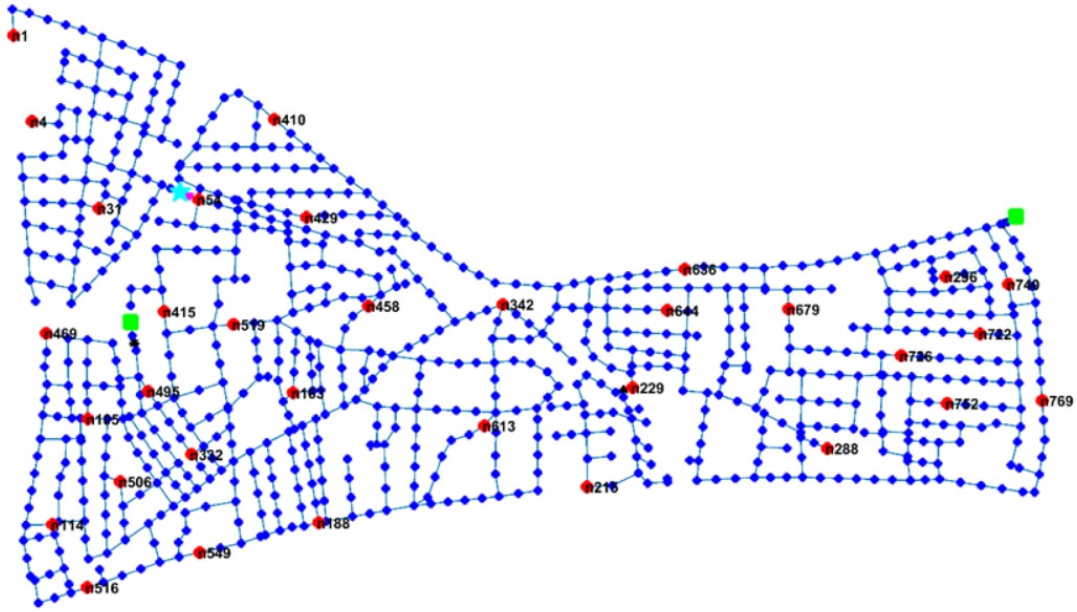

Figure 5: BattleDIM Challenge distribution of nodes with pressure measurement (=red nodes) versus nodes without pressure measurement (=blue nodes) [6].

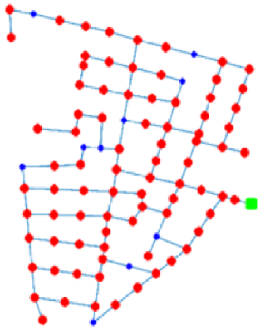

Figure 6: BattleDIM Challenge distribution of nodes in Area C with demand measurement (=red nodes) versus nodes without demand measurement (=blue nodes) [6].