Explaining Graph Neural Networks with Topology-Aware Node Selection: Application in Air Quality Inference

Esther Rodrigo Bonet, Tien Huu Do, Xuening Qin, *Graduate Student Member, IEEE*, Jelle Hofman, Valerio Panzica La Manna, Wilfried Philips, *Senior Member, IEEE*, and Nikos Deligiannis, *Member, IEEE*

***Abstract*—Graph neural networks (GNNs) have proven their ability in modelling graph-structured data in diverse domains, including natural language processing and computer vision. However, like other deep learning models, the lack of explainability is becoming a major drawback for GNNs, especially in health-related applications such as air pollution estimation, where a model's predictions might directly affect humans' health and habits. In this paper, we present a novel post-hoc explainability framework for GNN-based models. More concretely, we propose a novel topology-aware kernelised node selection method, which we apply over the graph structural and air pollution information. Thanks to the proposed model, we are able to effectively capture the graph topology and, for a certain graph node, infer its most relevant nodes. Additionally, we propose a novel *topological* node embedding for each node, capturing in a vector-shape the graph walks with respect to every other graph node. To prove the effectiveness of our explanation method, we include commonly employed evaluation metrics as well as fidelity, sparsity and contrastivity, and adapt them to evaluate explainability on a regression task. Extensive experiments on two real-world air pollution data sets demonstrate and visually show the effectiveness of the proposed method.**

***Index Terms*—Explainable deep learning, graph convolutional neural networks, geometric deep learning, air pollution.**

I. Introduction

RECENT years have witnessed deep neural networks (DNNs) achieving state-of-the-art performance in highly complex problems within various application domains such as natural language understanding and computational biology [1]. However, real-world data (e.g., social media data, IoT data) often exhibits a graph structure and can be represented by means of graphs [2], [3]. While conventional DNNs neglect this fact, graph neural networks (GNNs) have been proposed [4], [5] to capture this graph structure. Thanks to their capability in learning representations from graph-structured data, GNNs have proven a powerful architecture for various tasks with applications in bioinformatics, computer vision, natural language processing, recommendation systems and traffic forecasting, to name a few [2], [6].

Similarly, DNNs and GNNs have been successfully applied to predict unknown pollutant concentrations. Air pollution estimation based on deep learning aims at inferring or forecasting unknown air pollution levels given historical data. Inference is carried out by leveraging the input features and exploiting the spatio-temporal correlation of the data [2], [7], [8], [9]. This is mainly done by imposing constrains in the loss function and by incorporating other data types such as meteorological [8], [10] or traffic [9] information.

Nevertheless, one well-known limitation of DNNs and GNNs is their inability to provide the rationale for their decisions. This is due to the fact that DNNs comprise many non-linear transformations, which combine input features and parameters into activations. Eventually, a decision is made based on the final outputs of the DNNs, leading to a difficulty in tracing how particular input data drive a decision [11]. Hence, these models are usually considered *black-box* models. In many critical applications (e.g., healthcare and autonomous driving), however, *explainability* is a *must* in order to increase confidence, trust, transparency, and safety. The understanding of the inner workings of DNNs will allow experts and non-experts to better make use of the models' predictions. Whereas explanation techniques have been developed for non-graph-based deep models, the task remains challenging in graph-based models since they must additionally consider the topology of the underlying graphs. Most works have tackled the problem by adapting existing explainability methods for DNNs to GNNs, e.g., GNN-LRP [12] or Grad-CAM [13], or by designing post-hoc techniques such as feature selection, e.g., GraphLIME [14] or GNNExplainer [15].

Existing works in explainable air pollution inference aim at explaining the decisions taken by DNNs but not GNNs. These studies mainly focus on computing the relevancy weights of input features. For instance, [16], [17] perform feature selection on the input, mostly by incorporating an additional layer to the deep network. However, these approaches generally ignore the inherent topology of the air quality data.

To tackle the aforementioned problem, in this study, we propose a novel explainability method for air pollution inference, referred to as *NodeSel*. NodeSel follows the white-box explanation approach by deploying the interpretable HSIC Lasso model to obtain the most relevant nodes for a certain node's prediction. We then apply the proposed method to explain a GNN model—namely, the AVGAE model [2], [18]—in the context of air pollution prediction. Specifically, given

E. Rodrigo Bonet, T. H. Do, and N. Deligiannis are with the Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, 1050 Brussels, Belgium, and also with imec, 3001 Leuven, Belgium (e-mail: erodrigo@etrovub.be; thdo@etrovub.be; ndeligia@etrovub.be).

X. Qin and W. Philips are with the Department of Telecommunications and Information Processing, Ghent University, 9000 Ghent, Belgium, and also with imec, 3001 Leuven, Belgium.

J. Hofman and V. Panzica La Manna are with imec, 5656 Eindhoven, The Netherlands.

the estimated pollutant concentrations provided by the AVGAE model and the graph topology, our method aims at identifying the most influential graph nodes of a certain node. As a result, the estimated concentration at the selected node can be explained by looking at the concentrations and locations of its most influential nodes. Although our method is evaluated using air pollution data, our formulation is general and can be applied to any graph-structured data set.

To summarise, the contributions of this study are three-fold:

- We propose a novel post-hoc explainability method for GNN-based models on graph-structured data. The method identifies the most influential nodes of a given node, thereby explaining the node's prediction.
- We define a novel *topological* node embedding that is able to capture the global graph topology from a certain node's perspective. The topological node embeddings are then used in the formulation of the Hilbert-Schmidt Independence Criterion (HSIC) *least absolute shrinkage and selection operator* (Lasso) model.
- We present comprehensive experiments on two real-world air pollution data sets to evaluate the effectiveness of our method in explaining the predicted pollutant concentrations. Experimental results show that the model is able to maintain high-quality predictions while visually explaining the underlying graph data structure.

The remainder of the paper is organised as follows. Section II reviews related work on air pollution inference and explainability. Section III introduces our notation, describes the AVGAE inference model, and formulates our GNN explanation problem. Section IV presents the proposed method in detail and Section V describes our experimentation. Finally, Section VI draws the conclusion and discusses future work.

II. Related Work

Our work lies in the intersection of air pollution inference with GNNs and post-hoc explanations. In this section, we review these two topics and discuss the differences of our method with respect to existing ones. Section II-A introduces the problem of air pollution inference, focusing on GNN models. Section II-B elaborates on explainable DNN and GNN models and Section II-C reviews the prior art in explainable air pollution inference models.

A. Air Pollution Inference

Air pollution prediction aims at inferring or forecasting unknown air pollution levels, given historical data or related urban data such as temperature, wind flow, traffic, etc. Graph- and non-graph-based deep learning models have been applied to air quality estimation [19]. Apart from these *data-driven* approaches [7], [8], [19], [20], [21], a number of methods following a *deterministic* [22], [23] or *statistical* [23], [24] approach have been proposed. *Deterministic* approaches leverage fluid motion equations to model particles' shift in time. These models are computationally expensive and lack efficiency since they are based on numerical approximations. Moreover, they require knowledge of meteorological data or the distribution of pollution sources for parameter identification. Similarly, *statistical* models often make a linearity assumption which hinders their forecasting performance [20]. Recent advances in deep learning have led to DNN and GNN models surpassing deterministic and statistical methods in air pollution estimation [19]. Generally, estimating unmeasured pollutant concentrations with DNNs is carried out by exploiting the spatio-temporal correlations of the data [2], [7], [8], [9]. This is achieved by imposing constrains on the loss function or by incorporating other data types such as meteorological [8], [10] or traffic [9] information.

B. Explainability Methods for Deep Models

Recent studies have addressed the explainability shortcoming of deep learning models using different approaches. One approach investigates to which extent an input feature is responsible for a decision, either by studying the back-propagation of the gradients of the target prediction with respect to input features [25], [26], [27] or by perturbing the input samples [28], [29], [30]. Other methods seek to explain DNNs with *white-box* machine learning models (a.k.a., model distillation) [31], [32]. These white-box models are designed to be interpretable and trained to mimic the behaviour of the DNNs that are to be explained. A third approach attempts to provide the explanation as part of the model's output. Methods following this approach either use attention mechanisms [33], [34] or employ an additional explanation task to be trained jointly with the original task [35], [36].

While several explanation techniques have been developed for non-graph-based deep models, explainability remains a challenging task for graph-based models since these models additionally consider the topology of underlying graphs. Different methods have attempted to address this by taking into account information not only from the feature space but also from the graph structure, such as node [15], [37] and edge [12], [38] importance. A popular approach for explaining GNNs focuses on adapting existing explainability methods for DNNs to GNNs. To identify the importance of input features, these models look at the *gradients* or *hidden feature maps*. For instance, in the Guided Backpropagation (GB) [39] and Sensitivity Analysis (SA) [40] methods, the gradients of a class-specific prediction score (i.e., the score before the *softmax* function) versus the input features when freezing the learned model parameters are regarded as input importance scores. Despite their simplicity and efficiency, these methods suffer from the saturation problem; namely, the model output hardly changes in response to input changes in the saturated region [27]. Alternatively, an importance heat-map of the input features is generated in CAM [41] by mapping the final node embeddings to the input feature space. Grad-CAM [41] extends CAM by using gradients as weights in the linear combination of hidden feature maps to generate the importance heat-map. These methods are simple and effective; however, their assumption that node embeddings can reveal the importance of input features is heuristic [42]. Another group of explanation methods focuses on distributing final prediction scores to the input space; these are referred to as *decomposition-based* methods. Representative decomposition-based methods are Layer-wise Relevance Propagation (LRP) [43], Excitation-Backpropagation (EB) [41], and GNN-LRP [12]. Other works follow a *perturbation-based* approach [15], [38], which performs variations in the input layer to measure their effects on the output. Intuitively, if most of the relevant features are retained, the final prediction should only change marginally. These perturbations are often implemented via learnable mask matrices. For instance, GNNExplainer [15] learns a binary mask over the node features or graph edges and PGExplainer [38] learns an approximate discrete mask only over edges. Different from these approaches, there exist works that incorporate an interpretable model into a pre-existing GNN; the interpretable model is then used to explain the GNN model. This approach is similar to the white-box explanation approach mentioned earlier (see the previous paragraph). For example, GraphLIME [14] selects the most relevant features of the nodes of a graph by employing a nonlinear HSIC Lasso model. The method then uses the weights generated by the HSIC Lasso model for different features to select important input features, which are regarded as the explanation for the considered GNN. Similarly, our NodeSel model follows the white-box explanation approach by employing the HSIC Lasso model to explain an arbitrary GNN. This choice clearly differentiates our method from the gradient-based, feature-based, decomposition-based and perturbation-based approaches. Our method is similar to the GraphLIME model [14] in that both methods are based on HSIC Lasso. However, unlike GraphLIME, which provides the explanation on the feature level, our formulation focuses on explaining the interaction between nodes of the underlying graph, which is crucial in GNNs. In this regard, our work attempts to aggregate and interpret the joint effect of node interactions. Furthermore, GraphLIME selects the local vicinity by *n-hop* jumps, thereby foregoing important information from higher-order nodes. We, on the other hand, argue that collecting the subset of relevant nodes uniquely from the nearest *n-hop* vicinity might not provide sufficient support to explain a node's prediction. Hence, we mitigate this issue by considering a better choice in the vicinity selection process by proposing a novel topological node embedding.

C. Explainable Air Pollution Inference

In air pollution estimation using DNNs, limited work has addressed the problem of model explainability. While DNNs, including GNNs, are able to effectively address the problem of location-dependent prediction [2], [18], [44], [45], these models generally do not provide the rationale behind their decisions. Some authors [16], [46] have developed feature selection techniques, such as Lasso [47] and Group Lasso [48]. Other research proposed additional layers to the deep model for feature selection. For instance, Qi *et al.* [16] perform feature selection on the input layer by incorporating a sparsity Kullback-Leibler (KL) divergence constraint on an additional network layer. Similarly, Cheng *et al.* [17] attempt to learn the feature weights from collected data at monitoring stations by adding an attention-based pooling layer. Such approaches could be considered explainable in that they give a relevancy weighting of the input features. However, they generally ignore the inherent topology of the air quality data. It is noteworthy that the air quality data is commonly collected using a network of monitoring stations or sensors, hence this data type is location-dependent and has an underlying graph structure. As a result, the interaction between stations or sensors is essential in the modelling of air quality data.

III. Problem Formulation

In this section, we formulate the graph neural network (GNN) post-hoc explanation task. We first introduce our notation (Section III-A) and then briefly present our recent GNN model for air quality inference [2], [18] (Section III-B), which serves as the model to exemplify our post-hoc explanation method (Section III-C). It is worth noting that the proposed explanation method is not tailored to our prior GNN model but instead, it can be used to explain the impact of the graph structure in the decision of different GNN models.

A. Notation

In this work, we consider *row*-vectors and denote them by boldface lowercase letters. Matrices are denoted by boldface capital letters. Subscripts refer to elements of a certain vector or matrix; for instance, $\boldsymbol{x}_i$ is the i-th element of the vector $\boldsymbol{x}$ and $\boldsymbol{F}_{i,j}$ is the element in the i-th row and j-th column of the matrix $\boldsymbol{F}$. Superscripts in specific vectors (e.g., $\boldsymbol{\beta}^{(v)}$) or matrices (e.g., $\boldsymbol{F}^{(v)}$) refer to that these computations have been performed with respect to a certain graph node v. The overline notation over a matrix or vector represents the normalised centered version of the original. Lastly, constants are denoted by small Greek letters in regular font and sets of elements are denoted by capital calligraphic fonts (e.g., $\mathcal{V}$).

B. The Graph Neural Network Model

The Variational Graph Autoencoder (AVGAE), presented in [2], [18], aims at inferring missing air quality data from a set of known measurements. It employs a graph-based encoder-decoder approach that leverages the spatio-temporal correlation in air quality data. Let us consider a data set of measurements of pollutant concentrations in a specific city. Each measurement is expressed as $\{m, \boldsymbol{s}, t\}$, where m denotes the pollutant concentration, reported in terms of *micrograms per cubic meter* ($\mu g/m^3$) or *parts per billion* (ppb), $\boldsymbol{s}$ indicates the location (in terms of latitude and longitude) where the measurement was collected and t is the corresponding time instant. We aggregate the measurements at discrete time instants and locations; hence, the considered time interval of interest is divided into uniform slots of duration τ, obtaining a set of T timeslots $\{t_1, \ldots, t_T\}$. Similarly, the road network of the considered city is divided into a set of N points $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\}$. The spatial aggregation is adapted to the road network and the considered locations in it, meaning that we consider a non-uniform aggregation across space. Formally, for a certain time instant t_j, the measurements collected within a predefined distance r from $\boldsymbol{p}_i$ are averaged. Hence, the set of the considered locations $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\}$ and the timeslot duration τ determine the spatial and temporal resolution of the model, respectively. The aggregation process results in a matrix of measurements $\boldsymbol{M} \in \mathbb{R}^{N \times T}$, where N and T are the number of considered locations and time instants. Accordingly, an entry $\boldsymbol{M}_{i,j}$ with $i = 1, ..., N, j = 1, \ldots, T$, corresponds to the (averaged) measurement at location $\boldsymbol{p}_i$ and timeslot t_j.

The AVGAE model [2], [18] estimates the missing entries in $\boldsymbol{M}$ by solving a matrix completion on graphs problem. Each row of matrix $\boldsymbol{M}$ is associated with a location; hence, entries of $\boldsymbol{M}$ corresponding to nearby locations should have similar values within the same time instant. The AVGAE model considers the spatial correlation between matrix entries by constructing an undirected weighted graph based on the road network topology. The graph is built by computing the geodesic distances between the N discretised locations of the road network. Two graph nodes (i.e., locations) are connected if their distance is smaller than a predefined threshold δ or if they belong to the same street segment. The AVGAE model follows an encoder-decoder architecture over the constructed graph. The model inputs the highly-incomplete matrix $\boldsymbol{M}$ concatenated with the matrix $\boldsymbol{S}$ of geo-coordinates of the street locations. The encoder step is built of graph convolutional layers and outputs two matrices each containing the mean and standard deviation values of a multivariate Gaussian distribution. These matrices are then forwarded to the decoder step, which also consists of graph convolutional layers. In this step, the model approximates the known entries in the matrix $\boldsymbol{M}$ and infers the missing ones to finally recover the reconstructed matrix of measurements $\tilde{\boldsymbol{M}}$.

We briefly discuss the architecture of the AVGAE model [2], [18] in what follows. To infer the unknown pollutant concentrations in $\boldsymbol{M}$ and generate $\tilde{\boldsymbol{M}}$, the AVGAE model first learns a latent representation of the input data, which is denoted as $\boldsymbol{Z}$. In particular, the AVGAE assumes that $p(\boldsymbol{Z}) = \mathcal{N}(0, I)$ and $q(\boldsymbol{Z}|\boldsymbol{M}, \boldsymbol{S}, \boldsymbol{A}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\eta})$; where $\boldsymbol{M}$ is the matrix of known pollutant concentrations, $\boldsymbol{S}$ is the matrix of geo-coordinates of nodes (i.e., locations) and $\boldsymbol{A}$ denotes the weighted adjacency matrix of the considered graph. Simultaneously, the AVGAE model learns the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ and the generative process to produce $\tilde{\boldsymbol{M}}$. The former two parameters are learnt through two separate neural network branches, that is, $\boldsymbol{\mu} = f_\mu(\boldsymbol{M}, \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{\Theta_1})$ and $\boldsymbol{\eta} = g_\eta(\boldsymbol{M}, \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{\Theta_2})$, parameterised by $\boldsymbol{\Theta_1}$ and $\boldsymbol{\Theta_2}$, respectively. Concretely, to incorporate the graph knowledge into the model, functions f_μ and g_η are designed to be stacked graph convolutional layers (GCNs). The final generative process is described by another stack of GCN layers parameterised by $\boldsymbol{\Phi}$. The AVGAE model [2], [18] is described by the following set of operations:

$$\boldsymbol{\mu} = \text{GCN}_\mu(\boldsymbol{M}, \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{\Theta_1}) \tag{1}$$

$$\boldsymbol{\eta} = \text{GCN}_\eta(\boldsymbol{M}, \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{\Theta_2}) \tag{2}$$

$$\boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\eta}) \tag{3}$$

$$\tilde{\boldsymbol{M}} = \text{GCN}_z(\boldsymbol{Z}, \boldsymbol{A}, \boldsymbol{\Phi}) \tag{4}$$

In the previous equations, the functions GCN_μ, GCN_η and GCN_z are composed by stacking GCN layers, $\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2}$ and $\boldsymbol{\Phi}$ are parameters that are learned from the data, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\eta})$ represents a Gaussian distribution with mean and standard deviation. The matrix of geo-coordinates of nodes locations $\boldsymbol{S}$ is horizontally concatenated with $\boldsymbol{M}$. The AVGAE model utilises two separate branches for training $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$, thereby allowing to select proper activation functions for $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$.

C. Explainability Problem Formulation

In this work, we focus on explaining air quality inference at the street level of urban areas. Our key insight is that the graph structure, as seen by a certain node v, partially determines the information the GNN uses to generate the prediction of node v at a certain time instant. In particular, the GNN's aggregation mechanism mathematically defines how to compute the node embedding and directly influences node v's prediction.

Let $G = \{\mathcal{V}, \mathcal{E}, \boldsymbol{X}\}$ denote the graph structure, where G is an undirected and weighted graph, $\mathcal{V}$ is the set of graph nodes, $\mathcal{E}$ is the set of graph edges, and $\boldsymbol{X}$ is the associated *topology* matrix (computed in Section IV-B). Since each graph node represents a street location, we have $|\mathcal{V}| = N$. Each node in the graph has an associated *topological* node embedding $\boldsymbol{x}^{(i)} \in \mathbb{R}^D$ with D the predefined vector dimensionality, which mainly captures the observations of the graph structure as seen from the i-th node's perspective. Hence, all the *topological* node embeddings $\boldsymbol{x}^{(i)}$ of nodes in G are gathered in a $N \times D$ matrix, referred to as the associated *topology* matrix $\boldsymbol{X} \in \mathbb{R}^{N \times D}$. Let $v \in \mathcal{V}$ and $f : \mathbb{R}^{N \times T} \to \mathbb{R}^{N \times T}$ be the node and the GNN model to be explained, respectively. Moreover, let $\tilde{\boldsymbol{M}} = f(\boldsymbol{M})$ denote the output of this regression task, that is, the predicted pollutant concentrations for all the possible geo-locations $\boldsymbol{p}$ and timeslots t. Finally, let $\hat{\boldsymbol{y}} \in \mathbb{R}^N$ be vector of predicted pollutant concentrations at a pre-defined time instant t_j (i.e., column j of the reconstructed matrix of measurements $\tilde{\boldsymbol{M}}$) and $\hat{\boldsymbol{y}}^{(v)} \equiv \hat{\boldsymbol{y}}_v \in \mathbb{R}$ to denote the v-th element of vector $\hat{\boldsymbol{y}}$, namely, the predicted pollutant concentration of node v at the selected time instant t_j.

We aim at giving an explanation of node v's prediction $\hat{\boldsymbol{y}}^{(v)}$ by finding a subgraph of the graph structure that inherently contains the knowledge required for node v's prediction. Formally, given a GNN model f and its predictions $f(\boldsymbol{M})$, the time t_j and a node v whose prediction needs to be explained, the whole graph structure mapped to a *topology* matrix $\boldsymbol{X}$, the predefined size of relevant nodes K and our explanation model Ψ, the explanation for node v's prediction is obtained as follows:

$$(\mathcal{K}^{(v)}, \boldsymbol{\beta}^{(v)}) = \arg\min_{\Psi \in \Gamma} \Psi\left(f(\boldsymbol{M}), \boldsymbol{X}, (v, t_j), K\right), \tag{5}$$

where $\mathcal{K}^{(v)}$ is the subset of most relevant nodes and their respective *relevancy weight* with respect to node v is captured in $\boldsymbol{\beta}^{(v)}$. In effect, we employ $\gamma(v) = (\mathcal{K}^{(v)}, \boldsymbol{\beta}^{(v)})$ as the formal explanation of node v, generated based on the optimal explanation model $\Psi \in \Gamma$ from all classes of explanation models Γ.

IV. The Proposed Method

In this section, we describe the proposed *NodeSel* post-hoc explanation model, which is applicable to any trained GNN model (Section IV-A). *NodeSel* can exploit the topology

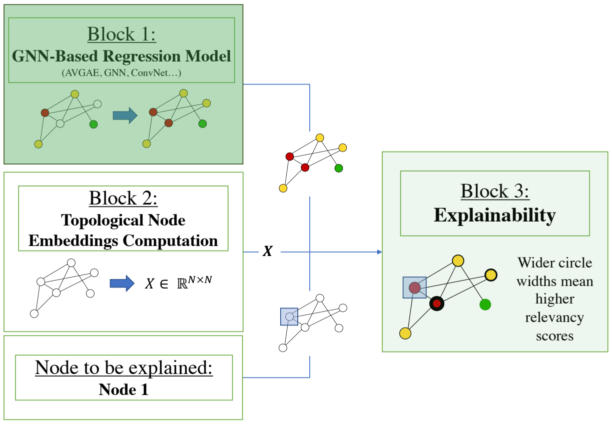

Fig. 1. A block diagram explaining the different elements of our method. Pre-trained blocks are represented in dark green while light green is employed for the blocks to be optimised during training. White blocks show elements of the model with no training required.

of the graph structure and node predictions, and infer the most relevant nodes (i.e., locations) for a particular node's prediction. Additionally, we propose a novel node embedding, which we refer to as the *topological-relevancy* node embedding (Section IV-B). With the proposed node embedding definition, we are able to capture the whole graph structure with respect to a certain node in a meaningful manner.

A. The Proposed GNN Explanation Method

We solve the node regression task by deploying any GNN-based model, in our case, the AVGAE model [2], [18]. We have first constructed the undirected weighted graph based on the road network topology. Then, the matrix of node embeddings $\boldsymbol{X}$ is built (see Section IV-B for a detailed description on this point). After that, the proposed NodeSel method determines per node the subset of nodes that have the most influence on its prediction (a.k.a., predicted pollutant concentration per location). We refer to the subset of nodes for a certain node v as $\mathcal{K}^{(v)}$. We assess the explanations of NodeSel that is, $\mathcal{K}^{(v)}$, $\forall v$, by contrasting the quality of the predictions made by a GNN model—in this case, the AVGAE model—when we train and test the model using either all graph nodes ($\mathcal{V}$) or the set of most influential ones ($\mathcal{K}$). We refer to Fig. 1 for a visual description of the different blocks of the proposed NodeSel architecture.

In essence, given a certain node and time instant for which we seek an explanation for the prediction, we aim at classifying every other node in the graph as relevant or not. In other words, we expect to select a subset $\mathcal{K}$ of graph nodes (corresponding to locations) of size $|\mathcal{K}| = K$, whose information will suffice to correctly predict a node's output. As such, the subset $\mathcal{K} \subset \mathcal{V}$ will contain the most relevant nodes for a certain node's prediction. We opt for a model that can capture non-linear decision boundaries. Motivated by this, we build our model upon the kernelised non-linear supervised selection method, HSIC Lasso [49], which aims at finding a non-redundant vector with a strong non-linear dependency with the output prediction. Lasso selection techniques are commonly used to eliminate redundant features of the node embeddings. Our approach, on the other hand, presents an HSIC Lasso-like optimisation function which efficiently performs a topology-aware selection over the existing graph nodes. Moreover, we perform the topology-aware selection with respect to the specific node we wish to explain. The architecture of the proposed NodeSel design is depicted in Fig. 2. Hence, given a certain node v to be explained, the proposed model minimises the following loss function with respect to $\boldsymbol{\beta}^{(v)} = [\boldsymbol{\beta}_1 \ldots \boldsymbol{\beta}_N] \in \mathbb{R}^N$, namely, the *topological-relevancy row*-vector of node v:

$$\begin{aligned}\min_{\boldsymbol{\beta}^{(v)} \in \mathbb{R}^N} & \frac{1}{2}||\overline{\boldsymbol{L}} - \overline{\boldsymbol{F}} \cdot \operatorname{diag}(\boldsymbol{\beta}^{(v)})||_F^2 \\ & + \frac{1}{2}||\overline{\boldsymbol{L}}^{(v)} - \overline{\boldsymbol{F}}^{(v)} \cdot \operatorname{diag}(\boldsymbol{\beta}^{(v)})||_F^2 \\ & + \rho||\boldsymbol{\beta}^{(v)}||_1 \\ & \text{s.t. } \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N \geq 0, \end{aligned} \tag{6}$$

where N is the number of nodes in the graph, $||\cdot||_F$ is the Frobenius norm, $||\cdot||_1$ is the ℓ_1-norm to enforce sparsity and $\rho \geq 0$ is the regularisation parameter. We define $\operatorname{diag}(\boldsymbol{\beta}^{(v)})$ as the diagonal matrix with the elements of $\boldsymbol{\beta}^{(v)}$ along its main diagonal. To simplify the notation, in what follows, we will refer to $\boldsymbol{\beta}^{(v)}$ simply as $\boldsymbol{\beta}$. $\overline{\boldsymbol{F}}$ and $\overline{\boldsymbol{L}}$ are the centered normalised Gram matrices for the input and predicted output, respectively. With v the selected node to be explained, $\overline{\boldsymbol{F}}^{(v)}$ is the centered normalised Gram matrix which captures the dependency of every graph node with respect to node v. Equivalently, $\overline{\boldsymbol{L}}^{(v)}$ is the centered normalised Gram matrix which captures the dependency between the predicted output of every graph node with respect to v's predicted output. We employ Gaussian kernels to compute the centered normalised Gram matrices as follows. First, we compute the kernelised matrices and vector as:

$$\boldsymbol{F}_{i,j} = \boldsymbol{F}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \exp\left(-\frac{||\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}||_2^2}{2\sigma_x^2}\right), \tag{7}$$

$$\boldsymbol{L}_{i,j} = \boldsymbol{L}(\hat{\boldsymbol{y}}_i, \hat{\boldsymbol{y}}_j) = \exp\left(-\frac{(\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{y}}_j)^2}{2\sigma_y^2}\right), \tag{8}$$

$$\boldsymbol{F}^{(v)}_{i,j} = \boldsymbol{F}^{(v)}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \exp\left(-\frac{(\boldsymbol{x}_i^{(v)} - \boldsymbol{x}_j^{(v)})^2}{2\sigma_{xv}^2}\right), \tag{9}$$

$$\boldsymbol{l}^{(v)}_j = \boldsymbol{l}^{(v)}(\hat{\boldsymbol{y}}_v, \hat{\boldsymbol{y}}_j) = \exp\left(-\frac{(\hat{\boldsymbol{y}}_v - \hat{\boldsymbol{y}}_j)^2}{2\sigma_{yv}^2}\right), \tag{10}$$

where $\boldsymbol{F}_{i,j}, \boldsymbol{L}_{i,j}$ and $\boldsymbol{F}^{(v)}_{i,j}$ are the elements in the i-th row and j-th column of matrices $\boldsymbol{F}, \boldsymbol{L}$ and $\boldsymbol{F}^{(v)} \in \mathbb{R}^{N\times N}$, respectively, $\boldsymbol{l}^{(v)}_j$ is the j-th element of the vector $\boldsymbol{l}^{(v)} \in \mathbb{R}^N$, and $\sigma_x, \sigma_y, \sigma_{xv}$ and σ_{yv} are Gaussian kernel widths which add a normalisation factor between input and output features.

Then, we obtain the normalised centered Gram versions of matrices $\boldsymbol{L}, \boldsymbol{F}$ and $\boldsymbol{F}^{(v)}$, namely, $\overline{\boldsymbol{L}}, \overline{\boldsymbol{F}}$ and $\overline{\boldsymbol{F}}^{(v)} \in \mathbb{R}^{N\times N}$, by applying the following normalisation operations:

$$\overline{\boldsymbol{L}} = \frac{\boldsymbol{HLH}}{||\boldsymbol{HLH}||_F}, \tag{11}$$

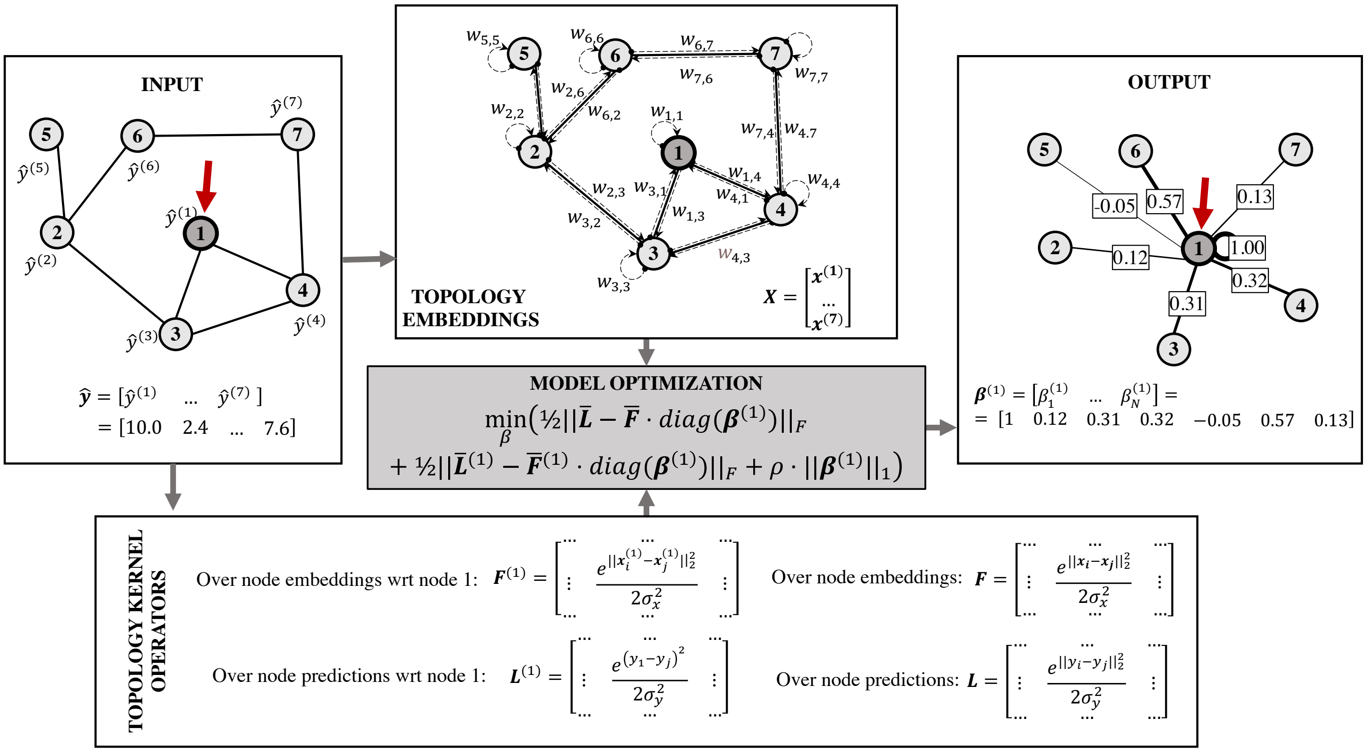
INPUT
5
6
7
2
1
4
3
$\hat{y}^{(5)}$
$\hat{y}^{(6)}$
$\hat{y}^{(7)}$
$\hat{y}^{(1)}$
$\hat{y}^{(2)}$
$\hat{y}^{(4)}$
$\hat{y}^{(3)}$
$\hat{\boldsymbol{y}} = [\hat{y}^{(1)} \;\; \dots \;\; \hat{y}^{(7)}]$
$= [10.0 \;\; 2.4 \;\; \dots \;\; 7.6]$
TOPOLOGY EMBEDDINGS
$w_{5,5}$
$w_{6,6}$
$w_{6,7}$
$w_{7,6}$
$w_{7,7}$
$w_{2,2}$
$w_{2,6}$
$w_{6,2}$
$w_{1,1}$
$w_{7,4}$
$w_{4,7}$
$w_{1,4}$
$w_{4,1}$
$w_{4,4}$
$w_{2,3}$
$w_{3,1}$
$w_{1,3}$
$w_{3,2}$
$w_{4,3}$
$w_{3,3}$
$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \dots \\ \boldsymbol{x}^{(7)} \end{bmatrix}$
MODEL OPTIMIZATION
$\min_{\beta}\big(½||\overline{\boldsymbol{L}} - \overline{\boldsymbol{F}} \cdot diag(\boldsymbol{\beta}^{(1)})||_F$
$+ ½||\overline{\boldsymbol{L}}^{(1)} - \overline{\boldsymbol{F}}^{(1)} \cdot diag(\boldsymbol{\beta}^{(1)})||_F + \rho \cdot ||\boldsymbol{\beta}^{(1)}||_1\big)$
OUTPUT
-0.05
0.57
0.13
0.12
1.00
0.32
0.31
$\boldsymbol{\beta}^{(1)} = [\beta_1^{(1)} \;\; \dots \;\; \beta_N^{(1)}] =$
$= [1 \;\; 0.12 \;\; 0.31 \;\; 0.32 \;\; -0.05 \;\; 0.57 \;\; 0.13]$
TOPOLOGY KERNEL OPERATORS
Over node embeddings wrt node 1: $\boldsymbol{F}^{(1)} = \begin{bmatrix} \dots & \dots & \dots \\ \vdots & \frac{e^{||x_i^{(1)} - x_j^{(1)}||_2^2}}{2\sigma_x^2} & \vdots \\ \dots & \dots & \dots \end{bmatrix}$
Over node embeddings: $\boldsymbol{F} = \begin{bmatrix} \dots & \dots & \dots \\ \vdots & \frac{e^{||x_i - x_j||_2^2}}{2\sigma_x^2} & \vdots \\ \dots & \dots & \dots \end{bmatrix}$
Over node predictions wrt node 1: $\boldsymbol{L}^{(1)} = \begin{bmatrix} \dots & \dots & \dots \\ \vdots & \frac{e^{(y_1 - y_j)^2}}{2\sigma_y^2} & \vdots \\ \dots & \dots & \dots \end{bmatrix}$
Over node predictions: $\boldsymbol{L} = \begin{bmatrix} \dots & \dots & \dots \\ \vdots & \frac{e^{||y_i - y_j||_2^2}}{2\sigma_y^2} & \vdots \\ \dots & \dots & \dots \end{bmatrix}$

Fig. 2. Proposed model architecture (NodeSel) to explain the prediction based on the graph topology of a network. The input of the model consists of the graph and the predicted air pollutant concentrations. In addition, the model is given the node to be explained (pointed out by a red arrow in the input rectangle) and the kernel operators to be employed. The model is able to compute the *topological* node embeddings $\boldsymbol{x}^{(i)}$ and optimise with respect to the *topological-relevancy* vector $\boldsymbol{\beta}$. The model outputs a relevancy score for every node of the graph with respect to the node to be explained, which is collected from the *topological-relevancy* vector $\boldsymbol{\beta}$. In the figure, we show this with line widths representing the strength of connections from node 1 to the other nodes.

$$\overline{\boldsymbol{F}} = \frac{\boldsymbol{HFH}}{||\boldsymbol{HFH}||_F}, \tag{12}$$

$$\overline{\boldsymbol{F}}^{(v)} = \frac{\boldsymbol{HF}^{(v)}\boldsymbol{H}}{||\boldsymbol{HF}^{(v)}\boldsymbol{H}||_F}, \tag{13}$$

where $\boldsymbol{H} = \boldsymbol{I}_N - \frac{1}{N}\boldsymbol{1}_N\boldsymbol{1}_N^T$ is the centering matrix, with $\boldsymbol{I}_N$ and $\boldsymbol{1}_N$ the identity matrix and the all-ones vector, respectively. By left and right multiplication with the symmetric and idempotent matrix $\boldsymbol{H}$, the input matrix becomes a doubly centered matrix whose row and column means are equal to zero.

Note that in (10), the vector $\boldsymbol{l}^{(v)} \in \mathbb{R}^N$. Hence, to compute the centered normalised matrix we perform the following diagonalisation plus normalisation operation:

$$\overline{\boldsymbol{L}}^{(v)} = \frac{\boldsymbol{H}\ \text{diag}(\boldsymbol{l}^{(v)})\boldsymbol{H}}{||\boldsymbol{H}\ \text{diag}(\boldsymbol{l}^{(v)})\boldsymbol{H}||_F}, \tag{14}$$

where $\overline{\boldsymbol{L}}^{(v)} \in \mathbb{R}^{N\times N}$ becomes a matrix.

In eqs. (7), (8), (9), and (10), the index $v \in \mathcal{V}$ denotes the graph node with respect to which the optimisation is performed, i.e., the node to be explained, and indices $i, j \in \mathcal{V}$ denote graph nodes. $\hat{\boldsymbol{y}}_i, \hat{\boldsymbol{y}}_j$ and $\hat{\boldsymbol{y}}_v \in \mathbb{R}$ respectively correspond to the i-th, j-th and v-th elements (i.e., i-th, j-th and v-th graph nodes) of the vector $\hat{\boldsymbol{y}}$ of the predicted pollutant concentrations. Finally, the vectors $\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)} \in \mathbb{R}^N$ correspond to the *topological* node embeddings of nodes i and j, respectively. Additionally, $\boldsymbol{x}_i^{(v)}, \boldsymbol{x}_j^{(v)}$ respectively correspond to the i-th and j-th elements of the *topological* node embedding $\boldsymbol{x}^{(v)}$. Essentially, a certain *topological* node embedding $\boldsymbol{x}^{(i)}$ is a vector which contains the structural dependency between node i and every other node in the graph. Hence, $\boldsymbol{x}_l^{(i)}$ is the l-th element of vector $\boldsymbol{x}^{(i)}$ and contains the graph dependency score between nodes i and l. The proposed topology embedding is based on the shortest graph path. In our case, the employed graph is undirected; hence, the property $\boldsymbol{x}_l^{(i)} \equiv \boldsymbol{x}_i^{(l)}$ is valid. For now, we assume that the *topological* node embeddings $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ ($\forall i, j \in \mathcal{V}$) are given; we will later discuss how to determine them (see Section IV-B). Finally, the σ-parameters in eqs. (7), (8), (9), and (10) are required to have normalised Gaussian kernels. In that case, the integral $\int_{-\infty}^{\infty} \exp\left(-\frac{x^2}{2\sigma^2}\right)dx = \sqrt{2\pi}\sigma$ must be true. Hence, the parameters σ_y, σ_x, σ_{yv} and σ_{xv} are set so that the area under the curve is always the unit. This allows the elements of matrices $\overline{\boldsymbol{L}}, \overline{\boldsymbol{F}}, \overline{\boldsymbol{L}}^{(v)}$ and $\overline{\boldsymbol{F}}^{(v)}$ to be comparable in value.

The first term in the objective function in (6) can be understood as a minimisation between the predicted output and input data. The second term can be interpreted as the minimisation between the predicted output and input data with respect to the node v to be explained. To be more specific, to optimise with respect to node v in the second term of (6), we only employ *topological* node embeddings and output predictions with respect to that certain node v.

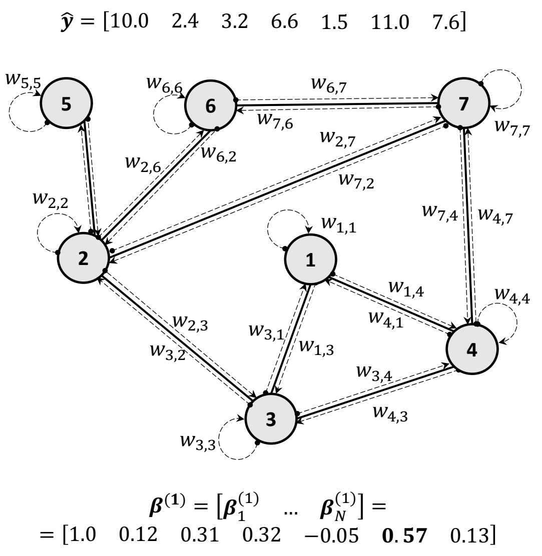
$\hat{\boldsymbol{y}} = [10.0 \quad 2.4 \quad 3.2 \quad 6.6 \quad 1.5 \quad 11.0 \quad 7.6]$
1
2
3
4
5
6
7
$w_{5,5}$
$w_{6,6}$
$w_{6,7}$
$w_{7,6}$
$w_{7,7}$
$w_{2,6}$
$w_{6,2}$
$w_{2,7}$
$w_{7,2}$
$w_{2,2}$
$w_{7,4}$
$w_{4,7}$
$w_{1,1}$
$w_{1,4}$
$w_{4,1}$
$w_{4,4}$
$w_{2,3}$
$w_{3,2}$
$w_{3,1}$
$w_{1,3}$
$w_{3,4}$
$w_{4,3}$
$w_{3,3}$
$\boldsymbol{\beta}^{(1)} = [\boldsymbol{\beta}_1^{(1)} \quad \dots \quad \boldsymbol{\beta}_N^{(1)}] =$
$= [1.0 \quad 0.12 \quad 0.31 \quad 0.32 \quad -0.05 \quad \mathbf{0.57} \quad 0.13]$

Fig. 3. A directed weighted graph with five nodes and five edges (solid lines) with directed edges indicated by dashed lines. Vector $\hat{\boldsymbol{y}}$ shows the predicted air pollution concentrations based on the output of the AVGAE at a certain time instant. The synthetic graph is built based on [2], [18], meaning that nodes represent locations in the road network and edges are built based on the nodes geodesic distances and the road network topology of the considered city. As an example, below the graph one can observe a toy-example output we might obtain after computing the *topological-relevancy* vector $\boldsymbol{\beta}^{(1)}$ with respect to node 1 and before applying the normalisation factor. Note how node 6 is not directly connected to node 1, for which we wish to obtain an explanation. However, node 6 may obtain a high topological relevancy score in $\boldsymbol{\beta}^{(1)}$, due to their structural similitude in the network and predicted output of the AVGAE. Concretely, we observe that nodes 1 and 6 share the following structural characteristics (i.e., structural similitude). First, only nodes 1 and 6 have high pollutant concentrations (and these are similar). Additionally, only these certain nodes (node 1 and 6) are connected to two nodes which are again connected amongst them: Node 1 connects to nodes 3 and 4 (which are connected amongst them) and node 6 is connected to nodes 2 and 7 (which are interconnected). Finally, observe that the pair of nodes to which nodes 1 and 6 are connected (nodes 3, 4 and nodes 2, 7 respectively) have similar pairs of pollutant concentrations: one of the neighbours has medium-high pollutant concentrations (nodes 4 and 7) and the other has low pollutant concentration (nodes 2 and 3).

The non-negative least angle regression [50] is used to optimise the loss function in (6), obtaining the *topological-relevancy* vector $\boldsymbol{\beta}$ for a certain node v. In the *output* block of Fig. 2, the *topological-relevancy* scores are illustrated by the thickness of the edges connecting the graph nodes. We introduce the variable $K = |\mathcal{K}| \leq N$, which determines the predefined number of most relevant nodes to be selected for a certain node's prediction. Hence, in the explanation process of node v, the indices of the K-th highest scores of its $\boldsymbol{\beta}$ compose the subset of $\mathcal{K}^{(v)}$ most relevant nodes (i.e., locations). To simplify the notation, $\mathcal{K}^{(v)}$ will be referred simply as $\mathcal{K}$. These nodes are then employed as explanations of the selected node v's prediction. Once the $\mathcal{K}$ set of top K nodes has been computed for this node, the GNN-based model, that is, the AVGAE model, is re-trained given the graph of the $\mathcal{K}$ set of nodes and their related data, that is, the subset of known concentrations and point locations.

Finally, we want to capture the $\mathcal{K}$ subset by leveraging similitude metrics amongst the nodes. In the model, we include knowledge coming from geodesic distance and output prediction similarity—in our application, pollutant concentration levels—amongst the nodes. See Fig. 3 for a possible output of this computation which leverages the geodesic distance and output prediction similarity amongst two nodes (1 and 6) to predict a high relevancy score for node 6 with respect to node 1. Note that the proposed explanation model is applied on air pollution data; however, it can be extended to other applications where the graph topology and node relevancy can be of interest.

B. The Proposed Topological Node Embedding Definition

We now describe how to compute the *topological* node embeddings, which capture the graph structure seen from a certain node perspective. In essence, the definition of $\boldsymbol{x}^{(v)}$ aims at capturing the inherent characteristics of the whole graph structure from node v's perspective. To this end, we introduce a novel definition to measure the influence a node has on the network with respect to node v's point of view. Among the possible multiple paths between two nodes, in this definition we only consider the shortest path. See Fig. 4 for an illustration of how to compute the *topological* node embeddings of an specific graph example.

The initial *topological* node embedding of node v can be written as $\boldsymbol{x}^{(v)} \in \mathbb{R}^D$, with D the vector dimensionality. Note that we compute the *topological* node embedding in such a way that the vector dimensionality equals the number of graph nodes N, i.e., $D = N$. An element $\boldsymbol{x}_i^{(v)}$ of the *topological* node embedding is obtained following the proposed topology-aware score, which is formulated leveraging notions from electricity flows through parallel and serial resistors. By definition, the element $\boldsymbol{x}_i^{(v)}$ corresponds to the relationship between nodes v and i and, since the *topological* node embedding must be understandable, the score of this relation must be meaningful. If the *topological* node embedding is made unitary after computing the score of its elements, the smaller the score $\boldsymbol{x}_i^{(v)}$ relating nodes i and v, the weaker their relationship and vice-versa. The importance score between two nodes will depend, among other variables, on the number of jumps in the graph between them. Subsequently, the *topological* node embedding must contain this knowledge. Before introducing the computation rules for the *topological* node embedding, the weight scores $w_{i,j}$ of the graph edges are normalised, i.e., $\Omega_{i,j} = \frac{w_{i,j}}{W}$, applying a normalisation factor W calculated as follows $W = \sum_{\forall(i,j)} w_{i,j}$.

The rules for the computation of the *topological* node embedding $\boldsymbol{x}^{(v)}$ of a certain node v with respect to any other graph node i are defined as follows:

- Rule (0): $\boldsymbol{x}_i^{(v)}$ will be equal to 1 if and only if $i = v$ and otherwise $\boldsymbol{x}_i^{(v)} \in [0, 1)$.
- Rule (1): If only one jump is needed to connect nodes v and i, that is, if v and i are neighbouring nodes:

$$\boldsymbol{x}_i^{(v)} = \Omega_{v,i}$$

where $\Omega_{v,i}$ is the normalised weight of the edge between nodes i and j in the graph.

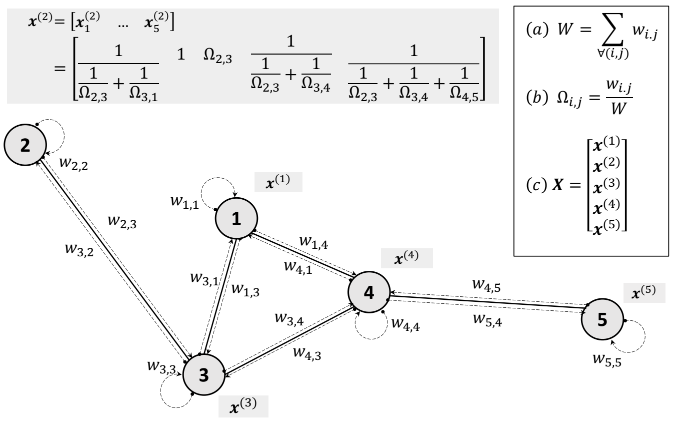

Fig. 4. A directed weighted graph with five nodes and five edges (solid lines) with directed edges indicated by dashed lines. To exemplify the computation process, we show the *topological* node embedding of node (2). Equation (a) gives the computation of the initial normalisation factor, where $w_{i,j}$ represents each of the direct edge weights. Equation (b) gives the computation of the normalised weight scores in the graph. (c) $\boldsymbol{X}$ is the matrix of *topological* node embeddings $\boldsymbol{x}^{(i)}$, used as input to the explanation model to compute the vector $\boldsymbol{\beta}$.

- Rule (2): If two jumps are needed to connect nodes v and i, that is, if the relevant path between v and i is though node k:
$$\boldsymbol{x}_i^{(v)} = \frac{1}{\frac{1}{\Omega_{v,k}} + \frac{1}{\Omega_{k,i}}}$$
- Rule (3): If three jumps are needed to connect nodes v and i, going though nodes k and l:
$$\boldsymbol{x}_i^{(v)} = \frac{1}{\frac{1}{\Omega_{v,k}} + \frac{1}{\Omega_{k,l}} + \frac{1}{\Omega_{l,i}}}$$
and so on.

Note that Rule (3) can be employed for any path of arbitrary length. For instance, let us solve for a path of length 4: $v \rightarrow k \rightarrow l \rightarrow i$ as follows. Given that nodes v and k are adjacent neighbours, $\boldsymbol{x}_i^{(v)}$ is calculated as

$$\boldsymbol{x}_i^{(v)} = \frac{1}{\frac{1}{\Omega_{v,k}} + \frac{1}{\boldsymbol{x}_i^{(k)}}}. \tag{15}$$

From Rule (2) we can conclude that $\Omega_{v,k}$ is the weight of the link connecting nodes v and k. On the contrary, the connection $k \rightarrow i$ is a two-jump path though l that can be calculated as:

$$\boldsymbol{x}_i^{(k)} = \frac{1}{\frac{1}{\Omega_{k,l}} + \frac{1}{\Omega_{l,i}}} \tag{16}$$

Substituting (16) into (15) we recover Rule (3). As such, we can generalise the computation of the topology-aware score for every pair of two nodes, independently of the walk distance amongst them.

Finally, consider two pairs of nodes (a, b) and (c, d) where the paths connecting a to b and c to d have an equal sum of edge scores amongst them $\sum w_{a,b} = \sum w_{c,d}$. Consider now that less jumps are required for path connecting a to b than for connecting c to d. Note that, even if the sum of edge scores between the nodes are equal, the more jumps needed, the lower the *topological* node embedding score amongst them will be. This is a desired behaviour because even if we want every node to be considered as a potential relevant node for v, nodes that are closer still have more probability of becoming an influential node. In the case of multiple shortest paths of the same length and jumps, the proposed *topological* node embedding definition has been designed so that it respects the dynamics of air pollution, namely, the physical equations of air flow. In the context of air pollution, the existence of multiple shortest paths does not have any physical meaning since pollution will flow in a defined direction, which is determined by external factors (i.e., wind, traffic, differences of pressure). To be more specific, as long as the selected path is the shortest in length, there is no difference which of them is followed for the explanation model.

V. Experiments

In this section, we perform an evaluation of both our explanation model and its explanations. In other words, we aim at providing a study on the quality of predictions and explanations. In our experiments, we consider the task of providing explanations of the output predictions given by the AVGAE model [2], [18] and alternative GNN models [4] [51]. We leverage the graph structure of the data and propose to infer, for a certain node, the most relevant nodes (i.e., locations) of the network. In addition, we study how the model performs under various conditions and evaluation metrics when the most and least important nodes are masked out. In Section V-A, we provide a summary of the characteristics of the considered data sets. Section V-B describes the evaluation metrics and the experimental procedure. Section V-C describes the hyper-parameter selection of the model. Section V-D refers to the experimental results based on the previously explained evaluation metrics. Finally, Section V-E includes a visual study of the model's explainability.

A. Data Sets

The results in this paper are obtained for two air quality data sets containing measurements of two air pollutants: NO_2, measured in parts per billion (*ppb*), and PM_{10}, measured in $\mu g/m^3$. Specifically, we employ the *imec* air pollution data set[1], which includes NO_2 measurements collected in the city of Antwerp, Belgium during April 2019 and the *Snuffelfiets* data set[2], which contains PM_{10} measurements of the city of Utrecht, Netherlands during June 2020. In both scenarios, we employ the settings reported in [2], [18] for the AVGAE model, namely, a time period equal to 30 days, a time span of temporal resolution of $\tau = 1$ h, and a distance threshold $\delta = 100$ m for the aggregation step, which results in the following two data sets: 1) an Antwerp NO_2 data set with $N = 4954$ and $T = 720$; and 2) an Utrecht PM_{10} data set with $N = 8292$ and $T = 720$. A statistical description of the data sets together with some relevant statistics concerning AVGAE original inference results is given in Table I. It is worth noting the scarcity of data: observed entries ranges from

[1] collected from https://obelisk.ilabt.imec.be/api/v2/docs/

[2] collected from https://snuffelfiets.nl/

0.355% in the Antwerp data set to 0.439% in the Utrecht data set, showing that mobile data collection still results in a lot of unmeasured locations, which highlights the need for air quality inference.

B. Experimental Setup

We select three pre-trained GNN-based models to solve the regression task of estimating air pollution concentrations. We test our explanation method with the AVGAE [2], [18] model, the recurrent graph convolutional neural network (RGCNN) [51] and the graph neural network (ChebyNet) [4]. To the best of our knowledge, AVGAE [2], [18] is the best performing GNN-based model for air quality estimation that does not leverage additional data other than the location of measurements. We also consider the RGCNN [51] and ChebyNet [4] models to demonstrate that our explanation method is independent of the underlying graph-based inference model. We test our explanation model in the two data sets containing air quality measurements described in Section V-A. The data, which contains measurements of air pollutant concentrations associated with time and geographical coordinates, is mapped to a weighted and undirected graph, where nodes correspond to specific locations in street segments of the city and edges depict geodesic similarity between nodes. Hence, as explained in Section III-B, an edge will be created amongst two nodes if the distance between them is smaller than $\delta = 100$ m or they belong to the same road segment. The weight of a connection is the inverse of the geodesic distance in meters computed by the Haversine formula [52].

We firstly compute the *topological* node embeddings as presented in Section IV-B. Then, we run the GNN model to obtain the predicted pollutant concentrations under different conditions. First, we obtain the predictions made by the GNN model before applying the NodeSel explanation method, that is, when all other nodes are considered when computing a node's value. Additionally, we obtain the predicted concentrations after applying the NodeSel method—in which case only the top K relevant nodes are considered when computing a node's prediction. Specifically, to train and test the AVGAE on each dataset, we randomly split the known data in train (90%) and test (10%) sets. We then train and test the AVGAE model (or the other GNN-based models) with the $\mathcal{V}$ set of graph nodes. For each node of the train and test set (i.e., the nodes to be explained), we apply the NodeSel model and find the set of its most relevant nodes $\mathcal{K}$. To be able to evaluate the quality of the NodeSel model in selecting the subset $\mathcal{K}$, we again train and test the AVGAE model using as input the data of subset $\mathcal{K}$ and compare the results with the AVGAE predictions when using $\mathcal{V}$.

We select the root-mean-square error (RMSE) and mean absolute error (MAE) to evaluate the prediction error made by the GNN model in the estimation of the air pollution concentrations. These results are reported in the first two columns of Tables II and III for the three GNN models. The rows with italic fonts in these tables show the RMSE and MAE scores of the GNN model when every other node in the network is considered for the prediction of a certain node (i.e., $K = N$). The remaining rows of RMSE and MAE are computed as follows. For each node in the network, the GNN model predicts the pollutant concentration of v when only the set of most relevant nodes $\mathcal{K}$ of v is considered. Then, we compute RMSE $= \sqrt{\frac{\sum_{i=1}^{N}(\boldsymbol{y}^{(i)} - \hat{\boldsymbol{y}}^{(i)})^2}{N}}$ and MAE $= \frac{\sum_{i=1}^{N}|\boldsymbol{y}^{(i)} - \hat{\boldsymbol{y}}^{(i)}|}{N}$, with $\boldsymbol{y}^{(i)}$ and $\hat{\boldsymbol{y}}^{(i)}$ the true and predicted pollutant concentrations of node i. As such, the RMSE and MAE metrics, calculated on the GNN model's prediction before and after applying the NodeSel method, can be used to indirectly assess the quality of the explanation.

In addition, we compute fidelity, contrastivity and sparsity to measure explainability in GNN-based regression tasks. Fidelity, contrastivity and sparsity were initially defined in [41] to assess explanations for graph-based deep learning models applied on the task of classification. In our work, however, the GNN model is solving a regression task; therefore, in order to apply these metrics, the predicted air pollution concentrations are turned into binary scores. To do so, we threshold the estimated air pollution values with the allowed guideline values defined by WHO [53]. Specifically, we use the upper limit of 42 $\mu g/m^3$ for NO_2 and 10 *ppb* for PM_{10}.

Fidelity captures the intuition that masking out the nodes highlighted by the explanation model should decrease the quality of the predictions. Concretely, fidelity is defined as the difference in accuracy when masking out its $\mathcal{K}$ set of K most relevant nodes. Consider $\mathcal{V}$ as the set of graph nodes and let $\mathcal{S} = \mathcal{V} \setminus \mathcal{K}$ be the set of less relevant nodes for a certain node v in the network that we wish to explain. The fidelity score of v is calculated as $\text{acc}_{\mathcal{V}} - \text{acc}_{\mathcal{S}}$, where $\text{acc}_{\mathcal{V}}$ and $\text{acc}_{\mathcal{S}}$ are the GNN model accuracy when the sets $\mathcal{V}$ or $\mathcal{S}$ are used in the prediction. The accuracy $\text{acc}_{\mathcal{V}}$ (respectively, $\text{acc}_{\mathcal{S}}$) is computed as the sum of the *accuracy scores* of the nodes in $\mathcal{V}$ (respectively, $\mathcal{S}$) divided by the number of nodes in $\mathcal{V}$ (respectively, $\mathcal{S}$). Note that, in order to follow the definition in [41], each graph node's values is classified as polluted or unpolluted based on the WHO threshold described above; therefore, its value becomes binary. The accuracy of a specific node v is then equal to 1 if both their true and predicted pollutant concentrations are classified as polluted or unpolluted, and 0 otherwise. Furthermore, to make differences more pronounced in our results, we express the accuracy score of a set in percentage, i.e., it belongs in the range $[0, 100]$. As a result, fidelity scores range between $[-100, 100]$; we shift the scores to the positive range of $[0, 200]$.

In classification problems, contrastivity was introduced to express the fact that highlighted class-specific features should differ between classes. We adapt the definition to our topology-aware explainability task by suggesting that two nodes $v, u \in \mathcal{V}$ with different predicted pollutant concentrations should have different sets of top K relevant nodes, which we refer to as $\mathcal{K}^{(v)}$ and $\mathcal{K}^{(u)}$. Hence, we calculate the overall contrastivity score by averaging the individual scores of node pairs with opposite pollutant concentrations (i.e., v polluted and u unpolluted node). For a node pair (v, u), their contrastivity is computed as $\frac{d_H\left(\zeta(\mathcal{K}^{(v)}), \zeta(\mathcal{K}^{(u)})\right)}{|\mathcal{K}^{(v)} \cup \mathcal{K}^{(u)}|}$ where $\zeta(\mathcal{K}) = (l_1, \dots, l_N) \in \{0,1\}^N$ with $l_i = 1$ if node $i \in \mathcal{K}$

TABLE I

Characteristics of the employed data sets. NO$_2$ data is measured in $\mu g/cm^3$ and PM$_{10}$ data is measured in *ppb*. Note that the maximum and minimum $predicted$ concentrations correspond to the maximum and minimum values of the predictions made by the AVGAE model.

Data Set	**Antwerp NO$_2$**	**Utrecht PM$_{10}$**
# Nodes	4,954	8,292
# Edges	69,032	99,738
Max. true concentration	124.41	11.0
Max. predicted concentration	164.64	14.76
Min. true concentration	0.0	0.0
Min. predicted concentration	-78.04	-0.80
Mean true concentration	81.91	3.64
Mean predicted concentration	66.47	3.42

belongs to the set $\mathcal{K}$ and otherwise 0, and $d_H(\cdot,\cdot)$ computes the Hamming distance between these binary strings of relevant nodes.

Sparsity aims at measuring how localised an explanation is in the graph and it is computed by averaging the sparsity score of every polluted or unpolluted node pair (v,u). First, we classify each node as polluted or unpolluted following the same procedure as for fidelity. Then, for every pair of nodes $(v,u \in \mathcal{V})$ where both have been classified as polluted or unpolluted, sparsity is defined as $1 - \frac{|\mathcal{K}^{(v)} \cup \mathcal{K}^{(u)}|}{N}$, where N is the total number of nodes. If the subsets of K most relevant nodes, $\mathcal{K}^{(v)}$ and $\mathcal{K}^{(u)}$, amongst two nodes greatly differ, i.e., $|\mathcal{K}^{(v)} \cup \mathcal{K}^{(u)}| \approx N$, the fraction will converge to 1 and the sparsity factor will be close to 0. On the contrary, as the subsets become more similar, the fraction will be smaller than 1 and the sparsity score will increase. Note that, for fidelity, contrastivity and sparsity, higher scores imply better explainability performance. Finally, note that in Tables II and III, sparsity and contrastivity are mainly defined as a control metric. By definition, a smaller size K of the relevancy set $\mathcal{K}$ directly implies higher sparsity and contrastivity scores. These two metrics only become meaningful in Tables V and VI, addressed in Section V-D, where the values are compared with other explanation methods.

C. Hyperparameter Settings

We follow the best configuration suggested in [2], [18] to define the hyperparameters of AVGAE. Specifically, we select a learning rate of 0.005 over 1000 epochs. Similarly, the hyperparameters of the RGCNN and ChebyNet models are set as in [51] and [4], respectively. Furthermore, the hyperparameters of the proposed explanation model are found empirically via tuning. We find the best results are obtained when the NodeSel model is trained with a small learning rate of 0.000001 and 500 epochs, we make use of early stopping to avoid over-fitting and employ a 4-fold cross validation procedure.

For each data set, we wish to select the optimal number of relevant nodes K. Accordingly, we perform an evaluation study based on the metrics described in Section V-B. To obtain the best K value for each data set, we firstly test our explanation model with different K values, from which we obtain the most relevant nodes for each location. Then, we run the GNN-based model only employing the data coming from the $\mathcal{K}$ set of more relevant nodes and collect the metric scores. In Tables II and III, we show the quality of the GNN-based model predictions for different K values, under the condition that $K \leq N$. Table II contains the results of the Antwerp NO$_2$ data set and Table III contains the results of the Utrecht PM$_{10}$ data set. Note that these results were obtained employing a 4-fold cross validation procedure. Lower values are better for RMSE and MAE while higher values are better for fidelity, contrastivity and sparsity. N/A stands for not applicable since, by definition, the computation of fidelity, contrastivity and sparsity needs subsets of the whole data set. Our goal is to select the lowest K value that provides good quality, namely, the K value with which the quality performance is comparable with the results of the italic row (where $K = N$). As such, scores in bold represent the best results for K values different than $K = N$. The results in Tables II and III show that with certain K values we can achieve similar performance to $K = N$ with a high fidelity score. This is because we infer the subset of most influential nodes in such a way that the most relevant information for the node of interest is inherently present in the subset. We notice that the scores for RMSE and MAE do not change monotonically with K. We believe this behaviour is due to the fact that increasing K over an optimal value (which depends on each data set), negatively affects the performance of the GNN-based model. To be more specific, for a certain node's prediction, by increasing K, the GNN-based model is receiving as input the relevant nodes as well as nodes that are less relevant, thereby lowering the performance of the GNN-based model. Finally, note that, in both data sets, the best performance is obtained when K is approximately $N/2$; specifically, $K = 2000$ for the Antwerp data set and $K = 4000$ for the Utrecht data set.

D. Experimental Results

In this section, we assess the explainability capability and performance of the NodeSel model using the aforementioned data sets and GNN-based models. We employ AVGAE [2], [18] as the underlying GNN-based model. Moreover, we perform the same study with RGCNN [51] and ChebyNet [4] as GNN-based models for comparison. We first aim to study the relevancy of the $\mathcal{K}$ set of K most relevant nodes found by the NodeSel model. Specifically, we assess the deviation in the GNN-based model performance when employing the subset of least relevant nodes, that is, the set $\mathcal{S}$ defined as $\mathcal{S} = \mathcal{V} \setminus \mathcal{K}$, and compare the results with those obtained with the $\mathcal{K}$ set. Secondly, we compare the effectiveness of the NodeSel model with respect to other benchmark models; namely, (i) a random-based selection method, which selects the most relevant graph nodes randomly; (ii) GraphLIME [14], which selects the *n-hop jumps* closest graph nodes maximising the spatial correlation; and (iii) GNNExplainer [15], which computes a binary mask that maximises the negative cross entropy between the label class and the model prediction. Note that, in the graph signal processing domain [54], method (i) is somehow equivalent to a node selection through random walks as in [55], [56]

TABLE II
PERFORMANCE OF VARIOUS GNN-BASED MODELS AFTER APPLYING OUR PROPOSED MODEL (NODESEL) WITH VARIOUS K VALUES USING THE ANTWERP NO$_2$ ($N = 4954$) DATA SET.

GNN Model	#Relevant Nodes	RMSE	MAE	Fidelity	Contrastivity	Sparsity
AVGAE [2], [18]	$\boldsymbol{K = 1000}$	11.173	9.4537	2.535	0.000571	0.646
	$\boldsymbol{K = 2000}$	**10.413**	**7.250**	**55.132**	0.000335	0.395
	$\boldsymbol{K = 3000}$	11.244	9.255	10.605	0.000255	0.230
	$\boldsymbol{K = 4000}$	10.951	7.977	9.424	0.000204	0.104
	$K = N = 4954$	*10.294*	*7.551*	*N/A*	*N/A*	*N/A*
RGCNN [51]	$\boldsymbol{K = 1000}$	15.049	10.867	15.751	0.000529	0.639
	$\boldsymbol{K = 2000}$	**12.995**	**9.008**	58.657	0.000359	0.401
	$\boldsymbol{K = 3000}$	13.724	9.035	**58.729**	0.000293	0.228
	$\boldsymbol{K = 4000}$	13.194	9.052	57.245	0.000215	0.101
	$K = N = 4954$	*12.678*	*8.263*	*N/A*	*N/A*	*N/A*
ChebyNet [4]	$\boldsymbol{K = 1000}$	12.167	9.001	51.092	0.000590	0.523
	$\boldsymbol{K = 2000}$	11.632	8.735	**54.890**	0.000462	0.398
	$\boldsymbol{K = 3000}$	11.927	8.976	54.821	0.000356	0.204
	$\boldsymbol{K = 4000}$	**11.553**	**8.434**	54.832	0.000231	0.116
	$K = N = 4954$	*11.348*	*8.245*	*N/A*	*N/A*	*N/A*

TABLE III
PERFORMANCE OF VARIOUS GNN-BASED MODELS AFTER APPLYING OUR PROPOSED MODEL (NODESEL) WITH VARIOUS K VALUES USING THE UTRECHT PM$_{10}$ ($N = 8292$) DATA SET.

GNN Model	#Relevant Nodes	RMSE	MAE	Fidelity	Contrastivity	Sparsity
AVGAE [2], [18]	$\boldsymbol{K = 1000}$	0.587	**0.225**	83.948	0.000618	0.802
	$\boldsymbol{K = 2000}$	0.527	0.343	77.995	0.000328	0.634
	$\boldsymbol{K = 3000}$	0.451	0.273	82.538	0.000228	0.466
	$\boldsymbol{K = 4000}$	0.371	0.287	**125.970**	0.000185	0.349
	$\boldsymbol{K = 5000}$	0.481	0.342	77.310	0.000151	0.199
	$\boldsymbol{K = 6000}$	**0.343**	0.248	83.551	0.000137	0.116
	$\boldsymbol{K = 7000}$	0.582	0.481	7.198	0.000123	0.039
	$K = N = 8292$	*0.539*	*0.449*	*N/A*	*N/A*	*N/A*
RGCNN [51]	$\boldsymbol{K = 1000}$	0.824	0.639	29.766	0.000661	0.742
	$\boldsymbol{K = 2000}$	0.810	0.613	59.201	0.000582	0.606
	$\boldsymbol{K = 3000}$	0.787	0.584	77.530	0.000464	0.524
	$\boldsymbol{K = 4000}$	**0.769**	**0.565**	88.453	0.000306	0.433
	$\boldsymbol{K = 5000}$	0.792	0.572	102.934	0.000171	0.295
	$\boldsymbol{K = 6000}$	0.753	0.567	**104.033**	0.000162	0.190
	$\boldsymbol{K = 7000}$	0.775	0.566	100.909	0.000110	0.044
	$K = N = 8292$	*0.745*	*0.560*	*N/A*	*N/A*	*N/A*
ChebyNet [4]	$\boldsymbol{K = 1000}$	1.102	0.894	78.026	0.000615	0.801
	$\boldsymbol{K = 2000}$	1.172	0.863	81.953	0.000332	0.629
	$\boldsymbol{K = 3000}$	0.976	0.865	84.250	0.000192	0.491
	$\boldsymbol{K = 4000}$	**0.840**	0.693	**111.403**	0.000177	0.359
	$\boldsymbol{K = 5000}$	0.897	0.704	101.728	0.000129	0.205
	$\boldsymbol{K = 6000}$	0.893	0.693	100.934	0.000120	0.113
	$\boldsymbol{K = 7000}$	0.888	**0.666**	100.982	0.000103	0.039
	$K = N = 8292$	*0.801*	*0.637*	*N/A*	*N/A*	*N/A*

TABLE IV
RESULTS ON THE EVALUATION METRICS FOR BOTH DATA SETS USING THE $\mathcal{S}$ SET OF LESS RELEVANT NODES WITH THE BEST K VALUE FOUND IN TABLES II AND III FOR ANTWERP AND UTRECHT DATA SETS, RESPECTIVELY. NO$_2$ REFERS TO THE ANTWERP NO$_2$ DATA SET WITH $N = 4954$ AND $K = 2000$. PM$_{10}$ REFERS TO THE UTRECHT PM$_{10}$ DATA SET WITH $N = 8292$ AND $K = 4000$.

	RMSE	MAE	Fidelity	Contrastivity	Sparsity
NO$_2$	69.489	64.3366	14.249	0.000112	0.469
PM$_{10}$	2.991	2.130	12.762	0.000177	0.652

and method (ii) is somehow equivalent to the aggregation GNN procedure of [57], [58]. Specifically, [57] presents two architectures that first select the n-hop jump neighbourhood per node and then, explain the model decision by using a feature selection procedure or a diffusion process.

Similarly to GNNExplainer, our model leverages spatial (i.e., locations) and label (i.e., pollutant level) correlations. However, NodeSel obtains a non-binary mask, which allows us to compare the relevancy score (including node similitude knowledge) amongst different nodes. We compare the performance in terms of the different evaluation metrics presented in Section V-B using both data sets.

In Section V-C, we were able to obtain the best K value for each data set. In addition, we showed that, independent of the underlying GNN-based model, the value of K remains unchanged for a certain data set. Given the $\mathcal{K}$ set of K most relevant nodes found by NodeSel for a certain node and time, notice that $\mathcal{S} = \mathcal{V} \setminus \mathcal{K}$ depicts the set of less relevant nodes in the network. We now conjecture that, for a certain node and time, the $\mathcal{K}$ set contains the sufficient information to obtain a high quality prediction. Consequently, the $\mathcal{S}$ set of less relevant

TABLE V
COMPARISON OF DIFFERENT EXPLANATION METHODS AND NODE EMBEDDINGS FOR VARIOUS GNN-BASED MODELS, USING THE ANTWERP NO_2 DATA SET AND $K = 2000$.

		RMSE	**MAE**	**Fidelity**	**Contrastivity**	**Sparsity**
AVGAE [2], [18]	**Random**	17.227	11.650	44.528	0.000313	0.356
	GraphLIME [14]	11.083	9.987	54.485	**0.000421**	0.551
	GNNExplainer [15]	**10.222**	8.332	50.223	0.000245	0.395
	NodeSel (Ours)	10.413	**7.250**	**55.132**	0.000335	0.395
RGCNN [51]	**Random**	16.923	10.273	15.685	0.000288	0.322
	GraphLIME [14]	13.082	9.131	55.924	0.000345	0.388
	GNNExplainer [15]	13.111	**9.006**	**58.932**	0.000351	0.394
	NodeSel (Ours)	**12.995**	9.008	58.657	**0.000359**	**0.401**
ChebyNet [4]	**Random**	15.020	10.399	20.734	0.000388	0.299
	GraphLIME [14]	**11.628**	8.770	54.004	**0.000404**	0.403
	GNNExplainer [15]	11.694	8.803	54.963	0.000401	**0.484**
	NodeSel (Ours)	11.632	**8.735**	**54.890**	0.000462	0.398

TABLE VI
COMPARISON OF DIFFERENT EXPLANATION METHODS AND NODE EMBEDDINGS FOR VARIOUS GNN-BASED MODELS, USING THE UTRECHT PM_{10} DATA SET WITH $K = 4000$.

		RMSE	**MAE**	**Fidelity**	**Contrastivity**	**Sparsity**
AVGAE [2], [18]	**Random**	0.744	0.570	62.305	0.000165	0.269
	GraphLIME [14]	0.385	0.339	75.584	0.000155	0.326
	GNNExplainer [15]	0.371	0.300	110.621	0.000182	0.344
	NodeSel (Ours)	0.371	**0.287**	**125.970**	**0.000185**	**0.349**
RGCNN [51]	**Random**	0.993	0.824	60.783	0.000191	0.304
	GraphLIME [14]	0.841	0.577	77.042	0.000204	0.357
	GNNExplainer [15]	0.806	0.591	**89.065**	**0.000309**	0.426
	NodeSel (Ours)	**0.769**	**0.565**	88.453	0.000306	**0.433**
ChebyNet [4]	**Random**	0.930	0.834	57.063	0.000126	0.280
	GraphLIME [14]	**0.811**	0.712	109.878	0.000173	0.351
	GNNExplainer [15]	0.834	0.703	102.894	**0.000198**	0.355
	NodeSel (Ours)	0.840	**0.693**	**111.403**	0.000177	**0.359**

nodes should produce a low quality prediction. To prove it, in Table IV we show the evaluation metrics when running the AVGAE model using the $\mathcal{S}$ set, composed of $S = N - K$ nodes. We compare these results with the numbers presented in Table II for Antwerp NO_2 and Table III for Utrecht PM_{10} data sets, with $K = 2000$ and $K = 4000$, respectively. It is evident that the prediction accuracy is much lower if the $\mathcal{S}$ set is used instead of the $\mathcal{K}$ set.

Secondly, we compare the proposed explanation model and the three benchmark models. The performance is assessed in terms of the different evaluation metrics presented in Section V-B. The experimental results are reported in Table V for the Antwerp data set and in Table VI for the Utrecht data set. From Tables V and VI, we observe that random selection is the worst performing model. This is an expected result since employing the spatial- or feature-based correlations between the data at hand should work better than a random selection. GraphLIME [14] leads to a better estimation accuracy than random selection. This model manages to capture properly the spatial correlation in the air quality measurements with respect to the geodesic distance. As a result, we believe that GraphLIME [14] will perform poorly when the relevant nodes are not in the closest neighbourhood while its performance will improve in the opposite case. On the contrary, our NodeSel model is able to learn from two sources of information, the geodesic distance and the pollution levels, allowing NodeSel to select which source of data is more relevant in each case. In addition, it is evident that GNNExplainer [15] is the second best-performing model. NodeSel achieves the best overall performance in terms of RMSE, MAE, contrastivity and fidelity in both data sets, with a few exceptions, such as the RMSE and contrastivity scores in the Antwerp NO_2 data set, where NodeSel is the second best-performing model. In contrast to other models, NodeSel effectively captures the spatial correlation and air pollution similarity in the data and leverages the underlying graph structure of the street network.

E. Visualisations

In this section we provide visual examples of the effectiveness of our explanation approach. We randomly select a set of node and time pairs from the test set, which will serve as samples for this purpose. Examples of resultant visualisations are presented in Figures 5 and 6 using the Antwerp data set, and Figures 7 and 8 using the Utrecht data set. For a certain time instant and given a node to be explained, we show the set of most relevant nodes $\mathcal{K}$ as found by *NodeSel* (recall that $K = 2000$ when employing the Antwerp data set and $K = 4000$ when employing the Utrecht data set). Specifically, we show the location of nodes in $\mathcal{K}$ and colour them by the relevancy score observed in $\boldsymbol{\beta}$. The node to-be-explained is highlighted with a red arrow.

In both scenarios, we observe that the closest nodes in terms of the geodesic distance are of high relevancy for a node's prediction. Visually, nodes that are in the same or neighbouring

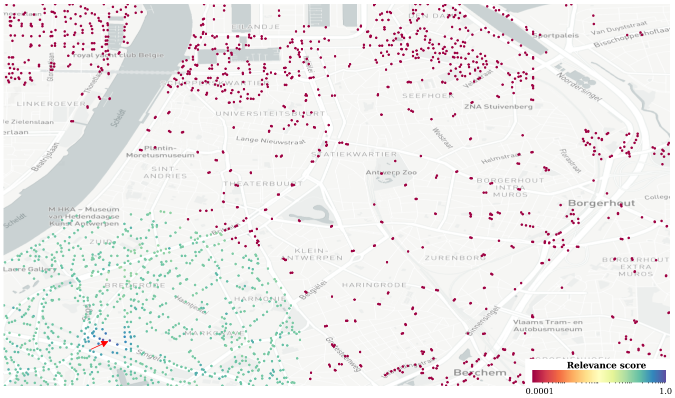

Fig. 5. Explanation of a node (pointed out by red arrow) of the city of Antwerp at time $t = 11$am.

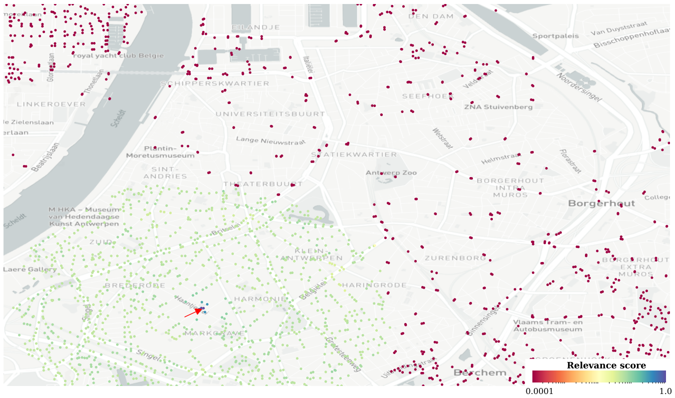

Fig. 6. Explanation of a node (pointed out by the red arrow) of the city of Antwerp at time $t = 10$am.

streets are coloured in purple or dark blue. As such, in terms of how pollution flows, it is evident that nodes in close-by streets are more likely to affect the predicted concentration in the node we wish to explain. Yet, a group of nodes that are further away (e.g., red-coloured top nodes in Fig. 5) also appear as important in the *topological-relevancy* vector $\boldsymbol{\beta}$, probably due to the fact that these nodes have similar pollution levels. We conjecture that is because these long-distance nodes have similar characteristics in terms of surrounding areas, e.g., closeness to green areas, industries or the river. For instance, in Fig. 8, we observe that the red-colored nodes are similar to the node we wish to explain in terms of closeness to the city center (e.g., both of them are far from the city center). Similarly, in Fig. 6, the node we wish to explain and the red-colored nodes are similar in terms of distance to high traffic roads or to the river. We conclude that our model is able to capture the most relevant nodes in terms of geodesic distance. In addition, it is capable to collect numerous nodes which are further away but might share similar characteristics with the original node to be explained.

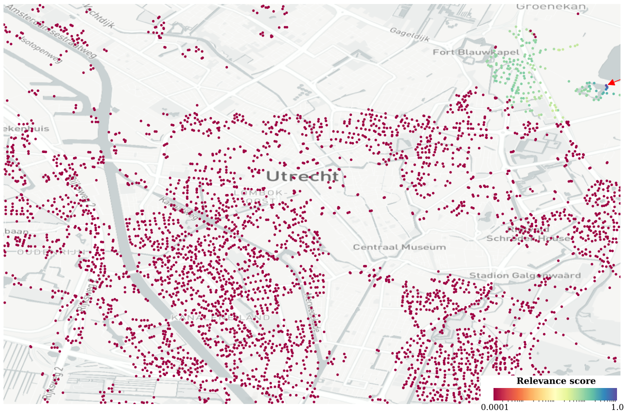

Fig. 7. Explanation of a node (pointed out by red arrow) of the city of Utrecht at time $t = 6$am.

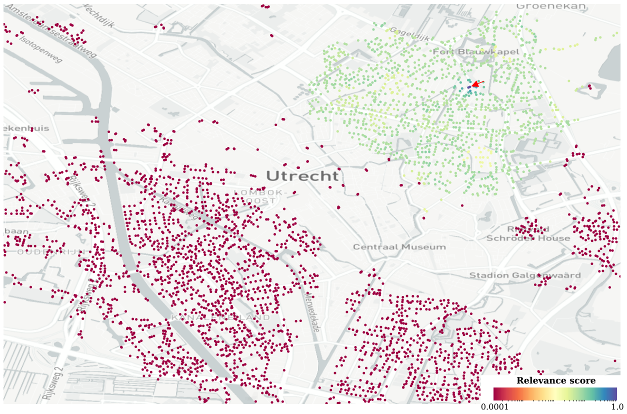

Fig. 8. Explanation of a node (pointed out by the red arrow) of the city of Utrecht at time $t = 4$pm.

VI. Conclusion and Future Work

We introduced a novel post-hoc explanation technique for GNN-based models, which is able to capture the most relevant locations for a certain graph node. By capturing the non-linear dependencies between predictions and the graph structure as seen from each node's perspective, the model is able to infer a vector which highlights the relevancy factor from such node to every other graph node. Although simple, the introduced technique upgrades the commonly used methods of either sampling the local neighbourhood via *n-hop* jumps or employing the complete set of nodes in the graph. As shown by experimental results, the proposed model delivers high-quality explanations, outperforming state-of-the-art explanation methods for deep learning on graphs. Additionally, we have proposed a novel definition for the node representation, referred to as *topological* node embedding, which intuitively captures the topology of the graph node surroundings as a measure of importance in the network. Jointly, we obtain a post-hoc explanation technique that can be easily integrated in any GNN-based model. The technique was experimentally assessed in the task of air quality estimation from mobile measurements using two air quality data sets. The model presented in this article can be extended to any GNN model that operates with graph-structured data. Future work aims at increasing the external non-graph-based input knowledge, for instance, including node individual features. Additionally, future research will aim at extrapolating the AVGAE model and the explanation method NodeSel to a time-aware domain where the dynamic aspect of the data is considered. For instance, leveraging recurrent neural networks such as graph recurrent NNs (GRNNs) during the prediction step or leveraging the dynamic evolution of the set of relevant nodes during the explanation step. Finally, we expect to generalise the model to other application domains.

Acknowledgment

This research was supported by the Research Foundation - Flanders (FWO) under the PhD Fellowship Strategic Basic Research (Project 1SC4521N) and imec under the AAA Project *AI-based Air Quality Map and Analytics*.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] T. H. Do, D. Minh Nguyen, E. Tsiligianni, A. L. Aguirre, V. Panzica La Manna, F. Pasveer, W. Philips, and N. Deligiannis, "Matrix Completion with Variational Graph Autoencoders: Application in Hyperlocal Air Quality Inference," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7535–7539.

[3] T. H. Do, D. M. Nguyen, E. Tsiligianni, B. Cornelis, and N. Deligiannis, "Multiview Deep Learning for Predicting Twitter Users' Location," 2017.

[4] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *International Conference on Neural Information Processing Systems (NeurIPS)*, 2016, p. 3844–3852.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016.

[6] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2020.

[7] X. Li, L. Peng, J. Shao, and T. Chi, "Deep learning architecture for air quality predictions," *Environmental Science and Pollution Research*, vol. 23, 11 2016.

[8] V. Reddy, P. Yedavalli, S. Mohanty, and U. Nakhat, "Deep Air : Forecasting Air Pollution in Beijing , China," *Environmental Science*, 2018.

[9] S. Khan, S. Nazir, I. García-Magariño, and A. Hussain, "Deep learning-based urban big data fusion in smart cities: Towards traffic monitoring and flow-preserving fusion," *Computers & Electrical Engineering*, vol. 89, p. 106906, 2021.

[10] Q. Tao, F. Liu, Y. Li, and D. Sidorov, "Air Pollution Forecasting Using a Deep Learning Model Based on 1D Convnets and Bidirectional GRU," *IEEE Access*, vol. 7, pp. 76 690–76 698, 2019.

[11] N. Xie, G. Ras, M. V. Gerven, and D. Doran, "Explainable deep learning: A field guide for the uninitiated," *ArXiv*, 2020.

[12] T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K. T. Schütt, K.-R. Müller, and G. Montavon, "Higher-Order Explanations of Graph Neural Networks via Relevant Walks," 2020.

[13] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability Methods for Graph Convolutional Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 772–10 781.

[14] Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, and Y. Chang, "GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks," 2020.

[15] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating Explanations for Graph Neural Networks," 2019.

[16] Z. Qi, T. Wang, G. Song, W. Hu, X. Li, and Z. Zhang, "Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-Grained Air Quality," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2285–2297, 2018.

[17] W. Cheng, Y. Shen, Y. Zhu, and L. Huang, "A neural attention model for urban air quality inference: Learning the weights of monitoring stations," in *AAAI*, 2018.

[18] T. H. Do, E. Tsiligianni, X. Qin, J. Hofman, V. P. La Manna, W. Philips, and N. Deligiannis, "Graph-deep-learning-based inference of fine-grained air quality from mobile iot sensors," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8943–8955, 2020.

[19] C. Bellinger, M. S. M. Jabbar, O. R. Zaiane, and A. Osornio-Vargas, "A systematic review of data mining and machine learning for air pollution epidemiology," *BMC Public Health*, vol. 17, 2017.

[20] J. Ma, J. C. Cheng, C. Lin, Y. Tan, and J. Zhang, "Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques," *Atmospheric Environment*, vol. 214, p. 116885, 2019.

[21] P. W. Soh, J. W. Chang, and J. W. Huang, "Adaptive Deep Learning-Based Air Quality Prediction Model Using the Most Relevant Spatial-Temporal Relations," *IEEE Access*, vol. 6, pp. 38 186–38 199, 2018.

[22] D. Byun and K. L. Schere, "Review of the Governing Equations, Computational Algorithms, and Other Components of the Models-3 Community Multiscale Air Quality (CMAQ) Modeling System," *Applied Mechanics Reviews*, vol. 59, no. 2, pp. 51–77, 03 2006.

[23] J. D. Marshall, E. Nethery, and M. Brauer, "Within-urban variability in ambient air pollution: Comparison of estimation methods," *Atmospheric Environment*, vol. 42, no. 6, pp. 1359–1369, Feb. 2008.

[24] Y. Wang, X. Zhang, and R. R. Draxler, "Trajstat: Gis-based software that uses various trajectory statistical analysis methods to identify potential sources from long-term air pollution measurement data," *Environmental Modelling & Software*, vol. 24, no. 8, pp. 938–939, 2009.

[25] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *CoRR*, vol. abs/1312.6034, 2013.

[26] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[27] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3145–3153.

[28] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014, pp. 818–833.

[29] R. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," *CoRR*, vol. abs/1704.03296, 2017. [Online]. Available: http://arxiv.org/abs/1704.03296

[30] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," *CoRR*, vol. abs/1702.04595, 2017. [Online]. Available: http://arxiv.org/abs/1702.04595

[31] M. T. Ribeiro, S. Singh, and C. Guestrin, """Why should i trust you?" Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[32] B.-J. Hou and Z.-H. Zhou, "Learning with interpretable structure from gated rnn," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2267–2279, 2020.

[33] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3156–3164.

[34] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2016, pp. 1480–1489.

[35] D. H. Park, L. A. Hendricks, Z. Akata, B. Schiele, T. Darrell, and M. Rohrbach, "Attentive explanations: Justifying decisions and pointing to the evidence," *arXiv preprint arXiv:1612.04757*, 2016.

[36] M. Hind, D. Wei, M. Campbell, N. C. Codella, A. Dhurandhar, A. Mojsilović, K. Natesan Ramamurthy, and K. R. Varshney, "Ted: Teaching ai to explain its decisions," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 123–129.

[37] X. Li and J. Saúde, "Explain graph neural networks to understand weighted graph features in node classification," in *Machine Learning and Knowledge Extraction*, A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl, Eds. Cham: Springer International Publishing, 2020, pp. 57–76.

[38] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang, "Parameterized Explainer for Graph Neural Network," 2020.

[39] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, “Striving for simplicity: The all convolutional net,” *arXiv:1412.6806*, 2014.

[40] M. Gevrey, I. Dimopoulos, and S. Lek, “Review and comparison of methods to study the contribution of variables in artificial neural network models,” *Ecological Modelling*, vol. 160, no. 3, pp. 249–264, 2003.

[41] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, “Explainability methods for graph convolutional neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 772–10 781.

[42] H. Yuan, H. Yu, S. Gui, and S. Ji, “Explainability in Graph Neural Networks: A Taxonomic Survey,” 2020.

[43] F. Baldassarre and H. Azizpour, “Explainability Techniques for Graph Convolutional Networks,” *CoRR*, vol. abs/1905.13686, 2019.

[44] H.-P. Hsieh, S.-D. Lin, and Y. Zheng, “Inferring air quality for station location recommendation based on urban big data,” in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 437–446.

[45] Y. Zheng, F. Liu, and H.-P. Hsieh, “U-air: When urban air quality inference meets big data,” in *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1436–1444.

[46] Y. Y. Chiang, Y. Y. Lin, M. Franklin, S. p. Eckel, J. L. Ambite, and W. Ku, “Building Explainable Predictive Analytics for Location-Dependent Time-Series Data,” in *IEEE International Conference on Cognitive Machine Intelligence (CogMI)*, Dec. 2019, pp. 202–209.

[47] R. Tibshirani, “Regression Shrinkage and Selection via the Lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[48] M. Yuan and Y. Lin, “Model Selection and Estimation in Regression With Grouped Variables,” *Journal of the Royal Statistical Society Series B*, vol. 68, pp. 49–67, 02 2006.

[49] M. Yamada, W. Jitkrittum, L. Sigal, E. P. Xing, and M. Sugiyama, “High-dimensional feature selection by feature-wise kernelized lasso,” *Neural computation*, vol. 26, no. 1, pp. 185–207, 2014.

[50] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *The Annals of Statistics*, vol. 32, no. 2, Apr 2004.

[51] F. Monti, M. M. Bronstein, and X. Bresson, “Geometric matrix completion with recurrent multi-graph neural networks,” *CoRR*, vol. abs/1704.06803, 2017. [Online]. Available: http://arxiv.org/abs/1704.06803

[52] G. V. Brummelen, *Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry*. Princeton University Press, 2013.

[53] WHO, “World Health Organization, Burden of disease from ambient air pollution for 2012 — Summary of results,” https://www.euro.who.int/__data/assets/pdf_file/0004/193108/REVIHAAP-Final-technical-report-final-version.pdf), 2014, accessed 1 July 2019.

[54] S. Chen, R. Varma, A. Sandryhaila, and J. Kovacevic, “Discrete signal processing on graphs: Sampling theory,” *CoRR*, vol. abs/1503.05432, 2015. [Online]. Available: http://arxiv.org/abs/1503.05432

[55] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, “Random sampling of bandlimited signals on graphs,” *CoRR*, vol. abs/1511.05118, 2015. [Online]. Available: http://arxiv.org/abs/1511.05118

[56] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Sampling of graph signals with successive local aggregations,” *CoRR*, vol. abs/1504.04687, 2015. [Online]. Available: http://arxiv.org/abs/1504.04687

[57] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, “Convolutional neural network architectures for signals supported on graphs,” *IEEE Transactions on Signal Processing*, vol. 67, no. 4, p. 1034–1049, Feb 2019. [Online]. Available: http://dx.doi.org/10.1109/TSP.2018.2887403

[58] A. Anis, A. Gadde, and A. Ortega, “Efficient sampling set selection for bandlimited graph signals using graph spectral proxies,” *CoRR*, vol. abs/1510.00297, 2015. [Online]. Available: http://arxiv.org/abs/1510.00297