Unsupervised Learning for Public Transport Delay Pattern Analysis

Cheng, Yuxing; Krishnakumari, Panchamy

DOI
10.1177/03611981231215333

Publication date
2024

Document Version
Final published version

Published in
Transportation Research Record

Citation (APA)
Cheng, Y., & Krishnakumari, P. (2024). Unsupervised Learning for Public Transport Delay Pattern Analysis. *Transportation Research Record*, *2678*(8), 358-373. https://doi.org/10.1177/03611981231215333

TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Research Article

Transportation Research Record
1–16
© National Academy of Sciences:
Transportation Research Board 2024
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/03611981231215333
journals.sagepub.com/home/trr
Sage

Unsupervised Learning for Public Transport Delay Pattern Analysis

Yuxing Cheng[1] and **Panchamy Krishnakumari**[1]

Abstract

To analyze inherent and diverse patterns within line-based public transport daily delay occurrences, we introduce a data-driven exploratory analysis focused on the spatial-temporal distribution of these delays. Our approach relies on the utilization of the image pattern recognition technique and *k*-means clustering algorithm. We extract daily punctuality information from the automatic vehicle location data for a singular public transport route. This information is then translated into a visual representation through aggregated daily delay distribution profile images, offering insights into the spatial and temporal distribution of delays. The delay distribution finds expression in the arrangement of pixels within these profile images. The essence of these images is further distilled through image pattern recognition using the neural network architecture of ResNet50. Employing the *k*-means algorithm, we cluster these images based on their similarity, revealing five distinct daily delay patterns. The analysis of these patterns offers insight into their unique characteristics, yielding noteworthy outcomes. These findings hold the potential to provide public transport operators with an enriched comprehension of the dynamics of delays occurring on a specific line.

Keywords

data and data science, visualization in transportation, analytic data visualization, public transportation, quality, performance measures, automatic vehicle location

Punctuality stands as a fundamental pursuit in public transport (PT) systems. Accomplishing this objective necessitates a cohesive combination of strategic, tactical, and operational management approaches. Successful management hinges on a comprehensive understanding of the inherent complexities characterizing PT operations. It is crucial to acknowledge that each PT line has distinct roles within the system, yielding varied functions and resulting in diverse delay patterns (*1*, *2*). Is there any spatial-temporal distribution pattern of delay that occurred on a specific PT line? Answering this question could provide PT operators with invaluable insights into the routine occurrence of delays on various lines. Such insights have the potential to assist operators in tailoring management strategies to address specific delay patterns and optimize operational efficiency (*3*). The increasing variety of PT-related data resources empowers operators to scrutinize diverse phenomena emerging during daily PT operations., especially automatic vehicle location (AVL) data and general transit feed specification (GTFS) data (*4*, *5*). AVL and GTFS datasets contain dynamic (e.g., locations) and static (e.g., stop information, geographic structure, and schedule) information collected when the PT lines are operated (*6*). This research aims to construct a methodology for data-driven exploration of PT line-based spatial-temporal patterns of delay occurrence. Understanding the spatial-temporal pattern of delay distribution on a single line could be meaningful for the operator (*7*).

Considerable research effort has been built on the AVL data to better understand the characteristics of phenomena that occur in PT lines. Previous studies have explored various methodologies for extracting operational information from this data, aimed at comprehending service reliability. An effective approach involves extracting the spatiotemporal load profile (*8*, *9*). This profile incorporates data from diverse sources through a well-constructed algorithm, transforming raw data into an informed visualization (*4*). The utility of this approach in visualizing PT

[1] Department of Transport & Planning, Delft University of Technology, Delft, The Netherlands

Corresponding Author:
Yuxing Cheng, y.cheng-1@tudelft.nl

operations has been widely recognized across diverse PT studies. For example, Degeler et al. (*5*), implemented an operational profile to illustrate the daily progression of a singular tram line. These profiles encompass real-time geolocation of PT vehicles incorporated with passenger loads throughout the journey. This framework detects and categorizes bunching phenomena through an unsupervised machine learning technique. Similarly, for short-term train load prediction, Bapaume et al. (*10*) introduced an image-processing-oriented methodology. This technique employs profile images to represent the train loads at each stop, facilitating load prediction.

Moreover, multiple studies carried out exploration analysis based on unsupervised learning (*11*). Unsupervised learning techniques have recently been employed to investigate spatial travel patterns and demand, given their natural advantages in solving clustering problems (*5*). In the area of PT, many analyses related to clustering rely on *k*-means, which permit one to cluster relatively large sets of data and require only a few parameters, especially the desired number of clusters (*12*). Most of the previous studies that implement the *k*-means algorithm in the PT field are based on the low-dimensional input, which means the number of attributes of each datapoint is relatively small and definite (*5*, *7*).

However, the disadvantage of the low-dimensional input for clustering algorithms is that the more complex spatiotemporal PT dynamics could not be well represented, and the clustering could only be based on the limited features among datapoints. The advantage of the profile derived from real-time PT data (e.g., AVL data) is that it can construct a complete view of the PT operation in time, space, and more dimensions. Still, the existing research methodology cannot fully use this advantage. To the best of our knowledge, very few studies have attempted to use clustering on high-dimensional input (e.g., images) that represents the PT dynamics. Combining clustering and representation learning is one of the most promising approaches for unsupervised learning of deep neural networks (*8*). Thus, this paper focuses on extracting the latent feature of delay occurrence patterns on a single PT line by a deep learning algorithm, and clustering these patterns in specific categories. The study aims to answer the following question: what is the spatial-temporal distribution pattern of delay occurrence on a single PT line? The contributions of this paper are as follows.

- We introduce an image-based approach to extract and visually represent daily punctuality information of a single PT line. The resulting daily delay profile image offers insights into the spatial-temporal distribution pattern of delay occurrence.
- This study applies a pre-trained convolution neural network architecture, Resnet50, for image feature recognition. The advantage of this feature recognition approach is that the abstract spatial-temporal distribution characteristics of delay occurrence could be extracted from the profile images. This technique is different from the previous studies that vectorize the daily punctuality data for the *k*-means algorithm or define the attributes of each sample manually.
- The clustering outcomes furnish a comprehensive depiction of diverse delay patterns on a specific PT line. This foundational knowledge serves as a precursor for subsequent studies, including supervised learning on PT dynamic patterns or applications in planning and management. Furthermore, this method holds potential for extension to other PT lines or systems.

Punctuality mentioned in this research refers to the time difference between the scheduled and real arrival time for the tram at each stop. So, the value of punctuality could be positive (representing delay) or negative (indicating early arrival). For simplification, we mainly focus on exploring the delay patterns. The feature of early arrival pattern can be obtained as complementary findings.

The next section of this paper presents the proposed methodology from two aspects, the details of implementing the *k*-means algorithm in this research and the image processing approaches. Then, the case study setup based on the cleaned AVL data is introduced. After that, the results of clustered delay patterns are presented, with the analysis of the spatial-temporal characteristics of each kind of delay pattern. Finally, the conclusion is drawn with discussion and suggestions for further research.

Methodology

This section presents the approach employed to identify spatial-temporal attributes associated with distinct daily delay patterns. We begin with a conceptual model overview, followed by a comprehensive explanation of the *k*-means clustering technique. In addition, we delve into the image processing strategies utilized in this study, serving as the intermediary linking the daily delay profile image to the *k*-means clustering algorithm.

Overview

An overview of the methodological framework is shown in Figure 1. The raw AVL and GTFS datasets are stored separately as input, which contains the daily running situation information and the PT lines network information. The raw data obtained from these two datasets are cleaned and useful information is extracted. Then, based

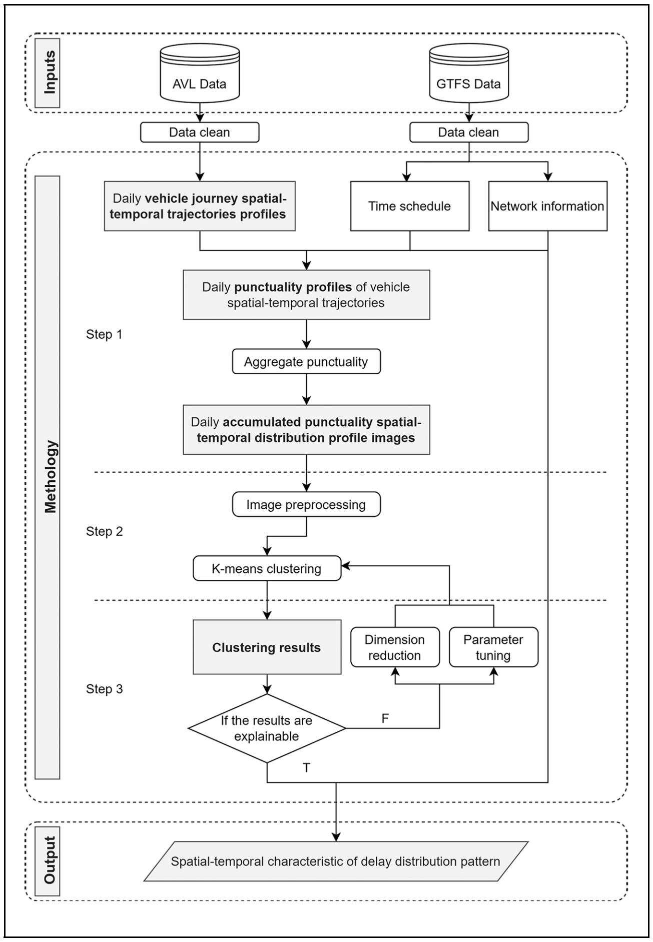
Inputs
AVL Data
GTFS Data
Data clean
Data clean
Methology
Daily vehicle journey spatial-temporal trajectories profiles
Time schedule
Network information
Daily punctuality profiles of vehicle spatial-temporal trajectories
Step 1
Aggregate punctuality
Daily accumulated punctuality spatial-temporal distribution profile images
Image preprocessing
Step 2
K-means clustering
Clustering results
Dimension reduction
Parameter tuning
Step 3
If the results are explainable
F
T
Output
Spatial-temporal characteristic of delay distribution pattern

Figure 1. Overview of the methodology.
Note: AVL = automatic vehicle location; GTFS = general transit feed specification.

on the three-step method, the daily delay patterns with different spatial-temporal characteristics are derived from the raw data and clustered based on the daily delay profile image. Finally, the identified daily delay patterns are interpreted to extract insights, which is this study's final output.

k-Means Algorithm (For Image Clustering)

The k-means algorithm is a concise clustering method within unsupervised learning, first introduced by MacQueen (*13*), and has found utility in diverse domains. Given a dataset X with n datapoints each having m dimensions, $X = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix}$, the algorithm aims to partition them into k ($k < n$) clusters $\{C_1, C_2, C_3, \ldots, C_k\}$. For each cluster, the centroid is defined as the mean of datapoints belonging to the cluster and is calculated iteratively until the algorithm process is terminated. The definition of centroid could also be generalized in high-dimensional space. For each iteration, each datapoint is assigned to the nearest cluster, based on the distance to each centroid. Multiple methods are used to calculate the distance between datapoints (*12*), and the Euclidean distance is the most used. The algorithm iteration is terminated when the assignment results of all datapoints no longer change. The generic k-means algorithm includes the following steps.

1. Select k samples $\{u_1, u_2, u_3, \ldots, u_k\}$ in the dataset randomly as initial cluster centroids of k clusters.
2. For all the other datapoints, calculate their distance to the initial cluster centroids, and assign them to the nearest cluster. The most common distance calculation is Euclidean distance D, which is as follows:

$$D_i(x_i,\ C_i)^2 = \sum_{d=1}^{m} (x_{id} - C_{id})^2 = \|x_i - C_{i2}^2\| \qquad (1)$$

where i denotes the label of a cluster, D_i denotes the Euclidean distance between datapoint x_i and centroid u_i, and both belong to the cluster C_i.

3. Calculate the mean of each cluster's datapoints to derive new centroids.
4. Iterate steps 2 and 3 iteratively until the limitation of iteration or the assignment no longer changes, which means the within-cluster sum of squared errors (SSE; the sum of the distance of each datapoint to the corresponding cluster centroid):

$$SSE(k) = \sum_{i=1}^{k} \sum_{x \in C_i} D_i(x_i,\ u_i)^2 \qquad (2)$$

In each cluster, the objective is to minimize the distance between each datapoint assigned to the cluster and its centroid while maximizing the differences among centroids of distinct clusters. Therefore, the k-means algorithm seeks to optimize the problem by minimizing the SSE within each cluster.

A common challenge in unsupervised machine learning algorithms is that clustering methods tend to generate clusters even in the absence of evident data clustering. It becomes essential to assess whether a substantial clustering tendency exists within the vectorized input dataset, leading to logical and meaningful cluster outcomes. To evaluate the clustering tendency of a specific dataset, we employ the Hopkins statistic h (*14*, *15*). The value of h lies between 0 and 1, with a higher value indicating closely grouped datapoints. A value closer to 0 signifies a lower cluster tendency, suggesting regular spacing among points. Conversely, a value closer to 1 indicates a stronger cluster tendency. When the data is uniformly distributed, h approaches 0.5 (*16*).

The key point of k-means clustering is to determine the number of clusters k. Two methods have been commonly used for solving the problem: the elbow method based on the SSE curve and silhouette analysis.

The elbow method is one of the most popular methods to select the optimal number of clusters by fitting the model with a range of values for k in the k-means algorithm. The elbow method requires a line plot between the SSE and the number of clusters and finding the point representing the "elbow point." However, the within-cluster SSE cannot determine the optimum cluster number independently. The elbow can be unclear, and the SSE can only reflect the data distribution within each cluster.

The other method is silhouette analysis. Silhouette analysis allows seeing how similar the points within the cluster are with the centroid point and how different they are from the points of other clusters (*17*). For each datapoint in a cluster, a silhouette score can be calculated on a scale from −1 to 1, and the average value of all silhouette scores represents the results of the silhouette analysis. The following equation calculates the average silhouette score:

$$S = \frac{1}{n}\sum_{i=1}^{n}\frac{b_i - a_i}{\max(a_i, b_i)} \tag{3}$$

where a_i represents the average distance between sample i and other samples in its cluster and b_i denotes the minimum average distance between sample i and the samples in other clusters. If a_i converges to 0 or b_i is high enough, the average silhouette score getting closer to 1 indicates that the clustering algorithm works better.

Although the mentioned methods could provide quantifiable evidence for choosing the value of k, for the number of clusters one should also consider the purpose of clustering based on prior knowledge. In this research, we need to observe if universal and generic features exist among the datapoints within each cluster. In this way, the patterns of daily delay distribution characteristics could be extracted, concluded, and generalized.

Image Processing Approaches

As introduced before, k-means clustering is based on real vectors with multiple dimensions. However, the daily delay information is represented by the delay profile images. So, how the images are vectorized and plugged into the k-means algorithm needs to be discussed.

In the previous research, the dimensions of the k-means input are limited (usually no more than 10^1 or 10^2 attributes [dimensions]). However, in this research, the spatial-temporal distribution of the delay phenomenon is represented by the daily delay profile image and the dimension of the image input we propose to obtain (more than 10^2 or 10^3, depending on the aggregation granularity of the delay profile images) could be relatively much higher. The challenges brought about by the high-dimensional input for k-means include two aspects.

1) The k-means algorithm determines the datapoint cluster affiliation based on the pair-wise distance, but all the points are at a similar distance from the others when the dimension increases. Thus, the notion of "nearest points" vanishes in the high-dimensional space (*18*).
2) The input for the k-means distance calculation is one-dimensional vectors. If an image is unfolded to a one-dimensional vector directly, any possible translation or disturbance of the image could significantly affect the clustering result because of the difference in pixel values (*19*). For example, in two identical images, if one of them is translated to a one-pixel distance, no difference could be found in their appearances. Still, they may be attributed to two different clusters.
3) The spatial relationship among the image pixels would be ignored if we directly unfold the image pixel values to a one-dimensional vector. One of the most significant advantages of representing the daily delay information by the punctuality profile image is that the spatial-temporal characteristics could be visualized and analyzed.

To solve the problems caused by the characteristics of the data in this research, image recognition based on the pre-trained deep neural network architecture Resnet-50 is implemented, combined with two kinds of dimensionality reduction approaches. Using the image feature recognition algorithm Resnet-50, the spatial-temporal distribution characteristics of daily delay could be extracted. The dimensionality reduction could help the k-means algorithm to be more efficient. Thus, the appropriate input attributes for the k-means algorithm can be obtained.

Image Feature Extraction.

For many image clustering or classification problems, replacing raw image data with features extracted by a pre-trained convolutional neural network (CNN) leads to better clustering performance (*20*). The previous research compared multiple neural network architectures and proved that the ResNet50 could perform relatively better than other prevailing architectures (*21*, *22*). The residual network is a classic neural network used as a backbone for many computer vision tasks, which was first proposed by He et al. (*23*) in

2015. ResNet-50 is a CNN with 50 layers. The pretrained Resnet-50 deep neural network architecture can effectively recognize the features of the images and has been widely used in computer vision, including image classification and detection applications. The process of implementing ResNet50 is done by the PyTorch deep learning framework.

Dimensionality Reduction. The dimensionality reduction method can transfer the high-dimensional data into low-dimensional space, vital in feature engineering, data visualization, and saving computation time (*18*, *24*). There are two kinds of dimensionality reduction methods: projection and manifold learning (*25*). Principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) are two typical algorithms belonging to these two kinds of dimensionality reduction methods, respectively.

As a traditional and popular dimensionality reduction technique, PCA is a linear technique that keeps the low-dimensional representations of different datapoints far apart. However, PCA cannot account for complex polynomial relationships between features and may cause inevitable reduction of image information, which is the internal limitation of the dimension reduction method. Unlike PCA, t-SNE is a nonlinear dimensionality reduction algorithm based on the probability distribution of random walks on the neighborhood graph to find the structure within the data. It maps multidimensional data to two or more dimensions suitable for human observation. In the research conducted by van der Maaten and Hinton (*24*), t-SNE is better than existing techniques at creating a single map that reveals structure at many different scales. It is essential for high-dimensional data that lie on several different but related, low-dimensional manifolds, such as images of objects from multiple classes seen from various viewpoints.

The k-means algorithm calculates which cluster a datapoint belongs to based on the distance among vectors with multiple dimensions, and the dimension is high in this research. Accordingly, the dimensionality reduction could be plugged into the clustering method in two roles: (1) processing the output of ResNet50 with $3 \times 3 \times 512$ dimensions to be the input of the k-means algorithm; (2) visualizing high-dimensional k-means output data by giving each datapoint a location in a two- or three-dimensional map. The first role may not be necessary, but for some specific data, the dimensionality reduction before k-means could help with feature selection and reducing time complexity (*26*). So, a comparison is made among the data preprocessing methods with (PCA or t-SNE) and without dimensionality reduction before k-means clustering. The second role is one of the critical parts of analyzing the effectiveness of the clustering algorithm, which is necessary. So, t-SNE is chosen to visualize the clustering results.

Case Study

In this study, we have employed the GTFS dataset along with the AVL dataset. These sources have enabled us to extract historical real-time operational information for public transport (PT) lines within The Hague. The dataset covers a span of 79 days across June, July, and August in 2019. Specifically, tram line 1 was chosen for our comprehensive analysis. Renowned for its lengthy operation within and around The Hague, tram line 1's significance extends across decades.

The line's route intersects diverse land use patterns, ensuring the presence of a multitude of daily delay patterns. As the city's oldest and longest-running tram line, its route spans from Scheveningen Noord to Delft Tanthof, passing through key locales such as The Hague city center, Hollands Spoor station, Rijswijk Haagweg, and Delft station. This route's layout is visually depicted in Figure 2. Throughout the day, the headway between tram vehicles fluctuates, with intervals ranging from 8.1 min at 8 a.m. to 14.5 min at 10 p.m. On average, the daily headway rests at 10.6 min. Consequently, we can gather punctuality data from approximately six tram vehicles at a single stop within an hour, on average.

The dynamic information is derived from the AVL data, shown in Table 1, which contains the line number, vehicle number, journey number, actual arrival/departure time to the platform of the vehicle, stop (platform) code, distance to the last platform, punctuality, and more related information of all the PT lines operated in The Hague. Besides, the static network information is derived

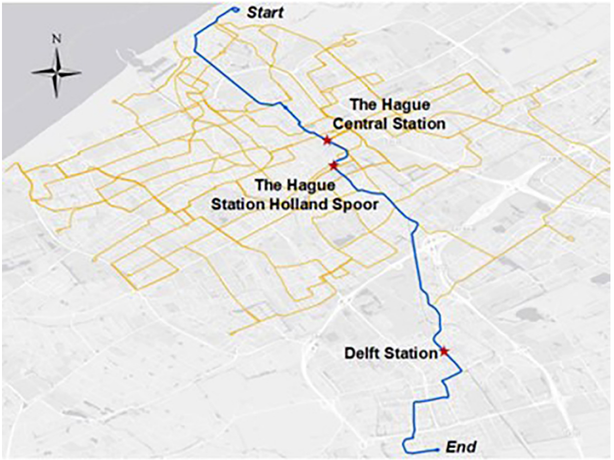

Figure 2. Tram line 1 from Scheveningen Noorderstrand to Delft Tanthof.

Table 1. Description of the Automatic Vehicle Location Data of Each Tram Vehicle

Name	Data type	Example	Description
Receive	DateTime	2019-06-05 04:51:52.086383	Time of sending the message by the source system
Messagetype	string	DEPARTURE	The status of the vehicle
Operating day	datetime	2019-06-05	A specific date the datapoint belongs to
Dataownercode	string	HTM	Operator company
lineplanningnumber	int	1	Line number of the journey belongs
Journeynumber	int	30004	Public journey number
Userstopcode	int	9594	Stop number of the stop where the arrival/leave is
Punctuality	int	20.0	Current deviation from the scheduled arrival time in seconds for this stop. Too early <0, too late >0, on time =0
rd_x&rd_y	float	86795.0, 454011.0	RDS in meters. RD coordinates refer to locations in the Netherlands according to the Rijksdriehoek system4
Vehiclenumber	int	4047	Vehicle identification number

Note: RD = Rijksdriehoeks-coordinates (Dutch grid coordinates).

from the GTFS dataset, which contains each platform's stop name, code, and geographic location. To extract the available information, we select the datapoint with the actual stop name and collect the exact time and position data when a tram leaves the platform. Finally, 78 platforms (34 stops in one direction) are identified and all the stops of tram line 1 are selected. In addition, for each datapoint, the punctuality is derived from the schedule and actual arrival time. Besides, some of the datapoints with extreme attribute values are deleted from the dataset, as these could be caused by a particular situation, such as rare equipment failure or temporary traffic control. The well-organized data could make the subsequent computation more efficient through data cleaning.

Results: Explored Daily Delay Patterns

Punctuality Visualization

To represent the daily delay spatial-temporal distribution pattern of tram line 1, the delay profile is obtained from the AVL and GTFS data, as Figure 3 illustrates. Figure 3 contains the spatial-temporal trajectories of tram line 1 tram in a single operation day, in both directions. The dots represent the time and location when a tram sends the signal that it is leaving a platform. The shade colors of the dots represent the punctuality in the unit of seconds, with the value from -200 to 200 s. The positive value represents delay, and the negative value represents early arrival. The dash lines connect multiple dots representing the trajectories of tram vehicles. The blank segmentation that appears between platforms 38 and 39 represents the end of a single-direction journey. The range of the y-axis from platform 1 ("Den Haag, Zwarte Pad") to platform 38 ("Delft, Abtswoudsepark") represents one direction, and this direction is defined as "direction 1." The range of the y-axis from platform 39 ("Delft, Abtswoudsepark") to platform 78 ("Den Haag, Zwarte Pad") is defined as "direction 2."

To facilitate the implementation of image clustering for pattern recognition, a crucial step involves transforming the original punctuality data. This transformation is essential to ensure the clear identification of spatial-temporal distribution features. To achieve this transformation, we adopt a strategy of aggregating the punctuality profile across a time scale. Specifically, we utilize the average punctuality value within each 1-h interval to effectively represent the average delay. This procedure yields what we term "aggregated punctuality profile images," illustrated in Figure 4. Two methods of punctuality aggregation are used: aggregate to the original value and aggregate to two punctuality types ("delay" or "on time"). Figure 4*a* shows the original punctuality value in space and time scales. Figure 4*b* shows the classified punctuality types (delay or no delay), which can indicate the temporal-spatial distribution of delay occurrence, but ignores the delay severity. For Figure 4*b*, the criteria of delay definition are derived from the distribution of the punctuality values in the whole dataset. The purpose of using two kinds of punctuality aggregation levels is to compare which one can better reflect the feature of the daily delay pattern and lead to clustering results with more distinctive features. Thus, the output images of 79 days aggregated daily delay profiles with different aggregation levels will be processed by ResNet50 and compared.

Punctuality Profile Image Clustering

In this section, we implement the proposed image preprocessing approaches through a two-fold process: firstly, extracting the spatial-temporal pattern of daily delay distribution by recognizing the pixel distribution pattern in

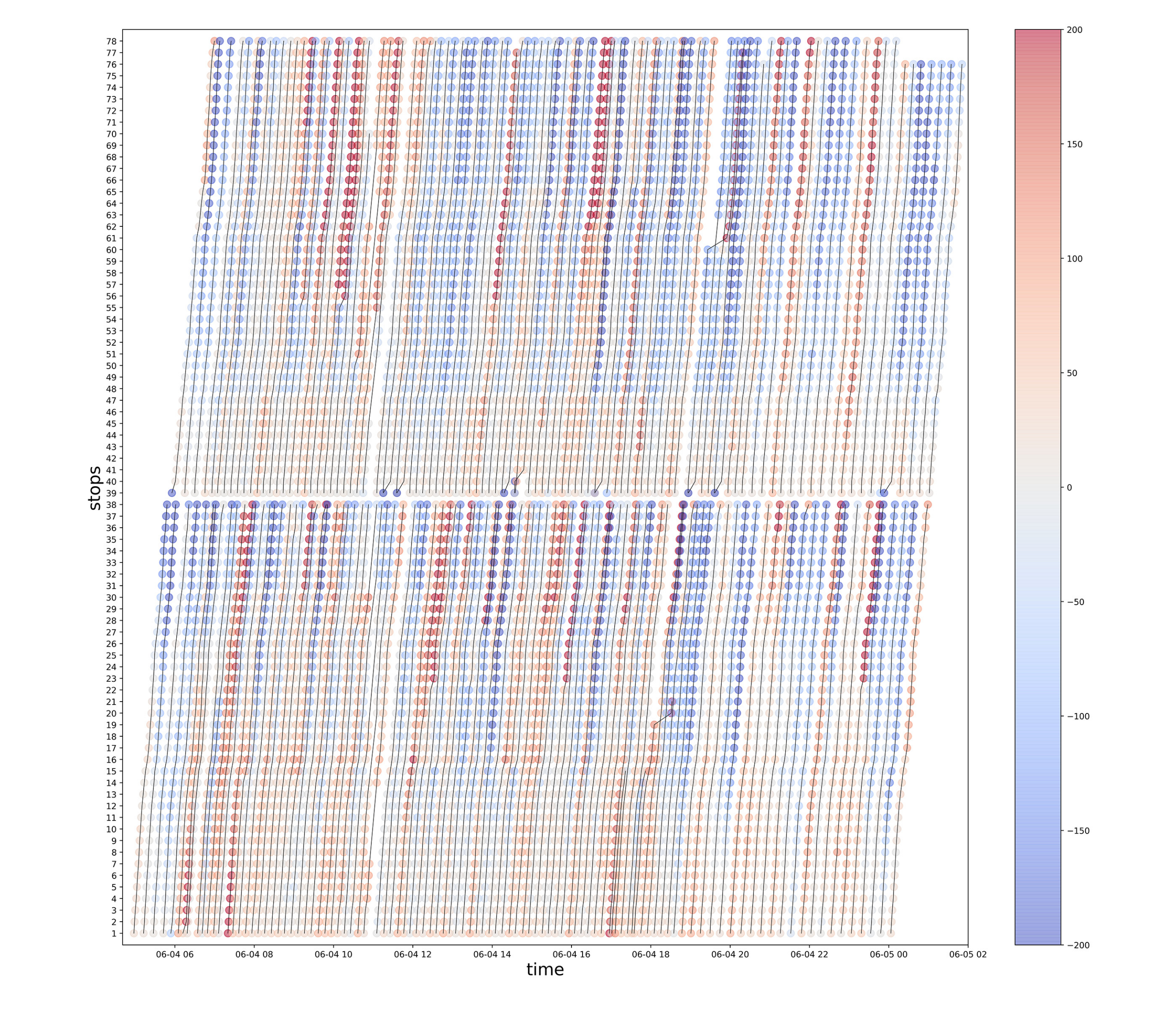

Figure 3. The punctuality profile of tram line 1 on July 11, 2019.

the aggregated punctuality profile images and, secondly, clustering these patterns into clusters that are maximally distinct from each other. These sequential steps are termed "image feature extraction" and "dimensionality reduction," respectively, and are each introduced in the *Methodology* section.

Based on the aggregated punctuality profile images, as depicted with an example in Figure 4, we employed ResNet50—a pre-trained neural network—to recognize the underlying patterns within the set of 79 input images. The process of image feature extraction is depicted in Figure 5. The output is the vector that can represent the feature of the corresponding image. Nevertheless, it remains necessary to implement dimensionality reduction to transform these high-dimensional vectors, making them conform to the low-dimensional criteria required for *k*-means clustering.

Various approaches can be employed to preprocess the input vectors for clustering. However, the precondition of meaningful clustering results is that the data be nonuniformly distributed and show a clustering tendency. The clustering tendency of the input datapoints obtained by different approaches should be compared to evaluate if these acquired datasets could lead to meaningful clustering results.

Thus, we select three image preprocessing approaches and implement them to compare their performance in transforming images into input data for *k*-means clustering with a significant clustering tendency: (1) ResNet50, (2) ResNet50 + PCA, and (3) ResNet50 + t-SNE.

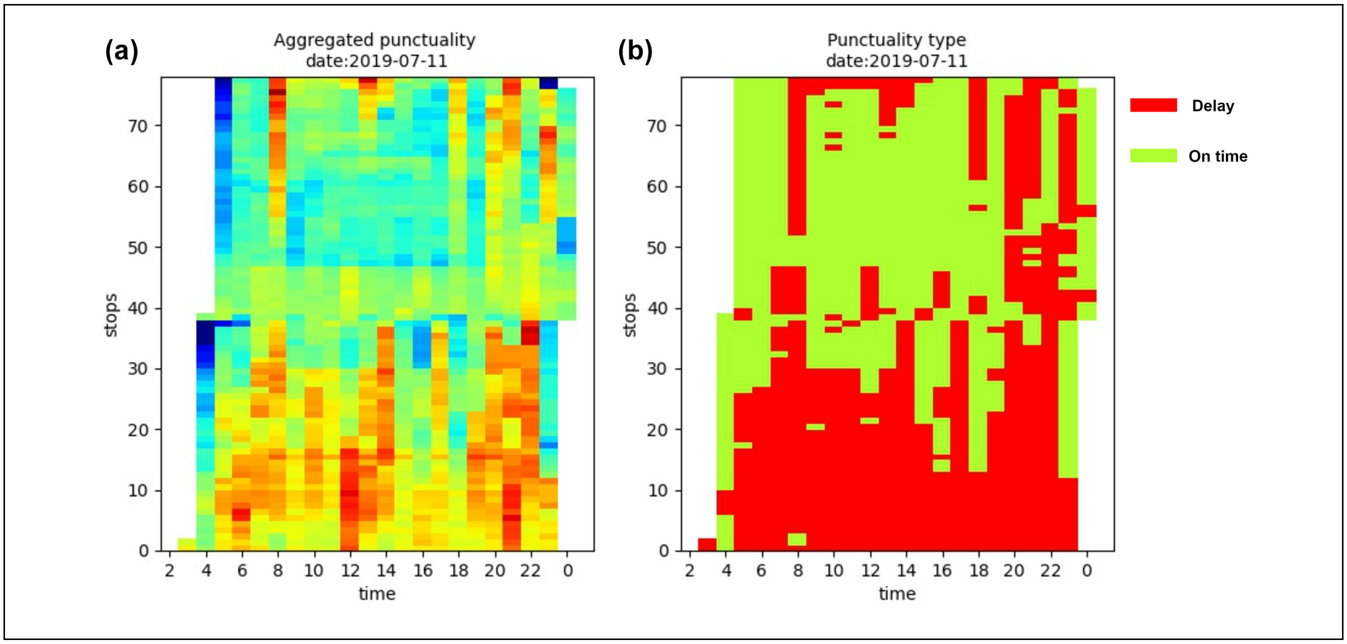

Figure 4. Two kinds of aggregated punctuality profile images extracted from Figure 3: (*a*) original punctuality values and (*b*) classified punctuality types.

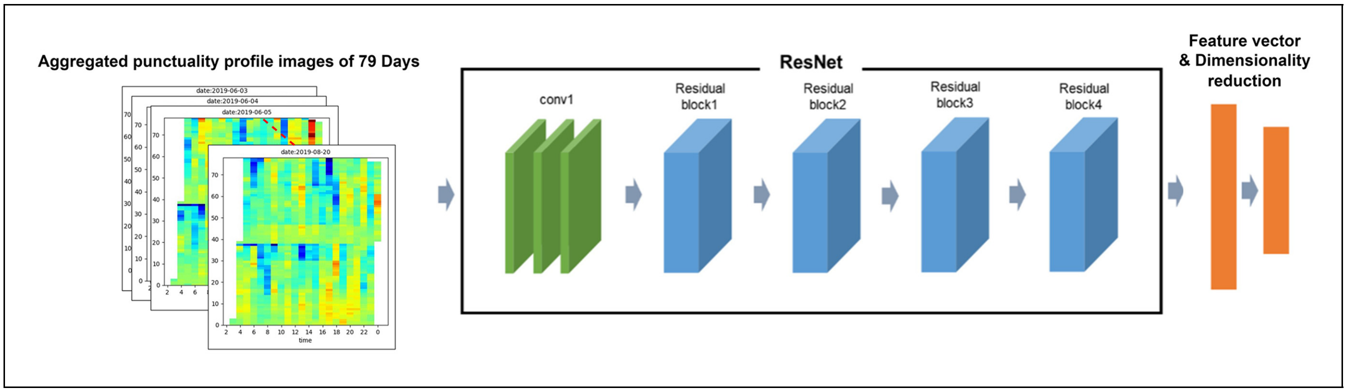

Figure 5. Process of image feature extraction.

Besides, two kinds of images we obtained in the former steps are used, which are "original punctuality value images" and "classified punctuality images." These two types of images contain different kinds of daily delay distribution characteristics with different granularities. We compare these two kinds of images to see which one can obtain clusters with clear distinctions. Combining the above-mentioned image preprocessing approaches and image types, six kinds of image preprocessing approaches are tested and compared using the Hopkins statistic h. The results are illustrated in Table 2 and Figure 6.

Table 2 and the boxplot in Figure 6 illustrate that for all the implemented image preprocessing approaches, the means of the h statistics are all higher than 0.5, which means these approaches can obtain the input data with a clustering tendency. Also, this could suggest the clustering tendency in daily delay dynamics. Thus, it is possible to classify the daily delay pattern and draw the conclusion of delay pattern characteristics. According to Figure 6, preprocessing methods (1), (4), (2), and (5) can obtain a dataset with a relatively more stable clustering tendency, with higher means and lower deviation of h. However, methods (3) and (6) have the probability of obtaining the image data with a higher clustering tendency with acceptable fluctuation.

The process aims to cluster and distinguish the different daily delay patterns. So, the higher the clustering tendency, the more preferred the results, even though the

Table 2. Comparison Among Multiple Images Preprocessing Approaches Before *k*-Means

	Hopkins statistic *h*					
Image preprocessing approaches	Mean	SD	Min.	25%	75%	Max.
Original punctuality profile images						
1) ResNet50	0.632	0.0068	0.6175	0.6311	0.6407	0.6505
2) ResNet50 + PCA	0.567	0.0084	0.5454	0.5571	0.5647	0.5770
3) ResNet50 + t-SNE	0.5972	0.0752	0.4424	0.5505	0.6555	0.7525
Classified punctuality profile images						
4) ResNet50	0.6224	0.0056	0.6058	0.6191	0.6269	0.6324
5) ResNet50 + PCA	0.5616	0.0060	0.5549	0.5707	0.5793	0.6000
6) ResNet50 + t-SNE	0.5960	0.0826	0.4684	0.5318	0.6453	0.8289

Note: PCA = principal component analysis; t-SNE = t-distributed stochastic neighbor embedding; SD = standard deviation; Min. = minimum; Max. = maximum.

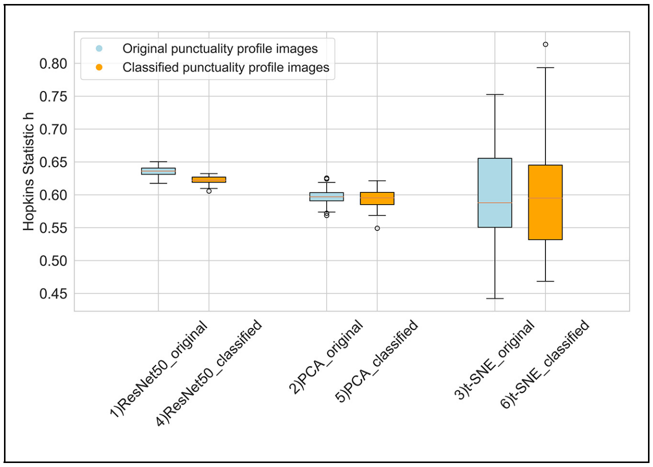

Figure 6. Comparison of the Hopkins statistic *h* of six image preprocessing approaches before *k*-means clustering.
Note: PCA = principal component analysis.

preprocessing method may not be so stable with the higher deviation of *h*. According to this, the "ResNet50 + t-SNE" combination is chosen as the preprocessing method before the *k*-means algorithm.

Cluster Number

By choosing a suitable number of clusters *k*, the daily punctuality profile can be separated into meaningful delay patterns with distinct features. For the *k*-means clustering algorithm, the first issue that needs to be solved is the appropriate number of clusters *k*. To determine the *k* value, the SSE and silhouette analysis are implemented on the scale of *k* from 2 to 30. Also, the application and the analysis objectives should be considered so that the generalized daily delay distribution patterns are more desirable from the planning perspective and are more explainable.

The curve of the SSE and silhouette score with different *k* values is shown in Figure 7. The within-cluster SSE reflects the data homogeneity in each cluster. Figure 7*a* indicates that the SSE declines rapidly before $k = 8$, and then the decline slope alleviates, and the approximate elbow point is in the range of 3–8. When *k* is lower than 8, the impact of changing the *k* value on the SSE is relatively significant, so the clustering performance of *k* values in this range should be compared carefully.

Figure 7*b* illustrates the fluctuation in silhouette scores across different values of *k*. Notably, when *k* equals 2, the score surpasses other *k* values by a substantial margin, indicating the optimal separation between clusters. Yet, considering the associated SSE value when *k* equals 2, this high score does not adequately capture the cluster characteristics because of the variability within the clusters. Furthermore, the red line in Figure 7*b* marks a score of 0.2. It is evident that more than 67% of all the silhouette scores fall below this threshold. As *k* surpasses 8, the scores progressively decline, indicating that the most appropriate *k* value for achieving satisfactory clustering results lies between 3 and 8. The red dot signifies that the second-highest silhouette score at *k* equals 5. This point holds the second-highest score after *k* equals 2 and highlights its significance in the context of silhouette scores.

Based on the analysis above, we visualize the clustering results for various *k* values (ranging from 2 to 7) in Figure 8, allowing for a detailed comparison among these outcomes. In each figure, individual datapoints correspond to one aggregated punctuality profile image. Distinguished by their respective colors, each cluster is denoted by specific numeric labels. The *xy*-coordinates in each figure represent the value of the two-dimensional (2D) feature vector, which is the output of the PCA dimension reduction. The distance among datapoints could indicate their similarity, which is unitless. Closely located datapoints signify higher similarity, while those farther apart suggest lower similarity.

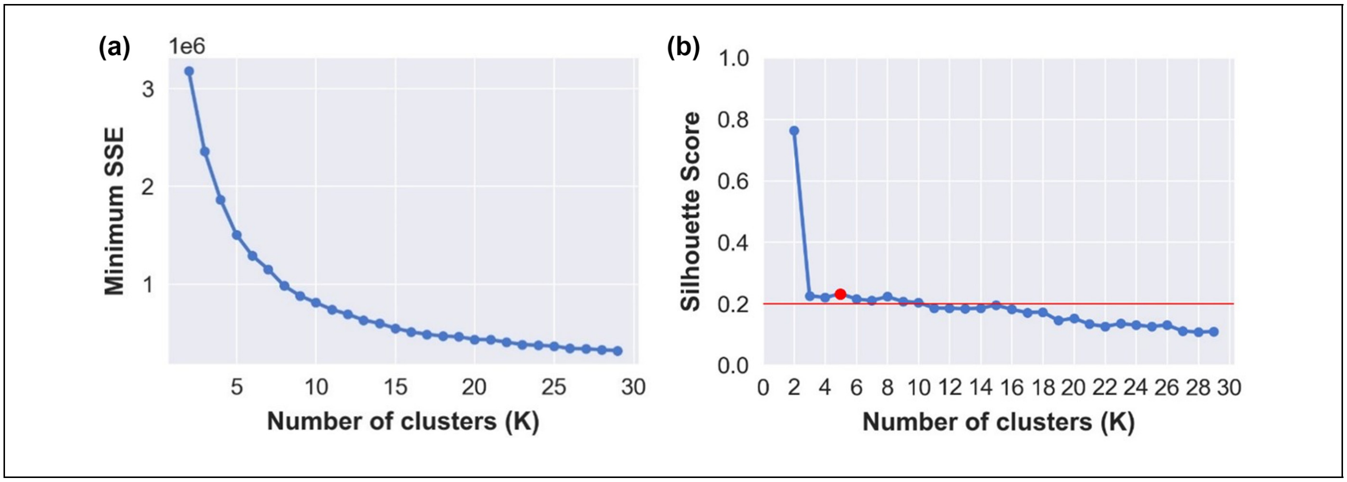

Figure 7. Analysis for determining the number of clusters (*k*): (*a*) the sum of squared errors (SSE) decreases exponentially as the number of cluster increases and (*b*) the silhouette score (color online only).

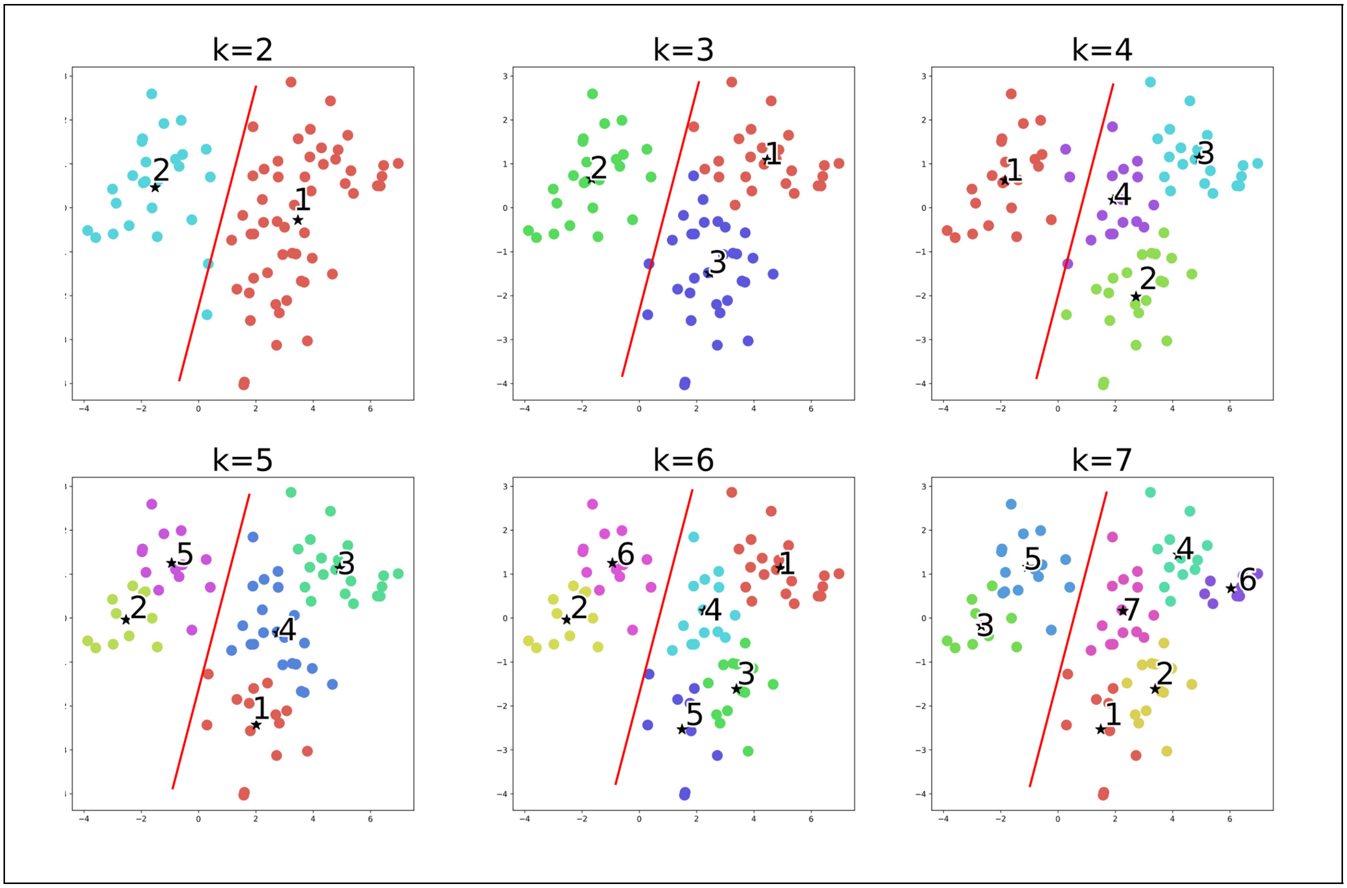

Figure 8. Clustering results with different *k* values. The red line denotes the approximate boundary between the two primary clusters (color online only).

It can be observed that the clustering results when *k* is larger than 2 are based on the clustering when *k* equals 2, and the red lines denote this phenomenon. This means the data could largely be divided into two clusters and, in each cluster, the data have the potential to be separated into multiple sub-clusters. This feature is marked by the red line in Figure 8, which is the approximate boundary between the two primary clusters.

The clusters should have as significant differences as possible, be self-contained, and be coherent. Also, the clusters should not be too large to make reasonable explanations and be easily generalized. Accordingly, $k = 5$ is chosen for the k-means clustering algorithm, and the result of clustering is shown in Figure 9.

In Figure 9, the results of $k = 5$ are shown in detail, with the clusters centroids and latent outliers marked. We notice that some latent outliers are lying approximately at the boundaries between multiple clusters, which means those datapoints' features may not be as clear as those datapoints clustered around the centroids, or these datapoints that have features of two adjacent clusters and therefore are ambiguous.

Clustering Results

As described in the earlier sections, the clustering is based on the daily delay profile images. Thus, the spatial-temporal feature of daily delay distribution can be represented. After processing the image feature recognition algorithm Resnet 50 and dimensionality reduction, the daily delay patterns of 79 days are clustered into five types, as shown in Table 3. For each cluster, the centroid is calculated iteratively in the k-means algorithm until stable. We define the image closest to the cluster centroid as a "centroid image," which can best represent the feature of the cluster members. The corresponding images of the cluster centroid and elements in each cluster are listed in the Appendix, followed by the analysis of each centroid image. The capital letter with a number in Table 3"spatial imbalance distribution" denotes the auxiliary lines in the Appendix figures, representing the platforms' locations or times with significant delay dynamics.

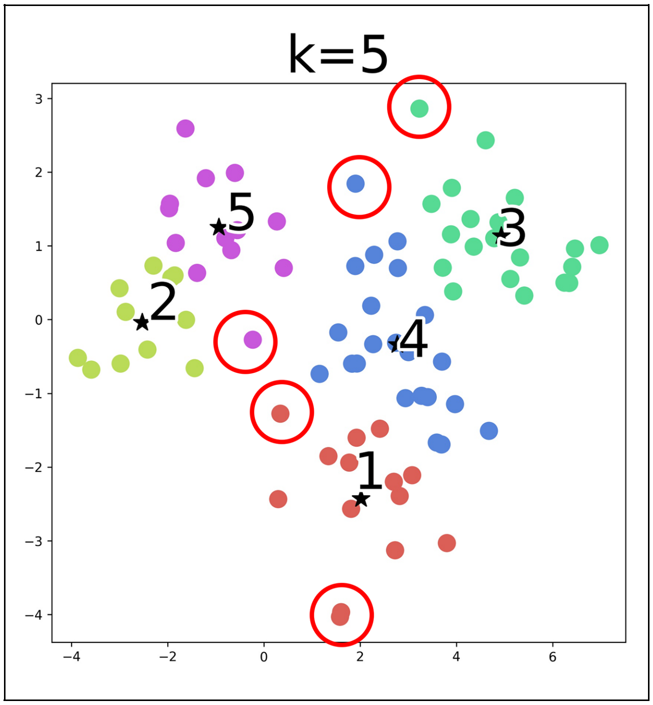

Figure 9. Clustering results with $k = 5$. The black stars denote the centroid of each cluster. The distance between clusters 2 and 5 and among clusters 1, 3, and 4 is relatively apparent. The red circles indicate the latent outliers (color online only).

The most significant feature difference among the five types of delay patterns is that each pattern has a specific day-of-week distribution. We found that there exist two main types of delay patterns: "weekend delay" (represented by clusters 2 and 5) and "weekday delay" (represented by clusters 1, 3, and 4). This feature can be identified in Figure 10. The number and color in the blocks denote the number of days belonging to the specific day and cluster. Most of the daily delay patterns that occur on weekends belong to the "weekend delay," while most of the delay that happens on weekdays belongs to one in three "weekday delay" clusters. Also, we found that more than half of Monday's daily delay distribution has a similar feature as cluster 3, and clusters 1 and 4 occur more frequently on Thursday and Friday.

Looking at the cluster-level features, we found a significant feature difference among the five types of delay patterns. These clusters have distinctive combinations of imbalance distribution on space, time, line directions, and corresponding delay severity. The feature differences are described in Table 3.

Moreover, the percentile values of the average daily punctuality of each cluster are illustrated in Figure 11. It is obvious that the "weekend delay" (clusters 2 and 5) have a higher delay time than the "weekday delay" (clusters 1, 3, and 4) on all percentiles, as well as a higher average daily delay. From this, we can conclude that the "weekend delay" patterns generally exhibit relatively more serious delays than the "weekday delay." Combined with Figure 12, we can see that there is a distinct periodic fluctuation in daily punctuality. The daily average punctuality has a significant peak almost every weekend and reaches a valley almost every Monday.

According to the analysis in this section, the prominent findings can be summarized as follows.

1) There are five types of daily delay patterns with distinct characteristics for tram line 1.
2) The delays on Monday, Saturday, and Sunday have significant distinct daily delay patterns. Weekends usually have more severe delays than weekdays. Tram line 1 is the most punctual on Monday, and least punctual on Sunday with a higher possibility of severe delay hampering large areas.

Table 3. Clustering Results and Attributes of the Delay Pattern of Each Cluster

Class	"Weekday delay"			"Weekend delay"	
Cluster (count)	Cluster 1 (14)	Cluster 3 (11)	Cluster 4 (22)	Cluster 2 (20)	Cluster 5 (13)
Feature of the delay distribution					
Temporal imbalance distribution	The heavy delay occurs during the evening peak hours	The heavy delay occurs between the morning peak end and evening peak end, from 9 a.m. to 7 p.m.	The heavy delay is evenly distributed all day	The heavy delay is evenly distributed all day	The delay occurs at intervals
Directional imbalance distribution	Heavier delay on direction 1	Slightly heavier delay on direction 1	Heavier delay on direction 1	Heavier delay on direction 1	More delay in direction 1; more early arrivals in direction 2
Spatial imbalance distribution	Always punctual between C2 and C2 (Delft)	Evenly distributed	Always delayed near Den Haag Kurhaus terminal (Den Haag, Scheveningen) More early arrivals between C1 and D (Delft)	Always delayed between A1 and Den Haag Kurhaus terminal (Den Haag)	Always arrive on time between C2 and D (Delft); always delayed between A1 and Den Haag Kurhaus terminal (Den Haag)
The mean of statistics of punctuality					
Average daily delay	28	3	13	48	45
SD	123.93	100.47	109.67	119.15	113.11
25%	−39	−52	−45	−18	−16
50%	18	6	11	34	34
75%	85	54	65	100	98
Day type distribution					
Workday	92.9%	100%	100%	18.2%	7.7%
Weekend	7.1%	0%	0%	81.8%	92.3%
Significant frequently	Even on weekdays	Monday	Even on weekdays	Sunday	Saturday

Note: SD = standard deviation.

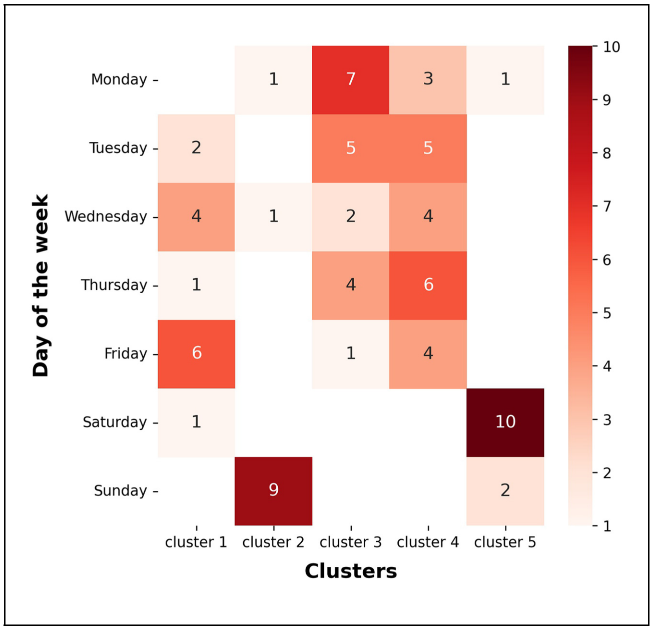

Figure 10. Cluster distribution on the day of the week.

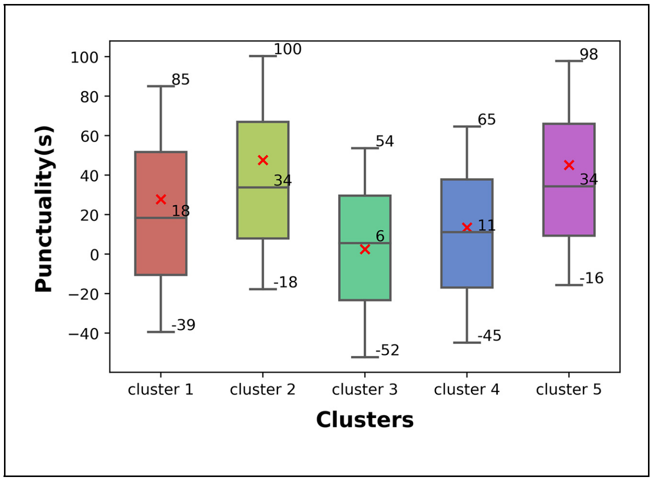

Figure 11. Distribution of daily punctuality values in each cluster. The lines within each box from top to bottom represent the 75%, 50%, and 25% percentile of punctuality. The red dots denote the mean value of average daily punctuality for each cluster (color online only).

3) Direction 1 (from "Den Haag Kurhaus" to "Delft, Tanthof") usually has a more serious delay than direction 2 (from "Delft, Tanthof" to "Den Haag Kurhaus"). Direction 2 has earlier arrival than direction 1.
4) The tram usually arrives earlier at stops in Delft than at stops in the Hague.
5) A latent boundary on the daily delay profile exists at the location of stop "Den Haag Frankenslag" in both directions, where the location is also the approximate boundary between the Den Haag Scheveningen (near the beach and the terminal of the tram line) and Den Haag downtown area. The spread of delay may be disturbed (intensified or weakened). This means the delay pattern of the Den Haag Scheveningen and Den Haag downtown area is different. The same boundary also exists at Delft station. This can be caused by the onboard/arriving passengers at those stops, especially at transit hubs, or caused by the land use pattern, which leads to less or more delay.

Conclusion

In this paper, we implement *k*-means clustering on the daily punctuality information of tram line 1 in The Hague. The patterns of daily delay distribution are detected, extracted, and clustered according to the proposed methodology. The case study results indicate that 79 days of daily delay profile images can be clustered into five clusters, corresponding to five types of delay patterns on the tram line 1, with different delay distribution features, distribution on the day of the week, and severity.

The data-driven explorative analysis proposed in this research can make significant contributions to PT operators and planners. Firstly, this analytical approach holds the potential to greatly enhance the operator's fundamental understanding of the recurring nature of delay along specific PT lines. Different types of delay patterns could offer insights into the likelihood of delays manifesting at distinct times and locations throughout the day. Alternately, certain patterns might suggest a more even distribution of delays across the entire line or day. Armed with this newfound understanding of delay characteristics, operators can intricately refine PT management strategies tailored to the unique delay patterns observed on different days. These strategies can then be put into action and assessed for their efficacy. Besides, clustering could provide the researchers with prior knowledge of typical delay patterns of PT networks. Further exploration of the causation of delay occurrence could be conducted based on the revealed pattern. Furthermore, the proposed methodology can easily be extended for other transit lines and other networks with GTFS data and AVL data and can explore and extract more abstract delay pattern characteristics.

This study does exhibit certain limitations that warrant acknowledgment. When translating the punctuality delay profile (Figure 3) into the aggregated delay profile image (Figure 4), the process of aggregating hourly delays experienced at each stop might inadvertently lead to the blending of delays from individual tram vehicles through averaging. Consequently, while the methodology sheds light on the broader macroscopic delay

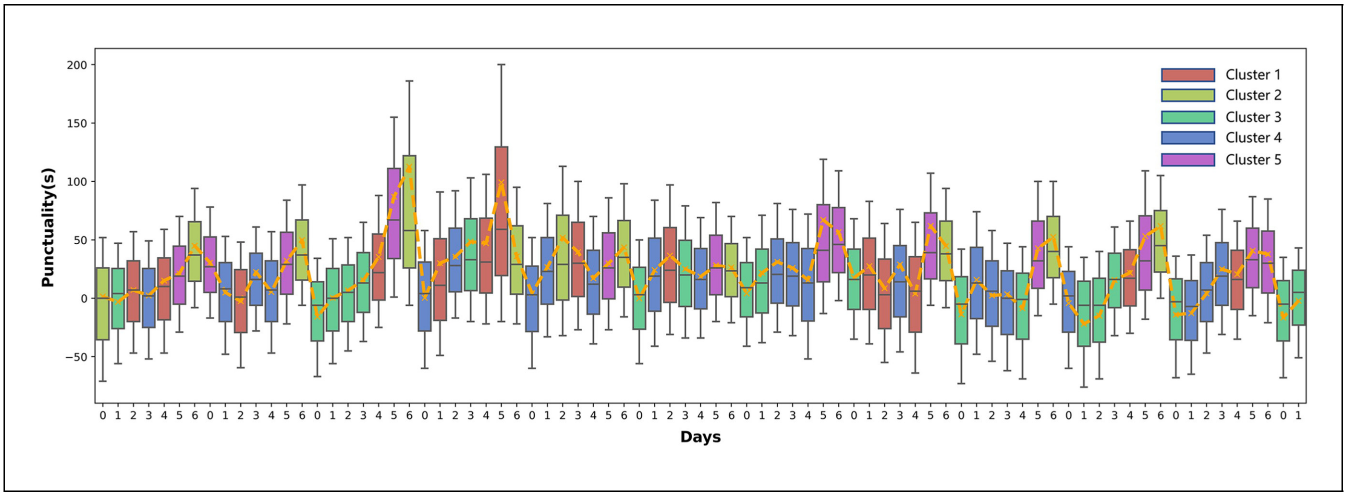

Figure 12. The daily delay pattern and punctuality statistics of each day. The orange dash line denotes the mean of daily punctuality. The color of boxplots denotes the clustering result of each day. The *x*-axis denotes the day of the week ("0" represents Monday), while the *y*-axis denotes the punctuality value in seconds (color online only).

distribution, it might overlook variations in delays among different vehicles. To mitigate this potential problem in practice, different temporal aggregation intervals could be selected based on the specific headways observed during the operation of a single line. In addition, while the *k*-means algorithm stands as a valuable tool for clustering, it is not the sole option available. We found that the clustering result in this research has ambiguity because of the outliers at the boundaries of the clusters. More advanced techniques, such as density-based clustering algorithms (DBSCANs), could potentially offer a solution, enhancing the clarity and reliability of the clustering results. Lastly, it is crucial to highlight that the present analysis primarily offers a qualitative assessment. While it provides PT operators with a foundational understanding of delay characteristics, it may be better suited for generating general insights rather than serving as a direct practical operation guide.

We envision four potential directions for further research. Firstly, the integration of automatic passenger count (APC) data holds the potential for estimating and analyzing average passenger delays on a per-passenger basis. The investigation of the pattern of passenger delay would provide a closer approximation of the actual delay experienced by commuters during their journeys. Secondly, there is room to extend our methodology beyond the line level to encompass network-wide analyses. This expansion would facilitate a broader understanding of delay propagation within a city or regional scale. By embracing this approach, operators can gain insights applicable to wider transport network management, potentially guiding the formulation of strategies for PT management advice on a larger scale. Thirdly, to provide a more quantitative assessment of each delay pattern, the adoption of a more interpretable clustering algorithm is warranted. Beyond recognizing distinct delay patterns, PT operators stand to benefit from understanding the intricate specifics of these patterns. A more interpretable clustering and image pattern recognition algorithm will make a clearer connection between the criterion of image cluster decision and delay characteristics, by providing a statistical description of these characteristics. Finally, future research could seek the possibility of complementing the delay profiles together with other data sources, which can provide more insights into delay profiles arising from internal and external causes.

Acknowledgments

We would like to thank NDOV (https://ndovloket.nl) for providing the AVL data.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: P. Krishnakumari, Y. Cheng; data collection: P. Krishnakumari; analysis and interpretation of results: Y. Cheng; draft manuscript preparation: Y. Cheng. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Yuxing Cheng https://orcid.org/0000-0001-7576-4775
Panchamy Krishnakumari https://orcid.org/0000-0003-3396-6314

Supplemental Material

Supplemental material for this article is available online.

References

1. Cats, O., and E. Jenelius. Dynamic Vulnerability Analysis of Public Transport Networks: Mitigation Effects of Real-Time Information. *Networks and Spatial Economics*, Vol. 14, No. 3, 2014, pp. 435–463. https://doi.org/10.1007/s11067-014-9237-7.
2. Park, Y., J. Mount, L. Liu, N. Xiao, and H. J. Miller. Assessing Public Transit Performance Using Real-Time Data: Spatiotemporal Patterns of Bus Operation Delays in Columbus, Ohio, USA. *International Journal of Geographical Information Science*, Vol. 34, No. 2, 2020, pp. 367–392. https://doi.org/10.1080/13658816.2019.1608997.
3. Zhang, H., Y. Liu, B. Shi, J. Jia, W. Wang, and X. Zhao. Analysis of Spatial-Temporal Characteristics of Operations in Public Transport Networks Based on Multisource Data. *Journal of Advanced Transportation*, Vol. 2021, 2021, p. e6937228. https://doi.org/10.1155/2021/6937228.
4. Luo, D., L. Bonnetain, O. Cats, and H. van Lint. Constructing Spatiotemporal Load Profiles of Transit Vehicles with Multiple Data Sources. *Transportation Research Record: Journal of the Transportation Research Board*, 2018. 2672: 175–186.
5. Degeler, V., L. Heydenrijk-Ottens, D. Luo, N. Oort, and J. W. C. Lint. Unsupervised Approach towards Analysing the Public Transport Bunching Swings Formation Phenomenon. *Public Transport*, Vol. 13, 2021, pp. 1–23. https://doi.org/10.1007/s12469-020-00251-z.
6. Wong, J. Leveraging the General Transit Feed Specification for Efficient Transit Analysis. *Transportation Research Record: Journal of the Transportation Research Board*, 2013. 2338: 11–19.
7. Szymanski, P., M. Zolnieruk, P. Oleszczyk, I. Gisterek, and T. Kajdanowicz. Spatio-Temporal Profiling of Public Transport Delays Based on Large-Scale Vehicle Positioning Data from GPS in Wrocław. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 11, 2018, pp. 3652–3661. https://doi.org/10.1109/TITS.2018.2852845.
8. Krishnakumari, P., O. Cats, and H. van Lint. Estimation of Metro Network Passenger Delay from Individual Trajectories. *Transportation Research Part C: Emerging Technologies*, Vol. 117, 2020, p. 102704. https://doi.org/10.1016/j.trc.2020.102704.
9. Liu, T. L. K., P. Krishnakumari, and O. Cats. Exploring Demand Patterns of a Ride-Sourcing Service Using Spatial and Temporal Clustering. *Proc., 6th International Conference on Models and Technologies for Intelligent Transportation Systems*, Cracow, Poland, 2019.
10. Bapaume, T., E. Côme, J. Roos, M. Ameli, and L. Oukhellou. Image Inpainting and Deep Learning to Forecast Short-Term Train Loads. *IEEE Access*, Vol. 9, 2021, pp. 98506–98522. https://doi.org/10.1109/ACCESS.2021.3093987.
11. Nguyen, T. T., P. Krishnakumari, S. C. Calvert, H. L. Vu, and H. van Lint. Feature Extraction and Clustering Analysis of Highway Congestion. *Transportation Research Part C: Emerging Technologies*, Vol. 100, 2019, pp. 238–258. https://doi.org/10.1016/j.trc.2019.01.017.
12. Agard, B., V. P. Nia, and M. Trépanier. Assessing Public Transport Travel Behaviour from Smart Card Data with Advanced Data Mining Techniques. 2013, p. 13. https://api.semanticscholar.org/CorpusID:145042381.
13. MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. In *Fifth Berkeley Symposium on Mathematics. Statistics and Probability*, University of California Press, Berkeley, CA, 1967, pp. 281–297.
14. Hopkins, B., and J. G. Skellam. A New Method for Determining the Type of Distribution of Plant Individuals. *Annals of Botany*, Vol. 18, No. 2, 1954, pp. 213–227. https://doi.org/10.1093/OXFORDJOURNALS.AOB.A083391.
15. Banerjee, A., and R. N. Dave. Validating Clusters Using the Hopkins Statistic. *Proc., IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, No. 1, Budapest, Hungary, IEEE, New York, 2004, pp. 149–153 vol.1.
16. Aggarwal, C. C.Cluster Analysis. In *Data Mining: The Textbook* (C. C. Aggarwal, ed.), Springer International Publishing, Cham, Switzerland, pp. 153–204.
17. Rousseeuw, P. J. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, Vol. 20, 1987, pp. 53–65. https://doi.org/10.1016/0377-0427(87)90125-7.
18. Giraud, C. Introduction to High-Dimensional Statistics CRC Press, Boca Raton, FL, 2021, p. 361.
19. Solem, J. E. Programming Computer Vision with Python: Tools and Algorithms for Analyzing Images. O'Reilly Media, Inc., Sebastopol, CA, 2012.
20. Picon, A., A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes. Deep Convolutional Neural Networks for Mobile Capture Device-Based Crop Disease Classification in the Wild. *Computers and Electronics in Agriculture*, Vol. 161, 2019, pp. 280–290. https://doi.org/10.1016/j.compag.2018.04.002.
21. Guérin, J., and B. Boots. Improving Image Clustering with Multiple Pretrained CNN Feature Extractors. *arXiv Preprint arXiv:1807.07760*, 2018.
22. Asano, Y. M., C. Rupprecht, and A. Vedaldi. Self-Labelling via Simultaneous Clustering and Representation Learning. *arXiv Preprint arXiv:1911.05371*, 2019.
23. He, K., X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *Proc., IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, NV, 2016.

24. van der Maaten, L., and G. Hinton. Visualizing Data Using T-SNE. *Journal of Machine Learning Research*, Vol. 9, No. 86, 2008, pp. 2579–2605.

25. Van Der Maaten, L., E. Postma, J. Van den Herik, Dimensionality Reduction: A comparative Review. *Journal of Machine Learning Research*, Vol. 10, No. 66–71, 2009, p. 13.

26. Noor Mathivanan, N. M., N. A. Md.Ghani, and R. Mohd Janor. A Comparative Study on Dimensionality Reduction between Principal Component Analysis and K-Means Clustering. *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 16, No. 2, 2019, p. 752. https://doi.org/10.11591/ijeecs.v16.i2.pp752-758.