Graph Convolutional Networks Based Contamination Source Identification Across Water Distribution Networks

Yujue Zhou[a,b], Jie Jiang[b,*], Kai Qian[b], Yulong Ding[b], Shuang-Hua Yang[b,c,*] and Ligang He[a]

[a]Department of Computer Science, University of Warwick, Coventry, CV4 7AL, UK
[b]Department of Computer Science and Engineering, Southern University of Science and Technology, 1088 Xueyuan Avenue, Shenzhen 518055, P.R. China
[c]Peng Cheng Laboratory, No.2, Xingke 1st Street, Shenzhen 518066, P.R. China

ARTICLE INFO

Keywords:
water distribution networks, contamination source identification, graph convolutional network, cross-networks learning

ABSTRACT

Water distribution Networks (WDNs) are one of the most important infrastructures for modern society. Due to accidental or malicious reasons, water contamination incidents have been repeatedly reported all over the world, which not only disrupt the water supply but also endanger public health. To ensure the safety of WDNs, water quality sensors are deployed across the WDNs for real-time contamination detection and source identification. In the literature, various methods have been employed to improve the performance of contamination source identification (CSI) and recent studies show that there is a great potential to tackle the CSI problem by deep learning models. The success of deep learning based CSI methods often requires a large size of training samples being collected. In real-world situations, the number of contamination events occurring in a single WDN is rather small, especially for a newly built WDN. However, the existing CSI methods in the literature mostly focus on the study of training and applying models on the same WDNs and the knowledge of CSI gained from one WDN cannot be reused by a different WDN. To these ends, based on the application of graph convolutional networks, this paper provides a solution for cross-network CSI that can transfer the CSI knowledge learned from one WDN to a different WDN. Empirically, based on a benchmark WDN in the task of contamination source identification, we show that the proposed cross-network CSI method can achieve comparable accuracy even trained on a different WDN.

1. Introduction

Water distribution networks (WDNs) are one of the most essential infrastructures for modern society. Due to accidental [1] or malicious reasons [28], water contamination incidents have been repeatedly reported all over the world, which not only disrupt the water supply but also endanger the public health. With the development of sensor technologies, water quality sensors are often deployed at the critical points of WDNs to collect real-time data regarding water quality. These data are used not only for water quality monitoring but more importantly for identifying the source of contamination in case of contamination being detected. The problem of contamination source identification (CSI) is difficult due to a number of reasons [19]. First, it is infeasible to do experimental analysis in real-world WDNs. Second, the water quality data are often sparse as the deployment of water quality sensors is limited in WDNs. Third, the dynamics of water demand patterns add uncertainty to the analysis. Fourth, it is important that the contamination source can be identified within a short period of time.

In the literature, simulation tools are often adopted as an alternative to real-world experimental analysis for water pollution analysis [6] and contamination source identification [18]. For example, the hydraulic modelling and simulation tool EPANET [20] developed by the US environmental protection agency has received great popularity for tackling CSI problems. A WDN is modelled by setting up the topology, hydraulic and water quality parameters, which can then be used to run simulations to observe the hydraulic and water quality changes. Combing with such simulation tools, researchers proposed to use evolutionary algorithms (EAs) to explore the contamination configurations by simulating a large number of contamination events and searching for the one that generates the most similar water quality data as to the real contamination event [18, 23, 29]. Though such simulation-optimisation based CSI methods show great potential, they are facing big challenges as they highly rely on the modelling capacity of the simulation tools and they come with a high computational cost due to the optimisation components.

Recently, deep neural networks have been investigated for tackling CSI problems and it is shown that deep neural networks achieved promising performance in identifying the contamination source node [19]. However, both the existing simulation-optimisation and the deep learning based CSI methods focus on the CSI of a single WDN. That is, the knowledge of identifying the contamination source gained from one WDN cannot be reused by a different WDN. Moreover, in real-world situations, the number of contamination events recorded in a single WDN is rather small especially for a newly built WDN. Therefore, for a new WDN or a new area of a large WDN, little contamination information is available for model training.

*This research is supported by the National Natural Science Foundation of China under Grant Nos. 61873119 and 92067109, National Key R&D Program of China under Grant Nos. 2019YFC0810705, the Science and Technology Innovation Commission of Shenzhen under Grant Nos. KQJSCX20180322151418232 and the Department of Education of Guangdong Province under Grant Nos. 2019KQNCX132.

*Corresponding author (E-mail address: jiangj@sustech.edu.cn; yangsh@sustech.edu.cn
ORCID(s):

To these ends, this paper proposed a method called GrassL (**Gra**ph Based Cro**ss**-Network **L**earning for Contamination Source Identification). GrassL investigates the application of graph convolutional networks (GCNs) to transfer the CSI knowledge learned from the historical contamination events (water quality sensor data together with the real contamination source node) that occurred in one WDN to a different WDN. The intuition behind such cross-WDN learning methods is that different WDNs or different areas of a big WDN share features from a local perspective. For example, the nodes of different WDNs all have upstream or downstream relations and the flow of the contaminants can be reflected from the changes of the data collected by the water quality sensors deployed over the WDNs. To extract such local features across WDNs, we investigate various deep learning methods to capture both the spatial and the temporal information of WDNs. The spatial information comes from the topology of a WDN while the temporal information comes from the data collected from the water quality sensors. By a set of experiments with a benchmark WDN, we provide an evaluation of the performance of different cross-network CSI methods.

In summary, the contributions of this paper are as follows:

- We propose a GCN based CSI method GrassL that can transfer the CSI knowledge learned from the historical contamination events collected from one WDN to a different WDN.
- We provide an extensive evaluation of GrassL with five baseline methods for the task of cross-network CSI through a set of experiments.
- The evaluation is performed using a benchmark WDN with 1786 nodes. We release the datasets used in this paper as well as the source code of GrassL [1] to facilitate the reproducibility of our work.

The rest of the paper is organised as follows. Section 2 discusses the related work. Section 3 provides a formalisation of the CSI problem. Section 4 introduces the proposed method GrassL. Section 5 presents the dataset used in this paper. Section 6 illustrates the experiment preparation and the evaluation metrics, and analyses the experiment results. Section 7 discusses the advantages and the limitations of the proposed CSI method. Finally, in Section 8, we conclude the paper with possibilities of future work.

2. Related Work

In the literature, a vast amount of research has been devoted to addressing the CSI problem based on the data collected from the water quality sensors deployed in WDNs. In general, the proposed CSI methods can be grouped in four categories.

The first category of CSI methods regards the CSI problem as an inverse problem which is solved by analysing the inner hydraulic model and water quality model of the WDN directly. For example, Shang et al. proposed a particle backtracking algorithm which aims to describe the relationship between water quality at input and output locations and by the relationship the inverse problem can be solved [21]. Laird at al. presented an origin tracking algorithm for solving the inverse problem of contamination source identification based on a nonlinear programming framework [10].

The second category of CSI methods is based on probabilistic modelling. For example, Huang and McBean proposed to combine a screening approach and a maximum likelihood method to identify the location and time of a contamination event based on limited sensor data [8]. Perelman and Ostfeld proposed a solution based on topological clustering and Bayesian Networks (BNs) [17]. Wang and Harrison proposed to use Markov Chain Monte Carlo (MCMC) to enable probabilistic inference of contamination events [24].

The third category of CSI methods adopts a simulation-optimisation approach, which is based on the idea that the accumulated errors between the contamination data observed by the water quality sensors and the contamination data returned by the simulation should be minimum if the simulated contamination event is the same as the real one. A common approach is combining the simulation tool EPANET with variants of EAs. For example, Preis and Ostfeld integrated a genetic algorithm (GA) with EPANET, in which GA provides heuristic information for searching while EPANET is used to evaluate the fitness of a candidate solution as to the real event [18]. Praveen et al. conducted experiments to show the effectiveness of simulation-optimisation methods by taking dynamic water demand into account [23]. Liu et al. developed an EA based adaptive dynamic optimisation procedure that uses multiple population-based search to provide a real-time response to contamination events [14]. Yan et al. investigated different optimization algorithms for CSI such as cultural algorithms [30] and variants of genetic algorithms [31, 32, 29].

The fourth category of CSI methods is motivated by the recent success of deep learning methods in domains such as computer vision and audio processing. For example, Qian et al. proposed a deep learning guided evolutionary algorithm for CSI, in which a convolutional neural network (CNN) and a recurrent neural network (RNN) is trained for identifying the contamination source node [19]. Sun et al. proposed to use CNN to capture the user complaint patterns corresponding to different contamination intrusion nodes and improve the performance of CSI based on consumer complaint information [22].

Though the CSI methods mentioned above provided solutions to the CSI problem from different perspectives, they mostly focus on the CSI of a single WDN. That is, the knowledge of identifying the source of the contamination events gained from the study of one WDN cannot be reused by a different WDN. Such limitation prohibits the application of data-driven CSI methods in real-world situations as the number of contamination events occurred in a single WDN is often rather small especially for a newly built WDN.

[1] https://github.com/yujue-zhou/Graph_CSI_WDS

To the best of our knowledge, there is no prior research that has investigated the application of Graph Convolutional Networks to transfer the CSI knowledge learned from the contamination events (water quality sensor data together with the real contamination source node) collected from one WDN to a different WDN.

Another line of related work focuses on the transfer of temporal and spatial knowledge in various prediction tasks. For example, with an aim of air quality prediction, Wei et al. leverages sparse coding and graph clustering to transfer the knowledge from a city with enough data and labels to the city where services and infrastructures are just established [27]. For the purpose of cross-city crowd flow prediction, Wang et al. used auxiliary data to obtain regional similarity between cities, and by building a deep neural network with convolutional LSTM layers to capture both spatial and temporal dependencies within the data [25]. Yao et al. proposed a meta-learning method for spatial-temporal prediction, in which CNN and LSTM are combined to capture the spatial dependencies and temporal correlation [33]. Mallick et al. used Diffusion Convolutional Recurrent Neural Network (DCRNN) to achieve cross-regions traffic prediction [15]. The success of these works show great potential of using deep neural networks to capture the temporal and spatial features for transfer learning, especially the recently developed graph convolution networks. With a similar requirement of transfer learning, this paper carries out an extensive study of the application of variants of graph based neural network models for the task of cross-network CSI, which has not been investigated before in the literature.

From a safety perspective, there is a body of relevant research that focuses on fault detection and diagnosis for pipeline networks. Pipeline faults like leakage, blockage and contamination not only bring financial losses but more seriously may lead to safety hazards. The accurate detection of the size and location of such faults is essential for the smooth operation of industries and environmental safety [5]. For example, to identify the leakage location and size in hydrocarbon transporting pipelines, Zadkarami et al. [34] proposed to use statistical techniques and wavelet-based approaches to extract features from the inlet pressure and outlet flow signals, which are then individually fed into two multilayer perceptron neural network classifiers to determine the leakage state. Moreover, the Dempster-Shafer technique is applied to integrate the outputs of the two classifiers. For leakage recognition in a gas pipeline valve, Li et al. proposed to use the acoustic signal obtained from an acoustic emission sensor as the feature source, and then apply kernel principal component analysis to reduce the dimensionality of the features and support vector machine to recognise the leakage level [12]. Also for gas pipelines, Wang et al. proposed a Bayesian network-based method for analysing the failure probability of urban buried gas pipelines [26]. To localise leakage in assembled pressurized liquid pipelines, Chen et al. proposed a modified method for calculating negative pressure wave propagation velocity considering the influence of rubber washers [3]. To detect quality faults in water distribution systems, Perelman et al. proposed a framework that combines an artificial neural network model for temporary analysis of multivariate water quality time series with Bayesian sequential analysis for estimating the probability of an event [16]. Based on various statistical and machine learning methods, these studies provide different solutions for detecting and diagnosing faults that may occur in pipeline networks such as gas pipelines and water distribution networks. Similarly, in this paper, we are aiming at providing an approach for diagnosing the quality faults in water distribution networks, i.e., identifying the contamination source when a contamination event occurs, which is essential for environmental protection as well as for the process safety of the water treatment plants. In particular, we are interested in the task of cross-network contamination source identification and the proposed method may be adapted to fault diagnosis in other types of pipeline networks.

3. Problem Statement

Given a WDN $\mathcal{W}$ with a set of historical contamination events (pairs of sensor data and the corresponding contamination source node), we regard $\mathcal{W}$ as the source domain. Let $G(V, A)$ represents the network topology (nodal connectivity) corresponding to $\mathcal{W}$, in which V indicates the nodes in $\mathcal{W}$, $N = |V|$ indicates the number of the nodes and A is the adjacency matrix representing the nodal connectivity, i.e., whether pairs of nodes are adjacent or not in $\mathcal{W}$. Assuming there are B historical contamination events, $X = \{X_1, X_2, \dots, X_B\}$ represents the sensor data of the B contamination events, and their corresponding contamination source nodes are indicated as $y = \{y_1, y_2, \dots, y_B\}$. For each contamination event, $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,T}\} \in M \times T, 1 \leq i \leq B$, where M indicates the number of sensors deployed in $\mathcal{W}$ and T indicates the number of time steps the sensor data are collected.

Suppose there is another WDN $\tilde{\mathcal{W}}$ for which no historical contamination event is available. We regard $\tilde{\mathcal{W}}$ as the target domain. Let $G(\tilde{V}, \tilde{A})$ represent the network topology of $\tilde{\mathcal{W}}$. Similarly, $\tilde{V}$ represents the nodes and $\tilde{A}$ is the adjacency matrix of $\tilde{\mathcal{W}}$. Assuming the set of contamination events for which the contamination source needs to be identified is represented as $\tilde{B}$, the corresponding sensor data is $\tilde{X} = \{\tilde{X}_1, \tilde{X}_2, \dots, \tilde{X}_{\tilde{B}}\}$, and the corresponding contamination source node is $\tilde{y} = \{\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_{\tilde{B}}\}$. For each contamination event there is $\tilde{X}_i = \{\tilde{x}_{i,1}, \tilde{x}_{i,2}, \dots, \tilde{x}_{i,T}\} \in \tilde{M} \times T, 1 \leq i \leq \tilde{B}$, where $\tilde{M}$ indicates the number of sensors deployed in $\tilde{\mathcal{W}}$ and T indicates the number of time steps the sensor data are collected.

The purpose of this research is to use the historical contamination event data (pairs of X and y) and the topological information (A) from the source domain to train a model h so that the model can accurately identify the source of a contamination event occurred in the target domain. That is, we want to minimize the following objective function:

$$\sum_{1 \leq i \leq \tilde{B}} \mathcal{L}\left(\mathrm{h}\left(\tilde{X}_i; \tilde{\mathrm{A}}\right), \tilde{y}_i\right) \tag{1}$$

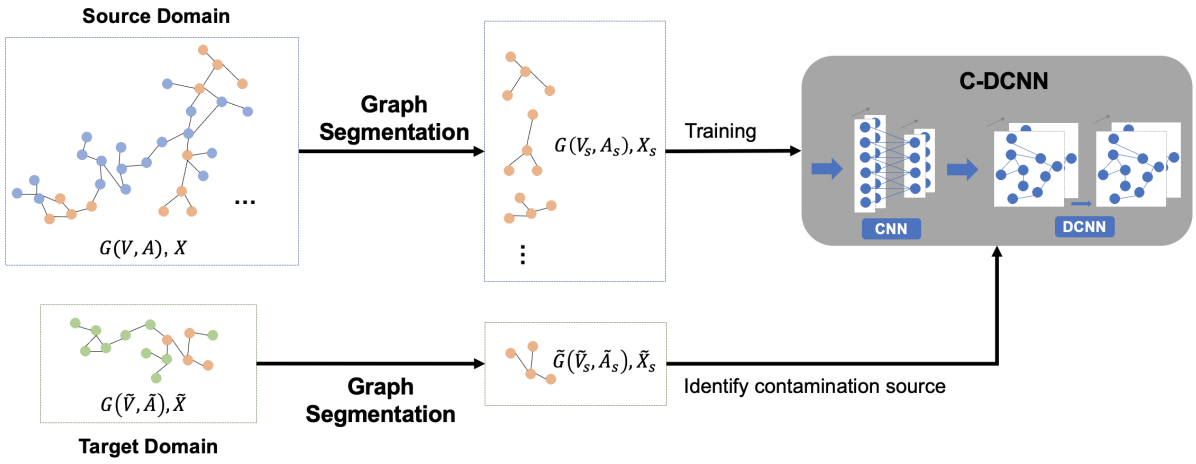

Figure 1: The framework of the Proposed Method GrassL

where h $\left(\tilde{X}_i; \tilde{\mathrm{A}}\right)$ is the predicted contamination source node, $\tilde{y}_i$ is the ground truth, and $\mathcal{L}$ indicates the cross entropy loss.

4. Proposed Method

In this section, we describe the proposed method GrassL for contamination source identification across water distribution networks. The framework of GrassL is shown in Figure 1, consisting of two steps. The first step is called Graph Segmentation, which splits a large WDN into sub-networks according to the occurrences of contamination events and their scopes of influence with respect to the network topology of the WDN. The second step is called Cross-Graph Learning, which aims at training a GCN based CSI model to transfer the CSI knowledge learned from the source domain to the target domain.

4.1. Graph Segmentation

In the task of cross-network CSI, the WDN of the source domain and that of the target domain may differ in terms of the size (number of nodes and pipes) and the network topology (nodal connectivity). This means that the CSI model trained directly on the entire WDN of the source domain cannot be transferred to the target domain. In order to achieve cross-network CSI, we train a model to capture the local features of the WDN from the source domain and expect that the network in the target domain share features from a local perspective. Specifically, we will use a variety of sub-networks extracted from the WDN and the sensor data contained in those sub-networks to train a CSI model so that the model can capture the temporal features from the sensor data together with the spatial features from the topology of the sub-networks. In this way, the trained CSI model can be used to identify the source of the contamination events in a new WDN.

To obtain the temporal and spatial features of the sub-networks, we carry out graph segmentation on both WDNs from the source and the target domains, respectively indicated as $\mathcal{W}$ and $\tilde{\mathcal{W}}$. For each contamination event occurred in $\mathcal{W}$, we extract a sub-network $\mathcal{W}_s$ and make sure the contamination source node presents in the sub-network. We then randomly select M_s nodes from $\mathcal{W}_s$ as the sensor nodes. Specifically, for the ith contamination event that occurred in $\mathcal{W}$, the extraction of a sub-network $\mathcal{W}_s$ starts from the contamination source node y_i based on the network topology of $\mathcal{W}$, i.e. $G(V, A)$. The breadth-first search is used to find the connected nodes with a probability p to keep the nodes. As a result, a sub-network $G(V_s, A_s)$ is obtained, where $V_s \subseteq V$ represents the nodes in the sub-network and $A_s = A(V_s; V_s)$ indicates the corresponding sub-adjacency matrix. The sensor data X_s of $\mathcal{W}_s$ is obtained from X of $\mathcal{W}$, and the sub-matrix $X_s = X(U_s; .) \in M_s \times T$ where $U_s \subseteq V_s$ is the set of sensor nodes in $\mathcal{W}_s$, $M_s = |U_s|$ is the number of sensors in the sub-network, T indicates the number of time steps the sensor data are collected. Similarly, for each contamination event in $\tilde{\mathcal{W}}$, we also use graph segmentation to extract the corresponding local topological information $G(\tilde{V}_s, \tilde{A}_s)$ and local sensor information $\tilde{X}_s \in \tilde{M}_s \times T$.

In addition, to facilitate the CSI model training, padding operations are performed on the local adjacency matrix and local sensor data such that the shapes of the inputs are the same. Specifically, for all the sub-networks extracted from $\mathcal{W}$ and $\tilde{\mathcal{W}}$, we first find the largest sub-network and represent the number of nodes it contains as $\mathcal{N}$. In this paper $\mathcal{N}$ is 100, which will be illustrated in Section 5. Then, for the rest of the sub-networks, we pad rows and columns filling with zeros to their adjacency matrix such that the shape of their adjacency matrix becomes $\mathcal{N} \times \mathcal{N}$. As a result, for all the sub-networks of $\mathcal{W}$ and $\tilde{\mathcal{W}}$, we obtain an extended adjacency matrix represented as $\mathcal{A}_s, \tilde{\mathcal{A}}_s \in \mathcal{N} \times \mathcal{N}$. For the padding of the local sensor data, $\mathcal{N}$ is used as the size of the first dimension (number of rows) while the number of time steps T is used as the size of the second dimension (number of columns). The rows corresponding to the non-sensor nodes are filled with zeros. As such, the resulting matrix contains both the data collected from the sensor nodes and the zeros padded for the non-sensor nodes, which will be called contamination matrix in this paper to differentiate it from the matrix that

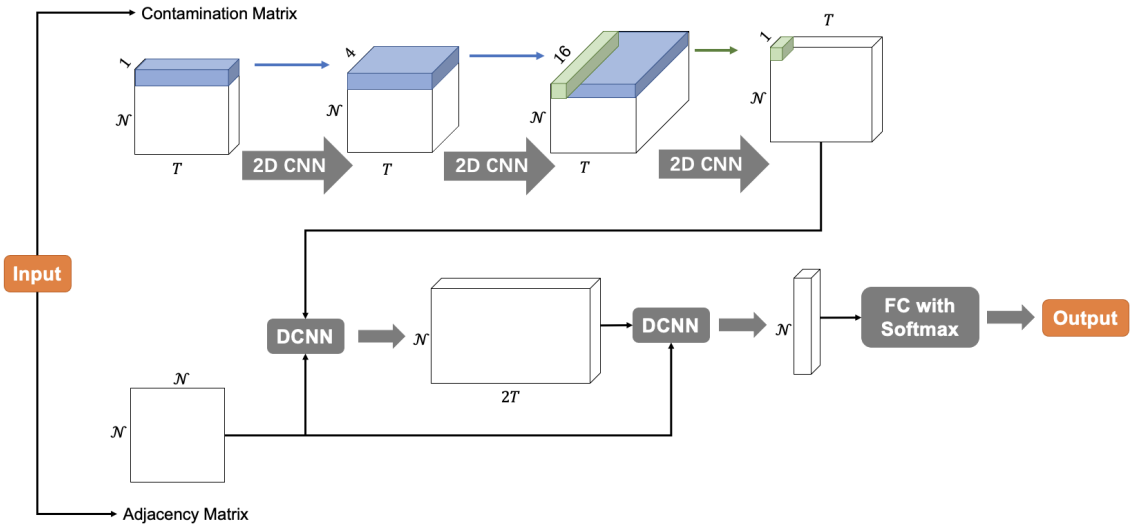

Figure 2: C-DCNN Model Structure

only consists of the sensor data. The local contamination matrix of a sub-network from $\mathcal{W}$ and $\tilde{\mathcal{W}}$ are indicated as $\mathcal{X}_s, \tilde{\mathcal{X}}_s \in \mathcal{N} \times T$.

4.2. Cross-Graph Learning

After extracting the local information with graph segmentation, we obtain the adjacency matrix $\mathcal{A}_s$ and the contamination matrix $\mathcal{X}_s$ of the source domain, and the corresponding adjacency matrix $\tilde{\mathcal{A}}_s$ and the contamination matrix $\tilde{\mathcal{X}}_s$ of the target domain. To capture the spatial-temporal features embedded in $\mathcal{A}_s$ ($\tilde{\mathcal{A}}_s$) and $\mathcal{X}_s$ ($\tilde{\mathcal{X}}_s$) for cross-network learning, we adopt a combination of Convolutional Neural Network (CNN) [11] with Diffusion Convolutional Neural Network (DCNN) [2], indicated as C-DCNN, in which CNN is used to model the temporal dynamics in the contamination matrix while DCNN is used to model the spatial features encoded by the adjacency matrix.

The model structure of C-DCNN is shown in Figure 2. C-DCNN first applies a three-layer CNN, with the contamination matrix $\mathcal{X}_s$ (or $\tilde{\mathcal{X}}_s$) as the input. The shape of $\mathcal{X}_s$ (or $\tilde{\mathcal{X}}_s$) is $\mathcal{N} \times T \times 1$, where $\mathcal{N}$ is the number of nodes in the sub-networks, T is the number of time steps the sensor data are collected. The filter sizes of the three CNN layers are $1 \times 3, 1 \times 3, 1 \times 1$. The number of filters are respectively 4, 16 and 1. Accordingly, convolution operations are only applied to the time dimension of the contamination matrix for each node. In this way, the temporal features of each node can be extracted while the relations between the nodes are maintained.

Thereafter, the outputs from the CNN model together with the adjacency matrix $\mathcal{A}_s$ (or $\tilde{\mathcal{A}}_s$) are fed into a two-layer DCNN to extract the spatial features of a network.

DCNN is a variant of spatial graph convolution networks. The main idea of DCNN is to use the operation of diffusion convolution to measure the probability of information spreading between any two nodes in a network through different paths. The formal definition of the DCNN layer is as follows:

$$Z = f\left(W^c \odot Q^* U\right) \tag{2}$$

where U represents the input of the DCNN layer; Q^* is the power series of Q, and Q is the degree-normalized transition matrix of the adjacency matrix $\mathcal{A}_s$ (or $\tilde{\mathcal{A}}_s$); W^c are the parameters; the $\odot$ operator represents element-wise multiplication; f represents a nonlinear activation function, and Z is the output of the DCNN layer.

In this paper, we apply two layers of DCNN in which three hops of graph diffusion is performed over U, i.e., the power series $Q^*=[Q, Q^2, Q^3]^T$. As such, the probability of information spreading between the neighbouring nodes within three hops are considered. As for f, the nonlinear activation function ReLu is used in this paper.

Finally, a fully connected (FC) layer followed by Softmax is used to obtain the outputs of the C-DCNN model, i.e., the probability of each node in $\mathcal{W}_s$ or $\tilde{\mathcal{W}}_s$ being the contamination source node. The node with the highest probability will be output as the final prediction.

5. Dataset

Following the literature [19, 29, 18], we use the simulation tool EPANET for data generation to evaluate the proposed CSI method. EPANET is a hydraulic modeling and simulation tool developed by the US environmental protection agency. It models a WDN by setting the topology, hydraulic and water quality parameters, which can then be used to run simulations to observe changes in hydraulics and water quality. In this paper, we adopt the benchmark Wolf-Cordera model [13], indicated by WDN_{wc}, as a basis for data generation. WDN_{wc} is a large-scale WDN which comprises of 1786 nodes and 1985 pipes together with a water hydraulic model and a water quality model. To get a source domain WDN and a target domain WDN, we divide

Table 1
Experimental parameters

Parameters	Value	Type
Contamination source	$0 \sim 1785$	Integer
Contamination concentration	$5 \sim 300$	Integer
Starting time (t_0)	$0 \sim 23$	Integer
Duration time	$1 \sim (23 - t_0)$	Integer
Nodal demand noise	$\max(0, \mathcal{N}(1, 0.25))$	Float
Nodal concentration noise	$\mathcal{U}(1 - 0.05, 1 + 0.05)$	Float

WDN_{wc} into two parts, as shown in Figure 3. The part on the left consisting of 1494 nodes is used as the WDN of the source domain while the smaller part on the right consisting of 292 nodes is used as the WDN of the target domain. The source domain here can be considered as the old town of a big city while the target domain here can be considered as a newly developed area in this city.

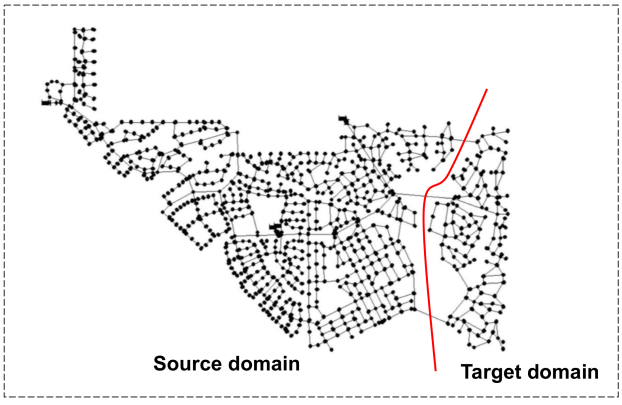

Figure 3: Water distribution networks of source and target domain

Based on EPANET, we simulate a number of contamination events that may occur in WDN_{wc}. For each contamination event, we use the default water demand pattern of WDN_{wc} to run a simulation over 24 hours and record the contamination concentration of all the nodes every 5 minutes. The contamination source node, concentration, starting time and duration are randomly set for each contamination event. To reflect the fact that the demand pattern of each node in a WDN may change over time, we add a Gaussian noise to the nodal demands. Moreover, to reflect the uncertainty caused by the fact that the sensor measurements may not be accurate, we add a random noise to the nodal concentration. The details of the parameter settings are shown in Table 1. In this way, we generated 10,000 contamination events with WDN_{wc}.

Next, we split the 10,000 contamination events into a training set and a test set according to the division of the source domain (training) and the target domain (testing) as shown in Figure 3. Furthermore, both the training samples and the test samples are processed following the procedure described in Section 4.1. Specifically, for a given contamination event, when the source node is in the source domain, we apply graph segmentation to the WDN of the source domain to search for a sub-network with a range of 80 to 100 nodes. In this case, the size of the largest sub-network $\mathcal{N}$ is 100. Then we randomly select M_s nodes in the sub-network as the sensor nodes. Taking the time when the contamination is first detected in any of the M_s sensor nodes as the starting time, we keep a sequence of sensor data for T time steps. If none of the above conditions is met, e.g., none of the sensor nodes detect any contamination, this event is discarded. Finally, padding operations are applied to the adjacency matrix and sensor data of the sub-networks for each simulated contamination event, as described in Section 4.1. As a result, we obtain an extended adjacency matrix with a size of 100×100, and a contamination matrix with a size of $100 \times T$. In this way, the training samples are generated. Similarly, when the contamination source node is in the target domain, the same procedures are used to generate the test samples.

Table 2
Dataset size

Time	$T = 12$ (30 minutes)		$T = 6$ (15 minutes)	
Sensor Node Number	$M_s = 6$	$M_s = 4$	$M_s = 6$	$M_s = 4$
Training data size	6539	5192	6883	5427
Test data size	1185	895	1237	952

For the evaluation of our proposed CSI method, different values of T and $M_s, \tilde{M}_s$ are used. Following [19], we experimented with $T = 12$ (30 minutes) and $T = 6$ (15 minutes), $M_s, \tilde{M}_s = 6$ and $M_s, \tilde{M}_s = 4$. As a result, we obtain four datasets under different sensor nodes selections and time limits, as shown in Table 2.

6. Experiments

As described in Section 5, the contamination events from WDN_{wc} are divided into a training set and a test set according to the division of the source domain and the target domain. We compare the performance of the proposed CSI method GrassL with five baseline methods on the test set in terms of the top K accuracy, indicated as Acc_K. The top K here represents the most suspicious K nodes predicted by a CSI model, i.e., these K nodes are considered by the CSI model to have the highest probability of being the source node. In this paper, we experimented with a set of values for K including 1, 5, and 10, and the corresponding accuracy is indicated as Acc_1, Acc_5 and Acc_{10}.

6.1. Baseline Methods

The existing CSI methods mostly focus on the study of a single WDN. To the best of our knowledge, there is no previous work that studied how to transfer the CSI knowledge learned from one WDN to a different WDN, i.e. cross-network CSI. Therefore, instead of directly comparing with an existing CSI method, we investigate the performance of different deep learning methods that could be used to capture both temporal and spatial dynamics. The baseline CSI methods used in this paper are as follows:

- CNN_DNN:
  This method respectively uses CNN to extract temporal features from the contamination matrix, and uses fully connected DNN to extract spatial features from the topological information encoded by the adjacency matrix. The outputs of these two parts are merged to identify the contamination source. The CNN module consists of three convolutional layers. The filter sizes are respectively $1\times3, 1\times3, 1\times1$, and their number of filters are respectively 4, 16 and 1. The DNN module consists of a fully connected layer with an input shape of $\mathcal{N}\times\mathcal{N}$ and an output shape of $\mathcal{N}$.

- LSTM_DNN:
  This method respectively uses LSTM (Long short-term memory) [7] to extract temporal features from the contamination matrix, and uses fully connected DNN to extract spatial features from the adjacency matrix. The outputs of these two parts are merged to identify the contamination source. The LSTM module consists of one LSTM layer with 128 hidden units and the DNN module is the same as that used in CNN_DNN.

- LSTM_DCNN:
  This method adopts a similar model structure as that used in GrassL. The difference is that this method replaces the CNN module of GrassL with the LSTM module used in LSTM_DNN. In specific, it first uses LSTM to extract the temporal features from the contamination matrix. Thereafter the outputs of the LSTM module and the adjacency matrix are fed into DCNN to identify the contamination source.

- DCNN_CNN:
  This method adopts the same set of models as that used in GrassL. The difference is that it changes the order of using CNN and DCNN.

- CNN_s-GCN:
  Similar to GrassL, this method combines a CNN with a graph convolutional network. The difference is that this method uses a spectral graph convolutional network (indicated as s-GCN) [9] instead of the spatial graph convolutional network DCNN.

The choice of these five baseline methods is to compare GrassL with a diverse set of methods that have been widely used in the literature for encoding temporal and spatial information [4][35]. In particular, we include two types of graph neural networks, i.e., spatial graph convolution networks (DCNN) and spectral graph convolution networks (s-GCN), which are widely used graph neural network models in the literature [35]. Among them, CNN_DNN and LSTM_DNN encode the topological information of WDNs through fully connected neural networks, while the other baseline methods LSTM_D-CNN, DCNN_CNN, CNN_GCN, as well as the C-DCNN used in GrassL adopt variants of graph based convolutional networks to capture the topological information. Adam is used as the optimiser to train the models of both GrassL and the baseline methods with a learning rate of 0.01.

6.2. Experimental Results and Analysis

Table 3 shows the performance of C-DCNN (the model used for the cross-graph learning module in GrassL) and the baseline methods under different parameter settings. It can be seen that under all the parameter settings, C-DCNN achieves the best performance with respect to the three evaluation metrics Acc_1, Acc_5 and Acc_{10}. The accuracy of C-DCNN increases 7.39% on average compared to the second best CSI method DCNN_CNN.

Specifically, compared with CNN+DNN, C-DCNN obtains an overall improvement of 17.78% for Acc_1, 34.04% for Acc_5, and 37.40% for Acc_{10}. The superiority of C-DCNN over CNN+DNN shows the importance of using graph-based convolutional networks to capture spatial information. Similarly, C-DCNN has a significant performance improvement compared to LSTM_DNN, with an increase of 29.58% for Acc_1, 40.85% for Acc_5, and 41.32% for Acc_{10}.

In addition, the superiority of C-DCNN over LSTM_D-CNN shows that the use of the CNN-based model to extract the temporal information of the contamination matrix is more effective than the use of the RNN based model. An explanation is that CNN uses a moving filter in the time dimension and the importance of each time point is the same when fusing the temporal features, while the RNN based model pays more attention to the most recent time points. For the task of CSI, contamination information from different time steps has the same importance along the time dimension. Compared with LSTM_DCNN, C_DCNN has an increase of 29.87% for Acc_1, an increase of 42.87% for Acc_5, and an increase of 44.23% for Acc_{10}. Similarly, CNN_DNN outperforms LSTM_DNN, which confirms that the CNN-based model is more capable of extracting the temporal information than the RNN-based model for the CSI task.

Compared with DCNN_CNN, C-DCNN obtains an overall improvement of 7.56% for Acc_1, 6.18% for Acc_5, and 8.43% for Acc_{10}. DCNN_CNN changes the order of applying CNN and DCNN in C-DCNN. The reason why C-DCNN achieves higher accuracy is that if DCNN is used first, the sequential contamination information for each node will be mixed in the output of DCNN, making it difficult for CNN to capture useful temporal features. However, if CNN is used first, it will only apply convolution operations along the time dimension. Therefore, the relations between the nodes are maintained such that DCNN could still capture the spatial features among the nodes from the outputs of CNN.

Finally, compared with CNN_GCN model, C-DCNN has an increase of 19.31% for Acc_1, an increase of 22.67% for Acc_5, and an increase of 22.88% for Acc_{10}. The superiority of C_DCNN shows that the spatial based GCN may be more suitable for the task of cross-network CSI than the spectral based GCN. The main reason is that DCNN is based on the idea of diffusion, which can integrate multi-level connection relationships in a single-layer DCNN. That is, in addition to

Table 3
CSI results on different methods

Methods		L=12		L=6	
		M=6	M=4	M=6	M=4
C-DCNN (GrassL)	Acc_1	**43.46%**	**36.43%**	**42.52%**	**37.08%**
	Acc_5	**58.31%**	**50.17%**	**59.34%**	**52.63%**
	Acc_{10}	**67.93%**	**59.33%**	**68.96%**	**61.66%**
CNN_DNN	Acc_1	35.36%	31.62%	36.14%	32.14%
	Acc_5	42.62%	38.66%	43.41%	39.60%
	Acc_{10}	49.20%	44.47%	49.15%	44.75%
LSTM_DNN	Acc_1	32.91%	27.71%	33.15%	29.31%
	Acc_5	40.59%	36.54%	40.99%	38.24%
	Acc_{10}	47.43%	42.91%	47.45%	44.54%
LSTM_DCNN	Acc_1	33.08%	28.16%	32.50%	28.99%
	Acc_5	40.51%	35.75%	41.07%	36.87%
	Acc_{10}	46.41%	42.57%	46.81%	42.86%
DCNN_CNN	Acc_1	39.16%	33.74%	40.26%	35.08%
	Acc_5	55.27%	46.82%	56.51%	49.16%
	Acc_{10}	65.06%	49.83%	65.40%	58.82%
CNN_s-GCN	Acc_1	36.46%	29.83%	35.49%	31.93%
	Acc_5	48.19%	40.67%	47.70%	43.17%
	Acc_{10}	55.27%	49.83%	54.08%	50.53%

fusing the information between directly connected nodes, it can also integrate the information between the nodes that are indirectly connected. Because the proportion of sensor nodes in WDNs is often small, using the operation of diffusion convolution is more effective for the task of CSI.

The experiment results above show that GrassL which takes C-DCNN as the model for cross-network learning outperforms all the baseline CSI methods. Moreover, GrassL provides a way to transfer the CSI knowledge learned from the historical contamination events collected from one WDN to a different WDN, and achieves an accuracy comparable with the approaches that can only apply to the same network [19].

7. Discussion and Limitation

In this paper, we investigate the problem of CSI in the context of knowledge transfer. That is, a CSI model trained on one WDN can be applied directly to a different WDN. Instead of training a CSI model based on the data of an entire WDN, we extract various sub-networks from the WDN and the sensor data contained in those sub-networks together with their topological information are used to train a CSI model. In this way, we relax the constraint of identical topology between the source domain and the target domain. Different WDNs or different areas of a big WDN usually share features from a local perspective, and the CSI model trained with those local features can be adapted to new WDNs.

In this paper, the adjacency matrix of GCNs only encodes the connectivity of the nodes of the WDN without the geometric information of the pipes such as their length and diameter. Though such geometric information is not directly included in the inputs, they may be partially inferred from the water quality sensor data given the nodal connectivity during the training of the graph neural networks. That is, when a contamination event occurs in a WDN, the length and diameter of the pipes determine the transmission speed of the contaminants which can be inferred from the difference of the time when the contaminants are detected by each sensor node. However, since the sensor nodes in a WDN are often sparse, the CSI model trained on one WDN may not work effectively when applying to a different WDN that consists of pipes with very different lengths and diameters without explicitly encoding the geometric information. Therefore, the CSI model presented in this paper is limited to WDNs with similar lengths and diameters. We leave the exploration of encoding such geometric information for future work.

8. Conclusion

In this paper, we propose GrassL, a graph convolutional network based method for the task of cross-network CSI, which is able to transfer the CSI knowledge learned from one WDN to a different WDN. In particular, GrassL provides a framework for extracting local contamination and topological information from a large WDN and using such information to train a model for identifying the contamination source node. Moreover, we adopt the state-of-the-art graph convolutional network model DCNN to capture the spatial features of the local topological information encoded by the adjacency matrix of WDNs. By carrying out a set of experiments with a benchmark WDN, we show the superiority of GrassL.

For future work, we will extend GrassL from two aspects. The first extension is on the amount of information encoded by the adjacency matrix. In this research, we only use the adjacency matrix to encode whether two nodes in a WDN are connected or not. Although graph convolutional networks such as DCNN or GCN could apply operations to extract deeper connections from the adjacency matrix, the current adjacency matrix is insufficient for expressing the complexity of a WDN. In future work, we will consider to encode geometric information such the length and diameter of the water pipelines. This may help graph convolutional networks to obtain finer-grained spatial information and provide more accurate predictions. The second extension is on the data values filled in the contamination matrix for the non-sensor nodes. In this research, to simplify the procedure of model training, the contamination data of the non-sensor nodes are all filled with zeros. Such operations may bring misleading information, as when there is no contamination detected the data values of the sensor nodes are also zeros. To alleviate the influence of this problem, we will investigate more effective ways of processing the non-sensor nodes.

References

[1] Al-Bedyry, N.K., Sathasivan, A., Al-Ithari, A.J., 2016. Ranking pipes in water supply systems based on potential to cause discolored water complaints. Process Safety and Environmental Protection 104, 517–522. Challenges in Environmental Science and Engineering.

[2] Atwood, J., Towsley, D., 2016. Diffusion-convolutional neural networks, in: NIPS.

[3] Chen, Q., Shen, G., Jiang, J., Diao, X., Wang, Z., Ni, L., Dou, Z., 2018. Effect of rubber washers on leak location for assembled pressurized liquid pipeline based on negative pressure wave method. Process Safety and Environmental Protection 119, 181–190.

[4] Cheng, X., Zhang, R., Zhou, J., Xu, W., 2018. Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting, in:

2018 International Joint Conference on Neural Networks (IJCNN), IEEE. pp. 1–8.
[5] Datta, S., Sarkar, S., 2016. A review on different pipeline fault detection methods. Journal of Loss Prevention in the Process Industries 41, 97–106.
[6] Guo, G., Cheng, G., 2019. Mathematical modelling and application for simulation of water pollution accidents. Process Safety and Environmental Protection 127, 189–196.
[7] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9, 1735–1780.
[8] Huang, J.J., McBean, E.A., 2009. Data mining to identify contaminant event locations in water distribution systems. Journal of Water Resources Planning and Management .
[9] Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 .
[10] Laird, C.D., Biegler, L.T., van Bloemen Waanders, B.G., Bartlett, R.A., 2005. Contamination source determination for water networks. Journal of Water Resources Planning and Management 131, 125–134.
[11] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural computation 1, 541–551.
[12] Li, Z., Zhang, H., Tan, D., Chen, X., Lei, H., 2017. A novel acoustic emission detection module for leakage recognition in a gas pipeline valve. Process Safety and Environmental Protection 105, 32–40.
[13] Lippai, I., . Wolf-cordera ranch. http://emps.exeter.ac.uk/engineering/research/cws/resources/benchmarks/expansion/wolf-cordera-ranch.php Accessed February 4, 2021.
[14] Liu, L., Ranjithan, S.R., Mahinthakumar, G., 2010. Contamination source identification in water distribution systems using an adaptive dynamic optimization procedure. Journal of Water Resources Planning and Management 137, 183–192.
[15] Mallick, T., Balaprakash, P., Rask, E., Macfarlane, J., 2020. Transfer learning with graph neural networks for short-term highway traffic forecasting. arXiv preprint arXiv:2004.08038 .
[16] Perelman, L., Arad, J., Housh, M., Ostfeld, A., 2012. Event detection in water distribution systems from multivariate water quality time series. Environmental science & technology 46, 8212–8219.
[17] Perelman, L., Ostfeld, A., 2013. Bayesian networks for source intrusion detection. Journal of Water Resources Planning and Management 139.
[18] Preis, A., Ostfeld, A., 2007. A contamination source identification model for water distribution system security. Engineering optimization 39, 941–947.
[19] Qian, K., Jiang, J., Ding, Y., Yang, S.H., 2021. Dlgea: a deep learning guided evolutionary algorithm for water contamination source identification. Neural Comput & Applic , 1–15.
[20] Rossman, L.A., 2000. Epanet 2: Users manual .
[21] Shang, F., Uber, J.G., Polycarpou, M.M., 2002. Particle backtracking algorithm for water distribution system analysis. Journal of environmental engineering 128, 441–450.
[22] Sun, L., Yan, H., Xin, K., Tao, T., 2019. Contamination source identification in water distribution networks using convolutional neural network. Environ Sci Pollut Res , 36786–36797.
[23] Vankayala, P., Sankarasubramanian, A., Ranjithan, S.R., Mahinthakumar, G., 2009. Contaminant source identification in water distribution networks under conditions of demand uncertainty. Environmental Forensics 10, 253–263.
[24] Wang, H., Harrison, K.W., 2013. Bayesian approach to contaminant source characterization in water distribution systems: adaptive sampling framework. Stoch Environ Res Risk Assess 27, 1921–1928.
[25] Wang, L., Geng, X., Ma, X., Liu, F., Yang, Q., 2018. Crowd flow prediction by deep spatio-temporal transfer learning. arXiv preprint arXiv:1802.00386 .
[26] Wang, W., Shen, K., Wang, B., Dong, C., Khan, F., Wang, Q., 2017. Failure probability analysis of the urban buried gas pipelines using bayesian networks. Process Safety and Environmental Protection 111, 678–686.
[27] Wei, Y., Zheng, Y., Yang, Q., 2016. Transfer knowledge between cities, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1905–1914.
[28] wikipedia, 2014. 2014 Elk River chemical spill. https://en.wikipedia.org/wiki/2014_Elk_River_chemical_spill. [Online; accessed 30-Feb-2021].
[29] Yan, X., Gong, J., Wu, Q., 2020. Pollution source intelligent location algorithm in water quality sensor networks. Neural Comput & Applic .
[30] Yan, X., Gong, W., Wu, Q., 2017. Contaminant source identification of water distribution networks using cultural algorithm. Concurrency Computat: Pract Exper 29, 1–11.
[31] Yan, X., Yang, K., Hu, C., 2018. Pollution source positioning in a water supply network based on expensive optimization. Desalin Water Treat 110, 308–318.
[32] Yan, X., Zhao, J., Hu, C., Zeng, D., 2019. Multimodal optimization problem in contamination source determination of water supply networks. Swarm and Evolutionary Computation 47, 66–71.
[33] Yao, H., Liu, Y., Wei, Y., Tang, X., Li, Z., 2019. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction, in: The World Wide Web Conference, pp. 2181–2191.
[34] Zadkarami, M., Shahbazian, M., Salahshoor, K., 2017. Pipeline leak diagnosis based on wavelet and statistical features using dempster–shafer classifier fusion technique. Process safety and environmental protection 105, 156–163.
[35] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M., 2020. Graph neural networks: A review of methods and applications. AI Open 1, 57–81.