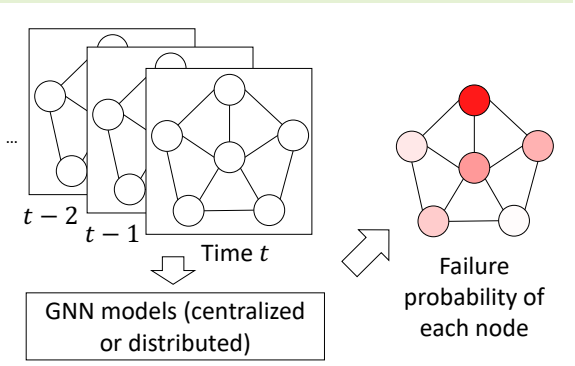
Detecting Inaccurate Sensors on a Large-Scale Sensor Network Using Centralized and Localized Graph Neural Networks

Dennis Y. Wu, Tsu-Heng Lin, Xin-Ru Zhang, Chia-Pan Chen, Jia-Hui Chen, and Hung-Hsuan Chen

***Abstract*— This paper conducts an empirical study on detecting faulty sensors in a large-scale sensor network containing approximately $10,000$ sensors distributed over $36,000$ $\mathbf{km}^2$. First, we discuss the practical challenge of this task. We compare rule-based models, traditional machine learning models, deep learning models without graph neural networks, and deep learning models with graph neural networks. The experimental results show that graph neural networks identify more problematic sensors in fewer trials than rule-based models and other machine learning and deep learning models. In addition to training the models in a central server, we also show that localized versions of the deep learning models with graph neural networks yield predictive power comparable to centralized training. Consequently, each sensor may perform a local inspection to identify its health status and only send reminder signals to a centralized server if it is self-diagnosed as a faulty sensor.**



***Index Terms*— Graph neural network, graph convolutional network, sensor network, anomaly detection, automatic inspection, PM2.5**

I. Introduction

AIR quality sensors play a crucial role in detecting air pollution, which is a critical environmental protection issue [1]. However, the quality of the monitored data may be unreliable, especially when the sensors are mostly low-cost sensors and are rarely inspected [2], [3]. As a result, downstream analyses might be problematic because they rely on the sensors' monitored values, which are likely distorted.

This paper aims to discover inaccurate sensors based on graph neural networks (GNNs) through the spatiotemporal information derived from sensors. In the following, we use the terms "inaccurate sensors", "problematic sensors", "anomalous sensors" and "malfunctioning sensors" interchangeably. While this task can be modeled as a standard binary classification problem, we usually have no ground truth information regarding sensors' health statuses as training targets, so directly applying supervised classification algorithms is impractical. Recently, Lin et al. suggested a two-stage approach — predicting sensors' monitored values first and then classifying the sensors whose monitored values vary most from the predicted values as the malfunctioning sensors [4]. This approach turns out to be effective, likely because of the myriad number of monitored data that are available from sensors. The same paper showed that predicting the future monitored values of a malfunctioning sensor is more difficult than doing so for a healthy sensor, so this two-stage approach is likely reasonable. They showed that when using deep neural networks as the predicting model for the first stage, the two-stage approach effectively discover the problematic sensors in terms of both precision and recall.

We continue this line of research by using GNNs in the first stage. In particular, we leverage both temporal convolutional and graphical convolutional neural networks to model the nonlinear and high-dimensional temporal and spatial information contained in the monitored values derived from sensors. In addition to centralized training models, we also test localized models and find that the predictive power of the localized models is comparable to that of the centralized models. As a result, each sensor may apply a local model to inspect its health status and then sends a signal to the centralized server only when it is self-diagnosed as problematic. Local models may largely reduce the required communication and computational costs.

We utilize a nationwide air quality sensor network as our experimental target. In particular, the Environmental Protec-

This work was supported in part by the National Science and Technology Council in Taiwan under grant numbers 110-2222-E-008-005-MY3 and 110-2634-F-008-008. (Corresponding author: H.-H. Chen)

Wu, Lin, and Zhang were with National Central University, Taoyuan 32001, Taiwan (e-mail: denniswu1126@gmail.com, neilbeeber@gmail.com). Chia-Pan Chen and Jia-Hui Chen are with the Industrial Technology Research Institute, Hsinchu 310401, Taiwan (e-mail: peter_chen@itri.org.tw, unachen0325@gmail.com). Hung-Hsuan Chen is with National Central University, Taoyuan 32001, Taiwan (e-mail: hhchen@acm.org).

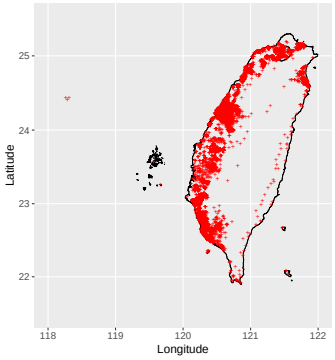


Fig. 1. The distribution of the working sensors in July 2021 in Taiwan and the outlying islands, including the Penghu Islands, Kinmen Island, Green Island, Orchid Island, and Liuqiu Island. The area of the Taiwan island is approximately $\mathbf{36,000}$ $\mathrm{km}^{\mathbf{2}}$.

tion Administration (EPA) of the Executive Yuan in Taiwan has gradually built and operated over $10,000$ air quality sensors since 2017, and this number continues to grow. A methodology operating at this scale should prove its practical effectiveness and robustness in discovering malfunctioning sensors. A geographical distribution of the working sensors in July 2021 is shown in Figure 1.

We summarize our threefold contributions below.

- We empirically validate the fact that the proposed two-stage framework can identify malfunctioning air quality sensors. This model requires no sensor health status labels. Consequently, this model is practical for identifying malfunctioning sensors in a large-scale sensor network, which usually has no ground-truth labels regarding sensors' health statuses.
- We picked two representative spatiotemporal GNN models, Graph WaveNet and spatiotemporal Graph Convolutional Network (STGCN), to empirically validate the effectiveness of centralized and localized GNNs in integrating spatiotemporal information to predict malfunctioning air quality sensors. We compare these GNNs with baseline models, including rule-based models, traditional machine learning models, and deep learning models, based on the dataset collected from the nationwide sensor network introduced above. The experimental results show that the GNNs outperform the baseline models in terms of area under the ROC curve, precision-at-n, and recall-at-n. As a result, the approach can effectively reduce the manpower required for inspections.
- To the best of our knowledge, our dataset is the largest available in the area of identifying problematic air quality sensors. With a vast amount of data points, our dataset provides a comprehensive view of sensor behavior and enables us to compare the effectiveness of various algorithms on a large scale.

The paper is organized as follows. Section II reviews previous methods for anomaly detection, especially for air quality prediction. Section III introduces the two-stage anomaly detection framework and deep spatiotemporal graph models. Section IV presents experiments on different methods. Finally, Section V discusses our discovery and future works.

II. Related Work

We review previous works on anomaly sensor detection and introduce the applications and research regarding spatiotemporal pattern mining in this section.

A. Anomalous Sensor Detection

The detection of anomalous sensors has been studied from different perspectives. Perhaps the most straightforward approach involves applying statistics-based methods and rule-based methods to find anomalous sensors. For example, the authors of [5], [6] discovered spatial anomalies and temporal anomalies based on human-defined rules. However, the effectiveness of rule-based methods is limited by the rule makers' expertise, and it is likely difficult to capture the high-dimensional and nonlinear interactions among the temporal, spatial, and perhaps other types of information.

An alternative approach for identifying anomalous sensors is through machine learning models. When target labels are unavailable, unsupervised approaches such as clustering [7] and principal component analysis [8], [9] are commonly used. However, supervised learning models such as simple linear models [10], support vector machines [11], random forests, Gaussian processes [12], or deep learning models [13] can be applied if the target variables are available. Unfortunately, the health statuses of sensors are often unavailable. To address this challenge, a two-stage approach was proposed by [4], [14]. While previous studies (e.g., [14]–[17]) have also suggested applying GNN models to detect anomalies in sensor networks, their experiments were conducted on smaller datasets (ranging from dozens to at most hundreds of nodes) [14], [15] or synthetic dataset [16], [17]. Our method, however, is tested on a network with tens of thousands of nodes, which is significantly larger than previous works. For further reading, surveys on anomaly detection for IoT systems and sensor networks are available in [18] and [19].

The discovered anomalies or outliers may not necessarily represent malfunctioning sensors. For example, the authors of [20] use topological characteristics, such as betweenness and closeness centralities, to infer the outliers. Another example is the PurpleAir dataset [21], which provides outdoor/indoor labels for each sensor; these labels are further used as the normal/abnormal labels in some studies, as the ground-truth information about sensors' statuses (normal or malfunctioning) is usually difficult to obtain on a large scale. In our paper, we define anomalous or malfunctioning sensors as those that yield inaccurate measurements when compared with those provided by a lab-calibrated sensor. We obtained the ground-truth labels of the test instances based on the on-field inspection records.

B. Spatiotemporal Pattern Mining

Spatiotemporal pattern mining involves studying the complex relationships among spatial, temporal, and perhaps other affiliated features, along with the target variables. However, integrating these various types of features into one model is sometimes tricky because the sizes of the spatial and temporal characteristics are likely dynamic, and the dependencies

among the data instances may violate the independent and identically distributed (i.i.d.) assumption of many learning models.

To deal with the dynamic sizes and data dependencies of temporal features, a possible approach is to utilize a Markov chain, which assumes that the memorylessness Markov property holds to simplify the constructed model [22]. Another strategy that has become popular in the last decade is the use of recurrent neural networks (RNN) and their variations (e.g., long short-term memory), which memorize sequential information via the internal states of neurons [23], [24]. Spatial features are sometimes integrated based on one-dimensional or two-dimensional convolutional neural networks. However, such designs assume that the spatial features form a grid-like shape [25], [26], which is probably an oversimplification of a natural environment. A graph is an essential structure for modeling the relationships among various objects, such as scientific literature coauthoring relationships [27], [28], the interactions between the characters of a novel [29], word co-occurrence relationships [30], recommender system relationships [31]–[33], disease relationships [34], and many more. A graph neural network (GNN) provides tools for bridging important data structure graphs with deep learning technology [35]. Among the various GNN models, graph convolutional networks (GCNs) are extremely popular, likely because of their elasticity and efficiency: a GCN's training time grows only linearly with the number of edges [36]. GCNs have been applied to model various spatial-temporal tasks, e.g., traffic prediction [37]–[40], trajectory prediction [41], link prediction [42], and detecting stealth false data injection attacks [43]. In this paper, we rely on a GCN to capture spatial information and integrate it with the temporal information captured by a temporal convolution network (TCN) or a gated linear unit (GLU).

III. Model

Classifying the health status (i.e., normal or inaccurate) for sensors using supervised learning is challenging due to the unavailability of their status labels. Therefore, supervised classifiers with cross-entropy loss cannot be directly applied. Instead, we leverage a large number of monitored PM2.5 values to identify inaccurate sensors using a two-stage framework, as proposed in [4]. In stage 1, we use a supervised regressor to predict the near-future PM2.5 values of each sensor. In stage 2, we identify problematic sensors by selecting those whose predicted values differ significantly from the monitored values. This approach avoids the need for direct classification while still enabling the accurate detection of inaccurate sensors.

This section introduces the two-stage approach for discovering inaccurate sensors in detail (Section III-A and Section III-B). In addition to centralized training and prediction, we also introduce a local model that can be trained based only on local information (Section III-C).

A. Stage 1: Predicting Future Monitored Values

In [4], the authors applied various supervised regressors (e.g., ridge regression, the Lasso, a random forest, a multilayer perceptron, and long short-term memory) to predict each sensor's monitored PM2.5 values in stage 1. In this paper, we apply deep spatiotemporal GNN models because these types of models naturally integrate spatial and temporal information.

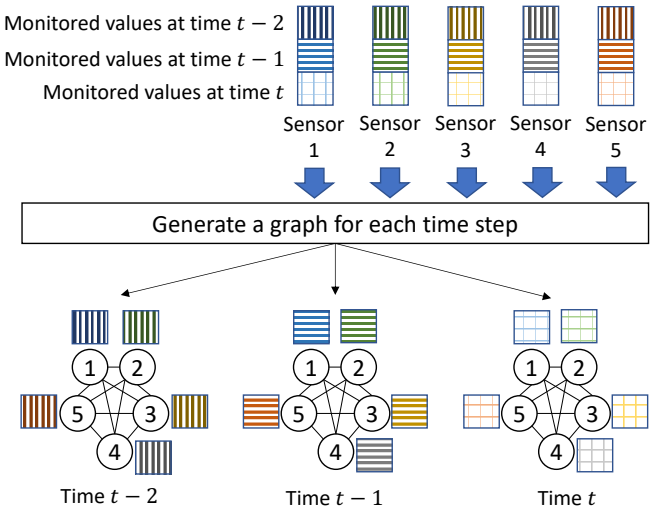


Fig. 2. An example of converting the sequential monitoring data derived from five sensors to graphs.

Figure 2 gives an example of converting the sequential PM2.5 information derived from $n = 5$ sensors into graphs. In particular, for each time unit, we treat each sensor as a node and connect every pair of sensors with an edge. Therefore, we generate a complete graph for each time unit. Although each graph is topologically identical, the monitored PM2.5 values of the same sensor at different time points may vary.

In our task, the conventional approach of constructing graphs with a threshold τ beyond which the weight between nodes is set to zero is not suitable. This is because of the non-uniform geographical distribution of sensors, as depicted in Figure 1, which presents a significant challenge. In particular, sensors in the middle or east regions of Taiwan are widely spaced apart, while those in the west are densely located. Therefore, if we want to ensure that the sensors in the middle or east regions are well connected to their neighboring sensors, we have to set τ to a small value, which ultimately results in a highly dense network. For simplicity, we opt for a fully connected network.

We test two different GNNs – Graph WaveNet model [39] and the spatiotemporal GCN (STGCN) [38] – because they are simple and have produced great prediction results in other spatiotemporal prediction tasks [38], [39]. We also study their variations that require only neighboring information in Section III-C.

1) Graph WaveNet: Graph WaveNet considers both temporal and spatial information. In particular, Graph WaveNet captures temporal information based on a dilated causal convolution and a gating temporal convolution layer (gated TCN). The dilated convolution can be considered a 1-dimensional convolutional neural network (CNN) applied on the time domain. The filter is applied over a time period larger than

its length by skipping input values with a certain step size. As a result, the dilated convolution has large receptive fields with few layers. The gating mechanism controls the flow of information to the next layer.

After dealing with the temporal information, Graph WaveNet iterates through each timestamp to capture the underlying spatial pattern based on the graph convolution layer module, which is composed of two components. First, a diffusion convolution models the information flow based on a k-step random walk to generate a graph convolution. Second, a self-adaptive adjacency matrix is automatically learned to capture the latent relationships among different nodes, which may alleviate the inefficiency of the fixed adjacency matrix.

In summary, the Graph WaveNet model applies a l-layer gated TCN to capture the temporal information ($l = 3$ in our setting). The outputs of the gated TCN are fed into the graph convolution layer to obtain spatial information. The Graph WaveNet model stacks m spatiotemporal layers through a dilated causal convolution to handle both the short-term and long-term temporal information ($m = 4$ in our setting).

2) Deep STGCN: The STGCN considers both temporal and spatial information based on a temporal attention layer and an approximated graph convolution layer. The temporal attention layer applies a one-dimensional convolution on the time domain to capture temporal relationships. The output is further transformed by gated linear units (GLUs) to capture nonlinearity. For the spatial information, the STGCN utilizes Chebyshev polynomials to approximate the first-order spectral graph convolution (i.e., the scaled graph Laplacian matrix) [36]. As a result, this structure requires fewer parameters and is therefore highly efficient for large networks. When increasing the depth of the layers, the approximation process resembles the k-th-order spectral graph convolution.

To apply Graph WaveNet (introduced in Section III-A.1) and the STGCN (introduced in Section III-A.2) to predict the future monitored values of a PM2.5 sensor, we construct a complete graph by treating each sensor as a node, and every node connects to every other node. We assign w_{ij} as the weight of an edge between node i and node j via the radial basis function (RBF) kernel, as shown in Equation 1.

$$w_{ij} = \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \tag{1}$$

where d_{ij} denotes the geographical distance between sensor i and sensor j, and $\sigma = 10$ is a hyperparameter. We set the value of the hyperparameter based on a grid search conducted on the validation set.

We apply the RBF kernel as it is commonly used as the similarity score between two locations in geographic information analyses, e.g., [44]. The RBF kernel is popular because the output is negatively correlated with d_{ij}, the geographical distance between a location i and a location j.

B. Stage 2: Anomaly Detection

The anomaly detection module calculates the R^2 score between the predicted PM2.5 values in the near future (using the results in Stage 1) and the actual monitored PM2.5 values for each sensor. We use the R^2 score instead of other regression evaluation metrics, such as the root-mean-square error or the mean absolute error, because the R^2 score is a normalized score, making it easier to interpret. In contrast, the value of the root-mean-square error is usually influenced by the magnitude of the true y-s.

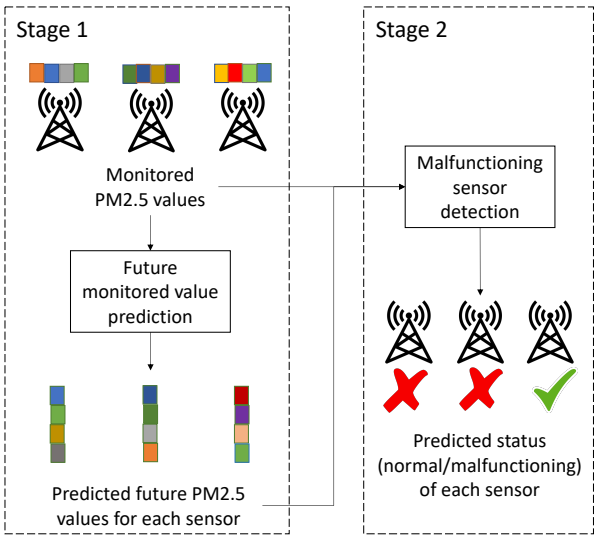


Fig. 3. The entire workflow of the two-stage method for discovering inaccurate sensors.

The R^2 score for a sensor i is defined by Equation 2.

$$R^2\left(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}^{(i)}\right) = 1 - \frac{\sum_{t=1}^{T}\left(y_t^{(i)} - \hat{y}_t^{(i)}\right)^2}{\sum_{t=1}^{T}\left(y_t^{(i)} - \bar{y}^{(i)}\right)^2}, \tag{2}$$

where $\boldsymbol{y}^{(i)} = \left[y_1^{(i)}, \ldots, y_T^{(i)}\right]$ represents the monitored PM2.5 values at the future T time units for sensor i, $\hat{\boldsymbol{y}}^{(i)} = \left[\hat{y}_1^{(i)}, \ldots, \hat{y}_T^{(i)}\right]$ denotes the predicted values at the future T time units for sensor i, and $\bar{y}^{(i)}$ is the average monitored value for sensor i based on the training data.

Rather than relying on a fixed threshold γ to identify problematic sensors, we propose ranking sensors by their R^2 scores in ascending order. This approach prioritizes sensors with the lowest R^2 scores, indicating that they are more likely to be inaccurate. To determine the number of sensors to inspect, we suggest the inspection team consider their available resources and choose the top n sensors for examination.

The entire workflow (Stage 1 and Stage 2) is shown in Figure 3.

C. Local GNN Models

A GNN-based centralized PM2.5 predictor (as explained in Section III-A.1 and Section III-A.2) requires the generation of a large complete graph (containing approximately $10,000$ nodes in our case) for each time unit. Additionally, all sensors must transmit their monitored values to a centralized computing unit for training and prediction, which might be an issue for low-cost sensors with limited computational and communication resources.

To economically utilize the limited capacity of such low-cost sensors, we discuss variations of the Graph WaveNet and STGCN models in which each sensor can diagnose its health status using local information from neighboring nodes. Our intuition is that a pair of distant nodes probably have smaller impacts on each other than nodes that are close together, so we can probably ignore the distant sensors and only focus on the local sensors. As shown in Equation 1, when the distance d_{ij} between sensor i and sensor j is large, w_{ij} is close to zero. Therefore, instead of generating a graph for all sensors, perhaps we can generate a small graph containing only the sensors that are local to each sensor.

Eventually, we train a distinct model for each sensor i based on a local complete graph that includes sensor i and its five nearest neighbor sensors. Therefore, sensor i's future monitored value is predicted by a small complete graph containing only six sensors. The observed PM2.5 values are transmitted only to local neighbors.

The Graph WaveNet model leverages a standard graph convolution network (GCN) to process information at each timestamp. However, the time complexity for each timestamp is $O(led+lnd^2)$, where l represents the number of layers, e is the number of edges, n denotes the number of nodes, and d is the number of features in each node [45]. Given that a complete graph has $n(n-1)/2$ edges, the training time for Graph WaveNet grows quadratically with the number of input nodes. Similarly, the Spatio-Temporal Graph Convolutional Network (STGCN) has a time complexity that grows linearly with the number of edges [38], which also results in quadratic growth of the training time with the number of inputs. Therefore, using the local GNN models significantly reduce the training time because the network size is much smaller.

IV. Experiment

This section introduces the utilized experimental dataset, the preprocessing steps conducted on the experimental dataset, the baseline methodologies employed for comparison purposes, the selected evaluation metrics, and the experimental results.

A. Experimental Dataset

Our experimental dataset includes two parts.

The first part contains the historical PM2.5 values monitored by the sensors. This dataset is released by the Environmental Protection Administration (EPA) of Taiwan and can be downloaded directly from the website "Civil IoT Taiwan".[1] However, since the majority of these sensors are low-cost sensors (only 77 out of the 10000+ sensors are highly expensive and accurate sensors that are continuously inspected every week), most of the monitored values may be unreliable.

The second dataset contains sensor maintenance records for 2018 retrieved from the Green Energy and Environment Research Laboratories of the Industrial Technology Research Institute — the institute that maintains the sensors studied in this paper. The maintenance records include 144 sensors that were inspected between January 2018 and December 2018. Among the 144 sensors, 116 are normal (labeled 0 in our paper), and 28 are out-of-order (labeled 1). All the sensors are located in Taichung City, the largest city in central Taiwan.

[1] https://ci.taiwan.gov.tw/air-quality

B. Data Preprocessing

The sensorsâĂŹ monitored PM2.5 values are occasionally missing in the dataset released by the EPA. To deal with a short-term missing value, we fill it in with the latest available value before the missing data point. If the missing period is longer than one day, we use the data from the previous day to fill in the missing values.

We standardize the monitored PM2.5 values using Z-score normalization, as defined below.

$$x' = \frac{x - \bar{x}}{\sigma_x} \tag{3}$$

where x is the original PM2.5 value, $\bar{x}$ denotes the mean of all the monitored PM2.5 values in the training data, σ_x represents the standard deviation, and x' is the normalized PM2.5 value.

C. Compared Baselines

We compare the proposed method with several baseline models, including a famous rule-based model that discovers inaccurate air quality sensors (ADF [5]), traditional machine learning models (a random forest [46], [47], the Lasso [48], [49], and ridge regression [50], [51]), and deep learning models without graphs (a multilayer perceptron (MLP) [52], [53] and long short-term memory (LSTM) [54]).

D. Evaluation Metrics

We use the area under the receiver operating characteristic curve (AUROC), precision-at-n ($P@n$), and recall-at-n ($R@N$) measures as the evaluation metrics. We select these metrics for the reasons described below.

The receiver operating characteristic (ROC) curve shows the relationship between the true positive rate and false positive rate by varying the discrimination threshold. The AUROC is equivalent to the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative instance. Therefore, the AUROC score is immune to various evaluation issues that are caused by an imbalanced dataset. Additionally, we do not need to worry about threshold values that may influence the scores of the evaluation metrics.

The precision-at-n metric measures the ratio of the number of true-positive instances (when classifying the top-n values as positives) to n. In our application scenario, the precision-at-n metric estimates the number of discovered inaccurate sensors if the inspection team can only afford to inspect n sensors. The definition of precision-at-n is shown in Equation 4.

$$P@n = \frac{\text{number of inaccurate sensors in the top } n}{n} \tag{4}$$

We also report the recall-at-n score, which is defined as the number of true-positive- instances (when classifying the top-n results as positives) in the top n divided by the total number of positives instances. In our application, recall-at-n

measures the proportion of discovered inaccurate sensors out of all inaccurate sensors when the inspection team can only inspect n sensors. The definition is given in Equation 5.

$$R@n = \frac{\text{number of inaccurate sensors in the top } n}{\text{number of inaccurate sensors in total}} \quad (5)$$

TABLE I
AUROC SCORES OF DIFFERENT METHODS. WE HIGHLIGHT THE LARGEST AND THE SECOND-LARGEST AVERAGE AUROC SCORES IN BOLD AND WITH A BOX, RESPECTIVELY.

Type	Model	AUROC (mean ± stdev)
Rule-Based	ADF-5	0.624
Traditional ML	Random Forest	0.6878 ± 0.0063
	Lasso	0.7000 ± 0.0157
	Ridge	0.7085 ± 0.0135
DNN	MLP	0.6940 ± 0.0072
	LSTM	0.7090 ± 0.0072
GNN	Graph WaveNet	0.7216 ± 0.0147
	STGCN	$\mathbf{0.7220} \pm 0.0142$

E. Results

Table I gives the AUROC scores of a rule-based model, traditional machine learning models, deep learning models, and GNN models. The rule-based ADF model detects a sensor's health status by comparing the sensor's monitored values with the neighboring sensors' monitored values based on the rules defined by experts. The original paper examined five nearest neighbors [5], which is denoted as ADF-5 in the following. Since all other models are training-based methodologies that may yield slightly different predictions every time, we repeat the experiments for each model five times and report both the mean and standard deviation of the AUROC scores obtained from the five trials. We list mean and standard deviation, instead of the traditional p-values, because (1) mean and standard deviation provide more information about the distribution of the scores, and (2) p-values are often harder to correctly interpret [55]. As shown in Table I, the GNN-based models (Graph WaveNet and the STGCN) yield better average AUROC scores stably (as shown by the small standard deviations) than all the baseline models. Furthermore, while the STGCN model exhibits a higher average AUROC score compared to the Graph WaveNet model, the observed difference may be deemed insignificant since the mean value of the Graph WaveNet's AUROC score plus its standard deviation is greater than the mean value of STGCN's AUROC score.

Table II gives the precision-at-n scores of various methods for $n = 10, 20, \ldots, 50$. Such a setup simulates the real-world scenario in which the inspection team is only able to inspect n sensors and wants to estimate the percentage of inaccurate sensors out of all inspected sensors. Based 2018 on the inspection records provided by the Green Energy and Environment Research Lab at the Industrial Technology Research Institute, 28 out of the 144 inspected sensors were malfunctioning. Since the inspected sensors are obtained from a randomly selected region, we estimate that approximately $(28/144) \times 100\% \approx 19.44\%$ of the sensors are inaccurate on average. As a result, the precision-at-n for a random inspection is 0.194 for all n-s. The precision-at-n values for a random inspection and the other compared methods are shown in Table II. As shown, the GNN-based methods are commonly in the top-2 in terms of the precision-at-n metric, especially when n becomes larger.

Table III displays the recall-at-n values of different methods. When the inspection team can only examine n sensors, the recall-at-n metric estimates the proportion of discovered problematic sensors out of all truly inaccurate sensors. Since 19.44% of the sensors are out-of-order, randomly inspecting n sensors is expected to yield $0.1944n$ problematic sensors, which leads to a recall-at-n value of $0.1944n/28$, as shown in Table III. We also show the results of the other comparison models in the same table. Again, the GNN-based methodologies are commonly among the two methodologies with the highest recall-at-n scores.

F. Global Models vs. Local Models

For each GNN model, we compare the AUROC scores produced by the global and local training methods.

The global method leverages all sensors' monitored information to build a unified model. This is exactly the methodology introduced in Section III-A and Section III-B. All the numbers related to the GNN models reported in previous tables are based on the global unified models.

The local models aim to use only the monitored PM2.5 values from a sensor i and its five closest neighbor sensors to build a local prediction model for sensor i. Although such a methodology requires building a distinct model for each sensor, each model can exchange and share information only among its local sensors.

Table IV gives the AUROC scores of the global and local versions of the Graph WaveNet and STGCN training models. There appears to be no significant difference between these two training methods. Table V and Table VI provide comparisons between the global and local methods in terms of their precision-at-n and recall-at-n values, respectively. The global and local methods produce no apparent winner here as well. Therefore, if transmitting all monitored values to a centralized server is an issue, we can always use the local methods to train GNN-based models while still obtaining comparable results.

G. The Influence of the Similarity Scores

We compare the AUROC scores using different similarity measures for Graph WaveNet and STGCN in this section. We compare the RBF kernel (Equation 1) with the polynomial kernel and Laplacian kernel.

The polynomial kernel uses the dot product to determine the similarity between two points. When the degree is larger than 1, the polynomial kernel determines not only the similarity between the original feature space but also a combination of the features. The formula of the polynomial kernel between two points i and j is shown in Equation 6.

TABLE II

PRECISION-AT-n VALUES ACHIEVED BY VARIOUS METHODS. WE HIGHLIGHT THE LARGEST AND THE SECOND-LARGEST AVERAGE AUROC SCORES IN BOLD AND WITH A BOX, RESPECTIVELY.

Type	Model	P@10	P@20	P@30	P@40	P@50
Random		0.194	0.194	0.194	0.194	0.194
Rule-Based	ADF-5	0.300	0.350	0.270	0.330	0.320
Traditional ML	Random Forest	0.380	0.370	0.400	0.342	0.320
	Lasso	0.580	0.430	0.394	0.370	0.320
	Ridge	**0.600**	0.433	0.395	0.375	0.337
DNN	MLP	0.500	0.430	0.374	0.344	0.312
	LSTM	**0.600**	0.410	0.368	0.332	0.336
GNN	Graph WaveNet	0.540	0.420	0.400	0.378	0.348
	STGCN	0.588	**0.438**	**0.419**	**0.414**	**0.372**

TABLE III

RECALL-AT-n VALUES FOR VARIOUS METHODS. WE HIGHLIGHT THE LARGEST AND THE SECOND-LARGEST AVERAGE AUROC SCORES IN BOLD AND WITH A BOX, RESPECTIVELY.

Type	Model	R@10	R@20	R@30	R@40	R@50
Random		0.069	0.139	0.208	0.278	0.347
Rule-Based	ADF-5	0.110	0.250	0.290	0.460	0.570
Traditional ML	Random Forest	0.136	0.266	0.428	0.484	0.570
	Lasso	0.204	0.306	0.422	0.524	0.570
	Ridge	0.210	0.308	0.423	0.533	0.603
DNN	MLP	0.180	0.308	0.398	0.484	0.560
	LSTM	**0.214**	0.293	0.394	0.474	0.600
GNN	Graph WaveNet	0.190	0.302	0.428	0.538	0.622
	STGCN	0.209	**0.314**	**0.449**	**0.594**	**0.665**

TABLE IV

GLOBAL MODELS VS. LOCAL MODELS BASED ON THEIR AUROC SCORES. FOR THE SAME GNN MODEL, THE GLOBAL AND LOCAL METHODS YIELD COMPARABLE AUROC SCORES.

Model	Type	AUROC (mean $\pm$ stdev)
Graph WaveNet	Global	0.7216 ± 0.01469
	Local	0.7260 ± 0.0108
STGCN	Global	0.7220 ± 0.0142
	Local	0.7214 ± 0.0186

TABLE V

GLOBAL MODELS VS. LOCAL MODELS IN TERMS OF PRECISION-AT-n. FOR THE SAME GNN MODEL, WE HIGHLIGHT THE WINNER BETWEEN THE GLOBAL AND THE LOCAL METHODS USING BOLD FONT.

Model	Type	P@10	P@20	P@30	P@40	P@50
Graph WaveNet	Global	0.540	**0.420**	0.400	0.378	0.348
	Local	**0.600**	0.417	**0.417**	**0.380**	**0.360**
STGCN	Global	0.588	0.438	**0.419**	**0.414**	**0.372**
	Local	**0.640**	**0.450**	0.398	0.386	0.360

$$w_{i,j} = (n_i^T n_j + c)^d, \tag{6}$$

where n_i and n_j are the latitude and longitude of sensor i and sensor j, respectively, c and d are two scalars.

The Laplacian kernel is similar to the RBF kernel, as shown by Equation 1 and Equation 7, respectively. However, the Laplacian kernel has a sharper peak than the RBF kernel, which means that the Laplacian kernel assigns high similarity to nearby points and low similarity to distant points.

$$w_{i,j} = \exp\left(-\frac{|d_{i,j}|}{\sigma^2}\right), \tag{7}$$

where $d_{i,j}$ is the geographical distance between sensors i and j, and σ is a hyperparameter.

Table VII gives a comparison of the AUROC scores of different kernels. The Poly0 and Poly2 in the table refer to the polynomial kernel with degrees $d = 0$ and $d = 2$, respectively.

When setting the degree to 0 (Poly0), the similarity score between each pair of sensors is one regardless of the distance. Such a naïve setting gives poor AUROC scores for both the Graph WaveNet and STGCN. The Poly2 setting produces unsatisfactory results, likely due to the polynomial kernel's inadequacy as a similarity measure for our application. This is primarily because the dot product is not a proper distance function for geographical locations that use latitude and longitude as input features. In contrast, the Laplacian kernel shows promise as its AUROC scores are comparable to those of the

TABLE VI
GLOBAL MODELS VS. LOCAL MODELS IN TERMS OF RECALL-AT-n. FOR THE SAME GNN MODEL, WE HIGHLIGHT THE WINNER BETWEEN THE GLOBAL AND THE LOCAL METHODS USING BOLD FONT.

Model	Type	R@10	R@20	R@30	R@40	R@50
Graph WaveNet	Global	0.190	**0.302**	0.428	0.538	0.622
	Local	**0.214**	0.300	**0.447**	**0.543**	**0.630**
STGCN	Global	0.209	0.314	**0.449**	**0.594**	**0.665**
	Local	**0.230**	**0.322**	0.428	0.550	0.642

TABLE VII
A COMPARISON OF THE AUROC SCORES OF DIFFERENT KERNEL FUNCTIONS.

Model	RBF kernel	Poly0	Poly2	Laplacian
Graph WaveNet	**0.7216**	0.600	0.620	0.7213
STGCN	**0.7220**	0.599	0.616	0.7176

RBF kernel, likely because the two kernel functions share similar formulas.

H. The Influence of the Number of Neighbors for the Local Models

TABLE VIII
AUROC SCORES OF THE LOCAL MODELS USING DIFFERENT NUMBERS OF NEIGHBORS.

	3	4	5	6	7
Graph WaveNet (local)	0.7242	0.7254	0.7260	0.7257	0.7262
STGCN (local)	0.7208	0.7210	0.7214	0.7215	0.7207

Table VIII shows the influence of the number of neighbors on the local models. We tested 3, 4, 5, 6, and 7 neighbors. The outputted AUROC scores are very similar. Eventually, we fix the number to 5 for the local models in other experiments for simplicity.

V. DISCUSSION AND FUTURE WORK

This paper leverages a two-stage framework to discover the problematic sensors in a nationwide sensor network containing $10000+$ sensors. In the first stage, we use deep spatiotemporal graph models to predict the future monitored PM2.5 values of each sensor. In the second stage, we claim that the k sensors that have the lowest R^2 scores among the monitored and predicted values are problematic sensors. Experimental results obtained on a real dataset show that GNN-based models outperform the baselines in terms of their AUROC, precision-at-n, and recall-at-n scores. We also experiment by training the GNN models with local methods in which the sensors only communicate with their neighboring sensors. The empirical results show that the local methods' predictions are comparable to those of the global methods, suggesting that we can use the local method to achieve more efficient training.

Currently, all our GNN models are applied on an undirected graph, which implies that two sensors have the same degree of influence on each other. This may not necessarily be true in real scenarios. For example, wind direction may be a key factor in determining the propagation flow of air pollution. Therefore, one of our future research directions will be to apply GNNs to a directed graph so that the models can capture these directional factors. We would also like to include more geographical or atmospheric features, such as relative humidity, wind direction, landform measurements, altitude, and time of day, to make the obtained predictions more accurate, as including these factors may yield better results [56].

Moreover, as the sensor deployment in Taiwan is not uniformly distributed, a key drawback of the proposed local models is that the nearest sensors for an urban sensor may still be spatially distant. Consequently, accurately diagnosing the status of the sensors in southeast Taiwan (as depicted in Figure 1) presents a challenge. As a future direction, we plan to extend our local models to address the sparsity of neighboring sensors in such areas.

Finally, since we leverage neighbors' information to infer a sensor's status, if a sensor's neighbors are faulty, this sensor may be falsely identified as faulty, which can affect the accuracy of our algorithm. Essentially, this issue is not unique to graph-based algorithms; it is a common challenge in all data-driven approaches because the accuracy of any algorithm ultimately depends on the quality of the input data. One possible remedy is to adjust the influence score matrix to account for the issue. For instance, if a large percentage of sensors are out of order in a small region, we might be able to adjust the matrix to give less weight to the neighbors of these sensors, reducing the impact of the faulty sensors on our algorithm's accuracy. Nonetheless, this solution may not be optimal, and the performance of our algorithm will still depend on the quality of the input data. As the saying goes, "garbage in, garbage out." Therefore, it's essential to ensure the quality of the input data to improve the algorithm's accuracy.

ACKNOWLEDGMENTS

This work was supported in part by the National Science and Technology Council in Taiwan under grant numbers 110-2222-E-008-005- MY3 and 110-2634-F-008-008.

REFERENCES

[1] S. Masoud, N. Mariscal, Y. Huang, and M. Zhu, "A sensor-based data driven framework to investigate pm2.5 in the greater detroit area," *IEEE Sensors Journal*, 2021.
[2] A. Polidori, V. Papapostolou, and H. Zhang, "Laboratory evaluation of low-cost air quality sensors," *South Coast Air Quality Management District: Diamondbar, CA, USA*, 2016.
[3] S. K. Jha, M. Kumar, V. Arora, S. N. Tripathi, V. M. Motghare, A. Shingare, K. A. Rajput, and S. Kamble, "Domain adaptation based deep calibration of low-cost pm2.5 sensors," *IEEE Sensors Journal*, 2021.
[4] T.-H. Lin, X.-R. Zhang, C.-P. Chen, J.-H. Chen, and H.-H. Chen, "Learning to identify malfunctioning sensors in a large-scale sensor network," *IEEE Sensors Journal*, pp. 1–1, 2021.
[5] L.-J. Chen, Y.-H. Ho, H.-H. Hsieh, S.-T. Huang, H.-C. Lee, and S. Mahajan, "ADF: An anomaly detection framework for large-scale pm2.5 sensing systems," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 559–570, 2018.

[6] E. S. Cross, L. R. Williams, D. K. Lewis, G. R. Magoon, T. B. Onasch, M. L. Kaminsky, D. R. Worsnop, and J. T. Jayne, "Use of electrochemical sensors for measurement of air pollution: correcting interference response and validating measurements," *Atmospheric Measurement Techniques*, vol. 10, no. 9, pp. 3575–3588, 2017.

[7] L. Yang, Y. Lu, S. X. Yang, Y. Zhong, T. Guo, and Z. Liang, "An evolutionary game-based secure clustering protocol with fuzzy trust evaluation and outlier detection for wireless sensor networks," *IEEE Sensors Journal*, vol. 21, no. 12, pp. 13 935–13 947, 2021.

[8] V. Chatzigiannakis and S. Papavassiliou, "Diagnosing anomalies and identifying faulty nodes in sensor networks," *IEEE Sensors Journal*, vol. 7, no. 5, pp. 637–645, 2007.

[9] O. Ghorbel, W. Ayedi, H. Snoussi, and M. Abid, "Fast and efficient outlier detection method in wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 6, pp. 3403–3411, 2015.

[10] L. Spinelle, M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola, "Field calibration of a cluster of low-cost available sensors for air quality monitoring. part a: Ozone and nitrogen dioxide," *Sensors and Actuators B: Chemical*, vol. 215, pp. 249–257, 2015.

[11] A. Bigi, M. Mueller, S. K. Grange, G. Ghermandi, and C. Hueglin, "Performance of no, no 2 low cost sensors and three calibration approaches within a real world application," *Atmospheric Measurement Techniques*, vol. 11, no. 6, pp. 3717–3735, 2018.

[12] T. Zheng, M. H. Bergin, R. Sutaria, S. N. Tripathi, R. Caldow, and D. E. Carlson, "Gaussian process regression model for dynamically calibrating and surveilling a wireless low-cost particulate matter sensor network in Delhi," *Atmospheric Measurement Techniques*, vol. 12, no. 9, pp. 5161–5181, 2019.

[13] E. Esposito, S. De Vito, M. Salvato, V. Bright, R. Jones, and O. Popoola, "Dynamic neural network architectures for on field stochastic calibration of indicative low cost air quality sensing systems," *Sensors and Actuators B: Chemical*, vol. 231, pp. 701–713, 2016.

[14] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 5, 2021, pp. 4027–4035.

[15] A. Protogerou, S. Papadopoulos, A. Drosou, D. Tzovaras, and I. Refanidis, "A graph neural network method for distributed anomaly detection in IoT," *Evolving Systems*, vol. 12, pp. 19–36, 2021.

[16] H. E. Egilmez and A. Ortega, "Spectral anomaly detection using graph-based filtering for wireless sensor networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1085–1089.

[17] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I 25*. Springer, 2018, pp. 362–373.

[18] A. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly detection for IoT time-series data: a survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6481–6494, 2019.

[19] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[20] A. Chaudhary, H. Mittal, and A. Arora, "Anomaly detection using graph neural networks," in *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE, 2019, pp. 346–350.

[21] J. Tryner, C. L'Orange, J. Mehaffy, D. Miller-Lionberg, J. C. Hofstetter, A. Wilson, and J. Volckens, "Laboratory evaluation of low-cost purpleair pm monitors and in-field correction using co-located portable filter samplers," *Atmospheric Environment*, vol. 220, p. 117067, 2020.

[22] H. Shi, Y. Li, H. Cao, X. Zhou, C. Zhang, and V. Kostakos, "Semantics-aware hidden markov model for human mobility," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[23] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 468–478, 2019.

[24] K. Zheng, Y. Zhao, D. Lian, B. Zheng, G. Liu, and X. Zhou, "Reference-based framework for spatio-temporal trajectory compression and query processing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 11, pp. 2227–2240, 2019.

[25] C.-J. Hsu and H.-H. Chen, "Taxi demand prediction based on LSTM with residuals and multi-head attention," in *Proceedings of the 6th International Conference on Vehicle Technology and Intelligent Transport Systems*, 2020, pp. 268–275.

[26] P.-Y. Ting, T. Wada, Y.-L. Chiu, M.-T. Sun, K. Sakai, W.-S. Ku, A. A.-K. Jeng, and J.-S. Hwu, "Freeway travel time prediction using deep hybrid model–taking sun yat-sen freeway as an example," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8257–8266, 2020.

[27] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles, "Collabseer: a search engine for collaboration discovery," in *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, 2011, pp. 231–240.

[28] H.-H. Chen, P. Treeratpituk, P. Mitra, and C. L. Giles, "Csseer: an expert recommendation system based on citeseerx," in *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, 2013, pp. 381–382.

[29] H.-H. Chen and C. L. Giles, "Ascos: an asymmetric network structure context similarity measure," in *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*. IEEE, 2013, pp. 442–449.

[30] ——, "Ascos++ an asymmetric similarity measure for weighted networks to address the problem of simrank," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 2, pp. 1–26, 2015.

[31] P. Chen and H.-H. Chen, "Accelerating matrix factorization by overparameterization." in *DeLTA*, 2020, pp. 89–97.

[32] H.-H. Chen and P. Chen, "Differentiating regularization weights–a simple mechanism to alleviate cold start in recommender systems," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 1, pp. 1–22, 2019.

[33] H.-H. Chen, C.-A. Chung, H.-C. Huang, and W. Tsui, "Common pitfalls in training and evaluating recommender systems," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 37–45, 2017.

[34] H.-H. Chen, L. Gou, X. L. Zhang, and C. L. Giles, "Towards the discovery of diseases related by genes using vertex similarity measures," in *2013 IEEE International Conference on Healthcare Informatics*. IEEE, 2013, pp. 505–510.

[35] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[36] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[37] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv: Learning*, 2018.

[38] T. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *IJCAI*, 2018.

[39] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *IJCAI*, 2019.

[40] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[41] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 424–14 432.

[42] J. Chen, X. Wang, and X. Xu, "Gc-lstm: Graph convolution embedded lstm for dynamic network link prediction," *Applied Intelligence*, pp. 1–16, 2022.

[43] O. Boyaci, M. R. Narimani, K. R. Davis, M. Ismail, T. J. Overbye, and E. Serpedin, "Joint detection and localization of stealth false data injection attacks in smart grids using graph neural networks," *IEEE Transactions on Smart Grid*, vol. 13, no. 1, pp. 807–819, 2021.

[44] M. Cellura, G. Cirrincione, A. Marvuglia, and A. Miraoui, "Wind speed spatial estimation for energy planning in sicily: A neural kriging application," *Renewable energy*, vol. 33, no. 6, pp. 1251–1266, 2008.

[45] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 257–266.

[46] T. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, pp. 832–844, 1998.

[47] ——, "A data complexity analysis of comparative advantages of decision forest constructors," *Pattern Analysis & Applications*, vol. 5, pp. 102–112, 2002.

[48] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the royal statistical society series b-methodological*, vol. 58, pp. 267–288, 1996.

[49] L. Breiman, "Better subset regression using the nonnegative garrote," *Technometrics*, vol. 37, pp. 373–384, 1995.

[50] A. E. Hoerl and R. Kennard, "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, pp. 80–86, 2000.
[51] M. H. Gruber, *Improving efficiency by shrinkage: the James-Stein and ridge regression estimators*. Routledge, 2017.
[52] Y. Chauvin and D. E. Rumelhart, *Backpropagation: theory, architectures, and applications*. Psychology press, 2013.
[53] D. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.
[54] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.
[55] S. Greenland, S. J. Senn, K. J. Rothman, J. B. Carlin, C. Poole, S. N. Goodman, and D. G. Altman, "Statistical tests, p values, confidence intervals, and power: a guide to misinterpretations," *European journal of epidemiology*, vol. 31, pp. 337–350, 2016.
[56] L. Stankovic, D. P. Mandic, M. Dakovic, I. Kisil, E. Sejdic, and A. G. Constantinides, "Understanding the basis of graph signal processing via an intuitive example-driven approach [lecture notes]," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 133–145, 2019.

Dennis Y. Wu Dennis Y. Wu is a graduate student pursuing his Master of Science degree in computer science. He received his Bachelor of Science (B. Sc) degree in the field of computer science and information engineering. His research interests lie in deep learning and learning theory.

Tzu-Heng Lin 's research interests include machine learning, deep learning, and data science. He obtained a Master's degree from the Department of Computer Science and Information Engineering in National Central Unversity in 2021. Tzu-Heng's GitHub: `https://github.com/NeilLin1117`.

Xin-Ru Zhang obtained a Master's degree and a Bachelor's degree from the Department of Computer Science and Information Engineering at National Central University in 2021 and 2019, respectively. She is interested in various machine learning topics such as deep learning and data analysis.

Chia-Pan Chen is an Environmental Management Senior Engineer at the Industrial Technology Research Institute. He is the Project Manager for the Taiwan Air Wide Array Network project, mainly in controlling and monitoring the progress of sensor deployment nationwide, data quality testing and verification and checking with third party. He is also the Principal Investigator for National Industrial Park Mutual Aid Plan, mainly in preventing disaster and assisting with chemical hazard emergency response and rescue. He received a MasterâĂŹs degree and BachelorâĂŹs degree from the Institute of Civil and Hydraulic Engineering, Feng Chia University and Department of Civil Engineering, Feng Chia University, respectively.

Jia-Huei Chen is an Associate Research Fellow at the Industrial Technology Research Institute and an Associate Researcher for the Taiwan Air Wide Array Network project, in which she applies spatial statistics, machine learning, and IoT technology to assist in air quality data analysis. She is a Ph.D. Candidate at National Taiwan Normal University. She was a GIS analyst for an environmental consulting company, mainly in designing the geography information system, analyzing spatial data and managing project progress. She obtained a MasterâĂŹs degree and BachelorâĂŹs degree both from the Department of Geography, National Taiwan Normal University.

Hung-Hsuan Chen is an Associate Professor at the Department of Computer Science and Information Engineering at National Central University (NCU). He is interested in data-related research topics such as machine learning, information retrieval, and graph analysis. He is also interested in applying these techniques to various application domains such as AIoT, recommender systems, and social networks. He was a researcher at the Industrial Technology Research Institute. He obtained his Ph.D. degree from the Pennsylvania State University in 2013.