University Library

MINERVA ACCESS

A gateway to Melbourne's research publications

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Sarteshnizi, IT;Sarvi, M;Bagloee, SA;Nassir, N

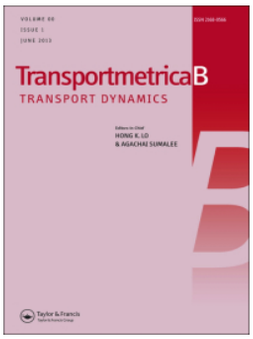
Title:
Temporal pattern mining of urban traffic volume data: a pairwise hybrid clustering method

Date:
2023

Citation:
Sarteshnizi, I. T., Sarvi, M., Bagloee, S. A. & Nassir, N. (2023). Temporal pattern mining of urban traffic volume data: a pairwise hybrid clustering method. Transportmetrica B: Transport Dynamics, 11 (1), pp.1186-1217. https://doi.org/10.1080/21680566.2023.2185496.

Persistent Link:
https://hdl.handle.net/11343/326422

License:
CC BY-NC-ND



Transportmetrica B: Transport Dynamics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ttrb20

Temporal pattern mining of urban traffic volume data: a pairwise hybrid clustering method

Iman Taheri Sarteshnizi, Majid Sarvi, Saeed Asadi Bagloee & Neema Nassir

To cite this article: Iman Taheri Sarteshnizi, Majid Sarvi, Saeed Asadi Bagloee & Neema Nassir (2023) Temporal pattern mining of urban traffic volume data: a pairwise hybrid clustering method, Transportmetrica B: Transport Dynamics, 11:1, 2185496, DOI: 10.1080/21680566.2023.2185496

To link to this article: https://doi.org/10.1080/21680566.2023.2185496

© 2023 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group

Published online: 09 Mar 2023.

Submit your article to this journal

Article views: 513

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=ttrb20

TRANSPORTMETRICA B: TRANSPORT DYNAMICS
2023, VOL. 11, NO. 1, 1186–1217
https://doi.org/10.1080/21680566.2023.2185496

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Temporal pattern mining of urban traffic volume data: a pairwise hybrid clustering method

Iman Taheri Sarteshnizi, Majid Sarvi, Saeed Asadi Bagloee and Neema Nassir

Department of Infrastructure Engineering, University of Melbourne, Melbourne, Australia

ABSTRACT

Multiple pattern analyses of traffic data have been conducted previously; however, it has yet to be explored with an awareness of temporal factors in big real-world traffic data. In this paper, we introduce a hybrid method to measure the intensity of differences among various temporal factors' data. The proposed method can efficiently process the historical data given temporal factors and provide insightful information about the intensity of variations. After data denoising with basis splines, we reshape the time series into a 2-D latent space using Principal Component Analysis (PCA) according to the type of analysis. Pairwise K-means clustering is then applied after anomaly elimination with DBSCAN to derive Adjusted Rand Index (ARI) matrices. Finally, these matrices are then systematically used to find similar patterns of different temporal perspectives. Multiple analyses are carried out with real data from Melbourne, Australia. Dissimilarities with intensities of up to 80% are detected that are not detectable with general clustering approaches.

ARTICLE HISTORY
Received 2 June 2022
Accepted 23 February 2023

KEYWORDS
Temporal pattern mining; traffic volume data; pairwise clustering

Introduction

The emergence of high-speed communication technologies, artificial intelligence, and advanced sensors can facilitate better management of our transport systems. A wide range of sensor devices and communication facilities are installed on roads and intersections daily (Sarvi, Asadi, and Van Uytsel 2021). The wealth of resulting data provides an unprecedented opportunity to better understand traffic network performance and address chronic traffic congestions, disruptions, and safety risks more effectively. However, the lack of proper platforms and techniques to process big data effectively beset the efficiency of applications.

Thanks to Intelligent Transportation Systems (ITS), historical traffic data has become readily available everywhere, bringing new challenges to scientists (Emami, Sarvi, and Asadi Bagloee 2019). Although handling big data becomes more complicated over time, new applications, insights, and methodologies emerge as a result of exploring these ever-growing databases. Among different data collection devices, loop detector traffic volume data is one of the earliest and mainly deployed data collectors in ITS with applications. It has shown significant advantages in several studies, from classical OD estimation (Hellinga and Aerde 1998) and traffic assignment (Vaze et al. 2009) to congestion detection (Kalinic and Krisp 2019), accident detection (Parsa et al. 2020) as well as traffic prediction (Chen et al. 2021; Emami, Sarvi, and Bagloee 2021).

CONTACT Iman Taheri Sarteshnizi itaherisarte@student.unimelb.edu.au Department of Infrastructure Engineering, University of Melbourne, Melbourne, VIC 3053, Australia

© 2023 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (http://creativecommons.org/licenses/by-nc-nd/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited, and is not altered, transformed, or built upon in any way. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

Data collected by loop detectors (e.g. speed, volume, and occupancy) represent the macroscopic behavior of traffic dynamics within a specific location, and it is subject to some contextual factors (Qu et al. 2021), mainly 'temporal' ones. For instance, the time of the day, day of the week, month of the year, season of the year, and even the year itself can significantly alter the underlying distribution of data (Horowitz et al. 2014). In addition, some other types of contextual factors exist, such as weather conditions or pandemic situations like COVID-19, which caused unprecedented disruptions to urban traffic systems (Liu and Stern 2021; Morita, Nakamura, and Hayashi 2020). It is noteworthy that some effects of contextual factors, especially temporal ones, on traffic data are predictable and easily understandable (e.g. the difference between workdays and holidays). However, some others cannot be fully detected or understood by preexisting knowledge, especially when it comes to big data.

Our aim is to mine the patterns of traffic big data with respect to temporal factors. In other words, we want to know how different the data of various temporal factors are compared to each other. Therefore, given the traffic data of each link in a network, we will return scores representing the degree of homogeneity between every pair of factors based on the data. This is valuable from two different practical and research aspects: 1) it reveals the underlying data patterns that might be hard to detect by simple data scanning and visualizations. 2) it helps data scientists to create a more meaningful feature space of temporal factors for further model training. There are plenty of recent works in the literature signifying the impacts of temporal factors on traffic-related models, specifically data-driven predictive ones (Ma, Song, and Li 2021; Qu et al. 2021). It is also shown that exploring traffic data patterns before model development results in superior outcomes (Habtemichael and Cetin 2016; Song et al. 2018; Leiser and Yildirimoglu 2021). However, despite these benefits, there is a lack of sufficient research on developing fast and efficient methodologies mining these patterns in big datasets before any model development.

Attempts are devoted to pattern mining of traffic data by applying statistical methods like ANOVA (Rakha and Aerde 1995) and clustering methods such as agglomerative and spectral clustering (Soriguera 2012; Yang et al. 2017). Functional Data Analysis (FDA) also proved helpful in this regard (Guardiola, Leon, and Mallor 2014; Crawford, Watling, and Connors 2017). Although the techniques recommended in these works are promising, pattern mining of traffic data with the temporal factors given as input still needs to be comprehensively studied. The main distinctions of this work compared to the literature can be stated as follows: 1) temporal factors are used as input in our methodology; therefore, the intensity of the differences between every possible pair of factors will indeed be reported. A more precise outcome is provided by conducting the pattern mining this way. For example, if clustering algorithms are applied to data of all days of the week, the difference between workdays and weekends prevents any model from checking other differences among the workdays. 2) our methodology can analyze the differences among the data of all temporal factors like time of the day, day of the week, month of the year, and the year itself. Earlier studies mainly focus on just one or a few of them. Since the temporal dimension of data is very long in some cases, various types of temporal patterns are presented in the data, and the method should be capable of exploring all of these types.

With attention to the above-mentioned points, we propose a hybrid method that automatically explores the patterns of big volume data with regard to temporal factors and returns matrices demonstrating the intensity of dissimilarities between every pair of factors. Principal Component Analysis (PCA), a computationally efficient method for dimension reduction, is utilized to change the shape of volume data into a 2-D latent space. This ability of PCA significantly reduces the complexity of data and makes it favorable for big data applications as it instantly compresses long daily time series into just two variables. It also adds to interpretability since 2-dimensional data is always easier to visualize and comprehend. Moreover, DBSCAN is employed to detect outliers related to sensor errors and public holidays in the latent space, as they can be deceptive for clustering algorithms. After outlier detection, pairwise clustering is also performed, given the data of every possible pair of temporal factors with the K-means algorithm. The main goal of pairwise clustering is to determine whether the targeted data pairs are meaningfully different. To measure the extent of this meaningfulness, Adjusted Rand

Index (ARI) matrices are utilized to compare the ground truth (temporal factor labels) with the outcome of K-means clustering. These matrices represent the intensity of the differences observed in the data and can be used as a dissimilarity score.

To the best of our knowledge, our approach discovers temporal data patterns with more details compared to the literature (Crawford, Watling, and Connors 2017; Yang et al. 2017; Guardiola, Leon, and Mallor 2014; Soriguera 2012; Stathopoulos & Karlaftis, 200) as it exploits pairwise clustering and benefits from temporal factors. We used six years of volume data recorded in different locations in Melbourne, Australia, to verify our method. The contributions of this research paper are listed below:

- We exploit the PCA to represent the traffic data in a latent space and then compare it in terms of given temporal factors. To the best of our knowledge, exploring different types of temporal factors with PCA is not investigated in the literature.
- The proposed methodology in this paper conducts clustering of the data pairwisely regarding temporal factors. This reveals more insights from the traffic data compared to the previous methodologies exploiting general perspectives for clustering.
- This paper analyzes different types of temporal factors such as time of the day, day of the week, month of the year, and the year itself, which is not the case in earlier efforts. Notable results and applications are provided in this research as a result of conducting this range of analyses.

The rest of the paper is organized as follows. In the next chapter, we discuss previous works and explain the novelty of this paper. In chapter 3, the structure of the methodology is described. Chapter 4 reviews the characteristics of the datasets used in this research. The results and outputs of this study are also explained in chapter 5. Finally, chapter 6 provides a summary and conclusion of the content.

Literature review

To comprehensively review the literature related to our investigation, we discuss efforts devoted to pattern mining in traffic data, along with studies establishing data-driven predictive models. The reason for including the latter type is that they are the most recent studies containing challenges related to temporal factors, and it is imperative to know how they deal with these factors. In the following sections, we first review the pattern mining studies and then discuss data-driven predictive models in the literature. In the end, we also explain how this research addresses current research gaps in the literature.

Pattern mining studies

Studies in this group aim to classify the daily traffic volume or speed patterns into different groups. The main objective of these efforts is to provide insights for traffic agencies about the daily traffic behavior in the network or a specific road. In terms of methodology, statistical tests and clustering algorithms are often used in the literature for data exploration. For instance, earlier studies (H. Rakha and Aerde 1995) used the Analysis of Variance test (ANOVA) to investigate the variations and distributions of traffic data. However, some researchers argue that traffic data only sometimes satisfies the assumptions of parametric tests such as ANOVA (Stathopoulos and Karlaftis 2001). Clustering algorithms were also found to be applicable and insightful for pattern mining. Without strict assumptions, these methods are capable of finding complicated patterns in data with higher dimensions. For instance, Soriguera (2012) and Yang et al. (2017), utilized agglomerative and spectral clustering to group daily traffic data patterns. However, their investigation was limited to daily variations in a short period and low-frequency collected data (hourly collected).

Other studies can also be found in the literature focusing on traffic pattern investigation; however, their methodologies were designed for specific applications. Jiang and Adeli (2004) developed

a Wavelet Packet-Autocorrelation Function Method to denoise the traffic data and identify its singularities. Pascale et al. (2015) also proposed an algorithm using Ward's method to extract daily speed patterns from the data of a few connected trucks. Tensor decomposition was used by Yang et al. (2019b) to detect homogenous subnetworks over time using two months of loop detector data. In addition, Functional Data Analysis (FDA) (Li and Chiou 2020) also captured attention for analyzing traffic volume data in previous works. For instance, Guardiola, Leon, and Mallor 2014; and Crawford, Watling, and Connors 2017 fitted spline functions into the aggregated data collected by loop detectors to benefit from functional PCA (FPCA) and functional linear modeling. Guardiola, Leon, and Mallor 2014 reshaped the daily traffic functions into a 2-D space using FPCA and clustered them to provide a traffic monitoring system. Crawford, Watling, and Connors 2017 also created multiple functional linear models to investigate the effects of the day of the week feature space on functional prediction accuracy.

Efforts in this group are substantial in developing practical approaches to explore historical traffic data; however, more attention needs to be paid to pattern mining with an awareness of temporal factors. Looking into the data without benefiting from temporal factors may result in imprecise clustering since some differences among daily profiles are not easily detectable.

Data-driven predictive models

Temporal factors are one of the main concerns that should be addressed before developing any data-driven predictive model. Previous efforts in this group can be divided into two main categories regarding their approach to tackling temporal factors. Using pre-existing knowledge and clustering of daily traffic patterns are mostly adopted in the literature for prediction, each of which will be discussed in the following.

A vast majority of previous works (Liu, Liu, and Jia 2019; Lv et al. 2015; Yang et al. 2019a; Zhao et al. 2020: and Yao, Zhang, and Long 2021) have used assumptions or pre-existing knowledge to develop a predictive model. This means that authors fed the data into their models without prior pattern mining to explore the exact extent of differences between the data of temporal factors. They utilized simple facts like 'there is a considerable difference between the data of workdays and weekends' for model training and assumed that the behavior of data during different months is similar. Yang et al. (2019a) even excluded weekends in their analysis. In addition to these studies, other efforts were observed in the literature employing a feature reduction algorithm for the temporal feature space (Ma, Song, and Li 2021; Qu et al. 2021). Although temporal feature space improved the prediction performance in these studies, defining it using pattern exploration of data is a more precise approach and avoids overfitting problems.

On the other hand, other research papers in the literature also emphasize pattern exploration of data prior to the prediction. Wang and Shi (2013) used simple visualization approaches to define the temporal factors feature space. Habtemichael and Cetin (2016) clustered daily volume profiles using enhanced K-Nearest Neighbors and made predictions based on the similarity of the data from the targeted day and the extracted clusters. Their algorithm is tested using data from different locations, and superior results are achieved compared to previous works. Song et al. (2018) also developed a Match-then-Predict method to predict the whole daily volume profile of a targeted day. In this work, historical profiles are first clustered using the density peak clustering method; then, predictions are made based on comparing the temporal factors of the target day and created clusters. The idea of clustering the data before making a prediction is promising; however, the approaches established in these works are similar to the ones in Pattern Mining studies and do not include pattern mining given temporal factors as input.

In this paper, we explore road traffic big volume data of different locations with respect to temporal factors and show how different they are compared to each other. For this objective, we develop an approach that takes advantage of Principal Component Analysis (PCA) to reshape the data with different dimensions and represent it in a 2-D latent space. Using a latent space to represent the traffic data

has been adopted recently and has shown practical applications (Boquet et al. 2020; Guardiola, Leon, and Mallor 2014). Moreover, this change of dimension represents any data within a unique shape that is insightful and representative and empowers clustering algorithms such as K-means and DBSCAN that have shown better performance in a 2-D space previously (Keogh and Lin 2005). It is also shown in the results that pattern mining of traffic data, given all the temporal factors, may reveal more insights about traffic behavior in a road section.

Problem statement

Traffic volume data of a whole day recorded in a specific location (either by a loop detector or any other sensor) can be stated as $X = \{(x_k)\}_{k=0}^{m}$ with x_k representing a single observation recorded in the k^{th} time interval and m representing the total number of recorded observations during a single day. As loop detectors always capture the aggregate volumes of traffic in discrete time intervals with duration T ($T = 5, 10$ or 15 min), m would be equal to $24 \times 60/T$. In this study, the volume data is collected every $T = 15$ minutes, and therefore, m would be equal to 96 data points. Furthermore, a feature space of $Z^{m\times n}$ is always assigned to the daily traffic data describing different temporal factors, as demonstrated in Equation 1:

$$Z^{m\times n} = [z^{k,l} \in \{0, 1\}] \tag{1}$$

In this equation, $z^{k,l}$ is a binary number denoting if a single record belongs to the l^{th} ($l = 1, 2, \ldots, n$) temporal feature or not, and the number of features (n) representing the temporal factors depends on the available information. Temporal factors like time of the day (96 features), day of the week (7 features), month of the year (12 features), and the year itself (6 features) are considered in this study leading to $n = 121$. Therefore, along with any volume data point recorded in a specific location, there are 121 other features clarifying its temporal characteristics.

Our aim in this paper is to explore the traffic volume data of each location and measure how different the data is considering these 121 temporal features. In other words, we intend to automatically analyze an extensive volume dataset collected from different locations and return a score stating the similarity of the data collected during each pair of different temporal factors in each location. Looking into the results, one could claim multiple statements according to the factors such as the degree of dissimilarity between the data of Mondays and Tuesdays of location A is very high (or very low, depending on the case). By the results, we can also define a revised feature space $Z'^{m'\times n'}$, a more meaningful feature space of temporal factors than the Z. The mathematical representation of Z' is provided below where $m' < m$ and $n' < n$:

$$Z'^{m'\times n'} = [z'^{k',l'} \in \{0, 1\}] \tag{2}$$

Method

This section elaborates on our proposed methodology for extracting dissimilarities in traffic data. A general overview of the model architecture and its inputs and outputs are depicted in Figure 1. We will first discuss the data pre-processing methods used in this paper and then describe different types of data inputs used for pattern exploration in our method. Finally, we will explain the details of the data reshaping and pairwise clustering part of our method, which leads to the model's output (ARI matrices).

Data pre-processing

Missing data imputation

Loop detector raw data usually contains missing values, which should be replaced or removed before further investigation. In this research, records with $x^i < 0$ or empty values are both considered missing

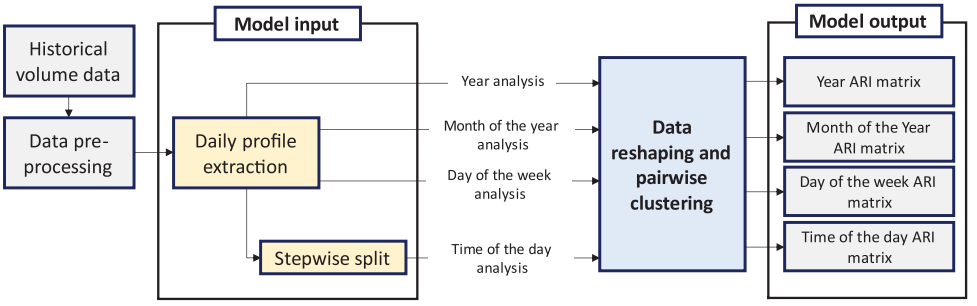


Figure 1. The proposed framework in this paper to identify the patterns of data according to its temporal factors.

values and replaced by other records possessing temporal factors like the targeted missing values. In other words, there are four x_i in each dataset for each specific z_i (as there are four weeks in each month), and any missing value in our dataset is replaced with the average of the other three records with the same z_i. This approach is exploited since the proportion of the missing values to the correct values is very low (less than 1%) in our dataset. Other methods should be considered and applied when this proportion is significant.

B-spline fitting

Superior results are achieved in data clustering by applying denoising approaches prior to the analysis. It is shown that clean data better describe the underlying nature of reality rather than raw and noisy data (Hitchcock, Booth, and Casella 2007). In addition to these proven facts, data denoising can prevent the effects of extreme data points in any analysis. Loop detector data, in our case, is always contaminated with point outliers and extreme values like the one shown in Figure 2. Although the daily pattern of data is recorded precisely, this extreme observation may mislead our linear dimension reduction in the following steps deriving the whole data pattern. Therefore, with just one extreme point, the whole pattern of data would be wrongly represented. Therefore, to avoid eliminating beneficial information, B-spline (basis spline) fitting is adopted in this study to denoise the volume data before reshaping it into a latent space. B-splines have been implemented previously in the literature (Crawford, Watling, and Connors 2017; Guardiola, Leon, and Mallor 2014) to describe the daily volume profiles as functions and benefit from their continuous nature. However, the main purpose of using them in this study is to smoothen the collected noisy data and better describe the underlying behavior of traffic data.

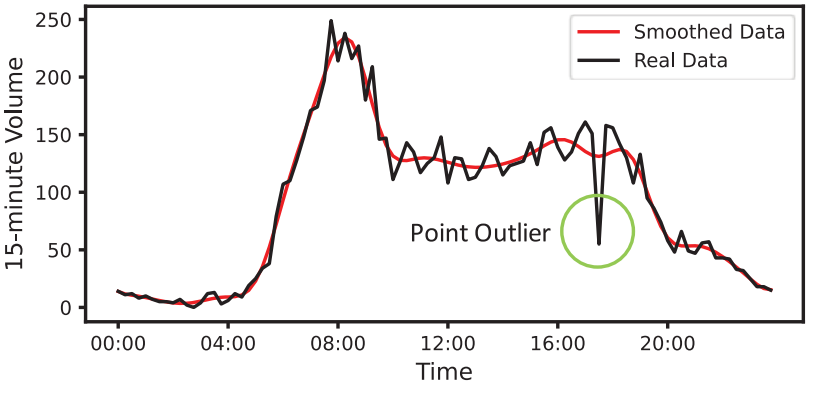


Figure 2. A sample of point outlier in volume data and the effect of B-spline smoothing.

B-spline curves (Bartels, Barsky, and Beatty 1987) are piecewise polynomials inspired by the classic Bézier curves. The main idea is to fit a linear combination of some basis splines into aggregated data to represent its continuous form. A d-degree B-spline curve $C(t)$ is defined by:

$$C(t) = \sum_{i=1}^{n} B_{i,d}(t) Q_i \tag{3}$$

where $\{Q_i\}_{i=1}^{n}$ is a set of $n+1$ control points ($1 \leq d \leq n$) and $B_{i,d}(t)$ represents the B-spline basis functions. A non-decreasing sequence of scalars t_i ($0 \leq i \leq n+d+1$) called knots are needed to derive the basis- functions of equation (3) recursively. A primary basis function is shown in equation (4) for $0 \leq i \leq n+d$ which can be used as a starting point for calculating the other ones. Equation (5) is also showing the recursion for $1 \leq j \leq d$ and $0 \leq i \leq n+d-j$:

$$B_{i,0}(t) = \begin{cases} 1, & t_i \leq t \leq t_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$B_{i,j}(t) = \frac{t - t_i}{t_{i+j} - t_i} B_{i,j-1}(t) + \frac{t_{i+j+1} - t}{t_{i+j+1} - t_{i+1}} B_{i+1,j-1}(t) \tag{5}$$

To fit a B-spline curve into the daily volume profiles, the control points $\{Q_i\}_{i=1}^{n}$ and the knots t_i mentioned above are unknown and should be determined. Given a set of knots, optimal control points can be extracted using the Least-Squares fitting approach. Let $\{(x_k)\}_{k=0}^{m}$ be the volume time series data of a single day. Since a B-spline curve is formulated using $t \epsilon [0, 1]$, $[0, m]$ should also be reshaped into this interval as $t_k = k/m$. By minimization of the error term $E(\hat{Q})$ stated in equation (6), one can achieve the desired control points resulting in a minimum amount of fit error:

$$E(\hat{Q}) = \frac{1}{2} \sum_{k=0}^{m} \left| \sum_{j=0}^{n} B_{j,d}(t_k) Q_j - x_k \right|^2 \tag{6}$$

As stated, the knot set should be specified prior to solving the above optimization problem, and the final shape of the fitted B-spline curve highly depends on it. Results using different sets of knots are visually evaluated in this research and will be discussed in the result section of this paper.

Data rescaling

Data rescaling is conducted after the B-spline curve fitting into the daily profiles. This is the case in most of the experiments in this paper, where we need to remove the effects of the size and just focus on profile shapes. In this rescaling, the raw data x_k becomes $x_k^{'}$ by the following equation:

$$x_k^{'} = \frac{C(t_k)}{(\sum_{k=0}^{m} C(t_k))/m} \tag{7}$$

where $C(t_k)$ is the smoothed value of the volume data achieved from the previous step and k is sample time (from 0 to m). This formula divides every data point of a daily volume profile ($C(t_k)$) by the average daily volume ($\sum_{k=0}^{m} C(t_k)$ is the area under the time series). A better representation of the parameters is depicted in Figure 3. In this figure, the area under the profile ($\sum_{k=0}^{m} C(t_k)$) is shadowed by the yellow color, and it is shown that m is the number of samples collected in a day. By this rescaling, we just take the shape of the profiles into account and neglect the differences in their amplitude.

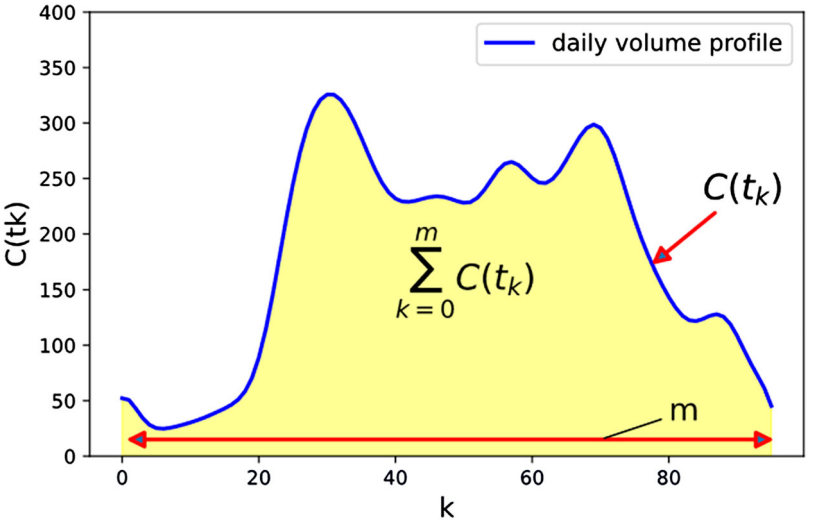


Figure 3. A depiction of the parameters in rescaling equation using a smoothed sample volume profile (equation 7). Based on the equation, every data point in a volume profile ($C(t_k)$: the blue curve) should be divided by the average of all values observed in that day ($\sum_{k=0}^{m} C(t_k)/m$: the yellow area under the curve).

Model inputs

According to Figure 1, data reshaping and pairwise clustering are implemented separately for each type of temporal factor. The input data for the year, month of the year, and day of the week are preprocessed daily volume profiles, and by daily volume profile, we mean all the filtered data points collected during a day from 00:00 AM to 11:59 PM. Using daily volume profiles as input, one should carefully avoid the effects of other factors while focusing on a single factor. For example, when our intention is to analyze the year, we should ensure that homogenous daily volume profiles in terms of the day of the week and month of the year are included in our analysis. On the other side, in time of the day analysis, as it is observed, a stepwise split of the data is used after extracting the daily profiles. To better illustrate the stepwise split after the daily profile extraction, Figure 4 is provided. We can see in this figure that the data of different time intervals (or steps) denoted as x_l are extracted from daily profiles $X = \{(x_k)\}_{k=0}^{96}$ during homogenous days of the week to be the method's input. Since we should use the homogenous daily profiles in the time of the day analysis, it is necessary to use the output of the year, month, and day of the week analysis. Therefore, daily volume profile analyses should be primarily implemented.

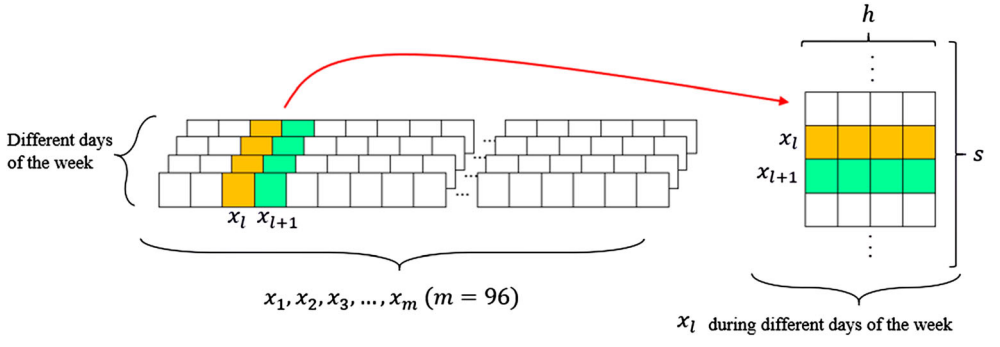


Figure 4. An illustration of stepwise split of data for time of the day analysis.

According to the above explanation, $D^{r\times m}$ is the model input (R is the number of days in the dataset) for the year, month of the year, and day of the week analysis consisting of homogenous daily volume profiles. $T^{s\times h}$ is the model input for the time of the day analysis (s and h are indicated in Figure 4).

Data reshaping and pairwise clustering

An overview of the process in this part is demonstrated in Figure 5 with more details. As observed, one main loop in this process calculates ARIs (dissimilarity scores) related to each pair of temporal factors and sends it to the final ARI matrix. In other words, when the method receives the input data ($D^{r\times m}$ or $T^{s\times h}$), it calculates a dissimilarity score between 0 and 1 for every possible pair of temporal factors. For example, if the input contains the filtered homogenous daily profiles of 2014–2019, the loop will be executed for each pair like (2014, 2015), (2014, 2016), … , (2018, 2019). The exact calculation of each step and its mathematical representation will be further discussed in the following subsections.

Dimension reduction with PCA

We reduce the dimension of data in our method before further analysis as it helps us better manage and discover the data. In big data cases, it is of paramount importance to keep the computational cost as low as possible and work with parts of the data that are substantially needed. Among different dimension reduction methodologies in the literature, from linear to non-linear ones, we selected Principal Component Analysis (PCA) as our method's backbone. There are some main reasons behind this decision:

1) PCA is a linear and deterministic approach that performs much faster than recent non-linear ones such as Autoencoders. Therefore, it can be utilized in big data applications where data from several different locations are available.
2) The objective function of this method maximizes the variance, and since our aim is to discover the differences in different data groups, this feature strongly suits our needs. In other words, PCA leads to more effortless capturing of differences among the data samples.
3) We also showed in Appendix 1 that PCA is robust to the data size and can be used for pattern exploration with low amounts of data (a few months). This is not possible with other learning-based dimension reduction algorithms.
4) PCA orthogonally projects the input data into a lower dimension principal subspace (or a latent space) such that the variance of the projected data is maximized (Bishop 2006) and no pre-training of the model is needed making it more practical for fast calculations.

Suppose that $\{x_n\}$ is a D-dimension set of data and $n = 1, 2, \ldots, N$ is the number of observations included in the experiment. The data should be projected into a lower dimension M ($M < D$) while the variance of the reshaped data is maximized. To keep the explanation simple and understandable,

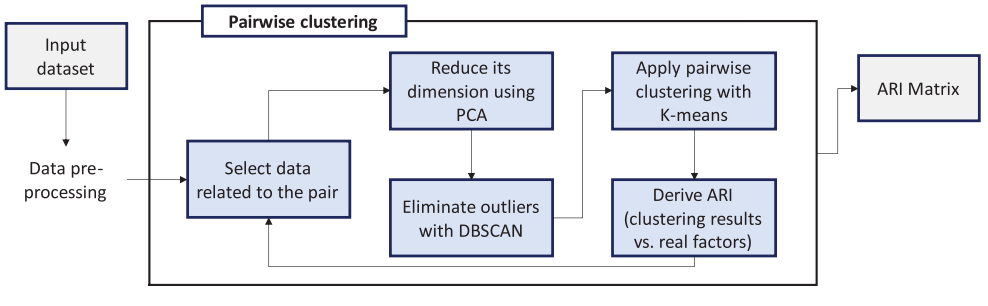


Figure 5. A detailed illustration of data reshaping and pairwise clustering process.

we first suppose that $M = 1$, and in the end, we will generalize the process. Let $\boldsymbol{u}_1$ be a D-dimensional unit vector denoting the direction of the latent space. As $\boldsymbol{u}_1$ is a unit vector, one can state $\boldsymbol{u}_1^{\boldsymbol{T}}\boldsymbol{u}_1 = 1$. Using PCA, each data point x_n in the data set will be demonstrated by a scalar which is $\boldsymbol{u}_1^T x_n$, and the following equations, respectively, provide the mean and variance of the projected data:

$$\boldsymbol{u}_1^T \bar{x} = \boldsymbol{u}_1^T \frac{1}{N}\sum_{n=1}^{N} x_n \tag{8}$$

$$\frac{1}{N}\sum_{n=1}^{N}\{\boldsymbol{u}_1^T x_n - \boldsymbol{u}_1^T \bar{x}\}^2 = \boldsymbol{u}_1^T \boldsymbol{S}\boldsymbol{u}_1 \tag{9}$$

where $\boldsymbol{S}$ is the covariance matrix of the data defined by:

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T \tag{10}$$

According to the above equations, we should maximize the variance (equation 9) with respect to the $\boldsymbol{u}_1$ to find the best direction of the principal subspace. To constrain the above optimization problem, a Lagrange multiplier denoted by λ_1 should be considered in the objective equation (variance equation), adding the previously mentioned constraint $\boldsymbol{u}_1^{\boldsymbol{T}}\boldsymbol{u}_1 = 1$ to it as below:

$$\boldsymbol{u}_1^T \boldsymbol{S}\boldsymbol{u}_1 + \lambda_1(1 - \boldsymbol{u}_1^{\boldsymbol{T}}\boldsymbol{u}_1) \tag{11}$$

The maximum value of the variance can be determined by setting the derivative of the above equation to $\boldsymbol{u}_1$, equal to zero, and multiplying it by $\boldsymbol{u}_1^T$ from the left side. This would result in equation 12:

$$\boldsymbol{u}_1^T \boldsymbol{S}\boldsymbol{u}_1 = \lambda_1 \tag{12}$$

which states that the optimum direction (or the first principal component) can be determined by the eigenvector having the largest λ_1. We can incrementally increase the number of principal components ($M > 1$) by maximizing the variance of the projected data in all the directions orthogonal to the previous ones. A simple implementation of the PCA technique is demonstrated in Figure 6 for a synthesized 2-D dataset projected into a 1-D subspace. Different parameters discussed in this section can be observed in this figure for better understanding. In our framework, we reshape the input data ($D^{r\times m}$ or $T^{s\times h}$) from their original dimension (m or h) into 2-dimension, and we will show that this is useful to capture the data patterns within a unified 2-D space.

Outlier detection

In addition to point outliers discussed previously, the general behavior of traffic data may also change due to different reasons, such as public holidays, long-time faulty sensors, and rare events happening in the network. These would result in a few complete volume profiles that the B-spline smoothing cannot handle alone. For these cases, we have no choice but to remove the abnormal data as they do not genuinely represent the dominant behavior of any specific temporal factor. In this work, we intend to compare the data completely related to a specific pair of factors. Therefore, any collected data during a public holiday has an extra factor (being related to a holiday) and does not purely belong to its temporal feature. As a result, it cannot be considered the same way as the rest. The literature also extensively shows that abnormal observations can significantly change the performance of different models. To better illustrate this point, Figure 7 is provided here. In Figure 7 (a), K-means clustering with the number of clusters = 2 is applied on an uncleaned 2-D dataset, and consequently, outliers are considered as a separate cluster. However, we know that the blue points in Figure 7 (a) are not independent, strong patterns in the data and are just anomalies. Since we do not want them to change the performance of clustering algorithms, we will first decontaminate the reshaped data and then

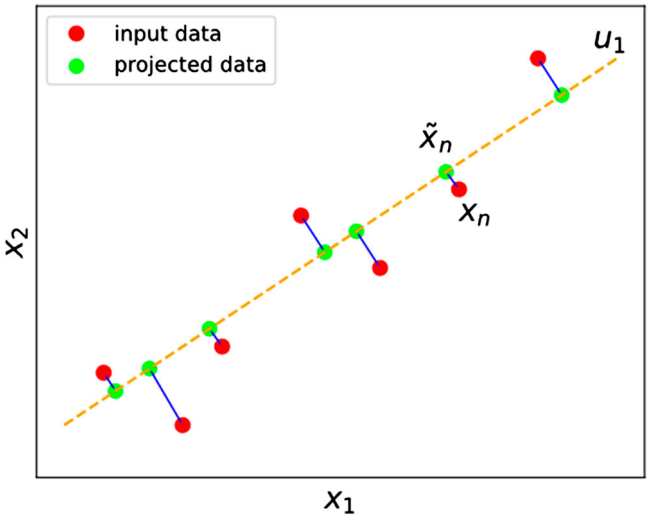


Figure 6. Dimension reduction of a 2-D dataset into 1-D using PCA.

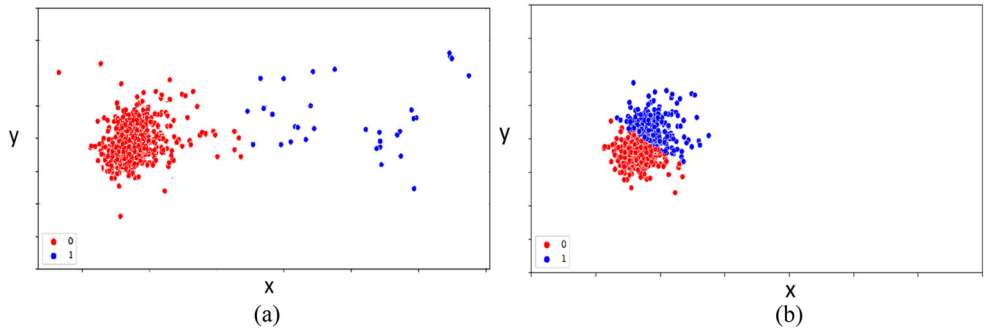


Figure 7. The effect of outliers applying clustering algorithms: K-means with #clusters = 2 applied on (a) data with outliers (b) cleaned data.

proceed with the clustering. In this manner, the clustering method will go for finding the best two clusters in the normal samples, as shown in Figure 7 (b).

For this aim, DBSCAN, with the help of KNN for hyperparameter selection, is applied before any pairwise clustering. DBSCAN (density-based spatial clustering of applications with noise) is a density-based and non-parametric clustering algorithm capable of detecting outliers in the data (Ester et al. 1996). It can be advantageous to apply DBSCAN-KNN for outlier detection in our methodology from different aspects. First, it works completely based on the distance and reachability of data samples. Theoretically, this aligns with our intention as we tend to eliminate data points considerably far from the clusters. Second, the hyperparameters of DBSCAN are challenging to set without any model implementation; however, with assistance from KNN, we show that it becomes completely straightforward to use DBSCAN within our methodology. Furthermore, it is shown in the literature that DBSCAN outperforms most other outlier detection and clustering algorithms when it comes to data with a low number of dimensions (Keogh and Lin 2005).

The basic concept behind this algorithm is to group very close data samples to each other and assign outlier labels to the data samples that are far from their neighbors. DBSCAN uses two main parameters to cluster the data: ε and *minPts*. ε is a distance measure specified before the clustering

algorithm's implementation to define reachability from one point to another point. *minPts* is the minimum number of points required to form a data cluster in a dense region. There is no need to specify the number of clusters in this algorithm since this approach determines the data clusters within a dataset solely based on ε and *minPts*. This is not the case in most other clustering approaches. The following steps are conducted in the DBSCAN clustering algorithm to assign cluster labels to the data:

- A data point is called a core point (p) if there are at least *minPts* number of data points within a specific distance of it (ϵ).
- q is a reachable data point from p if a path $p_1, \ldots, p_n$ is available with $p_1 = p$ and $p_n = q$, where every p_{i+1} is directly connected to p_i. The point here is that all the points on path $p_1, \ldots, p_n$ must be core points, with a potential exception for q.
- Data points not reachable from any other point are labeled as outliers in this algorithm.
- p forms a cluster with all other core or non-core points that are reachable from it.
- Non-core points included in any cluster form the edge, as there is no more point to be reached using them.

Although the main idea behind the DBSCAN algorithm is promising, selecting its hyperparameters, specifically the ε, is challenging. *minPts* (Minimum number of points per each cluster) can be determined by domain knowledge; however, we benefited from KNN (K Nearest Neighbors) algorithm to find the proper value for the ε (Schubert et al. 2017). In this algorithm, first, the Euclidean distance of each sample to its K^{th} nearest neighbor is calculated. Then, samples are sorted based on the derived distances. One candidate for the ε is the distance of the elbow (or knee) point (ε_{elb}) existing in this sorted set of samples (Satopaa et al. 2011). However, in this paper, we use $3 \times \varepsilon_{elb}$ instead of the ε in DBSCAN and experimentally show that it is more precise than the ε_{elb} using the data of different locations in different types of analyses. In this way, the problem of ε selection is changed to the selection of K in KNN algorithm, and we also show that the performance is not sensitive to the choice of the K value.

One should also be careful about 'super-long errors' in the datasets when applying our algorithm to the data of several links in a network. In some cases, long periods (like one or two months) might exist with messy data. Feeding such a dataset into our method without pre-screening would result in meaningless outcomes.

Pairwise clustering using K-means algorithm

The key idea of this research is to compare the reshaped traffic data related to pairwise temporal factors and determine a measure of dissimilarity. We will show in the result section that exploring data patterns reveals interesting and valuable points in this manner. We apply K-means clustering after the previously mentioned steps to achieve a dissimilarity measure for the data related to two distinct temporal factors. Then, we compare the results with the ground truth. Let $\{x_n^{'}\}_{n=1}^{N}$ be a reshaped data (from D to 2 dimensions) with no outlier derived from two specific temporal factors, and $\{z_n^{'}\}_{n=1}^{N}$ be the label of each data sample demonstrating its specific temporal factor. As we extracted the data of two specific factors for pairwise clustering, $z_n^{'}$ would be a binary value with 0 and 1 indicating the first and second temporal factors, respectively.

Our aim in this step is to partition the data $\{x_n^{'}\}_{n=1}^{N}$ into two groups to see whether they are well separated from each other in terms of their temporal factors. By applying the K-means clustering algorithm, a set of D-dimension vectors $\{\mu_k\}_{k=1}^{K}$ will be found ($K = 2$ and $D = 2$ in our case) such that the Euclidean distance of the $\{x_n^{'}\}_{n=1}^{N}$ data points to their closest μ_k is minimized. In this method, we assign a set of binary indicators $r_{nk} \in \{0, 1\}$ to each data point n to represent which one of the clusters k includes them. An objective function is then defined to be minimized by this method which is described below:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n' - \mu_k||^2 \tag{13}$$

The above optimization problem can be solved using a two-stage approach. First, a set of initial values are selected for μ_k, and r_{nk} is determined using the below equation:

$$r_{nk} = \begin{cases} 1 & if\ k = arg\ min_j ||x_n' - \mu_k||^2 \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Then, we optimize J given the calculated set of r_{nk}. Since equation 13 is a quadratic function given r_{nk}, one can minimize it by setting its derivative to μ_k equal to zero. The following equation can be derived as a result of this calculation for updating μ_k:

$$\mu_k = \frac{\sum_n r_{nk} x_n'}{\sum_n r_{nk}} \tag{15}$$

After calculating a new set of μ_k using the above formula, r_{nk} can be updated employing equation 13. This process is repeated until no significant change is observed in the value of J, where convergence happens. At this stage, μ_k would be the center of the created clusters, and r_{nk} demonstrates the final clustering labels of the dataset.

To achieve a dissimilarity score for the selected groups of data, a comparison should be made between the labels assigned by the K-means clustering algorithm and the real primary labels $\{z_n'\}_{n=1}^N$ expressing the temporal characteristics of the collected data. Since K is always equal to 2 in our research, a comparison between z_n' and r_{n1} or r_{n2} would represent how successful the K-means clustering algorithm is in separating the data samples of a pair of temporal factors in our dataset. Adjusted Rand Index (ARI) is utilized in this paper to compare z_n' and r_{n1}. To achieve ARI, a contingency table (Table 1) should be first determined to calculate equation 16. X_i and Y_i in Table 1 are two different sets of clusters, and n_{ij} denotes the number of elements in common between different clusters ($n_{ij} = |X_i \cap Y_j|$).

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}} \tag{16}$$

An ARI close to 1 implies clustering the data using the K-means algorithm reveals similar results to the ground truth (targeted temporal factors). For instance, in comparison between the reshaped data of Mondays and Tuesdays, we already know which data point belongs to which one of the groups (the targeted factors here are Monday and Tuesday). This label set, Monday or Tuesday, is the ground truth of data, and if K-means can clearly differentiate between these two groups, the ARI becomes equal to one. This explanation applies to any other pair of temporal factors considered in this study. On the other hand, an ARI close to 0 denotes that differentiation between the data with respect to the target labels is not possible for K-means. ARI will be used in the result section as a dissimilarity measure to investigate the data patterns.

Data

We utilized seven years and eight months of traffic volume data recorded by loop detectors in Melbourne, Australia, to demonstrate the applications of our proposed methodology in exploring the

Table 1. Contingency table for ARI calculation.

	Y_1	Y_2	...	Y_s	sums
X_1	n_{11}	n_{12}	...	n_{1s}	a_1
X_2	n_{21}	n_{22}	...	n_{2s}	a_2
	⋮	⋮	⋱	⋮	⋮
X_r	n_{r1}	n_{r1}	...	n_{rs}	a_r
sums	b_1	b_2	...	b_s	

patterns of big time series data. Several detectors are located on the road surface of the different intersections in Melbourne city, recording the behavior of vehicles in each direction. The data from some specific locations were selected based on the expert's opinion for this investigation. In our analysis, we tried to include data from multiple urban contexts (CBD areas, suburbs, highways, and streets).

The number of vehicles passing a specific way is collected by loop detectors every 15 min resulting in 96 data points for every single day (from 00:00–23:59). Data collection was also conducted from January 1st, 2014, to August 31st, 2021, for each location. The point is that loop detector data is not immune to outliers or missing values during such an extended period. Nevertheless, the selected locations in this study recorded a negligible number of missing values during the data collection (less than 1% of the total number of data points), and the approach to replacing them is explained previously. Temporal properties of the data, such as time, day, month, and year are also automatically recorded along with the volume data. One can also use our methodology to compare the data with regard to other characteristics like weather conditions or any other specific events.

To precisely mention any detector in the following section (result section), we refer to the data of each location using the name of two intersecting roadways (main roadway in the beginning) and a letter as an indicator of the direction (N, S, E, W). Therefore, interested readers can easily find the exact location of the detectors on the map. For example, Sydney Rd – Gaffney St (N) refers to the volume data collected in one direction (North Bound) of Sydney Road right before intersecting with Gaffney Street. A demonstration of this example is also depicted in Figure 8, where we can observe loop detectors of both directions in the Sydney Rd – Gaffney St intersection. In the following section, we will show important patterns observed in the data of different locations using our proposed method. We will recall different locations in the text with this format to be easily comprehended.

Results and discussion

Before discussing the applications of our proposed approach, we should mention the values selected for the hyperparameters of different methods introduced earlier. A summary of them is provided in Table 2. For B-spline fitting, a set of knots should be determined prior to curve fitting. The first and last data points are automatically considered knots in this method, and the number of interior knots, which can vary from 0 to 94, should be specified by the user. A very low and high number of interior knots cause underfitting and overfitting, respectively. Therefore, one should select a reasonable value

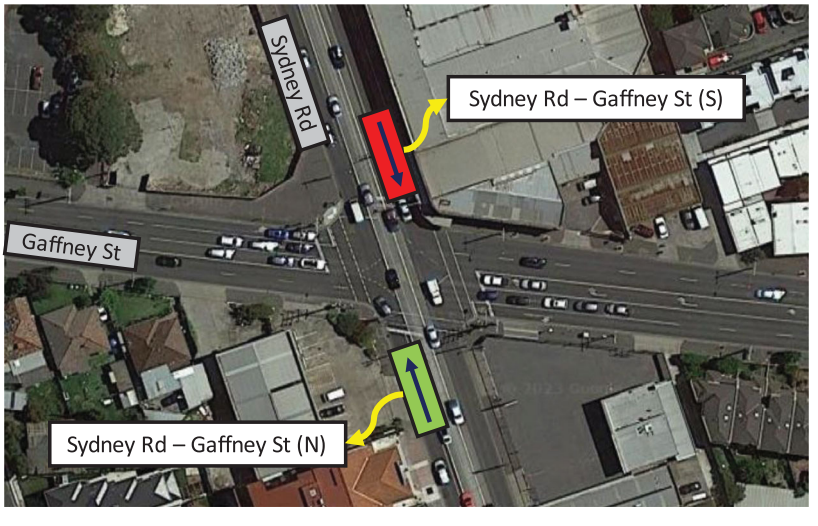


Figure 8. A sample depiction of loop detectors located at intersections in Melbourne.

Table 2. Derived hyperparameters for our proposed methodology.

Method	Parameter	Value
B-spline fitting	Number of Knots (t_i)	30 (uniform)
PCA	Number of components (M)	2
DBSCAN	ε	$3 \times \varepsilon_{elb}$
	$minPts$	30
	K_{KNN}	5
K-means	Number of clusters (K)	2

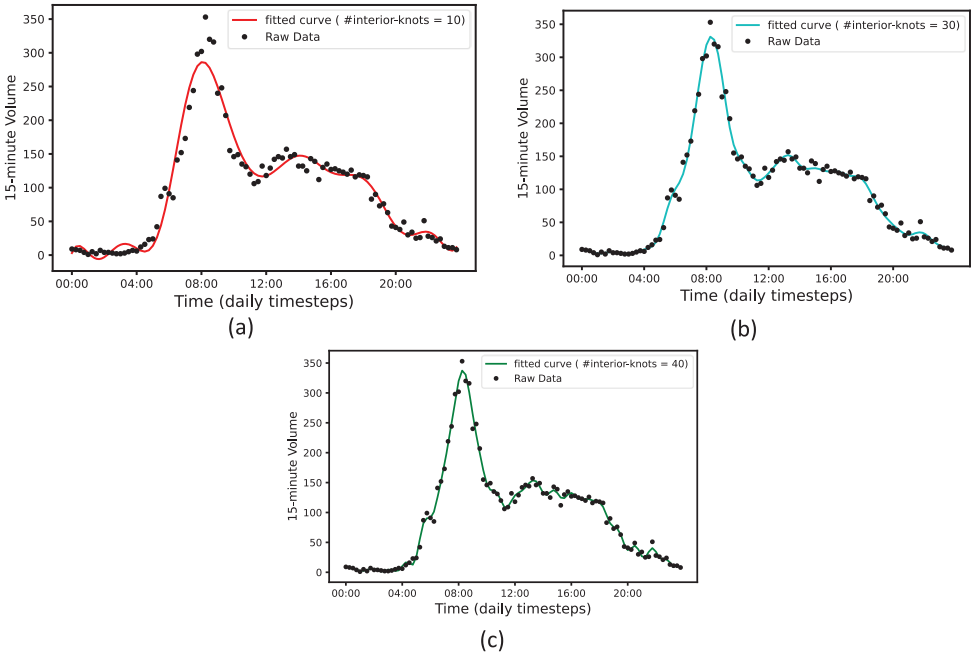


Figure 9. Effects of the number of interior knots on daily volume profiles (number of interior knots: (a) = 10, (b) = 30, (c) = 40).

for this parameter to fit an appropriate curve into the data. For better demonstration, fitted curves with different numbers of interior knots to a sample daily volume profile are provided in Figure 9. We used 30 as the number of interior knots (Figure 9 (b)) in our analysis since it is robust to noise and also better represents the underlying shape of the daily volume profiles despite the other lower and higher numbers (Figure 9 (a) and (c)). For PCA, we always use $M = 2$ to bring any data with different shapes into a 2-D space which is insightful and covers most of the variations among the data. Experiments with data from different locations showed an explained variance of more than 90 percent in most cases after applying PCA with $M = 2$ (with a few exceptions of 70 percent where faulty data was frequently observed). This means that the first two components are decent to capture the majority of the differences. Figure 10 (a) shows smoothed daily volume profiles of Mondays collected from Sydney Rd – Gaffney St (S) for six years (2014-2019). These profiles are reshaped into a 2-D latent space using PCA (Figure 10 (b)), and as it is observed, abnormal profiles are located far from the normal ones in this space. These abnormal samples should be detected using DBSCAN and removed before applying pairwise K-means clustering.

In DBSCAN, $minPts = 30$ is determined based on the minimum number of samples we expect to see in each group (background knowledge). $minPts = 30$ implies that we need at least 30 data points close to each other to consider them a unique cluster. The performance of DBSCAN is also not significantly affected by this parameter unless a very low (e.g. 5) or high value (e.g. 200) is assigned. We can use any value in a reasonable range according to the data size. Therefore, one can also get similar results using

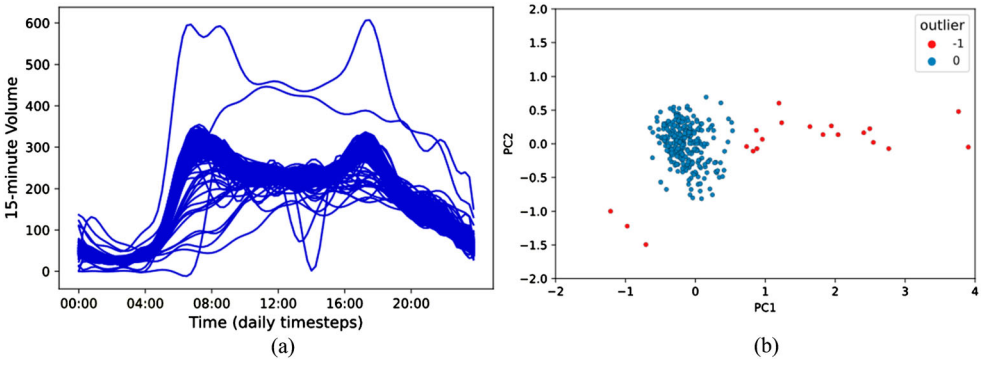


Figure 10. (a) Smoothed curves of daily volume profiles, and (b) their reshaped format using PCA. (Red points indicate potential outliers).

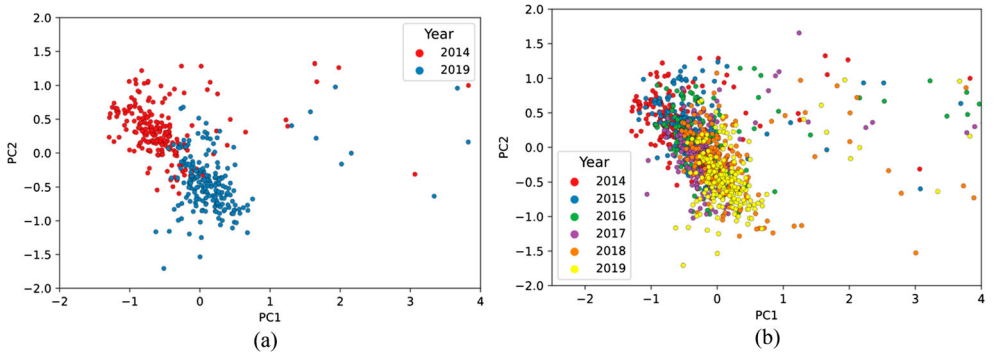


Figure 11. Pairwise comparison among data of 2014 and 2019 (a) vs. data of all years (2014–2019): location: Hoddle St – Victoria St (S).

$minPts = 40$ or $minPts = 50$. As we also mentioned in the Method section, we use $3 \times \varepsilon_{elb}$ determined by the KNN algorithm instead of ε. $K_{KNN} = 5$ is also chosen to perform the KNN algorithm. A detailed sensitivity analysis of the DBSCAN performance is discussed in Appendix 2.

As pairwise clustering is conducted in our analysis to compare the data of different temporal factors, the number of clusters in the K-means algorithm is always equal to 2. In our approach, K-means clusters the data related to every pair of temporal factors, and then its result is compared with reality using ARI. If a high ARI is calculated for each pair, we can conclude that the two data groups are totally different. An advantage of using ARI for analyzing K-means results is that it is not a binary number making strict decisions; however, it displays a number between 0 and 1, indicating the intensity of the differences.

In the following subsections, we will compare the patterns extracted from volume data of different locations and explore their variation with respect to the year, month, weekday, and time of the day. In each subsection, we will exclude the effect of other factors as much as possible to focus solely on the target factors using homogenous data. In the end, we also conduct a comparative study and show the strengths of pairwise clustering compared to the other clustering baselines.

Year analysis

In this section, our proposed methodology investigates the data patterns with respect to the year as a temporal factor. In other words, our aim in this section is to determine whether one specific year's data differs significantly from the others. We should note that to focus on annual variations, we just included Mondays to Thursdays and excluded the data for January and December in this analysis. An

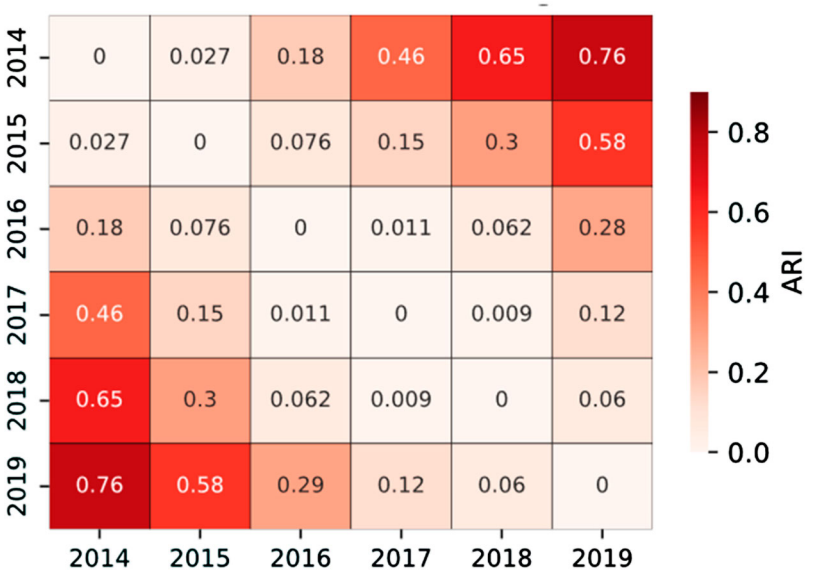


Figure 12. ARI matrix for year analysis of Hoddle St – Victoria St (S) (2014–2019).

essential point in this part is that annual data variations mostly happen gradually rather than abruptly. Therefore, a pairwise comparison of data related to different years is more revealing than observing the whole data together. Figure 11 shows the importance of pairwise comparison, which is overlooked by most previous literature investigations. Figure 11 (a) and (b) represent the volume profiles of every single day (collected from the Hoddle St – Victoria St loop detector) after pre-processing and being reshaped using the PCA method. In other words, every point in these figures (with two features of PC1 and PC2) depicts a reshaped daily volume profile with 96 features. As we can observe, when we consider the 2014 and 2019 data and exclude other years (Figure 11 (a)), two distinct clusters of data can be detected, indicating significant differences between the data of these two years. However, if we also include the data from other years (2015–2018), the differences are not clearly visible, and it is also impossible for clustering algorithms to detect these gradual changes. Although a gradual movement of each year's data is visible in Figure 11 (b), applying a clustering algorithm on these data does not detect this movement. Therefore, we will fail to make a valid conclusion.

The results of applying our method (pairwise clustering) on Hoddle St – Victoria St data are shown in Figure 12. The provided symmetric matrix in this figure shows the ARI of comparing every two data groups with respect to their years. As we can observe, our methodology can automatically extract the differences among the data considering their temporal factors (specifically the year in this case). It can also be inferred from this matrix that the data of two consecutive years are not significantly different; however, the data differs substantially in the long run. It should also be mentioned that as we rescaled the data before using PCA, the comparison here is just based on the shape rather than the magnitude. As we discussed previously, daily volume profiles can be normalized in any analysis, but one can also consider their magnitude by simply omitting data rescaling. Figure 13 demonstrates two rescaled sample volume profiles of Hoddle St – Victoria St, each collected in different years (2014 and 2019). Based on this figure, a different shape was recorded during 2019, which is the case for most 2019 profiles, according to Figure 11 (a).

The above analysis is also conducted for the data of other locations, and it is observed that these patterns vary from one location to another location. For instance, the ARI matrix of Exhibition St – Lonsdale St (S), a location in the CBD area, is shown in Figure 14, and no significant change is recorded for the data of different years in this specific location. Therefore, this should be remarked that long-term

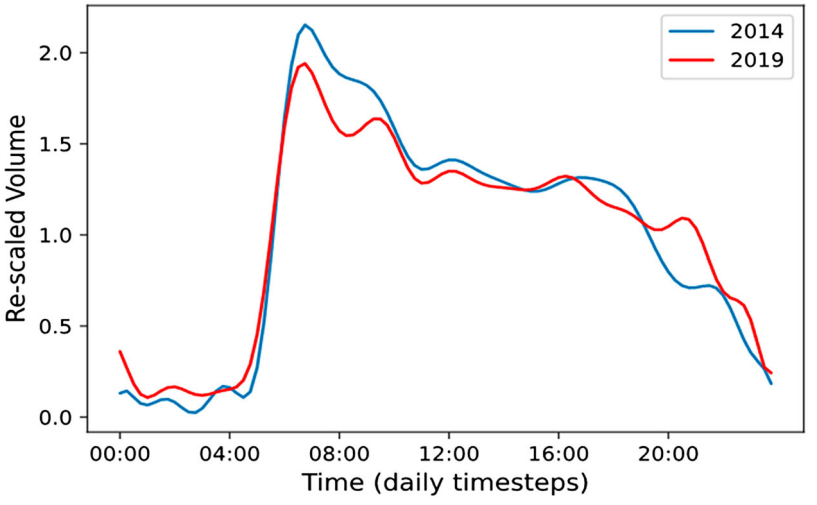


Figure 13. Two re-scaled volume profile samples from 2014 and 2019 collected by Hoddle St – Victoria St (S) loop detector.

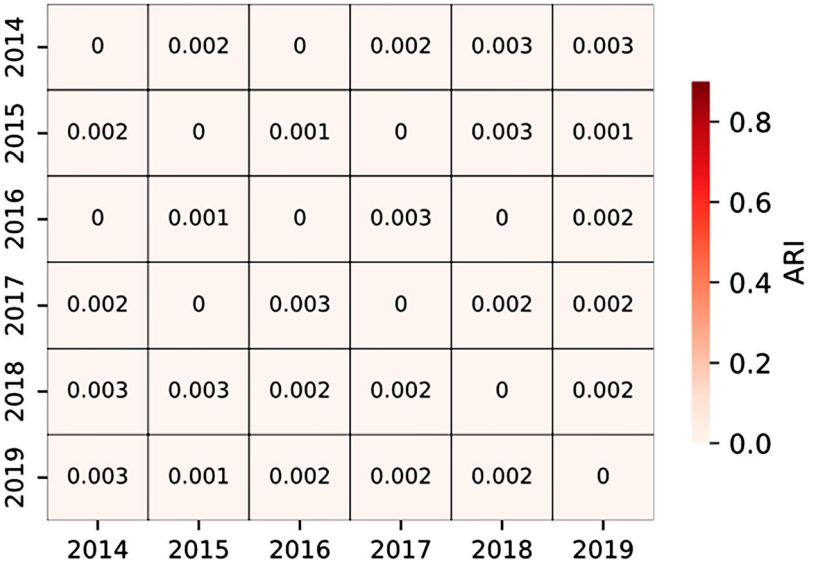


Figure 14. ARI matrix for year analysis of Exhibition St – Lonsdale St (N) (2014–2019).

historical data may not be homogenous, and our results highly recommend that one must first test the differences of big historical data prior to any model fitting or training. Specifically, it is a common belief that more data always empowers the predictive models (specifically learning-based models). However, the underlying nature of data must also be unique in the whole training set, and significant long-term variations can degrade the performance. Benefiting from the ARI matrices generated in this analysis, traffic experts can quickly scan the extent of dissimilarities in long historical data and then decide the best time scope in each location to be used in further analysis.

Furthermore, another experiment is conducted in this section to show our method's ability to explore the differences in volume data after and before COVID-19. For this aim, we considered the data from all the selected locations. Reshaped (2-D) daily volume profiles of each location were joined together, and again PCA was applied to reduce the dimension from 31×2 (number of locations $\times$ 2)

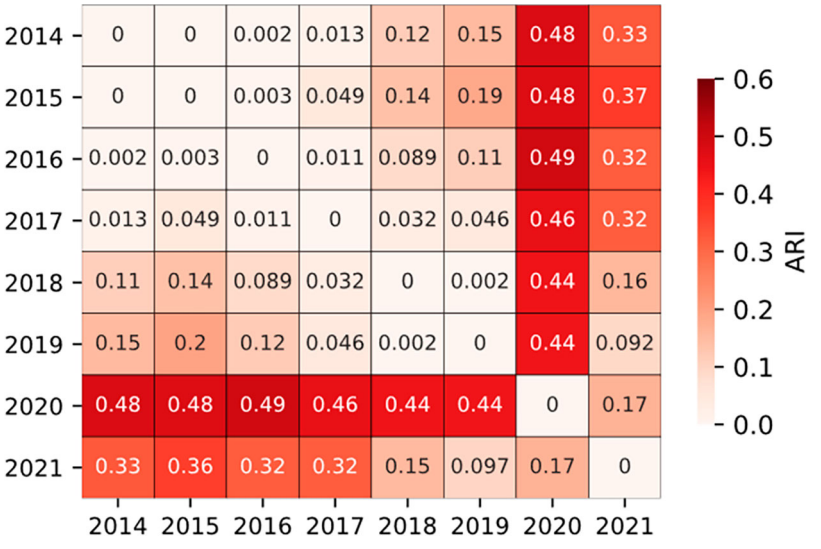


Figure 15. ARI matrix for year analysis of all available locations to show the effects of COVID-19 (2014–2021).

Table 3. School holidays in Melbourne.

School Holidays	Start	Finish
First Day of School	28 Jan	
Term 1 Holidays (Autumn)	2 Apr	18 Apr
Term 2 Holidays (Winter)	26 Jun	11 Jul
Term 3 Holidays (Spring)	18 Sep	3 Oct
Term 4 Holidays (Summer)	18 Dec	30 Jan

to a new 2-D latent space. Every point on the new latent space was a representation of daily volume profiles in all of the locations. Figure 15 shows the ARI matrix of this experiment, and as we can see, the data is clearly different before and after the COVID-19 pandemic. ARI for 2020 is also higher than in 2021 since more lockdowns were in place in Melbourne during 2020. Some differences are also detected between the data of 2019–18 and 2014-15. However, as we clarified previously, the exact differences should be analyzed for each location.

Month of the year analysis

Daily volume profiles of each location can also be explored with respect to the months of the year. Figure 16 shows the ARI matrix of two specific locations considering monthly data variations. These two figures are the most frequent patterns observed in the data of different locations, and we should note that the data used for them in this section was homogenous in terms of the year and the day of the week. We focused only on workdays and included the data with no significant annual pattern in this analysis. The main point inferred from these figures is that January and December (the first and the last month in each year) are more different from the other months in these locations. However, there are also some exceptions to this observation. For instance, the first month of the year (January) is significantly different from the other months in Figure 16 (a), but this happened in Figure 16 (b) with lower intensity. Furthermore, some months like April, July, and September showed lower conflict with January in Figure 16 (a), while this is true in Figure 16 (b) between December and other months like January, February, March, and November. These patterns and exceptions are because of the main characteristics of the traffic behavior in these locations. For example, Springvale Rd – Waverley Rd intersection in Melbourne is close to a primary school. The patterns observed in Figure 16 (a) are

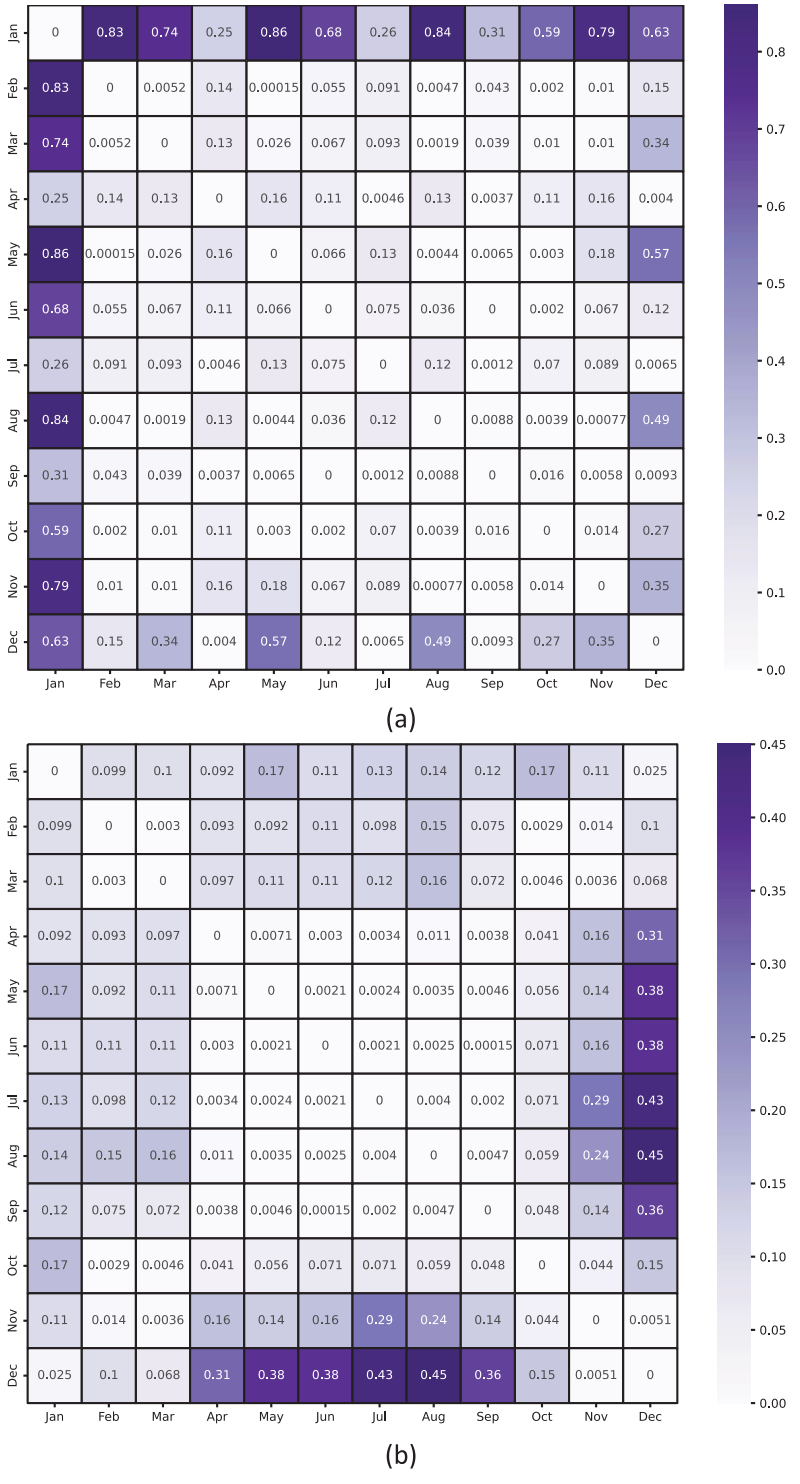


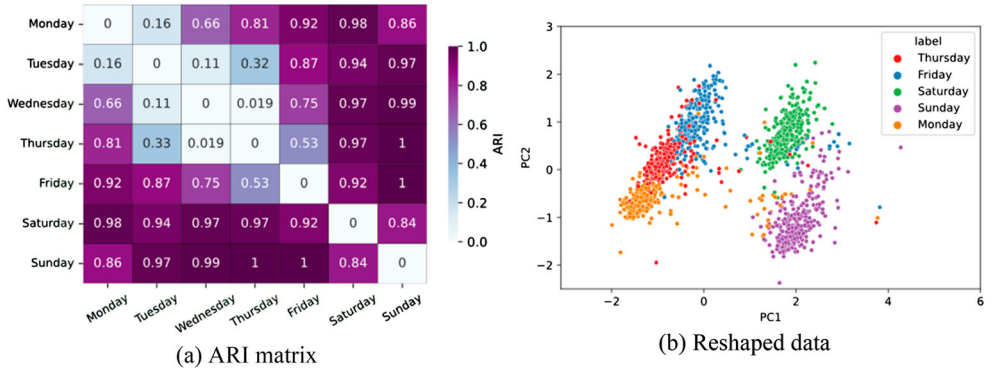
Figure 16. Analysis of month for two different locations: (a) Springvale Rd – Waverley Rd (N) vs. (b) Sydney Rd – Gaffney St (N).

adjustable with school holidays in this area. School holidays during 2021 in Melbourne are provided in Table 3, which is also approximately the same as the previous years (2014-2019). As we can see, January is the only month in which Melbourne schools are off, and there are also some other months, like August, July, and September, being partially off. These holidays affect the traffic volume data, and the patterns are depicted in Figure 16 (a). A seasonal pattern is also detected for Sydney Rd – Gaffney St data in Figure 16 (b). Summer starts in Melbourne from October to March. As we can see, a gradual variation is recorded, moving from cold weather (April to September) to warm weather (October to March) in this location, with the highest temperature in December.

Results in this section indicate that monthly traffic data variations significantly differ from location to location. Our methodology can detect them without information about land use or pre-knowledge of traffic behavior. Moreover, the extracted variations in this analysis also show the importance of pattern exploration before traffic modeling because these variations affect the performance and should be considered in the model.

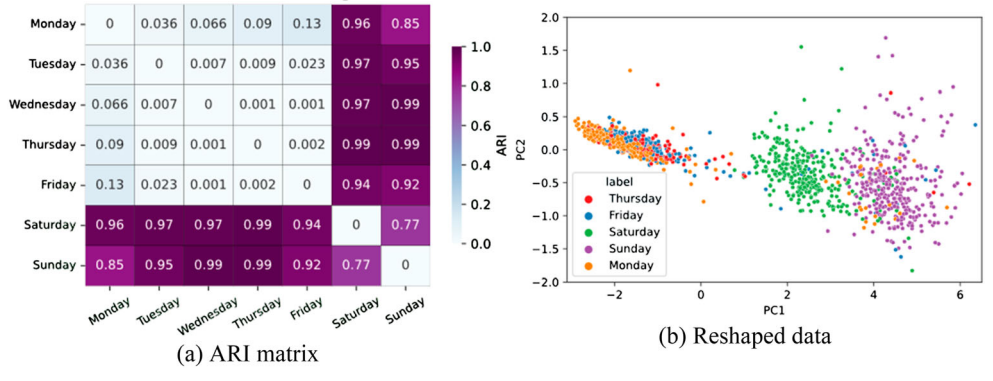
Day of the week analysis

Traffic behavior varies on different days of the week, and a well-known expression in this regard is that workdays are different from the weekends. In this subsection, we analyzed the volume data of different locations in Melbourne employing our methodology. The results indicate other differences among the daily volume profiles of different locations, which are worth exploring before further analysis.



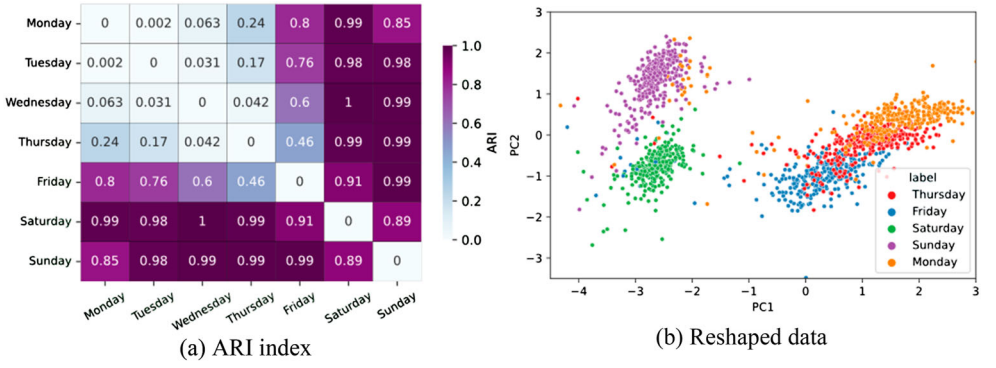
(a) ARI matrix (b) Reshaped data

Figure 17. Analysis of day for Victoria St – Elizabeth St (E).



(a) ARI matrix (b) Reshaped data

Figure 18. Analysis of day for Lorimer St – Salmon St (W).



(a) ARI index (b) Reshaped data

Figure 19. Analysis of day for Melville Rd – Moreland Rd (S).

Figures 17–19 show the results of the day of the week analysis for three different locations as well as some demonstrations of their reshaped data. Firstly, based on these results, one can accept significant differences between the workdays and the weekends for each location (as an ARI close to one is calculated for the weekends). However, another interesting point here is that even Saturdays and Sundays (weekend days) are also different from in terms of traffic behavior. Furthermore, other meaningful differences are also detectable between different days of the week, varying from one location to another. For instance, in the Victoria St – Elizabeth St intersection (Figure 17), a gradual movement for different workdays is recorded in the latent space (Figure 17 (b)) that enables the K-means clustering algorithm to detect some extra differences within them using a pairwise manner. In this location, Mondays and Fridays are significantly different from the other workdays; however, this is not the case in Figures 18 and 19. In Lorimer St – Salmon St intersection, traffic behavior is almost the same within the workdays (just a low difference is observed between Mondays and Fridays), and in Melville St, just Fridays possess different data among the other workdays.

These patterns can be justified with some characteristics of these roadways to show that they are meaningful and reasonable. In particular, the Victoria St – Elizabeth St intersection is located near a big shopping center (Queen Victoria Market), and this center is totally closed on Mondays and partially on Wednesdays. As it is observed, Mondays are recognized in Figure 17 (a) as a unique day, but this is not the case for Wednesdays. A few regular shopping stores also surround Lorimer St without any specific recreational facility, which leads to a Friday similar to the other workdays. On the other hand, Melville Rd – Moreland Rd intersection connects two typical major roadways, which help people with their work trips. Friday is the day before the holidays, and some leave work sooner than usual and go home. This can be why Fridays stand as a different day in most major roadways data. We should note here that the above reasons are stated only for justification and reasoning. They are not definitely the only factor affecting the traffic volume data, and there may be some other causes in these locations impacting the traffic volume data. Nevertheless, our approach in this paper is able to automatically detect the traffic behavior in a data-driven way without any prior information on the location properties.

Time of the day analysis

Daily volume profiles are different in each location. Given a historical set of volume data, one can also group different times of the day utilizing the proposed approach in this paper. Recognizing times

of the day with similar traffic volume values can be useful for different aims, such as missing data imputation and feature reduction of data. Since the traffic data is collected every T minutes ($T = 15$ minutes in this study), the feature space of the time of the day becomes equal to $24 \times \frac{60}{T} (= 96)$. Therefore, it sometimes becomes necessary to reduce this feature dimension for further analysis like predictive modeling. For this purpose, we reshaped the values recorded for each time slot during a week (only homogenous days) into a 2-D latent space using PCA. Then we conducted our pairwise clustering approach to construct an ARI matrix for every location. In other words, each x_n in this analysis is the value of traffic volume recorded during a week (different days are excluded). For instance, if it is observed that Mondays to Thursdays are similar in a specific location, $\{x_n\}_{n=1}^{N}$ becomes a 4-dimensional matrix that will be fed into the PCA. Each dimension or feature in x_n involves the data of one specific time interval (e.g. 2:00–2:15 PM) from each day of the week (Monday to Thursday). Different samples (n) are extracted from all other times and weeks included in historical data (this concept is also illustrated in Figure 4). Results for two different locations in our dataset, along with their daily volume profiles, are illustrated in Figures 20 and 21. As we can see, the ARI matrix for each location becomes a 96×96 matrix, with each cell demonstrating the degree of dissimilarity between the data of each pair of time (15 min intervals). For better representation, heatmaps of these matrices are shown in Figures 20 and 21, and any yellow cell in these figures indicates that the data of a specific pair of times are similar and indifferent based on our method.

Looking solely into the daily volume profiles of each location (right side of the figures) to determine similar time slots in a day is time-consuming and also imprecise and controversial in big datasets. For instance, one may claim that the data in Figure 20 (right side) does not significantly change from 10 AM to 4 PM, and one may state that the data is totally different in each time slot of this period. This paper's methodology can also be used as a baseline to overcome this issue. ARI matrices derived for different locations in this study can be employed to determine similar time intervals during a day automatically.

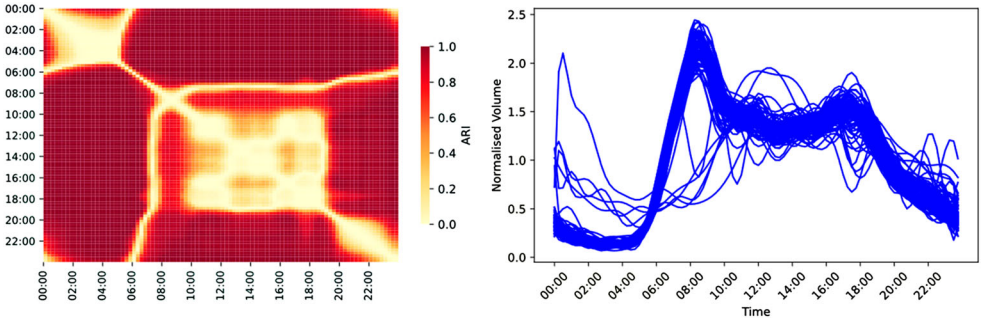


Figure 20. Time of the day analysis for Lonsdale St – William St (N). Left: ARI matrix. Right: homogenous daily volume profiles.

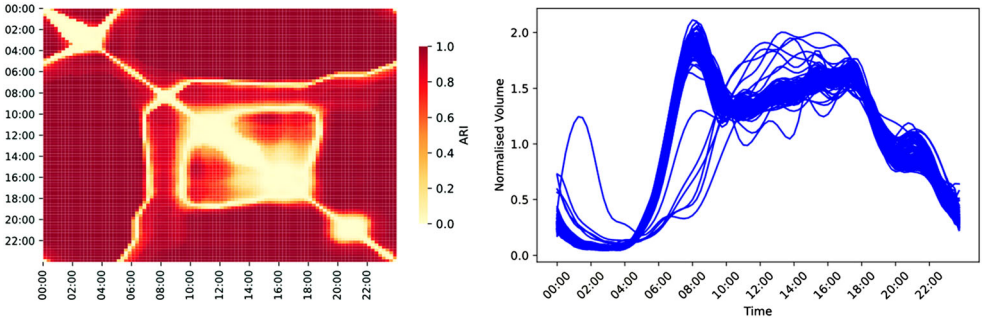


Figure 21. Time of the day analysis for Springvale St – Waverley St (N). Left: ARI matrix. Right: homogenous daily volume profiles.

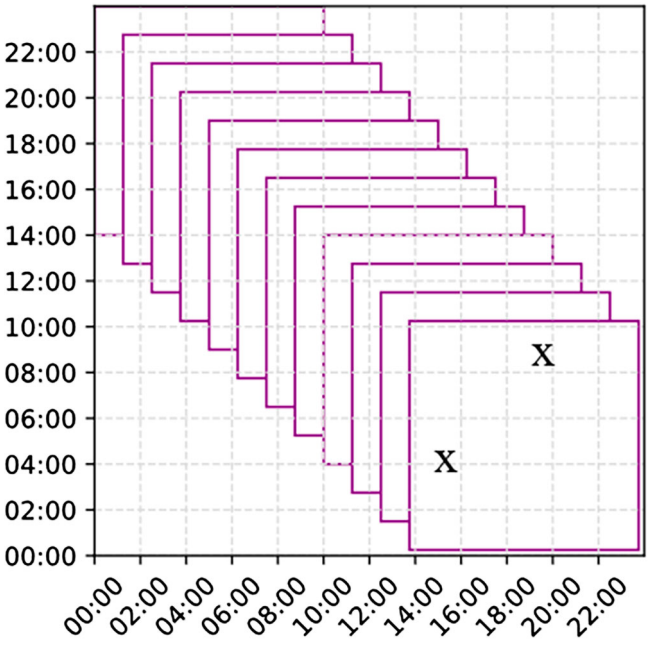


Figure 22. A demonstration of sliding windows used for detecting homogenous times (different values for x is tested).

Every yellow square (by square, we mean a set of cells connected to each other) on the diagonal of the ARI matrix shows pairs of time intervals with indifferent volume data. To avoid using personal judgments for detection of these squares, we employed 2-D sliding windows with variable length (x) to be rolled on ARI matrices (Figure 22). The exact algorithm for detecting these areas is discussed in Appendix 3. In this algorithm, we assign positive scores to the yellow-colored cells and very low negative scores to the red cells in each ARI matrix. Then windows with different lengths ($x = 2$–96) were rolled diagonally from the top left of each ARI matrix to the bottom right, and the sum of the scores within each window was calculated. Every square in the matrices was detected by sorting the windows based on their total scores and removing the overlapped windows with lower scores. A demonstration of the detected areas using sliding windows for Lonsdale St – William St and Springvale St – Waverley St intersections is provided in Figure 23. We can observe that three major time intervals with no significant variations were detected during the day for these two locations. This is not easily detectable by simple visualizations on the right side.

By this approach, one can also detect peak hours during a day (without using human judgment) employing the average volume of detected time intervals. In other words, if we compare the average volume of intervals in the new reduced feature space, we can mark the interval with the highest value as a peak interval. For example, the highest value (peak volume) can be observed for Springvale St – Waverley St (N) from 8:00 AM to 9:15 AM, as indicated in Figure 23.

Comparative study

According to the previous subsections, several types of information can be extracted from volume data of different locations by applying the pairwise clustering algorithm. To the best of our knowledge, no similar methodology from the literature can be systematically used for analyzing every type of temporal factor in such a way. However, to better demonstrate the preciseness of our method, clustering baselines are selected in this part and tested with Victoria St – Elizabeth St (E) data to analyze the day of the week patterns within it (like Figure 17 in our results). We compared daily volume

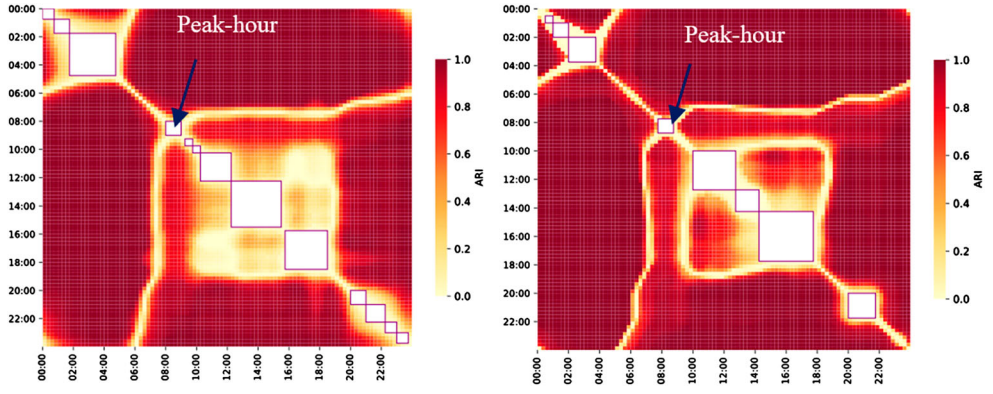


Figure 23. Detected areas showing time slots with homogenous data (Springvale St – Waverley St (N) vs. Lonsdale St – William St (N)).

Table 4. Suggested number of clusters applying different clustering algorithms on different days of the week in Victoria St – Elizabeth St (E).

	Silhouette coefficient for different number of clusters (components)						Suggested # of clusters
Method	2	3	4	5	6	7	
Euclidean Distance + K-means	**0.51**	0.33	0.34	0.34	0.34	0.24	2
DTW + K-means	**0.49**	0.45	0.47	0.32	0.34	0.33	2
Soft DTW + K-means	**0.50**	0.33	0.23	0.23	0.24	0.24	2
PCA + K-means	**0.65**	0.63	0.57	0.57	0.58	0.51	2
PCA + Agglomerative clustering	**0.65**	0.62	0.55	0.56	0.49	0.49	2
PCA + GMM	**0.60**	0.48	0.46	0.47	0.47	0.45	2
Pairwise clustering (our method)	Up to four dissimilarities are detected (Figure 17)						

profiles using similarity metrics named Euclidean distance, Dynamic Time Warping (DTW), and soft DTW (instead of using PCA). Then we applied K-means clustering as in Tavenard et al. (2020) to differentiate between different traffic data patterns. Since our aim is to know how many different patterns exist within the data, we adopted Silhouette Coefficient with variable numbers of clusters (Rousseeuw 1987) to find the optimum number of clusters. In another setting, we also used PCA to reduce the dimension of data. However, instead of pairwise clustering, we benefited from general clustering algorithms, like K-means, Agglomerative clustering, and Gaussian Mixture Model (GMM), where the whole data is passed into the model at once. Table 4 shows the Silhouette Coefficient achieved by applying these different settings with various numbers of clusters (or numbers of components in GMM). As we can see, the number of clusters $= 2$ is suggested by all methods as it has resulted in a higher Silhouette Coefficient. However, according to Figure 17, up to four different dissimilarities (Saturdays, Sundays, Fridays, and even Thursdays to some extent) can be detected with their own specified intensities. In other words, since the aim is to find a number-of-clusters which result in well-separated and internally intense groups of data (based on the definition of the Silhouette Coefficient), these models may neglect any particular pattern existing within the generally similar volume profiles. Applying PCA and pairwise clustering can provide a more detailed and accurate pattern exploration of data in this regard.

Another point in Table 4 is also the high marginal contribution of PCA in increasing the Silhouette Coefficients. Methods containing the PCA recorded higher cluster quality measures compared to the time series clustering algorithms. This shows the ability of PCA to reshape the data in a way that maximizes the variation. We also benefited from this ability in our proposed methodology to represent the data thoroughly for clustering and outlier detection.

Summary and conclusion

Pattern mining of traffic data considering temporal factors is advantageous for traffic modeling and analysis. It is highly suggested by the literature to explore the variations of data before developing practical and novel methodologies. In this work, we propose a hybrid approach that is able to automatically analyze the traffic data and report the intensity of dissimilarities existing among every pair of temporal factors. It is efficient and lightweight because it uses PCA, DBSCAN, and K-means. In cases where only large raw traffic data is available, traffic experts can also instantly apply it to reveal detailed information about the patterns of data. In this approach, B-spline fitted traffic volume data is first transferred into a 2-D latent space using PCA, and then abnormal samples are excluded using the DBSCAN clustering algorithm. Second, pairwise clustering using K-means is applied to derive Adjusted Rand Index matrices (which show the degree of dissimilarity of any data pairs). Using the ARI matrices, multiple analyses are carried out to elicit variations of traffic data regarding the time of the day, day of the week, month of the year, and the year itself. Seven years and eight months of volume data from different locations in Melbourne were used in our experiment. Some locations were found to have a different volume profile shape in the long run, while others demonstrated unique patterns in previous years. Moreover, monthly patterns and daily patterns within a week were found to be diverse as a result of surrounding land uses around each location. Daily variations were also analyzed, showing that it might be tricky to visually detect time intervals with similar volume data within a day.

The main findings of this paper can be summarized as follows:

- Traffic behavior may not be consistent with pre-existing knowledge in every location, and attention needs to be paid to data patterns regarding temporal factors before any further model developments.
- Pairwise clustering with respect to temporal factors is more illuminating than previous general clustering approaches in the literature since it revealed more insights in this study.
- Our pattern mining approach showed to be a proper feature reduction algorithm for temporal factors, specifically the time of the day. It has the capability of extracting time intervals with indifferent values.
- An automatic sliding window approach is also developed to detect each location's homogenous times during the day. This would be helpful to avoid human judgment in daily profile segmentation.

A comparison between PCA and other dimension reduction approaches, such as autoencoders, is highly suggested for future directions. PCA is robust to the amount of data and does not need any pre-training. Nevertheless, it is valuable to compare its results with other recent dimension reduction methods to see whether there are significant differences between them. Pattern analysis of traffic data using a higher number of principal components ($M > 2$) is also worth exploring, as clustering the data in higher dimensions has its own challenges and advantages. Moreover, comparing the performance of predictive machine learning models before and after feature reduction of temporal factors using the proposed methodology can also be another research direction revealing the impact of temporal factors' feature space on prediction results.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Iman Taheri Sarteshnizi http://orcid.org/0000-0002-7798-8788
Saeed Asadi Bagloee http://orcid.org/0000-0001-6078-6314

References

Bartels, R., B. A. Barsky, and J. C. Beatty. 1987. An Introduction to Splines for Use in Computer Graphics and Geometric Modeling.

Bishop, C. M., and N. M. Nasrabadi 2006. *Pattern Recognition and Machine Learning*. 1st ed., Vol. 4, 738. New York: Springer-Verlag.

Boquet, G., A. Morell, J. Serrano, and J. L. Vicario. 2020. "A Variational Autoencoder Solution for Road Traffic Forecasting Systems: Missing Data Imputation, Dimension Reduction, Model Selection and Anomaly Detection." *Transportation Research Part C: Emerging Technologies* 115: 102622. doi:10.1016/J.TRC.2020.102622.

Chen, X., H. Chen, Y. Yang, H. Wu, W. Zhang, J. Zhao, and Y. Xiong. 2021. "Traffic Flow Prediction by an Ensemble Framework with Data Denoising and Deep Learning Model." *Physica A: Statistical Mechanics and Its Applications* 565: 125574. doi:10.1016/J.PHYSA.2020.125574.

Crawford, F., D. P. Watling, and R. D. Connors. 2017. "A Statistical Method for Estimating Predictable Differences Between Daily Traffic Flow Profiles." *Transportation Research Part B: Methodological* 95: 196–213. doi:10.1016/J.TRB.2016.11.004.

Emami, A., M. Sarvi, and S. Asadi Bagloee. 2019. "Using Kalman Filter Algorithm for Short-Term Traffic Flow Prediction in a Connected Vehicle Environment." *Journal of Modern Transportation* 27 (3): 222–232. doi:10.1007/S40534-019-0193-2/FIGURES/5.

Emami, A., M. Sarvi, and S. A. Bagloee. 2021. "Network-Wide Traffic State Estimation and Rolling Horizon-Based Signal Control Optimization in a Connected Vehicle Environment." *IEEE Transactions on Intelligent Transportation Systems*, doi:10.1109/TITS.2021.3059705.

Ester, M., H.-P. Kriegel, J. Sander, and X. Xu. 1996. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise." In *Kdd*, edited by E. Simoudis, J. Han, and U. Fayyad, Vol. 96, 226–231. Portland, OR: AAAI Press. www.aaai.org.

Guardiola, I. G., T. Leon, and F. Mallor. 2014. "A Functional Approach to Monitor and Recognize Patterns of Daily Traffic Profiles." *Transportation Research Part B: Methodological* 65: 119–136. doi:10.1016/J.TRB.2014.04.006.

Habtemichael, F. G., and M. Cetin. 2016. "Short-Term Traffic Flow Rate Forecasting Based on Identifying Similar Traffic Patterns." *Transportation Research Part C: Emerging Technologies* 66: 61–78. doi:10.1016/J.TRC.2015.08.017.

Hellinga, B., and M. Van Aerde. 1998. *Estimating Dynamic O-D Demands for a Freeway Corridor Using Loop Detector Data*. Proceedings of the Canadian Society for Civil Engineering 1998 Annual Conference held in Halifax, Nova Scotia, Volume IVb, 185-197. ISBN 0-921303-86-6

Hitchcock, D. B., J. G. Booth, and G. Casella. 2007. "The Effect of Pre-Smoothing Functional Data on Cluster Analysis." *Journal of Statistical Computation and Simulation* 77 (12): 1043–1055. doi:10.1080/10629360600880684.

Horowitz, A. J., F. T. Creasey, R. M. Pendyala, M. Chen, and C. D. M. Smith. 2014. "Analytical Travel Forecasting Approaches for Project-Level Planning and Design." *TRB's National Cooperative Highway Research Program (NCHRP) Report* 765: 307. doi:10.17226/22366.

Jiang, X., and H. Adeli. 2004. "Wavelet Packet-Autocorrelation Function Method for Traffic Flow Pattern Analysis." *Computer-Aided Civil and Infrastructure Engineering* 19 (5): 324–337. doi:10.1111/J.1467-8667.2004.00360.X.

Kalinic, M., and J. M. Krisp. 2019. "Fuzzy Inference Approach in Traffic Congestion Detection." *Annals of GIS* 25 (4): 329–336. doi:10.1080/19475683.2019.1675760.

Keogh, E., and J. Lin. 2005. "Clustering of Time-Series Subsequences is Meaningless: Implications for Previous and Future Research." *Knowledge and Information Systems 2004* 8 (2): 154–177. doi:10.1007/S10115-004-0172-7.

Leiser, N., and M. Yildirimoglu. 2021. "Incorporating Congestion Patterns Into Spatio-Temporal Deep Learning Algorithms." *Transportmetrica B: Transport Dynamics* 9 (1): 622–640. doi:10.1080/21680566.2021.1922320.

Li, P. L., and J. M. Chiou. 2020. "Functional Clustering and Missing Value Imputation of Traffic Flow Trajectories." *Transportmetrica B: Transport Dynamics*, 1–21. doi:10.1080/21680566.2020.1781706.

Liu, Y., Z. Liu, and R. Jia. 2019. "DeepPF: A Deep Learning Based Architecture for Metro Passenger Flow Prediction." *Transportation Research Part C: Emerging Technologies* 101: 18–34. doi:10.1016/J.TRC.2019.01.027.

Liu, Z., and R. Stern. 2021. "Quantifying the Traffic Impacts of the COVID-19 Shutdown." *Journal of Transportation Engineering, Part A: Systems* 147 (5): 04021014. doi:10.1061/JTEPBS.0000527.

Lv, Y., Y. Duan, W. Kang, Z. Li, and F. Y. Wang. 2015. "Traffic Flow Prediction with Big Data: A Deep Learning Approach." *IEEE Transactions on Intelligent Transportation Systems* 16 (2): 865–873. doi:10.1109/TITS.2014.2345663.

Ma, D., X. Song, and P. Li. 2021. "Daily Traffic Flow Forecasting Through a Contextual Convolutional Recurrent Neural Network Modeling Inter- And Intra-Day Traffic Patterns." *IEEE Transactions on Intelligent Transportation Systems* 22 (5): 2627–2636. doi:10.1109/TITS.2020.2973279.

Morita, H., S. Nakamura, and Y. Hayashi. 2020. "Changes of Urban Activities and Behaviors Due to COVID-19 in Japan." *SSRN Electronic Journal*, doi:10.2139/SSRN.3594054.

Parsa, A. B., A. Movahedi, H. Taghipour, S. Derrible, and A. (Kouros) Mohammadian. 2020. "Toward Safer Highways, Application of XGBoost and SHAP for Real-Time Accident Detection and Feature Analysis." *Accident Analysis and Prevention* 136 (October 2019): 105405. doi:10.1016/j.aap.2019.105405.

Pascale, A., F. Deflorio, M. Nicoli, B. Dalla Chiara, and M. Pedroli. 2015. "Motorway Speed Pattern Identification from Floating Vehicle Data for Freight Applications." *Transportation Research Part C: Emerging Technologies* 51: 104–119. doi:10.1016/J.TRC.2014.09.018.

Qu, L., J. Lyu, W. Li, D. Ma, and H. Fan. 2021. "Features Injected Recurrent Neural Networks for Short-Term Traffic Speed Prediction." *Neurocomputing* 451: 290–304. doi:10.1016/J.NEUCOM.2021.03.054.

Rakha, H., and M. Aerde. 1995. "Statistical Analysis of Day-to-Day Variations in Real-Time Traffic Flow Data." *Transportation Research Record* 1510. http://onlinepubs.trb.org/Onlinepubs/trr/1995/1510/1510-004.pdf.

Rousseeuw, P. J. 1987. "Silhouettes: A Graphical aid to the Interpretation and Validation of Cluster Analysis." *Journal of Computational and Applied Mathematics* 20 (C): 53–65. doi:10.1016/0377-0427(87)90125-7.

Sarvi, M., S. Asadi, and S. Van Uytsel. 2021. "New Fixes for Old Traffic Problems: Connected Transport Systems and AIMES." *Perspectives in Law, Business and Innovation*, 185–196. doi:10.1007/978-981-15-9255-3_9.

Satopaa, V., J. Albrecht, D. Irwin, and B. Raghavan. 2011. "Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior." *Proceedings - International Conference on Distributed Computing Systems*, 166–171. doi:10.1109/ICDCSW.2011.20.

Schubert, E., J. Sander, M. Ester, H. P. Kriegel, and X. Xu. 2017. "DBSCAN Revisited, Revisited: Why and How Should (Still) Use DBSCAN." *ACM Transactions on Database Systems (TODS)* 42 (3), doi:10.1145/3068335.

Song, X., W. Li, D. Ma, D. Wang, L. Qu, and Y. Wang. 2018. "A Match-Then-Predict Method for Daily Traffic Flow Forecasting Based on Group Method of Data Handling." *Computer-Aided Civil and Infrastructure Engineering* 33 (11): 982–998. doi:10.1111/MICE.12381.

Soriguera, F. 2012. "Deriving Traffic Flow Patterns from Historical Data." *Journal of Transportation Engineering* 138 (12): 1430–1441. doi:10.1061/(ASCE)TE.1943-5436.0000456.

Stathopoulos, A., and M. Karlaftis. 2001. "Temporal and Spatial Variations of Real-Time Traffic Data in Urban Areas." *Transportation Research Record* 1768: 135–140. doi:10.3141/1768-16.

Tavenard, R., J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, et al. 2020. "Tslearn, A Machine Learning Toolkit for Time Series Data." *Journal of Machine Learning Research* 21 (1): 4686–4691. http://jmlr.org/papers/v21/20-091.html.

Vaze, V., C. Antoniou, Y. Wen, and M. Ben-Akiva. 2009. "Calibration of Dynamic Traffic Assignment Models with Point-to-Point Traffic Surveillance." *Transportation Research Record* 2090: 1–9. doi:10.3141/2090-01.

Wang, J., and Q. Shi. 2013. "Short-term Traffic Speed Forecasting Hybrid Model Based on Chaos–Wavelet Analysis-Support Vector Machine Theory." *Transportation Research Part C: Emerging Technologies* 27: 219–232. doi:10.1016/J.TRC.2012.08.004.

Yang, B., S. Sun, J. Li, X. Lin, and Y. Tian. 2019a. "Traffic Flow Prediction Using LSTM with Feature Enhancement." *Neurocomputing* 332: 320–327. doi:10.1016/J.NEUCOM.2018.12.016.

Yang, S., J. Wu, G. Qi, and K. Tian. 2017. "Analysis of Traffic State Variation Patterns for Urban Road Network Based on Spectral Clustering." *Advances in Mechanical Engineering* 9 (9), doi:10.1177/1687814017723790.

Yang, S., J. Wu, Y. Xu, and T. Yang. 2019b. "Revealing Heterogeneous Spatiotemporal Traffic Flow Patterns of Urban Road Network via Tensor Decomposition-Based Clustering Approach." *Physica A: Statistical Mechanics and Its Applications* 526: 120688. doi:10.1016/J.PHYSA.2019.03.053.

Yao, R., W. Zhang, and M. Long. 2021. "DLW-Net Model for Traffic Flow Prediction Under Adverse Weather." *Transportmetrica B: Transport Dynamics* 10 (1): 499–524. doi:10.1080/21680566.2021.2008280.

Zhao, L., Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li. 2020. "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction." *IEEE Transactions on Intelligent Transportation Systems* 21 (9): 3848–3858. doi:10.1109/TITS.2019.2935152.

Appendices

Appendix 1. PCA sensitivity to the amount of data

One of the most important benefits of using PCA as a baseline for data reshaping is its low sensitivity to the amount of data. The latent space can be derived precisely using this methodology even when there is a lower amount of data rather than six years, like one year, or even six or five months. Figure A1 is represented in this section showing the reshaped daily volume profiles of the Sydney Rd – Gaffney St (N) location to investigate the impacts of data size on PCA performance. As we can see, PCA is able to detect the differences between daily volume profiles of each day, even using just three months of data. The differences become more observable in the latent space using lower data sizes, such as one year. However, very low sample sizes like three, two, or one month may result in improper results as similar data groups (such as Mondays and Tuesdays in Figure A1) start creating clusters in very low sample sizes.

Compared to the recent dimension reduction approaches developed in the literature, like autoencoders (Boquet et al. 2020), PCA is advantageous not only when it comes to low data sample sizes (as indicated above) but also in terms of run-time. Although neural network autoencoders have their own benefits in dimension reduction of traffic data, such as exploring nonlinear relationships among the data, training them is a stochastic process and requires considerable time to be trained. However, there is no training phase while reshaping the data using PCA; and as shown, a huge amount of data does not significantly affect its performance.

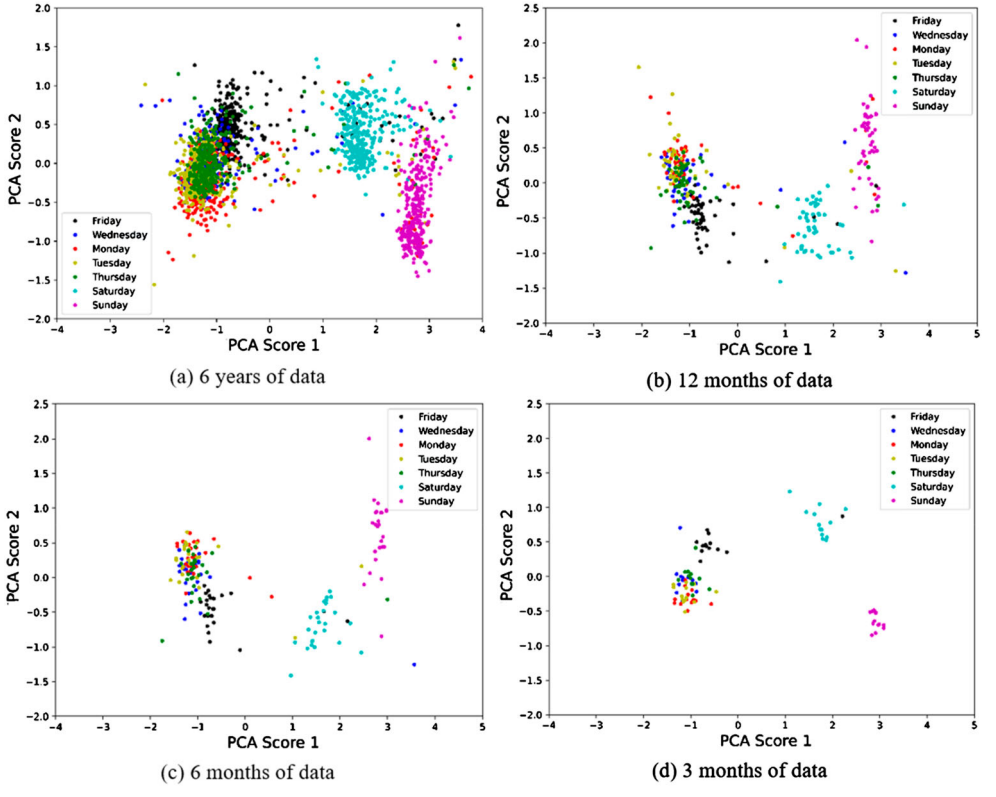


Figure A1. PCA performance using different amounts of data: daily volume profiles are reshaped in this figure to see the effects of day of the week (Sydney Rd – Gaffney St)

Appendix 2. Sensitivity analysis of ε in DBSCAN

Based on the method section, we utilized the KNN algorithm to find the best value of ε in DBSCAN and decided to use $\varepsilon = 3 \times \varepsilon_{elb}$. As there is no real ground truth label for anomalies in traffic data to calculate accuracy indicators, we employed visual inspection of data in every type of analysis using the data of multiple locations. In this experiment, DBSCAN with $\varepsilon = 3 \times \varepsilon_{elb}$ detected anomalies precisely and better than any other ε value in all sets of the available data. To keep the paper within a reasonable size, we randomly selected data from different types of temporal analysis to show the performance of DBSCAN. Results from this test are demonstrated in Figure A2. In this figure, we first show different pairs of data from different locations in the PCA latent space (first column). The detected outliers by DBSCAN with $\varepsilon = 3 \times \varepsilon_{elb}$ are illustrated in the second column. In the third column, we also applied pairwise K-means clustering to the decontaminated data to see if it can handle the clustering or not. As observed, all situations are successfully handled in our data by applying the DBSCAN algorithm with $\varepsilon = 3 \times \varepsilon_{elb}$.

To better clarify the reason behind choosing $\varepsilon = 3 \times \varepsilon_{elb}$ in our analysis, Figure A3 is provided. In this figure, we applied DBSCAN on a sample pair of data changing the coefficient of ε_{elb} from 1 to 6. According to this figure, low coefficient values like 1 or 2 result in the exclusion of several data samples. High values also may not detect some outliers existing in the data. Therefore, $\varepsilon = 3 \times \varepsilon_{elb}$ is the best option for applying DBSCAN in our case study. Furthermore, Figure A4 shows that the performance of DBSCAN does not heavily rely on the selection of K_{KNN} (low sensitivity).

Appendix 3. Detection of homogenous time intervals from time of the day ARI matrix

To automatically detect adjacent time intervals with indifferent volume data, the algorithm in Table A1 is used in this research. The main idea behind this algorithm is to first replace the ARI values in ARI matrices using a predefined threshold (0.2 in our case) with 1 and -1000 (a negative score would be assigned to dissimilar pairs of factors). This algorithm then employs sliding windows with all the possible sizes moving diagonally to calculate a score (Figure 22). The score in our algorithm is equal to the sum of values surrounded by each sliding window. After score calculation, windows overlapping with higher score windows are removed, and the remaining ones would indicate the desired areas in the ARI matrices.



Figure A2. Random visual evaluation of hyperparameter selection in DBSCAN using KNN algorithm ($3 \times \varepsilon_{elb}$). Performance on data of different locations and all types of temporal factors are included. (a) Raw data of the targeted pair of factors, (b) detected outliers by DBSCAN, and (3) pairwise clustering applied on decontaminated data.

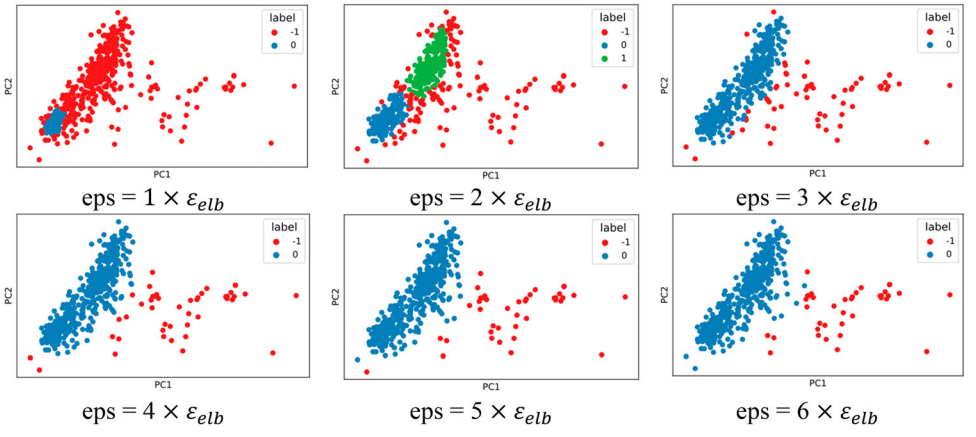


Figure A3. Effect of the elbow point coefficient in outlier detection: comparison of different values (1, 2, … , 6). Victoria St – Elizabet St (E) Tuesday vs. Friday.

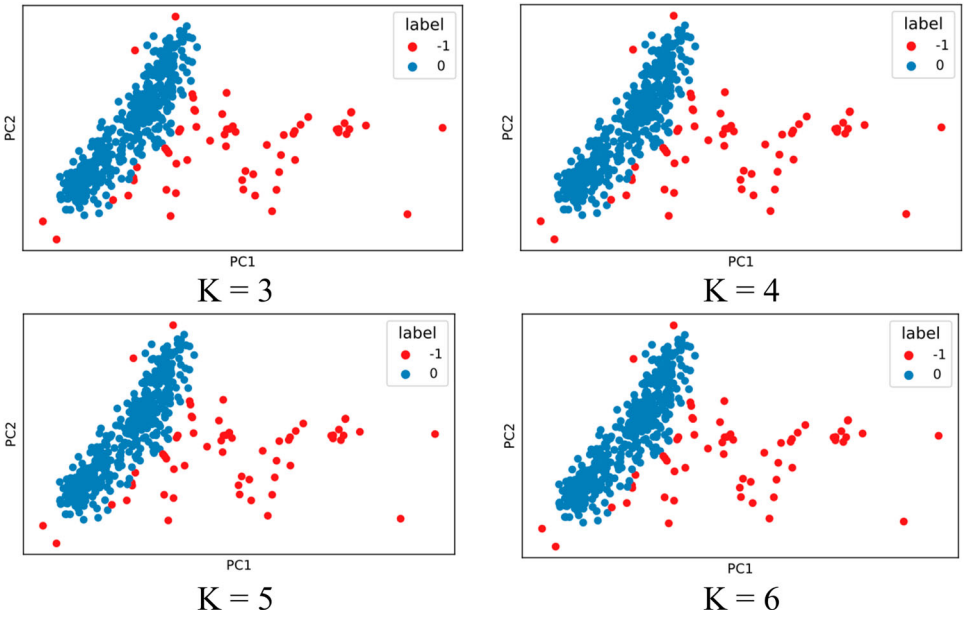


Figure A4. Effect of K in KNN method for determining the elbow/knee point: comparison of different values (3, 4, 5, 6). Victoria St – Elizabet St (E) Tuesday vs. Friday.

Table A1. Algorithm for detection of similar time intervals during a day.

Algorithm: Detection of similar time intervals

Input: Adjusted Rand Index matrix with values between 0 and 1
Output: Groups of adjacent homogenous time intervals (window locations)

```
threshold ← 0.2
negativescore ← −1000
for each ARI in ARI matrix do
  if ARI > threshold then
      ARI ← negative score
  else if ARI < threshold then
      ARI ← 1
  end if
end for
scores ← [ ]
for each window-size between 2 and 96 do
  W ← store all possible window locations (WL)
  for each WL do
      S ← sum all the ARI inside the window
      append S and WL to the scores
  end for
end for
sort scores based on S values
for each WL in scores do
  higher_WL ← WLs with higher S than the target WL in scores
  if WL overlaps with items in higer_wl then
      remove WL from the scores
  end if
end for
return WLs remained in scores
```