A MULTI-DOMAIN SPLITTING FRAMEWORK FOR TIME-VARYING GRAPH STRUCTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

The Graph Signal Processing (GSP) methods are widely used to solve structured data analysis problems, assuming that the data structure is fixed. In the recent GSP community, anomaly detection on datasets with the time-varying structure is an open challenge. To address the anomaly detection problem for datasets with a spatial-temporal structure, in this work, we propose a novel graph multi-domain splitting framework, called GMDS, by integrating the time, vertex, and frequency features to locate the anomalies. Firstly, by introducing the discrete wavelet transform into vertex function, we design a splitting approach for separating the graph sequences into several sub-sequences adaptively. Then, we specifically design an adjacency function in the vertex domain to generate the adjacency matrix adaptively. At last, by utilizing the learned graphs to the spectral graph wavelet transform, we design a module to extract vertices features in the frequency domain. To validate the effectiveness of our framework, we apply GMDS in the anomaly detection of actual traffic flow and urban datasets and compare its performances with acknowledged baselines. The experimental results show that our proposed framework outperforms all the baselines, which distinctly demonstrate the validity of GMDS.

1 INTRODUCTION

In the analysis of spatial-temporal structured data, graph signal processing (GSP) is an important type of method, taking advantages of graph model to represent the structure. Graph has abundant features, which should be captured by appropriate rules. These methods can be divided into 2 categories by whether graph structure is variable or not.The first, in the applications with invariant graph structure, there are no temporal differences on graph to consider. For instance, in traffic flow forecasting proposed by Yu et al. (2018), the road map is regarded as the graph structure (adjacency matrix), which is generated by fixed longitude and latitude without temporal features. The second, in the tasks that consider spatial-temporal structured datasets, the graph structure must be influenced by time lapse, such as, traffic (Guo et al. (2020)), urban, Covid-19, etc. Thus, in the tasks belong to second category, it is inevitable to consider the graph structure as time-varying. Also, time-varying data structures could be more common than the invariant cases in real world.

Then, anomaly detection is of great importance in modern data science, as singular or anomaly data is ubiquitous among real-world datasets, which are time-series collected from distributed sensor or receiver networks. Especially, detecting the anomalies in the time-varying structure is then becomes an open challenge Atluri et al. (2018), Bergman & Hoshen (2020). The applications include traffic events detection, neighbors discovery, pandemic spreading analysis and social network clustering. Significantly, the applications in traffic are the most considered task Zhang et al. (2020). The meticulous and effective analysis of graph structure are important for valuable detecting of anomalies of vertices Djenouri et al. (2019). However, how to divide graph sequence into appropriate sub-sequences adaptively is the key point in the challenge of temporal dynamic graph structure capturing.

Therefore, to break the limitation of time-varying structure analysis, the time-vertex-frequency multi-domain graph splitting framework, called GMDS, which is proposed to capture the time-varying graph structure. The first part is an augmented dickey-fuller (ADF) test based data preprocessing. The second is the discrete wavelet transform based graph time-series local splitting. Based on ARIMA, the third part of the GMDS is the graph generation to capture the variable graph structure that segmented by the second part. The last part is the global detection, which is designed as anomaly detection to extract the dependencies and eigenvalues among graph. The output of last part is the anomalies. And the implementations of our framework without training part are accessible at https://github.com/Zehua-Yu/TVF-anomaly-detection.

Our contributions are summarized as follows:

• A novel framework based on the appropriate temporal splitting in multi-domain called GMDS is proposed for time-varying graph structure anomaly detection.

• Through the dynamic adaptive partition graph, our framework can analyze the data of different periods more carefully and accurately. It breaks the limitation of global unified graph generation and prevents local features from being submerged in the global scope.

• The experiments show that GMDS has the generalization to different types of traffic data.

2 Related Work

There are many methods with abundant mechanisms and modules that have been applied to anomaly detection. Zhang et al. (2020) has done a survey of urban anomaly analysis approaches, including description, detection and prediction. These methods can be divided into sets by different standards. Djenouri et al. (2019) summarize the anomalies detection algorithms that are applicated in urban traffic. All the methods are divided into 2 categories: flow outlier detection and trajectory outlier detection. The front category includes statistical, similarity, and pattern mining methods. The latter one includes offline and online processing.

However, the above methods consider little about the spatial-temporal and interaction topology analysis among data. Graph is a widly used structure, which is good at modeling the data with complicated multi-meta to capture the saptial-temporal topology and other types linkages among vertices. Sofuoglu & Aviyente (2021) introduce low-rank matrix recovery on graphs into low-rank tensor recovery to imply the anomaly detection in spatial-temporal data. Tasneem et al. (2019) presents numerous examples and proofs to illustrate the validity of the theorems of using antimagic graph labeling for splitting. Then, Ioannidis et al. (2021) present GraphSAC to effectively detect anomaly vertices in graph with complicated dependency features. ITGCN have been proposed by Yu et al. (2021) to capture the interactions among vertices, and performed well in Covid-19 daily confirmed cases forecasting.

3 Preliminaries

In this section, we recap the preliminaries in graph signal, ARIMA, discrete wavelet transform (DWT) and graph wavelet transform.

3.1 Graph Signal

In our works, the graph signals are defined on the graphs, which are weighted, connected and undirected. Following the Zheng et al. (2019), the graphs considered in this work are denoted as $G = (V, E, A)$, where $V = v_0, v_1, ..., v_{N-1}$ is the set of vertices that contain features, E is the set of edges, $A \in \mathbf{R}^{N\times N}$ is the adjacency matrix that represent weights. The adjacency matrix is symmetric, shown as $A(i, j) = A(j, i)$, where $A(i, j) \in \mathbf{R}$ denotes the weight assigned to the edge $e(i, j)$ between the vertices v_i and v_j. The degree matrix D of the graph G is defined as a diagonal matrix whose $D(i, i)$ is given by the degree of vertex i, i.e., $D(i, i) = deg(v_i)$, where $deg(v_i)$ is the degree of vertex i. All the graphs considered in this work are undirected weighted graph without self-loops as shown in Fig. 1. The linkages between vertices are all nonnegative.

3.2 ARIMA

Auto regressive integrated moving average (ARIMA) is widly used in sequences modeling or forecasting tasks. No significant difference with stationarity and with a rapidly decreasing autocorrelation function are two satisfied requirements for ARMA modeling. Thus, for the data without the requirements above, the difference is introduced to solve this problem in ARIMA. Let d be a nonnegative integer, $\{X_t\}$ is $ARIMA(p, d, q)$, if $Y_t \triangleq (1-B)^d X_t$ is the causal $ARMA(p, q)$ process, the $\{X\}$ satisfies $\phi^*(B)X_t \equiv \phi(B)(1-B)^d X_t = \theta(B)Z_t, \{Z_t\} \sim WN(0, \sigma^2)$ where $\phi(z)$ and $\theta(z)$

are p-order and q-order polynomials respectively, B is the coefficient setted by mission requirement, d is the differences order.

3.3 DISCRETE WAVELET TRANSFORM

Signals defined on time are discrete. At the same time, the scale parameter a and time shift parameter b are also processed by discretion $(a = 2^j, b = k2^j, (j, k \in \mathbb{Z}))$. Under this definition, mother wavelet and other wavelets are all discrete, represented by $\phi(n)$ and $\phi_{j,k}(n)$ respectively, where $\phi_{j,k}(n)$ is defined as $\phi_{j,k}(n) = 2^{-j/2}\phi(2^{-j}n - k), \quad (j, k \in \mathbb{Z})$ where j, k are calculated according to the specific data. Let $f(n)$ be the input, the DWT of $f(n)$ according to $\phi_{j,k}(n)$ is $C_{j,k} \triangleq DWT_\phi f(2^j, k2^j) = \sum_{n=-\infty}^{\infty} f(n)\bar{\phi_{j,k}}(n) = 2^{-j/2}\sum_{n=-\infty}^{\infty} f(n)\bar{\phi}(2^{-j}n - k), \quad (j, k, n \in \mathbb{Z})$ where n is the total number of discrete time.

3.4 SPECTRAL GRAPH WAVELET TRANSFORM

Following the definition of Graph Fourier Transform Hammond et al. (2011), the spectral graph wavelet transform can be defined as follow. The transform will be determined by the choice of a kernel function $g : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, which is analogous to Fourier domain wavelet $\hat{\psi}^*$ in $(T^s\delta_a)(x) = 1/s\psi^*(a - x/s)$ This kernel g should behave as a band-pass filter, i.e. it satisfies $g(0) = 0$ and $lim_{x\rightarrow\infty}g(x) = 0$. We will defer the exact specification of the kernel g that we use until later.

The spectral graph wavelet transform is generated by wavelet operators that are operator-valued functions of the Laplacian L, which is $L = 1 - D^{-1/2}AD^{1/2}$. L is a real symmetric matrix, it has a complete set of orthonormal eigenvector denoted by χ_l for $l = 0, ..., N-1$, with associated eigenvalues λ_l. A measureable function of a bounded self-adjoint linear operator on a Hilbert space using the continuous functional calculus is defined to achieved using the spectral representation of the operator, which is equivalent to the graph Fourier transform defined in Hammond et al. (2011). In particular, for spectral graph wavelet kernel g, the wavelet operator $T_g = g(L)$ acts on a given function f by modulating each Fourier mode as $\hat{T_g f}(l) = g(\lambda_l)\hat{f}(l)$. Employing the inverse Fourier transform yields $(T_g f)(m) = \sum_{l=0}^{N-1} g(\lambda_l)\hat{f}(l)\chi_l(m)$. The wavelet operators at scale t is then defined by $T_g^t = g(tL)$. The spectral graph wavelets are then realized through localizing these opertors by applying them to the impulse on a single vertex, i.e. $\psi_{t,n} = T_g^t\delta_n$. Using the orthonormality of the χ_l, it can be seen that the wavelet coefficients can also be achieved directly from the wavelet opertors, as $W_f(t, n) = (T_g^t f)(n) = \sum_{l=0}^{N-1} g(t\lambda_l)\hat{f}(l)\chi_l(n)$. The above is the main part of the application of SGWT in our method. See Hammond et al. (2011) for other notes and proofs.

Then in next section, following the definition above, we will introduce the details of our method.

4 GMDS FRAMEWORK

In this section, the multi-domain splitting framework for graph structure is proposed, and we describe the detailed architecture of it shown in Figure 1. Firstly, we use preprocessing module to clean the original data, and form them into the appropriate format of next layer. Secondly, the Local Splitting is used to split the graph time-series that follow our framework order by local DWT analysis. Then, after the splitting, we use graph generation module to generate the adjacency matrix of each splitted graph time-series. In the last layer, we use spectral graph wavelet transform (SGWT) to implement the frequency domain global detection. And the output of our framework is the detection results that involve in the anomaly vertices finding, vertices classification and related application analysis. Furthermore, the local part is detection in each splitted period, the global part stands for splitting on whole time-series.

4.1 PREPROCESSING

With much of invalid eigenvalues and monitor stations, original data is used to construct the graph time-series model. Thus, we design Algorithm 1 that contains the discard of invalid points, vertices choosing and sequences stationary analysis to implement the preprocessing. According to the data characteristic requirements of the designed framework, we have formulated the following principles

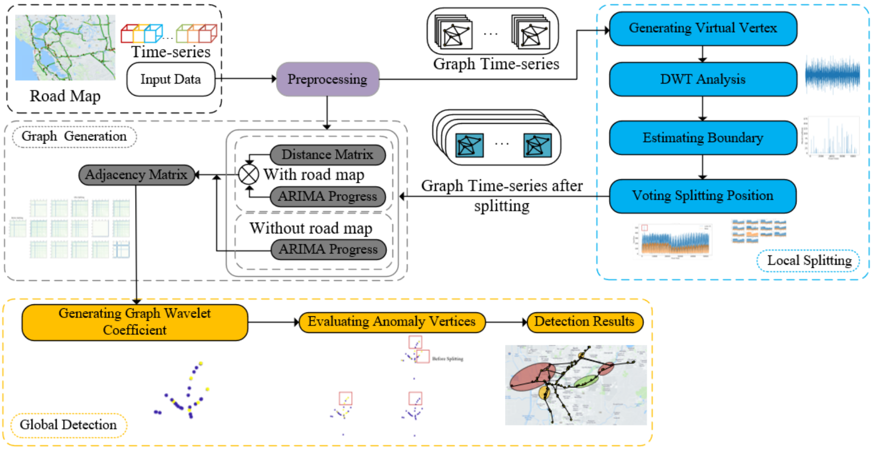


Figure 1: The flow chart of GMDS

Algorithm 1 Preprocessing

```
Input: Time-series;
Output: Graph Time-series;
Removing invalid vertices (unqualified vertices);
Extracting the target features from datasets;
Sampling the crucial vertices;
Stationarity and Validation Testing(3 requirements);
Calculating $m$;
return $G_{underpre}, m$;
```

for preprocessing.
1) Remove the vertices that with incompleted time-series.
2) Remove the vertices with below anomaly records that caused by the collecting: individual values are too large or too small relative to the whole, compared with other series, and the value of the whole series does not change with time.
3) Remove the vertices that are determined to be non-stationary and cannot be uniformly stabilized. The output of this layer is the graph time-series that needs splitting.

4.2 Local Splitting

Algorithm 2 Local Splitting

```
Input: Preprocessed Graph Time-series;
Output: Splitted Graph Time-series;
Extracting the target features from datasets;
Generate Virtual Vertex;
DWT analysis
for $j = 1; j <= vertices; j++$ do
    Discrete Wavelet Transform(DWT) for vertex $j$ time-series;
end for
Estimate Boundary
for $j = 1; j <= vertices; j++$ do
    Find the max wavelet time among vertex $j$;
end for
Set limitation value(Here, we have obtained that the number of splitted segments is from the experimental experience. Then, take the first n points with the largest wavelet coefficients, and the wavelet coefficient value of the $nth$ point is the boundary.);
Splitting the graph time-series follow the boundaries.
return Splitted graph time-series;
```

Algorithm 2 is designed to implement the Local Splitting module. Due to the graph wavelet transform needs a center vertex, we first generate the virtual vertex as the overall situation of the graph, the generation equation is

$$F_{v_N(t)} = \frac{1}{N} \sum_{i=0}^{N-1} F_{v_i(t)}, \quad t = 1, 2, 3, ... \tag{1}$$

where $F_{v_N(t)}$ is the feature of virtual vertex at t, N is the amount of vertices, $F_{v_i(t)}$ is the feature of i^{th} vertex among graph at t. So, the feature vector of virtual vertex is $F_{v_N} = \left\{F_{v_N(1)}, F_{v_N(2)}, ..., F_{v_N(t)}\right\}$. Then, analyzing the graph time-series by discrete wavelet transform. Each vertex of the graph is chosen as a local part, the time-series of that vertex are the eigenvalues. By discrete wavelet transformation, we can map the features of input from time domain to spectral domain. Then, each local part will be represented by a spectral signal. Secondly, considering the peaks of each spectral, the module calculates the splitting boundary and position by training. At last, the method votes the best trainable splitting position. The graph time-series is splitted to several parts, which have obvious different features compare to others.

4.3 Graph Generation

Algorithm 3 Graph Generation

```
Input: Splitted Graph Time-series, Road Map Matrix;
Output: Adjacency Matrix;
i=1
while $i <= n$ do
    $t - s = X^m_{i,t-n+1+m}, ..., X^m_{i,t}$;
    $u_i = ARMA(t - s, order = (p, q))$;
    $i + +$;
end while
//$u_i = (\phi_1, ..., \phi_p, \theta_1, ..., \theta_q)$, $G_x = (X_1, ..., X_n)$
for $j = 1; j <= n; j + +$ do
    for $k = 1; k <= n; k + +$ do
        $Dist_e(X_k, X_j) = \|\alpha_k - \alpha_j\|_2$;
    end for
end for
for $i = 1; i <= n \times n; i + +$ do
    Calculating the adjacency matrix $A$ by equation;
end for
return $A$;
```

Algorithm 3 is designed to implement the graph generation part. Adjacency matrix is the representation of the relation of graph between vertices. Thus, it is important to capture the most fitted representation of graph. This module combines physical relationships of graph and interaction coefficients to generate the adjacency matrix that plays an important role in the hole framework. The distance matrix comes from the road network, calculated by fixed Latitude and longitude coordinates of each station. This part captures the spatial relationships among graph.

The ARIMA progress is to capture the hidden interaction among vertices set. $ARMA(p, q)$ is used to generate the parameters vector Φ_i of each time-series, which represent the vertices uniquely in the Euclidean space.

The vectors are like, $u_k = (\Phi_k, \Theta_k) = (\phi_1, ..., \phi_p, \theta_1, ..., \theta_q) = (\alpha_1, ..., \alpha_{p+q})$, $k = 1, 2, ..., n; p, q = 1, 2, ...$ where p is the order of modeling, ϕ is the AR parameter, θ is the MA parameter. Basing on the u_k and Hannan & Rissanen (1982), we define a multivariate equation $f(u_k)$ as,

$$\begin{aligned} f(u_k) &= |\sum_{i=1,j=1}^{p,q} (\phi_i x_{t-i} + \theta_j \epsilon_{t-j}) - (x_t - \epsilon_t - C)| \\ &= \begin{cases} |\sum_{i=1,j=1}^{p,q} (\alpha_i x_{t-i} + \alpha_{j+p} \epsilon_{t-j}) - (x_t - \epsilon_t - C)|, \\ \qquad\qquad\qquad\qquad\qquad\qquad p, q \neq 0 \\ |\sum_{i=1}^{p} (\alpha_i x_{t-i}) - (x_t - C)|, \qquad q = 0, p \neq 0 \\ |\sum_{j=1}^{q} (\alpha_j \epsilon_{t-j}) + (\epsilon_t + C)|, \qquad p = 0, q \neq 0 \end{cases} \end{aligned} \tag{2}$$

where, x_t is the time sequence, ϵ is noise. C is constant, which consists the expectation of x_t. The goodness-of-fit between the u_k and the real model of time-series is inversely correlated with the value of $f(u_k)$. Then, the equation of U_k is, $U_k = \arg\min_{u_k} \left\{ f(u_k) | \exists u_k^* : \lim_{u_k \to u_k^*} f(u_k) = 0 \right\}$.

Ideally, we can find a u_k^*, which make $f(u_k^*) = 0$. However, in experiments, we have to find the most approximate value. In a graph with n nodes, each node has a unique U_k. The Euclidean distance between any pair of vertices u_k and u_l, $Dist_e(U_k, U_l)$ is given by, $Dist_e(U_k, U_l) = \sqrt{\sum_{j=1}^{p+q}(\alpha_{k,j} - \alpha_{l,j})^2} = \|\alpha_k - \alpha_l\|_2 \quad k, l = 1, 2, ..., n$. Here, if the data consists of features and spatial road map matrix R, the adjacency matrix is $A = Dist_e R$. If the data without the physical road map or fixed spatial matrix, the adjacency matrix is $A = Dist_e$.

4.4 Global Detection

Algorithm 4 Global Detection

```
Input: Adjacency Matrix $A$;
Output: Detection Results(Anomaly Vertices);
Generate Graph Wavelet Coefficient;
Setting adaptive boundary of classification of different anomaly levels by magnitude;
Detecting the anomaly vertex and classify them into different level categories.
return Detecting results;
```

After generating the adjacency matrix of each graph, global detection layer is needed to detect the anomaly vertices and classify the target set of vertices into different groups by training. The details are in Algorithm 4. In this module, we use SGWT to calculate the graph wavelet coefficients, the specific process have been mentioned in section 2. Finally, our framework is trained by using the binary cross-entropy loss functionTrinh et al. (2019), and optimized by Adam.

5 Experiments and Performances

In this section, we will deploy our framework in traffic and urban datasets. We set 3 parts of experiments to illustrate the performances and precision of our method. Firstly, we elaborate the performances by 5 metrics, among GMDS and baselines on the NYC urban datasets. Then, based on the PeMSD dataset, a small-scale dataset is sampled from PeMSD 3 to verify the effectiveness and rationality of our framework by visualization. At last, we apply GMDS to PeMSD 3 datasets to detect the anomaly vertices in different important period with landmark events. The details of our experiments are as follows.

5.1 Datasets and Settings

PeMSD : In our experiments, the traffic datasets are collected from California highway by the Caltrans Performance Measurement System(PeMS) PeM in the rate of one sampling every 5 minutes. The eigenvalues in traffic experiments are the total flows. We collected 70 points in PeMSD3 (district 3) by preprocessing module of our framework. Then, we resample 6 vertices among them for the small-scale experiment. The time range of data is from Jan/1st/2020 to Jun/30th/2020. Due to the less of recognized anomalies ground truth for this dataset, all the trainable parameters are set by experience.

NYC urban : This dataset consists of 2 parts. One is generated by the bike sharing system in NYC, which has 340 bike stations and about 7,000 bikes. The labels include time, bike ID, station ID, and an indication of check-out or return. The location of each station is also disclosed to the public. Another is generated by over 14,000 taxicabs in NYC. Labels include pick-up and drop-off locations and times, the duration and distance of each trip, taxi ID and the number of passengers for each trip. The aim of introducing this dataset in our experiments is to evaluate the performances of GMDS, thus the settings and preprocessing method come from Zheng et al. (2015). And we separate the data into 30%, 30% and 40% for training, validation and test.

All experiments are compiled by Python, and tested on a Windows10 workstation (CPU: Intel(R) i7-10700 GPU: NVIDIA RTX 2070 RAM: 32G).

There are 2 kinds of parameters that need to be set in this experiment. The first category of parameters can be trained according to the characteristics of the data. Such as, the difference order m in the preprocessing stage, the wavelet coefficient bounds of the local splitting module, the number of splitting periods. Others are needed to set by experience, such as the ARMA order in the generation of adjacency matrix, and the wavelet scale and order of the wavelet coefficients of the graph.

In evaluation experiments, the settings of our method are the same to the experiments below, except for the boundary of splitting is 100, and the anomaly limitations are trained in detection.

In PeMSD 3 anomaly detection, we set the number of time interval splitting to less than 15, and choose the splitting boundary according to the order of wavelet coefficients from large to small. In PeMSD 3, the splitting limit is 170. In the generation of adjacency matrix, we set ARMA (5,0). For the whole time series stabilization, we use first-order difference. In the global detection part, we use the generated virtual vertex as the center, set the wavelet scale to 3 and the order of Chebyshev polynomial to 20.

5.2 Baselines and Metrics

We compare our method with 4 classical and widely used anomaly detection methods on NYC urban datasets. The baselines are as follows:

Ind + int: An One-Class Support Vector Machine(SVM) based two-step method, which use a similarity-based algorithm and an one-class SVM based algorithm Zhang et al. (2018).

EE: Data is used to fit an elliptic envelope first. And Mahalanobis distance is introduced as the anomaly score Rousseeuw & Driessen (1999).

Neighbor: Use the Euclidean distance between the values of a vertex and the mean value of its nearby vertices as the anomaly score.

LRT: Fit a Poisson distribution on historical data and use likelihood ratio test as the anomaly score.

For evaluating the performances of our framework and baselines, we introduced 5 metrics into detection results analyzing. Before demonstrating of metrics, some basic statistics are needed explain. TP is true positive, FP is false positove, TN is true negative, FN is false negative. Positive means the anomaly vertices among detection results, negative means the normal vertices among detection results. True means the anomaly vertices in ground truth, false means the normal vertices in ground truth.

Therefore, the metrics are $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{ALL}$, $F1 - measure = \frac{2 \times P \times R}{P+R}$, $FPR = \frac{FP}{FP+TN}$ and $TPR = \frac{TP}{TP+FN}$, where P is precision, R is recall, ALL is all the sample features of vertices, TPR is the proportion of real positive samples in all positive samples currently assigned to positive samples, FPR is the proportion of real negative samples in the total number of negative samples in the category of positive samples wrongly assigned.

5.3 Performances evaluation

Table 1 demonstrates the comparison between our method and baselines on NYC urban data in metrics that introduced above. Hit rate here is $\frac{Hitevents}{Allevents}$. Following the same procedure in Zheng et al. (2015), a detected anomaly is regarded as a correct recall if the anomaly has an overlap with a reported event in spatial-temporal space. The results show that our method outperforms all other baselines.

Table 1: Results on Urban data anomaly detection

Method	P	F1-M	FPR	TPR	Hit Rate
ind + int	0.24	0.34	0.76	0.60	60%(12/20)
EE	0.15	0.19	0.90	0.25	25%(5/20)
Neighbor	0.21	0.14	0.56	0.10	10%(2/20)
LRT	0.16	0.20	0.90	0.25	25%(5/20)
GMDS(proposed)	**0.88**	**0.81**	**0.07**	**0.75**	**75%(15/20)**

After the evaluation comparison with baslines, for more intuitive and persuasive verification, we next deploy our method in a small-scale real-world traffic data that are sampled from PeMSD3. And we illustrate the splitting results, adjacency matrix, and detection results by visualization. Also, in order to show the structure of the framework more comprehensively, we will show the phased results of each step of our method in this experiment.

The yellow semi-transparent area in the Figure 4 is used to mark the area involved in the selected station (In our method, a virtual vertex is generated, but it is not identified in the image. Because it has no specific physical location). According to our method, after the data preprocessing, local splitting is the step to divided graph time-series into different parts in time domain.

Figure 2 in appendix shows the result of local splitting. This group of figures is the time-frequency information of two vertices, which as the splitting time voting vertices. Among them, Figure 2(a)(b) are the time domain characteristic values of two vertices selected as splitting time position after DWT respectively. Figure 2(c)(d) are the DWT spectrums of 2 vertices. In order to demonstrate the difference in the frequency spectrum more clearly, we have performed numerical significance processing on it, and the results are Figure 2(e)(f). It is possible to clearly find the moments with abnormal numerical characteristics, and these abnormal moments are the splitting points of voting. We mark these moments by black line in Figure 2(e)(f), by red line in Figure 2(c)(d), and by black box in Figure 2(a)(b). Figure 2(g) is the schematic diagram of all time domain splitting. Different colors represent different time periods. The red line in Figure 2(g) is the trained bound (DWT coefficient) of voting. According to the experiment of this module, it can be seen that the splitting of our method is effective, and the anomaly time can be accurately located.

The next step of the proposed method is to generate the adjacency matrix corresponding to each period, that is, to generate the representation of the graph structure. In this regard, in order to intuitively show the necessity and effect of splitting, Figure 3 in appendix shows the visual matrix of adjacency matrix. It can be seen from the figure that the graph in different periods has different detail characteristics.

As a comparison, Figure 3(d) shows the adjacency matrix that takes all half year data as a whole without splitting. It can be seen from the first three subgraphs that different periods have their own characteristics, but the results without splitting cannot show these different details. Splitting can locate the details of different periods, which is more conducive to accurate analysis.

The last step is global anomaly detection. The detection results are shown in Figure 4(a)(b)(c) in appendix, correspond to the 3 piecewise adjacency matrices in Figure 3. Here, we first divide the 6 vertices into 2 categories according to the magnitude of the graph wavelet coefficients. The red area is great change, and the yellow area is small change. In these 2 regions, we mark vertices with the biggest change in red, and it is detected as an anomaly vertex. For a more comprehensive comparison, Figure 5 in appendix shows the sequence diagram of the time domain, frequency domain and frequency domain saliency processing of the virtual vertex and the other 3 vertices.

According to the detection results, our method can locate the anomaly vertices in each time period, and can classify them accurately. At the same time, comparing with the results without splitting(Figure 4(d)), we can detect the events at each time in more details.

The above experiments are carried out on 2 different types of datasets. The first is to compare our method with baselines on multiple metrics. In addition, another is performed to visualize detection process of our method on PeMSD3. The results above have proven that GMDS has detection accuracy, rationality, generality and usability. Next, we will apply our framework to traffic flow historical data analysis in the Sacramental, CA, US.

6 Discussion

Figure 6 demonstrates the anomaly results of detection among PeMSD 3. The results show the traffic flow change in different periods. The colors in the figure, red, green, blue and yellow represent the degree of change from high to low. But, in different period, the same color does not represent the same degree. The colorful oval areas represent the average degree of change of traffic flow. The level of colors are the same to above. It has to be noticed that, the detected anomalies represent that the numerical variation of anomaly vertices are larger than other data entities. Following this criterion, the sudden expansion and decrease of traffic flow are all the anomaly phenomenon. And the ellipse areas are all the diagram of vertices without classification.

Basing on the events collected from official websites cag NAV in table 2, the analysis of our detection results are as follows. The events we select including holidays, Covid-19 containment measurements and traffic events. The traffic events consist of incidences, alerts and news. Due to the large amount of traffic events, we only list the numbers of events in each period. Because of

Table 2: The list of traffic, Covid-19 and holiday events in Sacramental, CA from 01/01/2020 to 06/30/2020.

No.	Time	Amount	Events category
1	Jan 1	1	New Year's Day
2	Jan 1 – Jan 15	42	Traffic archive
3	Jan 20	1	Martin Luther King, Jr. Day
4	Jan 16 – Jan 31	33	Traffic archive
5	Feb 17	1	Washington's Birthday
6	Feb	39	Traffic archive
7	Mar 1– Mar 20	2	Traffic archive
8	Mar 20 – May18	56	Covid-19 related
9	Mar 21– Mar 31	13	Traffic archive
10	Apr	16	Traffic archive
11	May 1 – May 18	9	Traffic archive
12	May 25	1	Memorial Day
13	May 19 – Jun 30	69	Covid-19 related
14	May 19– Jun 30	57	Traffic archive

the direct affection to the traffic flow, the holidays and traffic events are selected. Covid-19 is a big event, which can indirectly affect the traffic flow. Therefore, we collected these 3 categories of events.

In Figure 6, our method splits the time into 4 periods in the first half of 2020. Then, as another scale representation, Figure 6(e) is the result without splitting. In this range of time, our results illustrate that the state of traffic flow start to change in the period 1. And in period 1, the changeable center is the center of sacramental. Then, in period 2, the traffic flow variation range is gradually spread to a wider range, based on the urban center. In period 3, the maximum variation value and range indicates that there are many events affecting the change of traffic flow during this period. In the last period, the anomaly area has been sightly away from the center, showing a state of divergence from the center to the periphery. With the events in table 2, the data apparently shows that in period 3, the traffic related events and alerts are the least among 4 periods, but the Covid-19 related news and reports are the most. All these demonstrate that the travels of people are the least in this period, this conclusion is confirmed to our detection results. Besides, in period 1, the traffic events are the most without the Covid-19 reports. So, the people's travels are a normal and frequent situation, which are the same to history. Therefore, there are not many anomaly vertices in this period. From second period, the Covid-19 reports start appearing guadually, the traffic flow is indirectly affected by the gradually released containment policies and the gradually increasing number of Covid-19 infected cases, and the overall trend is decreasing.

Our method efficiently locates the specific vertices with anomaly changes of different levels. The significance of our method is that it can quikly locate the problem area in a more sensitive period and carry out relevant downstream tasks analysis. For example, the summary of historical traffic conditions is convenient for setting and alerting new traffic facilities and policies in the area where key vertices are located, and changing according to the time period.

7 Conclusion

In this paper, basing on the appropriate temporal splitting, a novel framework called GMDS is proposed to solve the problem of temporal variant graph structure capturing. Our framework consists 4 parts, data preprocessing, DWT based splitting, ARIMA based adjacency matrix generation and graph wavelet transform based global detection. In order to demonstrate the rationality and effectiveness of GMDS, we utilize our method in traffic task that consider time-varying graph structure seriously. The experimental results show that our proposed outperforms all the baselines. And the analysis of traffic flow illustrates that our method has the ability to locate detail events, and can effectively detect anomaly events.

ACKNOWLEDGMENTS

We acknowledge the support of the following funds: National Natural Science Foundation of China under Grants 61471229 and 61901116, and Guangdong Basic and Applied Basic Research Foundation under Grants 2019A1515011950 and 2019A1515010789, and by the Key Field Projects of Colleges and Universities of Guangdong Province (No. 2019KTSCX040).

REFERENCES

Sacramental traffic archives, 2020. URL https://www.navbug.com/california.

Caltrans performance measurement system (pems). URL http://pems.dot.ca.gov/.

Caltrans near me, district 3. URL https://dot.ca.gov/caltrans-near-me.

Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. Spatio-temporal data mining: A survey of problems and methods. *ACM Comput. Surv.*, 51(4), August 2018.

Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1lK_lBtvS.

Youcef Djenouri, Asma Belhadi, Jerry Chun-Wei Lin, Djamel Djenouri, and Alberto Cano. A survey on urban traffic anomalies detection algorithms. *IEEE Access*, 7:12192–12205, 2019.

Kan Guo, Yongli Hu, Zhen Qian, Yanfeng Sun, Junbin Gao, and Baocai Yin. Dynamic graph convolution network for traffic forecasting based on latent network of laplace matrix estimation. *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020. doi: 10.1109/TITS.2020.3019497.

David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

Edward J Hannan and Jorma Rissanen. Recursive estimation of mixed autoregressive-moving average order. *Biometrika*, 69(1):81–94, 1982.

Vassilis N Ioannidis, Dimitris Berberidis, and Georgios B Giannakis. Unveiling anomalous nodes via random sampling and consensus on graphs. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5499–5503, Toronto, Ontario, Canada, June 2021. IEEE.

Peter J Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

Seyyid Emre Sofuoglu and Selin Aviyente. Low-rank on graphs plus temporally smooth sparse decomposition for anomaly detection in spatiotemporal data. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5614–5618, Toronto, Ontario, Canada, June 2021. IEEE.

Zareen Tasneem, Farissa Tafannum, and Md Mahbubur Rahman. Revised scheme for antimagic graph labeling for splitting graph and shadow graph associated with cycle. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pp. 1–5, Dhaka, Bangladesh, May 2019. IEEE.

Hoang Duy Trinh, Lorenza Giupponi, and Paolo Dini. Urban anomaly detection by processing mobile traffic traces with lstm neural networks. In *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1–8, 2019. doi: 10.1109/SAHCN.2019.8824981.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3634–3640, 2018.

Zehua Yu, Xianwei Zheng, Zhulun Yang, Bowen Lu, Xutao Li, and Maxian Fu. Interaction-temporal gcn: A hybrid deep framework for covid-19 pandemic analysis. *IEEE Open Journal of Engineering in Medicine and Biology*, 2:97–103, 2021.

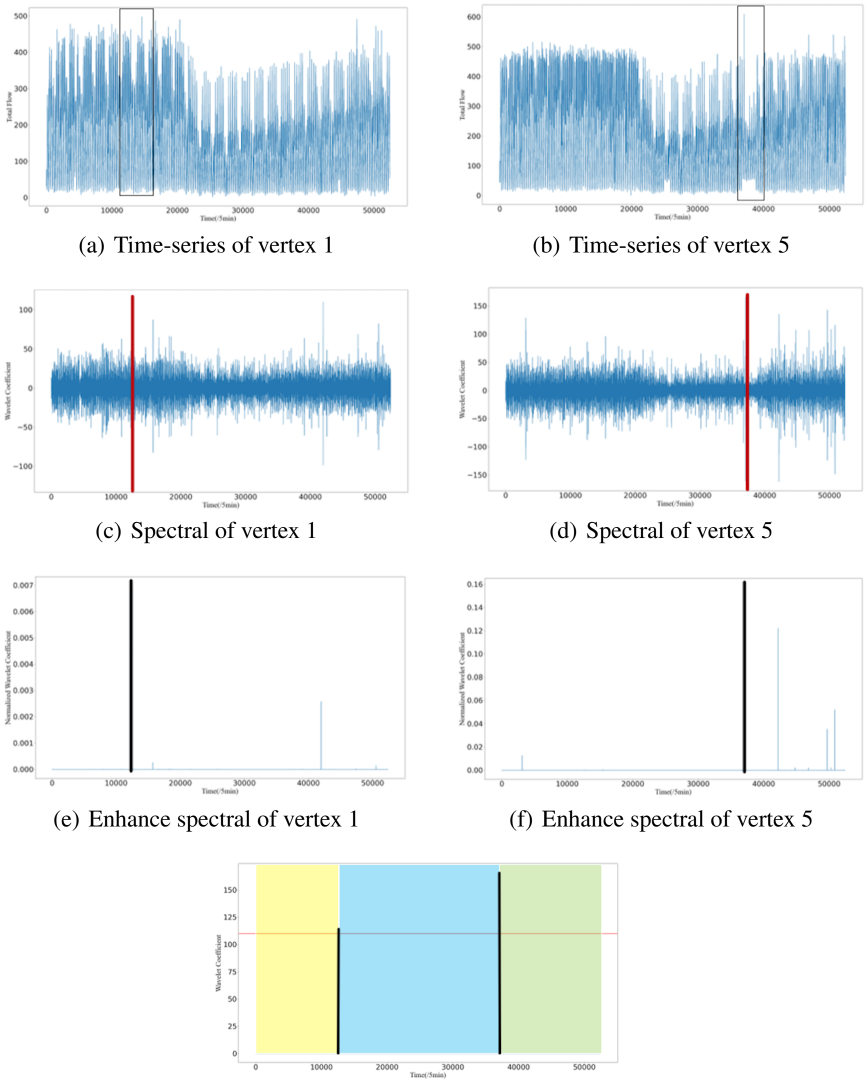
Huichu Zhang, Yu Zheng, and Yong Yu. Detecting urban anomalies using multiple spatio-temporal data sources. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):1–18, 2018.

Mingyang Zhang, Tong Li, Yue Yu, Yong Li, Pan Hui, and Yu Zheng. Urban anomaly analytics: description, detection and prediction. *IEEE Transactions on Big Data*, pp. 1–1, 2020.

Xian Wei Zheng, Yuan Yan Tang, and Jian Tao Zhou. A framework of adaptive multiscale wavelet decomposition for signals on undirected graphs. *IEEE Transactions on Signal Processing*, 67(7): 1696–1711, 2019.

Yu Zheng, Huichu Zhang, and Yong Yu. Detecting collective anomalies from multiple spatio-temporal datasets across different domains. In *Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems*, pp. 1–10, 2015.

A Appendix



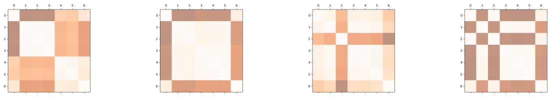
(a) Time-series of vertex 1 (b) Time-series of vertex 5

(c) Spectral of vertex 1 (d) Spectral of vertex 5

(e) Enhance spectral of vertex 1 (f) Enhance spectral of vertex 5

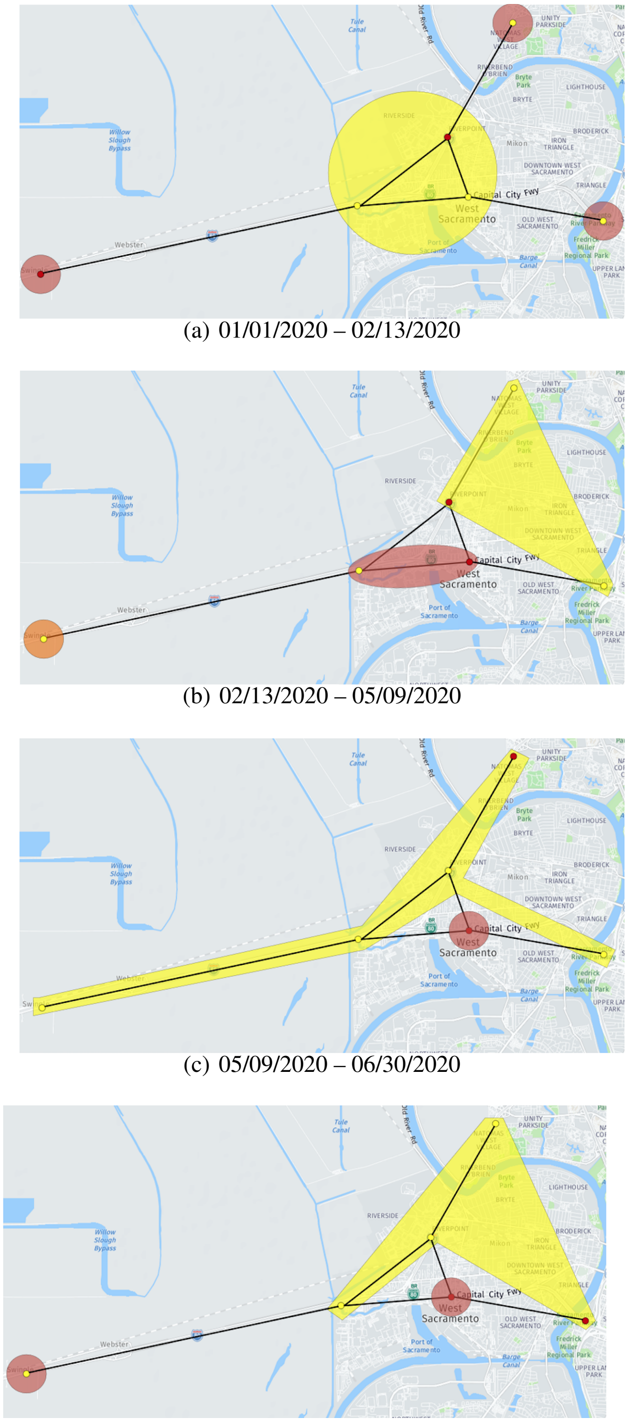
(g) Splitting on timeline

Figure 2: Time-series and spectral of voting vertices, splitting schematic diagram in PeMSD3 experiment.



(a) Period 1 (b) Period 2 (c) Period 3 (d) ALL

Figure 3: Visualization results of adjacency matrix in PeMSD3 experiment.



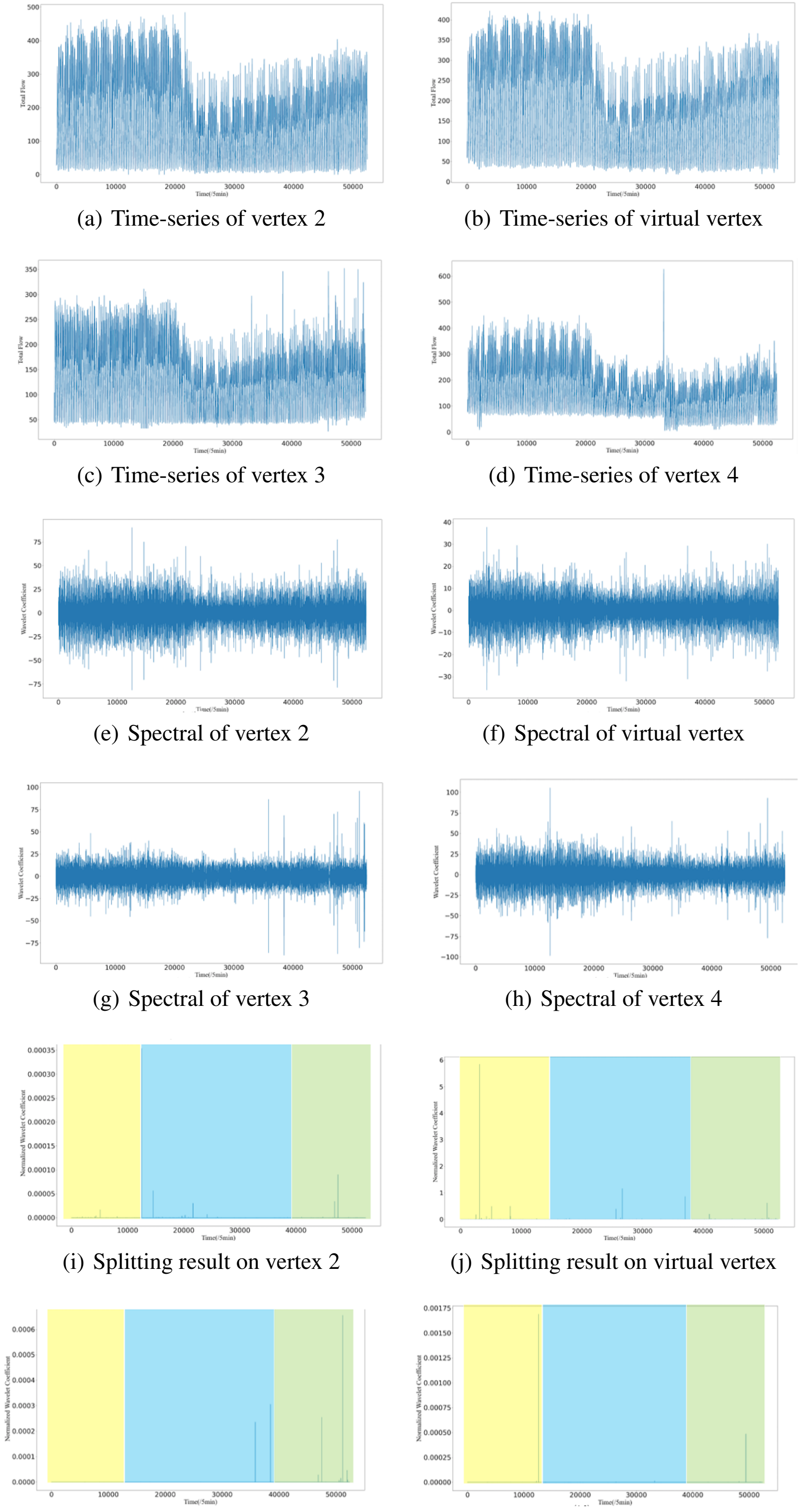
(a) 01/01/2020 – 02/13/2020

(b) 02/13/2020 – 05/09/2020

(c) 05/09/2020 – 06/30/2020

(d) 01/01/2020 – 06/30/2020

Figure 4: The results of 6 vertices verification experiment show that the yellow represents the minimum fluctuation of the sequence, and the red represents the maximum fluctuation of the sequence. (The translucent areas in figure are all diagram without classification.)



(a) Time-series of vertex 2

(b) Time-series of virtual vertex

(c) Time-series of vertex 3

(d) Time-series of vertex 4

(e) Spectral of vertex 2

(f) Spectral of virtual vertex

(g) Spectral of vertex 3

(h) Spectral of vertex 4

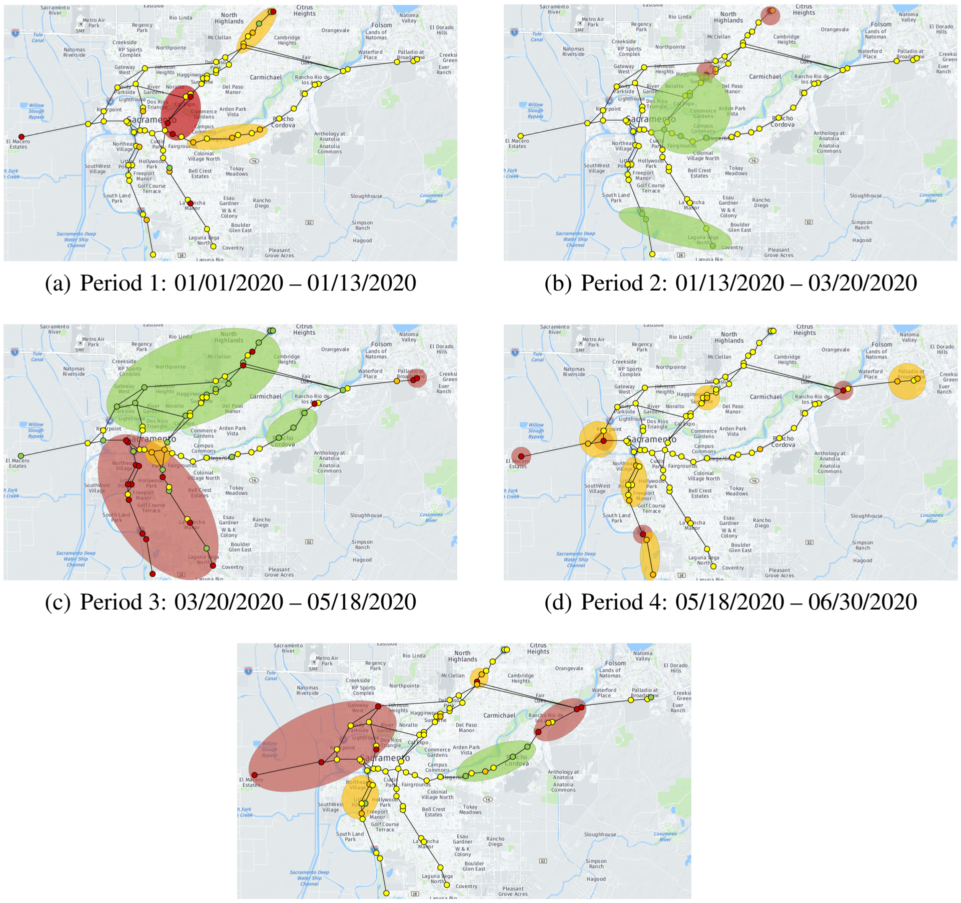
(i) Splitting result on vertex 2

(j) Splitting result on virtual vertex

(k) Splitting result on vertex 3

(l) Splitting result on vertex 4

Figure 5: Time-series, spectral and splitting result of vertex 2, 3, 4 and virtual vertex in PeMSD3 experiment.



(a) Period 1: 01/01/2020 – 01/13/2020

(b) Period 2: 01/13/2020 – 03/20/2020

(c) Period 3: 03/20/2020 – 05/18/2020

(d) Period 4: 05/18/2020 – 06/30/2020

(e) ALL: 01/01/2020 – 06/30/2020

Figure 6: The results of 70 vertices verification experiment show that the yellow represents the minimum fluctuation of the sequence, and the red represents the maximum fluctuation of the sequence. (The ellipses in figure are all diagram without classification.)