Finding Representative Sampling Subsets in Sensor Graphs Using Time-series Similarities

ROSHNI CHAKRABORTY, JOSEFINE HOLM, TORBEN BACH PEDERSEN, and PETAR POPOVSKI, Aalborg University, Denmark

With the increasing use of Internet-of-Things–enabled sensors, it is important to have effective methods to query the sensors. For example, in a dense network of battery-driven temperature sensors, it is often possible to query (sample) only a subset of the sensors at any given time, since the values of the non-sampled sensors can be estimated from the sampled values. If we can divide the set of sensors into disjoint so-called representative sampling subsets, in which each represents all the other sensors sufficiently well, then we can alternate between the sampling subsets and, thus, increase the battery life significantly of the sensor network. In this article, we formulate the problem of finding representative sampling subsets as a graph problem on a so-called sensor graph with the sensors as nodes. Our proposed solution, SubGraphSample, consists of two phases. In Phase-I, we create edges in the similarity graph based on the similarities between the time-series of sensor values, analyzing six different techniques based on proven time-series similarity metrics. In Phase-II, we propose six different sampling techniques to find the maximum number of representative sampling subsets. Finally, we propose AutoSubGraphSample, which auto-selects the best technique for Phase-I and Phase-II for a given dataset. Our extensive experimental evaluation shows that AutoSubGraphSample can yield significant battery-life improvements within realistic error bounds.

CCS Concepts: • Information systems → Sensor networks;

Additional Key Words and Phrases: Sampling sets, similarity graph, reconstruction error, stratification approach, time-series similarity, internet of things

ACM Reference format:
Roshni Chakraborty, Josefine Holm, Torben Bach Pedersen, and Petar Popovski. 2023. Finding Representative Sampling Subsets in Sensor Graphs Using Time-series Similarities. ACM Trans. Sensor Netw. 19, 4, Article 89 (June 2023), 32 pages.
https://doi.org/10.1145/3595181

1 INTRODUCTION

Recently, Internet-of-Things (IoT)–enabled sensors are being widely used for different applications, such as military operations, traffic management, home-service, healthcare, and several others [4, 15]. Irrespective of the application, the sensors generate continuous data, in the form of time-series. This massive production of data has led to new challenges in data processing, storage...
Fig. 1. An overview of a sensor network, SN, that comprises nine sensors, $S_1$ to $S_9$, where each of the sensor records data in time-series, $T_i$. The sensors communicate to the base-station $B$ for processing and computation. The data recorded by sensors, $S$, is shown in Table 1(a).

and analysis. To handle this massive information overload, there is a need to develop effective methods that can query the sensors efficiently, for example, to preserve battery life. Identifying disjoint representative sampling subsets such that only a representative sampling subset is queried at a given time is an efficient solution for querying the sensors. This is possible if the time-series generated by the different sensors are similar and each representative sampling subset represents the values of all the sensors sufficiently well. Therefore, in this article, we aim to identify the maximum number of disjoint representative sampling subsets given the sensors and their time-series data.

We now discuss this through a motivating example of a sensor network, SN, which comprises nine sensors, $S = S_1, S_2, \ldots, S_9$, as shown in Figure 1. Each of these sensors, $S_i$ records the temperature of a particular location as a time-series, $T_i$ with 5 time instances $1, 2, \ldots, 5$ resulting in $T_{i1}, T_{i2}, \ldots, T_{i5}$ for a sensor, $S_i$. We provide the data recorded by $S$ similar to existing datasets in Table 1(a).

To understand the similarity between the time-series data among sensors, we compute the Fast DTW distance between each pair of sensors (see Table 1(b)). Fast DTW [64] distance is inversely proportional to the similarity between a pair of sensors on the basis of the time-series. Although there are several ways to calculate the similarity or distance between a pair of sensors, we use Fast DTW on SN as an example to calculate distance and provide an intuition of how the similarity between sensors can be utilized to identify disjoint representative sampling subsets. We discuss Fast DTW in detail in Section 4.2.1 and its application on SN in Section 4.4. We highlight the results of Fast DTW in Table 1(b) to give an intuition of the research problem. For example, as shown in Table 1(b), $S_1$ and $S_2$ are similar with a low distances of 8 whereas $S_1$ and $S_9$, $S_2$ and $S_4$ are dissimilar with a high distances of 23 and 28, respectively. Therefore, we can query $S_1$ and $S_2$ at alternating timestamps, improving battery life of both $S_1$ and $S_2$ while still getting sufficiently accurate results. Extending this idea to the entire SN, we identify three disjoint representative sampling subsets, which each represent all the sensors within a given error bound on the time-series values. The intuition is that by identifying three disjoint representative sampling subsets from SN, we can increase the battery longevity by three times compared to the case of querying all sensors at all times. We represent this in Figure 2. It is therefore required to devise a system that can perform (a) creation of a similarity graph of the sensors on the basis of the time-series data and, (b) identification of the maximum number of representative sampling subsets from the similarity graph.

Several selection sampling techniques in graph signal processing domain have been proposed, including randomized [59, 69] or deterministic greedy sampling [9, 27] techniques, which focus on finding a single representative sampling subset. However, these approaches identify only one subset.
of sensors and, therefore, do not solve our objective. Furthermore, existing sampling approaches rely on the availability of the graph topology of the sensors, which might not be always available. Therefore, in this article, we propose a Two-phase framework, namely, SubGraphSample, that identifies the maximum number of representative sampling subsets on the basis of similarity among the sensors such that sensors that generate similar data belong to different representative sampling subsets. In SubGraphSample, we initially create a similarity graph of the sensors in Phase-I and then, iteratively identify representatives from each possible subgraph of the similarity graph to form the maximal number of possible representative sampling subsets in Phase-II. We highlight our contributions next:

1. We propose SubGraphSample that does not require the graph topology of the sensors and can directly identify the maximum number of possible sampling sets given only the readings of the sensors. To the best of our knowledge, there is no existing sampling algorithm that can directly identify the maximum number of sampling sets given only the readings of the sensors. We highlight our major contributions for SubGraphSample next.

(a) To generate the similarity graph of the sensors, we compare and tune six different types of similarity approaches. Although these are previously proposed approaches, there is no existing comprehensive research work that compares the performance of these different types of approaches and provide intuition about which approach to use given a dataset. Additionally, most of these approaches are mainly used only to identify the similarity between sensors for different applications. We propose an effective utilization of these approaches for similarity graph creation for sampling sets identification, which mitigates the dependency of existing sampling approaches on the availability of graph topology. We also provide recommendations of which similarity graph creation approach to use given a dataset.
(b) We propose six novel different types of sampling approaches in Phase-II, namely, Strat, MSV, JIP, SIP, Frob, and Par. In Strat, we propose three different approaches that utilize network science theoretic concepts to identify the maximum number of possible sampling sets. To the best of our knowledge, this is the first research work that proposes network science theoretic concepts for sampling sets identification and is designed specifically for time-series-based applications. In JIP and SIP, we propose utilization of the difference between the similarity graph signal and the signal of a sensor, namely, mean-square error, MSE, to select the sensor into a sampling set and thereby, iteratively create the maximum number of possible sampling sets. Additionally, in MSV, Frob, and Par, we extend methods that focus on identifying one optimum sampling set, to methods generating the maximum number of possible sampling sets. While we explore a measure using the eigen decomposition of the adjacency matrix in MSV, we explore different measures using the eigen decomposition of the graph laplacian matrix in Frob and Par, respectively.

(c) For our experiments, we compare the performance of Phase-I approaches in creating similarity graph given a dataset by the average path length, clustering coefficient, edge density, and measure how well the graph topology represents the dataset by total cumulative energy residual. Additionally, we use reconstruction error to compare the performance of Phase-II sampling approaches and SubGraphSample, respectively. Our experimental evaluations on 4 datasets show that the best combination of the graph creation approach and sampling approach can provide 5–13 times increase in battery life with a 20–40% error bound for a given dataset. We discuss our experimental evaluations in detail in Section 7.1 and Section 7.2, respectively.

(2) We propose an auto-tuned approach, AutoSubGraphSample, to select the best combination of Phase-I and Phase-II approaches for a given dataset.

(3) We evaluate AutoSubGraphSample on nine representative datasets. Our observations indicates that AutoSubGraphSample can generalize well to new datasets.

(4) We perform several additional experimental analyses, such as evaluate the performance of AutoSubGraphSample when only a fraction of time-series is provided (details in Section 7.8) and a comparative analysis of the AutoSubGraphSample with the optimum results at each phase (details in Sections 7.4 and 7.5).

The organization of the article is as follows. We discuss the existing research works in Section 2 followed by the problem statement in Section 3. In Sections 4 and 5, we discuss the proposed approach and the experiments setup. We show our observations in Section 6. We, finally, draw our conclusions and future works in Section 8.

2 RELATED WORK

2.1 Phase-I: Creation of Similarity Graphs

We categorize the existing research works that could be used to create a similarity graph of sensors, given the data generated by the sensors, into three types of approaches: Statistical, Time-series Analysis, and Graph Signal Processing.

2.1.1 Statistical Approaches. To identify graphs between sensors, a simple way is to calculate similarity between each pair of sensors and then, create an edge between them if their similarity is greater than the threshold [48]. Therefore, existing metrics, such as the Pearson correlation, the Jaccard coefficient, the Gaussian radial basis function and mutual information are used to compute the pairwise similarity and thereby, identify the graph topology [24, 31]. Feizi et al. [26] extended the pairwise correlation by including the indirect dependencies from the transitive correlations through a network deconvolution-based approach.
2.1.2 Time-series Analysis-based Approaches. Using time-series data, one can compute the similarity between a pair of sensors and use it as a basis to define an edge in the graph. The existing approaches based on time-series can be classified into distance/neighbourhood-based methods and feature-based methods [34]. Distance-based methods focus on identifying different distance metrics to align a pair of time-series [75]. Traditional distance metrics that are inspired by the concept of edit distance [11], include Lp-norms [82], Euclidean Distance [25], Dynamic Time Warping (DTW) [6], Longest Common Sub-sequence (LCSS) [73], Edit Sequence on Real Sequence (EDR) [12], Swale [50], Spatial Assembling Distance (SpADe) [14], and so on. Further, several existing research papers have proposed different variants [19] of these traditional distance metrics for different objectives, such as run-time [20], applicability to specific problem [83], and so on. There are several feature-based classification algorithms [67], such as traditional SVM and decision trees, naive logistic model (NL), the Fisher kernel learning (FKL) [33], Hidden State Conditional Random Field [56], and histogram-based method [85, 86], to determine time-series similarity. However, several studies [37, 80] have shown that feature-based approaches fail to capture the inherent intrinsic attributes particular to time-series data, which are better captured by distance-based approaches. Further, several recent research works have proposed integration of both neighbourhood-based metrics [29, 34] and distance-based metrics to train machine learning models, such as SVM, Random Forest, and ensemble models [44]. Recently, several research papers have proposed different neural network architectures, such as autoencoders [1], deep networks [49], meta-learning-based pre-training [52], attention modules [49, 81] to capture the complex temporal relationships in time-series.

2.1.3 Graph Signal Processing-based Approaches. To ensure analysis and processing of the graph signals in both the vertex and the spectral domain of the graph, several recent papers infer an optimum graph topology such that the input data form graph signals with smooth variations on the resulting topology [42, 72]. Dong et al. [23] propose a factor analysis-based model, which was extended by Kalofolias et al. [35] to include sparsity. However, these approaches assume that the graph signals used for training are smooth.

2.1.4 Summary of Insights. Considering the variety of existing approaches to infer the similarity graph based on the sensing data, there is still a lack of a comprehensive study that compares how different approaches perform on a given dataset. These approaches have mainly been used for different applications, such as to infer similarity or distance between two time-series, create an optimum graph topology given the sensor signals, and so on. In addition, to the best of our knowledge, there is no existing research that utilizes this similarity to generate a similarity graph for sampling approaches. Therefore, there is no existing research tailored to identify the maximum number of disjoint representative sampling subsets given the time-series data of the sensors. In this article, we select several prominent existing approaches from the three categories described above and compare them in their role in Phase-I for a given dataset.

2.2 Phase-II: Sampling Algorithms

Randomized sampling-based approaches [59, 60, 69] select nodes from a predetermined probability distribution. They have a low computational cost, but cannot ensure the same quality at each selection. Deterministic greedy sampling techniques resolve this by selecting the optimal sensor at each iteration. This deterministic operation scales with polynomial complexity [9, 27]. However, most of these sampling techniques search for only one optimal sampling set and do not consider the time dimension of the data [5, 17, 38, 63]. Therefore, these techniques do not resolve our objective of maximizing battery longevity. The works [54, 77] identify each sampling set representing a time-graph signal; nevertheless, the same node may participate in different representative sampling...
subsets, which is not suitable to maximize the battery longevity. The sampling technique from Reference [16] can ensure improvement in battery lifetime. However, Reference [32] has shown that the approach from Reference [16] is sub-optimal. Although Gedik et al. [28] and Liu et al. [45] intend to maximize the battery longevity of the sensor network, these techniques are not applicable for the application we are interested in. For example, Gedik et al. [28] assume that the set of clusters is given and therefore, requires both the similarity among sensors and the clustering information to be provided beforehand. Although Liu et al. [45] do not require the clustering information to be provided, it requires the location of the sensors to be provided and clusters the sensor network based on the spatial correlation among sensors. In comparison, we only require the time-series data of the sensors. Thus, we propose six different sampling techniques to identify the maximum number of representative sampling subsets of which we propose two novel ones and extend four existing ones [9, 13, 71].

3 PROBLEM STATEMENT AND FRAMEWORK

3.1 Problem Statement

Given a sensor graph comprising n IoT-enabled sensors, \( S = (S_1, S_2, \ldots, S_n) \), the time-series data for the sensor \( S_i \) is denoted by \( T^i \). Let \( CP \) denote the set of all (say, \( q \) in this case) possible complete partitions of the network \( CP = (SP_1, SP_2, \ldots, SP_q) \). A partition, \( SP_u \) consists of several non-empty subsets of \( S \), i.e., \( SP_u = (SP_u^1, SP_u^2, \ldots, SP_u^k) \) such that \( \bigcup_{i=1}^{k} SP_u^i = S \). The sensors are battery-powered and have low computing power. We also assume that the time-series data have no missing values. We identify the optimum partition, \( OSP \), from all the possible complete partitions, \( CP \), such that

\[
OSP = \arg\max_{x \in CP} (|SP_x|)
\]

\[
\text{s.t. } \text{Error}(SP_x) \leq \epsilon \ \forall \ SP_x \in CP
\]

\[
SP_x^i \cap SP_x^j = \emptyset \ \forall (SP_x^i \neq SP_x^j) \in CP
\]

The optimum sampling partition, \( OSP \), is the partition that comprises the maximum number of representative sampling subsets, \( SP_u \), such that each of these non-empty subsets, \( SP_u^i \) can represent the values of all sensors well enough, i.e., the error in the information recorded by \( SP_u^i \) when compared to \( S \) must be less than the threshold, \( \epsilon \), as \( \text{Error}(SP_u^i, S) \leq \epsilon \). We consider reconstruction error to calculate \( \text{Error}(SP_u^i, S) \), which we discuss in details in Section 6.2.1. Additionally, we assume only periodic round-robin scheduling of each representative sampling subset, \( SP_u^i \) of \( OSP \) in this article. Furthermore, we consider constraint that no two subsets of \( OSP \) can overlap, i.e., \( SP_u^i \cap SP_j^i = \emptyset \).

This problem can be reformulated to identify the optimal partition \( OSP \) that minimizes the error for a given number \( K \) representative sampling subsets:

\[
OSP = \arg\min_{x \in CP} (\text{Error}(SP_x, S))
\]

\[
\text{s.t. } |SP_x| = K
\]

\[
SP_x^i \cap SP_x^j = \emptyset \ \forall (SP_x^i \neq SP_x^j) \in CP
\]

We show a brute-force solution to Equation (1) in Algorithm 1, where we initially identify all the possible complete partitions of \( G \). We then enumerate all of these complete partitions to identify the partition that has the maximum number of sampling partitions such that each sampling partition is within the error threshold. Identifying one complete partition of a graph is NP-hard [30] (Algorithm 1, line 1). The number of possible complete partitions in a \( G \) of \( n \) sensors are given by the Bell numbers \( B_n \) \((B_0 = 1, \ldots) \) [18] (where, \( B_0 = 1, B_1 = 1, B_2 = 2, B_3 = 5, B_4 = 15, B_5 = 52, \text{etc.}) \) [78]. Therefore, the time-complexity of the inner loop of Algorithm 1 (lines 3–18) is \( O(B_n s^2) \), where \( s \) represents the maximum possible size of a sampling set, \( SP_x^i \) and \( B_n \) is number of possible complete
ALGORITHM 1: Brute-force Solution

**Input** Time-series Similarity Graph \( G = (V,E) \) and error threshold, \( \epsilon \)

**Output** Optimum Sampling Partition \( OSP \)

1. Find the set of all possible complete partitions, \( CP = (SP_1, SP_2, \ldots, SP_q) \) of \( G \)
2. Initialize max=0 \( \triangleright \) variable to store the maximum number of sampling sets among all partitions
3. for \( x \) in \( CP \) do \( \triangleright \) iterate through all the possible complete partitions
4. Initialize count=0 \( \triangleright \) variable to store the number of sampling sets of \( SP_x \) within \( \epsilon \)
5. for \( i \) in \( SP_x \) do \( \triangleright \) \( SP_x \) is a complete partition of \( CP \) and \( SP^i_x \) is \( i^{th} \) sampling set of \( SP_x \)
6. if (Error\( (SP^i_x, S) \leq \epsilon \)) then
7. count=count+1
8. Initialize inter=0 \( \triangleright \) inter is 1 if \( SP^i_x \) is non disjoint
9. for \( j \) in \( SP_x \) do
10. if \( (SP^i_x \cap SP^j_x) \neq \emptyset \) then
11. inter=1
12. if (count==|\( SP_x \)| and (inter==0) and (|\( SP_x \)| \( \geq \) max)) then
13. max=|\( SP_x \)|
14. \( OSP = SP_x \) \( \triangleright \)
15. Return \( OSP \)

The partitions of \( CP \) [78]. Therefore, Algorithm 1 given \( CP \) (from line 3–16) is exponential with respect to the number of sensors, \( n \) [2]. Therefore, we instead propose a heuristic solution based on a 2-phase framework, SubGraphSample, which can solve either of the dual problems, Equation (1) or Equation (2). In Phase-I, we create a similarity graph, \( G = (V,E) \) such that the vertices \( V \) are the sensors, \( S \) and the edges \( E \) represent the similarity of the recorded data between each pair of sensors, \( S_i \) and \( S_j \) using existing approaches. In Phase-II, we identify the \( OSP \) from \( G \). An overview of the proposed approach on \( SN \) is shown in Figure 2. We discuss graph creation approaches for Phase-I in Section 4 and propose sampling approaches for Phase-II in Section 5. Furthermore, we propose Algorithm AutoSubGraphSample to recommend the most suitable algorithm for both Phase-I and Phase-II, respectively, for a given dataset.

3.2 Preliminaries

We now discuss the graph signal processing preliminaries needed to understand the proposed sampling techniques and evaluation metrics. We consider a dataset that comprises \( n \) sensors, \( t \) as the length of the time-series of each sensor such that \( T = 0,1, \ldots, (t-1) \), \( S = S_0, S_1, \ldots, S_{n-1} \) is the set of sensors and \( s \) as signal for the rest of the article. We provide a summary of the notations in Table 2.

- **Degree Matrix**, \( D \): A diagonal matrix that contains the degree of each node, i.e., with entries \( D_{ii} = \sum_{j=1}^{n} A_{ij} \) and \( D_{ij} = 0 \) for \( i \neq j \), where \( A \) is the adjacency matrix.
- **Graph Laplacian**, \( L \): \( L \) is calculated as \( L = A - D \), where \( A \) is the adjacency matrix and \( D \) is the degree matrix [66].
- **Signal**: A signal represents a time-dependent function that conveys information [53]. For example, the signal is \( s_T = s_0, s_1, \ldots, s_{t-1} \) and \( s_j \) is a sample of the signal \( s_T \).
- **Graph Signal**, \( x \): A signal whose samples are indexed by the nodes of a graph [53]. In this article, we consider graph-time signal, i.e., one graph signal, \( x^r \) per time stamp, \( r \) where \( r \in 0,1, \ldots, (t-1) \). Therefore, a graph signal represents the values of each sensor at a time-stamp,
Table 2. Summary of the Notations

| Notation          | Description                                      |
|-------------------|--------------------------------------------------|
| \( S = (S_1, S_2, ..., S_n) \) | list of sensors                                 |
| \( n \)           | number of sensors in \( S \)                    |
| \( m \)           | length of time-series of a sensor, \( S_i \)    |
| \( K \)           | number of representative sampling subsets        |
| \( SPP_u \)       | \( u \)th sampling partition                    |
| \( OSP \)         | optimum sampling partition                      |
| \( D \)           | degree matrix                                   |
| \( L \)           | graph Laplacian                                 |
| \( \Lambda \)     | eigenvalues of \( L \)                          |
| \( V \)           | eigenvectors of \( L \)                         |
| \( x^r \)         | graph signal, \( x \) at \( r \)th time-stamp where \( r \in [0,1,\ldots,(t-1)] \) |
| \( \mathcal{A} \) | adjacency matrix                                |
| \( \Sigma \)      | eigenvalues of \( \mathcal{A} \)                |
| \( U \)           | eigenvectors of \( \mathcal{A} \)               |
| \( \Sigma \)      | eigenvalues of \( \mathcal{A} \)                |
| \( U \)           | eigenvectors of \( \mathcal{A} \)               |
| \( \hat{x}^k \)   | Graph Fourier Transform of \( x^k \)            |
| \( SPP^i_u \)     | \( i \)th sampling set of \( SPP_u \)          |
| \( \epsilon \)    | error threshold                                 |
| \( \mathcal{G} \) | similarity graph                                |
| \( P_y \)         | similarity graph creation approach, \( y \)     |
| \( \mathcal{G}_y \) | similarity graph created by \( y \)           |
| \( \rho(S_i, S_j) \) | Pearson Correlation Coefficient, of \( S_i \) and \( S_j \) |
| NodeScore(\( S_i \)) | importance of \( S_i \)                         |
| \( R_i \)         | relevance of \( S_i \)                          |
| \( C_k \)         | \( k \)th community of \( \mathcal{G} \)       |
| \( IG(S_i, SPP^i_u) \) | information gain provided by \( S_i \) with respect to \( SPP^i_u \) |
| \( E_d \)         | edge density of \( \mathcal{G} \)               |
| \( SamSErr(SPP^i_u, S) \) | sampling error of \( SPP^i_u \) w.r.t. to \( S \) |
| \( Err(OSP, S) \) | reconstruction error of \( SPP_u \) calculated for \( K \) sampling sets |
| Avg \( P_l \)     | average path length of \( \mathcal{G} \)        |
| Avg CC            | clustering coefficient of \( \mathcal{G} \)     |
| TCER              | total cumulative energy residual                |

- \( x^r \) comprises \( n \) samples (for \( n \) sensors) where each sample \( x^r_j \) is the value for sensor, \( S_j \) at the time stamp \( r \).
- **Smoothness:** A graph signal \( x^r \) at the \( r \)th time-stamp is smooth if it has similar values for the neighbouring nodes of \( \mathcal{G} \).
- **Graph Fourier transform, GFT:** GFT is the eigendecomposition of the graph Laplacian, \( L \) or adjacency matrix, \( \mathcal{A} \) into eigenvalues, \( \Lambda \) and eigenvectors, \( V \). The eigendecomposition of \( L \) is \( L = \Lambda V^{-1} \), GFT of \( x^k \), i.e., \( \hat{x}^k \), which is defined as \( \hat{x}^k = V^{-1} x^k \) [66].
- **Bandlimited Signal:** This is a signal that is limited to have non-zero spectral density only for frequencies that are below a given frequency. If \( \hat{x}^k \) is bandlimited, i.e., there exist a \( B \in [0,1,\ldots,F-1] \) such that \( \hat{x}_i = 0 \) for all \( i \geq B \), then \( x^k \) is compressible and can be sampled [53].
4 PHASE I: SIMILARITY GRAPH CREATION

In this section, we discuss the creation of the similarity graph, \( G \), by selecting different approaches from Statistical Approaches, Time-series-based Approaches and Graph Signal Processing-based Approaches.

4.1 Statistical Approaches

We discuss two statistical approaches next.

4.1.1 Correlation-based Approach, \( P_{\text{corr}} \). From Reference [48], we calculate Pearson Correlation Coefficient, \( \rho(S_i, S_j) \) to determine the similarity between \( S_i \) and \( S_j \) as

\[
\rho(S_i, S_j) = \frac{\sigma(S_i, S_j)}{\sqrt{\text{var}(S_i) \cdot \text{var}(S_j)}}
\]  

(3)

where \( \sigma(S_i, S_j) \) is the co-variance between \( S_i \) and \( S_j \) and \( \text{var}(S_i) \) calculates the variance of the data for \( S_i \). Therefore, we create an edge between \( S_i \) and \( S_j \) in \( G_{\text{corr}} \) if \( \rho(S_i, S_j) \) is greater than the threshold. The time complexity of \( P_{\text{corr}} \) is \( O(n^3) \) [48].

4.1.2 Network Deconvolution \( P_{\text{conv}} \). We use network deconvolution [26, 68] to create \( G_{\text{conv}} \) from the adjacency matrix \( A \). Network deconvolution calculates \( A \) based on the co-variance matrix, \( \Sigma \), determined from the data \( S \) generated by the sensors:

\[
A = \Sigma(I + \Sigma)^{-1}.
\]  

(4)

The time complexity of \( P_{\text{conv}} \) is \( O(n^3) \) [26].

4.2 Approaches based on Time-series

We discuss three approaches that determine similarity based on the time-series of each pair of sensors, \( S_i \) and \( S_j \).

4.2.1 Dynamic Time Warping (DTW), \( P_{\text{dtw}} \). \( P_{\text{dtw}} \) measures the distance between a pair of sensors, \( S_i \) and \( S_j \) by calculating the distance, \( \text{DisDTW}(S_i, S_j) \) based on the Euclidean distance of the respective time-series of the sensors, \( S_i \) and \( S_j \), at the particular time-stamp and the minimum of the cumulative distances of adjacent elements of the two-time-series. However, \( P_{\text{dtw}} \) incurs high computational cost, which runs across different time-series. Therefore, we use Fast DTW [64], which being an approximation of \( P_{\text{dtw}} \) runs in linear time and space [64]. We calculate the distance between \( S_i \) and \( S_j \) as \( \text{DisFDTW}(S_i, S_j) \) and create an edge between \( S_i \) and \( S_j \) in \( G_{\text{dtw}} \) if the \( \text{DisFDTW}(S_i, S_j) \) is less than the threshold. The time complexity of \( P_{\text{dtw}} \) is \( O(n^2m) \) as the calculation of \( \text{DisFDTW}(S_i, S_j) \) is \( O(m) \) [64] (\( m \) being the length of the time-series), and we calculate \( \text{DisFDTW}(S_i, S_j) \) for \( n \) pairs of sensors to create \( G_{\text{dtw}} \).

4.2.2 Edge Estimation Based on Haar Wavelet Transform, \( P_{\text{haar}} \). The data generated from the sensors is inherently unreliable and noisy. Therefore, we compress the time-series of sensor, \( S_i \) to effectively handle the unreliability in the data by Haar wavelet transform [10]. We select the \( K \)-largest coefficient for \( S_i \) and \( S_j \) to get a compressed approximation as \( S'_i \) and \( S'_j \), respectively [79]. We create \( G_{\text{haar}} \), in which an edge between \( S_i \) and \( S_j \) exists if the Euclidean distance between \( S'_i \) and \( S'_j \) is less than the threshold. The time complexity of \( P_{\text{haar}} \) is \( O(n^2\log_2m) \) as the calculation of Haar distance between a pair of sensors is \( O(\log_2(m)) \) [46] (\( m \) being the length of the time-series), and we calculate the distance for \( n \) pairs of sensors to create \( G_{\text{haar}} \).
4.2.3 **K-NN Approach, \( P_{nei} \).** We follow \( K \) nearest neighbours, where a class of a node is assigned on the basis of its \( K \) nearest neighbours [3]. We initially calculate the distance between a pair of sensors, \( S_i \) and \( S_j \) based on Euclidean distance and create an edge between \( S_i \) and \( S_j \) in \( G_{nei} \) if the distance between them is among the least \( K \)-distances. The time complexity of \( P_{nei} \) is \( O(m \log n) \) [18].

4.3 **Approaches based on Graph Signal Processing, \( P_{gsp} \)**

We follow Reference [35] to infer the graph topology from signals under the assumption that the signal observations from adjacent nodes in a graph form smooth graph signals. The solution from Reference [35] is scalable and the pairwise distances of the data in matrix

\[
Z_{ij} = \|x_i - x_j\|^2
\]

are introduced as in

\[
W^* = \min_W \|W \circ Z\|_{1,1} - \alpha 1^T \log(W 1) + \frac{\beta}{2} \|W\|_F^2,
\]

where \( W^* \) is the optimal weighted adjacency matrix, \( \| \cdot \|_{1,1} \) is the elementwise 1-norm, \( 1^T \log(W 1) \) ensures overall connectivity of the graph by forcing the degrees to be positive while allowing sparsity, and \( \alpha \) and \( \beta \) are parameters to control connectivity and sparsity, respectively. The time complexity to create \( P_{gsp} \) is \( O(n^2) \) [35]. We follow the implementation in Reference [57] to determine the weighted adjacency matrix, \( W \). We create an unweighted adjacency matrix, \( A \) and graph, \( G_{gsp} \) by creating an edge in \( A \) and \( G_{gsp} \) if the edge weight in \( W \) is greater than threshold. However, we observe that most of the edge weights are around 0 and very few edge weights are within \( 0.5 - 1 \), therefore, it is difficult to create graphs with every edge density by \( P_{gsp} \).

4.4 **Summary of Insights**

Therefore, in Phase-I, we explore a variety of approaches to generate the similarity graph. These approaches vary immensely on the basis of their methodology. Additionally, to the best of our knowledge, there is no existing research work that studies these different approaches and provide insights on the applicability of these approaches to generate the similarity graph topology for a sampling technique. We experimentally analyze the performance of these approaches in Section 7 to provide heuristics to select the best approach given a dataset. On the basis of the time complexity of these approaches, we observe that \( P_{nei} \) performs the best. To illustrate how the similarity graph is created given the time-series, we show how \( P_{dtw} \) works on SN. On the basis of distance calculated between each pair of sensors as shown in Table 1(b), we create an edge between each pair of sensors, \( S_i \) and \( S_j \) in \( G_{dtw} \) if the distance between \( S_i \) and \( S_j \) is less than the threshold, say 15 for SN. Therefore, we show \( G_{dtw} \) in Phase-I of Figure 2, where \( S_1 \) and \( S_2 \), \( S_3 \) and \( S_0 \) are connected as the distances are 8 and 4, which are less than 15. Additionally, \( S_1 \) and \( S_9 \), \( S_2 \) and \( S_4 \) with distance 23 and 28 are not connected.

5 **PHASE II: IDENTIFYING OPTIMUM SAMPLING PARTITION (OSP)**

We propose several sampling approaches that utilize \( G \) to identify representative sampling subsets are representative of the values of all sensors.

5.1 **Network Stratification-based Approach, Strat**

We propose a network stratification-based sampling approach, \( Strat \), that captures the inter-relationship among sensors at group level to inherently handle the sparsity at individual connections and the generic global attributes at the network level.\(^1\) Therefore, in \( Strat \), we initially group similar sensors together into communities by Modularity Maximization [7] followed by

---

\(^1\)https://en.wikipedia.org/wiki/Level_of_analysis.
ALGORITHM 2: SRel

Input: Time-series Similarity Graph \( \mathcal{G} = (V,E) \), number of representative sampling subsets \( K \)

Output: Optimum Sampling Partition \( OSP \)

1. \( C = \) Modularity Maximization algorithm(\( \mathcal{G} \))
2. \( ComMem = \min_{v \in C} \text{Size}(v)/K \)
3. for \( u \) in \( S \) do
4. \( \text{NodeScore}(S_u) = \) Eigenvector Centrality(\( S_u \))
5. Initialize \( OSP = [\] \)
6. for \( i \) in range(0,\( K \)) do
7. for \( y \) in \( C \) do
8. for \( z \) in \( \text{ComMem} \) do
9. \( \text{MNodeScore} = \max_{u \in S} \text{NodeScore}(S_u) \)
10. Identify \( S_u \) with \( \text{MNodeScore} \)
11. Add \( S_u \) to \( SP_i \)
12. Add \( SP_i \) to \( OSP \)
13. Return \( OSP \)

selecting representatives from each of these communities to create a representative sampling subset. We use Modularity Maximization-based community detection to group similar sensors as it is similar to the problem of community detection in large networks, as in online social networks [40, 41]. Among the multiple available community detection algorithms, we have opted for Modularity Maximization as it is efficient and scalable to large networks [7]. To create a representative sampling subset, we select a sensor from each community based on their importance to that community. We denote the importance of a sensor, \( S_i \) by \( \text{NodeScore}(S_i) \). We propose three different mechanisms to calculate \( \text{NodeScore}(S_i) \), which we discuss next in detail. Therefore, we iteratively select sensors from each community in the decreasing order of \( \text{NodeScore}(S_i) \) to form a representative sampling subset. We tune the selection method depending on whether we solve Equation (1) or Equation (2).

5.1.1 Selection by Relevance, SRel. In SRel, we calculate \( \text{NodeScore}(S_i) \) as the relevance of \( S_i \), i.e., \( R_i \), with respect to \( C_k \) to capture the centrality of \( S_i \). Centrality score of a node measures the importance of that node in the network [21]. In this scenario, \( R_i \) measures the importance of \( S_i \) with respect to all the other sensors present in \( C_k \). Since, an edge between any two sensors in \( C_k \) represents the similarity between the time-series of those two sensors, \( R_i \) captures the similarity of \( S_i \) with all the other sensors of \( C_k \). Therefore, by selecting the sensor with the highest centrality, we can ensure maximum representation of all the sensors present in \( C_k \). We utilize the concept of centrality to generate the maximum number of possible representative sampling subsets. We measure \( R_i \) by Eigenvector Centrality [62]. We determine the number of sensors to be selected from \( C_k \) by \( \text{ComMem} \). The calculation of \( \text{ComMem} \) varies based on whether we solve Equations (1) or (2). For Equation (1), we select the minimum number of possible nodes from each community such that the selected nodes can represent all the nodes from the community sufficiently well. We set \( \text{ComMem} \) to be 1 and then, iteratively select \( \text{ComMem} \) sensors from \( C_k \) in decreasing order of \( \text{NodeScore}(S_i) \) to create a representative sampling subset such that the error of the representative sampling subset with respect to all the sensors is less than \( \epsilon \). We repeat these steps to create the maximum number of possible representative sampling subsets, i.e., \( OSP \) and thus, optimize Equation (1).

To solve Equation (2), we set \( \text{ComMem} \) as the ratio of the size of the smallest community and the given number of representative sampling subsets, \( K \). We then iteratively select \( \text{ComMem} \) sensors from
**Algorithm 3: SMMR**

**Input** Time-series Similarity Graph \( G = (V,E) \), number of representative sampling subsets \( K \)

**Output** Optimum Sampling Partition \( OSP \)

1. \( \text{ComMem} = \min_{v \in C} \text{Size}(v) / K \)
2. for \( u \) in \( S \) do
3. \( \text{NodeScore}(S_u) = \text{Eigenvector Centrality}(S_u) \)
4. Initialize sampling subsets, \( OSP = [] \)
5. for \( i \) in range(0, \( K \)) do
6. for \( y \) in \( C \) do
7. for \( z \) in \( \text{ComMem} \) do
8. \( A[S_u] = \text{Adjacency List of Node } S_u \)
9. \( D(a,b) = \text{Difference between a set } a \text{ and a set } b \)
10. Calculate \( I_G(S_u, S^P_x) = |D(A[S_i], A[S^P_x])| \)
11. Calculate \( M\text{NodeScore} \) by Equation (7)
12. Identify sensor, \( S_u \) with \( M\text{NodeScore} \)
13. Add \( S_u \) to \( S^P_x \)
14. Add \( S^P_x \) to \( OSP \)
15. Return \( OSP \)

A \( C_k \) in decreasing order of \( \text{NodeScore}(S_i) \) to form a \( S^P_x \) and repeat this step for \( K \) times to create \( OSP \). We show the algorithm of \( SRel \) in Algorithm 2. The time complexity of \( SRel \) depends on the time complexity of the community detection algorithm \( (O(E)) \) [7, 70], calculation of eigen-vector centrality \( (O(n^2)) \) [62] and calculation and finding the sensor with the highest \( \text{NodeScore}(S_i) \) \( (O(|C| \log_2(|C_k|)). \) Therefore, the calculation of eigen-vector centrality is most computationally expensive for \( SRel \), which makes the time complexity of \( SRel \) be \( O(n^2) \). We follow the same procedure as \( SRel \) in SMMR and SEMMR to calculate \( \text{ComMem} \) and determine \( OSP \) for either Equation (1) or Equation (2). However, we calculate \( \text{NodeScore}(S_i) \) differently in SMMR and SEMMR, which we discuss next.

5.1.2 Selection by Maximum Marginal Relevance, SMMR. In SMMR, we consider both relevance and information gain of a sensor to calculate \( \text{NodeScore}(S_i) \). We propose Maximum Marginal Relevance [8]-based score to calculate \( \text{NodeScore}(S_i) \), which is the weighted average of the relevance, \( R_i \) and the information gain provided by \( S_i \) with respect to \( S^P_x \), \( IG(S_i, S^P_x) \). We measure \( R_i \) as in \( SRel \) and \( IG(S_i, S^P_x) \) as the difference between the adjacency list of \( S_i \) and the adjacency list of the already selected sensors in \( S^P_x \). Therefore, we select the sensor with maximum node score, \( M\text{NodeScore} \) and further, repeat this for \( \text{ComMem} \) times for each \( C_k \) iteratively. The calculation of \( M\text{NodeScore} \) is as follows:

\[
M\text{NodeScore} = \max_{\{S_i \in C_k\}} \left[ \text{NodeScore}(S_i) \right]
\]

\[
= \max_{S_i \in C_k} [\beta R_i - (1 - \beta) IG(S_i, S^P_x)], \tag{7}
\]

where \( \beta \) is the weight for relevance and \( (1 - \beta) \) for information gain, respectively. For our experiments, we consider \( \beta \) as 0.5 to give equal importance to both relevance and information gain, respectively. We show the pseudocode of SMMR in Algorithm 3. The time complexity of SMMR is similar to SRel except for the calculation of Maximum Marginal Relevance [8], which has a time complexity of \( O(|C_k|^2) \). However, as the number of sensors in a community, i.e., \( |C_k| \) is much less than the number of sensors in the network, i.e., \( n \), the time complexity for eigen-vector
ALGORITHM 4: SEMMR

Input Time-series Similarity Graph $G = (V, E)$, number of representative sampling subsets $K$
Output Optimum Sampling Partition $OSP$

1: $\text{ComMem} = \min_{v \in C} \text{Size}(v)/K$
2: for $u$ in $S$ do
3: \text{NodeScore}(S_u) = \text{Eigenvector Centrality}(S_u)$
4: Initialize sampling subsets, $OSP = []$
5: for $i$ in range(0, $K$) do
6: for $y$ in $C$ do
7: for $z$ in $\text{ComMem}$ do
8: $T[S_u] =$ Time-series Data of $S_u$
9: $\text{Er}(S_u, SP_i^z) =$ Average Minimum-square Error of $S_u$ w.r.t. $SP_i^z$
10: Calculate $I_G(S_u, SP_i^z) = \text{Er}(T[S_u], T(SP_i^z))$
11: Calculate $M\text{NodeScore}$ by Equation (7)
12: Identify sensor, $S_u$ with $M\text{NodeScore}$
13: Add $S_u$ to $SP_i^z$
14: Add $SP_i^z$ to $OSP$
15: Return $OSP$

centrality is the most computationally expensive for SMMR. Therefore, the time complexity for SMMR is same as for SRel, i.e., $O(n^2)$.

5.1.3 Selection by Error-based Maximum Marginal Relevance, SEMMR. In SRel and SMMR, we consider the edges between the sensors in $G$ to calculate $\text{NodeScore}(S_i)$ and do not consider the actual data generated by $S_i$. In SEMMR, we incorporate this information by calculating $I_G(S_i, SP_i^z)$ as the average of the minimum-square error between the data generated by $S_i$ and the other sensors already selected in $SP_i^z$. We follow the same procedure of SMMR to calculate $M\text{NodeScore}$ and finally, follow the same procedure as discussed in SRel to determine ComMem and resolve either Equation (1) or Equation (2) accordingly. Algorithm 4 shows the pseudocode of SEMMR. The time complexity of SEMMR is same as for SMMR, i.e., $O(n^2)$ as it follows same approach as SMMR except for utilizing the time-series of sensors instead of the adjacency list. SEMMR is redundant when compared to all the proposed sampling techniques as it also requires time-series of the sensors and is not just dependent on the $G$.

5.2 Minimum Singular Value-based Approach, MSV

Chen et al. [13] proposed a greedy-selection-based sampling algorithm in which they iteratively select the node that maximizes the minimum singular value of the eigen vector matrix under the assumption that the signal is bandlimited. Under the assumption of bandlimitedness, selection of the best $|B|$ nodes ensures almost complete reconstruction of the graph signal given that there is no sampling noise. Therefore, by choosing the node that maximizes the minimum singular value, Chen et al. optimize the information in the graph Fourier domain and form a greedy approximation of the best $|B|$ nodes. The authors consider eigen decomposition of the adjacency matrix, $A = U \Sigma U^{-1}$, for graph Fourier transform, $\hat{x} = U^{-1}x$, and create only one representative sampling subset with $|B|$ nodes by selecting the nodes iteratively according to

$$m = \arg \max_q \sigma_{\min}(UB, SS_p + \{q\}),$$

ACM Transactions on Sensor Networks, Vol. 19, No. 4, Article 89. Publication date: June 2023.
Algorithm 5: Minimum Singular Value, MSV

**Input** Time-series Similarity Graph $G = (V, E)$, number of representative sampling subsets $K$, bandwidth, $B$

**Output** Optimum Sampling Partition, OSP

1. Initialize OSP as a set of $K$ empty sets
2. Initialize $A$ as the adjacency matrix of $G$
3. Calculate eigendecomposition $A = U \Sigma U^{-1}$
4. for $i$ in the number of nodes do
   5. $p = i \mod K$
   6. $m = \text{argmax}_q \sigma_{\text{min}}(U_B, OSP_p + \{q\})$
   7. $OSP_p \leftarrow OSP_p + \{m\}$

Return OSP

where $U_{B, SP_p}$ is the first $|B|$ rows of $U$, $SP_p$ represents the set of columns of $U$ and $\sigma_{\text{min}}(U)$ is the function for the minimal singular value of $U$. In this article, we propose MSV, which is an extension of Reference [13], where we generate $K$ representative sampling subsets by iteratively adding nodes to each representative sampling subset according to Equation (8) until all nodes have been assigned. We provide the pseudocode of MSV in Algorithm 5. The time complexity of MSV is $O(n^3K^2)$ as it depends on the time complexity of singular value decomposition of the matrix, $U_{B, OSP_p + \{q\}}$, which is $O(nK^2)$ [65], and we calculate the singular value decomposition for $n^2$ sensors. Applying MSV for $SN$ generates three representative sampling subsets results in $(5, 4, 7)$, $(2, 6, 3)$, and $(0, 8, 1)$. In this article, we consider $B = V$.

5.3 Greedy MSE-based Approach, JIP and SIP

We propose two sampling techniques, i.e., Joint Iterative Partitioning, JIP, and Simultaneous Iterative Partitioning, SIP, that consider Mean-square Error to select a node into a representative sampling subset. By considering Mean-square Error, we ensure that each representative sampling subset generated can reconstruct the original graph within an error bound. We discuss JIP and SIP in detail next. We estimate $MSE$ as in Reference [9]:

$$MSE(OSP_p) = Tr[Q(OSP_p)],$$

where

$$Q(OSP_p) = V_B \left( \Lambda^{-1} + \sum_{i \in OSP_p} \eta_i^{-1} v_i v_i^H \right)^{-1} V_B^H.$$

Here $OSP_p$ is the $p$th representative sampling subset, $V_B$ is the first $|B|$ columns of the eigenmatrix, $V$, $v_i$ is the $i$th row of $V$, and $\eta_i$ is the $i$th entry in $\eta$, which is the variance of the noise. Therefore, $MSE$ is calculated iteratively as nodes are added to a representative sampling subset. The reformulation is thoroughly described in Reference [32] as

$$MSE(OSP_{p,j} \cup v_s) = Tr[Q_j] - \frac{v_s^H Q_j V_{BS} Q_{j-1} v_s}{\eta_s + v_s^H Q_j v_s},$$

where

$$Q_j = Q_{j-1} - V_{BS}^H V_{BS} Q_{j-1} v_s v_s^H Q_{j-1} v_s.$$

ACM Transactions on Sensor Networks, Vol. 19, No. 4, Article 89. Publication date: June 2023.
Finding Representative Sampling Subsets in Sensor Graphs Using Time-series Similarities

**ALGORITHM 6:** Joint Iterative Partitioning, JIP

**Input** Time-series Similarity Graph $G = (V,E)$, number of representative sampling subsets $K$, upper bound of $MSE \epsilon$

**Output** Optimum Sampling Partition, OSP

1: Initialize $A$ as the adjacency matrix of $G$
2: Initialize $L$ as the graph Laplacian, $L = A - D$
3: Calculate eigendecomposition $L = \Lambda V V^{-1}$
4: $Q_0 = \Lambda, p = 0, j = 1$
5: $L_s = MSE(v_1)$
6: $L_{\text{index}} = \text{argsort}(L)$ (largest first)
7: while $L_{\text{index}} \neq \emptyset$
8:   $u = \text{pop}(L_{\text{index}} - 1)$
9:   $OSP_{p,j} = u$
10:   while $MSE(OSP_{p,j}) > \epsilon$ and $L_{\text{index}} \neq \emptyset$
11:      Calculate $Q_j$ according to (12)
12: for $i$ in $L_{\text{index}}$
13:     if $MSE(OSP_{p,j} \cup L_i) < \epsilon$ then
14:        $OSP_{p,j+1} = \{OSP_{p,j} \cup L_i\}$
15:        $j = 0, p = p + 1, \text{delete}(L_{\text{index}} = i)$
16:     goto 7
17: $OSP_{p,j+1} = \{OSP_{p,j} \cup \text{argmin}(MSE(L))\}$
18: remove chosen node from $L_{\text{index}}, j = j + 1$
19: if $MSE(OSP_{-1,-1}) > \epsilon$ then
20:   Split the nodes in $OSP_{-1}$ among the other sets
21: Return $OSP$

$Q_0 = \Lambda$, and $u$ is the index for the most recently added node:

$$MSE(v_u) = MSE(\emptyset \cup v_u).$$

(13)

In JIP, we iteratively create representative sampling subsets such that the MSE of each representative sampling subset is within the $MSE$ threshold. Therefore, we initially create a representative sampling subset by adding the node with the least $MSE$ according to Equation (13) until the $MSE$ of that representative sampling subset is less than the threshold. We, further, repeat this for the maximum number of possible representative sampling subsets. If there are any nodes left that cannot form a representative sampling subset on their own, then they are divided among existing representative sampling subsets. The pseudocode of JIP is shown in Algorithm 6 and the time complexity is $O(n^2B^2)$ [32]. In the results presented in this article, we have used a uniform random vector of $[0,0.001]$ for $\eta$.

In SIP, we aim to generate representative sampling subsets such that every representative sampling subset has similar reconstruction error to Equation (2). Given the number of representative sampling subsets, $K$, at each iteration, SIP creates the representative sampling subsets simultaneously unlike JIP. After sorting the nodes according to Equation (13), the $K$ nodes with the lowest $MSE$ are added to the representative sampling subsets, such that each representative sampling subset has been assigned one node. At each iteration, we add the best node according to Equation (11) to the representative sampling subset with the largest $MSE$ and repeat this for all the representative sampling subsets in the same order. We iterate this until all the nodes are allocated to a representative sampling subset. The pseudocode of SIP is given in Algorithm 7 and the time complexity is same as...
**Algorithm 7:** Simultaneous Iterative Partitioning, SIP

**Input** Time-series Similarity Graph $\mathcal{G} = (V,E)$, number of representative sampling subsets $K$

**Output** Optimum Sampling Partition, OSP

1. Calculate the eigenvalues, $\Lambda$ and eigenvectors, $V$ of $\mathcal{G}$
2. Initialize $OSP$ as a list of $K$ empty sets
3. $Q = \text{array}(\Lambda, K), err = \text{zeros}(K)$
4. $L_s = \text{MSE}(v_s)$
5. $L_{\text{index}} = [0,...,N]$
6. for $i$ in range($K$) do
   7. $m = \text{argmin}(L_s)$
   8. append $m$ to $OSP_i$
   9. $err[i] = L_s[m]$
   10. delete $L_s[m]$ and $L_{\text{index}}[m]$
11. Update $Q_i$ according to (12)
12. while $L_{\text{index}} \neq \emptyset$ do
   13. $j = \text{argmax}(err)$
   14. $m = \text{ones}(N)\text{max}(L_s)$
   15. for $i$ in $L_{\text{index}}$ do
       16. $m[i] = \text{MSE}(OSP_i \cup v_i)$
   17. append argmin($m$) to $OSP_j$
   18. $err[j] = \text{min}(m)$
   19. Update $Q_i$ according to (12)
   20. delete $L_{\text{index}} = m$
21. Return $OSP$

JIP, i.e., $O(n^2|\mathcal{B}|^2)$ [32]. The representative sampling subsets by JIP are (5,4,7), (2,6,3), and (0,8,1) and by SIP are (5,1,3), (4,8,0), and (2,6,7), respectively.

5.4 Minimum Frobenius Norm, Frob, and Maximum Parallelepiped Volume, Par

Tsitsvero et al. [71] proposed two different greedy-based sampling algorithms, $GFrob$ and $GPar$, which are based on eigendecomposition of graph Laplacian, $L = \text{\textit{adj}}V \Lambda^{-1}$, under the assumption that the signal is $\mathcal{B}$-bandlimited. $Frob'$ aims to find the representative sampling subset of size $|\mathcal{B}|$ that minimizes the Frobenius norm for the pseudo-inverse for the eigenvector matrix restricted to the first $|\mathcal{B}|$ columns and the rows corresponding to the chosen representative sampling subset, i.e.,

$$OSP_i = \arg\min_{SS_i \in S: |SS_i| = |\mathcal{B}|} \| (V_{\mathcal{B},SS_i})^+ \|_F,$$

where $V_{\mathcal{B},SS_i}$ is and the columns of the arbitrary sampling set $A_i$ of $V$, $V^+$ denotes the pseudo-inverse and $S$ is the set of all nodes. However, $GFrob$ generates only one representative sampling subset by adding $|\mathcal{B}|$ nodes in a greedy manner according to

$$m = \arg\min_q \sum_j \frac{1}{\sigma_j^2(V_{\mathcal{B},SS_q} + {q})},$$

where $\sigma_j^2(V)$ denotes the $j$th singular value of $V$.

In this article, we extend $GFrob$ as $Frob$ to generate $K$ representative sampling subsets by adding a node to a representative sampling subset on the basis of Equation (15) in a round-robin manner until all the nodes have been assigned to a representative sampling subset. An overview of $Frob$ is shown in

ACM Transactions on Sensor Networks, Vol. 19, No. 4, Article 89. Publication date: June 2023.
Algorithm 8: Minimum Frobenius Norm, Frob

**Input**
Time-series Similarity Graph $G = (V,E)$, number of representative sampling subsets $K$, bandwidth $B$

**Output**
Optimum Sampling Partition, OSP

1. Initialize OSP as an assembly of $K$ empty sets
2. Initialize $A$ as the adjacency matrix of $G$
3. Calculate eigen-decomposition $L = VAV^{-1}$
4. for $i$ in the number of nodes do
   5. $p = i \mod k$
   6. $m = \arg\min_q \sum_j \sigma_j^{-1}(U_B,OSP_p+\{q\})$
   7. $OSP_p \leftarrow OSP_p + \{m\}$
5. Return $OSP$

Algorithm 9: Maximum Parallelepiped Volume, Par

**Input**
Time-series Similarity Graph $G = (V,E)$, number of representative sampling subsets $K$, bandwidth $B$

**Output**
Optimum Sampling Partition, OSP

1. Initialize OSP as an assembly of $K$ empty sets
2. Initialize $A$ as the adjacency matrix of $G$
3. Calculate eigen-decomposition $L = VAV^{-1}$
4. for $i$ in the number of nodes do
   5. $p = i \mod k$
   6. $m = \arg\min_q \prod_j \lambda_j(V_B,OSP_p+\{q\}V_B,OSP_p+\{q\})$
   7. $OSP_p \leftarrow OSP_p + \{m\}$
5. Return $OSP$

Algorithm 8 and the time complexity is $O(n^3K^2)$, as it depends on the time complexity of singular value decomposition, which is $O(nK^2)$ [65], and we calculate the singular value decomposition for $n^2$ nodes. In the results presented in this article, we have used $B = V$. Similarly, Par selects $|B|$ nodes in a greedy manner according to

\[ m = \arg\min_q \prod_j \lambda_j(V_B,SS_p+\{q\}V_B,SS_p+\{q\}), \]  

where $\lambda_j(V)$ is the $j$th eigenvalue of $V$. For Par, we follow the same approach as proposed in Frob to identify the maximum number of possible representative sampling subsets that optimizes Equation (16) instead of Equation (15). An overview of Par is shown in Algorithm 9 and the time complexity is $O(n^3K^2)$ as it depends on the time complexity of eigen-value decomposition, which is $O((n/K)^3)$ [84], and the eigen-value decomposition is calculated for $n^2$ nodes. In the results presented in this article, we have used $B = V$. The representative sampling subsets generated by Frob and Par on $G_{dtw}$ for SN are the same, which are (8,2,7), (4,5,3), and (0,6,1), respectively.

5.5 AutoSubGraphSample

We have discussed six existing graph creation approaches for Phase-I and proposed six sampling approaches for Phase-II. However, the performance of these approaches differ across different datasets.
ALGORITHM 10: AutoSubGraphSample: Recommendation for Phase-I and Phase-II

Input $S = S_1, S_2, \ldots, S_n$ and time-series data of each sensor, $T^i$ is the time-series of $i$th sensor

1: Let, $e_d$ be the desired edge density
2: if $n < 90$ then
3: Use $P_{haar}$ in Phase-I
4: if $e_d < 0.40$ then
5: Use SMMR or Frob in Phase-II
6: else
7: Use SRel or Frob in Phase-II
8: else
9: if $e_d < 0.40$ then
10: Use $P_{nei}$ in Phase-I
11: Use SMMR or Frob in Phase-II
12: else
13: Use $P_{haar}$ in Phase-I
14: Use SRel or Frob in Phase-II

as they have different properties. Therefore, there is a need to automatically select the most suitable approach for Phase-I and Phase-II, respectively, given a dataset. In this subsection, we propose an Algorithm AutoSubGraphSample that considers the meta data of the dataset, such as number of sensors, $n$ and edge density, $E_d$ to do this. AutoSubGraphSample recommends $P_{haar}$ in Phase-I for any edge density in smaller networks (when $n$ is less than 90) and high edge density (when $E_d$ is greater than 0.40) in large networks (when $n$ is greater than 90). It recommends $P_{nei}$ in Phase-I for large networks (when $n$ is greater than 90) with low edge density (less than 0.40). It recommends SMMR or Frob when edge density is low and SRel or Frob, otherwise. Our decision of the threshold for $n$ as 90 and $E_d$ as 0.40 is based on our observations from our experiments, which we discuss in Section 6. The pseudocode of AutoSubGraphSample is shown in Algorithm 10. We validate the generalizability of AutoSubGraphSample to a dataset in Section 7.3 by comparing the performance of AutoSubGraphSample and the best manually selected algorithms for Phase-I and Phase-II on nine representative datasets.

6 EXPERIMENTAL SETUP

In this section, we describe the datasets used in experiments and discuss the different evaluation metrics.

6.1 Dataset Details and Preprocessing

The datasets used for our experiments are:

- $D_{epa}$: This dataset comprises 92 sensors and their edge relationships, which is simulated with EPANET [61]. EPANET is a tool for simulating water distribution network.
- $D_{temp}$: This dataset is based on a sensor network that comprises 74 sensors and their hourly temperature [22].
- $D_{pol}$: This dataset is based on a sensor network deployed at Aarhus, Denmark that comprises 37 sensors and their Ozone level recording.
- $D_{ws}$: We create a synthetic dataset of 100 nodes that follows the Watts-Strogatz Model [76] with $\beta = 0.5$. We create the data for each of the sensors such that it is strictly bandlimited in the graph Fourier domain.

2http://iot.ee.surrey.ac.uk:8080/datasets/pollution/index.html.
6.2 Evaluation Metrics

In this subsection, we discuss the different metrics that we used to compare the different approaches for Phase-I, Phase-II, and their combinations. For Phase-I, we compare the approaches in creating different graph topology given a dataset through average path length, clustering coefficient, edge density and measure how well the graph topology represents the dataset by total cumulative energy residual. Furthermore, we use reconstruction error to measure the performance of Phase-II and the combination of both. We do not discuss average path length, clustering co-efficient, and edge density further as they are well known. We detail how we calculate reconstruction error and total cumulative energy residual next.

6.2.1 Reconstruction Error. Reconstruction of signals on graphs is a well-known problem [51, 74] that provides an estimation of the whole graph, \( G \) by a representative sampling subset. For our experiments, we compare the sampling techniques on the basis of the reconstruction error. Given the OSP, which comprises \( K \) representative sampling subsets and \( t \) as the length of the time-series, we calculate the reconstruction error of a representative sampling subset of OSP, \( SP^i_u \), with the \( G \) by measuring the difference between the signal generated by \( SP^i_u \), \( \hat{x}^i \), with respect to the signal of \( G \), \( x \), at a time-stamp, say \( r \) as

\[
\|x^r - \hat{x}^i_r\|_2.
\]

We repeat this for all \( K \) and \( t \), respectively, for each sampling technique. We calculate the reconstruction error of OSP, Err(OSP,S) as the average of the total reconstruction error (TErr(OSP,S)) over \( K \) representative sampling subsets. TErr(OSP,S) is the sum of the average reconstruction error of each representative sampling subset, i.e., \( SP^i_u \) such that \( i \) ranges between 1 to \( K \). We calculate the reconstruction error of a representative sampling subset, \( SP^i_u \) as SamSErr\((SP^i_u,S)\) over \( t \) time stamps. Therefore, we calculate Err(OSP,S) as follows:

\[
\text{SamSErr}(SP^i_u,S) = \sum_{m=1}^{t} \frac{(||x^m - \hat{x}^i_m||_2)}{||x^m||},
\]

\[
TErr(OSP,S) = \sum_{i=1}^{K} \frac{(\text{SamSErr}(SP^i_u,S)/t)}{K}.
\]

Err(OSP,S) = \frac{TErr(OSP,S)}{K}.

For our results, we show the quartile of TErr(OSP,S), which represents the reconstruction error by a sampling technique.

6.2.2 Total Cumulative Energy Residual, TCER. TCER [36] measures the expected energy given a data set to understand how the graph structure represents the data by total cumulative energy of the data. Total cumulative energy of the data is measured by

\[
\mathcal{T}(X,Q) = \sum_{r=1}^{N} (N+1-r)||q^\top r X||^2,
\]

where \( Q \) is an orthogonal basis, TCER can then be calculated as

\[
1 - \frac{\mathcal{T}(X,V)}{\sum_{r=1}^{N} \sum_{r=1}^{N} \theta^2_{r,r}},
\]

where \( V \) is the eigen vectors of the graph Laplacian and \( \theta \) are the singular values of \( X \). The values of TCER are in the range of \([0,1]\) where a high value indicates that the graph represents the dataset well. We follow Reference [58] for the implementation.
Table 3. \(E_d\), Avg \(P_t\), and Avg CC for Different Values of Threshold for Phase-I Algorithms for \(D_{temp}\)

| Phase-I | \(E_d\) | Avg \(P_t\) | Avg CC | TCER | Th | Phase-I | \(E_d\) | Avg \(P_t\) | Avg CC | TCER | Th |
|---------|--------|-------------|--------|------|----|---------|--------|-------------|--------|------|----|
| \(G_{dtw}\) | 0.19   | inf         | 0.80   | 0.95 | 20 | 0.18   | inf        | 0.65 | 0.98 | 16 |
|         | 0.41   | inf         | 0.83   | 0.96 | 55 | 0.40   | inf        | 0.75 | 0.97 | 18 |
|         | 0.60   | inf         | 0.85   | 0.98 | 80 | 0.60   | 1.41       | 0.82 | 0.96 | 70 |
|         | 0.76   | 1.31        | 0.91   | 0.97 | 120| 0.75   | 0.87       | 1.25 | 0.88 | 160|
| \(G_{nei}\) | 0.20   | 1.79        | 0.81   | 0.77 | 8  | 0.39   | 1.61       | 0.87 | 0.92 | \(-1.11\times10^{-6}\) |
|         | 0.40   | 1.59        | 0.84   | 0.79 | 17 | 0.22   | 1.78       | 0.90 | 0.93 | \(-1\times10^{-7}\) |
|         | 0.60   | 1.38        | 0.84   | 0.78 | 28 | 0.59   | 1.41       | 0.89 | 0.90 | \(-1.1559\times10^{-6}\) |
|         | 0.75   | 1.247       | 0.87   | 0.90 | 37 | 0.77   | 1.23       | 0.96 | 0.92 | \(-1.1576\times10^{-6}\) |

7 RESULTS AND DISCUSSIONS

In this section, we initially evaluate the performance of the approaches for Phase-I and Phase-II separately followed by the validation of Algorithm AutoSubGraphSample on 4 representative datasets. We also analyze which combination of algorithms for Phase-I and Phase-II provides most optimal solutions. Last, we evaluate the performance of SubGraphSample when the whole time-series is not available and discuss additional experiments, such as studying the impact of \(E_d\) on reconstruction error, frequency analysis of \(G\).

7.1 Phase-I Results: Comparison of the Similarity Graph Creation Approaches

We evaluate the Phase-I algorithms by analyzing two specific properties of the similarity graph topology, TCER, and reconstruction error.

7.1.1 Evaluation of the Similarity Graph Topology. To analyze the properties of the graphs created by different graph creation approaches, we vary the values of the threshold for each approach of Phase-I to create graphs with a specific \(E_d\) and then study the average path length and clustering coefficient of these graphs. For our experiments, we consider four different edge densities, 0.20, 0.40, 0.60, and 0.75 for all the datasets. We show our observations for \(D_{temp}\) in Table 3. Our observations show that there is a significant variance in the properties of the graphs created by the different approaches even for the same \(E_d\) and same dataset. \(G_{dtw}\) is disconnected when the \(E_d\) is less than 0.70 and the number of sensors is greater than 70 and when the number of sensors is greater than 90 for any \(E_d\). \(G_{haar}\) is disconnected when the \(E_d\) is less than 0.40 and the number of sensors is above 90 and \(G_{nei}\) is always connected irrespective of the number of sensors and \(E_d\). Additionally, analyzing the possible values of threshold for different edge densities, we observe \(P_{dtw}\), \(P_{haar}\), and \(P_{nei}\) can create graphs with any \(E_d\). However, a very small difference in the values of threshold for \(P_{gsp}\), \(P_{corr}\), and \(P_{deconv}\) can create graphs with highly different \(E_d\). As previously discussed in Section 4.3, we observe that it is difficult for \(P_{gsp}\) to generate graphs of different edge densities given a dataset.

The reason for the performance of \(P_{corr}\) and \(P_{deconv}\) is that they utilize correlation of the time-series between a pair of sensors to create an edge and find similarity even when the values of the two time-series vary. Therefore, we do not consider \(P_{corr}\) and \(P_{deconv}\) henceforth. On the basis of our observations, we find that \(P_{nei}\) can be used irrespective of the dataset and \(E_d\), \(P_{haar}\) can be used only for datasets with small number of sensors or sensors when \(E_d\) is greater than 0.40 while \(P_{dtw}\) can be used for small networks. \(P_{gsp}\) can be used only if the threshold is tuned for different edge densities.

7.1.2 Total Cumulative Energy Residual. We compare the TCER value of a graph to that of a random graph for a dataset. Our observations indicate that \(G_{dtw}\) and \(G_{haar}\) always yield the best TCER values, around 0.88–0.98 irrespective of the \(E_d\) and the dataset and \(G_{nei}\) has the lowest TCER values. We show our observations in Table 3. We observe that the TCER for \(D_{ws}\) is bad irrespective
7.1.3 Reconstruction Error. We compare the reconstruction error of all the graph creation approaches on $D_{epa}$ for when $E_d$ are 0.20, 0.40, 0.60, and 0.75 and the number of representative sampling subsets are 5, 7, 10, and 13, respectively. We perform this experiment to understand how the choice of the graph creation approach affects the reconstruction error. For our experiments, we select Frob here. We perform this experiment only for connected graphs. Our observations indicate that $P_{corr}$ has significantly higher reconstruction error than others whereas $P_{nei}$, $P_{haar}$, and $P_{dtw}$ have similar results for $D_{epa}$. We observe $P_{nei}$ has a higher reconstruction error than $P_{haar}$, $P_{gsp}$, and $P_{dtw}$ for $D_{temp}$ and $D_{pol}$. Furthermore, $P_{dtw}$ and $P_{haar}$ performs the best followed by $P_{gsp}$ irrespective of the dataset and $E_d$. We show the results for $D_{epa}$ in Figure 3, where we observe that $P_{haar}$ performs the best followed by $P_{dtw}$, while $P_{nei}$ and $P_{corr}$ performs the worst.

7.1.4 Summary for Phase-I. We conclude that $P_{haar}$ followed by $P_{dtw}$ is the best choice for Phase-I for a dataset with more than 90 nodes and high $E_d$ (more than 0.40) and for any $E_d$ for a dataset with less than 90. However, for graphs with more than 90 nodes and $E_d$ less than 0.40, we recommend $P_{nei}$ followed by $P_{gsp}$. We use these observations to propose AutoSubGraphSample. As already discussed, we do not recommend $P_{corr}$ and $P_{deconv}$.

7.2 Phase-II Results: Comparison of the Sampling Techniques

We compare the performance of the sampling approaches for Phase-II and a random representative sampling subset selection algorithm on the basis of their solution for Equation (2). For our experiments, we compare the reconstruction error generated by the sampling techniques for each graph creation approach with different $E_d$, such as 0.20, 0.40, 0.60, and 0.75 and vary $K$ from 5 to 13 and calculate the reconstruction error quartile.

We find that irrespective of $K$ and the dataset, SRel ranks 1–3 among all sampling techniques when the $E_d$ is greater than 0.40, whereas SMMR ranks 1–3 when the $E_d$ is less than 0.40. SEMMR has similar mean reconstruction error as SMMR but, the maximum reconstruction error is much higher. SRel has around 0.1–0.40 reconstruction error when $E_d$ is greater than 0.40 and 0.20–0.60 otherwise. SMMR and SEMMR has around 0.2–0.40 when $E_d$ is less than 0.40 and 0.10–0.60
Fig. 4. Comparing the reconstruction error of sampling techniques and Random Sampling Approach for $K$ is 5, $E_d$ is 0.20 on $G_{\text{nei}}$ in panel (a) of $D_{\text{temp}}$, for $K$ is 7 when $E_d$ is 0.75 on $G_{\text{haar}}$ in panel (b) of $D_{\text{temp}}$, and $K$ is 5, $E_d$ is 0.75 on $G_{\text{haar}}$ in panel (c) of $D_{\text{pol}}$.

otherwise. SRel, SMMR, and SEMMR has the highest maximum reconstruction error for $D_{\text{epa}}$ when $E_d$ is greater than 0.60 and is $K$ greater than 7. MSV has around 0.20–0.40 reconstruction error when $E_d$ is around 0.75 but the performance degrades for low $E_d$ to around 0.30–0.80 reconstruction error. MSV also has the highest maximum reconstruction error at low $E_d$. On comparing JIP and SIP, which follow similar approaches, we observe that SIP yields better performance than JIP in every scenario irrespective of $K$, $E_d$ or dataset. On comparison with the other sampling approaches, we observe that SIP has around 0.40–0.60 reconstruction error when $E_d$ is high and 0.40–0.90 otherwise. Par has the worst performance among all sampling techniques. Although Frob ranks in the top 3–4 among all sampling techniques based on the minimum reconstruction error, it produces the maximum reconstruction error among all sampling techniques. As expected, we also observe that the reconstruction error increases with increase in $K$ irrespective of the sampling technique.

Based on our observations, we conclude that SRel is the best choice for graphs with high $E_d$ (greater than 0.40) and SMMR for graphs with $E_d$ less than 0.40. However, if we need to choose a sampling technique that performs irrespective of the $E_d$, then Frob should be selected. We use these observations to propose AutoSubGraphSample. Due to the huge number of results, we only show three representative examples in Figure 4.

7.3 Evaluation of AutoSubGraphSample

Based on our observations for Phase-I and Phase-II, we decide the values for $Th_n$ and $Th_e$ in Algorithm AutoSubGraphSample as 90 and 0.40, respectively. We analyze the generalizability of AutoSubGraphSample on 9 new representative datasets now.

1. $D_{\text{ps}}$: A dataset that records temperature of 55 sensors.
2. $D_{\text{in}}$: A dataset that records humidity of 54 sensors.
3. $D_{\text{hum}}$: A dataset that records humidity of 100 sensors.
4. $D_{\text{gas}}$: A dataset that records acetone of 16 sensors.
5. $D_{\text{sof}}$: A dataset that records temperature of 170 sensors.
6. $D_{\text{elec}}$: A dataset that records the electricity consumption of 124 sensors.

3https://archive.ics.uci.edu/ml/datasets.php.
4https://www.kaggle.com/hmavrodiev/air-quality-dataset?select=2017-09_bme280sof.csv.
5https://archive.ics.uci.edu/ml/datasets/.
6https://archive.ics.uci.edu/ml/datasets/.
7https://www.kaggle.com/hmavrodiev/sofia-air-quality-dataset.
8https://archive.ics.uci.edu/ml/datasets/.
Based on AutoSubGraphSample, we apply \( P_{\text{haar}} \) in Phase-I irrespective of the \( E_d \) and SMMR or \( SRel \) in Phase-II on the basis of \( E_d \) for \( D_{ps}, D_{in}, \) and \( D_{gas} \). However, for \( D_{hum}, D_{sof}, D_{syn}, D_{syn1}, \) and \( D_{elec} \), we apply \( P_{\text{haar}} \) in Phase-I and \( SRel \) in Phase-II when \( E_d \) is greater than 0.40 and \( P_{\text{nei}} \) in Phase-I followed by SMMR in Phase-II otherwise. We consider the number of representative sampling subsets, \( K \) as 5, 7, and 10 for all datasets except \( D_{gas} \). As \( D_{gas} \) comprises only 16 sensors, we consider \( K \) as 3, 5, and 7. We observe that the reconstruction error by AutoSubGraphSample is similar for all datasets except for \( D_{gas} \) and \( D_{elec} \). To our previous observations for other datasets irrespective of the number of representative sampling subsets and \( E_d \). Our observations indicate that the reconstruction error for \( D_{elec} \) is high when \( K \) increases more than 5, therefore we can increase the battery life for \( D_{elec} \) by maximum 5 times within the error margin of 0.40. We observe for \( D_{gas} \), the highest reconstruction error is greater than 0.60, whereas the average is within 0.40 for \( K \) values ranging from 3 to 7. The reason being the number of sensors being very less, i.e., only 16. We, further, observe that AutoSubGraphSample can ensure similar performance irrespective of the number of sensors as the number of sensors in the datasets range from 16 to 1,000. We show some representative examples of our observations in Figure 5. To understand the significance of AutoSubGraphSample, we compare the performance by AutoSubGraphSample and Manual Selection, i.e., if we manually select the best combination of Phase-I and Phase-II algorithms specifically for a dataset. We apply all combinations of Phase-I and Phase-II algorithms on a dataset and calculate the respective reconstruction errors for a specific \( E_d \). We select that combination of Phase-I and Phase-II algorithm, which provides the least reconstruction error as the Manual Selection. We repeat this for all the nine datasets when \( E_d \) is 0.25–0.75 and \( K \) as 3, 5, 7, and 10. We show our observations as shown in Table 4 for \( K = 5, E_d = 0.25 \) and 0.75 shows that AutoSubGraphSample can ensure similar results as compared to Manual Selection with a small margin of around 2–6% for \( D_{gas} \) and \( D_{elec} \) and same results for other datasets. Therefore, based on our observations, we can conclude that AutoSubGraphSample generalizes to a dataset irrespective of the size of the dataset and \( E_d \).

### 7.4 Comparing SubGraphSample with Exhaustive Search

In theory, identification of the optimum representative sampling subsets given the data of the sensors is possible through a joint exhaustive search for both the best graph topology and best sampling partition. However, this requires us to perform an exhaustive search for best sampling partition for every possible graph topology, which is so computationally expensive that we consider it to be infeasible. Therefore, we consider this in two phases, where in Phase-I, we search for an optimum graph topology and in Phase-II, we search for the optimum sampling partition given the optimum graph topology. For our experiments, we consider a subset of eight sensors of \( D_{pat} \), namely, \( D_{ex} \), as exhaustive analysis is not possible on the complete dataset.

To identify the optimum graph, we explore the relationship between existing graph topology measures, like average path length, clustering coefficient, and TCER with optimal graph. Based on our observations, we conclude that the TCER values are indirectly proportional to reconstruction error, i.e., higher the TCER values, lower is the reconstruction error. Furthermore, if different graphs have similar TCER values, then we observe that as the average path length decreases, the reconstruction error also decreases. We calculate the TCER for all possible connected graphs to a precision of 2 significant digits for \( D_{ex} \). We consider the graph that has the highest TCER and shortest average as "optimal".
Fig. 5. Comparing the reconstruction error of $D_{gas}$ in panel (a) for $E_d = 0.25$, $D_{hum}$ in panel (b) for $E_d = 0.25$, $D_{pre}$ in panel (c) for $E_d = 0.60$, $D_{syn}$ in panel (d) for $E_d = 0.60$, $D_{sof}$ in panel (e) for $E_d = 0.60$, and $D_{syn1}$ in panel (f) for $E_d = 0.75$. The x-axis represents K and y-axis represents the reconstruction error.

Table 4. Comparing the Reconstruction Error by AutoSubGraphSample and Manual Selection on $D_{ps}$, $D_{gas}$, $D_{in}$, $D_{elec}$, $D_{sof}$, and $D_{hum}$ when $E_d$ is 0.25 and 0.75 for $K = 5$

| Dataset | $E_d$ | AutoSubGraphSample | Manual | Dataset | $E_d$ | AutoSubGraphSample | Manual |
|---------|-------|---------------------|--------|---------|-------|---------------------|--------|
| $D_{ps}$ | 0.25  | 0.51                | 0.51   | $D_{gas}$ | 0.25  | 0.49                | 0.47   |
|         | 0.75  | 0.61                | 0.61   |         | 0.75  | 0.56                | 0.56   |
| $D_{hum}$ | 0.25 | 0.31                | 0.31   | $D_{in}$ | 0.25  | 0.33                | 0.32   |
|         | 0.75  | 0.42                | 0.42   |         | 0.75  | 0.49                | 0.49   |
| $D_{sof}$ | 0.25 | 0.41                | 0.41   | $D_{elec}$ | 0.25 | 0.47                | 0.45   |
|         | 0.75  | 0.38                | 0.38   |         | 0.75  | 0.50                | 0.47   |

Table 5. $E_d$, $P_l$, CC, and TCER for $D_{ex}$

| Method | $E_d$ | Avg $P_l$ | Avg CC | TCER |
|--------|-------|-----------|--------|------|
| $G_{opt}$ | 0.64  | 1.36      | 0.45   | 0.88 |
| $G_{rel}$ | 0.46  | 1.54      | 0.82   | 0.82 |
| $G_{haar}$ | 0.50  | 1.79      | 0.87   | **0.86** |
| $G_{dtw}$ | 0.50  | 1.79      | 0.87   | **0.86** |
| $G_{gsp}$ | 0.57  | 1.54      | 0.80   | 0.83 |

path length as the optimal graph, $G_{opt}$. As the $E_d$ of $G_{opt}$ is 0.64, we try to find graphs with similar $E_d$ using the proposed methods. We show the $E_d$, average path length, clustering co-efficient, TCER of $G_{opt}$ with $G_{rel}$, $G_{haar}$, $G_{dtw}$, and $G_{gsp}$ in Table 5, which indicates that $G_{haar}$ and $G_{dtw}$ has the most similar TCER values with $G_{opt}$. As the graphs produced with $P_{haar}$ and $P_{dtw}$ are identical, so we only show results for $P_{dtw}$ henceforth.
To find the $O_s$, we search all possible sampling partitions on $G_{opt}$ such that the maximum reconstruction error is the lowest. We perform an exhaustive search to find $O_s$ on $G_{nei}, G_{dtw}, G_{dspr},$ and $G_{opt}$. Our observations as shown in Figure 6 indicate that $P_{haar}$ and $P_{dtw}$ ensures the least reconstruction error. Therefore, our observations indicate that it is possible to find a graph that gives lower reconstruction error than the graph with the highest TCER. To evaluate the different sampling algorithms for Phase-II, we compare the reconstruction error by Frob, MSV, SIP, SMMR, and SRel on $G_{opt}$ in Figure 7. Our observations indicate that by Frob, SIP and SMMR has the least reconstruction error with respect to Opt. However, these observations varies with network size and edge density. As it is not possible to confirm every scenario of different edge densities and for different network sizes with exhaustive analysis, we compare the performance of the graph creation approaches and sampling algorithms on a synthetic dataset whose representative sampling subsets are already provided next in Section 7.5.

### 7.5 Comparison of SubGraphSample with optimum Sampling Sets

We now evaluate how close the representative sampling subsets found by SubGraphSample are to the optimum sampling sets, $O_s$. As we do not have $O_s$ for any real dataset, we construct a dataset such that we know $O_s$. We assume the optimal number of sampling sets, $K$ as 6, the total number of sensors, $N$ as 40, the length of the time-series as 10, sensors as $S_1, S_2, \ldots, S_N$, and we denote this dataset as $D_{st}$. We simulate $D_{st}$ such that it records temperature. We generate $O_s$ of 6, $O_1, O_2, \ldots, O_6$ sampling sets by randomly allocating each sensor to a $O_i$ on the basis of $D_{st}$. Based on $O_s$, we generate the time-series of $S$ such that while the mean values of the distributions vary by 3–5 between different sampling sets, i.e., the constructed sampling sets are indeed the optimal. We calculate the reconstruction error of $O_s$ for $D_{st}$ to understand the performance of $O_s$. As we do not know the true $E_d$ and the graph topology of $D_{st}$, which is required to calculate the reconstruction error, we consider four different $E_d$, such as 0.20, 0.40, 0.60, and 0.75, and the four similarity graph creation algorithms, $P_{dtw}, P_{haar}, P_{nei}$, and $P_{gsp}$. We compare $P_{dtw}, P_{nei}, P_{gsp}$, and $P_{haar}$ on the basis graph topology, TCER, and the reconstruction error for all $E_d$ in Table 6. Our observations shows that $O_s$ has minimum reconstruction error when $E_d$ is 0.60 and similarity graph creation approach is $P_{haar}$. On comparing the sampling techniques on $G_{haar}$ when $E_d$ is 0.60, our observations as shown in Figure 8 indicate that SRel produces similar reconstruction error to $O_s$. Therefore, the combination of $P_{haar}$ and SRel can ensure most similar results to $O_s$. On the basis of our observations from Section 7.5 and
Fig. 8. Comparison of the reconstruction error of when $E_d$ is 0.60 and $P_{haar}$ in Phase-I for $D_{st}$.

Table 6. Average Path Length ($P_l$), Clustering Co-efficient (CC), TCER, and the Threshold Value for $D_{st}$

| Phase-I | $E_d$ | Avg $P_l$ | Avg CC | TCER | Th | Phase-I | $E_d$ | Avg $P_l$ | Avg CC | TCER | Th |
|---------|-------|-----------|--------|------|----|---------|-------|-----------|--------|------|----|
| $G_{dtw}$ | 0.23  | inf       | 0.91   | 0.49 | 120| $G_{haar}$ | 0.23  | 2.64      | 0.81   | 0.97 | 60 |
|         | 0.40  | 2.26      | 0.84   | 0.97 | 210|          | 0.40  | 1.77      | 0.77   | 0.96 | 100|
|         | 0.60  | 1.47      | 0.89   | 0.97 | 330|          | 0.61  | 1.39      | 0.81   | 0.93 | 185|
| $G_{nei}$ | 0.23  | 1.78      | 0.89   | 0.89 | 5  | $G_{gap}$ | 0.20  | 2.05      | 0.39   | 0.90 | 0.01 |
|         | 0.40  | 1.60      | 0.86   | 0.85 | 9  |          | 0.40  | 1.69      | 0.52   | 0.89 | -4e-06|
|         | 0.60  | 1.39      | 0.84   | 0.87 | 15 |          | 0.59  | 1.47      | 0.71   | 0.89 | -1.1e-05|

In this subsection, we find that the proposed recommendations for Algorithm AutoSubGraphSample can ensure most similar results to $O_s$. For example, we observe that $P_{haar}$ in Phase-I, SMMR or Frob in Phase-II has the best performance. Although it is not possible to confirm every scenario by exhaustive analysis, our results from empirical analysis supports the recommendations by Algorithm AutoSubGraphSample when $P_{nei}$ is used in Phase-I and when SRel could be used in Phase-II.

7.6 Studying the Impact of $E_d$ on Reconstruction Error

To study the relationship between $E_d$ and reconstruction error, we calculate the reconstruction error for different edge densities $0.20–0.75$. For this experiment, we consider Frob for Phase-II and $K$ as 7. Our observations differ with respect to datasets. For example, we observe that the higher the $E_d$, the lower is the reconstruction error for $D_{epa}$ as shown in Figure 9(a). However, we observe that the reconstruction error is the highest when $E_d$ is 0.2 and decreases with increase in $E_d$ for $D_{temp}$ as shown in Figure 9(b). We did not observe any relationship between $E_d$ and reconstruction error in $D_{pol}$ and $D_{ws}$ as shown in Figure 9(c). Based on these observations, we conclude there is an optimal $E_d$ for which the reconstruction error is the lowest for a dataset. However, the optimal $E_d$ differs across datasets.

7.7 Frequency Analysis

To understand the performance of the graph creation approaches discussed in Phase-I, we visualize the Frequency transform of the graphs created given a dataset. As discussed in Section 3.2, Graph Fourier Transform (GFT) is the eigen decomposition of the graph Laplacian, $L$ into eigenvalues, $\Lambda$ and eigenvectors, $V$, i.e., GFT of $L$ is $L = \Lambda \Lambda^{-1}$. Additionally, GFT of the graph signal at the $k$–th time-stamp, $x^k$, is $\hat{x}^k$, which is defined as $\hat{x}^k = V^{-1}x^k$. Therefore, given a dataset, the GFT of the
Finding Representative Sampling Subsets in Sensor Graphs Using Time-series Similarities

Fig. 9. Comparison of the reconstruction error for different edge densities for $G_{gsp}$ on $D_{epa}$ in panel (a), $G_{nei}$ on $D_{temp}$ in panel (b), and $G_{dtw}$ in panel (c). The x-axis represents the sampling techniques and y-axis represents the reconstruction error.

Fig. 10. Graph Fourier transforms of $G_{nei}$ with $E_d 0.43$ on $D_{pol}$ in panel (a), $G_{haar}$ with $E_d 0.40$ on $D_{temp}$ in panel (b), and $G_{dtw}$ with $E_d 0.60$ on $D_{pol}$ in panel (c).

optimal graph topology should comprise of the maximum number of possible distinct eigenvalues, which are evenly spread. Additionally, the GFT of the optimal graph topology should be such that the lower the eigenvalues, the higher the amplitudes and vice-versa. Our observations indicate that the GFT of $G_{haar}$ and $G_{dtw}$ ensures optimal graph topology created given a dataset and the $E_d$ whereas $G_{nei}$ has fewer distinct eigenvalues and therefore, is not optimal for $D_{pol}$ and $D_{temp}$. We show some representative examples of our observations in Figure 10.

7.8 Evaluation of AutoSubGraphSample on Partial Time-series

In our previous experiments, we generate $G$ based on the complete time-series given a dataset. In this subsection, we analyze the performance of AutoSubGraphSample when the complete time-series is not available. Therefore, in this experiment, we select a fraction, $p$, of the time-series for which to generate the similarity graph, $G'$, and identify $K$ sampling sets on $G'$. We compare the reconstruction error of $G'$ with $G$ for different values of $p$ while keeping $K$, $E_d$, Phase-I and Phase-II approaches constant. We repeat this experiment by varying $p$ as 0.75, 0.50, and 0.25 of the time-series, $K$ between 5 and 13 and $E_d$ between 0.20 and 0.75, respectively. We repeat this for all the 13 datasets. Our observations as shown in Figure 11 indicate that the difference in average reconstruction error is minimal (1–9%) across datasets irrespective of $p$. Therefore, we can conclude that AutoSubGraphSample can ensure similar increase in battery longevity within the error bound irrespective of the size of the time-series. We intuitively believe the reason being AutoSubGraphSample identifies a list of sampling sets such that similar sensors are allocated to different sampling
Fig. 11. Comparing the reconstruction error of $D_{temp}$ when $K = 7$, $E_d = 0.75$ in panel (a), $D_{pol}$ when $K = 5$, $E_d = 0.60$ in panel (b), $D_{sof}$ when $K = 5$, $E_d = 0.60$ in panel (c), $D_{hum}$ when $K = 7$, $E_d = 0.40$ in panel (d), $D_{ps}$ when $E_d = 0.75$, when $K = 5$ in panel (e), and $D_{temp}$ when $K = 5$, $E_d = 0.20$ in panel (f). The x-axis represents the fraction of time-series, $p$, and y-axis represents the reconstruction error.

sets by utilizing the similarity graph topology. Therefore, although the sampling sets might vary as we vary $p$ of the time-series, it does not impact the reconstruction error. Furthermore, there is generally a correlation in the sensor’s data irrespective of the $p$, which is another reason that the performance of AutoSubGraphSample is not impacted and AutoSubGraphSample can handle small fluctuations in the sensing data easily by the utilization of similarity graph topology.

7.9 Identifying the Maximum Number of Sampling Subsets, $K$

In this subsection, we compare the performance of Phase-II for Equation (1), i.e., we find the $K$ generated by a sampling approach given $\epsilon$. However, the proposed sampling approaches except SRel, SMMR and SEMMR cannot provide a solution for Equation (1) as they focus on identifying the maximum error given $K$ and therefore, they require $K$ to be pre-specified and cannot be modified to identify the maximum number of sampling subsets given $\epsilon$. Therefore, we select SRel, SMMR, and SEMMR for this experiment. We compare $K$ generated by SRel, SMMR, and SEMMR for each graph creation approach with different $E_d$, such as $0.20$–$0.75$, and vary $\epsilon$ from $0.40$–$0.60$. Our observations indicate that SRel generates the maximum value for $K$ for high $E_d$, whereas SMMR generates the maximum value for low $E_d$. Although SEMMR performs similar to SMMR for all datasets, the performance for SEMMR is the worst among all for $D_{pol}$ irrespective of $E_d$. We show our observations on $G_{dtw}$, $G_{rei}$, and $G_{haar}$ for $E_d$ as $0.20$, $0.40$, $0.75$ on $D_{pol}$ and $D_{temp}$ datasets for $\epsilon$ as $0.40$ and $0.50$, respectively, in Table 7.

8 CONCLUSIONS AND FUTURE WORKS

In this article, we propose SubGraphSample, which finds the maximum number of representative sampling subsets given a sensor graph. By finding the maximum number of representative sampling
Finding Representative Sampling Subsets in Sensor Graphs Using Time-series Similarities

Table 7. Number of Sampling Sets, $K$, Generated by SRel, SMMR, and SEMMR when the Mean Reconstruction Error, $\epsilon$, Is Given for $D_{\text{temp}}$ and $D_{\text{pol}}$

| Dataset | $\epsilon$ | $E_d$ | Phase-I | Phase-II | $K$ | Dataset | $\epsilon$ | $E_d$ | Phase-I | Phase-II | $K$ |
|---------|------------|-------|---------|----------|-----|---------|------------|-------|---------|----------|-----|
| $D_{\text{temp}}$ | 0.40 | 0.75 | $\mathcal{G}_{\text{nei}}$ | SRel | 6 | $D_{\text{temp}}$ | 0.40 | 0.75 | $\mathcal{G}_{\text{haar}}$ | SRel | 5 |
| | | | | SMMR | 4 | | | | SMMR | 4 |
| | | | | SEMMR | 4 | | | | SEMMR | 4 |
| $D_{\text{temp}}$ | 0.40 | 0.20 | $\mathcal{G}_{\text{nei}}$ | SRel | 3 | $D_{\text{temp}}$ | 0.40 | 0.40 | $\mathcal{G}_{\text{haar}}$ | SRel | 9 |
| | | | | SMMR | 4 | | | | SMMR | 10 |
| | | | | SEMMR | 4 | | | | SEMMR | 10 |
| $D_{\text{pol}}$ | 0.50 | 0.75 | $\mathcal{G}_{\text{nei}}$ | SRel | 7 | $D_{\text{pol}}$ | 0.50 | 0.75 | $\mathcal{G}_{\text{dtw}}$ | SRel | 8 |
| | | | | SMMR | 7 | | | | SMMR | 7 |
| | | | | SEMMR | 4 | | | | SEMMR | 5 |
| $D_{\text{pol}}$ | 0.50 | 0.20 | $\mathcal{G}_{\text{nei}}$ | SRel | 3 | $D_{\text{pol}}$ | 0.50 | 0.20 | $\mathcal{G}_{\text{dtw}}$ | SRel | 4 |
| | | | | SMMR | 4 | | | | SMMR | 4 |
| | | | | SEMMR | 3 | | | | SEMMR | 3 |

subsets, we can alternate querying between these and, thus, increase battery longevity significantly. Unlike existing sampling approaches, SubGraphSample do not require prior knowledge of the similarity of the sensors and automatically identifies the maximum number of representative sampling subsets. We explore six graph creation approaches and propose six sampling approaches in SubGraphSample. However, the suitability and performance of a graph creation approach and sampling approach varies across datasets. Therefore, we propose Algorithm AutoSubGraphSample, which can autoselect the most suitable approaches given a sensor graph, and we further show the generalizability of AutoSubGraphSample given a dataset. We evaluate all possible combinations of approaches of SubGraphSample on four datasets, which show that the best combination of algorithms can provide 5–13 times increase in battery life within a 20–40% error bound.

As a future work, we will extend AutoSubGraphSample to handle multivariate time-series and scale to large time-series using deep learning-based time-series embedding. Furthermore, we aim to merge the current two phases into one in a deep reinforcement learning-based model.

REFERENCES

[1] Abubakar Abid and James Zou. 2018. Autowarp: Learning a warping distance from unlabeled time series using sequence autoencoders. Retrieved from https://arXiv:1810.10107.
[2] Martin Aigner. 1999. A characterization of the Bell numbers. Discrete Math. 205, 1-3 (1999), 207–210.
[3] Naomi S. Altman. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. Amer. Stat. 46, 3 (1992), 175–185.
[4] Shahzad Ashraf, Sehrish Saleem, and Tauqeer Ahmed. 2020. Sagacious communication link selection mechanism for underwater wireless sensors network. Int. J. Wirel. Microw. Technol. 10, 2 (2020), 12–25.
[5] Yuanchao Bai, Fen Wang, Gene Cheung, Yuji Nakatsukasa, and Wen Gao. 2020. Fast graphsampling set selection using gershgorin disc alignment. IEEE Trans. Signal Process. 68 (2020), 2419–2434.
[6] Donald J. Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In Proceedings of the KDD Workshop. 359–370.
[7] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. J. Stat. Mech.: Theory Exper. 2008, 10 (2008), P10008.
[8] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 335–336.
[9] Luiz F. O. Chamon and Alejandro Ribeiro. 2017. Greedy sampling of graph signals. IEEE Trans. Signal Process. 66, 1 (2017), 34–47.
[10] F. K.-P. Chan, A. W.-C. Fu, and Clement Yu. 2003. Haar wavelets for efficient similarity search of time-series: With and without time warping. IEEE Trans. Knowl. Data Eng. 15, 3 (2003), 686–705.
[11] Lei Chen and Raymond Ng. 2004. On the marriage of lp-norms and edit distance. In Proceedings of the 30th International Conference on Very Large Data Bases-Volume 30. 792–803.

[12] Lei Chen, M Tamer Özsu, and Vincent Oria. 2005. Robust and fast similarity search for moving object trajectories. In Proceedings of the ACM SIGMOD International Conference on Management of Data. 491–502.

[13] Siheng Chen, Rohan Varma, Aliaksei Sandryhaila, and Jelena Kovačević. 2015. Discrete signal processing on graphs: Sampling theory. IEEE Trans. Signal Process. 63, 24 (2015), 6510–6523.

[14] Yueguo Chen, Mario A. Nascimento, Beng Chin Ooi, and Anthony K. H. Tung. 2007. Spade: On shape-based pattern detection in streaming time series. In Proceedings of the IEEE 23rd International Conference on Data Engineering. IEEE, 786–795.

[15] Yan-Bin Chen, Ido Nevat, Pengfei Zhang, Sai Ganesh Nagarajan, and Hung-Yu Wei. 2018. Query-based sensors selection for collaborative wireless sensor networks with stochastic energy harvesting. IEEE Internet Things J. 6, 2 (2018), 3031–3043.

[16] Alessandro Chiumento, Nicola Marchetti, and Irene Macaluso. 2019. Energy efficient WSN: A cross-layer graph signal processing solution to information redundancy. In Proceedings of the 16th International Symposium on Wireless Communication Systems (ISWCS). IEEE, 645–650.

[17] Mario Coutino, Sundeep Prabhakar Chepuri, and Geert Leus. 2018. Subset selection for kernel-based signal reconstruction. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'18). IEEE, 4014–4018.

[18] Padraig Cunningham and Sarah Jane Delany. 2021. K-nearest neighbour classifiers—a tutorial. ACM Comput. Surveys 54, 6 (2021), 1–25.

[19] Marco Cuturi. 2011. Fast global alignment kernels. In Proceedings of the 28th International Conference on Machine Learning (ICML’11). 929–936.

[20] Marco Cuturi and Mathieu Blondel. 2017. Soft-dtw: A differentiable loss function for time-series. In Proceedings of the International Conference on Machine Learning. PMLR, 894–903.

[21] Kousik Das, Sovan Samanta, and Madhumangal Pal. 2018. Study on centrality measures in social networks: A survey. Soc. Netw. Anal. Min. 8, 1 (2018), 1–11.

[22] Howard J. Diamond, Thomas R. Karl, Michael A. Palecki, C. Bruce Baker, Jesse E. Bell, Ronald D. Leeper, David R. Easterling, Jay H. Lawrimore, Tilden P. Meyers, Michael R. Helfert et al. 2013. U.S. climate reference network after one decade of operations: Status and assessment. Bull. Amer. Meteorol. Soc. 94, 4 (2013), 485–498.

[23] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst. 2015. Laplacian matrix learning for smooth graph signal representation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’15). IEEE, 3736–3740.

[24] Hilmi E. Egilmez, Eduardo Pavez, and Antonio Ortega. 2017. Graph learning from data under Laplacian and structural constraints. IEEE J. Select. Topics Signal Process. 11, 6 (2017), 825–841.

[25] Christos Faloutsos, Mudumbai Ranganathan, and Yannis Manolopoulos. 1994. Fast subsequence matching in time-series databases. ACM SIGMOD Rec. 23, 2 (1994), 419–429.

[26] Soheil Feizi, Daniel Marbach, Muriel Médard, and Manolis Kellis. 2013. Network deconvolution as a general method to distinguish direct dependencies in networks. Nature Biotechnol. 31, 8 (2013), 726–733.

[27] Akshay Gadde, Aamir Anis, and Antonio Ortega. 2014. Active semi-supervised learning using sampling theory for graph signals. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 492–501.

[28] Bugra Gedik, Ling Liu, and S. Yu Philip. 2007. ASAP: An adaptive sampling approach to data collection in sensor networks. IEEE Trans. Parallel Distrib. Syst. 18, 12 (2007), 1766–1783.

[29] Zhichen Gong and Huanhuan Chen. 2018. Sequential data classification by dynamic state warping. Knowl. Info. Syst. 57, 3 (2018), 545–570.

[30] Ram P. Gupta. 1968. Bounds on the Chromatic and Achromatic Numbers of Complimentary Graphs. Technical Report. Department of Statistics, North Carolina State University.

[31] Sepideh Hassan-Moghaddam, Neil K. Dhingra, and Mihailo R. Jovanović. 2016. Topology identification of undirected consensus networks via sparse inverse covariance estimation. In Proceedings of the IEEE 55th Conference on Decision and Control (CDC'16). IEEE, 4624–4629.

[32] Josefine Holm, Federico Chiariotti, Morten Nielsen, and Petar Popovski. 2021. Lifetime maximization of an internet of things (IoT) network based on graph signal processing. IEEE Communications Letters 25, 8 (2021), 2763–2767.

[33] Tommi Jaakkola, Mark Diekhans, and David Haussler. 2000. A discriminative framework for detecting remote protein homologies. J. Comput. Biol. 7, 1-2 (2000), 95–114.

[34] Weiwei Jiang. 2020. Timeseriesclassification:Nearestneighborversusdeeplearningmodels. SNAppl.Sci. 2, 4 (2020), 1–17.

[35] Vassilis Kalofolias. 2016. How to learn a graph from smooth signals. In Artificial Intelligence and Statistics. PMLR, 920–929.
Finding Representative Sampling Subsets in Sensor Graphs Using Time-series Similarities

[36] Vassilis Kalofolias, Andreas Loukas, Dorina Thanou, and Pascal Frossard. 2017. Learning time varying graphs. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’17). IEEE, 2826–2830.

[37] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. 2019. Multivariate LSTM-FCNs for time series classification. Neural Netw. 116 (2019), 237–245.

[38] Yoon Hak Kim. 2020. QR factorization-based sampling set selection for bandlimited graph signals. Signal Process. (2020), 107847.

[39] Tim Van Der Lee, Georgios Exarchakos, and Sonia Heemstra De Groot. 2020. Distributed reliable and energy-efficient scheduling for Ir-wpans. ACM Trans. Sensor Netw. 16, 4 (2020), 1–20.

[40] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. 2010. Empirical comparison of algorithms for network community detection. In Proceedings of the 19th International Conference on World Wide Web. 631–640.

[41] Ian X. Y. Leung, Pan Hui, Pietro Lio, and Jon Crowcroft. 2009. Towards real-time community detection in large networks. Phys. Rev. E 79, 6 (2009), 066107.

[42] Tianxiao Liao, Wen-Qin Wang, Bang Huang, and Jan Xu. 2019. Learning Laplacian matrix for smooth signals on graph. In Proceedings of the IEEE International Conference on Signal, Information and Data Processing (ICSIDP). IEEE, 1–5.

[43] Deyu Lin, Quan Wang, Weidong Min, Jianfeng Xu, and Zhiquiang Zhang. 2020. A survey on energy-efficient strategies in static wireless sensor networks. ACM Trans. Sensor Netw. 17, 1 (2020), 1–48.

[44] Jason Lines and Anthony Bagnall. 2015. Time series classification with ensembles of elastic distance measures. Data Min. Knowl. Discov. 29, 3 (2015), 565–592.

[45] Chong Liu, Kui Wu, and Jian Pei. 2007. An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation. IEEE Trans. Parallel Distrib. Syst. 18, 7 (2007), 1010–1023.

[46] Jose A. Rosendo Macias and Antonio Exposito. 2005. Efficient computation of the running discrete Haar transform. IEEE Trans. Power Deliv. 21, 1 (2005), 504–505.

[47] Lei Mao and Lisa Jackson. 2016. Selection of optimal sensors for predicting performance of polymer electrolyte membrane fuel cell. J. Power Sources 328 (2016), 151–160.

[48] Gonzalo Mateos, Santiago Segarra, Antonio G. Marques, and Alejandro Ribeiro. 2019. Connecting the dots: Identifying network structure via graph signal processing. IEEE Signal Process. Mag. 36, 3 (2019), 16–43.

[49] Shinnosuke Matsuo, Xiaomeng Wu, Gantugs Atarsaikhan, Akisato Kimura, Kunio Kashino, Brian Kenji Iwana, and Seiichiro Uchida. 2021. Attention to warp: Deep metric learning for multivariate time series. Retrieved from https://arXiv:2103.15074.

[50] Michael D. Morse and Jignesh M. Patel. 2007. An efficient and accurate method for evaluating time series similarity. In Proceedings of the ACM SIGMOD International Conference on Management of Data. 569–580.

[51] Suniti K. Narang, Akshay Gadde, and Antonio Ortega. 2013. Signal processing techniques for interpolation in graph structured data. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 5445–5449.

[52] Jyoti Narwariya, Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and T. V. Vishnu. 2020. Meta-learning for few-shot time series classification. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD. ACM, 28–36.

[53] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M. F. Moura, and Pierre Vandergheynst. 2018. Graph signal processing: Overview, challenges, and applications. Proc. IEEE 106, 5 (2018), 808–828.

[54] Guillermo Ortiz-Jiménez, Mario Coutino, Sundey Prabbakar Chepuri, and Geert Leus. 2018. Sampling and reconstruction of signals on product graphs. In Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP’18). IEEE, 713–717.

[55] John Paparrizos, Chunwei Liu, Bruno Barbarioli, Johnny Hwang, Ikbrudaya Edian, Aaron J. Elmore, Michael J. Franklin, and Sanjay Krishnan. 2021. VergeDB: A database for IoT analytics on edge devices. In Proceedings of the Conference on Innovative Data Systems Research (CIDR’21).

[56] Wenjie Pei, Hamdi Dibeklioğlu, David J. M. Tax, and Laurens van der Maaten. 2017. Multivariate time-series classification using the hidden-unit logistic model. IEEE Trans. Neural Netw. Learn. Syst. 29, 4 (2017), 920–931.

[57] Nathanaël Perraudin, Johan Paratte, David Shuman, Lionel Martin, Vassilis Kalofolias, Pierre Vandergheynst, and David K. Hammond. 2014. GSPBOX: A toolbox for signal processing on graphs. Retrieved from https://arXiv:1408.5781.

[58] Nathanaël Perraudin, Benjamin Ricaud, David I. Shuman, and Pierre Vandergheynst. 2018. Global and local uncertainty principles for signals on graphs. APSIPA Trans. Signal Inf. Proces. 7 (2018).

[59] Gilles Puy, Nicolas Tremblay, Rémi Gribonval, and Pierre Vandergheynst. 2018. Random sampling of bandlimited signals on graphs. Appl. Comput. Harmon. Anal. 44, 2 (2018), 446–475.

[60] LewisA.Rossmanetal.2000.EPANET2:Usersmanual.

[61] Britta Ruhnau. 2000. Eigenvector-centrality—Anode-centrality? Phys.Rev.E.

[62] Akie Sakiyama, Yuichi Tanaka, Toshihisa Tanaka, and Antonio Ortega. 2019. Eigendecomposition-free sampling set selection for graph signals. IEEE Trans. Signal Process. 67, 10 (2019), 2679–2692.

ACM Transactions on Sensor Networks, Vol. 19, No. 4, Article 89. Publication date: June 2023.
[64] Stan Salvador and Philip Chan. 2007. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.* 11, 5 (2007), 561–580.

[65] Alok Sharma, Kuldirp K. Paliwal, Seiya Imoto, and Satoru Miyano. 2013. Principal component analysis using QR decomposition. *Int. J. Mach. Learn. Cybernet.* 4, 6 (2013), 679–683.

[66] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and P. Vanderghenst. 2013. The emerging field of signal processing on graphs. *IEEE Signal Process. Mag.* (2013).

[67] Vladimir Stojov, Nikola Koteli, Petre Lameski, and Eftim Zdravevski. 2018. Application of machine learning and time-series analysis for air pollution prediction. *Proceedings of the 6th IASTED International Conference on Communications, Internet, and Information Technology (CIIT’18).*

[68] Nurgazy Sulaimanov and Heinz Koeppl. 2016. Graph reconstruction using covariance-based methods. *EURASIP J. Bioinform. Syst. Biol.* 2016, 1 (2016), 19.

[69] Yuichi Tanaka, Yonina C. Eldar, Antonio Ortega, and Gene Cheung. 2020. Sampling signals on graphs: From theory to applications. *IEEE Signal Process. Mag.* 37, 6 (2020), 14–30.

[70] V. A. Traag. 2015. Faster unfolding of communities: Speeding up the Louvain algorithm. *Phys. Rev. E* 92 (2015), 032801. Issue 3.

[71] Mikhail Tsitsvero, Sergio Barbarossa, and Paolo Di Lorenzo. 2016. Signals on graphs: Uncertainty principle and sampling. *IEEE Trans. Signal Process.* 64, 18 (2016), 4845–4860.

[72] Arun Venkitaraman, Saikat Chatterjee, and Peter Händel. 2019. Predicting graph signals using kernel regression where the input signal is agnostic to a graph. *IEEE Trans. Signal Info. Process. Netw.* 5, 4 (2019), 698–710.

[73] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. 2002. Discovering similar multidimensional trajectories. In *Proceedings of the 18th International Conference on Data Engineering*. IEEE, 673–684.

[74] Xiaohan Wang, Pengfei Liu, and Yuantao Gu. 2015. Local-set-based graph signal reconstruction. *IEEE Trans. Signal Process.* 63, 9 (2015), 2432–2444.

[75] Xiaoyue Wang, Abdullah Museen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. 2013. Experimental comparison of representation methods and distance measures for time series data. *Data Min. Knowl. Discov.* 26, 2 (2013), 275–309.

[76] Robin Wilson and John J. Watkins. 2013. *Combinatorics: Ancient & Modern*. Oxford University Press, Oxford, UK.

[77] Yi-Leh Wu, Divyakant Agrawal, and Amr El Abbadi. 2000. A comparison of DFT and DWT-based similarity search in time-series databases. In *Proceedings of the 9th International Conference on Information and Knowledge Management*. 488–495.

[78] Zhengzheng Xing, Jian Pei, and Eamonn Keogh. 2010. A brief survey on sequence classification. *ACM SIGKDD Explor. Newsletter* 12, 1 (2010), 40–48.

[79] K.-B. Yu. 1991. Recursive updating the eigenvalue decomposition of a covariance matrix. *IEEE Trans. Signal Process.* 39, 5 (1991), 1136–1145.

[80] K.-B. Yu. 1991. Recursive updating the eigenvalue decomposition of a covariance matrix. *IEEE Trans. Signal Process.* 39, 5 (1991), 1136–1145.

[81] Eftim Zdravevski, Petre Lameski, Riste Mingov, Andrea Kulakov, and Dejan Gjorgjevikj. 2015. Robust histogram-based feature engineering of time series data. In *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 381–388.

[82] Eftim Zdravevski, Petre Lameski, Vladimir Trajkovik, Andrea Kulakov, Ivan Chorbev, Rossitza Goleva, Nuno Pombo, and Nuno Garcia. 2017. Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering. *IEEE Access* 5 (2017), 5262–5280.

Received 29 June 2022; revised 23 January 2023; accepted 8 April 2023