Title

Harnessing Sheaf Theory for Enhanced Air Quality Monitoring: Overcoming Conventional Limitations with Topology-Inspired Self-correcting Algorithm

ISBN
9783031474538

Authors
Pham, Anh-Duy
Le, An Dinh
Le, Chuong Dinh
et al.

Publication Date
2023

DOI
10.1007/978-3-031-47454-5_8

Harnessing Sheaf Theory for Enhanced Air Quality Monitoring: Overcoming Conventional Limitations with Topology-Inspired Self-Correcting Algorithm

Anh-Duy Pham[1], An Dinh Le[2] Chuong Dinh Le[3], Hoang Viet Pham[3], and Hien Bich Vo[3]

[1] Hochschule Bonn-Rhein-Sieg, Sank Augustin, Germany,
duy.pham@smail.inf.h-brs.de
[2] University of California San Diego, California, USA
[3] Vietnamese-German University, Binh Duong, Vietnam

Abstract. Sheaf theory is a potent but intricate tool that is supported by topological theory. It offers more accuracy and adaptability than traditional graph theory when modeling the connections between several characteristics. This is especially valuable in air quality monitoring, where sudden changes in local dust particle density can be hard to measure accurately using commercial instruments. Conventional air quality measurement techniques often depend on calibrating the measurement with standard instruments or calculating the measurement's moving average over a fixed period. However, this can result in an incorrect index at the measurement location, as well as an excessive smoothing effect on the signal. To address this issue, this study proposes a self-correcting algorithm that employs sheaf theory to account for vehicle counts as a local air quality change-causing factor. By deducing the number of vehicles and incorporating it into the recorded PM2.5 index from low-cost air monitoring sensors, we can achieve real-time self-correction. Additionally, the sheaf-theoretic approach enables straightforward scaling to multiple nodes for further filtering effects. By integrating sheaf theory into air quality monitoring, we can overcome the limitations of conventional techniques and provide more precise and dependable results.

Keywords: sheaves, consistency structure, low-cost sensors, air quality monitoring, causality

1 Introduction

Air quality, impacting global health, is influenced by factors like weather and traffic. Worldwide monitoring stations and legislation help mitigate pollution. Various sensors collect atmospheric contaminant data to analyze pollution patterns and develop solutions. Handling heterogeneous data is vital in air quality monitoring, as multiple sources provide partial information, such as PM10, PM2.5 indices, video footage, vehicle counts, and social media data. A model integrating diverse data types optimizes sensor resources and delivers comprehensive information. This study demonstrates mathematical sheaves as an effective processing framework for integrating supplementary factors and enabling fine-grained analytics tailored to specific deployments. The research focuses on multi-sensor fusion and simulations of related methodologies, including air monitoring and vehicle tracking systems.

Vehicle to PM2.5 concentration. A 2018 study from Bangkok, Thailand, explored the relationship between PM2.5 levels and traffic flow on various roadsides using linear regression and path analysis [30]. The results suggest a direct impact of traffic flow on particle levels in both open and enclosed areas, with path analysis offering more accurate and efficient outcomes than linear regression at different PM concentration levels. The European Environment Agency's guidelines [3] address exhaust emissions, including PM2.5, N2O, CO2, and NH3, considering factors such as total distance driven, technology-specific equipment, and vehicle numbers to estimate emissions more accurately. The study also produced tables with estimated emission factors for each commercial vehicle technology in Europe. For Ho Chi Minh City, Vietnam, a formula calculates a vehicle's PM2.5 emissions [23], based on a 2020 California Air Resource Board guidebook [20] detailing the relationship between various vehicle model technologies and pollutant emissions like NOx, PM2.5, CO, and ROG.

Multi-sensor fusion and sheaf-theoretical approach. Data fusion involves combining data from multiple sources [34], with various experimental programs demonstrating that merging sensor detections improves coverage and performance [5, 13, 18, 21, 31, 32]. Techniques like [36] use a heterogeneous sensor collection, but often require sensor registration to a common coordinate system [4, 10, 12, 16, 33, 35]. Changes in sensor placement can impact such methods due to the lack of a standard coordinate system. An alternative is "possibilistic" information theory [6–8], which encodes sensor models as propositions and inference rules. Although these methods can handle heterogeneous sensor collections, they lack theoretical guarantees for homogeneous collections and face combinatorial complexity issues. The proposed methodology, which uses local consistency models and sheaves, requires careful modeling before examining observations, reducing sample requirements. Sheaves offer a canonical and practical formalization for merging local data sets into more consistent global data [19]. They are an effective tool for organizing heterogeneous sensor deployments [15] and addressing data fusion questions [24]. Consistency radius [27] quantifies the relationship between sheaf geometry and observation sets, leading to practical data fusion algorithms [29]. Sheaves enable finer-grained consistency analysis through consistency filtration [28]. Cohomology, a technique for understanding local observation fusion, has applications in various fields like network structure [11, 22, 26] and quantum information [1, 2], with straightforward computation [25] and efficient methods [9].

This study introduces a unique framework for air quality monitoring using mathematical sheaves to integrate various data types, such as PM2.5 indices, video footage, and vehicle counts. Highlighting the substantial impact of traffic flow on particle levels, the research leverages data fusion techniques, necessitating careful initial modeling, to optimize sensor use and generate comprehensive pollution data. The contributions of this study are twofold: (1) it demonstrates how to build a sheaf on a simplicial complex using data from cameras and sensor streams; (2) it applies filtration as a self-adjusting sensor fusion technique to calculate the most reliable air quality value.

2 Overview

This study mainly focuses on the algorithm, simulating hardware results for better input signal control, and explores the sheaf modeling's increased flexibility and accuracy in air quality monitoring. Our sheaf-theoretic approach is profoundly guided by the work of Joslyn et al. [14]. However, we modify this methodology to exploit the correlation between vehicle count and air quality indices, fusing data from various sensor types such as cameras and dust sensors.

Table 1. Emission factors of typical vehicles in Vietnam

	Emission Factor $(g * km^{-1})$
Two-wheeled vehicle	0.047
Four-wheeled vehicle	0.117

Vehicle counting and PM2.5 concentration conversion algorithms. We simulate the vehicle counting algorithm, adapted from Le et al.'s work [17], which accurately and efficiently counts vehicles in real-time, offering valuable data to analyze the relationship between PM2.5 concentration and traffic density. Based on the vehicle counts, a formula exists for calculating the PM2.5 emissions from vehicles, as described by Phung, Nguyen Ky, et al. [23]:

$$E_m = N_m \times EF_m \times VKT_m, \tag{1}$$

where E_m represents the mass of emitted PM2.5 (g), N_m denotes the number of vehicles of type m, EF_m is the emission factor for vehicle type m $(g * km^{-1})$, and VKT_m signifies the length of the recorded street segment (km). The emission factor utilized in our calculations can be found in the California Air Resources Board Emission Factor Table [20], which is condensed into two typical vehicle types in Vietnam for our implementation, as shown in Table 1. By using this factor and formula 1, we can compute the PM2.5 concentration by dividing the emission rate by the volume of air in which the particles are dispersed. In our calculations, we assume that the PM2.5 particles are uniformly distributed in a cube with a side length equal to the vehicle kilometers traveled (VKT). This enables us to accurately estimate the PM2.5 concentration in a specific area and time period as follows:

$$C_m = E_m \times 10^6 / (VKT_m \times 1000)^3, \tag{2}$$

where C_m is the target PM2.5 concentration $(\mu g/m^3)$.

We have a formula linking vehicle numbers to PM2.5 concentration, but PM2.5 from vehicles isn't immediately detected by sensors; there's a delay. Le et al. [17] developed a method calculating the average delay, and we refine this approach using cross-correlation to determine the lag between vehicle and PM2.5 signals. After each 24-hour period, the signals are associated to find the most viable lag, which is then used for shifting the converted signal the next day. Vehicles aren't the sole PM2.5 source, so we estimate other sources' contributions, assuming they remain constant for a given hour. We calculate the pattern for one day and use it for the following day. The total PM2.5 concentration is derived by adding the pattern to the PM2.5 concentration from vehicles. The following algorithm summarizes the process of converting vehicle counts to the total PM2.5 concentration.

Algorithm 1 Vehicle2PM2.5 (VI2PM) algorithm

```
Require: $N_m$: Number of vehicles of every type $m$
Ensure: $P_{v,t,h}$: PM2.5 concentration of emission from vehicles in the current day $t$ of an hour $h$
        $P_{T,t,h}$: aligned total PM2.5 concentration from the number of vehicles in the image in
  the current day $t$ of an hour $h$
  $\{E_m\} \leftarrow N_m \times EF_m \times VKT_m$ for $m$ in $M$
  $\{C_m\} \leftarrow E_m \times 10^6/(VKT_m \times 1000)^3$ for $m$ in $M$
  $P_{v,t,h} \leftarrow \sum_m^M C_m$
  $P_{T,t,h} \leftarrow P_{b,t-1,h} + P_{v,t,h-L}$
  return $P_{v,t,h}$, $P_{T,t,h}$
```

Specifically, the algorithm takes inputs such as an image, a vector of target vehicle types, the lag between camera-converted and sensor PM2.5 index, and the base PM2.5 concentration of other emission sources from the previous day. It then outputs the PM2.5 concentration of emissions from vehicles and the aligned total PM2.5 concentration for the current day t and hour h. The latter two arguments are recalculated once at the end of the day based on Algorithm 2 to prepare for the next calculation round of Algorithm 1. In Algorithm 1, there are some intermediate variables that have not been specified, such as $\mathbf{O}$, which is the vector of extracted foreground objects, and $\mathbf{M}_i : \mathbf{O}_j$, which represents a dictionary listing desired vehicle types and their corresponding list of belonging objects. N_m, E_m, and C_m are the previously introduced factors, which represent the number of vehicles of type m, the corresponding preliminary PM2.5 emission, and the target PM2.5 concentration over a given area and time, respectively.

Algorithm 2 PM2.5 concentration base from other sources than vehicles and lag between camera-converted and sensor PM2.5 index calculation algorithm

```
Require: $P_{v,0...23}$: PM2.5 concentration of emission from vehicles spreading over 24 hours,
         $P_{s,0...23}$: PM2.5 concentration of emission from an air sensor spreading over 24 hours
Ensure: $lag$: lag between camera-converted and sensor PM2.5 index
        $P_{b,0...23}$: base PM2.5 concentration of other emission sources spreading over 24 hours
  $\{lag\} \leftarrow$ Cross-correlation$(P_{v,0...23}, P_{s,0...23})$
  $lag_{max} \leftarrow$ max$(\{lag_i\})$
  $P_{b,0...23} \leftarrow P_{s,0...23} - P_{v,0-lag_{max}...23-lag_{max}}$
  return $lag$, $P_{b,0...23}$
```

3 Sheaf-theoretic self-filtering network modeling

Various tracking methods exist for integrating heterogeneous information, with aggregation being common but imprecise for highly dissimilar data points. Sheaf-based tracking models offer an alternative, aggregating specific quantities and enabling accurate heterogeneous information calculation by individually evaluating data points. This simplifies the process compared to pooling models, as evaluating a point in a sheaf only requires locating the corresponding point in space.

A sheaf is a data structure storing information over a topological space, with the space outlining sensor relationships and the sheaf mapping sources into a common comparison framework. An abstract simplicial complex (ASC) represents sensors as vertices and their multi-way interactions as higher-order faces. The ASC forms the sheaf's base, while the faces of the stalks store information. Restriction functions model interactions between sensor combinations and data, completing the sheaf model. Assignments are defined as recorded readings, sections as consistent assignments based on the sheaf model, and partial assignments and sections over a subset of sensors.

Then, we present the following strategies for sheaf modeling. Specifically, when global or partial sections suggest entirely consistent data, we add *consistency structures* to describe data that are only partially consistent. *Consistency structures* instantiated for the sensor nodes model take the form of n-way standard deviations, albeit being specified entirely broadly. In turn, *consistency structures* enable us to create approximation sections that can quantify the degree of sensor consistency. The *consistency radius* is a native global measure of the uncertainty among the sensors present in any given reading. Moreover, *consistency filtration* [27] provides a comprehensive description of the contributions of individual sensors and sensor combinations to the total uncertainty.

3.1 Sheaf simplicial constructions

We start by defining an abstract simplicial complex, the type of topological space used to represent our sensor network. Usually, specifying all of the simplices in a simplicial complex is laborious. Rather, it is much more efficient to provide a generating set X of subsets of the vertex set. The abstract simplicial complex formed by X is the unique smallest simplicial complex containing the generating set, and formally defined, according to [14], as follows.

Definition 1 (Abstract Simplicial Complex). *An abstract simplicial complex X on a set V_X is a collection of subsets of V_X, where if $\sigma \in X$ and $\gamma \subseteq \sigma$, then $\gamma \in X$. Each $\sigma \in X$ is a simplex of X, and each element of V_X is a vertex of X. Any subset γ of a simplex σ is called a* ***face*** *of σ. More generally, each $\sigma \in X$ with d+1 elements is referred to as a* ***d-face*** *of X, where* **d** *represents its dimension.* ***Vertices*** *are zero-dimensional faces (singleton subsets of V_X), and* ***edges*** *are one-dimensional faces.*

The following explains how an abstract simplicial complex can be used to represent connections within a sensor network. Let the base set V_X be the set of sensors in the network, and let X include every set of sensors that measure the same quantity. Based on the sensor network configuration, the following labels are assigned to the four sensors,

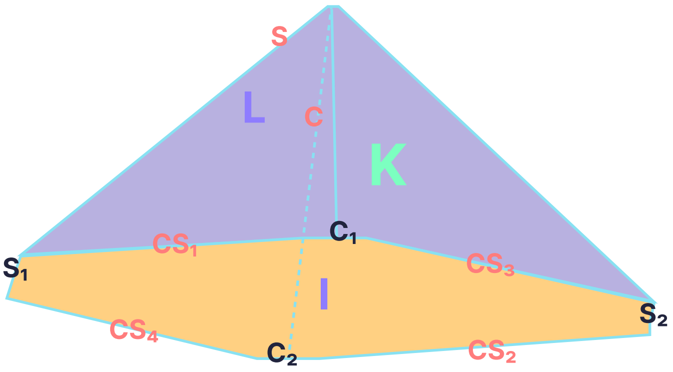

Fig. 1. Simplicial complex of our network of air monitoring sensors.

including two camera nodes and two dust particle sensory nodes, which are connected to monitor the local air quality signal: C_1, C_2, S_1, and S_2, respectively. Since all four sensors contribute to a single final piece of information, the PM2.5 index, we define $U = C_1, C_2, S_1, S_2$ as our base sensor set, where $C = C_1, C_2$ is the set of cameras that count and classify traffic, and $S = S_1, S_2$ includes two air sensor devices that measure the PM2.5 index of the air. In this manner, our air-monitoring network's ASC X has a total of ten faces, counting both C and S and their subsets:

$$\begin{aligned} X = \{\{C_1,C_2\},\{S_1,S_2\}, \\ \{C_1,S_1\},\{C_2,S_2\},\{C_1,S_2\},\{C_2,S_1\}, \\ \{C_1\},\{S_1\},\{C_2\},\{S_2\}\} \end{aligned} \tag{3}$$

The remaining pairwise sensor interaction faces are represented as $CS_1 = C_1, S_1$, $CS_2 = C_2, S_2$, $CS_3 = C_1, S_2$, $CS_4 = C_2, S_1$; therefore, the entire sheaf diagram is illustrated in Fig. 1, where the highest dimensional face (the final air PM2.5 index K) and all sub-faces are shown. The sensors are displayed as black singleton faces, while the higher-dimensional faces are shown in red. Additionally, the solid rectangle represents the three-way interaction I.

The Abstract Simplicial Complex (ASC) represents a topological space that describes complex interactions among sensor data. It consists of hypertetrahedrons that form a single connected component, but the complexity can vary depending on the quantity and configuration of observables. Sheaf theory accounts for all interactions and provides a directed acyclic graph (DAG) 2 with nodes representing the faces of the ASC and directed edges ascending from a face to its corresponding higher-dimensional co-face.

In essence, a *sheaf of vector spaces* assigns a vector space to each face and a linear map to every attachment analogously. Moreover, a *sheaf of groups* designates a group to each face and a group homomorphism to every attachment. Specifically, the stalk above

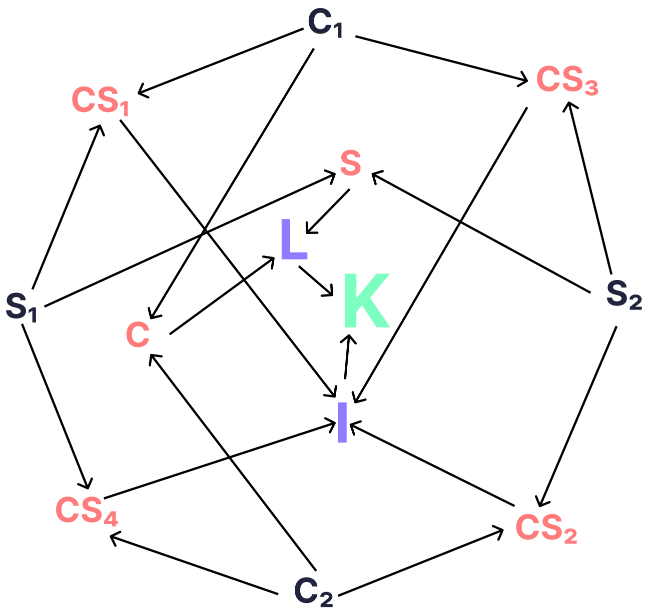

Fig. 2. Corresponding attachment diagram of our network of air monitoring sensors.

a face constitutes the locus where data associated with that face is situated. Restriction functions establish the foundation upon which interacting data can be deemed consistent or inconsistent.

Concerning our air monitoring network, we have two cameras that encode vehicle counts over time from the input video stream; thus, the stalks above vertices $C1$ and $C2$ are $\mathbb{R}$. Similarly, the stalks above the vertices of the sensory nodes are also $\mathbb{R}$, as they directly monitor PM2.5 levels. With the stalks for the vertices defined, the stalks for the higher-order faces and the restriction maps must be determined. Nevertheless, the data formats are incompatible. Consequently, to compare these measures, the information should first be transformed into standard units along the edges, and subsequently conveyed to the rectangular face. We opted for PM2.5 as our common coordinate system, given that most sensors report their readings in this format. Hence, $\mathbb{R}$ serves as the foundation for the higher-order facets of the rectangle. Utilizing a guide book in [23], we non-linearly convert vehicle counts from the camera vertices to the common unit in all other faces. Consequently, the attachment diagram in Fig. 2 evolves into our sheaf model for the air monitoring network, depicted in Fig. 3.

The sheaf model characterizes the temporal alignment of sensors as delineated below. At a specific time t, each vertex is assigned the most recent sensor reading, a data point from its stalk space. These measurements are subsequently transmitted to the higher-order faces for comparison through the restriction maps. If the two measurements obtained by an edge are identical, this single value is attributed to that edge, and

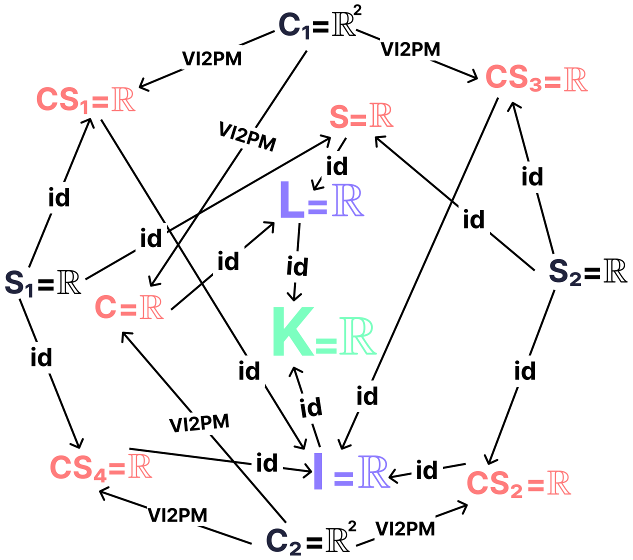

Fig. 3. Sheaf model of the air monitoring network. GB stands for guide book, which is used to translate from the vehicle numbers to PM2.5. The other abbreviation, id, stands for "identity", which then refers to the identity transformation, or in the sheaf language, the identity restriction map.

the algorithm advances. During this process, an agreement occurs when a single value is established for faces of the same type, e.g., Cameras (the same value is set for C, C_1, and C_2). When there is the potential for a complete assignment, we refer to it as a *global section*. The possibility of disagreement always exists, which is where the concept of an *assignment* originates. This lays the foundation for the following definition, according to [14].

Definition 2 (Assignment and global section). *Let $\mathscr{S}$ be a sheaf on an abstract simplicial complex X. A function $\gamma : X \to \Pi_{x\in X}\mathscr{S}(x)$ that assigns a value $\gamma(x) \in \mathscr{S}(x)$ to each face $x \in X$ is defined as an* ***assignment****. Subsequently, the definition of a* ***partial assignment*** *is constrained by that of an* ***assignment****: a function $\alpha : X' \to \Pi_{x\in X'}\mathscr{S}(x)$ that assigns a value $\gamma(x) \in \mathscr{S}(x)$ to each face $x \in X' \subset X$. An assignment s is denoted as a* ***global section*** *if, for every inclusion $x \rightsquigarrow \beta$ of faces, $\mathscr{S}(x \rightsquigarrow \beta)(s(x)) = s(\beta)$.*

At a specific instant, a global section of our tracking model corresponds to the sensor data concurrently concurring. The equality condition of a global section might be overly restrictive for certain applications, including our tracking model.

3.2 Loosening Section constraints with Consistency Structures

Consistency structures relax the constraint of matching sensor data by allowing agreement between data points rather than demanding a singular global section. This approach uses a boolean function for each face to assess the level of agreement, forming a consistency structure when combined with a sheaf. Formally, a consistency structure is defined, according to [14], as follows:

Definition 3 (Consistency structures). *A consistency structure is defined as a triple (X, $\mathscr{S}$, **C**) in which X is an abstract simplicial complex, $\mathscr{S}$ is a sheaf over X, and **C** is the assignment to each non-vertex d-face $\beta \in X, d > 0$, of the following function:*

$$\boldsymbol{C}_\beta : \left(\!\!\left(\begin{matrix}\mathscr{S}(\beta)\\ dim\beta+1\end{matrix}\right)\!\!\right) \rightarrow \{0,1\},$$

where the double bracket pairs derive the set of subsets with length $\beta + 1$ over $\mathscr{S}(\beta)$.

The multiset in the domain of C_β represents the various sheaf values to be compared for all vertices impinging on a non-vertex face β, while the codomain $0,1$ indicates whether or not they match ”sufficiently.” Formally, a **conventional consistency structure** for a sheaf $\mathscr{S}$ applies an equality test to each face that is not a vertex $\beta = v_1, v_2, \dots, v_k$, according to [14]:

$$\mathbf{C}_\beta([z_1, z_2, \dots, z_k]) = \begin{cases} 1, & if\ z_1 = z_2 = \cdots = z_k \\ 0, & otherwise, \end{cases} \tag{4}$$

where the square bracket pair represents the multiset and $z_j = \mathscr{S}(v_i \rightsquigarrow \beta)(s(v_j))$. A consistency structure extends the equality criterion inherently available in a sheaf to encompass classes of values that are considered equivalent. Moreover, each stalk in our tracking model is a metric space. As such, we can use the natural metric to determine if two points are merely ”close enough” rather than entirely congruent or comparable. Utilizing ε to represent the amount of error present or acceptable in an assignment, we construct the **ε-approximate consistency structure** for a sheaf $\mathscr{S}$ and each of the non-vertex $\beta = v_1, v_2, \dots, v_k, k > 1$, according to [14], as follows:

$$\mathbf{C}_\beta([z_1, z_2, \dots, z_k]) = \begin{cases} 1, & if\ \delta([z_1, z_2, \dots, z_k]) \leq \varepsilon \\ 0, & otherwise, \end{cases} \tag{5}$$

where the function δ serves to measure the consistency as a general dispersion of the multivariate data. Specifically, it is defined, according to [14], as:

$$\delta(Y) = \sqrt{\frac{1}{|Y|}\mathbf{Tr}(\Sigma_Y)}, \tag{6}$$

where Σ_Y represents the covariance matrix of the multidimensional data Y.

Subsequently, a consistency structure’s counterpart to a global section for sheaves is referred to as a pseudosection. A pseudosection assignment s ensures that, for each non-vertex face β, (i) the restrictions of its vertices to the face are adequate, and (ii) the value assigned to the face is congruent with the restricted vertices. Formally, the pseudosection definition can be delineated, according to [14], as follows:

Definition 4 (Pseudosection). *For each non-vertex $\beta = \{v_1, v_2, \dots, v_k\}$, an assignment $s \in \Pi_{x\in X}\mathscr{S}(x)$ is defined as a $(X, \mathscr{S}, \boldsymbol{C})$-pseudosection if*

i. $\boldsymbol{C}_\beta([\mathscr{S}(\{v_i\} \rightsquigarrow \beta)s(\{v_i\}) : i = 1,\dots,k]) = 1$, *and*

ii. $\boldsymbol{C}_\beta([\mathscr{S}(\{v_i\} \rightsquigarrow \beta)s(\{v_i\}) : i = 1,\dots,j-1,j+1,\dots,k]) \cup s(\{v_i\})) = 1$,

$for\ j\ in\ 1,2,\dots,k.$

Consequently, the pseudosections of a conventional consistency structure correspond to the global sections of its associated sheaf because when **C** represents the conventional consistency structure, the condition for an assignment for each non-vertex face $\beta = v_1, v_2, \dots, v_k$ to become a pseudosection is as follows:

$s(\beta) = \mathscr{S}(v_1 \rightsquigarrow \beta)s(v_1) = \mathscr{S}(v_2 \rightsquigarrow \beta)s(v_2) = \cdots = \mathscr{S}(v_k \rightsquigarrow \beta)s(v_k).$

In contrast, if **C** represents an ε-approximate consistency structure, pseudosections for each non-vertex face $\beta = v_1, v_2, \dots, v_k$ are defined as:

$$\begin{cases} \delta([\mathscr{S}(\{v_i\} \rightsquigarrow \beta)s(\{v_i\}) : i = 1,\dots,k]) \leq \varepsilon \\ \delta([\mathscr{S}(\{v_i\} \rightsquigarrow \beta)s(\{v_i\}) : i = 1,\dots,j-1,j+1,\dots,k]) \cup s(\{v_i\})) \leq \varepsilon, \\ for\ j\ in\ 1,2,\dots,k. \end{cases} \tag{7}$$

A pseudosection of our tracking model's ε-approximate consistency structure allocates the PM2.5 signals in a manner that the ”spread” of all measurements attributed to each face is constrained by ε. As per the subsequent theorem found in [14], the minimal ε for which a pseudosection arises is exclusively determined by constraining the images of the vertices.

Theorem 1 (Consistency Radius). *Let $\mathscr{S}$ be a sheaf on an abstract simplicial complex X where each stalk is a metric space and let s be an assignment belonging to the set $\Pi_{x\in X}\mathscr{S}(x)$. The minimal ε such that s is a pseudosection of the ε-approximate consistency structure $(X, \mathscr{S}, \boldsymbol{C})$ is given by*

$$\varepsilon^* = \max_{\beta \in X \setminus \{\{v\} : v \in V\}} \delta([\mathscr{S}(w \rightsquigarrow \beta)s(\{w\}) : w \in \beta]),$$

where V represents the set of vertices of the sheaf, and the proof can be found in [14].

Lemma 1 (Consistency Radius). *Should a set of real numbers $Z = \{z_1, z_2, \dots, z_k\}$ that has a mean of μ_Z, $\forall z \in Z, \delta(Y_z) \leq \delta(Z)$, where $Y_z = Z \setminus z \cup \mu_Z$.*

The consistency radius prompts the examination of the network system's consistency as ε varies from its minimum to the defined radius, given data from multiple sources. Identifying maximally consistent sections enables the introduction of key metrics like consistency filtration and cover measures.

Theorem 2 (Maximally Consistent Subcomplexes). *Let a consistency structure $(X, \mathscr{S}, \boldsymbol{C})$ be with a sheaf partial assignment s on vertices $\mathscr{U}$. There exists a unique group of subsets $\{W_i\}$ of $\mathscr{U}$ that leads to subcomplexes $\{X_{W_i}\}$ of X with the following characteristics:*

i. The assignment s is consistent within each X_{W_i}, and any subcomplex where s is consistent also has at least one X_{W_i} as a supercomplex.

ii. $\bigcup star(X_{W_i})$ is a cover of X, where $star(X_{W_i})$ is defined as the set of faces containing X_{W_i}, formally $star(X_{W_i}) = \{k \in X : X_{W_i} \subseteq k\}$.

Lemma 2 (Maximally Consistent Subcomplexes). *Let a consistency structure $(X, \mathscr{S}, \boldsymbol{C})$ be with a sheaf partial assignment s on $\mathscr{U}$. if for some non-vertex face βinX such that $C_\beta(\mathscr{S}(\{v\} \rightsquigarrow \beta)s(\{v\})) = 0$, there exists subsets $\{W_i\}$ of $\mathscr{U}$ which leads to the subcomplexes $\{X_{W_i}\}$ of X such that:*
i. $\beta \notin \{X_{W_i}\}$ for all i
ii. The assignment s is consistent within each X_{W_i}, and any subcomplex where s is consistent also has at least one X_{W_i} as a supercomplex.
iii. $\bigcup star(X_{W_i})$ is a cover of X, where $star(X_{W_i})$ is defined as the set of faces containing X_{W_i}, formally $star(X_{W_i}) = \{k \in X : X_{W_i} \subseteq k\}$.

Theorem 2 offers a maximal vertex cover for our tracking model when supplied with an assortment of sensor readings and a specific ε value, ensuring that each corresponding subcomplex maintains approximate consistency within a predetermined error margin. To establish a measure for the vertex cover associated with a collection of maximal consistent subcomplexes, we regard the set of covers as a graded poset and employ the poset's rank function as a metric.

Definition 5 (Graded posets and rank function). *Let $\mathscr{P} = \langle P, \trianglelefteq \rangle$ be a poset, it is **graded** if there exists a **rank** function $r : P \to \mathbb{N} \cup \{0\}$ such that $r(s) = 0$ if s is a minimal element of P as well as $r(j) = r(k) + 1$ if $j \lhd k$ in P. s is said to have rank i, if $r(s) = i$ and the maximum rank, which is defined as $\max\limits_{p \in P}\{r(p)\}$, stands for the rank of $\mathscr{P}$.*

Ultimately, the rank function can be employed to quantify the vertex cover connected to a collection of maximally consistent subcomplexes, as elaborated in the comprehensive proof presented in [14]. The subsequent definition delineates the manner in which this process unfolds.

Definition 6 (Graded posets and rank function). *Let a consistency structure $(X, \mathscr{S}, \boldsymbol{C})$ that has the number of vertices $|V| = n$ and $\mathscr{A} = \{W_i\}$ be a vertex cover obtained from Theorem 2. The measure of such vertex cover is defined as*

$$\bar{r}(\mathscr{A}) = |\downarrow \mathscr{A}| - (n+1)$$

Note that $0 \leq \bar{r}(\mathscr{A}) \leq 2^n - (n+1)$, and a bigger number suggests subcomplexes with more consistency and given $\downarrow \mathscr{A}_1 \subseteq \downarrow \mathscr{A}_2$ entails $\bar{r}(\mathscr{A}_1) \leq \bar{r}(\mathscr{A}_2)$.

Ultimately, the rank function can be employed to quantify the vertex cover connected to a collection of maximally consistent subcomplexes, as elaborated in the comprehensive proof presented in [14]. The subsequent definition delineates the manner in which this process unfolds.

3.3 Consistency Filtrations

Thus far, we have established a simplicial complex X, a sheaf $\mathscr{S}$ on the simplicial complex, a partial assignment s to the vertices of X, and an ε-approximate consistency structure $(X, \mathscr{S}, \mathbf{C}\varepsilon)$. These components facilitate the construction of a consistency filtration by varying ε and, owing to Theorem 2, enable the acquisition of the set of maximal consistent subcomplexes. The filtration of vertex covers corresponds to landmarks spanning over ε values enumerated as $\varepsilon_0 = 0 < \varepsilon_1 < \cdots < \varepsilon t - 1 < \varepsilon_t = \varepsilon^*$ – the consistency radius, which represents the smallest feasible ε such that it renders the assignment s a $(X, \mathscr{S}, \mathbf{C}\varepsilon)$-pseudosection. The series of vertex covers, $C_0 \trianglelefteq C_1 \trianglelefteq \cdots \trianglelefteq Ct - 1 \trianglelefteq C_t$, can be refined such that each set comprises subcomplexes whose union constitutes X. Furthermore, the sequence of cover measures, $p_0 \trianglelefteq p_1 \trianglelefteq \cdots \trianglelefteq p_{t-1} \trianglelefteq p_t$, can be computed for each set of covers, C_i. The consistency filtration serves as a technique for assessing the consistency among a group of sensors. If the distance between two consecutive landmark values, ε_i and ε_{i+1}, is considerably larger than the other distances, this suggests a disagreement between at least two groups of sensors. By contrasting the covers C_i and C_{i+1}, it becomes straightforward to identify which sensors are responsible for the disagreement.

4 Integrated algorithm

The algorithm delineated in this section constitutes a crucial step in comprehending the temporal behavior of our system. Utilizing the information from the nodes at a specific time, we can ascertain the system's consistency filtration, a metric indicative of coherence and stability at that instance. This enables tracking the system's evolution and pinpointing potential issues or instabilities.

The system asynchronously populates data to each node in the set C_1, C_2, S_1, S_2, with camera nodes receiving data every 10 minutes and air sensor nodes every 15 seconds. The data is initially retrieved by the sheaf's lowest-ranking face and subsequently disseminated, or "lifted up," to higher faces level by level until reaching the highest-ranking face. Pertaining to the sheaf model depicted in Fig. 3, the second level encompasses the set of edges $CS_1, CS_2, CS_3, CS_4, C, S$, while the highest level is K. It is important to note that K is represented as a 3-dimensional space constituted by 2-dimensional surfaces I and L. Each time the data is lifted within the sheaf model, the corresponding conventional consistency structure must exhibit pairs of adjacent nodes/edges that are strictly equal. Consequently, the values of all high-level faces in the sheaf model must be zero, forming a pseudosection of the conventional consistency structure or a global section of the sheaf model. However, as previously mentioned, achieving a global section in real-world applications is unrealistic due to the inherent imperfections of any working system; thus, an ε-approximate consistency structure is employed to permit a degree of errors among the faces. The propagated values along the faces in the consistency structure are described in Eq. 6.

Owing to the definition of consistency radius and consistency filtration, it is possible to determine which sensor groups are consistent with one another. Similar to the work of Joslyn et al. [14], we can employ the consistency measure to identify faulty sensing nodes and decide to isolate their results prior to examining their status. In summary, the entire process can be determined through the following algorithm, specifically tailored for the sheaf model in Fig. 3.

The algorithm uses data aggregation to evaluate sheaf model consistency with a mean operator. However, it's not a simple aggregation method as it (i) integrates data from heterogeneous sensors at different times, and (ii) applies the mean operator hierar-

Algorithm 3 Self-consistency management from heterogeneous input sources algorithm

```
Require: $\mathscr{S}$: sheaf model, $\{C_1, C_2, S_1, S_2\}$: vertices set
Ensure: $\varepsilon_c$: consistency threshold,
        $V_c$: value of the face at $\varepsilon_c$,
        $S_c$: a set of consistent sensors
  $0\text{-}faces \leftarrow \{C_1, C_2, S_1, S_2\}$    ▷ Original vertices, or $0\text{-}faces$
  $CS_1() \leftarrow \{\mathscr{S}(C_1 \rightsquigarrow 1\text{-}faces)s(\{C_1\}), \mathscr{S}(S_1 \rightsquigarrow 1\text{-}faces)s(\{S_1\})\}$
  $CS_2() \leftarrow \{\mathscr{S}(C_2 \rightsquigarrow 1\text{-}faces)s(\{C_2\}), \mathscr{S}(S_2 \rightsquigarrow 1\text{-}faces)s(\{S_2\})\}$
  $CS_3() \leftarrow \{\mathscr{S}(C_1 \rightsquigarrow 1\text{-}faces)s(\{C_1\}), \mathscr{S}(S_2 \rightsquigarrow 1\text{-}faces)s(\{S_2\})\}$
  $CS_4() \leftarrow \{\mathscr{S}(C_2 \rightsquigarrow 1\text{-}faces)s(\{C_2\}), \mathscr{S}(S_1 \rightsquigarrow 1\text{-}faces)s(\{S_1\})\}$
  $C() \leftarrow \{\mathscr{S}(C_1 \rightsquigarrow 1\text{-}faces)s(\{C_1\}), \mathscr{S}(C_2 \rightsquigarrow 1\text{-}faces)s(\{C_2\})\}$
  $S() \leftarrow \{\mathscr{S}(S_1 \rightsquigarrow 1\text{-}faces)s(\{S_1\}), \mathscr{S}(S_2 \rightsquigarrow 1\text{-}faces)s(\{S_2\})\}$
  $1\text{-}faces \leftarrow \{CS_1, CS_2, CS_3, CS_4, C, S\}$    ▷ Configuring $1\text{-}faces$
  $I \leftarrow \{\mathscr{S}(CS_1 \rightsquigarrow 2\text{-}faces)s(\{CS_1\}), \mathscr{S}(CS_2 \rightsquigarrow 2\text{-}faces)s(\{CS_2\}), \mathscr{S}(CS_3 \rightsquigarrow 2\text{-}faces)s(\{CS_3\}), \mathscr{S}(CS_4 \rightsquigarrow 2\text{-}faces)s(\{CS_4\})\}$
  $L \leftarrow \{\mathscr{S}(C \rightsquigarrow 2\text{-}faces)s(\{C\}), \mathscr{S}(S \rightsquigarrow 2\text{-}faces)s(\{S\})\}$
  $2\text{-}faces \leftarrow \{I, L\}$    ▷ Configuring $2\text{-}faces$
  $K \leftarrow \{\mathscr{S}(I \rightsquigarrow 2\text{-}faces)s(\{I\}), \mathscr{S}(L \rightsquigarrow 2\text{-}faces)s(\{L\})\}$
  $3\text{-}faces \leftarrow \{K\}$    ▷ Configuring $3\text{-}faces$
  $V \leftarrow \{\}$    ▷ Initializing an empty dictionary that stores complexes' values
  $\varepsilon \leftarrow \{\}$    ▷ Initializing an empty dictionary that stores complexes' corresponding $\varepsilon$ threshold
  while $d \leq 3$ do
      $\Omega \leftarrow d\text{-}faces$
      if $d > 0$ then
          for $\omega \in \Omega$ do
              $\{\omega_1, \omega_2 \ldots, \omega_n\} \leftarrow \omega()$
              $V[\omega] \leftarrow \{\omega_1, \omega_2, \ldots, \omega_n\}$
              $\varepsilon[\omega] \leftarrow \delta(\{\omega_1, \omega_2, \ldots, \omega_n\})$    ▷ $\delta(.)$ is determined in Eq. 6
          end for
      else
          $V[\omega] \leftarrow s(\omega)$    ▷ Initializing with partial assignment
          $\varepsilon[\omega] \leftarrow 0$
      end if
  end while
  $S_c \leftarrow []$
  $\varepsilon_c \leftarrow 0$
  $V_c \leftarrow 0$
  for $\omega \in \Omega$ do
      if $\varepsilon[\omega] \leq mean(\varepsilon) + 0.5 * std(\varepsilon)$ then    ▷ Checking if the consistency threshold belong to such part of the filtration
          $S_c.append(\omega)$
          $\varepsilon_c \leftarrow \varepsilon[\omega]$
          $V_c \leftarrow V[\omega]$
      end if
  end for
  return $\varepsilon_c$, $V_c$, $S_c$
```

chically to the most consistent sensors based on the filtration distribution. More details can be found in Section 5.

5 Toy Examples

This section uses toy examples with simulated sensor measurements and generated data to illustrate sheaf modeling, helping readers grasp the main concepts and practical applications in data analysis and interpretation.

5.1 Example 1: Sheaf Global Section or Consistency Structure Pseudo-section

In this example, we demonstrate the sheaf behavior with a global section or pseudo-section, considering the consistency structure, where all assignments seamlessly fit together, forming a cohesive global structure. This scenario provides insight into how the sheaf represents and analyzes data when local assignments perfectly align.

Specifically, we code up our sensor network sheaf model, illustrated in Fig. 3. We use the Python *networkx* package to visualize the sheaf in Fig. 4.

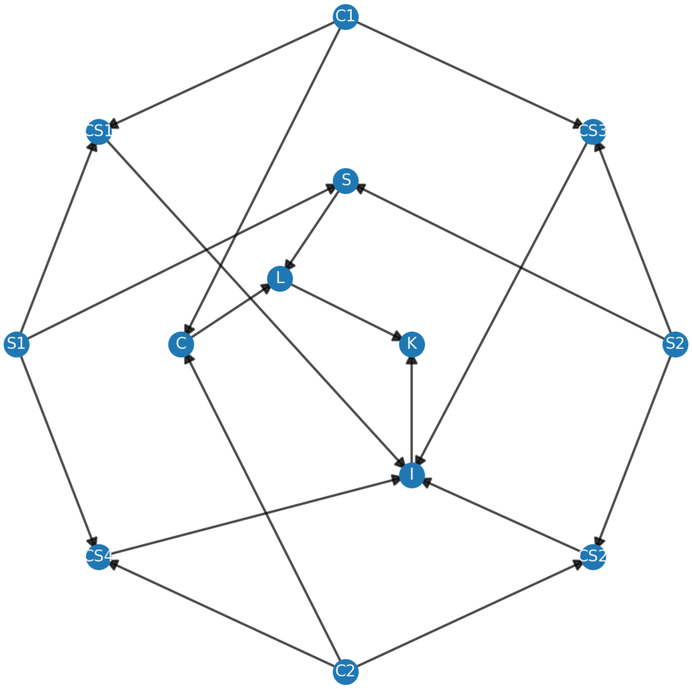

Fig. 4. Sheaf model of the air monitoring network, drawn by the Python *networkx* package.

Table 2. Sheaf consistency structure pseudo-section assignment

	C1	C2	S1	S2
Assignments	[200,30]	[200,30]	12.91	12.91

Then, we give assignments to the vertices, such that all of them can be inferred to match perfectly with each other. The assignments are demonstrated in Table 2, where we have two unique assignments such as 12.91 and $[200, 30]$, in which 200 stands for the number of motorbikes and 30 stands for the number of cars and they then can be inferred as exactly 12.91 by the numbers of emission factors shown in Table 1. Because the values are perfectly matched right on the vertices, the values propagated to the higher faces are also consistent and therefore completely matched. This situation is called a global section of the sheaf model or the pseudo-section of the consistency structure of the sheaf. All the consistency thresholds in the consistency filtration equal to 0 as a result. Hence, the filtration in this case is a dictionary, as such

```
{'C': 0.0,
 'S': 0.0,
 'CS1': 0.0,
 'CS2': 0.0,
 'CS3': 0.0,
 'CS4': 0.0,
 'L': 0.0,
 'I': 0.0,
 'K': 1.8553442084620055e-15
}
```

5.2 Example 2: Sheaf Data Aggregation with simulated signals

In this example, there are four nodes that are being used to measure a simulated signal. These nodes consist of two cameras and two sensors. The cameras have a sampling frequency rate of once every 600 seconds, while the sensors have a sampling frequency rate of once every 15 seconds. The goal is to demonstrate how sheaf-based data fusion outperforms naive data aggregation, providing more accurate results.

We create a simulation of the actual PM2.5 signal by using a sinusoidal function that varies over time, with the unit of time being seconds, and limited in the range between 100 and 200 by incorporating coefficients to the sinusoidal function, such as $y = 50sin(x) + 150$. The simulation covers a period of 48 hours, illustrated in Fig. 5. Then, to simulate the measurement of the sensor signals, we add different levels of Gaussian noise to each type of sensor. The noise levels for the sensor nodes C1, C2, S1, and S2 are 2.8%, 8.3%, 11.7%, and 16.9%, respectively. Additionally, the sensors have different resolutions, so their measurements are not in sync. To ensure that we can propagate the values across the sheaf faces, we need to fill in the values for the vertices every time we update the measurements. To handle the misalignment of the measurements, we assign the previous values to the vertices that have not received a new measurement when we update. The data from the sensors are depicted in Fig. 6.

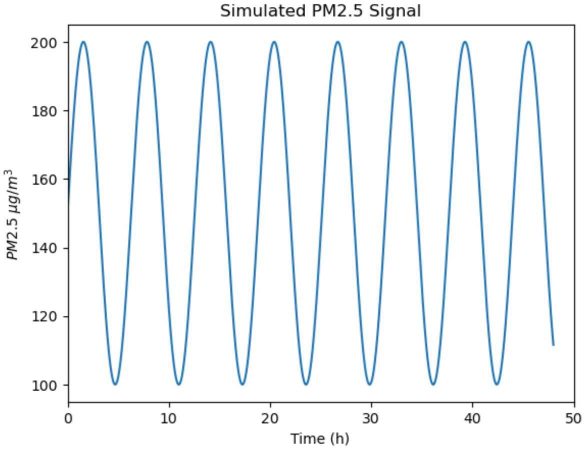

Fig. 5. Simulated PM2.5 signal by using a sinusoidal function that varies over time.

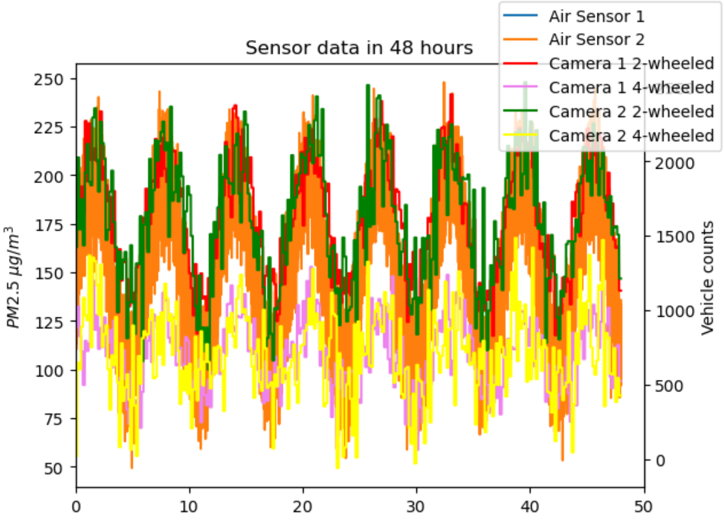

Fig. 6. Simulated PM2.5 sensor measurement over time.

In this comparison, we analyze the sheaf data averaging with cut-off along with baseline methods including naive data averaging and Kalman filter using simulated sensor signals. The outcomes presented in both Table 3 and Fig. 7 indicate that the performance of the Kalman Filter is inferior (with an error rate of (22.81%) to that of all individual sensors when juxtaposed with the actual value. The naive approach averages sensor values at each update, while the sheaf modeling system aggregates only consistent faces based on a threshold. Both methods improve overall errors (measured by Mean Absolute Percentage Error), but the sheaf with cut-off reduces errors by 12.43% more than naive averaging. Figure 7 highlights the lower error values for the sheaf approach (orange line) compared to naive averaging (blue line) over an extended simulation period, which is more evident in their moving averages.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N} \frac{|y - \hat{y}|}{y} \times 100 \tag{8}$$

Table 3. Average errors of individual sensors and data aggregation approaches

	MAPE (%)
C1	11.74%
C2	16.93%
S1	2.83%
S2	8.25%
Kalman Filter	22.81%
Naive Averaging	5.92%
Sheaf Averaging with cut-off	5.18%

Apparently, by cutting off the inconsistent face out of the consistency structure of the sheaf model would reduce the overall errors of the system. We would like to demonstrate how cutting off inconsistent faces would decrease the error by analyzing the most spread filtration among the consistency filtration sets of the sheaf model over the simulation time, illustration in Fig. 8.

5.3 Example 3: Sheaf Filtration at the minute 1545.5

In Fig. 8, over time, the consistency of the filtration tends to align with the errors that each sensor produces. Usually, the face sensors $CS2$, $CS4$, and C are the most prone to errors, and these correspond to the two least accurate nodes, $C1$ and $C2$. At the second #92730, or the minute #1545.5, there is a maximum in the data spread as measured by the consistency filtration. We would like to demonstrate the impact of removing faces with high consistency thresholds. At that point of time, the filtration is a dictionary, as such

```
{'CS1': 8.283882338353894,
'S': 33.620082580239966,
'CS3': 41.903964918593864,
'K': 50.105373654118516,
```

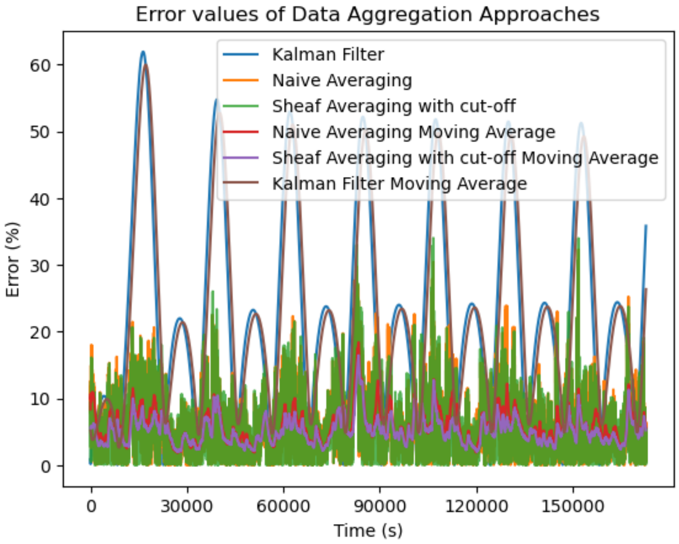

Fig. 7. Error values of Data Aggregation Approaches.

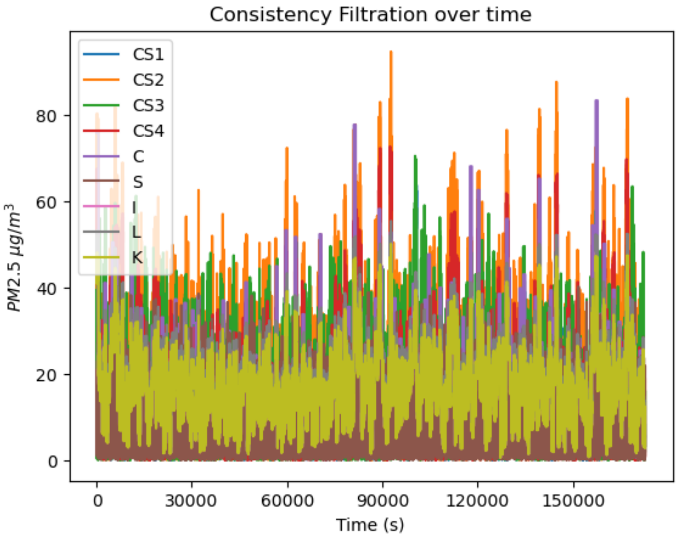

Fig. 8. Sheaf filtration of the sensor network over time.

```
'I': 51.284484945400095,
'C': 52.72299262385046,
'L': 55.393574797089926,
'CS4': 61.00687496220436,
    'CS2': 94.62695754244432}
```

, where the cut-off point is defined as the mean plus half a standard deviation of the filtration distribution, which is in this case equalled to 60.71. Figure 9 illustrates the faces in the filtration along with their values and also the cut-off value in order to visualize how the distribution is spread.

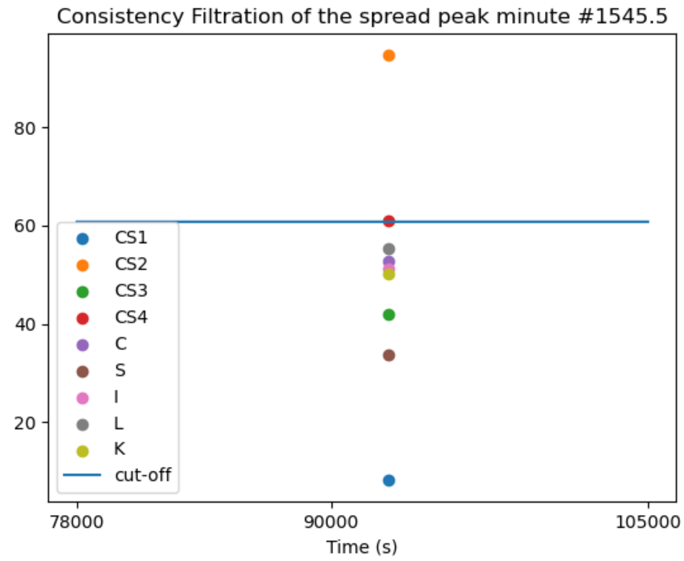

Fig. 9. Sheaf filtration spread of the sensor network at the minute 1545.5.

Visually, it seems that only the face *CS*2 is eliminated off the sheaf averaging process, but the threshold actually is 60.71, which means that the face *CS*4 lying exactly on the threshold line but it has the value of 61.0 and thus is also eliminated. This gives the results in Table 4, where sheaf averaging without the cut-off is exactly the same with the naive averaging approach. This again stresses the importance of using the consistency filtration to filter out the inconsistency faces off the averaging process.

Table 4. Filtration errors assessment

	MAPE (%)
Kalman Filter	16.07%
Naive Averaging	11.03%
Sheaf Averaging without cut-off	11.03%
Sheaf Averaging with cut-off	**8.05%**

6 Conclusion and Outlook

Sheaf theory provides increased accuracy and adaptability for modeling connections between multiple features, particularly in air quality monitoring, compared to traditional graph theory. This study introduced a self-correcting algorithm utilizing sheaf theory to account for vehicle counts affecting local air quality, achieving real-time self-correction and enabling straightforward scaling to multiple nodes. Integrating sheaf theory into air quality monitoring provides several advantages such as effectively handling heterogeneous data, reducing sample requirements, and ensuring global data consistency. However, the complexity of sheaf theory and the challenges associated with careful pre-observation modeling can serve as drawbacks.

As for future research, there's potential in merging deep learning with sheaf theory to optimize real-time monitoring and predictive analysis. Leveraging deep learning could help simplify the complex models associated with sheaf theory, improving prediction accuracy by learning intricate patterns in the data. Future work could also focus on integrating more varied data sources into the sheaf-deep learning model, such as weather and traffic data, thereby enhancing the comprehensiveness of air quality forecasts.

References

1. Abramsky, S., Barbosa, R.S., Kishida, K., Lal, R., Mansfield, S.: Contextuality, cohomology and paradox. arXiv preprint arXiv:1502.03097 (2015)
2. Abramsky, S., Brandenburger, A.: The sheaf-theoretic structure of non-locality and contextuality. New Journal of Physics **13**(11), 113,036 (2011)
3. Agency, E.E.: European environment agency's home page (2022). URL https://www.eea.europa.eu/
4. Alparone, L., Aiazzi, B., Baronti, S., Garzelli, A., Nencini, F., Selva, M.: Multispectral and panchromatic data fusion assessment without reference. Photogrammetric Engineering & Remote Sensing **74**(2), 193–200 (2008)
5. Bailer, C., Pagani, A., Stricker, D.: A superior tracking approach: Building a strong tracker through fusion. In: European Conference on Computer Vision, pp. 170–185. Springer (2014)
6. Benferhat, S., Sossai, C.: Reasoning with multiple-source information in a possibilistic logic framework. Information Fusion **7**(1), 80–96 (2006)
7. Benferhat, S., Titouna, F.: Fusion and normalization of quantitative possibilistic networks. Applied Intelligence **31**(2), 135–160 (2009)
8. Crowley, J.L., Demazeau, Y.: Principles and techniques for sensor data fusion. Signal Processing **32**(1-2), 5–27 (1993)
9. Curry, J., Ghrist, R., Nanda, V.: Discrete morse theory for computing cellular sheaf cohomology. Foundations of Computational Mathematics **16**(4), 875–897 (2016)
10. Dawn, S., Saxena, V., Sharma, B.: Remote sensing image registration techniques: A survey. In: International Conference on Image and Signal Processing, pp. 103–112. Springer (2010)
11. Ghrist, R., Hiraoka, Y.: Network codings and sheaf cohomology. IEICE Proceedings Series **45**(A4L-C3) (2011)
12. Guo, Z., Sun, G., Ranson, K.J., Ni, W., Qin, W.: The potential of combined lidar and sar data in retrieving forest parameters using model analysis. In: IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium, vol. 5, pp. V–542. IEEE (2008)

13. Hall, D.L., McMullen, S.A.: Mathematical techniques in multisensor data fusion. Artech House (2004)
14. Joslyn, C.A., Charles, L., DePerno, C., Gould, N., Nowak, K., Praggastis, B., Purvine, E., Robinson, M., Strules, J., Whitney, P.: A sheaf theoretical approach to uncertainty quantification of heterogeneous geolocation information. Sensors **20**(12), 3418 (2020)
15. Joslyn, C.A., Hogan, E., Robinson, M.: Towards a topological framework for integrating semantic information sources. In: STIDS, pp. 93–96. Citeseer (2014)
16. Koetz, B., Sun, G., Morsdorf, F., Ranson, K., Kneubühler, M., Itten, K., Allgöwer, B.: Fusion of imaging spectrometer and lidar data over combined radiative transfer models for forest canopy characterization. Remote Sensing of Environment **106**(4), 449–459 (2007)
17. Le, C.D., Pham, H.V., Pham, D.A., Le, A.D., Vo, H.B.: A pm2. 5 concentration prediction framework with vehicle tracking system: From cause to effect. In: 2022 RIVF International Conference on Computing and Communication Technologies (RIVF), pp. 714–719. IEEE (2022)
18. Leal-Taixé, L., Milan, A., Reid, I., Roth, S., Schindler, K.: Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint arXiv:1504.01942 (2015)
19. Malcolm, G.: Sheaves, objects, and distributed systems. Electronic Notes in Theoretical Computer Science **225**, 3–19 (2009)
20. Mitigation, C., Board, A.Q.I.C.P..C.A.R.: California air resources board (2022). URL https://ww2.arb.ca.gov/resources/documents/congestion-mitigation-and-air-quality-improvement-cmaq-program
21. Newman, A.J., Mitzel, G.E.: Upstream data fusion: History, technical overview, and applications to critical challenges. Johns Hopkins APL Technical Digest **31**(3), 215–233 (2013)
22. Nguemo, M.A., Tcheka, C.: Sheaf cohomology on network codings: maxflow-mincut theorem. Applied General Topology **18**(2), 219–230 (2017)
23. Phung, N.K., Long, N.Q., Van Tin, N., Le, D.T.T., et al.: Development of a pm2. 5 forecasting system integrating low-cost sensors for ho chi minh city, vietnam. Aerosol and Air Quality Research **20**(6), 1454–1468 (2020)
24. Purvine, E., Aksoy, S., Joslyn, C., Nowak, K., Praggastis, B., Robinson, M.: A topological approach to representational data models. In: International Conference on Human Interface and The Management of Information, pp. 90–109. Springer (2018)
25. Robinson, M.: Topological signal processing, vol. 81. Springer (2014)
26. Robinson, M.: Imaging geometric graphs using internal measurements. Journal of Differential Equations **260**(1), 872–896 (2016)
27. Robinson, M.: Sheaves are the canonical data structure for sensor integration. Information Fusion **36**, 208–224 (2017)
28. Robinson, M.: Assignments to sheaves of pseudometric spaces. arXiv preprint arXiv:1805.08927 (2018)
29. Robinson, M., Henrich, J., Capraro, C., Zulch, P.: Dynamic sensor fusion using local topology. In: 2018 IEEE Aerospace Conference, pp. 1–7. IEEE (2018)
30. Sahanavin, N., Prueksasit, T., Tantrakarnapa, K.: Relationship between pm10 and pm2. 5 levels in high-traffic area determined using path analysis and linear regression. Journal of Environmental Sciences **69**, 105–114 (2018)
31. Smith, D., Singh, S.: Approaches to multisensor data fusion in target tracking: A survey. IEEE Transactions on Knowledge and Data Engineering **18**(12), 1696–1710 (2006)
32. Solera, F., Calderara, S., Cucchiara, R.: Towards the evaluation of reproducible robustness in tracking-by-detection. In: 2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6. IEEE (2015)
33. Varshney, P.K.: Multisensor data fusion. Electronics & Communication Engineering Journal **9**(6), 245–253 (1997)

34. Wald, L.: Some terms of reference in data fusion. IEEE Transactions on Geoscience and Remote Sensing **37**(3), 1190–1193 (1999)
35. Zhang, J.: Multi-source remote sensing data fusion: status and trends. International Journal of Image and Data Fusion **1**(1), 5–24 (2010)
36. Zhu, S., Guo, Y., Chen, J., Li, D., Cheng, L.: Integrating optimal heterogeneous sensor deployment and operation strategies for dynamic origin-destination demand estimation. Sensors **17**(8), 1767 (2017)