MDPI

*Article*

# An Adaptive Cascade Clustering Approach to High-Fidelity Traffic Pattern Recognition in Smart Transportation Systems

**Vitaliy Pavlyshyn [1], Eduard Manziuk [1], Oleksander Barmak [1], Pavlo Radiuk [1]* and Iurii Krak [2,3]**

[1] Department of Computer Science, Khmelnytskyi National University, 11 Instytuts'ka Str., 29016 Khmelnytskyi, Ukraine; vitaliy@ualeaders.com (V.P.); manziuk.e@khmnu.edu.ua (E.M.); barmako@khmnu.edu.ua (O.B.)
[2] Department of Theoretical Cybernetics, Taras Shevchenko National University of Kyiv, 4d Akademika Glushkova Ave, 03680 Kyiv, Ukraine; iurii.krak@knu.ua
[3] Laboratory of Communicative Information Technologies, V.M. Glushkov Institute of Cybernetics, 40 Akademika Glushkova Ave, 03187 Kyiv, Ukraine
* Correspondence: radiukp@khmnu.edu.ua; Tel.: +380-97-854-9146

## Highlights

**What are the main findings?**

- An adaptive clustering approach enhances traffic pattern analysis by synergizing density- and centroid-based algorithms, improving cluster compactness by up to 13%.
- The framework automatically and accurately identifies distinct traffic modes with up to 95.0% accuracy, providing a robust tool for analyzing dynamic urban mobility patterns.

**What is the implication of the main finding?**

- The research provides a foundational analytical tool for intelligent traffic management systems, enabling dynamic signal control that adapts to real-time traffic conditions.
- By facilitating optimized traffic flow, the approach directly contributes to building sustainable smart cities by reducing congestion, vehicle emissions, and fuel consumption.

**Abstract:** Static traffic management in modern urban centers is a critical barrier to achieving sustainable mobility, leading to inefficient traffic flow and significant environmental impact. The development of smart cities requires analytical methods capable of autonomously understanding the complex, dynamic patterns of urban traffic. This paper introduces an adaptive cascade clustering approach that synergizes HDBSCAN and k-means algorithms to address this challenge. By employing a data-driven weighted voting mechanism, our framework enhances traffic pattern recognition by integrating robust structural analysis with precise cluster refinement. Validated using a high-fidelity simulation of the Khmelnytskyi, Ukraine, transport network, the approach demonstrated a superior ability to identify true traffic modes, achieving a V-measure of 0.79–0.82 and improving cluster compactness by 4–13% over standalone algorithms. The model 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 provides a foundational technology for intelligent transport systems, enabling more responsive, efficient, and sustainable urban mobility management.

**Keywords:** smart cities; intelligent transport systems (ITS); sustainable mobility; urban traffic management; transport data clustering; HDBSCAN; k-means; fuel consumption and emissions

Received:
Revised:
Accepted:
Published:

**Citation:** Pavlyshyn, V.; Manziuk, E.; Barmak, O.; Radiuk, P.; Krak, I. An Adaptive Cascade Clustering Approach to High-Fidelity Traffic Pattern Recognition in Smart Transportation Systems. *Smart Cities* **2025**, *1*, 0. https://doi.org/



## 1. Introduction

The evolution toward smart cities is intrinsically linked to the challenge of sustainable urban mobility. As urban populations grow, the complexity of traffic flow presents a

major obstacle to environmental sustainability, economic efficiency, and public well-being. Conventional traffic management, reliant on static, pre-programmed signal timings, is fundamentally misaligned with the dynamic nature of a smart city. These legacy systems cannot adapt to real-time fluctuations in traffic demand, leading to chronic congestion, increased fuel consumption, and a significant rise in greenhouse gas emissions. Addressing these inefficiencies is paramount for creating resilient and livable urban environments. This requires a paradigm shift toward intelligent transport systems (ITS) that can perceive, interpret, and dynamically respond to the network's state. Central to such systems is high-fidelity traffic pattern recognition—the ability to automatically identify the network's distinct operational modes.

This paper introduces an adaptive cascade clustering approach as an enabling technology for smart and sustainable urban mobility. By synergistically integrating the complementary strengths of density-based and centroid-based clustering algorithms, our approach provides a robust and automated foundation for the next generation of intelligent, responsive, and eco-conscious urban transport control systems.

### *1.1. State of the Art*

The push toward smarter, more sustainable urban mobility is a core pillar of the smart city vision, which demands a new generation of intelligent transport systems that are safe, resilient, and eco-friendly [1,2]. A foundational element of this vision is the ability to accurately model and forecast traffic dynamics using real-time data analytics. Research highlighting the importance of time-evolving mobility patterns for prediction tasks underscores the need to capture temporal dynamics in any traffic analysis [3]. Our study contributes to this goal by proposing an advanced unsupervised learning framework that uses cascade clustering and weighted voting to deconstruct complex traffic flows into their fundamental patterns, moving beyond the limits of traditional, monolithic analytical methods.

Unsupervised clustering is a cornerstone of traffic analysis as it can uncover latent structures in data without pre-labeled examples. However, existing methods have limitations within a smart city context. For instance, hybrid approaches combining k-medoids with spectral clustering have shown high accuracy but can be sensitive to initialization, and their geometric assumptions may not hold for heterogeneous urban traffic data [4]. Similarly, spatially constrained hierarchical clustering has improved forecasting in bike-sharing systems, but its reliance on fixed spatial constraints makes it less adaptable to the fluid nature of vehicular traffic [5]. More recent advancements, such as Bayesian ensembles [6] and self-learning clustering schemes [7], have enhanced performance but often at the cost of significant model complexity, which can obscure interpretability and require substantial computational resources that limit their use in real-time applications. Inspired by the proven efficacy of ensemble methods in traffic analysis [8–10], our research aims to develop a more adaptive, lightweight, and automated framework for identifying hidden patterns [11,12] in dynamic urban environments [13,14]. Our focus differs from approaches that create static typologies of road infrastructure, as we aim to identify the dynamic, time-varying operational modes of the entire network [15].

A critical application of advanced traffic analysis is mitigating the transport sector's environmental footprint, a key goal of sustainable urban mobility. Technologies integral to the smart city, such as deep learning, the Internet of Things (IoT), and decentralized control, offer powerful tools for real-time monitoring and intelligent traffic management essential for minimizing environmental harm [16,17]. Comprehensive sensor networks and dynamic traffic signal adjustment algorithms are also proving instrumental in this effort [18,19]. Modern analytical methods have successfully established the link between traffic flow

patterns and vehicle emission levels [20–22]. Our prior research laid the groundwork for this study by demonstrating the utility of cluster analysis for traffic pattern identification [20] and developing foundational designs for environmentally oriented transport management systems [23]. This study directly addresses a key limitation of that work—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 identification of transport modes, which is critical for designing intelligent traffic management strategies that reduce vehicular pollution.

### *1.2. Objectives and Tasks*

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

The primary goal of this study is to enhance sustainable urban mobility by developing and validating an adaptive approach for the automated determination 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 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 to diverse datasets.
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 clustering.
4. Rigorously validate the proposed adaptive approach through controlled simulation experiments, comparing its 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 light regulation. This leads to tangible sustainability 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 outcome.

### *1.3. Motivation and Contributions*

This study is motivated by the urgent need for intelligent, eco-friendly, and safe transportation infrastructures in smart cities. Traffic intensification and outdated control systems contribute 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 approach to high-fidelity traffic pattern recognition. Accurate identification of these patterns is directly linked to mitigating environmental impact, as it enables intelligent systems to optimize traffic signals, thereby minimizing inefficient driving modes [25]. Building upon our foundational research [20,23], 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 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 accuracy and stability.
- 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, which ensures adaptability and eliminates the need for manual algorithm selection.
- A sophisticated 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.

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.

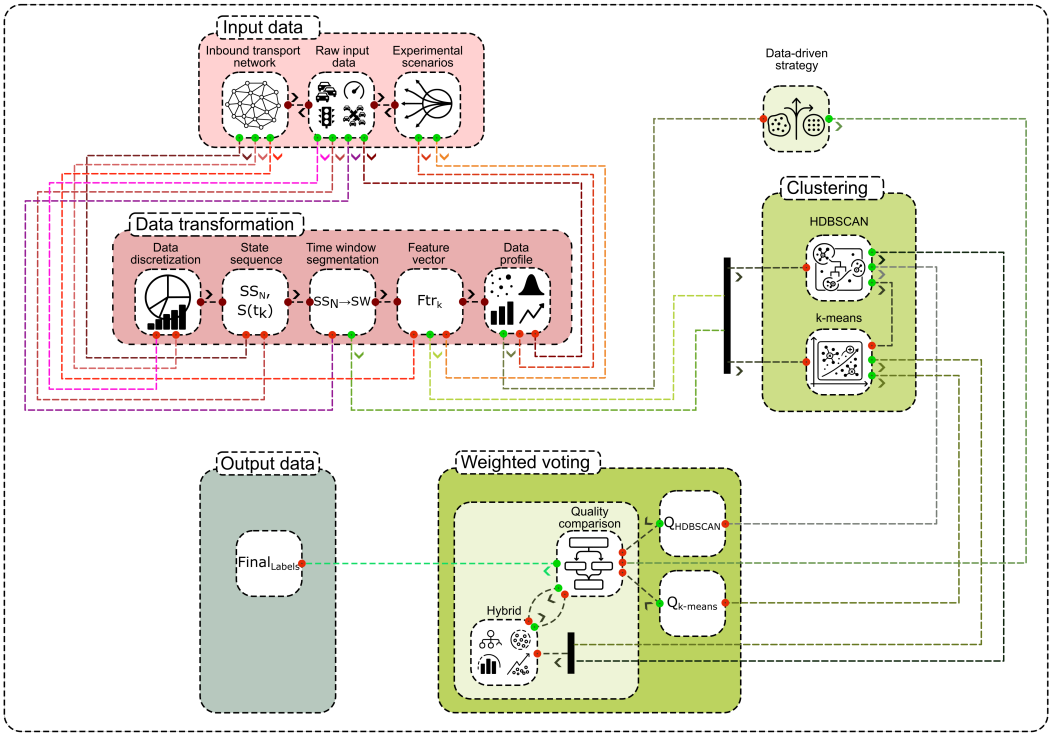



**Figure 1.** The proposed adaptive cascade clustering architecture. The process commences with data acquisition from the urban transport network, which is then subjected to a feature extraction process within discrete time windows. A data-driven weighted voting mechanism then selects the optimal clustering 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 foundation of our analysis is a hybrid methodology that integrates real-world empirical data with a high-fidelity simulation environment. The empirical data was collected by processing visual information from key intersections in Khmelnytskyi, Ukraine, using computer vision techniques. This raw transport data, which underpins our model, is provided in an external repository, as detailed in the Data Availability Statement.

This real-world data was used to construct and meticulously calibrate a digital twin of the city's transport network within the SUMO package v1.22.0 [28]. This approach ensures that the model's outputs accurately reflect authentic traffic dynamics. For this study, the simulation was run for a 22-hour period, with key traffic parameters such as vehicle speeds and queue lengths sampled at 10-minute intervals, yielding a time series of 132 distinct observations.

Crucially, this hybrid approach provides a controlled and repeatable experimental environment. By embedding known ground-truth scenarios into the simulation, we can

conduct a direct and objective quantitative assessment of the clustering algorithms' performance, validating their ability to identify genuine traffic patterns.

### *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:

$$G = (V, E), \tag{1}$$

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:

$$SS_N = \{S(t_0), S(t_1), \ldots, S(t_N)\}, \tag{2}$$

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:

$$SW = \{W_1, W_2, \ldots, W_K\}, \quad K \in \mathbb{N}, \tag{3}$$

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

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

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:

$$Ftr_k = (\mu_k, \sigma_k, \delta_k, \tau_k), \tag{5}$$

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:

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

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

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

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 when detailed, intersection-level data is preserved, leading to a high-dimensional feature space. This "curse of dimensionality" can severely degrade the performance of clustering algorithms, as distances between points become less meaningful. To address this, our approachology incorporates strategies for dimensionality mitigation. Before clustering, feature selection can be performed using metrics like mutual information or Gini importance (derived from a preliminary tree-based model) to identify and retain only the most informative features.

Alternatively, dimensionality reduction techniques can be applied. Principal Component Analysis (PCA) is a linear method that can be used to project the data onto a lower-dimensional subspace while preserving the maximum amount of variance. For capturing more complex, non-linear structures, manifold learning techniques such as Uniform Manifold Approximation and Projection (UMAP) are more suitable. UMAP is particularly effective at preserving both the local and global structure of the data in the low-dimensional embedding. The choice of which strategy to employ, i.e,. feature selection, PCA, or UMAP, can be integrated into the adaptive framework, with the decision based on an initial profiling of the data's linearity and intrinsic dimensionality.

### *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:

$$mcs = \lceil N_{ob} \cdot s_{cl} \rceil, \tag{7}$$

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 $mcs$:

$$ms = \lceil mcs \cdot \beta \rceil, \tag{8}$$

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:

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

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:

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

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:

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

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 $Cnt(C_k)$, 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:

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

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:

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

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

$$Dns(C_k) = \frac{|C_k|}{\text{Vol}_{\text{rad}}(C_k)}, \tag{14}$$

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

$$\text{Vol}_{\text{rad}}(C_k) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} \cdot r(C_k)^d, \tag{15}$$

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:

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

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:

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

#### 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:

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

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:

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

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 innovation 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 HDBSCAN, the metric balances structural correctness, stability, and interpretability:

$$\text{Quality}_{\text{HDBSCAN}} = \alpha \cdot \text{Silhouette} + \beta \cdot \text{Stability} + \gamma \cdot \text{Interpretability}, \tag{20}$$

where $\alpha, \beta, \gamma$ are weighting factors.

For the k-means stage, the quality metric emphasizes the geometric properties of the clusters:

$$\text{Quality}_{\text{k-means}} = \alpha \cdot \text{Silhouette} + \beta \cdot \text{Compactness} + \gamma \cdot \text{Separation}. \tag{21}$$

These metrics provide a balanced assessment of geometric quality, internal consistency, and semantic value. For this study, the weighting factors were set to $1/3$ each for a comprehensive evaluation.

To ensure system stability and prevent frequent switching between strategies due to minor performance fluctuations—a common issue in noisy, real-world data—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:

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

When the quality scores are within this tolerance, indicating comparable performance, a hybrid result is generated. This is achieved through a principled label fusion process at the individual data point level. For any point where the algorithms disagree, the final label is assigned based on a local confidence score, derived from the outlier score for HDBSCAN and the distance to the centroid for k-means. This leverages the local strengths of both methods, often producing a superior final partition. This refined decision-making process, formalized in Equation (22) and visualized in Figure 2, ensures that the final clustering result is the most robust and meaningful choice for the given data.

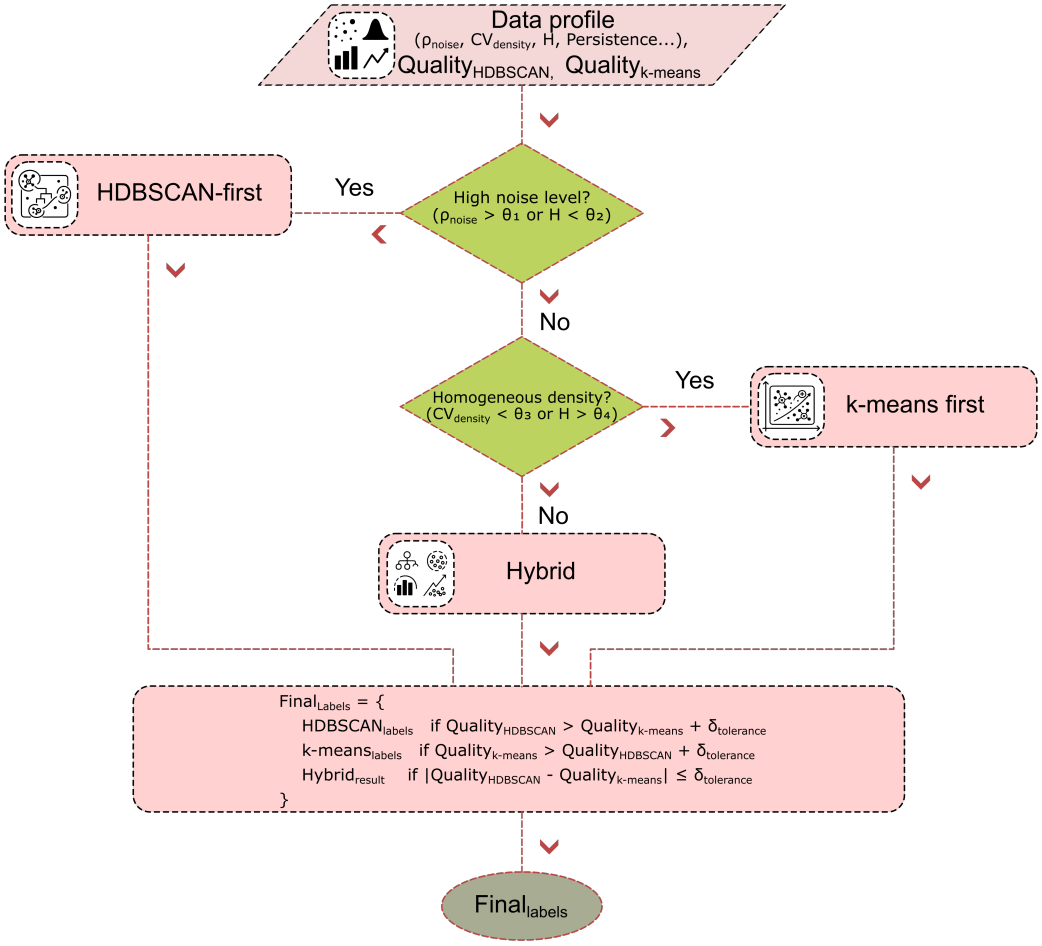



**Figure 2.** Logical scheme of the weighted voting and adaptive selection process. Characteristics of the input data, such as the noise ratio and density variation, are evaluated to inform the initial selection between HDBSCAN and k-means. The quality of both models is then assessed using a combination 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.

#### 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:

$$\rho_{\text{noise}} = \frac{|\text{outliers}|}{|\text{Data}|}. \tag{23}$$

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

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

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, $H$:

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

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:

$$CV_{\text{density}} = \frac{\sigma_{\text{density}}}{\mu_{\text{density}}}, \tag{26}$$

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

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

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:

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

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:

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

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** Adaptive Cascade Clustering for Traffic Pattern Recognition.

**Require:** Traffic data $D$; window parameters ($w_{size}$, $w_{step}$); thresholds ($O_{stab}$, $O_{val}$).
**Ensure:** Traffic patterns $P$; transition matrix $T$; quality metrics $Q$.

Initialize: $W \leftarrow \{\}$, $P \leftarrow \{\}$ ▷ *Phase 1. Data Preparation & Analysis*
**for** $i = 1$ to $N - w_{size}$ step $w_{step}$ **do**
    window $\leftarrow D[i : i + w_{size}]$
    features $\leftarrow$ ComputeFeatures(window)
    $W \leftarrow W \cup \{\text{features}\}$
**end for**
DataProfile $\leftarrow$ ComputeDataProfile($W$) ▷ *Phase 2. Adaptive Clustering*
**if** DataProfile suggests high noise or complex structure **then**
    strategy $\leftarrow$ HDBSCAN_first
**else if** DataProfile suggests low noise and simple structure **then**
    strategy $\leftarrow$ k_means_first
**else**
    strategy $\leftarrow$ hybrid
**end if**
$L_h \leftarrow \text{HDBSCAN}(W, \text{auto\_params})$
$L_k \leftarrow \text{k\_means}(W, |\text{unique}(L_h)|, \text{init} = \text{centroids\_from}(L_h))$
$Q_h \leftarrow \text{EvaluateQuality}(W, L_h)$
$Q_k \leftarrow \text{EvaluateQuality}(W, L_k)$ ▷ *Phase 3. Weighted Voting & Validation*
**if** $|Q_h - Q_k| > \delta_{\text{tolerance}}$ **then**
    $L_{final} \leftarrow \arg\max(Q_h, Q_k)$
**else**
    $L_{final} \leftarrow \text{HybridResult}(L_h, L_k)$
**end if**
**for** each cluster $C_i$ in $L_{final}$ **do**
    **if** stability$(C_i) \geq O_{stab}$ AND length$(C_i) \geq O_{val}$ **then**
        $P \leftarrow P \cup \{C_i\}$
    **end if**
**end for**
$T \leftarrow \text{ComputeTransitionMatrix}(P)$
**return** $P, T, \text{ComputeQualityMetrics}(P, T)$

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.

### *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-hour 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 xmltodict 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).

## 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.

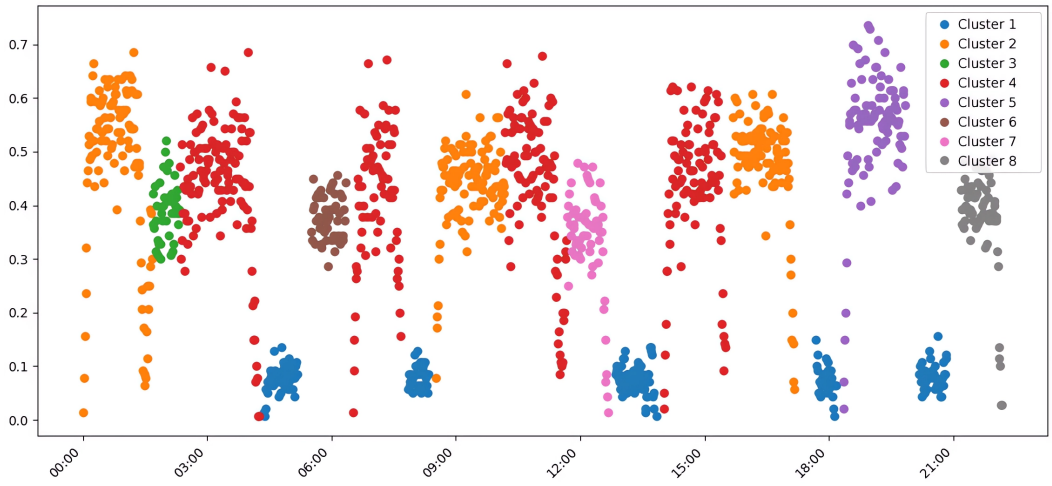



**Figure 3.** Clustering of aggregated average traffic data using HDBSCAN. The algorithm automatically identified eight distinct clusters, effectively separating the different simulated traffic modes and demonstrating a strong alignment with the ground-truth data structure. Each color represents a distinct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

In stark contrast, the k-means algorithm, which requires the number of clusters to be specified "a priori," 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.

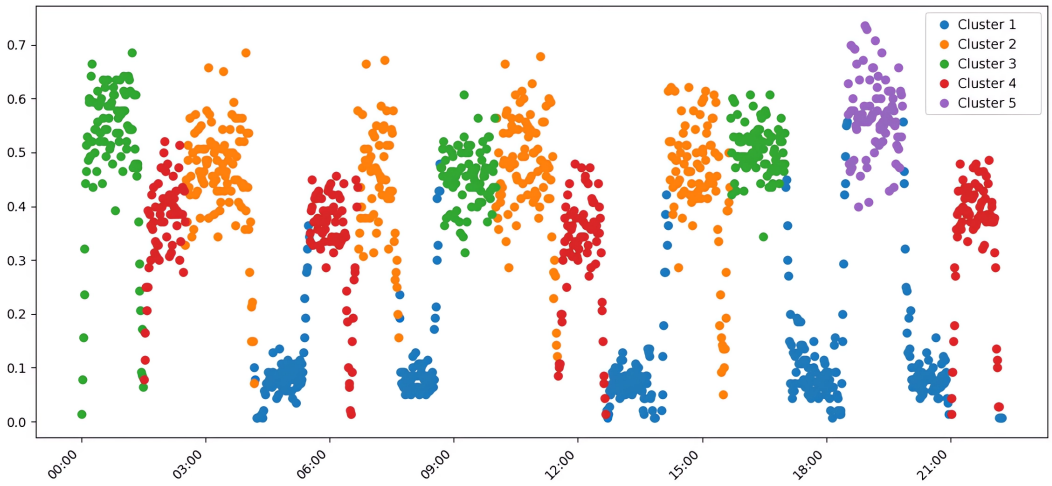



**Figure 4.** Clustering of aggregated average traffic data using k-means with K=5. This approach produced compact, well-defined spherical clusters, which resulted in high internal validation scores. However, it also merged some distinct traffic scenarios (e.g., morning and evening peaks) into single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

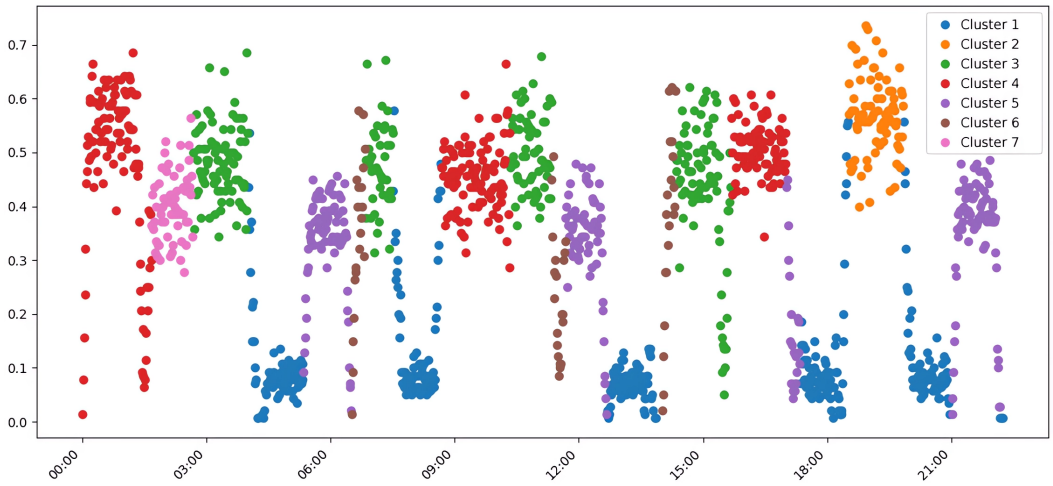



**Figure 5.** Clustering of aggregated average traffic data using k-means with K=7. Increasing the cluster count resulted in the over-detailing and fragmentation of the data, where minor variations in traffic flow were incorrectly classified as separate patterns, thereby reducing the semantic clarity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

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.

#### 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 B. 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*

A detailed semantic analysis of the cluster assignments for the aggregated average data reveals how the architectural differences between HDBSCAN and k-means translate into their interpretive value for real-world traffic management. The Hrechany district scenarios, which represent a well-defined and highly structured transport corridor from a large residential area to the city center and a major clothing market, showed remarkable uniformity of recognition across all tested approaches. As can be seen in Table 3, the time windows corresponding to these periods were consistently and correctly assigned to a single cluster (Cluster 1) by all three approaches. This indicates the presence of a clear and dominant spatiotemporal structure in the data for this scenario, which both the density-based and centroid-based algorithms could easily and accurately capture.

**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 6/3 |
| 07:30–08:30 | Hrechany (variation) | Cluster 1 | Cluster 1 | Cluster 1 |

The fundamental divergence between the algorithms becomes apparent when analyzing more nuanced or complex scenarios. The chronological analysis of the cluster assignments, as illustrated in the timeline plot in Figure 6, showcases the exceptional stability and semantic consistency of HDBSCAN's output. All of the morning periods simulated in the experiment (e.g., 0:00–1:30, 8:40–10:00), which were characterized by traffic flow towards the city center and the clothing market, were consistently grouped into a single cluster (Cluster 2). In contrast, the k-means algorithm distributed these same scenarios across multiple different clusters, which suggests a higher sensitivity to minor, quantitative variations in traffic intensity (e.g., distinguishing between the early, rising peak and the full peak). This highlights a key complementary aspect of the two algorithms: HDBSCAN is adept at identifying semantically homogeneous modes (e.g., the general concept of a "morning peak"), while k-means can be used to further partition these modes based on their quantitative intensity (e.g., "low-intensity peak" versus "high-intensity peak"). Similarly, HDBSCAN grouped most of the evening periods, which were characterized by return traffic, into a single, distinct Cluster 4, further confirming its ability to detect functionally similar traffic modes regardless of their specific time of occurrence.

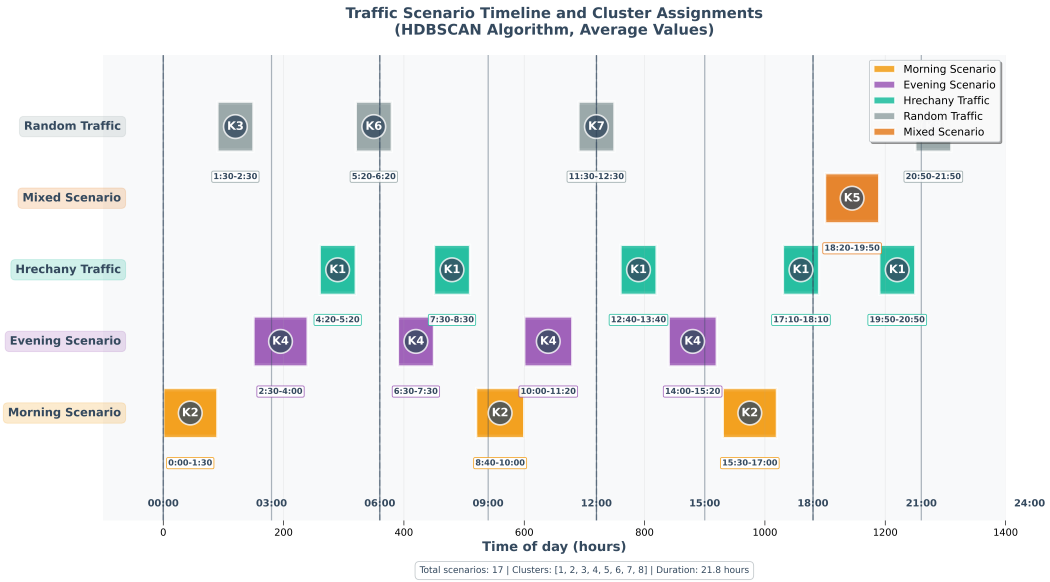
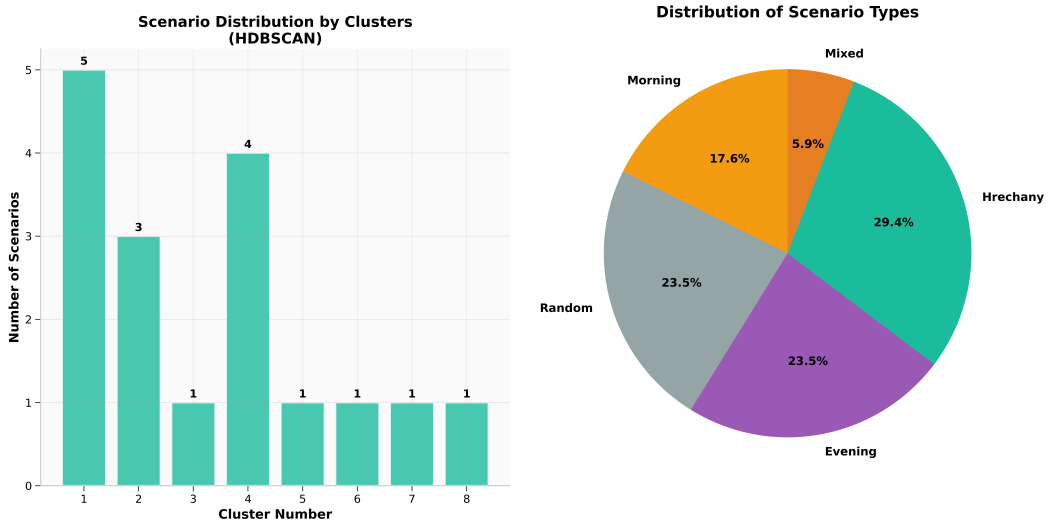



**Figure 6.** Temporal distribution of transport scenarios and their corresponding cluster assignments by the HDBSCAN algorithm on aggregated average data. Each colored block represents a specific cluster, showing a clear, non-overlapping, and chronologically consistent temporal sequence that aligns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

A critical advantage of HDBSCAN in the context of transport systems analysis is its ability to automatically determine the optimal number of clusters without requiring any "a priori" specification. In our experiments, HDBSCAN automatically identified eight distinct clusters. This result logically and accurately corresponds to the real organization of the simulation scenarios: four main types of transport modes (morning, evening, Hrechany, and mixed) plus four distinct variations of random, low-intensity movement with different characteristics of intensity and spatial distribution. This precision in automatic structure detection is particularly valuable for practical, real-world applications where expert knowledge of the number of traffic modes may not be available or accurate. In contrast, the k-means algorithm with five clusters showed a tendency to combine semantically different scenarios, which complicates interpretation, while increasing the cluster count to seven led to over-fragmentation without providing any significant improvement in semantic understanding. The distribution of the different scenarios across HDBSCAN's eight identified clusters is visualized in Figure 7.




(**a**) (**b**)

**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 C.

### *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 4–13% 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 (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 (+4–13%) |

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 [1] (%) |
|---|---|---|---|---|
| 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** |

[1] 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 [1] |
|---|---|---|---|---|
| 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%) |

[1] The performance of the cascade approach is based on its preferred selection of HDBSCAN at high noise levels.

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 results, contextualizing them within the broader field of intelligent transport systems. We analyze the trade-offs between density-based and centroid-based clustering that our findings revealed, highlighting the critical role of data representation in algorithmic performance. The advantages and limitations of the proposed adaptive cascade approach are critically evaluated, and its implications for developing next-generation, environmentally-oriented traffic management systems are explored.

### *4.1. Principal Findings and Their Implications*

The findings of this study confirm the significant potential of adaptive, hybrid clustering to decipher the complex dynamics of urban transport systems. Our proposed cascade approach, which synergistically integrates HDBSCAN and k-means, demonstrated a statistically significant improvement over its standalone components. This outcome aligns with the growing consensus favoring hybrid and ensemble methods in traffic analysis for their ability to combine the strengths of multiple models [4,8]. However, our work advances this paradigm by introducing an intelligent adaptive selection layer. Governed by a data-driven weighted voting mechanism, this layer addresses the pivotal challenge of choosing the optimal analytical model for datasets with unknown characteristics. The success of this pattern recognition was rooted in a comprehensive feature engineering strategy, where the selected metrics ($\mu, \sigma, \delta, \tau$) provided a balanced representation of both static and dynamic

traffic properties, enabling the identification of key traffic modes without overcomplicating the feature space.

Our experiments exposed a crucial trade-off contingent on data representation. On lower-dimensional, aggregated data, HDBSCAN proved superior for semantic pattern recognition, achieving a V-measure of 0.79. In contrast, k-means excelled in geometric quality, producing more compact clusters as shown by its higher Silhouette Score (0.57 vs. 0.52). This dichotomy supports our hypothesis that no single algorithm is universally optimal. When confronted with high-dimensional data, the "curse of dimensionality" degraded the performance of both algorithms. In this challenging scenario, the simpler geometric optimization of k-means was more resilient than the density-based approach of HDBSCAN. This result powerfully validates the need for our adaptive architecture, which can pivot its strategy based on the data's intrinsic structure.

A key advantage of our approach is its high degree of automation and ability to produce interpretable results. By automatically determining the optimal number of clusters, identifying eight modes that perfectly corresponded to our simulation scenarios, the approach obviates the need for the "a priori" parameter specification that limits algorithms like k-means. This automation directly tackles the challenges of model complexity and usability highlighted in related research [40,41]. The practical value of this design is reflected in the high scenario identification accuracy (up to 95.0%) and exceptional temporal coherence (0.94) of the cascade model. Furthermore, the robustness of the adaptive selection mechanism is enhanced by its use of a tolerance threshold. This feature ensures system stability by preventing frequent switching between strategies due to minor performance fluctuations, which is critical for deployment in real-world environments with noisy and variable data. Ultimately, these results confirm that the identified patterns are meaningful, actionable representations of traffic behavior, which is vital for downstream tasks like optimizing traffic signals to reduce $CO_2$ emissions [21,25]. By refining our previous work [20,23], this study delivers a more powerful and reliable tool for environmentally-oriented traffic management.

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

Our adaptive cascade approach is conceptually distinct from traditional ensemble methods. While existing ensembles, such as Bayesian techniques [6] or multi-algorithm voting schemes, typically seek to find a static consensus among different models, our approach uses an intelligent selection mechanism. It dynamically chooses the optimal algorithm based on data profiling rather than simply averaging results. This makes a direct comparison methodologically complex, as our framework solves a problem of adaptive algorithm selection, not consensus clustering.

When contextualized with other state-of-the-art methods, our approach offers a unique balance of performance, interpretability, and efficiency. For example, Spectral Clustering can identify non-convex shapes like HDBSCAN but is often more computationally expensive and sensitive to parameter choices, making it less suitable for a fully automated system. Gaussian Mixture Models offer a probabilistic framework for modeling fuzzy transitions between traffic states, but their underlying assumption of Gaussian-distributed data may not hold for the complex, irregular distributions characteristic of real-world traffic. Unlike these methods or hybrid spectral approaches [4], our cascade architecture relies on fewer geometric assumptions and provides a transparent, data-driven decision process, representing a clear innovation over simple model switching or parallel ensembles.

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

Despite its demonstrated strengths, the proposed approach has several limitations that define clear and important trajectories for future research. A primary limitation is the study's reliance on a synthetic dataset. While the use of a meticulously calibrated simulation provided an ideal environment for controlled validation, it introduces a degree of circularity, as the algorithm is tasked with recovering known, predefined patterns. This setting does not fully account for the non-stationary and often chaotic phenomena of real-world traffic. Therefore, future work must prioritize validating the approach on large-scale, real-world data from diverse urban sensor networks to test its generalizability and robustness. Beyond the source of the data, its scope presents another limitation. The current feature set does not account for crucial exogenous variables such as weather conditions, public events, or accidents, all of which significantly influence traffic patterns. A crucial next step is to integrate these external data sources into the feature engineering process to create a more context-aware and predictive model.

Further limitations relate to the structural generalizability of our approach, both in the feature space and the physical environment. As our results showed, the performance of the clustering algorithms degrades significantly in high-dimensional spaces due to the "curse of dimensionality." While our study highlighted this issue, a more systematic investigation into mitigation strategies is needed. Future research should explore the integration of advanced dimensionality reduction techniques, such as PCA or UMAP, as a formal pre-processing step within the adaptive pipeline. Similarly, the model's performance may be specific to the radial-concentric network topology of Khmelnytskyi. Its effectiveness and optimal parameters would likely require re-evaluation for cities with fundamentally different layouts, such as the grid-based systems common in North America. A comprehensive cross-city comparative study is therefore necessary to assess the true generalizability of the approach.

### *4.4. Computational Complexity and Scalability*

A key practical consideration is the computational cost of our cascade approach. The increased overhead from executing multiple algorithms and an evaluation layer could pose latency challenges for real-time applications. The overall complexity is governed by HDBSCAN, which scales quadratically with the number of time windows analyzed ($O(K^2)$). While the observed runtime of 6.84 seconds for our dataset is acceptable for offline analysis and strategic planning, optimization is necessary for deployment in real-time control systems. Potential strategies include implementing more efficient HDBSCAN variants, exploring approximate nearest neighbor methods, or developing a lightweight online version of the algorithm that updates cluster assignments incrementally.

## 5. Conclusions

This study introduced and validated a novel adaptive cascade clustering approach for high-fidelity urban traffic pattern recognition, a cornerstone for intelligent mobility in smart cities. By synergistically integrating HDBSCAN and k-means through a data-driven voting mechanism, our approach overcomes the inherent limitations of standalone algorithms. 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, an exceptional temporal coherence of 0.94 ensures the identified patterns are chronologically consistent and semantically meaningful representations of real-world traffic dynamics. 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. The computational demands and feature engineering sensitivity also present challenges for real-time implementation that future work must address.

Future research will focus on bridging the gap between simulation and real-world deployment by validating the system on live 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. This research provides a critical foundation for developing the intelligent transport systems needed to mitigate congestion and reduce environmental impact, paving the way toward realizing the vision of smarter, more sustainable cities.

**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.

**Data Availability Statement:** **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-trafic-simulate-cluster (accessed on 01 September 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 |
| 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 details the sensitivity of the HDBSCAN algorithm's performance to variations in the automated tuning hyperparameters $\beta$ (used for calculating 'min_samples') and $\gamma$ (used for calculating 'cluster_selection_epsilon'). As shown in Table A1, the resulting performance, measured by the ARI, is stable across a reasonable range of parameter values.

This confirms the robustness of our automated tuning approach, as the final outcome is not highly sensitive to the specific choices of these meta-parameters within their typical ranges. The values used in the main study ($\beta = 0.7$, $\gamma = 1.25$) are located within the observed high-performance plateau.

**Table A1.** Sensitivity of HDBSCAN performance (ARI) to variations in the hyperparameters $\beta$ and $\gamma$.

| **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. Additional Clustering Results for High-Dimensional Data

This appendix provides supplemental figures that visualize the clustering results on the high-dimensional combined traffic data, as discussed in Section 3.1.2. These plots illustrate the challenges posed by the "curse of dimensionality," where clear visual separation of clusters is significantly more difficult to achieve compared to the low-dimensional aggregated data. The data has been projected onto a 2D plane using PCA for visualization purposes.

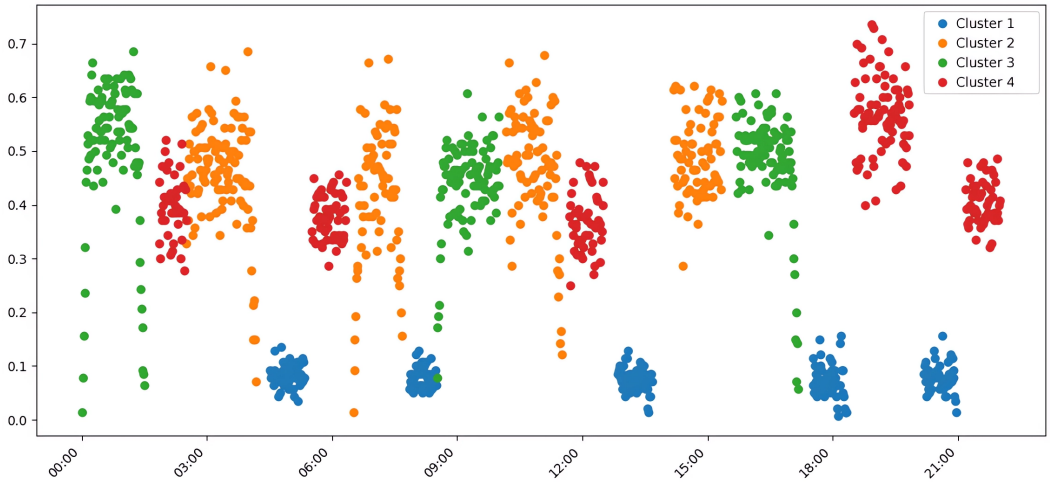



**Figure A1.** Visualization of HDBSCAN clustering results on high-dimensional combined traffic data (projected to 2D via PCA). The plot illustrates the algorithm's performance in a more complex feature space, where density estimation is more challenging.

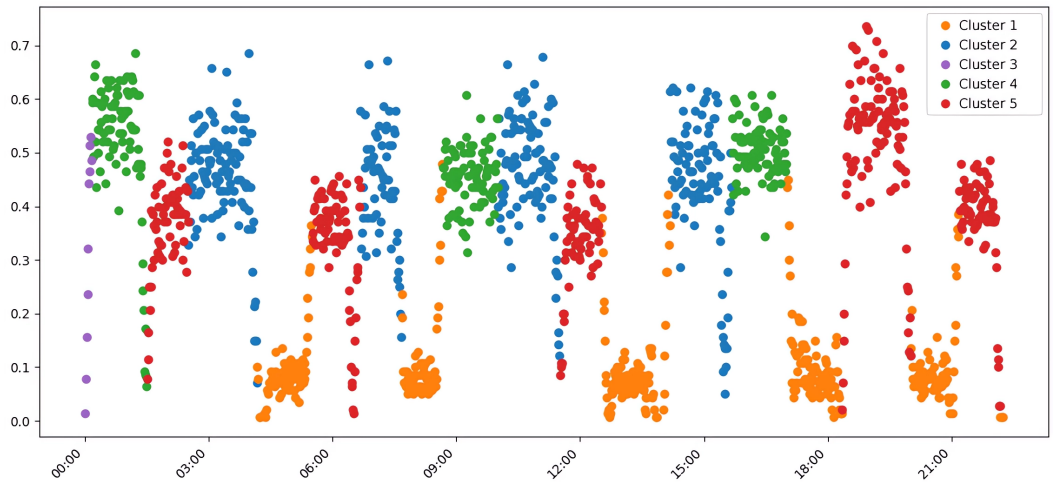



**Figure A2.** Visualization of k-means clustering (K=5) on high-dimensional combined traffic data (projected to 2D via PCA), showing how the algorithm partitions the data into five clusters in the high-dimensional space.

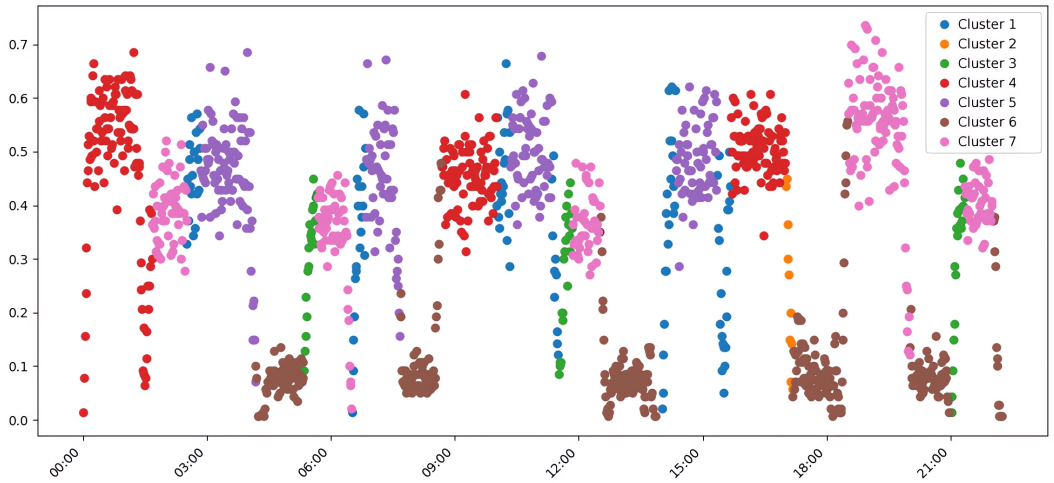



**Figure A3.** Visualization of k-means clustering (K=7) on high-dimensional combined traffic data (projected to 2D via PCA), illustrating the effect of increasing the number of clusters in a high-dimensional setting.

## Appendix C. Cluster Assignment Comparison Matrix

This appendix contains a detailed matrix that compares the cluster assignments for each individual time window across all three baseline clustering approaches, providing support for the analysis in Section 3.2.

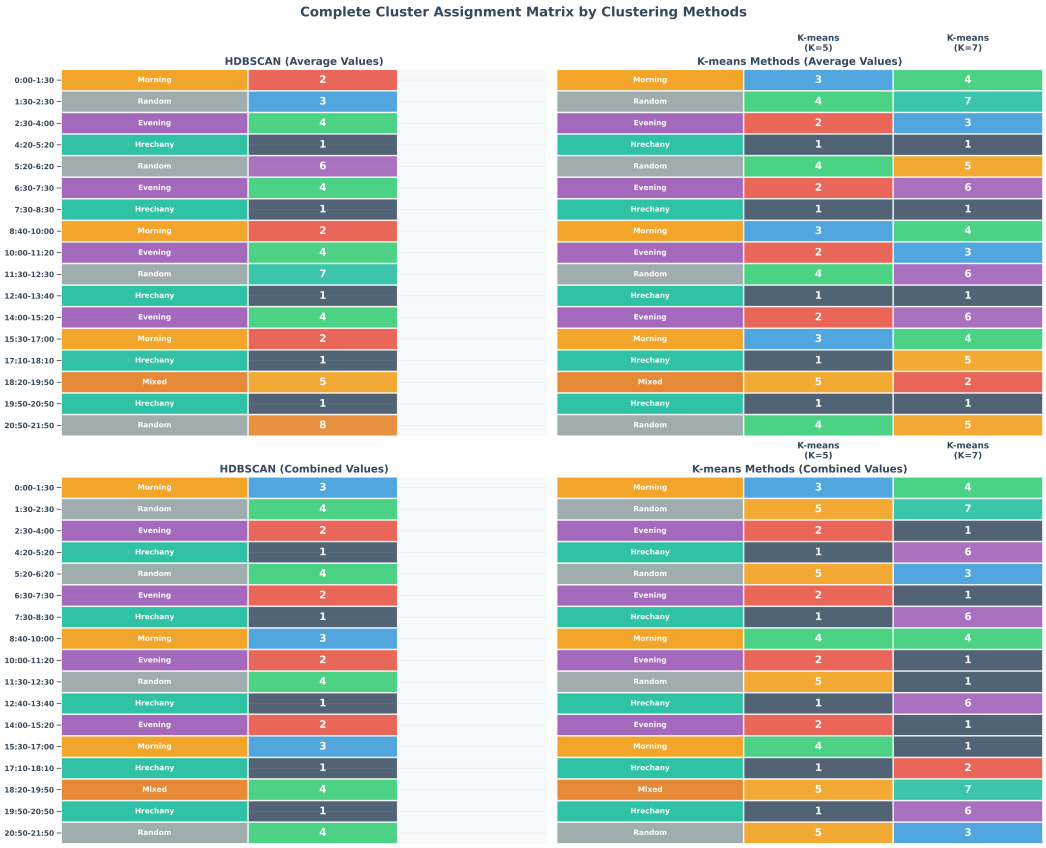



**Figure A4.** A matrix comparing the cluster assignments for each time window across the different clustering approaches (HDBSCAN, k-means K=5, and k-means K=7). The color-coded matrix visually represents the level of agreement and disagreement between the algorithms in classifying individual traffic states. Each row represents a time window, and each column represents a clustering algorithm. The color of each cell indicates the cluster label assigned to that time window by that algorithm.

## 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. https://doi.org/10.1109/TITS.2023.3305380.
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. https://doi.org/10.1109/TIV.2023.3349324.
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. https://doi.org/10.1109/TMC.2021.3079117.
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. https://doi.org/10.3390/su141711068.
5. Kim, K. Spatial contiguity-constrained hierarchical clustering for traffic prediction in bike sharing systems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 5754–5764. https://doi.org/10.1109/TITS.2021.3057596.
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. https://doi.org/10.1016/j.trc.2023.104032.
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. https://doi.org/10.1016/j.heliyon.2023.e17530.
8. Manziuk, E.; Krak, I.; Barmak, O.; Mazurets, O.; Kuznetsov, V.; Pylypiak, O. Structural alignment method of conceptual categories of ontology and formalized domain. In Proceedings of the International Workshop of IT-professionals on Artificial Intelligence (ProfIT AI 2021) 2021, Kharkiv, Ukraine, 20–21 September 2021, 2021; Vol. 3003, *CEUR Workshop Proceedings*, pp. 11–22. Available online: https://ceur-ws.org/Vol-3003/paper2.pdf (accessed on 01 September 2025).
9. 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, 2020; Vol. 2711, *CEUR Workshop Proceedings*, pp. 53–67.

10. Barmak, O.; Krak, Y.; Manziuk, E. Characteristics for choice of models in the ensembles classification. *Probl. Program.* **2018**, *2–3*, 171–179. https://doi.org/10.15407/pp2018.02.171.
11. Majstorović, Ž.; Tišljarić, L.; Ivanjko, E.; Carić, T. Urban traffic signal control under mixed traffic flows: Literature review. *Appl. Sci.* **2023**, *13*, 4484. https://doi.org/10.3390/app13074484.
12. 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. https://doi.org/10.3390/sym15091679.
13. 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. https://doi.org/10.3390/en16083585.
14. 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. https://doi.org/10.1080/21650020.2025.2481962.
15. Yarahmadi, A.; Morency, C.; Trepanier, M. New data-driven approach to generate typologies of road segments. *Transp. A Transp. Sci.* **2024**, *20*, 2163206. https://doi.org/10.1080/23249935.2022.2163206.
16. Khan, H.; Thakur, J. Smart traffic control: Machine learning for dynamic road traffic management in urban environments. *Multimed. Tools Appl.* **2025**, *84*, 10321–10345. https://doi.org/10.1007/s11042-024-19331-4.
17. Almukhalfi, H.; Noor, A.; Noor, T. Traffic management approaches using machine learning and deep learning techniques: A survey. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108147. https://doi.org/10.1016/j.engappai.2024.108147.
18. 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, 2022; pp. 1–4. https://doi.org/10.1109/INFOTEH53737.2022.9751263.
19. 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. https://doi.org/10.30574/wjarr.2024.22.3.1886.
20. 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); Pitsun, O.; Dyvak, M., Eds., Ternopil-Skomorochy, Ukraine, 8–9 May, 2025, 2025; Vol. 3974, *CEUR Workshop Proceedings*, pp. 1–9. Available online: https://ceur-ws.org/Vol-3974/paper01.pdf (accessed on 01 September 2025).
21. Jiang, J.; Han, C.; Zhao, W.; Wang, J. PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. *AAAI Conference on Artificial Intelligence* **2023**, *37*, 4365–4373. https://doi.org/10.1609/aaai.v37i4.25556.
22. Li, M.; Zhu, Z. Spatial-temporal fusion graph neural networks for traffic flow forecasting. *AAAI Conference on Artificial Intelligence* **2021**, *35*, 4189–4196. https://doi.org/10.1609/aaai.v35i5.16542.
23. 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); Hovorushchenko, T.; Savenko, O.; Popov, P.; Lysenko, S., Eds., Khmelnytskyi, Ukraine, 28 March 2024, 2024; Vol. 3675, *CEUR Workshop Proceedings*, pp. 118–136. Available online: https://ceur-ws.org/Vol-3675/paper9.pdf (accessed on 01 September 2025).
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*, 2013. https://doi.org/10.1038/s41467âĂŚ025âĂŚ56701âĂŚ4.
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), 2024, pp. 1609–1615. https://doi.org/10.1109/ICSCSS60660.2024.10625356.
26. Lv, Z.; Shang, W. Impacts of intelligent transportation systems on energy conservation and emission reduction of transport systems: A comprehensive review. *Green Technol. Sustain.* **2023**, *1*, 100002. https://doi.org/10.1016/j.grets.2023.100002.
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 Switzerland: Cham, 2025; pp. 154–171. https://doi.org/10.1007/978-3-031-98334-4_15.
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, 2018; pp. 2575–2582. https://doi.org/10.1109/ITSC.2018.8569938.
29. Hejlsberg, A.; Wiltamuth, S.; Golde, P. C# Language Specification. ECMA Standard ECMA-334, 1st Edition, 2001. Available online: https://www.ecma-international.org/wp-content/uploads/ECMA-334_1st_edition_december_2001.pdf (accessed on 01 September 2025).
30. Oliphant, T.E. Python for Scientific Computing. *Computing in Science & Engineering* **2007**, *9*, 10–20. https://doi.org/10.1109/MCSE.2007.58.

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 Proceedings of the Positioning and Power in Academic Publishing: Players, Agents and Agendas, 2016, pp. 87–90. https://doi.org/10.3233/978-1-61499-649-1-87.
32. Behnel, S.; Faassen, M.; Bicking, I. lxml – XML and HTML with Python. Software documentation, 2025. Available online: https://lxml.de/ (accessed on 01 September 2025).
33. Blech, M. xmltodict 0.14.2. PyPI Software Documentation, 2024. Available online: https://pypi.org/project/xmltodict/#description (accessed on 01 September 2025).
34. McKinney, W. Data Structures for statistical computing in Python. In Proceedings of the Proceedings of the 9th Python in Science Conference; van der Walt, S.; Millman, J., Eds., 2010, pp. 56–61. https://doi.org/10.25080/Majora-92bf1922-00a.
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. https://doi.org/10.1038/s41586-020-2649-2.
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. https://doi.org/10.21105/joss.00205.
38. Hunter, J. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. https://doi.org/10.1109/MCSE.2007.55.
39. Waskom, M. seaborn: Statistical data visualization. *J. Open Source Softw.* **2021**, *6*, 3021. https://doi.org/10.21105/joss.03021.
40. Afandizadeh, S.; Abdolahi, S.; Mirzahossein, H. Deep learning algorithms for traffic forecasting: A comprehensive review and comparison with classical ones. *J. Adv. Transp.* **2024**, *2024*, 9981657. https://doi.org/10.1155/2024/9981657.
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. https://doi.org/10.3390/s24123884.