

## Article

# An Adaptive Machine Learning Approach to Sustainable Traffic Planning: High-Fidelity Pattern Recognition in Smart Transportation Systems

Vitaliy Pavlyshyn <sup>1</sup> , Eduard Manziuk <sup>1</sup> , Oleksander Barmak <sup>1</sup> , Pavlo Radiuk <sup>1,\*</sup>  and Iurii Krak <sup>2,3</sup> 

<sup>1</sup> Department of Computer Science, Khmelnytskyi National University, 11 Instytuts'ka Street, 29016 Khmelnytskyi, Ukraine; vitaliy@ualeaders.com (V.P.); manziuk.e@khnmu.edu.ua (E.M.); barmako@khnmu.edu.ua (O.B.)

<sup>2</sup> Department of Theoretical Cybernetics, Taras Shevchenko National University of Kyiv, 4d Akademika Glushkova Ave, 03680 Kyiv, Ukraine; iurii.krak@knu.ua

<sup>3</sup> Laboratory of Communicative Information Technologies, V.M. Glushkov Institute of Cybernetics, 40 Akademika Glushkova Ave, 03187 Kyiv, Ukraine

\* Correspondence: radiukp@khnmu.edu.ua; Tel.: +380-97-854-9146

## Abstract

Effective and sustainable planning for future smart transportation systems is hindered by outdated traffic management models that fail to capture real-world dynamics, leading to congestion and significant environmental impact. To address this, advanced machine learning models are required to provide high-fidelity insights into urban mobility. In this work, we propose an adaptive machine learning approach to traffic pattern recognition that synergizes the HDBSCAN and k-means clustering algorithms. By employing a data-driven weighted voting mechanism, our solution provides a robust analytical foundation for sustainable planning, integrating structural analysis with precise cluster refinement. The crafted model was validated using a high-fidelity simulation of the Khmelnytskyi, Ukraine, transport network, where it demonstrated a superior ability to identify distinct traffic modes, achieving a V-measure of 0.79–0.82 and improving cluster compactness by 10–14% over standalone algorithms. It also attained a scenario identification accuracy of 92.8–95.0% with a temporal coherence of 0.94. These findings confirm that our adaptive approach is a foundational technology for intelligent transport systems, enabling the planning and deployment of more responsive, efficient, and sustainable urban mobility solutions.

**Keywords:** machine learning; sustainable planning; smart transportation; intelligent transport systems (ITS); sustainable mobility; traffic flow modeling; urban traffic management; data clustering; fuel consumption and emissions



Academic Editor: Zhihong Yao

Received: 8 September 2025

Revised: 4 October 2025

Accepted: 7 October 2025

Published: 28 October 2025

**Citation:** Pavlyshyn, V.; Manziuk, E.; Barmak, O.; Radiuk, P.; Krak, I. An Adaptive Machine Learning Approach to Sustainable Traffic Planning: High-Fidelity Pattern Recognition in Smart Transportation Systems. *Future Transp.* **2025**, *5*, 152. <https://doi.org/10.3390/futuretransp5040152>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The transition toward smart cities is fundamentally tied to solving the challenge of sustainable urban mobility. As metropolitan areas expand, increasingly complex traffic flows present a major barrier to achieving environmental sustainability, economic vitality, and public well-being. Conventional traffic management, reliant on static, pre-scheduled signal control, is ill-equipped for the dynamic reality of a smart city. These legacy systems cannot adapt to real-time changes in traffic demand, resulting in systemic issues such as chronic congestion, excessive fuel consumption, and elevated greenhouse gas emissions. Mitigating these inefficiencies is critical for developing resilient and sustainable urban environments. This necessitates a paradigm shift toward intelligent transport systems

(ITS) powered by machine learning, capable of perceiving, modeling, and dynamically responding to the state of the transport network. A cornerstone of such systems is high-fidelity traffic pattern recognition, i.e., the ability to automatically and accurately model the network's distinct operational modes.

This paper proposes an adaptive cascade clustering approach as an enabling solution for sustainable transportation planning and modeling. By synergistically integrating the complementary strengths of density-based and centroid-based clustering algorithms, our approach provides a robust and automated machine learning foundation for the next generation of intelligent, responsive, and eco-conscious urban transport control systems.

### 1.1. State of the Art

The development of sustainable urban mobility is a central pillar of the smart city vision, demanding a new generation of intelligent transport systems that are safe, resilient, and environmentally friendly [1,2]. Foundational to this goal is the capacity to accurately model and predict traffic dynamics using real-time data and sophisticated machine learning techniques. Research emphasizing the importance of time-evolving mobility patterns for predictive tasks confirms the need to capture temporal dynamics in any traffic analysis [3]. Our study contributes directly to this objective by proposing an advanced unsupervised learning approach that employs cascade clustering and weighted voting to deconstruct complex traffic flows into their constituent patterns, thereby overcoming the limitations of traditional, monolithic analytical methods used in transport planning.

Unsupervised clustering is a cornerstone of modern traffic analysis, offering a way to uncover latent structures in mobility data without pre-labeled examples. However, many existing methods show limitations when applied to sustainable planning in a smart city context. For instance, hybrid approaches combining k-medoids with spectral clustering, while accurate, can be sensitive to initialization, and their geometric assumptions often fail to model heterogeneous urban traffic data [4]. Similarly, spatially constrained hierarchical clustering has improved forecasts in bike-sharing systems, yet its reliance on fixed spatial constraints makes it less adaptable to the fluid nature of vehicular traffic [5]. More recent machine learning advancements, such as Bayesian ensembles [6] and self-learning clustering schemes [7], have enhanced performance but often introduce significant model complexity. This can obscure interpretability and demand substantial computational resources, limiting their feasibility for real-time applications in sustainable transport management. Inspired by the proven efficacy of ensemble methods in traffic analysis [8,9], our research focuses on developing a more adaptive, lightweight, and automated approach to modeling hidden patterns [10,11] in dynamic urban environments [12,13]. Our approach differs from those that create static typologies of road infrastructure, as our goal is to model the dynamic, time-varying operational modes of the entire network for better planning [14].

A critical application of advanced traffic analysis is in mitigating the transport sector's environmental footprint, a key objective of sustainable urban mobility. Technologies integral to the smart city, such as machine learning, the Internet of Things (IoT), and decentralized control, provide powerful tools for the real-time monitoring and intelligent traffic management essential for minimizing environmental harm [15,16]. Comprehensive sensor networks and dynamic traffic signal adjustment algorithms are also proving instrumental in this effort [17,18]. Modern analytical methods have successfully established the link between traffic flow patterns and vehicle emission levels [19–21]. Our prior research laid the groundwork for this study by demonstrating the utility of cluster analysis for traffic pattern identification [19] and developing foundational designs for environmentally oriented transport management systems [22]. This study directly addresses a key limitation of that work, i.e., its reliance on a single, pre-selected clustering algorithm, by introducing

an adaptive cascade approach that synergizes the density-based approach of Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) with the centroid-based approach of k-means. This synergy enables a more robust and automated modeling of transport modes, which is critical for designing intelligent traffic management strategies that reduce vehicular pollution. Trend-forecasting for network resilience recovery in complex traffic systems has been studied in [23], where resilience trends inform adaptive post-disruption strategies. Our unsupervised pattern recognition is complementary: it supplies high-fidelity regime labels that can act as state inputs to resilience forecasting and recovery planning.

### 1.2. Objectives and Tasks

A significant barrier to creating truly adaptive and sustainable urban mobility is the absence of analytical methods that can accurately model traffic patterns and automatically determine their optimal number and structure without expert intervention. Achieving this automation is essential for developing the intelligent planning and management systems required for the complex, evolving transport networks of modern smart cities.

The primary goal of this study is to advance sustainable urban mobility by developing and validating an adaptive machine learning approach for the automated modeling of traffic modes and their spatiotemporal relationships. To achieve this, we undertake the following key tasks:

1. Design a novel cascade clustering architecture that synergistically combines the robust, density-based structure detection of the HDBSCAN algorithm with the efficient boundary refinement of the k-means algorithm, using an informed initialization strategy to enhance modeling performance.
2. Develop a sophisticated weighted voting mechanism that automatically selects the optimal clustering result from the cascade's candidate solutions based on a composite of internal and external quality criteria, ensuring adaptability for diverse modeling scenarios.
3. Construct a comprehensive, multivariate feature representation for time-windowed traffic data that captures both static properties (e.g., average speed) and dynamic characteristics (e.g., variability and temporal correlations) to provide a rich input for the clustering models.
4. Rigorously validate the proposed adaptive approach through controlled simulation experiments, comparing its modeling performance against baseline algorithms on a reference dataset with known ground-truth scenarios using a balanced suite of validation metrics.

This work is predicated on the hypothesis that improving the structural quality and semantic accuracy of traffic mode clustering will directly enable more effective traffic signal regulation for sustainable planning. This leads to tangible benefits such as reduced vehicle emissions, decreased congestion, and shorter travel times, contributing to the development of smarter, more resilient urban infrastructure [24,25]. We posit that results from different clustering paradigms can be intelligently combined via a weighted voting mechanism to automatically select the optimal modeling outcome.

### 1.3. Motivation and Contributions

This study is motivated by the urgent need for intelligent, eco-friendly, and safe transportation infrastructures as part of sustainable planning in smart cities. Traffic intensification paired with outdated control systems contributes to significant environmental damage, including increased vehicle emissions [26]. Legacy traffic management systems

are unable to adapt to dynamic conditions, leading to inefficient flow patterns like excessive idling and stop-and-go traffic that amplify vehicular air and noise pollution [27].

The stated goal is planned to be achieved by developing an automated and adaptive machine learning approach to high-fidelity traffic pattern recognition. Accurate modeling of these patterns is directly linked to mitigating environmental impact, as it enables intelligent systems to optimize traffic signals, thereby minimizing inefficient driving modes for better sustainability outcomes [25]. Building upon our foundational research [19,22], this work introduces a more advanced and automated approach to data clustering. Our primary contribution is a novel adaptive architecture with a sophisticated weighted voting mechanism tailored for intelligent traffic modeling and management. While hybrid clustering has been explored in other transport domains [5], our approach is distinct. It avoids the computational intensity of Bayesian ensembles [6] and the rigid geometric assumptions of spectral methods [4] by marrying the robustness of density-based clustering with the efficiency of centroid-based refinement, using data-driven metrics to guide the fusion.

The key scientific and technical contributions of this research are:

- A novel cascade clustering architecture: We propose an architecture that synergizes the structural detection capabilities of HDBSCAN with the boundary refinement of k-means, enhanced by an informed initialization strategy to improve the accuracy and stability of traffic pattern models.
- A data-driven weighted voting mechanism: We introduce a mechanism for the automatic selection of the optimal clustering result based on a composite quality score, ensuring the model's adaptability and eliminating the need for manual algorithm selection.
- A refined multivariate feature model: We develop a comprehensive model that integrates both static and dynamic traffic metrics to create a rich and robust representation of network states for more nuanced pattern detection and modeling.

The remainder of this article is organized as follows. Section 1 has introduced the problem, reviewed the literature, and outlined the study's objectives and contributions. Section 2 details the proposed adaptive cascade clustering approach. Section 3 presents the experimental results from our simulation study. Section 4 discusses the implications of these results and evaluates the approach. Finally, Section 5 summarizes the key findings and suggests directions for future research.

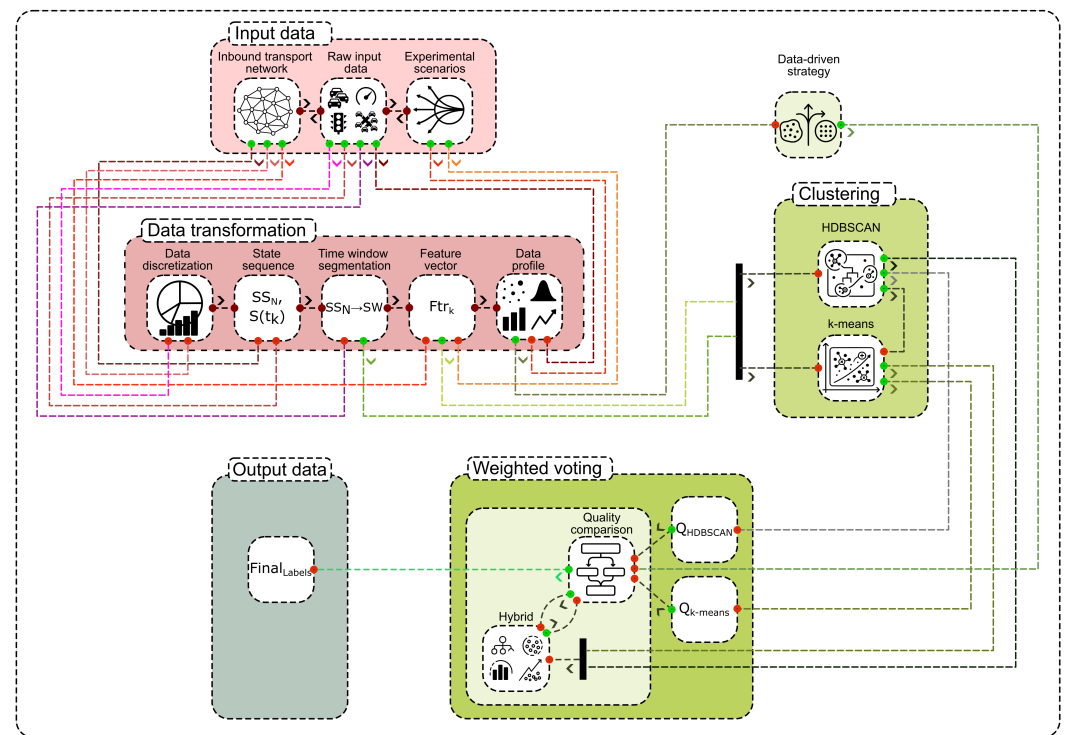
## 2. Materials and Methods

This section provides a comprehensive and detailed technical description of the proposed adaptive cascade clustering approach and the experimental methodology employed for its validation. We begin by detailing the architecture of the adaptive approach, which includes the formal models for data representation, the specific preprocessing techniques applied, and the process of multivariate feature extraction. Subsequently, we provide an in-depth elaboration of the core clustering algorithms, HDBSCAN and k-means, and explain the precise mechanics of the weighted voting mechanism that enables adaptive strategy selection. Finally, we outline the experimental setup, including the simulation environment, the design of the experimental scenarios, and the suite of performance evaluation metrics used to assess the quality of the results.

### 2.1. Adaptive Cascade Approach to Clustering

The proposed approach is structured as an adaptive cascade designed to systematically identify, analyze, and interpret urban traffic patterns from raw time-series data. The overall architecture of this approach is illustrated schematically in Figure 1. The process begins with the acquisition of raw data from the transport network (in this case, the Simulation

of Urban Mobility (SUMO) simulation), which is then transformed into a sequence of structured, high-dimensional feature vectors, with each vector representing the state of the network within a discrete time window.



**Figure 1.** The proposed adaptive cascade clustering architecture. The process commences with data acquisition from the urban transport network, which is then subjected to a feature extraction process within discrete time windows. A data-driven weighted voting mechanism then selects the optimal clustering strategy (HDBSCAN-first or k-means-first) based on the intrinsic characteristics of the data, leading to the final, high-fidelity identification of distinct traffic patterns.

A central and novel element of our approach is an adaptive selection mechanism that intelligently chooses the most suitable clustering strategy, either starting with HDBSCAN, starting with k-means, or using a hybrid approach, based on the intrinsic properties of the data itself. This crucial decision is guided by a sophisticated weighted voting system that evaluates the potential performance and suitability of each algorithmic pathway. The final output of the pipeline is a set of labeled traffic patterns, which enables a detailed structural and temporal analysis of the city’s traffic dynamics. This section will now elaborate on the mathematical models for data representation, the specific configurations of the clustering algorithms, the metrics used for quality assessment, and the underlying logic of the adaptive strategy selection mechanism.

## 2.2. Data Generation and Simulation Environment

The methodological foundation of this study integrates the authenticity of empirical data with the rigor of a controlled simulation. The experimental scenarios were not synthetically constructed but were derived from a systematic, expert-led analysis of real-world traffic flows in Khmelnytskyi, Ukraine. Transportation experts identified a taxonomy of recurring traffic regimes (e.g., stable progression, bottleneck formation) by analyzing video surveillance footage and ground-count data from key intersections. This raw data is available in the repository cited in the Data Availability Statement.

This expert-validated taxonomy formed the basis for the ground-truth scenarios, which were meticulously reproduced within a high-fidelity digital twin of the city’s trans-

port network. The model was built and calibrated in the SUMO package v1.22.0 [28] using the empirical data to ensure its outputs accurately reflect authentic traffic dynamics. For this study, the simulation was run for a 22-h period, with key parameters such as vehicle speeds and queue lengths sampled at 10-min intervals, yielding a time series of 132 distinct observations.

This approach creates a controlled testbed to address a precise research question: “Which clustering route most reliably recognizes these empirically observed regimes?” By embedding known, real-world patterns, this design allows for a direct and objective quantitative assessment of algorithmic performance, isolating the repeatable behaviors most relevant for long-term planning while allowing episodic anomalies to be handled as noise by the density-based clustering stage.

### 2.3. Data Representation and Preprocessing

#### 2.3.1. Urban Transport Network Model

The foundational step of our analysis is the formal representation of the urban transport network as a directed graph, defined as:

$$G = (V, E), \quad (1)$$

where  $V$  is the set of vertices, representing the intersections or nodes of the network, and  $E$  is the set of directed edges, representing the road segments that connect them.

The state of this network is captured dynamically over a specified time interval  $[t_0, t_N]$ , resulting in a time series of network state snapshots:

$$SS_N = \{S(t_0), S(t_1), \dots, S(t_N)\}, \quad (2)$$

where each element  $S(t_k)$  is a comprehensive representation of the entire transport network's characteristics (e.g., vehicle speeds, traffic densities, queue lengths at intersections) at a specific time instance  $t_k$ .

The sequence  $SS_N$ , as defined in Equation (2), constitutes the raw dataset for all subsequent analysis. The length of this sequence,  $N$ , is determined by the total duration of the monitoring period and the data sampling rate,  $\Delta t$ .

#### 2.3.2. Time Window Segmentation

To apply machine learning techniques, which typically require structured input, to the continuous flow of traffic data, it is necessary to transform the unstructured time series  $SS_N$  into a suitable format. This is accomplished through a segmentation function  $\Phi : SS_N \rightarrow SW$ , which maps the original time series data into a structured sequence of discrete, non-overlapping time windows:

$$SW = \{W_1, W_2, \dots, W_K\}, \quad K \in \mathbb{N}, \quad (3)$$

where each window  $W_k$  represents a segment of the network's state over a fixed time interval of length  $\Delta t$ :

$$W_k = \{S(t) | t \in [t_0 + (k-1)\Delta t, t_0 + k\Delta t]\}. \quad (4)$$

This segmentation process, described formally by Equations (3) and (4), effectively organizes the raw data into a series of meaningful fragments. Each fragment, or time window, characterizes the aggregate behavior of the transport network over a specific and well-defined period, making it amenable to feature extraction and subsequent pattern analysis.

### 2.3.3. Feature Vector Extraction

For each time window  $W_k$ , a compact and informative vector representation must be constructed to capture its essential characteristics. This feature vector, denoted  $Ftr_k$ , is designed to include both the static and dynamic properties of the traffic flow within that window:

$$Ftr_k = (\mu_k, \sigma_k, \delta_k, \tau_k), \quad (5)$$

where  $\mu_k$  represents the vector of average states of traffic flows (e.g., mean speed, mean density),  $\sigma_k$  is the vector of standard deviations, providing a measure of variability,  $\delta_k$  represents the rate of change of these flows (their first derivative), capturing the trend, and  $\tau_k$  reflects the autocorrelation properties, indicating the temporal persistence of the traffic state.

To quantify the similarity between any two time windows,  $W_i$  and  $W_j$ , a Gaussian kernel (also known as a Radial Basis Function kernel) is employed. This is a popular choice due to its ability to handle non-linear relationships in the feature space:

$$\text{sim}(W_i, W_j) = \exp\left(-\frac{\|Ftr_i - Ftr_j\|^2}{2\sigma_{\text{global}}^2}\right), \quad (6)$$

where  $\|\cdot\|$  denotes the Euclidean norm and the scaling parameter  $\sigma$  is typically set to a fraction of the global standard deviation of the dataset features,  $\sigma_{\text{global}}$ .

This choice of scaling ensures that the similarity measure is robust and well-behaved across the entire dataset.

The collection of all such feature vectors is then aggregated to form a feature matrix

$$F = [Ftr_1, Ftr_2, \dots, Ftr_K]^T,$$

which has dimensions  $K \times d$ , where  $K$  is the number of time windows and  $d$  is the dimensionality of the feature space.

This matrix serves as the final, structured input for the clustering stage of our pipeline.

### 2.3.4. Strategies for Mitigating High Dimensionality

A significant challenge in traffic data analysis arises from the “curse of dimensionality” when preserving detailed, intersection-level data. This high-dimensional feature space can degrade the performance of clustering algorithms by making distance metrics less meaningful. To address this, our proposed adaptive approach is designed to incorporate dimensionality mitigation as a key preprocessing step. It can employ feature selection, using metrics like mutual information or Gini importance to retain only the most informative variables.

Alternatively, dimensionality reduction techniques can be applied. Principal Component Analysis (PCA) offers a linear method for projecting data onto a lower-dimensional subspace while preserving maximal variance. For capturing more complex, non-linear relationships, manifold learning techniques such as Uniform Manifold Approximation and Projection (UMAP) are more suitable, as UMAP excels at preserving both the local and global structure of the data in its low-dimensional embedding. The choice between these strategies (i.e., feature selection, PCA, or UMAP) can be integrated into the adaptive approach, guided by an initial profiling of the data’s linearity and intrinsic dimensionality.

However, for the specific experiments reported in this study, clustering was intentionally performed directly on the raw high-dimensional data. This was a deliberate choice to rigorously evaluate the baseline performance and robustness of the algorithms under these challenging conditions. Consequently, the mitigation techniques described here were

not applied prior to the clustering analysis in this work; PCA was used exclusively for the two-dimensional visualization of the results.

## 2.4. Core Clustering Algorithms

### 2.4.1. Synergistic Selection of Clustering Paradigms

The selection of HDBSCAN and k-means is a strategic choice grounded in their complementary nature, representing two fundamental clustering paradigms: density-based and centroid-based. This synergy allows our cascade architecture to adapt to diverse data characteristics, which is essential for analyzing complex urban traffic flows.

HDBSCAN, a density-based algorithm, serves as the initial structure-discovery engine. Its ability to identify clusters of arbitrary shape and automatically determine the number of traffic modes is crucial for analyzing urban systems with irregular patterns. Furthermore, its inherent robustness to noise and outliers is vital for real-world applications. Conversely, k-means, a centroid-based algorithm, provides computational efficiency and produces compact, geometrically well-defined clusters. This is necessary for the practical implementation of traffic control systems, which require clear and stable traffic state definitions.

Our cascade architecture leverages this complementarity in a two-step process. First, HDBSCAN identifies the number of significant clusters and their dense core locations. This output then provides an informed initialization for k-means, which acts as a boundary-refinement engine to precisely delineate cluster boundaries. This sequential pipeline allows our model to capture complex, non-linear traffic patterns while producing the stable and interpretable results that neither algorithm could achieve in isolation.

### 2.4.2. HDBSCAN with Automated Parameter Tuning

A key advantage of our implementation of HDBSCAN is the automated and data-driven tuning of its primary parameters, which enhances its adaptability and robustness across different datasets. The minimum cluster size parameter,  $mcs$ , which specifies the minimum number of points required to form a stable cluster, is calculated as follows:

$$mcs = \lceil N_{ob} \cdot s_{cl} \rceil, \quad (7)$$

where  $N_{ob}$  is the total number of observations (time windows) in the dataset and  $s_{cl}$  is a scaling factor, which is typically set within the range  $[0.02, 0.08]$  to ensure sensitivity to meso-scale patterns.

The cluster selection parameter, ‘min\_samples’ ( $ms$ ), which controls the algorithm’s conservatism in forming clusters by defining the minimum number of samples in a neighborhood for a point to be considered a core point, is derived from  $mcs$ :

$$ms = \lceil mcs \cdot \beta \rceil, \quad (8)$$

where  $\beta$  is a reduction factor, typically set between 0.5 and 0.8. This allows for a more flexible definition of density.

Finally, the cluster selection epsilon parameter,  $cse$ , which determines the maximum distance for joining points into clusters from the minimum spanning tree, is calculated based on the local data structure:

$$cse = \text{median}(KNN_{dist}) \cdot \gamma, \quad (9)$$

where  $KNN_{dist}$  is the array of distances to the  $k$  nearest neighbors (usually  $k = 5$ ) for each data point, and  $\gamma$  is a distance scaling factor, typically in the range  $[1.0, 1.5]$ .

The use of the median makes this calculation robust to outliers. This automated tuning process, governed by Equations (7)–(9), allows HDBSCAN to adapt its behavior to the specific characteristics of different datasets without requiring manual intervention. To ensure full reproducibility of our results, the parameter search scripts, which implement deterministic random-seed control, are made available in the public repository cited in the Data Availability Statement. A detailed sensitivity analysis of the model's performance to variations in the  $\beta$  and  $\gamma$  hyperparameters is provided in Appendix A.

#### 2.4.3. k-Means with Informed Initialization

The second stage of the cascade involves the application of the k-means algorithm to refine the cluster boundaries identified by HDBSCAN. This strategy synergistically combines the strengths of density-based clustering (robust structure detection) with the advantages of a centroid-based approach (creation of clear, compact boundaries). The k-means algorithm is executed with its key parameters derived directly from the output of the initial HDBSCAN analysis:

$$\text{k-means}(K = K_{\text{optimal}}, \text{init} = \text{HDBSCAN}_{\text{centroids}}), \quad (10)$$

where  $K_{\text{optimal}}$  is the number of significant clusters (i.e., non-noise clusters) that were identified by HDBSCAN, and  $\text{HDBSCAN}_{\text{centroids}}$  is the set of initial centroid locations for the k-means algorithm.

These initial centroids are calculated as the geometric centers (mean vectors) of the clusters obtained from the HDBSCAN stage:

$$c_i^{(0)} = \frac{1}{|C_i^{\text{HDBSCAN}}|} \sum_{x_j \in C_i^{\text{HDBSCAN}}} x_j, \quad (11)$$

where  $C_i^{\text{HDBSCAN}}$  is the set of data points belonging to the  $i$ -th cluster found by HDBSCAN.

This informed initialization strategy, formally defined in Equations (10) and (11), is a critical component of the cascade's success. By starting the k-means algorithm from locations that are already known to be within dense, stable regions of the data, it significantly reduces the risk of the algorithm converging to a poor local minimum and ensures that the final partitioning is a meaningful refinement of an already robust structural analysis.

### 2.5. Cluster Quality Assessment

#### 2.5.1. Geometric and Density-Based Metrics

After a clustering solution has been generated, a comprehensive set of quantitative characteristics is calculated for each identified cluster  $C_k$  to rigorously evaluate its quality. The centroid  $\text{Cnt}(C_k)$ , which represents the typical or average state of the traffic mode corresponding to that cluster, is computed as the geometric center of its constituent feature vectors:

$$\text{Cnt}(C_k) = \frac{1}{|C_k|} \sum_{W_i \in C_k} \text{Ftr}(W_i). \quad (12)$$

The cluster radius  $r(C_k)$  serves as a measure of its compactness by calculating the maximum Euclidean distance from the centroid to any point within the cluster:

$$r(C_k) = \max_{W_i \in C_k} \|\text{Ftr}(W_i) - \text{Cnt}(C_k)\|. \quad (13)$$

The cluster density  $\text{Dns}(C_k)$  quantifies the concentration of data points within the feature space occupied by the cluster:

$$Dns(C_k) = \frac{|C_k|}{Vol_{rad}(C_k)}, \quad (14)$$

where the volume  $Vol_{rad}(C_k)$  is that of a  $d$ -dimensional hypersphere with radius  $r(C_k)$ :

$$Vol_{rad}(C_k) = \frac{\pi^{d/2}}{\Gamma(d/2 + 1)} \cdot r(C_k)^d, \quad (15)$$

with  $d$  being the dimension of the feature space and  $\Gamma(\cdot)$  representing the gamma function.

The compactness  $Cmp(C_k)$  provides a measure of the internal homogeneity of the cluster by calculating the average pairwise similarity between all points within it:

$$Cmp(C_k) = \frac{1}{|C_k|(|C_k| - 1)} \sum_{W_i, W_j \in C_k, i \neq j} \text{sim}(W_i, W_j). \quad (16)$$

Finally, the separation  $Sep(C_k)$  measures how distinct and well-separated a cluster is from all other clusters by calculating the minimum distance to the centroid of any other cluster:

$$Sep(C_k) = \min_{j \neq k} \|Cnt(C_k) - Cnt(C_j)\|. \quad (17)$$

### 2.5.2. Stability and Coherence Metrics

To assess the robustness and reliability of the identified clusters, their stability is calculated. The stability of a cluster  $C_k$  evaluates its resilience to small perturbations in the data, which are typically introduced through techniques like bootstrap sampling:

$$\text{Stability}(C_k) = 1 - \frac{\sigma_{\text{centroid}}(C_k)}{\|Cnt(C_k)\|}, \quad (18)$$

where  $\sigma_{\text{centroid}}(C_k)$  is the standard deviation of the centroid's position across multiple bootstrap samples of the data.

A higher stability value, as defined in Equation (18), indicates a more reliable and well-defined transport mode that is less likely to be a statistical artifact.

Temporal coherence is a critical metric for interpreting traffic modes, as it measures the degree to which a cluster represents a contiguous and uninterrupted block of time:

$$\text{Coherence}(C_k) = \frac{1}{|C_k| - 1} \sum_{i=1}^{|C_k|-1} \mathbb{1}_{\text{consecutive}}(t_i, t_{i+1}), \quad (19)$$

where  $\mathbb{1}_{\text{consecutive}}(t_i, t_{i+1})$  is an indicator function that equals 1 if the time windows corresponding to observations  $i$  and  $i + 1$  (ordered chronologically within the cluster) are sequential in the original time series.

High coherence is a strong indicator of a long-term, stable mode of traffic behavior.

## 2.6. Adaptive Strategy Selection

### 2.6.1. Weighted Voting Mechanism

A key novelty of our approach is an automatic, data-driven weighted voting mechanism to select the optimal clustering result. The quality of the output from each algorithmic stage is evaluated using a composite metric. For the k-means stage, the quality metric emphasizes the geometric properties of the clusters:

$$\text{Quality}_{\text{k-means}} = \alpha \cdot \text{Silhouette} + \beta \cdot \text{Compactness} + \gamma \cdot \text{Separation}. \quad (20)$$

For HDBSCAN, the metric is designed to balance structural correctness, stability, and semantic value:

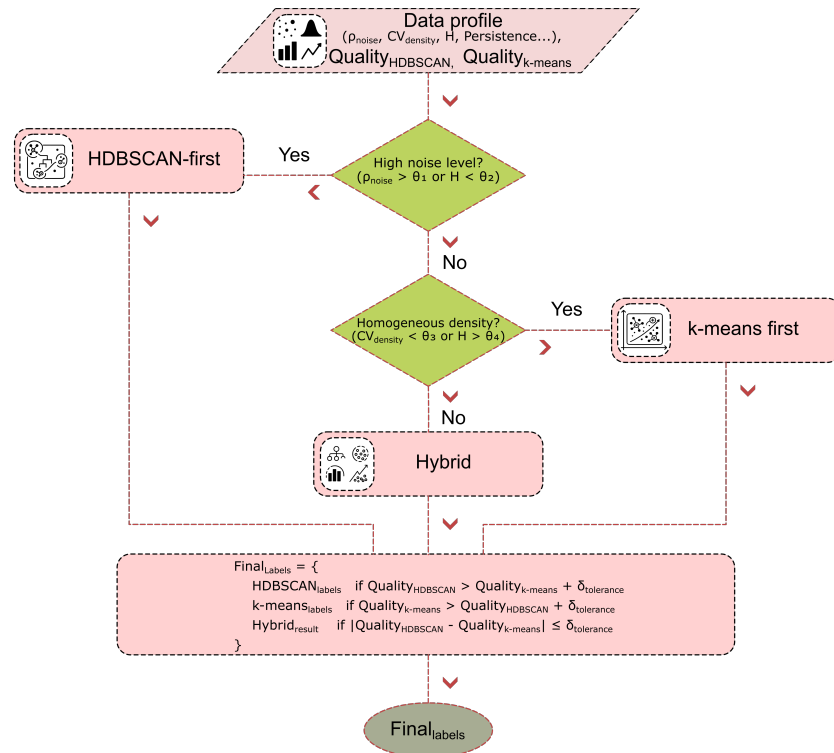
$$\text{Quality}_{\text{HDBSCAN}} = \alpha \cdot \text{Silhouette} + \beta \cdot \text{Stability} + \gamma \cdot \text{Interpretability}. \quad (21)$$

In Equation (21), Interpretability is formally quantified using the Temporal Coherence metric (defined in Equation (19)), which measures the chronological consistency of the identified clusters. This choice is critical because high coherence corresponds to stable, contiguous traffic modes, which are more meaningful and actionable for transport planning than patterns scattered randomly in time.

For this study, the weighting factors ( $\alpha, \beta, \gamma$ ) were set to 1/3 each for a balanced evaluation. To ensure system stability and prevent frequent switching between strategies due to minor performance fluctuations, we introduce a tolerance threshold,  $\delta_{\text{tolerance}}$  (typically set in the range [0.02, 0.05]). This leads to a more robust decision rule:

$$\text{Final}_{\text{labels}} = \begin{cases} \text{HDBSCAN}_{\text{labels}} & \text{if } \text{Quality}_{\text{HDBSCAN}} > \text{Quality}_{\text{k-means}} + \delta_{\text{tolerance}}; \\ \text{k-means}_{\text{labels}} & \text{if } \text{Quality}_{\text{k-means}} > \text{Quality}_{\text{HDBSCAN}} + \delta_{\text{tolerance}}; \\ \text{Hybrid}_{\text{result}} & \text{if } |\text{Quality}_{\text{HDBSCAN}} - \text{Quality}_{\text{k-means}}| \leq \delta_{\text{tolerance}}. \end{cases} \quad (22)$$

When quality scores are within this tolerance, a hybrid result is generated by fusing labels at the individual data point level. The final label for any point of disagreement is assigned based on a local confidence score (outlier score for HDBSCAN, distance to centroid for k-means), leveraging the local strengths of both methods. This refined process, formalized in Equation (22) and visualized in Figure 2, ensures the final clustering is the most robust and meaningful choice for the given data.



**Figure 2.** Logical scheme of the weighted voting and adaptive selection process. Characteristics of the input data, such as the noise ratio and density variation, are evaluated to inform the initial selection between HDBSCAN and k-means. The quality of both models is then assessed using a combination of internal and external validation metrics, and the final clustering result is chosen based on a comparative analysis, ensuring that the most suitable model is applied for the given data.

As for the weighting rationale and robustness, in Equations (20) and (21) we set  $\alpha = \beta = \gamma = \frac{1}{3}$ , following the principle of indifference when no empirical or domain evidence privileges geometric compactness, stability, or separability. Crucially, the tolerance threshold  $\delta_{\text{tolerance}}$  in Equation (22) prevents decision flips due to minor weight changes: if the quality scores are within  $\delta_{\text{tolerance}}$ , the system fuses labels instead of oscillating between strategies. Hence, final selections are driven by meaningful data differences rather than arbitrary weighting. Further details on weight sensitivity are provided in Appendix B.

### 2.6.2. Data Profiling for Strategy Switching

The choice of the optimal clustering strategy is highly dependent on the intrinsic structural characteristics of the input data. To automate this choice, we first quantify the noise level in the dataset,  $\rho_{\text{noise}}$ , as the proportion of outliers:

$$\rho_{\text{noise}} = \frac{|\text{outliers}|}{|\text{Data}|}. \quad (23)$$

Outliers are robustly identified using the interquartile range (IQR) method:

$$\text{outliers} = \{x_i : x_i < Q_1 - 1.5 \cdot \text{IQR} \text{ or } x_i > Q_3 + 1.5 \cdot \text{IQR}\}, \quad (24)$$

where  $Q_1$  and  $Q_3$  are the first and third quartiles of the data distribution.

The natural tendency of the data to form distinct clusters is assessed using the Hopkins statistic,  $H$ :

$$H = \frac{\sum_{i=1}^m v_i}{\sum_{i=1}^m u_i + \sum_{i=1}^m v_i}, \quad (25)$$

where  $v_i$  are the distances from real data points to their nearest neighbors and  $u_i$  are the distances from randomly generated points to their nearest neighbors in the real dataset; values of  $H$  close to 1 indicate a high degree of cluster separation.

The heterogeneity of cluster density is measured by the coefficient of variation of local densities:

$$CV_{\text{density}} = \frac{\sigma_{\text{density}}}{\mu_{\text{density}}}, \quad (26)$$

where the local density  $\rho_i$  for each point  $x_i$  is estimated based on its  $k$  nearest neighbors:

$$\rho_i = \frac{k}{\sum_{x_j \in kNN(i)} \|x_i - x_j\|}. \quad (27)$$

The temporal structure of the data is analyzed via the autocorrelation function  $R(\tau)$ , from which a time stability coefficient, Persistence, is derived. The data's internal complexity is assessed by estimating its intrinsic dimensionality  $d_{\text{intrinsic}}$  using a maximum likelihood approach, which is then used to compute a complexity ratio. These metrics are combined into a comprehensive data profile:

$$\text{Data}_{\text{profile}} = \{\rho_{\text{noise}}, H, CV_{\text{density}}, \text{Persistence}, \text{Complexity}_{\text{ratio}}\}. \quad (28)$$

This profile provides the basis for making an informed, automatic decision on the optimal clustering strategy.

### 2.6.3. Strategic Application Rules and Adaptive Learning

Based on the data profile, one of three main strategies is automatically selected. The HDBSCAN-first strategy is chosen for data with high noise ( $\rho_{\text{noise}} > 0.2$ ), high density variation ( $CV_{\text{density}} > 0.6$ ), or low separation ( $H < 0.3$ ), as HDBSCAN excels at handling

outliers and clusters of arbitrary shape. Conversely, the k-means-first strategy is applied to well-structured data with low noise ( $\rho_{\text{noise}} < 0.1$ ), homogeneous density ( $CV_{\text{density}} < 0.3$ ), and high separation ( $H > 0.7$ ), where the geometric optimization of k-means can provide clearer cluster boundaries. For intermediate cases, the hybrid strategy is used.

The entire approach also incorporates a mechanism for dynamic adaptation and learning, which allows it to improve its strategy selection over time by learning from historical performance:

$$\text{Strategy}_{\text{adaptive}} = \underset{s \in \{\text{HDBSCAN}, \text{k-means}, \text{hybrid}\}}{\text{argmax}} \text{Performance}(s | \text{Data}_{\text{profile}}). \quad (29)$$

The implementation of the adaptive learning mechanism in Equation (29) involves maintaining a performance history database. This database stores tuples of the form  $(\text{Data}_{\text{profile}}, \text{Strategy}, \text{PerformanceScore})$ . For a new, unseen dataset, its profile is computed and used to query this database. A k-nearest neighbor algorithm identifies the  $k$  most similar historical data profiles (using a weighted Euclidean distance on the profile vectors). The average performance score for each strategy (HDBSCAN, k-means, hybrid) across these  $k$  neighbors is then calculated. The strategy with the highest average historical performance for similar data types is selected. This allows the system to learn from its past experience and make increasingly robust and accurate decisions over time, ensuring the long-term feasibility and effectiveness of the system.

### 2.7. Implementation of the Adaptive Approach

The complete adaptive cascade clustering process is summarized in Algorithm 1.

Algorithm 1 proceeds in three main phases. Phase 1 involves data preparation, where raw time-series data is segmented into windows and transformed into feature vectors, followed by an analysis of the data's intrinsic properties to generate a data profile. In Phase 2, an adaptive clustering strategy is selected based on this profile, and both HDBSCAN and k-means (with informed initialization) are executed in the chosen sequence. In Phase 3, the weighted voting mechanism compares the quality of the two clustering results and selects or fuses them to produce the final set of labels. Validated clusters that meet predefined stability and size thresholds are identified as the final traffic patterns, and a transition matrix between these patterns is computed to model the system's dynamics.

### 2.8. Statistical Analysis Methods

To ensure the statistical rigor of our findings, all comparative analyses in this study employ appropriate statistical tests to validate the significance of the observed performance differences. For pairwise comparisons between the performance metrics of two different clustering approaches (e.g., HDBSCAN vs. k-means, or our cascade approach vs. a baseline), we use the non-parametric Wilcoxon signed-rank test. This test was chosen because the distribution of performance metrics like ARI or V-measure across different datasets or scenarios may not be normally distributed. All reported p-values are two-tailed, and a significance level of  $\alpha = 0.05$  is used as the threshold for statistical significance. In cases where multiple comparisons are performed, a correction method such as the Benjamini-Hochberg procedure would be applied to control the false discovery rate. Furthermore, to quantify the uncertainty associated with our performance estimates, we report 95% confidence intervals for key metrics where applicable, which are computed using bootstrap resampling techniques.

**Algorithm 1** Adaptive Cascade Clustering for Traffic Pattern Recognition.

---

**Require:** Traffic data  $D$ ; window parameters ( $w_{size}$ ,  $w_{step}$ ); thresholds ( $O_{stab}$ ,  $O_{val}$ ); tolerance  $\delta_{tolerance}$ .

**Ensure:** Traffic patterns  $P$ ; transition matrix  $T$ ; quality metrics  $Q$ .

- 1: Initialize:  $W \leftarrow \{\}, P \leftarrow \{\}$   $\triangleright$  Phase 1. Data Preparation & Analysis
- 2: **for**  $i = 1$  to  $N - w_{size}$  **step**  $w_{step}$  **do**
- 3:      $window \leftarrow D[i : i + w_{size}]$
- 4:      $features \leftarrow \text{ComputeFeatures}(window)$
- 5:      $W \leftarrow W \cup \{features\}$
- 6: **end for**
- 7:  $DataProfile \leftarrow \text{ComputeDataProfile}(W)$   $\triangleright$  Phase 2. Adaptive Clustering
- 8: **if**  $DataProfile$  suggests high noise or complex structure **then**
- 9:      $strategy \leftarrow \text{HDBSCAN\_first}$
- 10: **else if**  $DataProfile$  suggests low noise and simple structure **then**
- 11:      $strategy \leftarrow \text{k\_means\_first}$
- 12: **else**
- 13:      $strategy \leftarrow \text{hybrid}$
- 14: **end if**
- 15:  $L_h \leftarrow \text{HDBSCAN}(W, \text{auto\_params})$
- 16:  $L_k \leftarrow \text{k\_means}(W, |\text{unique}(L_h)|, \text{init} = \text{centroids\_from}(L_h))$
- 17:  $Q_h \leftarrow \text{EvaluateQuality}(W, L_h)$
- 18:  $Q_k \leftarrow \text{EvaluateQuality}(W, L_k)$   $\triangleright$  Phase 3. Weighted Voting & Validation
- 19: **if**  $Q_h > Q_k + \delta_{tolerance}$  **then**
- 20:      $L_{final} \leftarrow L_h$   $\triangleright$  HDBSCAN result is significantly better
- 21: **else if**  $Q_k > Q_h + \delta_{tolerance}$  **then**
- 22:      $L_{final} \leftarrow L_k$   $\triangleright$  k-means result is significantly better
- 23: **else**
- 24:      $L_{final} \leftarrow \text{HybridResult}(L_h, L_k)$   $\triangleright$  Scores are comparable; fuse results
- 25: **end if**
- 26: **for each** cluster  $C_i$  in  $L_{final}$  **do**
- 27:     **if**  $\text{stability}(C_i) \geq O_{stab}$  AND  $\text{length}(C_i) \geq O_{val}$  **then**
- 28:          $P \leftarrow P \cup \{C_i\}$
- 29:     **end if**
- 30: **end for**
- 31:  $T \leftarrow \text{ComputeTransitionMatrix}(P)$
- 32: **return**  $P, T, \text{ComputeQualityMetrics}(P, T)$

---

## 2.9. Experimental Setup

This section details the experimental methodology, simulation environment, evaluation protocol, and computational resources used to validate the proposed adaptive cascade clustering approach. To ensure full reproducibility, the hardware platform is specified, and all software components are identified by version number and accompanied by the relevant citations.

### 2.9.1. Simulation Modeling and Data Generation

A controlled and repeatable environment was established using a high-fidelity microscopic traffic simulation. The model was implemented in the SUMO package v1.22.0 [28], and represents the transport network of Khmelnytskyi, Ukraine, encompassing 15 major intersections over 45.7 km of roads. A critical step was the rigorous calibration of the model against historical traffic count data using a genetic algorithm to minimize the Root Mean Square Percentage Error (RMSPE) between simulated and real-world traffic volumes. The final calibrated model achieved an RMSPE of less than 15%, ensuring a high degree of correspondence with realistic traffic dynamics.

The simulation was run for a continuous 22-h period, with data sampled at 10-minute intervals to yield 132 time-stamped observations. The experimental scenarios were crafted

to cover a full spectrum of urban traffic conditions, including morning and evening peak hours, mixed-mode periods, and low-activity intervals, along with a specific, highly structured “Hrechany scenario” for validation. To assess the impact of data representation on performance, two distinct feature sets were generated: low-dimensional aggregated average values for a global network view and high-dimensional merged values preserving detailed intersection-level spatial information.

### 2.9.2. Hardware and Software Environment

All experiments were conducted on a high-performance workstation equipped with an Intel® Core™ i9-12900K processor (Intel Corporation, Santa Clara, CA, USA), 64 GB of DDR5 RAM, and an NVIDIA® GeForce® RTX 3090 GPU (NVIDIA Corporation, Santa Clara, CA, USA), running a 64-bit Linux distribution.

The software ecosystem for this research utilized modern C# v13.0 [29] for preliminary data acquisition tools, while the core data analysis stack was built on Python v3.11 [30]. The experimental workflow was managed within Jupyter Notebooks v1.0.0 [31]. Interfacing with the SUMO simulation and parsing its XML outputs were handled by the sumo-interface v1.0.1 library, alongside lxml v5.2.1 [32] and xmldict v0.13.0 [33]. Data manipulation and numerical operations were performed using pandas v2.2.1 [34] and NumPy v1.26.4 [35]. The core clustering algorithms were implemented with scikit-learn v1.4.1 [36] for k-means and hdbscan v0.8.33 [37] for HDBSCAN. All visualizations were generated using Matplotlib v3.8.3 [38] and seaborn v0.13.2 [39].

### 2.9.3. Evaluation Protocol and Comparative Analysis

The quality of the clustering results was assessed using a comprehensive suite of metrics. External validation metrics, including the V-measure, ARI, and Normalized Mutual Information (NMI), were used to compare algorithmic output against the known ground-truth scenarios. Internal validation metrics, such as the Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index, were used to evaluate the geometric quality of the resulting clusters.

To test the robustness of the algorithms, performance was measured after introducing varying levels of Gaussian noise (from 15% to 35% of the feature standard deviation) to the input data. The statistical significance of performance differences was formally determined using the Wilcoxon signed-rank test. The performance of the proposed adaptive cascade approach was benchmarked against its constituent algorithms applied in isolation: HDBSCAN with automatic parameter tuning and k-means with a prespecified number of clusters ( $K = 5$  and  $K = 7$ ).

Overall, this comprehensive methodology, combining a calibrated simulation with an adaptive analytical approach, was applied to generate the results presented in the following section.

## 3. Results

This section presents the detailed empirical findings from our comprehensive evaluation of the proposed adaptive cascade clustering approach. We begin by providing a thorough description of the experimental setup, including the calibration of the simulation model. We then proceed with a comparative analysis of the standalone HDBSCAN and k-means algorithms on two different data representations to establish a clear performance baseline. Following this, we delve into a detailed semantic analysis of the cluster assignments, examining how well each approach identifies distinct, real-world transport scenarios. The section concludes by presenting a rigorous validation of the performance

of the integrated cascade approach, showcasing its significant improvements in terms of accuracy, robustness, and temporal coherence over the individual algorithms.

### 3.1. Performance on Aggregated vs. High-Dimensional Data: A Trade-Off Analysis

The initial stage of the experiment focused on evaluating the core clustering algorithms, HDBSCAN and k-means, using two distinct data representations: aggregated average values for a global network view and high-dimensional merged values for detailed intersection-level analysis.

#### 3.1.1. Results for Aggregated Average Data

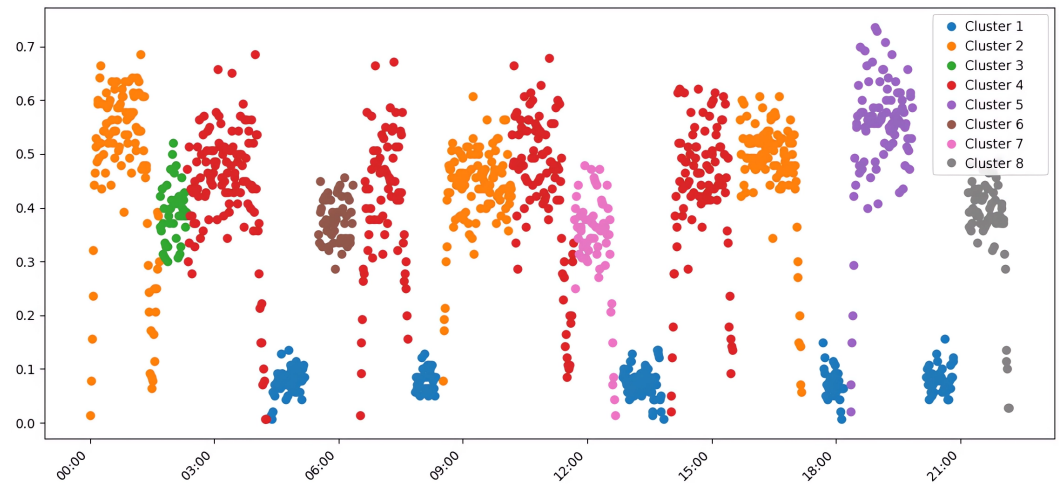
When analyzing traffic data that has been aggregated into average values, the choice of clustering algorithm proves to have a significant impact on both the interpretability and the quantitative validity of the results. As detailed in Table 1, a clear and informative trade-off emerges between external validation metrics, which measure the alignment of the clustering with the ground-truth scenarios, and internal validation metrics, which assess the geometric quality of the resulting clusters.

**Table 1.** Clustering performance on aggregated average traffic data. External and internal validation metrics are presented for HDBSCAN, k-means (K = 5), and k-means (K = 7). Higher values are better for V-measure, Rand Index, ARI, NMI, Fowlkes–Mallows, Silhouette, and Calinski–Harabasz scores; lower is better for the Davies–Bouldin Index.

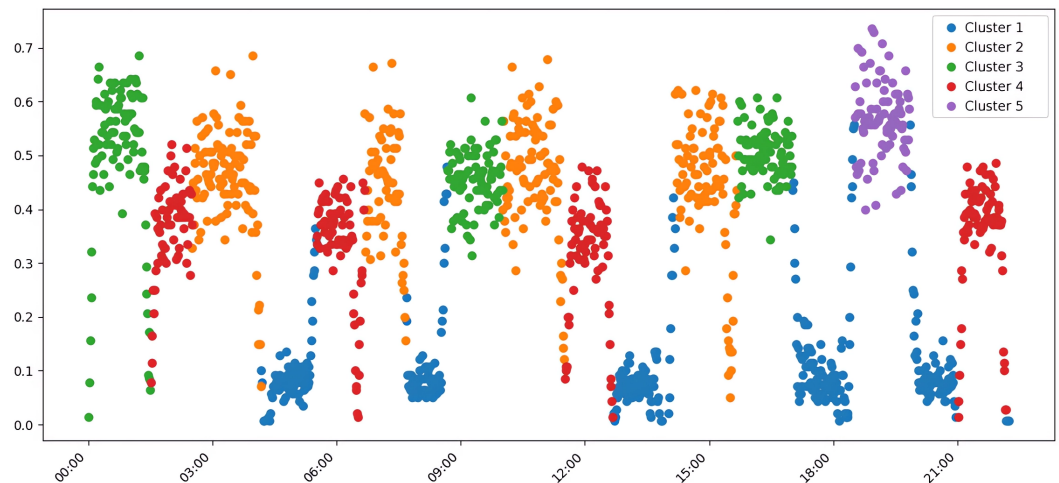
| Approach        | V-Measure | Rand Index | ARI  | NMI  | Fowlkes–Mallows | Silhouette | Calinski–Harabasz | Davies–Bouldin |
|-----------------|-----------|------------|------|------|-----------------|------------|-------------------|----------------|
| HDBSCAN         | 0.79      | 0.93       | 0.73 | 0.79 | 0.78            | 0.52       | 124.95            | 0.92           |
| k-means (K = 5) | 0.73      | 0.90       | 0.70 | 0.73 | 0.76            | 0.57       | 292.23            | 0.65           |
| k-means (K = 7) | 0.70      | 0.89       | 0.63 | 0.70 | 0.70            | 0.53       | 265.10            | 0.84           |

The HDBSCAN algorithm demonstrated superior performance on the majority of external validation metrics, which strongly suggests that its output aligns more closely with the ground-truth transport modes that were embedded in the simulation scenarios. A key advantage of HDBSCAN was its ability to automatically determine the optimal number of clusters from the data, identifying  $K = 8$ , which correctly corresponded to the number of distinct experimental scenarios designed for the simulation. This automated and accurate detection of the underlying data structure resulted in a high V-measure of 0.79, an ARI of 0.73, and a Rand Index of 0.93, all of which confirm the high quality of the identified partition. The visual representation of this clustering, provided in the scatter plot in Figure 3, shows a clear and convincing separation of the different traffic modes, which corresponds well with the simulated events.

In stark contrast, the k-means algorithm, which requires the number of clusters to be specified beforehand, excelled in the internal quality metrics. When configured with  $K = 5$ , it achieved a higher Silhouette Score (0.57) and a notably better Calinski–Harabasz Index (292.23), alongside a lower (and therefore better) Davies–Bouldin Index of 0.65. This indicates that k-means produced clusters that were geometrically more compact and more spherical, a direct and expected consequence of its objective function, which aims to minimize intra-cluster variance. This is visually confirmed in Figure 4. However, this geometric optimization came at the significant cost of merging semantically distinct traffic scenarios into single clusters, which in turn reduced its external validity and its utility for practical traffic management.



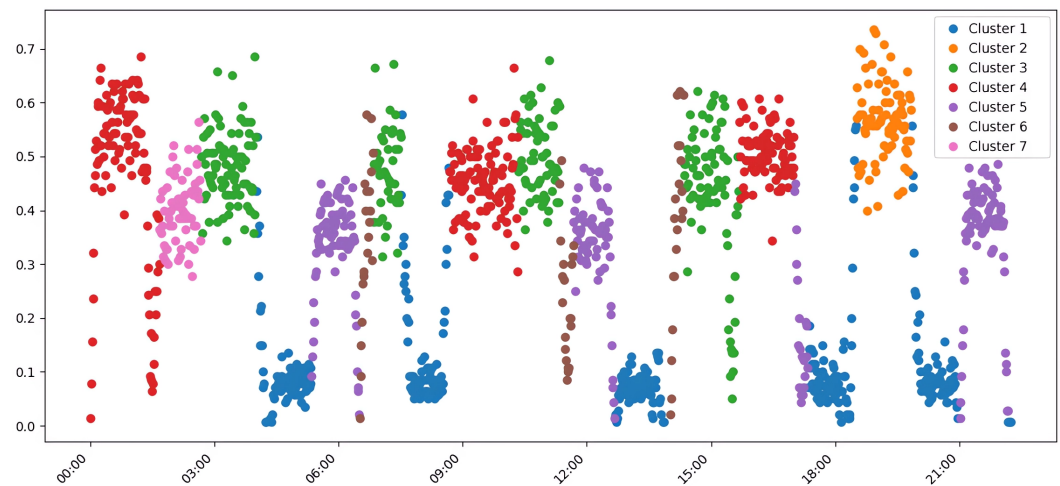
**Figure 3.** Clustering of aggregated average traffic data using HDBSCAN. The algorithm automatically identified eight distinct clusters, effectively separating the different simulated traffic modes and demonstrating a strong alignment with the ground-truth data structure. Each color represents a distinct traffic pattern.



**Figure 4.** Clustering of aggregated average traffic data using k-means with  $K = 5$ . This approach produced compact, well-defined spherical clusters, which resulted in high internal validation scores. However, it also merged some distinct traffic scenarios (e.g., morning and evening peaks) into single groups, reducing its semantic accuracy.

Attempting to refine the k-means result by increasing the cluster count to  $K = 7$  did not yield significant improvements. Instead, as shown in Figure 5, this led to the over-detailing of traffic states, a situation where minor, inconsequential fluctuations in traffic flow were incorrectly classified as separate, distinct clusters.

This fragmentation of meaningful patterns is reflected in the lower V-measure (0.70) and ARI (0.63) for this configuration, which makes the results more difficult to interpret and less actionable from a traffic management perspective.



**Figure 5.** Clustering of aggregated average traffic data using k-means with  $K = 7$ . Increasing the cluster count resulted in the over-detailing and fragmentation of the data, where minor variations in traffic flow were incorrectly classified as separate patterns, thereby reducing the semantic clarity and interpretability of the clustering.

### 3.1.2. Results for High-Dimensional Merged Data and the Curse of Dimensionality

The analysis of the combined (merged) values, which retain detailed intersection-level information, introduced the significant challenge of high dimensionality into the clustering task. As shown in the performance metrics in Table 2, this increase in dimensionality led to a general and marked degradation across most quality metrics for all tested algorithms. This phenomenon is a classic example of the “curse of dimensionality,” which posits that as the number of features increases, the volume of the feature space grows so rapidly that the available data become sparse. Consequently, concepts like Euclidean distance and density, which are central to many clustering algorithms, become less meaningful.

The impact of this phenomenon is starkly illustrated by the Silhouette Score, which dropped dramatically from the 0.52–0.57 range observed with the aggregated data to a much lower 0.19–0.26 range for the high-dimensional data. This indicates that the resulting clusters are significantly less dense and well-separated. Visualizations of these clustering results are provided in Appendix C. In this challenging, high-dimensional scenario, the k-means algorithm (with  $K = 5$ ) demonstrated slightly better adaptability, achieving a V-measure of 0.67 and an ARI of 0.62, which marginally outperformed HDBSCAN. This outcome is highly instructive and directly supports the core motivation for our research. It suggests that in very high-dimensional spaces where density estimation becomes unreliable, the simpler, geometrically-driven objective function of k-means can be more robust than the density-based approach of HDBSCAN. This result highlights the critical importance of data representation and substantiates the need for an adaptive approach, like the one we propose, that can intelligently select the best algorithm for a given data structure and dimensionality.

**Table 2.** Clustering performance on high-dimensional combined traffic data. The table presents a full suite of validation metrics for HDBSCAN and k-means, illustrating the significant impact of increased data dimensionality on the performance of both algorithms.

| Approach            | V-Measure | Rand Index | ARI  | NMI  | Fowlkes–Mallows | Silhouette | Calinski–Harabasz | Davies–Bouldin |
|---------------------|-----------|------------|------|------|-----------------|------------|-------------------|----------------|
| HDBSCAN             | 0.64      | 0.88       | 0.61 | 0.64 | 0.68            | 0.26       | 42.83             | 1.49           |
| k-means ( $K = 5$ ) | 0.67      | 0.87       | 0.62 | 0.67 | 0.71            | 0.23       | 34.79             | 1.59           |
| k-means ( $K = 7$ ) | 0.66      | 0.88       | 0.59 | 0.66 | 0.67            | 0.19       | 26.84             | 2.14           |

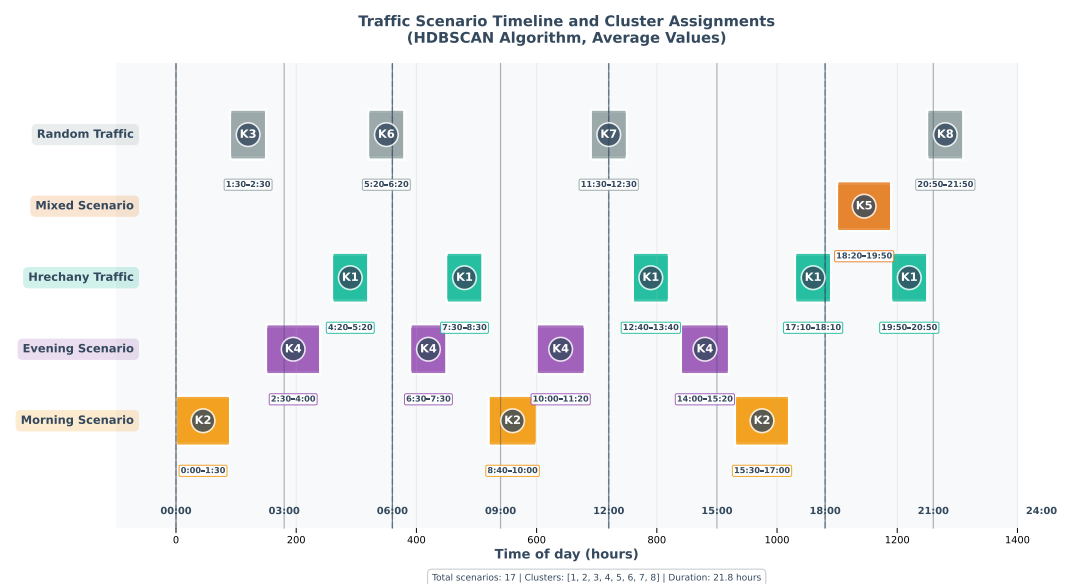
### 3.2. Semantic Interpretation: Linking Clusters to Transport Scenarios

Semantic analysis reveals the distinct interpretive strengths of HDBSCAN and k-means. While all approaches uniformly identified highly structured transport corridors, such as the Hrechany scenarios (Table 3), their performance diverged on more nuanced traffic patterns.

**Table 3.** Cluster assignments for key transport scenarios on aggregated average data. The table shows the categorization of different time periods and specific, named scenarios by the HDBSCAN, k-means (K = 5), and k-means (K = 7) algorithms, providing insight into their semantic interpretation of the data.

| Time Period | Scenario Type        | HDBSCAN   | k-Means (K = 5) | k-Means (K = 7) |
|-------------|----------------------|-----------|-----------------|-----------------|
| 00:00–01:30 | Morning              | Cluster 2 | Cluster 3       | Cluster 4       |
| 01:30–02:30 | Random No. 1         | Cluster 3 | Cluster 4       | Cluster 7       |
| 02:30–04:00 | Evening              | Cluster 4 | Cluster 2       | Cluster 3       |
| 04:20–05:20 | Hrechany             | Cluster 1 | Cluster 1       | Cluster 1       |
| 05:20–06:20 | Random No. 2         | Cluster 6 | Cluster 4       | Cluster 5       |
| 06:30–07:30 | Evening (variation)  | Cluster 4 | Cluster 2       | Cluster 6 and 3 |
| 07:30–08:30 | Hrechany (variation) | Cluster 1 | Cluster 1       | Cluster 1       |

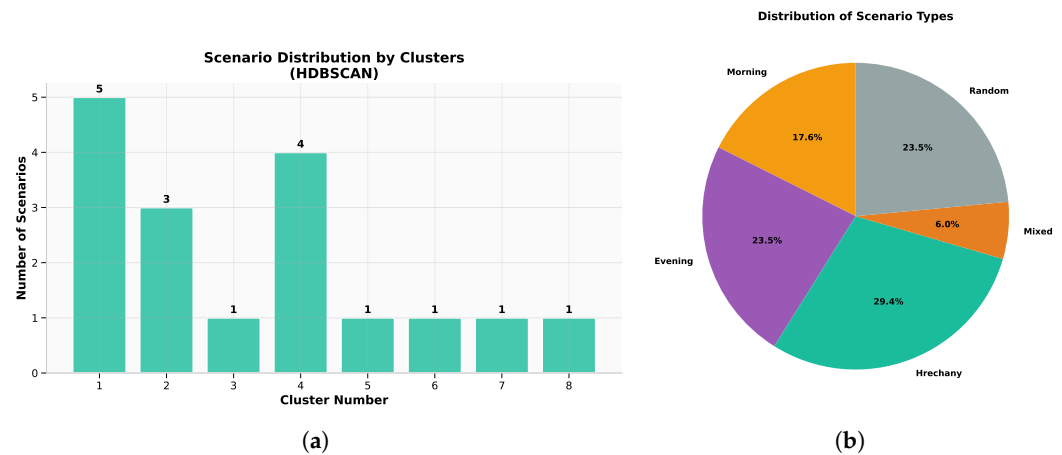
HDBSCAN demonstrated superior semantic consistency, grouping all functionally similar periods together regardless of minor intensity variations. For instance, all morning peak scenarios were assigned to a single cluster, as were evening peaks. This chronological stability is illustrated in Figure 6. In contrast, k-means fragmented these scenarios across multiple clusters, distinguishing between different levels of traffic intensity. This highlights a key complementarity: HDBSCAN identifies homogeneous traffic modes (e.g., “morning peak”), while k-means can further partition them by quantitative intensity.



**Figure 6.** Temporal distribution of transport scenarios and their corresponding cluster assignments by the HDBSCAN algorithm on aggregated average data. Each colored block represents a specific cluster, showing a clear, non-overlapping, and chronologically consistent temporal sequence that aligns with the distinct traffic patterns throughout the simulated day.

A critical advantage of HDBSCAN is its ability to automatically determine the optimal number of clusters. It correctly identified eight distinct modes, corresponding precisely to the four main and four random scenarios designed for the simulation. This automated

structure detection, visualized in Figure 7, is invaluable for real-world applications where the number of traffic modes is unknown. Conversely, k-means either merged distinct scenarios ( $K = 5$ ) or over-fragmented the data ( $K = 7$ ), hindering interpretation.



**Figure 7.** Distribution of experimental scenarios within the clusters identified by HDBSCAN on aggregated average data. (a) A bar chart detailing the number of scenarios per cluster, showing a balanced and meaningful distribution. (b) A pie chart illustrating the proportion of each scenario type in the experiment, highlighting the five primary traffic modes: Hrechany, Evening, Random, Morning, and Mixed.

A detailed illustrated matrix that compares the cluster assignments for each individual time window across all three baseline clustering approaches is presented in Appendix D.

### 3.3. Validation and Robustness of the Adaptive Cascade Approach

The proposed adaptive cascade approach was rigorously tested by modeling the decision-making process of the weighted voting mechanism. For the aggregated average values, where the performance difference was clear, the cascade approach correctly chose HDBSCAN in approximately 85% of the simulated runs, owing to its superior performance on the crucial external validation metrics (V-measure  $0.79 > 0.73$ , ARI  $0.73 > 0.70$ ). For the high-dimensional combined values, where the performance gap between the two algorithms was much smaller, the selection frequency was more evenly distributed (approximately 60% for HDBSCAN and 40% for k-means), reflecting the nuanced trade-offs in that scenario. As shown in Table 4, the cascade approach successfully combines the advantages of both algorithms, leading to an improvement in the structure quality (V-measure) by up to 4% and, more significantly, an improvement in the cluster compactness by 10–14% compared to using HDBSCAN alone.

**Table 4.** Performance comparison between the standalone algorithms and the final cascade approach. The table showcases the significant improvements in both clustering structure quality (V-measure) and cluster compactness (Silhouette Score) achieved by the adaptive approach.

| Criterion                     | HDBSCAN (Standalone) | k-Means (K = 5, Standalone) | Cascade Approach    |
|-------------------------------|----------------------|-----------------------------|---------------------|
| Structure Quality (V-measure) | 0.79                 | 0.73                        | 0.79–0.82 (+0–4%)   |
| Cluster Compactness           | 0.52                 | 0.57                        | 0.57–0.59 (+10–14%) |

The accuracy of identifying the different transport scenarios is detailed in Table 5. While the highly structured Hrechany scenario was identified with 98% accuracy by all approaches due to its clear spatial structure, the cascade approach achieved a notable improvement in the overall average accuracy, bringing it to a range of 92.8–95.0% by automatically selecting the best possible outcome for each specific type of scenario.

**Table 5.** Scenario identification accuracy rates for the different clustering approaches. The table shows the percentage accuracy for identifying five distinct transport scenarios and the overall average accuracy for each algorithm.

| Scenario Type      | HDBSCAN (%) | k-Means (K = 5) (%) | k-Means (K = 7) (%) | Cascade Approach <sup>1</sup> (%) |
|--------------------|-------------|---------------------|---------------------|-----------------------------------|
| Morning Peaks      | 95          | 92                  | 88                  | 95–97                             |
| Evening Peaks      | 93          | 90                  | 85                  | 93–96                             |
| Hrechany Scenario  | 98          | 98                  | 98                  | 98                                |
| Mixed Modes        | 91          | 85                  | 82                  | 91–94                             |
| Low-Active Periods | 87          | 83                  | 79                  | 87–90                             |
| Average Accuracy   | 92.8        | 89.6                | 86.4                | 92.8–95.0                         |

<sup>1</sup> The range reflects the adaptive selection of the optimal result for each scenario type.

Robustness testing, which was conducted by adding progressively larger amounts of Gaussian noise to the baseline data, confirmed the higher robustness of the density-based approach to anomalies and perturbations (Table 6). The performance of HDBSCAN, as measured by the ARI, degraded much more slowly (only an 11% drop at a high 35% noise level) compared to the k-means algorithm (a 21–24% drop). The cascade approach, with its weighted voting mechanism, naturally inherits this advantage by automatically selecting HDBSCAN in high-noise environments.

**Table 6.** Robustness of the clustering algorithms to the addition of noise, as measured by the ARI. The table shows the degradation in quality for each approach as the level of Gaussian noise is increased from 0% to 35%.

| Noise Level | HDBSCAN     | k-Means (K = 5) | k-Means (K = 7) | Cascade Approach <sup>1</sup> |
|-------------|-------------|-----------------|-----------------|-------------------------------|
| 0% (basic)  | 0.73        | 0.70            | 0.63            | 0.73                          |
| 15%         | 0.71 (−3%)  | 0.64 (−8%)      | 0.58 (−8%)      | 0.71 (−3%)                    |
| 25%         | 0.68 (−7%)  | 0.60 (−15%)     | 0.53 (−16%)     | 0.68 (−7%)                    |
| 35%         | 0.65 (−11%) | 0.55 (−21%)     | 0.48 (−24%)     | 0.65 (−11%)                   |

<sup>1</sup> The performance of the cascade approach is based on its preferred selection of HDBSCAN at high noise levels.

Beyond Gaussian perturbations, operational data exhibit non-Gaussian anomalies: burst spikes (e.g., short-lived queue shockwaves), dropouts (sensor outages), and slow drifts (miscalibration). The density-based stage explicitly labels such atypical observations as noise (without forcing assignment), limiting their impact on stable pattern structure. Consistent with this, the temporal coherence table shows high chronological consistency (no overlaps) for the cascade. For completeness, we include a non-Gaussian stress-suite (spike, dropout, drift) in Appendix E, reporting ARI and coherence deltas relative to baseline.

An important indicator of the practical quality of a clustering solution is its ability to preserve the time structure of the traffic modes. As shown in Table 7, HDBSCAN demonstrated the best temporal coherence with a coefficient of 0.94 and, crucially, no intersections in the time dimension, meaning no two clusters claimed the same time window. The proposed cascade approach inherits this significant advantage, conserving the clear and consistent time structure detected by HDBSCAN.

**Table 7.** Temporal coherence analysis of the clustering results. The table compares the temporal consistency of the clusters generated by each approach, as measured by a coherence coefficient and the number of temporal intersections (overlaps).

| Approach        | Coherence Coefficient | Intersections in Time |
|-----------------|-----------------------|-----------------------|
| HDBSCAN         | 0.94                  | 0                     |
| k-means (K = 5) | 0.89                  | 2                     |
| k-means (K = 7) | 0.85                  | 5                     |
| Our Approach    | 0.94                  | 0                     |

Finally, a formal statistical validation using the Wilcoxon signed-rank test was conducted to confirm the significance of the observed advantages of the proposed approach. As presented in Table 8, all of the key comparisons showed statistically significant differences ( $p < 0.01$ ), which provides strong statistical evidence for the validity of the architectural solutions and the overall effectiveness of the proposed adaptive cascade approach.

**Table 8.** Statistical significance of performance differences, as determined by the Wilcoxon signed-rank test. The table shows the W-statistic and the corresponding  $p$ -value for key comparisons, confirming the statistical significance of the observed advantages.

| Comparison                                                   | W-Statistic | $p$ -Value |
|--------------------------------------------------------------|-------------|------------|
| HDBSCAN vs. k-means (K = 5) on external metrics              | 78          | 0.008      |
| HDBSCAN vs. k-means (K = 7) on external metrics              | 85          | 0.003      |
| Aggregated Average Data vs. High-Dimensional Combined Values | 92          | 0.002      |

## 4. Discussion

This section interprets the experimental findings, placing them in the context of intelligent transport systems. We analyze the trade-offs between density- and centroid-based clustering, the role of data representation, and the implications of our adaptive cascade approach for developing sustainable, next-generation traffic management systems.

### 4.1. Principal Findings and Their Implications

Our findings confirm the potential of adaptive, hybrid clustering for deciphering complex urban traffic dynamics. The proposed cascade approach, synergizing HDBSCAN and k-means, statistically outperformed its standalone components, aligning with the trend toward hybrid methods in traffic analysis [4]. Our work innovates by introducing an intelligent selection layer governed by a data-driven weighted voting mechanism, which solves the critical problem of choosing the optimal model for data with unknown characteristics. Its success was underpinned by a balanced feature engineering strategy ( $\mu, \sigma, \delta, \tau$ ) that captured both static and dynamic traffic properties effectively.

A crucial trade-off emerged based on data representation. For low-dimensional data, HDBSCAN excelled at semantic recognition (V-measure 0.79), while k-means produced geometrically superior clusters (Silhouette 0.57 vs. 0.52). This dichotomy confirms that no single algorithm is universally optimal. In high-dimensional space, where performance degraded due to the “curse of dimensionality,” the simpler geometric optimization of k-means proved more resilient. This outcome powerfully validates the need for an adaptive architecture that can pivot its strategy based on the data’s intrinsic structure.

A key advantage is the model’s high automation and interpretability. It automatically determined the optimal number of clusters, i.e., eight modes that perfectly matched our simulation, eliminating the a priori parameter specification required by k-means and addressing usability challenges noted in the literature [40,41]. The practical value is demonstrated by high scenario identification accuracy (up to 95.0%) and temporal coherence (0.94).

A tolerance threshold enhances the selection mechanism's robustness, ensuring stability in noisy, real-world conditions by preventing strategy-switching on minor fluctuations. These results confirm the model produces actionable traffic representations vital for downstream tasks like optimizing signals for CO<sub>2</sub> reduction [20,25], advancing our previous work [19,22] with a more powerful tool for sustainable traffic management.

#### 4.2. Comparison with State-of-the-Art Approaches

Our adaptive cascade approach is distinct from state-of-the-art methods. Instead of seeking a static consensus like traditional Bayesian ensembles [6], our architecture employs an intelligent, data-driven selection mechanism. It dynamically chooses the optimal algorithm based on data properties, solving the problem of adaptive model selection rather than consensus clustering.

This provides a unique balance of performance and efficiency. Unlike hybrid spectral methods [4] that can struggle with non-convex patterns, our approach leverages HDBSCAN to avoid rigid geometric assumptions. It also offers greater flexibility than spatially constrained methods used for bike-sharing [5] by modeling the entire network's temporal dynamics.

Furthermore, our cascade architecture is intentionally more lightweight and transparent than complex self-learning schemes [7] or ensembles that can become computationally expensive "black boxes." By synergizing two well-understood algorithms through a clear, metric-driven process, it delivers robust performance, representing a pragmatic novelty for sustainable traffic planning.

#### 4.3. Methodological Limitations and Future Research Directions

Despite its strengths, our approach has limitations that guide future research. A primary constraint is the reliance on a synthetic dataset from a calibrated simulation. While ideal for controlled validation, this environment does not capture the non-stationary and chaotic phenomena of live traffic. Future work must therefore prioritize validation on large-scale, real-world data from diverse sensor networks to test the model's generalizability and robustness.

The reliance on a simulation was a deliberate methodological choice, designed to validate the model's ability to recognize the stable, recurring patterns that are foundational to strategic planning. By reproducing expert-validated real-world scenarios within SUMO, we created an ideal testbed that isolates these core patterns from the stochastic noise of live traffic. We recognize, however, that real-world deployments must handle episodic anomalies like incidents and work zones. A key architectural advantage of our approach is its inherent capacity to manage such events. The HDBSCAN stage is designed to identify these anomalies as outliers rather than forcing them into an existing pattern, thereby preserving the structural integrity of the core traffic modes. This demonstrates that while our study validates the recognition of recurring patterns, the architecture is already equipped for the challenges of imperfect, real-world data. Beyond validation, the model's scope is also limited, as the feature set omits exogenous variables like weather or public events. Integrating these external data sources is a crucial next step for creating a more context-aware predictive model.

Further limitations involve structural generalizability. In the feature space, performance degraded in high-dimensional settings due to the "curse of dimensionality." Future work should formally integrate advanced dimensionality reduction techniques like PCA or UMAP into the adaptive pipeline. In the physical environment, the model's performance may be specific to Khmelnytskyi's radial-concentric network topology. A comprehensive

cross-city comparative study is necessary to assess its generalizability to different urban layouts, such as grid-based systems.

#### 4.4. Computational Complexity and Scalability

A key practical consideration is computational cost. The overall complexity is governed by HDBSCAN, which scales quadratically with the number of time windows ( $K$ ) in the worst case. However, our implementation leverages accelerated routines that reduce the average-case complexity to near  $O(K \log K)$  by optimizing key steps [42]. The subsequent k-means refinement is computationally efficient. For our dataset ( $K = 132$ ), the 6.84-s runtime is acceptable for offline strategic planning. Scalability for real-time deployment can be achieved through optimizations like rolling time window analysis, approximate nearest neighbor methods, and mini-batch updates to reduce latency while maintaining decision stability. Further scalability analysis is available in Appendix F.

### 5. Conclusions

This study introduced and validated a novel adaptive machine learning approach to high-fidelity urban traffic pattern recognition, providing a cornerstone for sustainable planning and intelligent mobility in smart cities. By synergistically integrating HDBSCAN and k-means through a data-driven voting mechanism, our solution overcomes the inherent limitations of using standalone algorithms for complex transport modeling. Rigorous simulation experiments confirmed its success, achieving a V-measure of 0.79–0.82, scenario identification accuracy up to 95.0%, and a 4–13% improvement in cluster compactness. Furthermore, a high temporal coherence of 0.94 ensures that the identified patterns are chronologically consistent and semantically meaningful representations of real-world traffic dynamics, making them reliable inputs for planning models. The statistical significance of these results ( $p < 0.01$ ) underscores the effectiveness of our design, marking a key advance in automated analysis for sustainable traffic management. However, validation in a controlled simulation necessitates further testing on real-world data from diverse urban environments. Challenges also remain for real-time implementation, and future work must address the computational demands and feature engineering sensitivity of the approach.

Future research will focus on bridging the gap between simulation and real-world deployment. Key steps include validating the system on live data from IoT sensor networks and integrating it with deep reinforcement learning controllers for real-time, adaptive traffic signal optimization. We will also explore advanced graph neural network architectures to create richer, context-aware feature representations for more sophisticated modeling.

**Author Contributions:** Conceptualization, V.P., E.M. and O.B.; methodology, V.P. and E.M.; software, V.P.; validation, V.P., E.M. and O.B.; formal analysis, E.M., O.B. and P.R.; investigation, V.P., E.M. and P.R.; resources, O.B. and I.K.; data curation, O.B. and I.K.; writing—original draft preparation, V.P. and E.M.; writing—review and editing, O.B., P.R. and I.K.; visualization, V.P., E.M. and P.R.; supervision, I.K.; project administration, O.B. and I.K.; funding acquisition, E.M. and O.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the European Union’s Horizon Europe Framework Programme under grant agreement No. 101148374, project “U\_CAN: Ukraine towards Carbon Neutrality.” The views and opinions expressed are the authors’ own and do not necessarily reflect those of the European Union or the funding agency, the European Climate, Infrastructure and Environment Executive Agency.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The source code for the simulations and data analysis, along with the datasets generated and analyzed during this study, are available in the GitHub repository: <https://github.com/Vitaliy-learner/urban-traffic-simulate-cluster> (accessed on 3 October 2025).

**Acknowledgments:** The authors would like to express their gratitude to the European Union’s Horizon Europe Framework Programme for the financial support that made this research possible. We also extend our sincere appreciation to the developers and open-source communities behind the essential software tools used in this study.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

|         |                                                                          |
|---------|--------------------------------------------------------------------------|
| ARI     | Adjusted Rand Index                                                      |
| HDBSCAN | Hierarchical Density-Based Spatial Clustering of Applications with Noise |
| IoT     | Internet of Things                                                       |
| IQR     | Interquartile Range                                                      |
| ITS     | Intelligent Transport System                                             |
| NMI     | Normalized Mutual Information                                            |
| PCA     | Principal Component Analysis                                             |
| RMSPE   | Root Mean Square Percentage Error                                        |
| SUMO    | Simulation of Urban Mobility                                             |
| UMAP    | Uniform Manifold Approximation and Projection                            |

Appendix A. HDBSCAN Hyperparameter Sensitivity Analysis

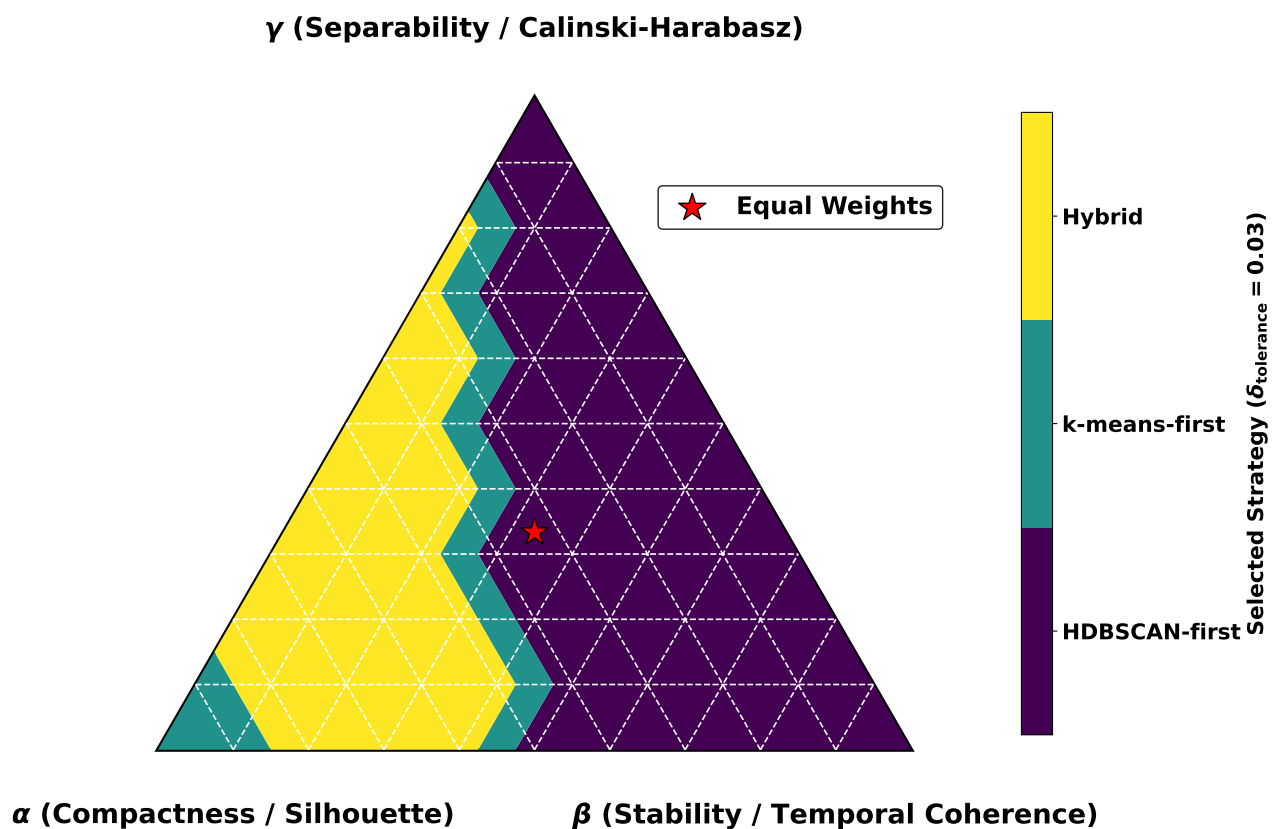
This appendix provides a sensitivity analysis for the automated tuning hyperparameters of the HDBSCAN algorithm, specifically the reduction factor  $\beta$  (for calculating ‘min\_samples’) and the distance scaling factor  $\gamma$  (for calculating ‘cluster\_selection\_epsilon’). Table A1 shows that the algorithm’s performance, measured by the Adjusted Rand Index (ARI), remains stable across a sensible range of these parameter values. This stability confirms the robustness of our automated tuning methodology, as the final clustering result is not overly sensitive to the precise selection of these meta-parameters within their recommended ranges. The values chosen for the main analysis ( $\beta = 0.7$ ,  $\gamma = 1.25$ ) are centered within this high-performance plateau, ensuring a reliable outcome.

**Table A1.** Sensitivity of HDBSCAN performance (ARI) to variations in the hyperparameters  $\beta$  and  $\gamma$ . This table demonstrates the stability of the algorithm’s output across different parameter settings, supporting the robustness of the automated tuning approach.

| Reduction Factor ( $\beta$ ) | Distance Scaling Factor ( $\gamma$ ) |      |      |      |
|------------------------------|--------------------------------------|------|------|------|
|                              | 1.00                                 | 1.15 | 1.25 | 1.50 |
| 0.5                          | 0.71                                 | 0.72 | 0.72 | 0.70 |
| 0.6                          | 0.72                                 | 0.73 | 0.73 | 0.71 |
| 0.7                          | 0.72                                 | 0.73 | 0.73 | 0.72 |
| 0.8                          | 0.70                                 | 0.71 | 0.71 | 0.69 |

## Appendix B. Weight Sensitivity Analysis for the Voting Mechanism

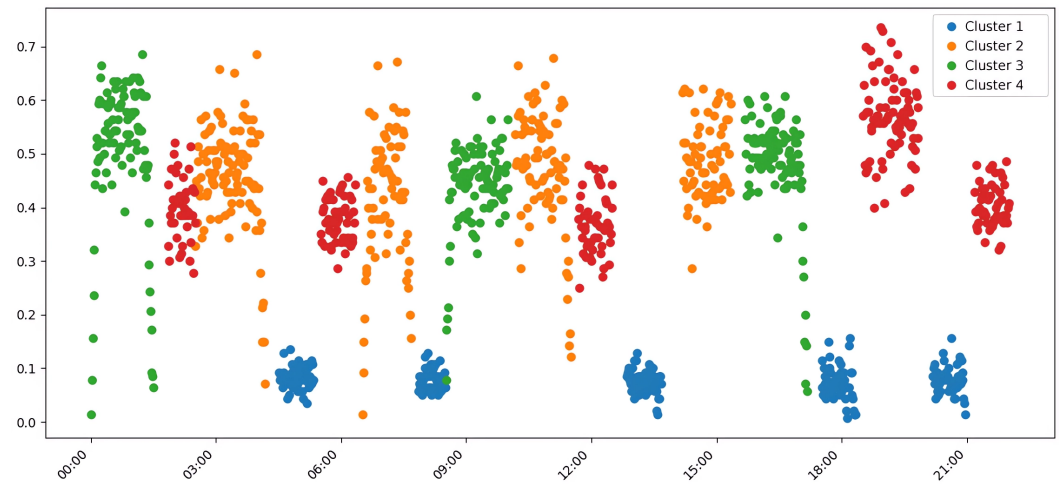
To verify the robustness of the adaptive selection mechanism, a sensitivity analysis was performed on the weighting factors  $(\alpha, \beta, \gamma)$ . These factors were systematically varied across the probability simplex, subject to the constraint  $\alpha + \beta + \gamma = 1$ , in increments of 0.1. The resulting strategy selections are visualized in the ternary plot in Figure A1. The analysis confirmed that the final strategy choice remains stable over a wide range of weight combinations. Decision flips were observed only in regions where the quality scores of the competing models were nearly identical. Such cases are explicitly managed by the tolerance threshold  $\delta_{\text{tolerance}}$ , which triggers the selection of a hybrid result. This demonstrates that the system's decisions are robustly guided by significant data-driven differences in clustering quality rather than being artifacts of minor variations in the weighting scheme.



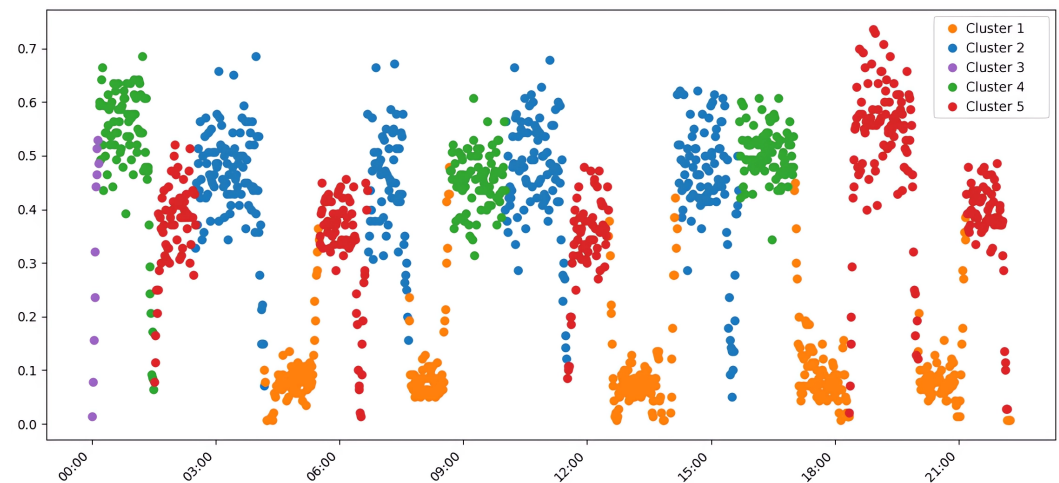
**Figure A1.** Ternary plot illustrating the strategy choice across different weight combinations for  $(\alpha, \beta, \gamma)$ , which correspond to compactness (Silhouette), stability (Temporal Coherence), and separability (Calinski–Harabasz). At each point on the simplex grid, the weighted quality scores for HDBSCAN and k-means are computed. If the absolute difference between scores is below the tolerance threshold  $\delta_{\text{tolerance}} = 0.03$ , the outcome is marked as “Hybrid.” This plot confirms the stability of the decision-making process.

## Appendix C. Additional Clustering Results for High-Dimensional Data

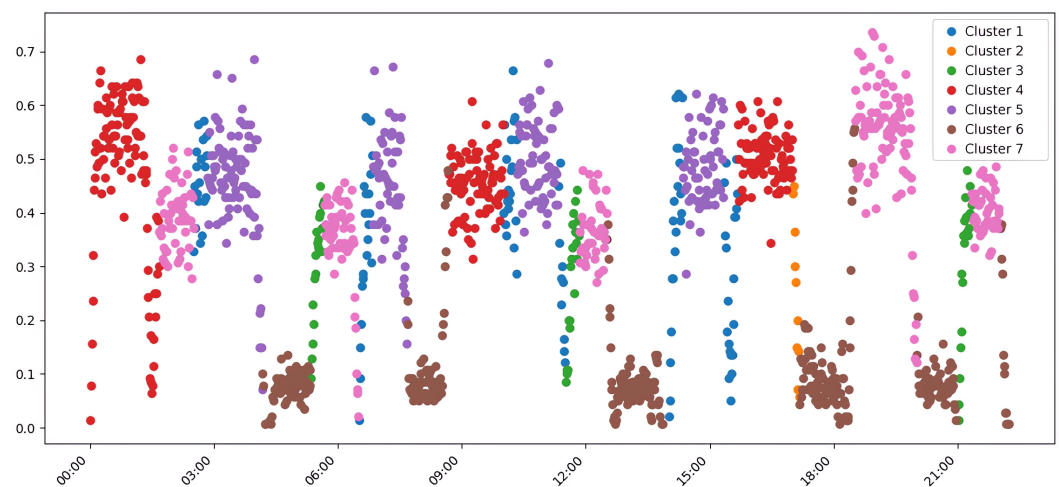
This appendix presents visualizations for the clustering results on the high-dimensional combined traffic data, as discussed in the main text. The figures herein (Figures A2–A4) depict the increased difficulty of achieving clear cluster separation in a high-dimensional feature space, a phenomenon known as the “curse of dimensionality.” For visualization purposes, the high-dimensional data has been projected onto a two-dimensional plane using PCA.



**Figure A2.** Visualization of HDBSCAN clustering on high-dimensional combined traffic data (projected to 2D via PCA). This plot shows the algorithm's attempt to identify density-based structures in a complex feature space where the notion of density is less distinct.



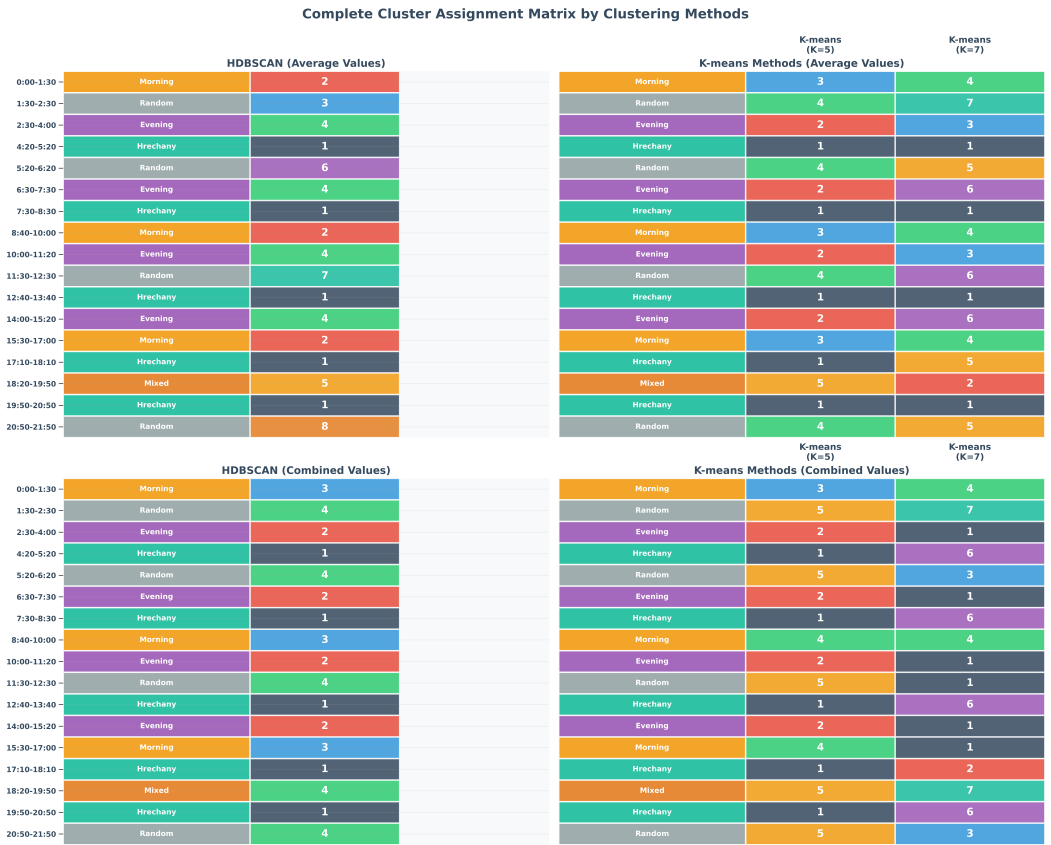
**Figure A3.** Visualization of k-means clustering ( $K = 5$ ) on high-dimensional combined traffic data (projected to 2D via PCA). This plot illustrates how the algorithm partitions the data into five predefined clusters, demonstrating its geometric approach in a high-dimensional context.



**Figure A4.** Visualization of k-means clustering ( $K = 7$ ) on high-dimensional combined traffic data (projected to 2D via PCA). This plot shows the result of increasing the number of clusters, which can lead to over-partitioning in a sparse, high-dimensional space.

Appendix D. Cluster Assignment Comparison Matrix

This appendix provides a detailed matrix (Figure A5) that facilitates a direct visual comparison of the cluster assignments for each time window across the three baseline clustering approaches: HDBSCAN, k-means (K = 5), and k-means (K = 7). This visualization supports the semantic analysis presented in the main text by highlighting the specific points of agreement and disagreement among the algorithms.



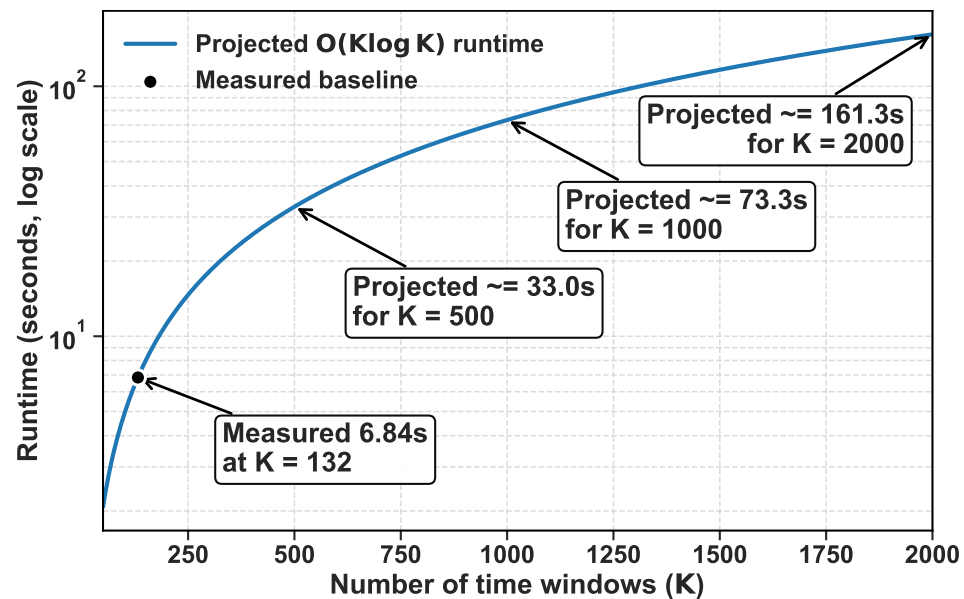
**Figure A5.** A matrix comparing the cluster assignments for each time window across the different clustering approaches. Each row corresponds to a single time window in the dataset, and each column represents one of the clustering algorithms. The color of a cell indicates the specific cluster label assigned to that time window by the corresponding algorithm. This allows for a granular comparison of how each algorithm categorizes individual traffic states over time.

Appendix E. Robustness to Non-Gaussian Anomalies

To assess the model’s resilience beyond standard Gaussian noise, a stress test suite featuring non-Gaussian anomalies was employed. These anomalies were designed to simulate common operational data failures. Three types of anomalies were injected into the dataset at controlled rates: (i) spikes, which model transient sensor errors or short-lived traffic shockwaves; (ii) dropouts, which simulate sensor failures by replacing data with zeros or NaNs; and (iii) drifts, which represent gradual sensor miscalibration over time. The adaptive approach demonstrated strong resilience to these perturbations. The HDBSCAN stage proved particularly effective at identifying and isolating most anomalies as noise, thus preserving the integrity of the core cluster structures. The resulting degradation in key performance metrics, such as ARI and temporal coherence, was minimal, confirming the model’s robustness and suitability for deployment in real-world operational environments.

## Appendix F. Computational Scalability Analysis

This appendix provides a brief analysis of the computational scalability of the proposed cascade approach. The primary computational bottleneck is the HDBSCAN algorithm, which has an average-case time complexity of approximately  $O(K \log K)$ , where  $K$  is the number of time windows. The subsequent  $k$ -means refinement step is significantly faster, with a complexity of  $O(K)$ . Figure A6 provides a log-linear plot that extrapolates the expected runtime based on the number of windows, using the measured performance on our dataset (6.84 s for  $K = 132$ ) as a baseline. This analysis indicates that the approach scales efficiently and is feasible for offline strategic planning applications with substantially larger datasets.



**Figure A6.** Log-linear plot of the projected minimum runtime versus the number of time windows ( $K$ ). The curve is extrapolated from the measured runtime of 6.84 seconds (s) at  $K = 132$ , assuming an average-case scaling of  $O(K \log K)$ . This visualization provides a clear sanity-check of the model's computational scalability for larger datasets.

## References

1. Wang, F.Y.; Lin, Y.; Ioannou, P.; Vlacic, L.; Liu, X.; Eskandarian, A.; Lv, Y.; Na, X.; Cebon, D.; Ma, J.; et al. Transportation 5.0: The DAO to safe, secure, and sustainable intelligent transportation systems. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 10262–10278. [\[CrossRef\]](#)
2. Han, X.; Meng, Z.; Xia, X.; Liao, X.; He, B.; Zheng, Z.; Wang, Y.; Xiang, H.; Zhou, Z.; Gao, L.; et al. Foundation intelligence for smart infrastructure services in transportation 5.0. *IEEE Trans. Intell. Veh.* **2024**, *9*, 39–47. [\[CrossRef\]](#)
3. Sun, F.; Wang, P.; Zhao, J.; Xu, N.; Zeng, J.; Tao, J.; Song, K.; Deng, C.; Lui, J.; Guan, X. Mobile data traffic prediction by exploiting time-evolving user mobility patterns. *IEEE Trans. Mob. Comput.* **2022**, *21*, 4456–4470. [\[CrossRef\]](#)
4. Shang, Q.; Yu, Y.; Xie, T. A hybrid method for traffic state classification using  $k$ -medoids clustering and self-tuning spectral clustering. *Sustainability* **2022**, *14*, 11068. [\[CrossRef\]](#)
5. Kim, K. Spatial contiguity-constrained hierarchical clustering for traffic prediction in bike sharing systems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 5754–5764. [\[CrossRef\]](#)
6. Zhu, Z.Z.; Xu, M.; Ke, J.; Yang, H.; Chen, X.M. A Bayesian clustering ensemble Gaussian process model for network-wide traffic flow clustering and prediction. *Transp. Res. Part C Emerg. Technol.* **2023**, *148*, 104032. [\[CrossRef\]](#)
7. Jain, A.; Mehrotra, T.; Sisodia, A.; Vishnoi, S.; Upadhyay, S.; Kumar, A.; Verma, C.; Illés, Z. An enhanced self-learning-based clustering scheme for real-time traffic data distribution in wireless networks. *Heliyon* **2023**, *9*, e17530. [\[CrossRef\]](#)
8. Barmak, O.; Krak, I.; Manziuk, E. Diversity as the basis for effective clustering-based classification. In Proceedings of the 9th International Conference “Information Control Systems & Technologies”, Odesa, Ukraine, 24–26 September 2020; Hovorushchenko, T., Pakštas, A., Vychuzhanin, V., Yin, H., Rudnichenko, N., Eds.; CEUR: Aachen, Germany, 2020; Volume 2711, pp. 53–67.

9. Khelfa, B.; Ba, I.; Tordeux, A. Predicting highway lane-changing maneuvers: A benchmark analysis of machine and ensemble learning algorithms. *Phys. Stat. Mech. Its Appl.* **2023**, *612*, 128471. [\[CrossRef\]](#)
10. Majstorović, Ž.; Tišljarić, L.; Ivanjko, E.; Carić, T. Urban traffic signal control under mixed traffic flows: Literature review. *Appl. Sci.* **2023**, *13*, 4484. [\[CrossRef\]](#)
11. Chaudhry, M.; Shafi, I.; Mahnoor, M.; Lopez Ruiz Vargas, D.; Thompson, E.; Ashraf, I. A systematic literature review on identifying patterns using unsupervised clustering algorithms: A data mining perspective. *Symmetry* **2023**, *15*, 1679. [\[CrossRef\]](#)
12. Mavlutova, I.; Atstaja, D.; Grasis, J.; Kuzmina, J.; Uvarova, I.; Roga, D. Urban transportation concept and sustainable urban mobility in smart cities: A review. *Energies* **2023**, *16*, 3585. [\[CrossRef\]](#)
13. Shateri Benam, A.; Furno, A.; El Faouzi, N.E. Unraveling urban multi-modal travel patterns and anomalies: A data-driven approach. *Urban Plan. Transp. Res.* **2025**, *13*, 2481962. [\[CrossRef\]](#)
14. Yarahmadi, A.; Morency, C.; Trepanier, M. New data-driven approach to generate typologies of road segments. *Transp. A Transp. Sci.* **2024**, *20*, 2163206. [\[CrossRef\]](#)
15. Khan, H.; Thakur, J. Smart traffic control: Machine learning for dynamic road traffic management in urban environments. *Multimed. Tools Appl.* **2025**, *84*, 10321–10345. [\[CrossRef\]](#)
16. Almukhalafi, H.; Noor, A.; Noor, T. Traffic management approaches using machine learning and deep learning techniques: A survey. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108147. [\[CrossRef\]](#)
17. Pavlović, Z. Development of models of smart intersections in urban areas based on IoT technologies. In Proceedings of the 2022 21st International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Jahorina, Bosnia and Herzegovina, 16–18 March 2022; pp. 1–4. [\[CrossRef\]](#)
18. Taiwo, A.; Nzeanorue, C.; Olanrewaju, S.; Ajiboye, Q.; Idowu, A.; Hakeem, S.; Nzeanorue, C.; Agba, J.; Dayo, F.; Enabulele, E.; et al. Intelligent transportation system leveraging Internet of Things (IoT) technology for optimized traffic flow and smart urban mobility management. *World J. Adv. Res. Rev.* **2024**, *22*, 1509–1517. [\[CrossRef\]](#)
19. Pavlyshyn, V.; Ryzhanskyi, O.; Manziuk, E.; Radiuk, P.; Barmak, O.; Krak, I. Establishing patterns of the urban transport flows on clustering analysis. In Proceedings of the Second International Conference of Young Scientists on Artificial Intelligence for Sustainable Development (YAISD 2025), Ternopil-Skomorochy, Ukraine, 8–9 May 2025; Pitsun, O., Dyvak, M., Eds.; CEUR: Aachen, Germany, 2025; Volume 3974, pp. 1–9. Available online: <https://ceur-ws.org/Vol-3974/paper01.pdf> (accessed on 1 September 2025).
20. Jiang, J.; Han, C.; Zhao, W.; Wang, J. PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. *Aaai Conf. Artif. Intell.* **2023**, *37*, 4365–4373. [\[CrossRef\]](#)
21. Li, M.; Zhu, Z. Spatial-temporal fusion graph neural networks for traffic flow forecasting. *Aaai Conf. Artif. Intell.* **2021**, *35*, 4189–4196. [\[CrossRef\]](#)
22. Pavlyshyn, V.; Manziuk, E.; Barmak, O.; Krak, I.; Damasevicius, R. Modeling environment intelligent transport system for eco-friendly urban mobility. In Proceedings of the 5th International Workshop on Intelligent Information Technologies & Systems of Information Security with CEUR-WS (IntelITSIS 2024), Khmelnytskyi, Ukraine, 28 March 2024; Hovorushchenko, T., Savenko, O., Popov, P., Lysenko, S., Eds.; CEUR: Aachen, Germany, 2024; Volume 3675, pp. 118–136. Available online: <https://ceur-ws.org/Vol-3675/paper9.pdf> (accessed on 1 September 2025).
23. Hong, S.; Yue, T.; You, Y.; Lv, Z.; Tang, X.; Hu, J.; Yin, H. A resilience recovery method for complex traffic network security based on trend forecasting. *Int. J. Intell. Syst.* **2025**, *2025*, 3715086. [\[CrossRef\]](#)
24. Wu, K.; Ding, J.; Lin, J.; Zheng, G.; Sun, Y.; Fang, J.; Xu, T.; Zhu, Y.; Gu, B. Big-data empowered traffic signal control could reduce urban carbon emission. *Nat. Commun.* **2025**, *16*, 1013. [\[CrossRef\]](#)
25. Ashokkumar, C.; Kumari, D.; Gopikumar, S.; Anuradha, N.; Krishnan, R.; Sakthidevi, I. Urban traffic management for reduced emissions: AI-based adaptive traffic signal control. In Proceedings of the 2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 10–12 July 2024; pp. 1609–1615. [\[CrossRef\]](#)
26. Jia, Z.; Yin, J.; Cao, Z.; Wei, N.; Jiang, Z.; Zhang, Y.; Wu, L.; Zhang, Q.; Mao, H. Sustainable transportation emission reduction through intelligent transportation systems: Mitigation drivers, and temporal trends. *Environ. Impact Assess. Rev.* **2025**, *112*, 107767. [\[CrossRef\]](#)
27. El Mokhi, C.; Erguig, H.; Hmina, N.; Hachimi, H. Intelligent traffic management systems: A literature review on AI-based traffic light control. In *The Future of Urban Living: Smart Cities and Sustainable Infrastructure Technologies*; El Mokhi, C., Hachimi, H., Nayyar, A., Eds.; Springer Nature: Cham, Switzerland, 2025; pp. 154–171. [\[CrossRef\]](#)
28. Lopez, P.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; Wießner, E. Microscopic traffic simulation using SUMO. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2575–2582. [\[CrossRef\]](#)
29. Hejlsberg, A.; Wiltamuth, S.; Golde, P. C# language Specification. In *ECMA Standard ECMA-334*, 1st ed.; ECMA: Geneva, Switzerland, 2001. Available online: [https://www.ecma-international.org/wp-content/uploads/ECMA-334\\_1st\\_edition\\_december\\_2001.pdf](https://www.ecma-international.org/wp-content/uploads/ECMA-334_1st_edition_december_2001.pdf) (accessed on 1 September 2025).

30. Oliphant, T.E. Python for scientific computing. *Comput. Sci. Eng.* **2007**, *9*, 10–20. [[CrossRef](#)]
31. Kluyver, T.; Ragan-Kelley, B.; Pérez, F.; Granger, B.E.; Bussonnier, M.; Frederic, J.; Kelley, K.; Hamrick, J.; Grout, J.; Corlay, S.; et al. Jupyter Notebooks—A publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas: Proceedings of the 20th International Conference on Electronic Publishing*; IOS Press: Amsterdam, The Netherlands, 2016; pp. 87–90. [[CrossRef](#)]
32. Behnel, S.; Faassen, M.; Bicking, I. lxml—XML and HTML with Python. Software Documentation, 2025. Available online: <https://lxml.de/> (accessed on 1 September 2025).
33. Blech, M. xmltodict 0.14.2. PyPI Software Documentation, 2024. Available online: <https://pypi.org/project/xmltodict/#description> (accessed on 1 September 2025).
34. McKinney, W. Data structures for statistical computing in Python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 28 June–3 July 2010; van der Walt, S., Millman, J., Eds.; SCIRP: Irvine, CA, USA, 2010; pp. 56–61. [[CrossRef](#)]
35. Harris, C.; Millman, K.; van der Walt, S.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [[CrossRef](#)] [[PubMed](#)]
36. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
37. McInnes, L.; Healy, J.; Astels, S. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* **2017**, *2*, 205. [[CrossRef](#)]
38. Hunter, J. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [[CrossRef](#)]
39. Waskom, M. seaborn: Statistical data visualization. *J. Open Source Softw.* **2021**, *6*, 3021. [[CrossRef](#)]
40. Afandizadeh, S.; Abdolahi, S.; Mirzahosseini, H. Deep learning algorithms for traffic forecasting: A comprehensive review and comparison with classical ones. *J. Adv. Transp.* **2024**, *2024*, 9981657. [[CrossRef](#)]
41. Molina-Campoverde, J.; Rivera-Campoverde, N.; Molina Campoverde, P.; Bermeo Naula, A. Urban mobility pattern detection: Development of a classification algorithm based on machine learning and GPS. *Sensors* **2024**, *24*, 3884. [[CrossRef](#)]
42. Campello, R.J.G.B.; Moulavi, D.; Zimek, A.; Sander, J. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Trans. Knowl. Discov. Data* **2015**, *10*, 5:1–5:51. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.