Systematic clustering method to identify and characterise spatiotemporal congestion on freeway corridors

Article *in* IET Intelligent Transport Systems · September 2018

CITATIONS
2

READS
400

5 authors, including:

Jishun Ou
Yangzhou University (China)
8 PUBLICATIONS **33** CITATIONS
SEE PROFILE

Shu Yang
University of South Florida
15 PUBLICATIONS **86** CITATIONS
SEE PROFILE

Yao-Jan Wu
The University of Arizona
66 PUBLICATIONS **796** CITATIONS
SEE PROFILE

Jingxin Xia
Southeast University (China)
51 PUBLICATIONS **493** CITATIONS
SEE PROFILE

IET Intelligent Transport Systems

Research Article

IET Journals
The Institution of Engineering and Technology

Systematic clustering method to identify and characterise spatiotemporal congestion on freeway corridors

ISSN 1751-956X
Received on 2nd November 2017
Revised 30th March 2018
Accepted on 17th April 2018
E-First on 21st May 2018
doi: 10.1049/iet-its.2017.0355
www.ietdl.org

Jishun Ou[1], Shu Yang[2], Yao-Jan Wu[3], Chengchuan An[1], Jingxin Xia[1] ✉

[1]*Intelligent Transportation System Research Center, Southeast University, No. 2, Southeast University Road, Jiangning District, Nanjing City, Jiangsu Province, 211189, People's Republic of China*

[2]*Center for Urban Transportation Research, University of South Florida, 4202 E. Fowler Ave., CUT 100, Tampa, Florida, 33620, USA*

[3]*Department of Civil Engineering and Engineering Mechanics, The University of Arizona, 1209 E. 2nd St., Tucson, Arizona, 85721, USA*

✉ *E-mail: xiajingxin@seu.edu.cn*

Abstract: Many analytical procedures, technical methods, and tools have been developed to facilitate manual inspection of traffic congestion and support the decision-making process for traffic authorities. However, lacking an automatic mechanism, it would be a time-consuming and labour-intensive process for day-to-day and location-by-location analyses. This study presents a method based on a three-stage framework that is capable of automatically identifying and characterising spatiotemporal congested areas (STCAs) by parsing, extracting, analysing and quantifying the knowledge contained in traffic heatmaps. The key components of the proposed method are two unsupervised clustering procedures: (i) a mini-batch *k*-means clustering algorithm to separate the congested and non-congested areas and (ii) a graph-theory-based clustering algorithm to distinguish between different STCAs. Twenty weekdays of dual loop detector data collected from a 26-mile stretch of Interstate 10 in Phoenix, Arizona was analysed for the case study. The new method identified and quantified 102 STCAs without the need for human intervention. Based on 14 traffic measures calculated for each STCA, 19 active bottlenecks along the study corridor were identified. Top-ranked bottlenecks identified in this study were consistent with those reported in previous studies but were produced with less effort, demonstrating the new method's potential utility for traffic congestion management systems.

1 Introduction

Traffic congestion remains a serious concern and an extremely challenging issue in urban areas all over the world. As the American Highway Users Alliance [1] notes, the time lost due to traffic congestion in the most congested 30 metro-areas costs the United States up to $2.4 billion annually. Much of this congestion is due to bottlenecks in the road systems that impede the free flow of traffic [2, 3]. Mitigation or elimination of these traffic bottlenecks could thus significantly reduce traffic congestion [4]. A number of well-designed algorithms and tools [5–10] have been developed to identify and characterise traffic congestion and the associated bottlenecks. Among these, traffic speed heatmaps have received extensive attention from researchers and practitioners because they provide an overall picture of how traffic states evolve to facilitate decision making. Fig. 1 shows an example of a heatmap with dimensions of $m \times n$, presenting traffic congestion on a spatiotemporal plane. The four major congested areas in the figure, or spatiotemporal congested areas (STCAs), may be caused by a single active bottleneck or an active bottleneck associated with several deactivated bottlenecks. An active bottleneck is defined as 'a point upstream of which there is a queue and downstream of which there is freely-flowing traffic' [11]. Bottlenecks usually arise due to poor roadway design, sharp curves, incidents, or traffic merging and waving. Active bottlenecks are marked with a pair of mileposts denoting the locations of the closest upstream and downstream detector stations. A deactivated bottleneck may become so due to the spill over from a bottleneck further downstream when the two queues merge together [12]. Several useful measures can be estimated from a speed heatmap, including the onset and clearance times of active bottlenecks, the approximate locations of active bottlenecks, and the spatiotemporal extent of each STCA.

Although transportation researchers and practitioners often diagnose (or analyse) traffic congestion and bottlenecks by visually inspecting traffic heatmaps, gathering specific spatiotemporal information as well as the corresponding performance measures, such as delay, for traffic congestion and the associated bottlenecks can be challenging; it is also time-consuming and labour-intensive to conduct day-to-day and location-by-location heatmap analyses. This study proposes a systematic clustering method to identify and characterise traffic congestion and the associated bottlenecks that can be effectively and efficiently applied to practical traffic operations to facilitate decision making.

2 Related work

Various methods have been developed to identify and characterise traffic congestion and the associated bottlenecks in recent years. Generally, these methods can be categorised into two groups: segment-based and corridor-based methods. As the names suggest, segment-based methods diagnose traffic congestion and the associated bottlenecks by detecting and tracking upstream and downstream traffic states at the segment level, while corridor-based methods attempt to achieve the same goal by analysing traffic state evolution at the corridor level.

A classic segment-based method uses oblique cumulative curves [5, 11], plotting cumulative curves for vehicle count or occupancy along an oblique time axis rather than a conventional orthogonal axis to amplify temporal changes in traffic states. Congestion is determined by carefully inspecting changes in the curves, and the activation time and location of the associated bottleneck is identified by locating the point where a sudden decrease in the slope of a cumulative curve occurs. The oblique cumulative curve is a powerful way to analyse congestion and characterise bottlenecks, but it is vulnerable to accumulated errors from biased counts [13]. Another major drawback is that it is challenging to automate because it requires manual inspection and adjustment. To overcome the latter drawback, several automatic methods have been proposed. Chen *et al.* [6] presented a systematic method to detect congestion based on a predefined cut-off speed, identifying and quantifying bottlenecks according to the speed differences between consecutive detectors; Zhang and Levinson [7] implemented a similar process using occupancy

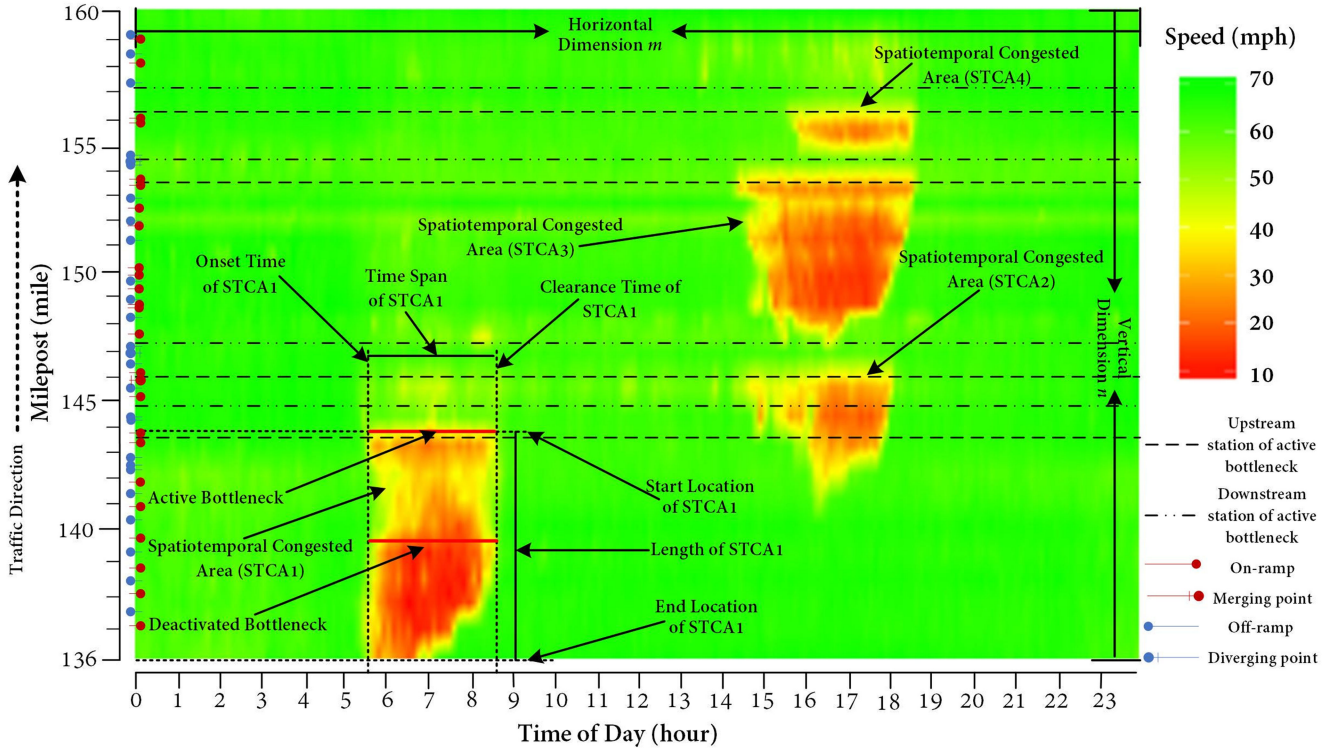


Fig. 1 *Example of a speed heatmap showing STCAs*

differentials. Das and Levinson [14] diagnosed traffic congestion with fundamental diagrams in which four traffic phases were defined and partitioned and the associated bottlenecks identified through statistical analysis. The common drawback of all these automatic methods is that not only are they subject to noise in the traffic data, thus leading to unreliable diagnoses [9, 10], but the predefined thresholds tend to be subjective and site-specific. In an attempt to address the uncertainty introduced by unreliable measurements, fuzzy logic rules were incorporated into the bottleneck identification process by Liu and Fei [9]. However, their model still required predefined thresholds. Zheng *et al.* [10] utilised wavelet transforms to analyse important features related to bottleneck activation and identify bottleneck locations. Although wavelet-based methods are capable of smoothing out negative impacts caused by traffic noise and finding the exact times of congestion onset and clearance at each station, as with most segment-based methods it remains difficult to distinguish and quantify spatiotemporal congestion caused by multiple bottlenecks in the same corridor.

For corridor-based methods, one of the most commonly used approaches is to use speed contour maps, which have been successfully applied in several advanced intelligent transportation systems (ITS), including the Caltrans Performance Measurement System [15] and Portland Oregon Regional Transportation Archive Listing [16]. Speed contour maps present traffic conditions using a time-space plane, where contour lines are used to represent a range of speeds. Although such maps can provide good spatial coverage and a useful overall picture of traffic state evolution [10], as a graphical tool they are usually best suited to preliminary analyses. Ban *et al.* [8] proposed an automatic congestion and bottleneck identification method based on a binary speed contour map that uses several days of speed percentile data to identify congestion and bottlenecks. Their method is appropriate for urban congested freeways and computationally efficient for bottleneck calibrations in traffic simulations, but may not be able to distinguish and characterise multiple bottlenecks along a corridor. The speed heatmap, which is similar to the speed contour map, is another popular visualisation tool for examining and tracking congestion and bottlenecks on a time-space plane [17, 18], also requires a manual inspection to provide an effective diagnosis as well as suffering from a limited capacity to provide accurate spatiotemporal congestion information.

Once identified, congestion and bottlenecks are characterised by a series of measures, including activation time [5, 6, 10, 11], queue discharge flows [7], duration [4, 8], spatial extent [4], intensity [4], queue length [1, 8], severity [9], delay [1, 4, 6], and frequency [6]. Despite the fact that these traffic measures can be defined and calculated in different ways, the ultimate goal is to quantify the impact caused by the congestion and bottlenecks identified and help decision makers prioritise congestion mitigation strategies.

To the best of the authors' knowledge, few researchers have sought to identify and characterise traffic congestion using traffic heatmaps in an automatic and systematic way. To explore their strengths and address the shortcomings of existing methods, this study aims to parse, extract, analyse and quantify the knowledge contained in traffic heatmaps. This paper is organised as follows. The new three-stage methodology framework is introduced in the next section, along with details of the STCA identification, refinement and characterisation processes utilised, after which the results of an empirical study to demonstrate the feasibility and effectiveness of the proposed method will be presented. The paper will conclude by summarising the study's findings, discussing its contributions and making recommendations for future work.

3 Methodology

3.1 Framework

Fig. 2 presents the three-stage framework of the proposed method. The three stages consist of (i) 'STCA identification' where STCAs are identified in speed heatmaps by using two unsupervised clustering procedures; (ii) 'STCA refinement' where a post-processing procedure is conducted to refine the identified STCAs; and (iii) 'STCA characterisation' where the refined STCAs are characterised by a series of measures after finding their contours.

3.2 STCA identification

The first stage of the proposed method automatically identifies the STCAs in the speed heatmaps. This stage involves two steps: (i) distinguish the STCAs from the non-congested areas (non-STCAs) by using the mini-batch *k*-means algorithm; (ii) assign a corresponding label to each STCA by using the proposed graph-theory-based clustering (GTBC) algorithm.

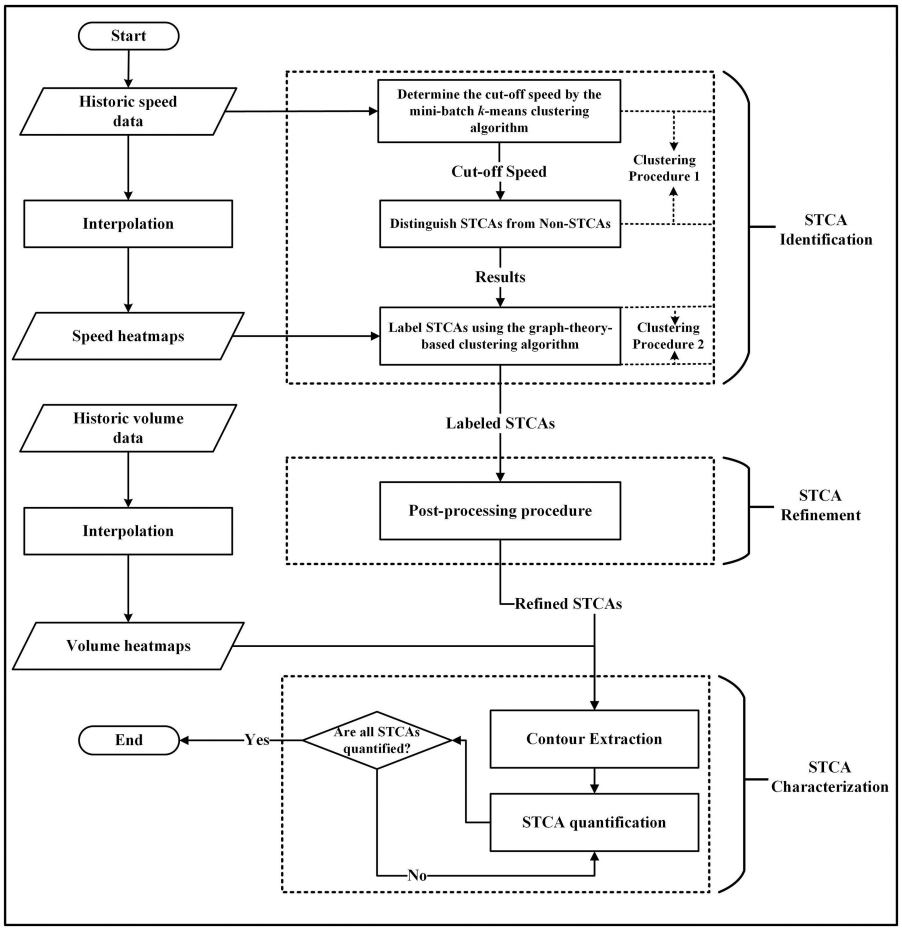


Fig. 2 *Methodology framework*

3.2.1 Distinguishing STCAs from non-STCAs: To distinguish STCAs from non-STCAs, a common practice is to set a predefined cut-off speed threshold based on the experience of decision-makers [6]. As indicated by Wieczorek *et al.* [19], the user's optimal choice of the cut-off speed relies on variable factors in terms of geography, traffic patterns, and driver behaviours in a certain region. To automatically determine an appropriate cut-off speed for different cases, the mini-batch *k*-means algorithm [20] is utilised in the new method to cluster the historic speed data. The mini-batch *k*-means algorithm uses per-centre learning rates and a stochastic gradient descent strategy to speed up convergence of the clustering algorithm, enabling high-quality solutions to identify several orders of magnitude faster than the conventional batch *k*-means method [21]. Fig. 3 shows the pseudo-code and the details of the mini-batch *k*-means algorithm.

To distinguish STCAs from non-STCAs in the speed heatmaps, an appropriate cut-off speed has first to be established as the threshold. The final cut-off speed is estimated as follows:

$$v_{\text{cut}-\text{off}} = \frac{\max\{v_{\text{STCA}}\} + \min\{v_{\text{non}-\text{STCA}}\}}{2}, \tag{1}$$

where v_{STCA} are the speed values in the STCA cluster, and $v_{\text{non}-\text{STCA}}$ are the speed values in the non-STCA cluster. The final output of this step is a binary heatmap where class label 1 represents non-STCAs and class label 2 represents STCAs.

3.2.2 Labelling STCAs by GTBC: In general, more than one STCA may be presented in a given corridor so it is necessary to make a clear distinction between these different STCAs. An intuitive approach is to use another clustering algorithm to partition these STCAs. Although a wide spectrum of clustering algorithms have been proposed, most depend on prior assumptions or extra validity indices to determine the optimal number of clusters [22]. A truly automated clustering procedure does not require the number of clusters to be pre-defined, so to solve this problem a novel clustering algorithm based on graph theory is proposed in this study. This algorithm automatically groups and labels the different STCAs without the need for either prior assumptions or extra validity indices. The proposed algorithm was inspired by the visual inspection process of humans: an STCA in a given speed heatmap can always be distinguished from other STCAs by determining the connectivity of the data points within the STCA, which means that any two STCAs in a speed heatmap should be treated differently if they are not closely connected with each other. The clustering process can thus be deemed a problem that is solved by finding connected components in an undirected graph. Fig. 4 shows the pseudo-code and the details of the proposed GTBC algorithm.

The input data set of the proposed algorithm is composed of a series of data points, the class label of which is 1. Each data point is described by two variables: time and space. To check the connectivity between the data points in STCAs, the connectivity distance must first be estimated. Intuitively, two points in a speed heatmap are connective if they are contiguous in either or both the horizontal direction or the vertical direction. The connectivity distance is estimated as the maximum distance between the minimum neighbour distance in the horizontal direction and the minimum neighbour distance in the vertical direction; the adjacency matrix can thus be generated according to the connectivity distance. The clustering procedure can now be translated into a problem to be solved by searching the connected components in an undirected graph represented by the adjacency matrix. A connected component is defined as a subgraph where a path exists between any two vertices in a graph. For an undirected graph $G = \langle V, E \rangle$, where V is the set of vertices and E is the set of edges, the search process for connected components is as follows:

For each vertex $v \in V$ in G

(a) If v has no owner, it belongs to an undiscovered connected component. Conduct a depth-first search [23] starting from v and mark all the vertices as being owned by v.

(b) If v has an owner, it is part of a connected component already discovered. Ignore v and move on to the next vertex.
(c) If all vertices have been traversed, exit the search process.

Once all connected components have been searched, each cluster corresponding to data points in the same connected component is assigned to an STCA. This means that the number of clusters (connected components) is the same as the number of STCAs. The strength of the GTBC algorithm lies in its ability to automatically determine the number of STCAs in a given speed heatmap without the need for either prior assumptions or extra validity indices. The time complexity of the search process is $(|V| + |E|)$, where $|V|$ is the number of vertices and $|E|$ is the number of edges of the created graph. Given the resulting reasonable time complexity, this clustering procedure can be easily implemented in practical applications.

3.3 STCA refinement

Traffic flow data often contains noise, leading to unreasonable or unreliable identification results [6, 19]. A two-step post-processing procedure was thus developed to refine the identified STCAs.

- Step 1: Remove data points from the STCAs according to the following two rules:
  For each data point $p \in P_{\text{STCA}}$, where P_{STCA} is the set of data points in STCA,
  a. If count(P_{STCA}) $< \alpha$, remove all data points in P_{STCA}.
  b. Find the data points located in the furthest downstream segment P_{down}. If the duration of P_{down} is less than β minutes, remove all points in P_{down}. Repeat until the duration is equal to or great than β minutes.
- Step 2: Fill the gaps or 'holes' within each STCA assuming each STCA is a continuous congestion entity.

In Step 1, Rule (a) smooths out small isolated STCAs caused by moving jams, measurement errors or temporary fluctuations in the traffic flow as these are not the focus of this study. Here, α was set as 30 based on the interpolation resolution (200×200 in the later experimental settings). The setting of the value of α will not influence the robustness of the proposed methodology since those unfiltered small STCAs are still able to be characterised in the STCA characterisation stage. Rule (b) was applied because STCAs are usually caused by a sustained active bottleneck lasting at least 25 min [6]. Thus, β was set as 25 in this study.

3.4 STCA characterisation

After STCA identification and refinement, the last stage of the proposed framework is to characterise the identified STCAs. This stage consists of two steps:

- Step 1: contour extraction
  The contour of an STCA is composed of a series of outermost data points, allowing several traffic parameters to be measured. Given an empty set of contour points P_{contour}, for each point $p \in P_{\text{STCA}}$.
  a. If $p \in P_{\text{bound}}$, where P_{bound} is the set of boundary points of the heatmap, add p to P_{contour}.
  b. Find the nearest neighbours of p in the horizontal and vertical directions. Denote the set of neighbour points as P_{neig}. If the cluster label of any point in P_{neig} is different from the cluster label of p, then add p to P_{contour}.
- Step 2: Quantification

A number of relevant traffic measures can be extracted after contour extraction. As the delay measure requires the corresponding traffic volume for each data point, a flow heatmap is generated using a linear interpolation technique [24]. Given an STCA with a contour point set $P_{\text{contour}} \in P_{\text{STCA}}$, each data point $p \in P_{\text{STCA}}$ carries four kinds of information, namely temporal

Algorithm 1: Mini-batch *k*-means Clustering

Input: data set $D = \{x_1, x_2, \dots, x_n\}$; cluster number k; mini-batch size b; iterations t

Output: clusters $C = \{C_1, C_2, \dots, C_k\}$

Process:

1: Initialize k cluster centers $\mu = \{\mu_{C_1}, \mu_{C_2}, \dots, \mu_{C_k}\}$ with k samples randomly picked from D

2: $C_i \leftarrow \emptyset\ (1 \le i \le k)$ # *initialize clusters*

3: $N_{C_i} \leftarrow 0\ (1 \le i \le k)$ # *initialize sample number for each cluster*

4: **for** $j = 1, 2, \dots, t$ **do**

5: $M \leftarrow \{x_m | 1 \le m \le b\}$ # *M is the batch data set, and x_m is the sample randomly picked from D*

6: **for** $m = 1, 2, \dots, b$ **do** # *step 6 ~ step 8 are to catch cluster center for each sample in the batch set*

7: $\mu_{C_i}(x_m) \leftarrow \frac{1}{|C_i|}\sum_{x_m \in C_i} x_m\ (x_m \in M)$ # *$\mu(x_m)$ is the cached cluster center nearest to data sample x_m*

8: **end for**

9: **for** $m = 1, 2, \dots, b$ **do** # *step 9 ~ step 14 are to update the cluster centers with each batch set*

10: $\mu_{C_i} \leftarrow \mu_{C_i}(x_m)$ # *get the cached center for x_m*

11: $N_{C_i} \leftarrow N_{C_i} + 1$ # *update sample number for each cluster center*

12: $\eta \leftarrow 1/N_{C_i}$ # *calculate learning rate for each cluster center*

13: $\mu_{C_i} \leftarrow (1 - \eta)\mu_{C_i} + \eta x_m$ # *take gradient step to update cluster center*

14: **end for**

15: **end for**

Fig. 3 *Pseudo-code of the mini-batch k-means algorithm*

Algorithm 2: Graph-theory-based Clustering (GTBC)

Input: data set $D = \{x_1, x_2, \dots, x_n\}$

Output: clusters $C = \{C_1, C_2, \dots, C_k\}$; the number of clusters k

Process:

$C \leftarrow \emptyset$ # *initialize clusters*

$k \leftarrow 0$ # *initialize the number of clusters*

for $i = 1, 2, \dots, n$ **do** # *step 3 ~ step 5 are to scale data samples with mean and standard deviation*

  $x_i \leftarrow \big(x_i - mean(x_i)\big)/sdv(x_i)$

end for

for $i = 1, 2, \dots, n$ **do** # *step 6 ~ step 11 are to calculate distance matrix*

  for $j = 1, 2, \dots, n$ **do**

    $M(i,j) \leftarrow \left\|x_i - x_j\right\|_2$ # *$M(i,j)$ is the element of the distance matrix M*

    $M(j,i) \leftarrow M(i,j)$ # *distance matrix is a symmetric matrix*

  end for

end for

$d_h \leftarrow \min_{1<j\le n} M(1,j)$ # *find the minimum neighbor distance in the horizontal direction*

$d_v \leftarrow \min_{1<i\le n} M(i,1)$ # *find the minimum neighbor distance in the vertical direction*

$d_{conn} \leftarrow max(d_h, d_v)$ # *estimate the connectivity distance*

for $i = 1, 2, \dots, n$ **do** # *step 15 ~ step 23 are to generate adjacency matrix*

  for $j = 1, 2, \dots, n$ **do**

    if $0 \le M(i,j) \le d_{conn}$ **then**

      $A(i,j) \leftarrow 1$ # *$A(i,j)$ is the element of the adjacency matrix A*

    else

      $A(i,j) \leftarrow 0$

    end if

  end for

end for

Create undirected graph $G = \langle V, E \rangle$ with the adjacency matrix A

Search the connected components $C = \{C_1, C_2, \dots, C_k\}$ in G by using the DFS algorithm

Assign the same cluster labels to data points (samples) in the same connected component

Fig. 4 *Pseudo-code of the graph-theory-based algorithm*

information t, spatial information s, speed information v, and volume information q. Assume the time interval of collected data is T_{interv} (5 min in our study), the number of data points in P_{STCA} is J, and the number of data points in P_{contour} is $I (I < J)$.

The extracted measures are defined and described in Table 1. Note that the total delay due to the STCA, D_{total}, and the delay due to the active bottleneck, D_{ab} are treated differently in our study: D_{ab} calculates the delay based solely on the data points directly associated with the active bottleneck as there may be multiple bottlenecks in some STCAs, in which case the delay due to deactivated bottlenecks is excluded from D_{total}. Chen *et al.* [6] defined the delay of an active bottleneck as the total delay caused by the contiguous group of congested segments upstream of the active bottleneck location, arguing that the total delay is attributable to the active bottlenecks. Their definition of delay is thus equivalent to D_{total} in Table 1.

4 Empirical study

4.1 Study corridor and data description

The performance of the proposed method was evaluated using data collected along a 26-mile stretch of I-10 (eastbound) that passes through the central business district of Phoenix, Arizona. Traffic flow is monitored using 29 detectors, as seen in Fig. 5*a*, and 5-min volume and speed data are collected by dual-loop detectors. Data for 20 weekdays between 2 and 27 March 2015 was used for this study. Note that data for the high-occupancy vehicle (HOV) lane along this corridor was excluded because the traffic flow characteristics in an HOV lane differ significantly from those in non-HOV lanes.

Fig. 5*b* shows the speed heatmaps for all 20 weekdays studied. At least four STCAs can be observed in the data for each day, implying the study corridor has at least four active bottlenecks. The study corridor experienced varying traffic conditions over the study period, suggesting that different STCAs must be individually analysed and quantified each day. For example, on the 6, 13, 18 19, and 27 March, traffic appeared to exhibit stronger fluctuations than on other days; traffic patterns on Fridays were also different from those on other weekdays, especially during the afternoons.

4.2 Identification of STCAs

The first stage of the new method proposed here identifies different STCAs along a given corridor using two clustering procedures, as described earlier. The first of these clustering procedures distinguishes congested areas from non-congested areas. The cut-off speed determined by the mini-batch *k*-means algorithm is 45.5 mph, which is very close to the empirical threshold of 45 mph

adopted in a previous study [25] for the same I-10 corridor. The FHWA report [4] recommended the use of a cut-off speed 28% lower than the free-flow speed, which for this stretch of the interstate is 65 mph. The cut-off speed automatically determined here is thus consistent with both of these two studies.

Once the congested areas are identified, the GTBC algorithm proposed here was used to label the extracted STCAs. Three popular clustering algorithms, batch *k*-means [21], fuzzy *c*-means [26] and density-peaks [27], were also implemented for comparison. Both of the *k*-means and fuzzy *c*-means algorithms require a predefined number of clusters to be assigned before the clustering procedure can proceed, thus extra validity measures were employed to determine the corresponding number of clusters, while the density-peaks and GTBC algorithms are able to automatically determine the optimal number of clusters by using their own mechanism.

For the *k*-means algorithm, we used the Elbow method [28] to define the proper number of clusters. The procedure of this method to select the optimal cluster number is described as follows:

- Conduct the *k*-means clustering algorithm by varying a range of values of k ($1 \le k \le K$), where K is the largest possible value of the optimal cluster number.
- For each k calculates the within-cluster sum of the square index, defined as follows:

$$\mathrm{WSS}_k = \sum_{j=1}^{k} \sum_{x_i \in C_j} (x_i - c_j)(x_i - c_j)^{\mathrm{T}}, \tag{2}$$

  where C_j is the jth cluster of the clustering result, x_i is the ith sample in cluster C_j, c_j is the centroid of cluster C_j.
- Plot the curve of the WSS according to the number of clusters k, and choose k, the WSS value of which corresponds to an obvious bend of the curve, as the optimal cluster number.

For the fuzzy *c*-means algorithm, the average Silhouette method [29] was used to determine the optimal number of clusters. This method is similar to the Elbow method while selecting the corresponding k that leads to the maximum average silhouette index, defined as follows:

$$\mathrm{AvgSil} = \frac{\sum_{i=1}^{n} \dfrac{b(i) - a(i)}{\max\{a(i); b(i)\}}}{n}, \tag{3}$$

where

$a(i) = \left(\left(\sum_{j \in \{C_r \setminus i\}} d_{ij}\right)/(n_r - 1)\right)$ is the dissimilarity of the ith sample to all other samples of cluster C_r, and $d_{ij} = \sqrt{x_i x_j^{\mathrm{T}}}$ is the dissimilarity of sample x_i to sample x_j,

n_r is the number of samples in cluster C_r,

$b(i) = \min_{s \ne r}\left\{\left(\left(\sum_{j \in C_s} d_{ij}\right)/n_s\right)\right\}$ is the minimum average dissimilarity of the ith sample to all samples of cluster C_s,

n_s is the number of samples in cluster C_s,

n is the total number of clustering samples.

The density-peaks algorithm can determine the optimal number of clusters by estimating the local density ρ and the distance δ from points of higher density, defined as follows, respectively

$$\rho(x_i) = \sum_{j=1}^{n} \gamma(d_{ij} - d_{\mathrm{c}}), \tag{4}$$

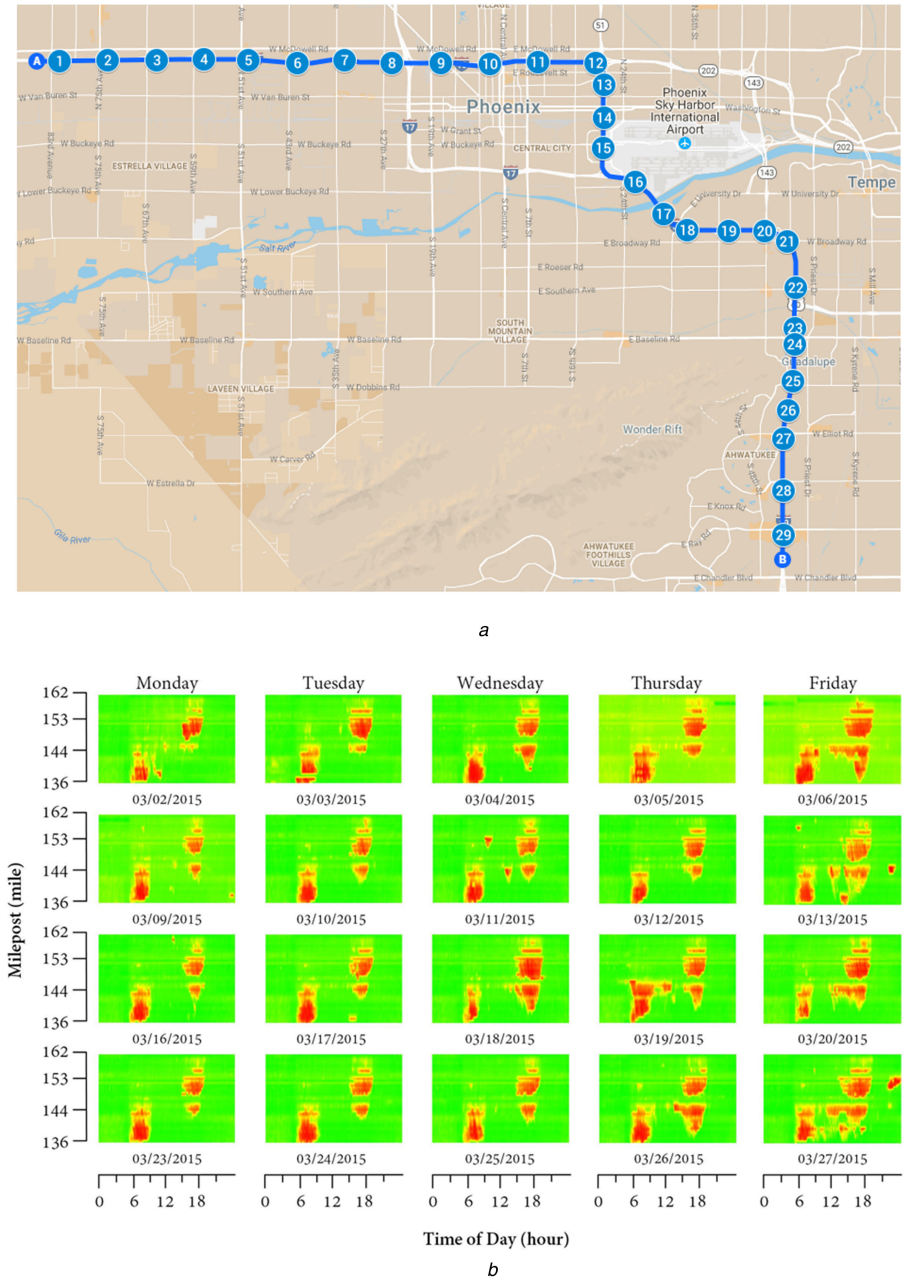
$$\gamma(d_{ij} - d_{\mathrm{c}}) = \begin{cases} 1 & \text{if} \quad d_{ij} < d_{\mathrm{c}}, \\ 0 & \text{if} \quad d_{ij} \ge d_{\mathrm{c}}, \end{cases} \tag{5}$$

$$\delta(x_i) = \min_{j:\rho(x_j) > \rho(x_i)} d_{ij}, \tag{6}$$

where d_{ij} is the distance between sample x_i and sample x_j, d_{c} is a cut-off distance, which can be selected so that the average number

Table 1 Extracted STCA measures

No.	Traffic measures	Definition
1	$T_{\mathrm{onset}} = \min_{1 \le i \le I}(t_{p_i}) \times T_{\mathrm{interv}}$, where $p_i \in P_{\mathrm{contour}}$	onset time of the STCA, min
2	$T_{\mathrm{clear}} = \max_{1 \le i \le I}(t_{p_i}) \times T_{\mathrm{interv}}$, where $p_i \in P_{\mathrm{contour}}$,	clearance time of the STCA, min
3	$T_{\mathrm{span}} = T_{\mathrm{clear}} - T_{\mathrm{onset}}$	time span of the STCA, min
4	$S_{\mathrm{start}} = \min_{1 \le i \le I}(s_{p_i})$, where $p_i \in P_{\mathrm{contour}}$	start location of the STCA, miles
5	$S_{\mathrm{end}} = \max_{1 \le i \le I}(s_{p_i})$, where $p_i \in P_{\mathrm{contour}}$	end location of the STCA, miles
6	$S_{\mathrm{length}} = S_{\mathrm{end}} - S_{\mathrm{start}}$	length of the STCA, miles
7	$N_{\mathrm{cov\_stn}}$	number of covered stations
8	$N_{\mathrm{cov\_seg}}$	number of covered segments
9	$D_{\mathrm{total}} = \sum_{j=1}^{J} D_{p_j}$, where $p_j \in P_{\mathrm{stca}}$, and D_{p_j} is the delay at the jth point, $D_{p_j} = \left(q_{p_j} \times 60/T_{\mathrm{interv}}\right) \times \left(\dfrac{\max_{1 \le k \le m*n}(s_{p_k}) - \min_{1 \le k \le m*n}(s_{p_k})}{n-1}\right) \times \left(1/v_{p_j} - 1/v_f\right)$, q_{p_j} is the volume (veh-5 min) at the jth point in the STCA, s_{p_k} is the spatial location of the kth point in heatmap, v_{p_j} is the speed (mph) at the jth point in the STCA, and v_f is the free-flow speed (taken to be 65 mph here)	total delay associated with the STCA, veh-h
10	L_{ab} is defined as the furthest downstream segment of the STCA, represented by a station-pair, i.e. $\langle s_{\mathrm{start\_stn}}, s_{\mathrm{end\_stn}} \rangle$	location of the active bottleneck in the STCA
11	$T_{\mathrm{onset\_}ab} = \min_{p_i \in P_{ab}}(t_{p_i}) \times T_{\mathrm{interv}}$, where P_{ab} is the set of points in the segment containing the active bottleneck	onset time of the active bottleneck in the STCA, min
12	$T_{\mathrm{clear\_}ab} = \max_{p_i \in P_{ab}}(t_{p_i}) \times T_{\mathrm{interv}}$, where P_{ab} is the set of points in the segment containing the active bottleneck	clearance time of the active bottleneck in the STCA, min
13	$T_{\mathrm{dur\_}ab} = T_{\mathrm{clear\_}ab} - T_{\mathrm{onset\_}ab}$	duration of the active bottleneck in the STCA, min
14	$D_{ab} = \sum_{p_i \in P_{ab}} D_{p_i}$, where D_{p_i} is the delay of the ith point in the area directly associated with the active bottleneck	delay associated with the active bottleneck in the STCA, veh-h



a

b

Fig. 5 *Study corridor*

(a) I-10 study corridor (eastbound) in Phoenix, AZ, ***(b)*** Corresponding speed heatmaps for study period

of neighbours is around 1 to 2% of the total number of samples, n is the total number of samples.

The speed data for 12 March 2015 were extracted to generate the speed heatmap for this comparison as the traffic conditions on this particular day were relatively uncomplicated and it thus serves as a useful baseline for comparing the performance of the different clustering algorithms. After carefully calibrating the parameters, the number of clusters and the clustering partitions was determined as shown in Fig. 6*a*. Each colour in the figure represents a specific cluster identified by a relevant clustering algorithm. As illustrated by a number of existing studies [4, 6, 8, 19, 25], each STCA associated with a bottleneck should be a contiguous entity on a spatiotemporal plane. Therefore, it is reasonable for a clustering algorithm to label a specific STCA with a single colour rather than two or more colours. That is, the clustering results are determined to be inaccurate or unreasonable if one STCA is labelled with two or more colours or more than one STCA are labelled with the same colour. Moreover, as opposed to supervised classification problems, there is often no ground-truth data to test the accuracy of unsupervised clustering algorithms [30]. Therefore, this study evaluates the accuracy of the clustering algorithms by visually inspecting whether each STCA is appropriately labelled with a single specific colour.

As can be seen from the figure, the optimal number of clusters identified with the extra validity measures for the k-means and fuzzy c-means algorithms are five and six, respectively. The optimal ρ and δ in the density-peaks algorithm were determined to be 40 and 0.15. Accordingly, the most suitable cluster number was identified as five. For the GTBC algorithm, it identified a total of nine clusters, corresponding to four major STCAs and five small isolated STCAs (see the specific location of each STCA at the bottom right of Fig. 6*a*), which can be confirmed by visually inspecting the corresponding speed heatmap in Fig. 5*b*. It can also be observed that all three comparative algorithms found it hard to identify the proper cluster number as well as the clustering partitions, due to the fact that some STCAs (e.g. STCA2 and STCA3 clustered with the k-means algorithm; STCA3 clustered with the fuzzy c-means algorithm; STCA1 clustered with the density-peaks algorithm) were labelled with two or more colours, while some other STCAs (e.g. STCA4 and STCA6 clustered with the k-means algorithm; STCA2, STCA4, and STCA6 clustered with the fuzzy c-means algorithm; STCA2, STCA3, STCA4 and STCA6 clustered with the density-peaks algorithm) were assigned

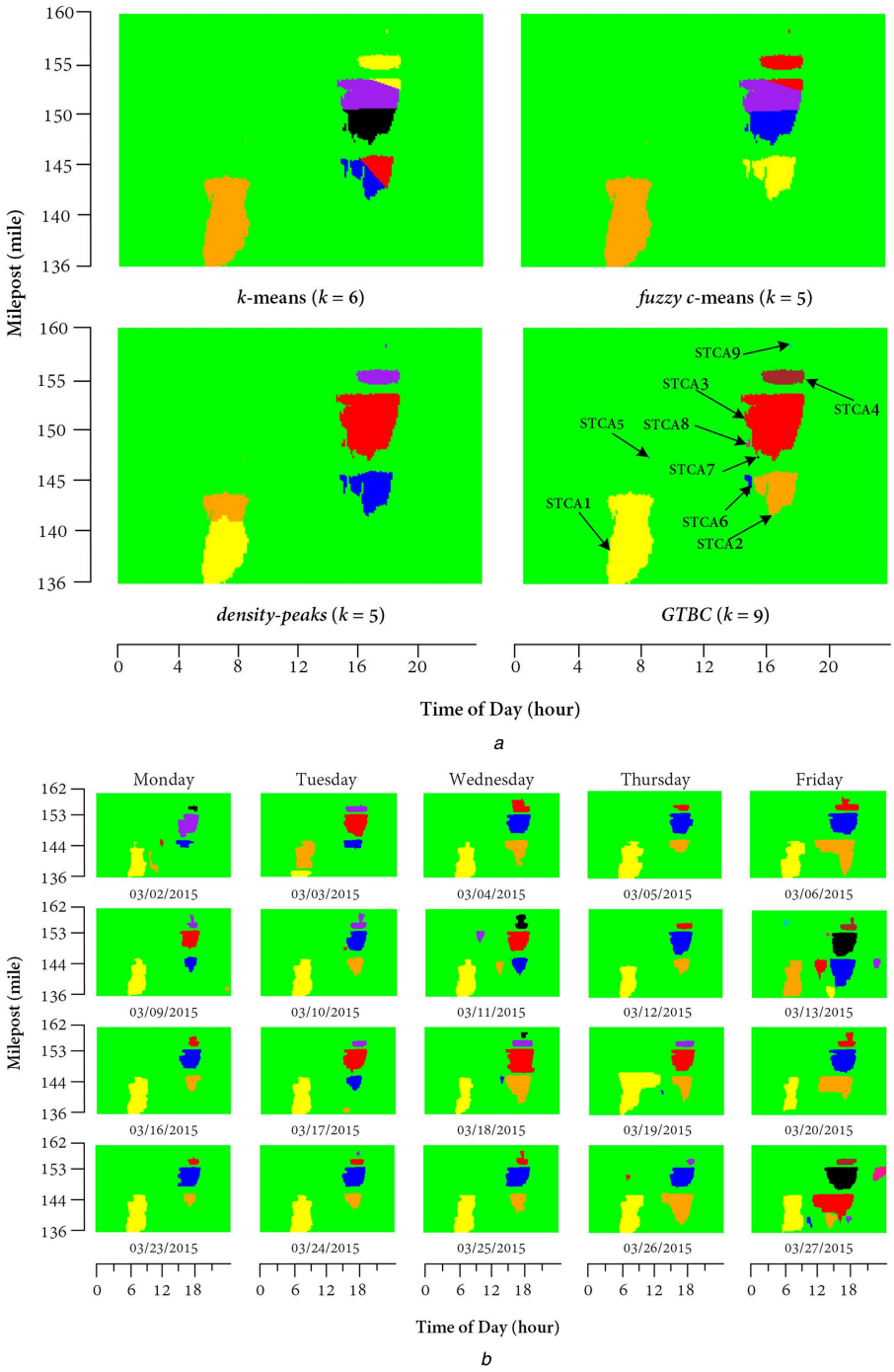


Fig. 6 *Results of STCA identification*

(a) Comparisons of clustering algorithms (*k*-means, fuzzy *c*-means, density-peaks and GTBC), ***(b)*** Identified STCAs

the same colour with more than one STCA. In contrast, the proposed GTBC algorithm identified all of the STCAs accurately.

Fig. 6*b* shows the clustering results for the entire study period obtained using the GTBC algorithm. Each cluster was assigned a specific colour representing an individual STCA on that day. Small isolated STCAs were excluded because these STCAs are often caused by moving jams, measurement errors or temporary fluctuations in traffic flow. Note also that the number and shape of the STCAs vary over different days, implying the clustering procedure should be able to automatically determine the number of clusters. Comparing Fig. 5*b* with Fig. 6*b*, the GTBC algorithm has accurately determined the optimal cluster number and reasonable cluster partitions.

4.3 Stability of GTBC

One basic philosophy in clustering is that the clustering results achieved from a clustering algorithm should be stable. That is, if applied to different data sets from the same underlying model or of the same data generating process, a clustering algorithm should obtain similar results [30]. To verify the stability of the GTBC algorithm, we tested the clustering procedure on several different data sets generated with varying interpolation resolutions, i.e. the resolution r is defined as 50×50, 80×80, 100×100, 140×140, 170×170, and 200×200, respectively. The final clustering results are shown in Fig. 7. It can be observed that the number of STCAs increases with the resolution. This is due to the fact that the higher the resolution value was set as, the more the number of congested points was interpolated in the speed heatmaps. Not surprisingly, the GTBC algorithm is still able to accurately identify all STCAs as well as their correct number, no matter what the resolution was used, implying it belongs to a kind of stable clustering algorithms.

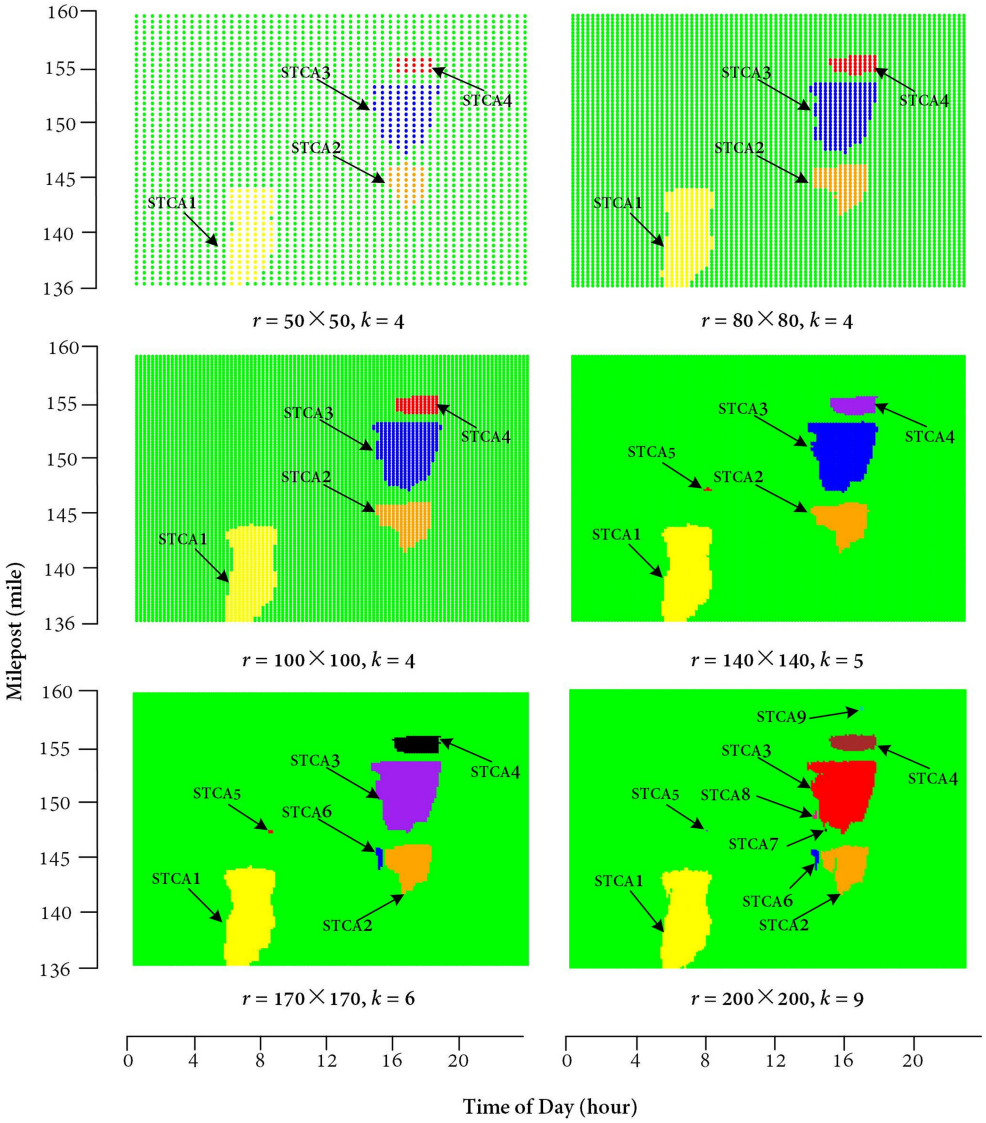


Fig. 7 *Clustering results of GTBC with varying resolutions*

4.4 Characterisation of STCAs

The 14 measures listed in Table 1 can be extracted by the proposed method to characterise each STCA identified for each study day. The first five indices are obtained by analysing the contours of the STCA, as shown in Fig. 8*a*. It is easy to see that the contour of each STCA can be accurately extracted. In addition, the calculated delays for each day are shown in Fig. 8*b*, which are unevenly distributed over the different STCAs. It is worth noting that two recurrent congested areas can be observed by visually inspecting this figure. Perhaps unsurprisingly, the first of these occurs during the morning peak hours (5:30–9:00) and the second during the evening peak hours (15:00–18:30).

Table 2 shows the quantitative results for the first 5 weekdays during the study period. By analysing the S_{length}, $N_{\text{cov\_stn}}$, $N_{\text{cov\_seg}}$, and D_{total} measures, it is easy to justify the most significant STCAs in each weekday. Similarly, the specific onset time and location as well as their corresponding delays of the associated bottlenecks, including the active and deactivated bottlenecks, can also be estimated and diagnosed. These consequent measures could be translated into the financial cost of wasted time and fuel, environmental cost of pollutants and carbon emissions etc., which could further help transportation planners to prioritise their congestion mitigation efforts.

The 19 active bottlenecks identified by the analysis are listed in Table 3, ordered based on their frequency, duration and the delay incurred. Of these, three significant active bottlenecks occurred at least 10 times during the study period, with the top-ranked bottleneck occurring at least twice a day and delaying drivers for a total of 6,704 min across the 20 days and the second-ranked at least once a day with delays totalling 5,072 min. Even though the second bottleneck had a shorter total duration, this bottleneck had a higher total delay compared with the first-ranked bottleneck. Interestingly, the 16th-ranked bottleneck had the third highest total delay, possibly because a severe accident occurred at this bottleneck location.

The results from two reports were consistent with our findings. The top-ranked bottleneck identified here is consistent with the findings of a recent AHUA report *Unclogging America's Arteries 2015: Prescriptions for Healthier Highways* [1], which identified the two most congested areas in the Phoenix region as being: (i) the I-10 segment between N 16th Street and N 7th Street, identified in our study, and (ii) the I-17 segment between Sky Harbor Cir and S 24th Street, which is outside the study area. Kandala [25] also identified and quantified 23 bottlenecks causing recurrent congestion during the morning and evening peak hours across the entire Phoenix metropolitan region: the first two bottlenecks identified in our study are both included in Kandala's list. It is important to note that as our new method has been implemented in a computer program, it can be automatically applied to the whole freeway network and process any new datasets with minimal additional effort.

5 Conclusion and future work

The identification and characterisation of traffic congestion and the associated bottlenecks is a challenging problem and as yet no effective tool for traffic congestion analysis based on a series of

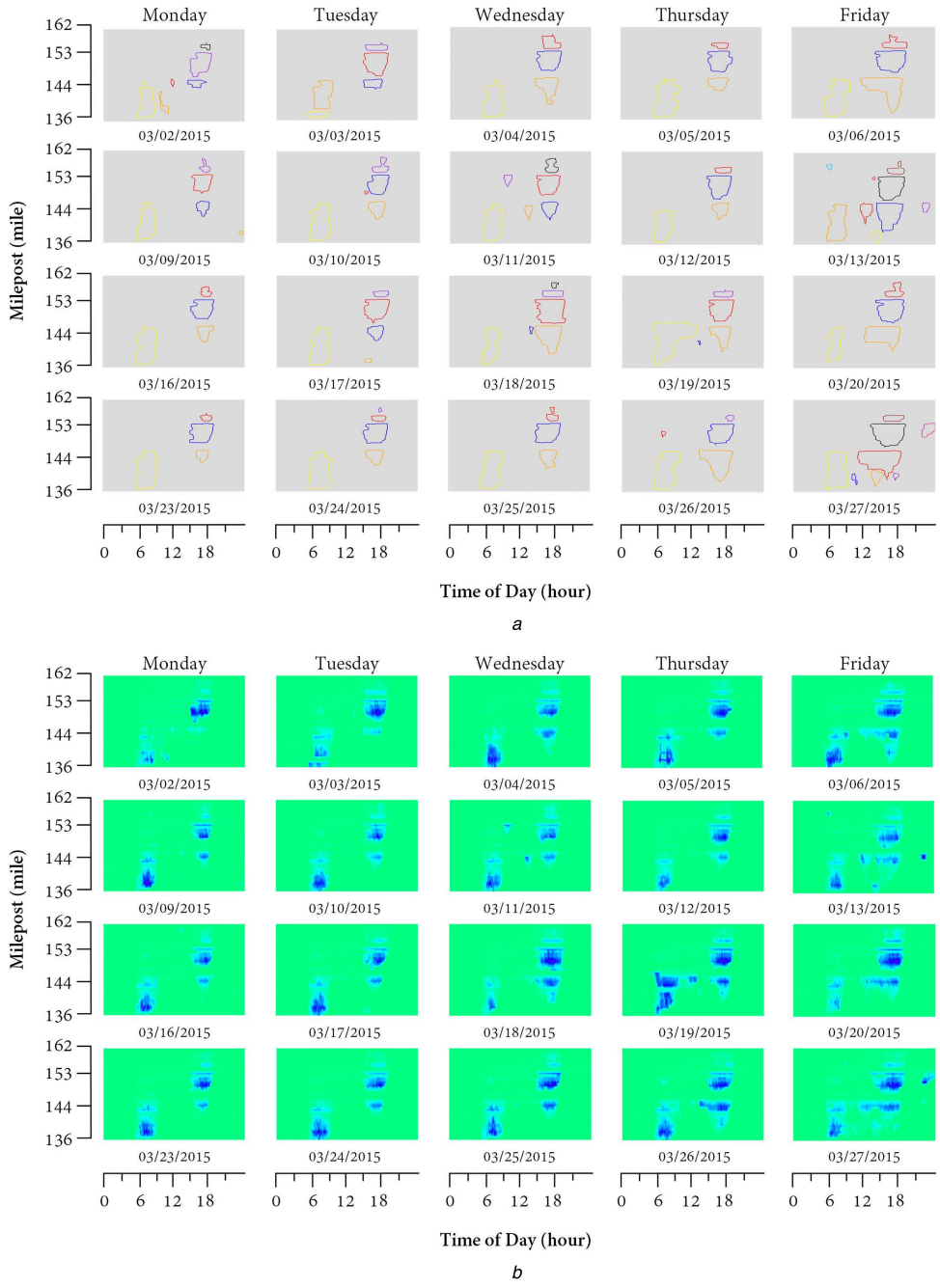


Fig. 8 *Contour extraction and delay calculation*
(a) Extracted contours, ***(b)*** Calculated delays

traffic heatmaps has been fully explored and developed. In this study, a novel method based on a three-stage framework for automatically identifying and characterising the STCAs is proposed. The first stage uses two unsupervised clustering procedures to distinguish and quantify different STCAs along a corridor. During the second stage, a two-step post-processing procedure was developed to refine the identified STCAs and the refined STCAs were further characterised with a series of measures in the final stage. An empirical study of the new method successfully identified and quantified 102 STCAs with 14 traffic measures using 20 weekdays of real world data. At least four significant STCAs were identified on each weekday along the study corridor. After all the STCAs were characterised, 19 active bottlenecks were identified and ranked. The top two bottlenecks identified in our studies were consistent with the findings of two previous studies [1, 25].

The major contributions of this study are as follows:

(1) A novel three-stage framework for identifying and characterising traffic congestion and its associated bottlenecks was developed based on the use of traffic heatmaps and successfully implemented by using a kind of computer programming language.

(2) The new method uses the mini-batch *k*-means algorithm to cluster historic speed profiles as this is several orders of magnitude faster than the conventional *k*-means algorithm while still providing high-quality solutions. The successful implementation of mini-batch *k*-means offers the practitioners an opportunity to address the challenging of increasing data size and face the big-data era. Furthermore, the cut-off speed threshold value can be automatically determined, which makes the method applicable in different cases.

(3) A GTBC algorithm was proposed to automatically distinguish individual STCAs in a given speed heatmap. The clustering procedure achieved this goal by searching for connected components in an undirected graph. The new algorithm automatically determines the optimal number of clusters without the need for prior assumptions or extra validity indices. Experimental results showed that this algorithm outperformed two classical clustering algorithms, *k*-means and fuzzy *c*-means, and a state-of-the-art clustering algorithm, density peaks.

Table 2 STCA quantification results

Date	ID	T_{onset}, min	T_{clear}, min	T_{span}, min	S_{start}, mi	S_{end}, mi	S_{length}, mi	N_{cov_stn} (−)	N_{cov_seg} (−)	D_{total}, veh-h	L_{ab}<mi, mi>	T_{onset_ab}, min	T_{clear_ab}, min	T_{dur_ab}, min	D_{ab}, veh-h
03/02	1	344	546	202	135.84	146.14	10.30	11	11	14,238.3	<146.06, 147.33>	409	438	29	30.9
	2	567	668	101	137.19	143.57	6.38	6	7	1757.9	<142.76, 143.84>	567	596	29	1757.9
	3	856	1051	195	144.55	146.63	2.08	2	3	1952.2	<146.06, 147.33 >	863	1051	188	675.2
	4	690	719	29	144.79	146.88	2.09	2	3	430.1	<146.06, 147.33>	690	719	29	430.1
	5	885	1101	216	147.62	154.36	6.74	9	10	24,092.8	<153.54, 154.63>	942	1101	159	3267.6
	6	993	1087	94	155.10	157.06	1.96	3	4	1189.8	<156.55, 157.12>	1000	1087	87	107.9
03/03	1	322	539	216	135.84	137.56	1.72	2	2	5648.2	<136.90, 137.92>	322	539	217	5306.3
	2	373	575	202	138.29	146.39	8.10	8	9	19,074.4	<146.06, 147.33>	430	567	137	1803.0
	3	885	1079	194	143.81	146.76	2.95	3	4	4821.8	<146.06, 147.33>	885	1079	194	4821.8
	4	885	1137	252	147.62	154.24	6.62	9	10	28,951.1	<153.54, 154.63>	892	1137	245	6645.3
	5	906	1130	224	154.98	157.06	2.08	3	4	3917.4	<156.55, 157.12>	914	1108	194	211.5
03/04	1	337	560	223	135.84	146.14	10.30	11	11	31,771.1	<146.06, 147.33>	452	481	29	31.5
	2	870	1108	238	139.27	146.76	7.49	7	8	14,912.7	<146.06, 147.33>	870	1108	238	14,912.7
	3	885	1137	252	148.72	154.24	5.52	7	8	22,421.6	<153.54, 154.63>	891	1137	246	5654.7
	4	942	1130	188	154.98	158.78	3.80	5	6	3917.7	<157.82, 158.83>	942	1065	123	3917.7
03/05	1	329	604	275	135.84	146.51	10.67	11	11	28,907.7	<146.06, 147.33>	430	553	123	169.1
	2	885	1101	216	143.08	146.76	3.68	3	4	8039.6	<146.06, 147.33>	885	1101	216	8039.6
	3	885	1137	252	148.35	154.24	5.89	8	9	22,721.1	<153.54, 154.63>	885	1108	223	4891.4
	4	928	1094	166	154.98	156.57	1.59	3	4	2119.3	<156.55, 157.12>	1007	1058	51	43.2
03/06	1	329	582	253	135.84	146.27	10.43	11	11	31,434.2	<146.06, 147.33>	402	524	122	182.6
	2	690	1116	426	136.58	146.76	10.18	10	11	25,611.8	<146.06, 147.33>	690	1116	426	25,611.8
	3	834	1144	310	148.23	154.24	6.01	8	9	28,954.7	<153.54, 154.63>	834	1144	310	28,954.7
	4	906	1159	253	154.98	158.78	3.80	5	6	5843.5	<157.82, 158.83>	978	1058	80	872.1

Future work could focus on the following research directions. First, as the traffic heatmaps employed in this study were generated by using the linear interpolation technique. Some other interpolation techniques will be tested and discussed in the future. Second, the sensitivity of the distance of consecutive detectors and the time intervals of data collection to the proposed methodology could also be a topic worth exploring. Third, a potential real-time system could be implemented by combining our new method with another online method, such as Chen's [6]. Fourth, this study mainly focused on identifying and characterising STCAs. It needs to further explore and elaborate the causal factors of these STCAs by considering a wider range of conditions including, for example (a) normal recurrent bottleneck activation, (b) queuing due to incidents, (c) small isolated moving jams, and (d) low speeds not associated with queuing, resulting from severe geometry and/or rough surfaces, as suggested by Kurada *et al.* [13]. In addition, we will test other freeway corridors in different states or countries to further evaluate the feasibility and effectiveness of the proposed method. Since reliability is a very useful measure for traffic authorities and practitioners, another possible future research direction is to develop a model to track, capture and quantify the reliability of each identified STCA.

6 Acknowledgments

The authors thank the anonymous reviewers for their useful comments and suggestions to help improve this paper. Moreover, the authors gratefully acknowledge the financial support received from the Key Research and Development Program of Jiangsu Province (no. BE2017027). We would like to express our sincere appreciation to the ADOT (Arizona Department of Transportation) for providing the study data. Special thanks go to Vahid Goftar and Brent Cain for their professional support to this research.

Table 3 Ranking of the identified active bottlenecks

No.	Marked mileposts (mi, mi)	Location	Frequency (−)	Duration, min	Total delay, veh-h
1[a,b]	<146.06, 147.33>	I-10 EB 7TH ST/I-10 EB 19TH ST	42	6704	227,166.2
2[b]	<153.54, 154.63>	I-10 EB BROADWAY RD/I-10 EB SOUTHERN AVE	23	5072	230,347.2
3	<156.55, 157.12>	I-10 EB N OF GUADALUPE RD/I-10 EB S OF GUADALUPE RD	10	1076	1176.7
4	<157.82, 158.83>	I-10 EB ELLIOT RD/I-10 EB WARNER RD	5	311	6220.7
5	<158.83, 160.25>	I-10 EB WARNER RD/I-10 EB S OF RAY RD	4	252	2857.7
6	<157.12, 157.82>	I-10 EB S OF GUADALUPE RD/I-10 EB ELLIOT RD	3	151	1866.9
7	<136.90, 137.92>	I-10 EB 75TH AVE/I-10 EB 67TH AVE	2	289	5872.1
8	<139.79, 140.81>	I-10 EB 51ST AVE/I-10 EB 43RD AVE	2	86	1387.0
9	<137.92, 138.80>	I-10 EB 67TH AVE/I-10 EB 59TH AVE	1	36	182.9
10	<138.80, 139.79>	I-10 EB 59TH AVE/I-10 EB 51ST AVE	1	43	2890.3
11	<140.81, 141.83>	I-10 EB 43RD AVE/I-10 EB 35TH AVE	1	65	2503.3
12	<141.83, 142.76>	I-10 EB 35TH AVE/I-10 EB 27TH AVE	1	29	152.2
13	<142.76, 143.84>	I-10 EB 27TH AVE/I-10 EB 19TH AVE	1	29	1757.9
14	<143.84, 144.96>	I-10 EB 19TH AVE/I-10 EB 5TH AVE	1	180	6047.6
15	<144.96, 146.06>	I-10 EB 5TH AVE/I-10 EB 7TH ST	1	79	3069
16	<147.33, 147.93>	I-10 EB 19TH ST/I-10 EB N OF WASHINGTON ST	1	375	29,067.2
17	<149.17, 150.24>	I-10 EB BUCKEYE RD/I-10 EB 24TH ST	1	29	242.6
18	<151.66, 152.41>	I-10 EB 32ND ST/I-10 EB 40TH ST	1	43	465.5
19	<155.88, 156.55>	I-10 EB BASELINE RD/I-10 EB N OF GUADALUPE RD	1	145	2215.3

[a]Top bottleneck identified in AHUA [1].

[b]Top bottlenecks identified in Kandala [25].

7 References

[1] CPCS Transcom Limited: '*Unclogging America's arteries 2015: prescriptions for healthier highways*' (American Highway Users Alliance, Washington, DC, USA, 2015), pp. 4–5

[2] Cambridge Systematics, Inc. and Texas Transportation Institute: '*Traffic congestion and reliability: linking solutions to problems*'(FHWA, U.S. Department of Transportation, Washington, DC, USA, 2004), pp. 5–6

[3] Cooner, S.A., Ranft, S.E., Zimmerman, C.: '*An agency guide on overcoming unique challenges to localized congestion reduction projects*' (FHWA, U.S. Department of Transportation, Washington, DC, USA, 2011), pp. 2–6

[4] Hale, D., Jagannathan, R., Xyntarakis, M., *et al.*: '*Traffic bottlenecks: identification and Solutions*' (FHWA, U.S. Department of Transportation, Washington, DC, USA, 2016), pp. 1–4

[5] Cassidy, M.J., Bertini, R.L.: 'Some traffic features at freeway bottlenecks', *Transp. Res. B, Methodol.*, 1999, **33**, (1), pp. 25–42

[6] Chen, C., Skabardonis, A., Varaiya, P.: 'Systematic identification of freeway bottlenecks', *Transp. Res. Rec., J. Transp. Res. Board*, 2004, (1867), pp. 46–52

[7] Zhang, L., Levinson, D.: 'Some properties of flows at freeway bottlenecks', *Transp. Res. Rec., J. Transp. Res. Board*, 2004, (1883), pp. 122–131

[8] Ban, X., Chu, L., Benouar, H.: 'Bottleneck identification and calibration for corridor management planning', *Transp. Res. Rec., J. Transp. Res. Board*, 2007, (1999), pp. 40–53

[9] Liu, K., Fei, X.: 'A fuzzy-logic-based system for freeway bottleneck severity diagnosis in a sensor network', *Transp. Res. C, Emerg. Technol.*, 2010, **18**, (4), pp. 554–567

[10] Zheng, Z., Ahn, S., Chen, D., *et al.*: 'Applications of wavelet transform for analysis of freeway traffic: bottlenecks, transient traffic, and traffic oscillations', *Transp. Res. B, Methodol.*, 2011, **45**, (2), pp. 372–384

[11] Bertini, R.L.: 'Detecting signals of bottleneck activation for freeway operations and control', *J. Intell. Transp. Syst.*, 2005, **9**, (1), pp. 35–45

[12] Daganzo, C.F.: '*Fundamentals of transportation and traffic operations*' (Pergamon Press, Oxford, 1997, 1st edn.)

[13] Kurada, L., Öğüt, K., Banks, J.: 'Evaluation of N-curve methodology for analysis of complex bottlenecks', *Transp. Res. Rec., J. Transp. Res. Board*, 2007, (1999), pp. 54–61

[14] Das, S., Levinson, D.: 'Queuing and statistical analysis of freeway bottleneck formation', *J. Transp. Eng.*, 2004, **130**, (6), pp. 787–795

[15] 'Caltrans Performance Measurement System (PeMS)'. Available at http://pems.dot.ca.gov/, accessed 6 June 2017

[16] 'Portland Oregon Regional Transportation Archive Listing (PORTAL)'. Available at http://portal.its.pdx.edu/, accessed 6 June 2017

[17] Leontiadis, I., Marfia, G., Mack, D., *et al.*: 'On the effectiveness of an opportunistic traffic management system for vehicular networks', *IEEE Trans. Intell. Transp. Syst.*, 2011, **12**, (4), pp. 1537–1548

[18] Liu, S., Pu, J., Luo, Q., *et al.*: 'Vait: a visual analytics system for metropolitan transportation', *IEEE Trans. Intell. Transp. Syst.*, 2013, **14**, (4), pp. 1586–1596

[19] Wieczorek, J., Fernández-Moctezuma, R., Bertini, R.: 'Techniques for validating an automatic bottleneck detection tool using archived freeway sensor data', *Transp. Res. Rec., J. Transp. Res. Board*, 2010, (2160), pp. 87–95

[20] Sculley, D.: 'Web-scale *k*-means clustering'. Proc. 19th Int. Con. on World Wide Web, Raleigh, USA, April, 2010, pp. 1177–1178

[21] Lloyd, S.: 'Least squares quantization in PCM', *IEEE Trans. Inf. Theory*, 1982, **28**, (2), pp. 129–137

[22] Charrad, M., Ghazzali, N., Boiteau, V., *et al.*: 'Package NbClust: an R package for determining the relevant number of clusters in a data set', *J. Stat. Softw.*, 2014, **61**, (6), pp. 1–36

[23] Even, S.: '*Graph algorithms*' (Cambridge University Press, Cambridge, 2011, 2nd edn.)

[24] Akima, H.: 'Algorithm 761: scattered-data surface fitting that has the accuracy of a cubic polynomial', *ACM Trans. Math. Softw.*, 1996, **22**, (3), pp. 362–371

[25] Kandala, S.S.: 'Analysis of freeway bottlenecks'. PhD thesis, Arizona State University, 2014

[26] Pal, N.R., Bezdek, J.C., Hathaway, R.J.: 'Sequential competitive learning and the fuzzy c-means clustering algorithms', *Neural Netw.*, 1996, **9**, (5), pp. 787–796

[27] Rodriguez, A., Laio, A.: 'Clustering by fast search and find of density peaks', *Science*, 2014, **344**, (6191), pp. 1492–1496

[28] Kodinariya, T.M., Makwana, P.R.: 'Review on determining number of cluster in *K*-means clustering', *Int. J. Adv. Res. Comput. Sci. Manage. Stud.*, 2013, **1**, (6), pp. 90–95

[29] Rousseeuw, P.J.: 'Silhouettes: a graphical aid to the interpretation and validation of cluster analysis', *J. Comput. Appl. Math.*, 1987, **20**, pp. 53–65

[30] Luxburg, U.V.: 'Clustering stability: an overview', *Found. Trends Mach. Learn.*, 2010, **2**, (3), pp. 235–274