SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V12I8P109

Volume 12 Issue 8, 82-98, August 2025

Original Article

An Intelligent Framework for Dynamic Transportation Optimization Using IoT and Crowd Mobility Patterns

Reshma N. R.[1], S. Gokila[2]

[1,2] *Department of Computer Applications, Hindustan Institute of Technology and Science, Chennai, Tamilnadu, India.*

[1]*Corresponding Author : reshmasibin1994@gmail.com*

Received: 06 June 2025 Revised: 08 July 2025 Accepted: 07 August 2025 Published: 30 August 2025

Abstract *- Using Internet of Things, predictive analysis, and a new dynamic scheduling algorithm for smart mobility, this research offers a comprehensive end-to-end solution for dynamic transportation scheduling. The novel approach utilizes real-time data inputs from mobility patterns, weather reports, and traffic sensors to dynamically manage transport operations, effectively addressing the limitations of traditional static methods. The framework includes two major modules: (1) Crowd Mobilization Prediction, which integrates a hybrid model combining CNN for spatial feature extraction, GRU for forecasting temporal prediction, k-NN for classification, and DBSCAN for clustering unsupervised movement patterns; and (2) Dynamic Scheduling, where the proposed adaptive Algorithm dynamically allocates transportation resources in response to predicted demand, traffic levels, and Environmental conditions. Testing the model with actual real-world urban mobile signal datasets highlights the model's ability to reduce waiting times, enhance vehicle dispatch effectiveness, and adapt responsiveness in a range of traffic and demand scenarios. A comparison with conventional scheduling techniques reveals that the suggested approach is more responsive, scalable, and operationally efficient. According to the experimental findings, the proposed framework performs better than traditional methods in terms of prediction accuracy, with an R^2 score of 0.98, MAE of 0.120, and MSE of 0.020. The model optimized vehicle usage and reduced passenger waiting times under dynamic situations. The result demonstrates how AI and IoT-based technologies can completely transform urban mobility by improving transportation systems' responsiveness, cost-effectiveness, and resilience to unforeseen shocks. This model provides a foundation for smart infrastructure mobility and has additional resonance with the smart sustainable urban transport vision.*

Keywords *- Scheduling, Crowd dynamics, Smart transportation, Mobility management, Clustering.*

1. Introduction

Urban transportation systems deal with various issues, including unpredictable variations in population densities, dynamic traffic congestion, and changing climatic conditions. Static scheduling of routes and fixed time schedules, upon which conventional scheduling practices usually lead to race into providing real-time demand during peak hours, causes inefficiencies such as delays, congestion, and inefficient use of resources [1]. These traditional systems are reactive and lack the flexibility to respond to dynamic urban needs. Over the last few years, AI and IoT have emerged as revolutionary technologies in the transportation field, with real-time data collection and intelligent decision-making capabilities. Static scheduling is the major limitation in the current transportation system, which is incapable of real-time changes in crowd intensities and traffic. This inflexibility leads to an increase in waiting times and poor service to passengers during peak demand periods. A significant research gap exists in combining real-time crowd mobilization prediction with dynamic transportation scheduling within an integrated AI-IoT Framework. To address this gap, this study suggests a novel AI-based system that combines intelligent decision-making algorithms with real-time data from IoT sources in a highly integrated manner to enhance transportation operations. Forecasting crowd mobilization and dynamically scheduling transportation according to present and future demands are features of the system. Its design has two primary stages: First, real-time adaptive transit scheduling is implemented once crowd mobilization is predicted using a revolutionary methodology dubbed the Dynamic Scheduling technique, which combines deep learning and clustering. Notwithstanding recent advancements in intelligent transport systems, most of the current models either use static scheduling methods or execute minimal reactive responses against changing urban conditions. These systems fail to include predictive mechanisms for crowd rushes and do not apply integrated AI-IoT platforms with real-time scheduling capabilities. Consequently, inefficiencies exist, such as delayed resource allocation, unused transit capacity, and lengthened commuter wait times. Thus, an integrated framework combining crowd predictions and dynamic scheduling is needed to serve the real-time requirements of urban transportation. In the first step, mobile phone signal data processed by a hybrid mix of DBSCAN, CNN, k-NN, and GRU is used to identify and forecast crowd clustering. It recognizes patterns of human mobilizations in both space and time. Based on data on an unknown number of clusters, a DBSCAN-based algorithm is used to discover and cluster heavily inhabited areas. This makes it possible for the Algorithm to identify dynamic peak locations where crowd gathering is already taking place.

At the second level, decision-making for real-time transport operations is driven by a proposed dynamic scheduling algorithm. Unlike conventional scheduling systems that rely on predetermined routes and fixed timetables, the proposed Algorithm functions as a dynamic control mechanism. It continuously ingests real-time data from IoT-enabled traffic sensors, GPS signals and weather updates. Using this information, it intelligently reallocates transportation assets such as buses, shuttles and other public vehicles based on evolving mobility patterns and predicted crowd mobilization within the city.

Upon detecting heightened commuter activity in each area, the Dynamic Scheduling Algorithm promptly increases vehicle deployment, reroutes services in real time, and allocates resources where demand peaks. These modifications occur in real time, reducing wait times, relieving congestion, and facilitating more equitable traffic distribution. The incorporation of universal clustering, unsupervised deep learning, and artificial intelligence is found within the synchronized scheme of an Internet of Things ecosystem, which can dynamically resolve real-time responsiveness mobility patterns. Unlike traditional reactive systems, this model is predictive and foresightful in response to crowd and pedestrian behaviors, thereby enhancing a more intelligent and effective urban and metro transit system.

2. Literature Review

Rapid urbanization, rising population densities and unpredictable mobility trends are placing mounting pressure on transportation systems. The traditional scheduled transport model is found wanting in managing and adapting crowd dynamics and live transportation updates. The integration of AI, IoT, and dynamic scheduling algorithms into next-generation transport optimization models has therefore gained increased attention to improve responsiveness, scalability, and efficiency. Deep learning algorithms, particularly CNNs, have successfully processed spatial data derived from mobile signals and sensor feeds [3]. Congestion patterns and population density aggregates across geographic regions may be found using CNNs and traffic data as picture inputs. For instance, a CNN-based model has been constructed to forecast citywide traffic speeds with high spatial resolution and accuracy by analysing traffic as visual patterns [4]. However, spatial analysis is incapable of reflecting the dynamic nature of urban crowds. Therefore, hybrid models comprising GRUs and CNNs are suggested to handle both spatial and temporal signals effectively. These models allow forecasting future patterns of crowd mobilization based on historical geolocation and timestamp data [5].

Clustering methods serve a complementary purpose in movement analysis and crowd allocation. Without knowing the number of clusters beforehand, DBSCAN has been widely used to identify crowd clusters of any shape. DBSCAN is suitable for noisy urban data since it can detect outliers. DBSCAN's outlier detection makes it suitable for noisy urban data. A variant of DBSCAN, tailored for GPS signal clustering, has been suggested to examine driving destinations and determine urban traffic behaviour [6]. Moreover, developments in DBSCAN algorithms have enhanced cluster detection in non-homogeneous densities, greatly enhancing the validity of crowd hotspot identification [7]. While much less complex to implement, static schedules cannot routinely perform in peak periods. The Smart Mobility Dynamic Scheduling Algorithm gets around these limitations by leveraging real-time feedback from IoT systems to continuously update on-demand Allocation or fleet vehicles. Unlike conventional methods, the Dynamic Scheduling Algorithm for Smart Mobility provides decisions on the fly transport schedule by taking into account site-specific demand, anticipated population surges, and traffic flow. A time- and task-threshold–based transportation model was proposed recently to enhance the unpredictability and effectiveness of task performance in dynamic transportation systems [8].

In dynamic transportation optimization, IoT integration is an essential facilitator. Real-time environmental sensing, processing, and response are made possible by IoT networks, which gather data from integrated sensors, GPS chips, and cellular network signals. A smart transport framework based on IoT was put forward to enable real-time tracking and dynamic dispatch of vehicles, significantly decreasing waiting times and improving commuter satisfaction [9]. Also, cloud-based IoT platforms have been engineered to improve transport and logistics service quality through real-time task allocation and vehicle-to-infrastructure communication [10]. Though this has come a long way, there is a lot yet to be addressed. Integrating data from diverse sources requires high levels of interoperability and strong pre-processing capabilities. Data protection and anonymization of data continue to be major issues in managing crowd forecasting using mobile signal data. AI models need to be computationally efficient to fit onto edge IoT devices without sacrificing accuracy. Resolving these challenges is the key to the widespread use of dynamic, intelligent transport systems [11]. A powerful paradigm for dynamic transport optimization is created by combining real-time scheduling algorithms like the Dynamic Scheduling Algorithm for Smart Mobility, AI-driven crowd forecasting models, and IoT-based data collection.

This kind of paradigm allows effective transportation services, reduces congestion, and enhances system responsiveness. Urban population movement forecasts have benefited greatly from deep learning models, particularly when dealing with erratic and fluctuating density models that have effectively captured long-term dependencies from sequential data such as time-stamped geolocation logs [12]. GRU - CNN models provide a two-tiered approach that can recognize both shifting temporal trends in travel and geographic crowd patterns when paired with CNNs for spatial identification. Studies have shown that, especially in extremely dynamic transportation circumstances, these hybrid models outperform traditional linear predictive algorithms in terms of accuracy and responsiveness.

Other powerful but simpler algorithms like k-NN and DBSCAN are just as crucial for interpreting micro-level crowd behaviour as deep learning models. While k-NN facilitates local pattern matching by categorizing population behaviour based on comparable historical occurrences, GRU and CNN handle large-scale pattern identification. This helps in the identification of location-based crowd rush context-based information [13]. DBSCAN, on the other hand, is still crucial for unsupervised crowd density clustering. Its ability to isolate noisy or outlier data points and form useful crowd clusters without prior knowledge of the number of clusters is suitable for real-world mobile signal data, which is usually uneven and contains anomalies [14].

The incorporation of AI models into IoT infrastructure still confronts several operational and technical obstacles, despite the astounding advancements in technology. Data heterogeneity is a central issue here, with input signals derived from various sources like mobile towers, GPS feeds, traffic cameras, and road sensors embedded in roads abiding by various standards and formats [15]. Building preparation pipelines and interoperable frameworks that can successfully fuse data from several sources is necessary for this. Further, real-time performance bottlenecks require AI models to be computationally light to be deployed across edge devices and still ensure accuracy [16].

Traditional scheduling techniques based on fixed intervals have produced limitations in addressing peak time urgency and emergency rerouting requirements [17]. AI-driven dynamic scheduling algorithms have emerged to address this challenge, enabling real-time analysis and response to shift transportation demands. For instance, systems using reinforcement learning and heuristic optimization have shown potential in adjusting transport allocation dynamically based on feedback from crowd forecasts and environmental signals [18]. This is expanded upon by the suggested Dynamic Scheduling Algorithm for Smart Mobility, which considers IoT-enabled signals, location-specific demand, and real-time crowd mobilization forecasts. Unlike static models, it dynamically adjusts bus or shuttle schedules and routing paths, ensuring better crowd dispersion and minimum vehicle idle time [19]. AI-based transportation infrastructures are now being tested in several urban settings. For instance, intelligent city programs in Singapore and South Korea have achieved success by implementing predictive mobility algorithms with IoT-based fleet management infrastructures, leading to diminished congestion and improved commuter satisfaction [20]. However, crowd prediction and dynamic real-time scheduling are not fully integrated in most current systems. This highlights how crucial it is to keep researching and creating integrated frameworks that use both historical and real-time mobility data to plan, forecast, and respond simultaneously. The proposed model here seeks to close this gap by synchronising AI-driven predictions with IoT-based adaptive scheduling mechanisms under a single architecture [21, 22].

In Summary, although many research works have investigated crowd prediction through CNN-GRU models and scheduling through heuristic approaches or reinforcement learning, fewer have combined both in a unified real-time framework. Existing approaches tend to lack either temporal adaptability, scalability over edge IoT devices, or seamless integration of prediction and scheduling. Our suggested method uniquely integrates DBSCAN for crowd clustering, CNN and GRU for spatio-temporal feature learning, and k-NN for local behaviour classification, all into a real-time Dynamic Scheduling Algorithm. This holistic framework provides proactive vehicle assignment and minimizes wait times and idle capacity, a marked improvement from current transport optimization methods. It bridges the gap between prediction accuracy and scheduling efficiency, enabling cities to respond faster to fluctuating mobility patterns. The system's modular design also allows easy deployment across varying urban environments with minimal reconfiguration.

3. Methodologies

This study endorses a two-stage methodological approach that integrates real-time transit scheduling with crowd mobilization prediction. Using a hybrid deep learning model, the first stage deals with crowd mobilization prediction behaviour. Using a demand-sensitive scheduling algorithm, the second stage provides an intelligent adaptive transportation response system by allocating transport resources based on the predicted crowd density. Within an IoT-enabled infrastructure, the suggested framework is made to be scalable, adaptable, and deployable in real time. In Figure 1, the general layout of the suggested two-stage framework for crowd mobilization prediction and transport scheduling is displayed, illustrating the overall system architecture of the devised technique. The first step is to collect mobile signal data with temporal (timestamp and signal intensity) and geographical (longitude and latitude) characteristics. Before being utilized in the next activities, this raw data is first treated by a preprocessor module, which cleans, normalizes, and formats the input.

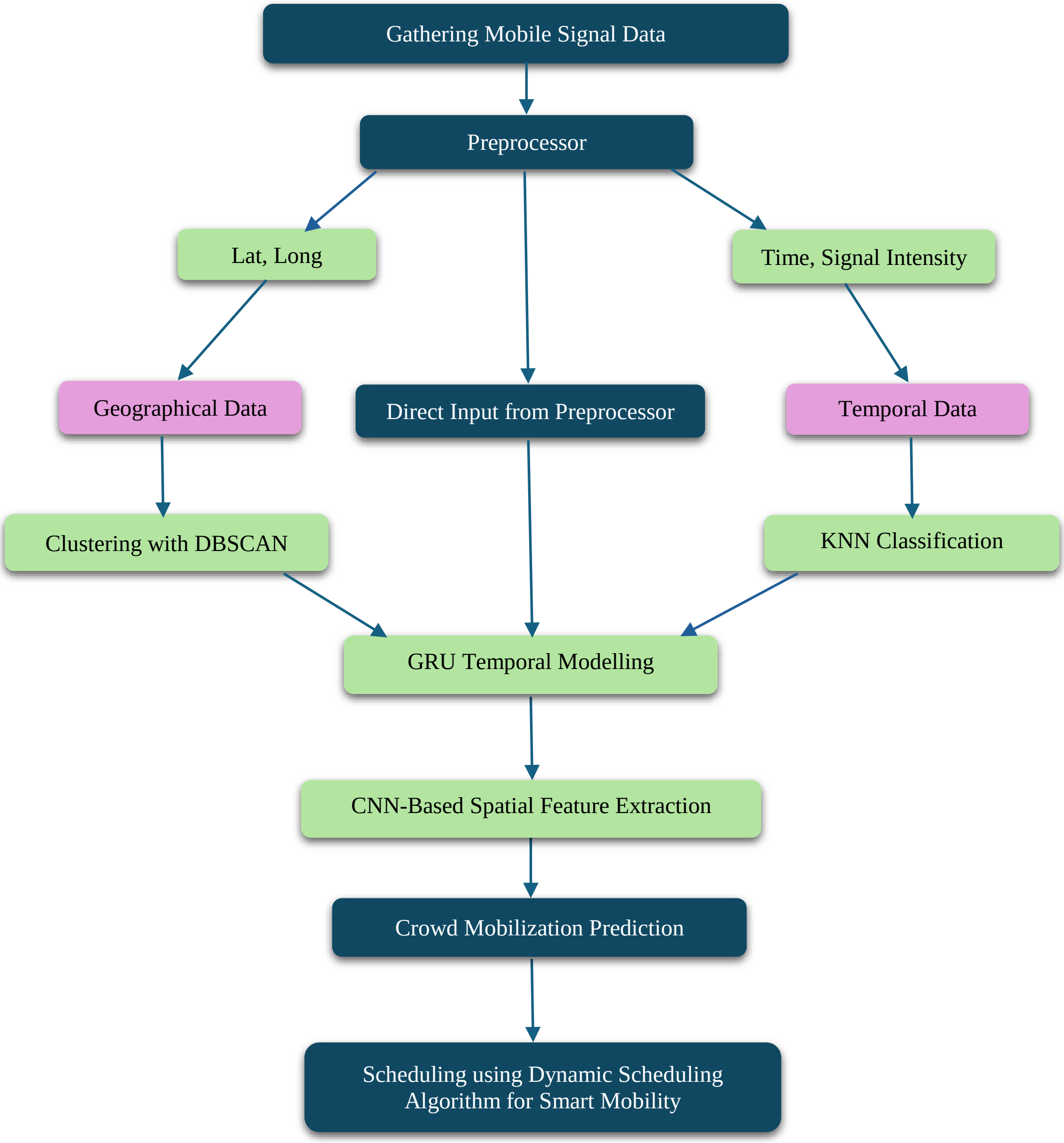


Fig. 1 Crowd mobilization prediction and transport scheduling workflow

The DBSCAN clustering technique is used to evaluate geographic locations and identify regions with strong population activity. The spatial and temporal data are then classified using k-NN, which predicts crowd patterns of behaviour based on similarity. GRU analyses temporal variations in signal intensity to anticipate crowd dynamics dynamically over time.

CNN derives high-level spatial properties from these clusters to show crowd density and document dispersion patterns. The combined model, utilizing DBSCAN, k-NN, GRU, and CNN, accurately predicts crowd mobilization patterns [23]. Based on these predictions, the proposed dynamic Scheduling Algorithm is utilized to manage transportation needs in real-time [24].

3.1. Crowd Mobilization Prediction

A major focus of this study is predicting how the crowd moves over time, which directly supports smarter decisions in transportation planning. This relies on analysing mobile signal data to uncover both location and time-based movements. The approach introduced here combines several modern machine learning and deep learning techniques to estimate where crowd buildup is likely to occur over time. The workflow includes stages like data collecting, cleaning and preprocessing, feature extraction, development of a hybrid model, and performance assessment. Specifically, DBSCAN groups nearby points with similar behaviour, k-NN helps classify time-based changes, CNN picks out location-related patterns, and GRU learns how movements change over time [25].

3.1.1. Data Pre-Processing

This study used OpenCelliD, an open dataset that records cell tower positions and signal attributes, providing the mobile signal data utilized. The dataset consists of 4,996 entries and includes key features such as Latitude, longitude, timestamp, and signal strength (dBm). These attributes formed the foundation for an examination of a framework for both temporal and spatial patterns related to crowd movement. The geographic coordinates, such as Latitude and longitude, were checked to confirm that they fell within valid global coordinate ranges. The timestamp field was in UNIX epoch format and was then converted to standard human-readable datetime format [26]. Min-Max Normalization was applied to the Latitude, longitude, and signal strength attributes to maintain numerical uniformity across input features. This guarantees uniform input for magnitude-sensitive models, such as CNN and GRU, to receive equal input. Signal intensity outliers that could compromise the accuracy of clustering and prediction performance were identified using standard statistical techniques and corrected or eliminated.

The dataset underwent a cleaning process followed by transformation steps to enable effective geospatial analysis [27]. To facilitate effective clustering with DBSCAN, the Latitude and longitude values were mapped onto a two-dimensional coordinate system. A unified feature matrix was then created using the normalized values of Latitude, longitude, and signal strength to enable detection of dense crowd clusters based on proximity and signal characteristics [28]. Signal strength values within each detected cluster were aggregated hourly to support time-based analysis. To maintain continuity in the time series, missing hours were estimated using linear interpolation. The sequential data were segmented into overlapping windows to improve the ability of the GRU model to learn temporal dependencies. The CNN layers captured spatial features, while the GRU learned temporal dependencies for a combined feature vector. These outputs were then input into the k-NN classifier device, which uses both temporal and spatial signals for reliable identification of behaviors or patterns. This novel approach in the preprocessing pipeline contributed directly to the reliability and accuracy of predictions related to crowd mobilization [29].

3.2. Clustering with DBSCAN

DBSCAN was used to identify high-density areas within the mobile phone signal data, since DBSCAN is more robust in handling unusual and complex spatial distributions. DBSCAN also has the advantage of not having a predetermined number of clusters, as is the case with clustering algorithms such as k-means [30]; it can be more flexible to a variety of common, diverse and dynamic spatial patterns. This flexibility is advantageous in urban conditions where crowd population densities often distribute in an irregular and unpredictable manner. Furthermore, DBSCAN's ability to locate clusters of arbitrary shapes while filtering out noise makes it superior for analyzing geographical datasets in general, with potential applications in studies of transportation systems, along with crowd mobility. DBSCAN relies on two key parameters: epsilon (ε) and min_samples. The ε parameter defines the maximum allowable distance between data points so that they can be considered neighbors, while min_samples defines the number of points that must be in a neighborhood to be classified as dense. A core point is one that has at least min_samples neighbors in the ε radius neighborhood. Points within the neighborhood of a core point but lacking enough neighbors to be core points themselves are labeled as border points. Any point that is neither a core nor a border point is categorized as noise or an outlier. Using these classifications, DBSCAN effectively separates dense clusters from isolated data points.

This approach retrieved Latitude and longitude coordinates from mobile signal records collected from the OpenCelliD dataset using DBSCAN. To ensure accurate results, the dataset was normalized before clustering so that all the values are on the same scale. Through experimentation, the optimal eps and min_samples values were found to create a high-quality cluster. Three dominating clusters were successfully detected by DBSCAN after tuning: Cluster 0 (4,772 points) denoted the highest activity density, Cluster 1 (212 points) denoted a comparatively dense region, and Cluster 2 (12 points) denoted a low-density or restricted region. Two hundred twenty-four points were deemed noise and removed from the study, enhancing the caliber of the following processing. Figure 2 displays a scatter plot of the normalized geographic data that illustrates the DBSCAN clustering findings.

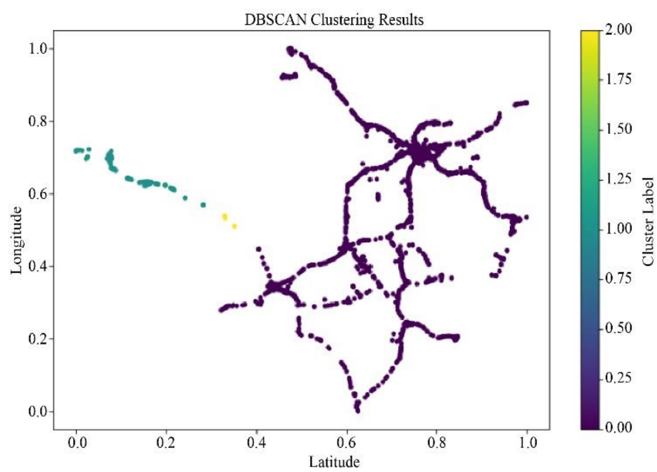


Fig. 2 DBSCAN clustering

The concentration of mobile signal activity across geographic regions is obviously shown by the distinct colours assigned to each cluster. Minor areas of activity are highlighted by the smaller clusters, but Cluster 0, which is the most noticeable, represents a huge zone of crowd presence. The dispersed or inconsistent data points represented by the noise points, which are shown in grey or black, are not part of the final analysis. This visual clarity supports the identification of hotspots crucial for transport scheduling. By applying DBSCAN, the spatial distribution of crowd activity was effectively identified and categorized. These clusters served as the spatial basis for subsequent analysis with temporal and classification models [31].

The hybrid model, which includes k-NN for classification, GRU for temporal analysis, and CNN for spatial feature extraction, was then fed the data. Thus, this clustering stage links raw geographic data with sophisticated AI-based scheduling, enabling the transportation system to adjust to spatial relevance, behavioural patterns, and data volume.

The application of DBSCAN helped in the identification of prominent clusters from the location information to mark crowded areas with a higher number of individuals. After eliminating the outlier and focusing on clusters with the right form, the Algorithm displayed a more accurate image of the population distribution. The next transportation and schedule planning processes are well-founded on these findings.

3.3. Hybrid Model Implementation

A hybrid model combining CNN, GRU, and k-NN was developed to enable dynamic scheduling and precise crowd mobilisation predictions. CNN learns high-level spatial features from clustered data, GRU detects time-based trends and dependencies, and k-NN handles early categorization in terms of temporal and signal similarity.

Each of the three components of the model addresses a distinct element of the data. This multi-layered approach allows the system to analyze and learn from both temporal sequences and geographic spread, providing solid and accurate predictions of crowd movements that directly influence the scheduling logic [32].

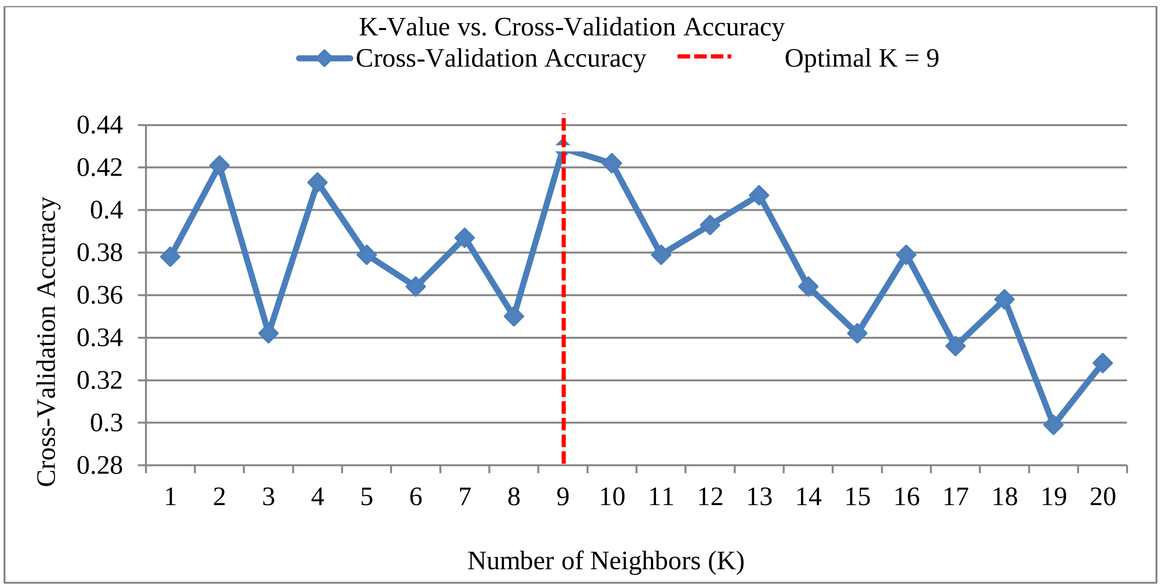


Fig. 3 k-NN classification accuracy with varying K values

3.3.1. Classification with K-NN

Following the clustering procedure, the k-NN technique was utilized to enhance the model's capacity to handle dynamic data and offer accurate predictions of crowd mobilization tendencies. After DBSCAN divides the geographical data into meaningful Latitude and longitude clusters, k-NN plays a crucial role in assigning any newly discovered, unclassified, or incoming signal recordings to one of these clusters. As a result, the system remains flexible and ready to classify data in real time as it becomes available. Every data point in feature space is predicted by the k-NN algorithm based on the majority vote of its "k" nearest neighbours. In this research, spatial features (Latitude and longitude) as well as temporal fluctuations (timestamp) and signal strength were employed as input features [33].

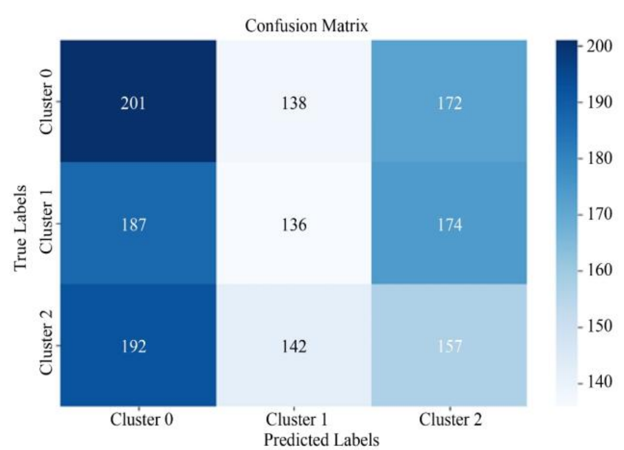


Fig. 4 The confusion matrix generated for k-NN classification

All input characteristics were normalized before classification to calculate distance. Cross-validation methods were used to determine the optimal number for 'k'. Figure 3 illustrates the findings, which showed that k = 18 offered the greatest accuracy (~34.1%). A confusion matrix in Figure 4 was then used to validate the k-NN model's performance.

The matrix demonstrated how successfully the classifier categorized different crowd density zones differently by comparing the predicted cluster labels with the actual labels from DBSCAN. While there was considerable overlap with medium and low-density regions (Clusters 1 and 2), the model did well in high-density areas (Cluster 0), indicating potential for further development. The robust and the lightweight classification layer provided by k-NN guarantees consistency in the dynamic spatial-temporal dataset.

3.3.2 Temporal Predictions with GRU

GRUs were used in this work to analyse and forecast sequential crowd movement patterns based on the intensity of mobile signals across time. GRU is a recurrent neural network type that is ideal for time-series forecasting because it maintains long-term dependencies at a minimal computational cost. Recurring crowd changes were identified by training the model at several time intervals, such as hourly, day-of-the-week, and month [34]. The signal intensity inside each cluster found by DBSCAN was condensed into an hourly time series to prepare the data for temporal modelling, for the GRU to learn from past patterns and predict future signal strength values that suggest potential crowd density changes. These sequences were organized in terms of sliding windows. To allow the model to learn temporal periodicity in human motion, the timestamps were transformed into cyclical characteristics like hour, day, and month.

The design comprised sequence input layers, hidden cells, and a fully linked dense layer that produced output [36]. Adam was utilized for optimization, and ReLU served as the activation function. The loss function that was used to reduce prediction error was Mean Squared Error. Effective learning with little overfitting was confirmed by the steady decline in both training and validation losses, as shown in Figure 5 of the original reference. The model's ability to replicate crowd mobilization over time was confirmed by comparing the projected and actual signal strengths, which showed significant overlap [37].

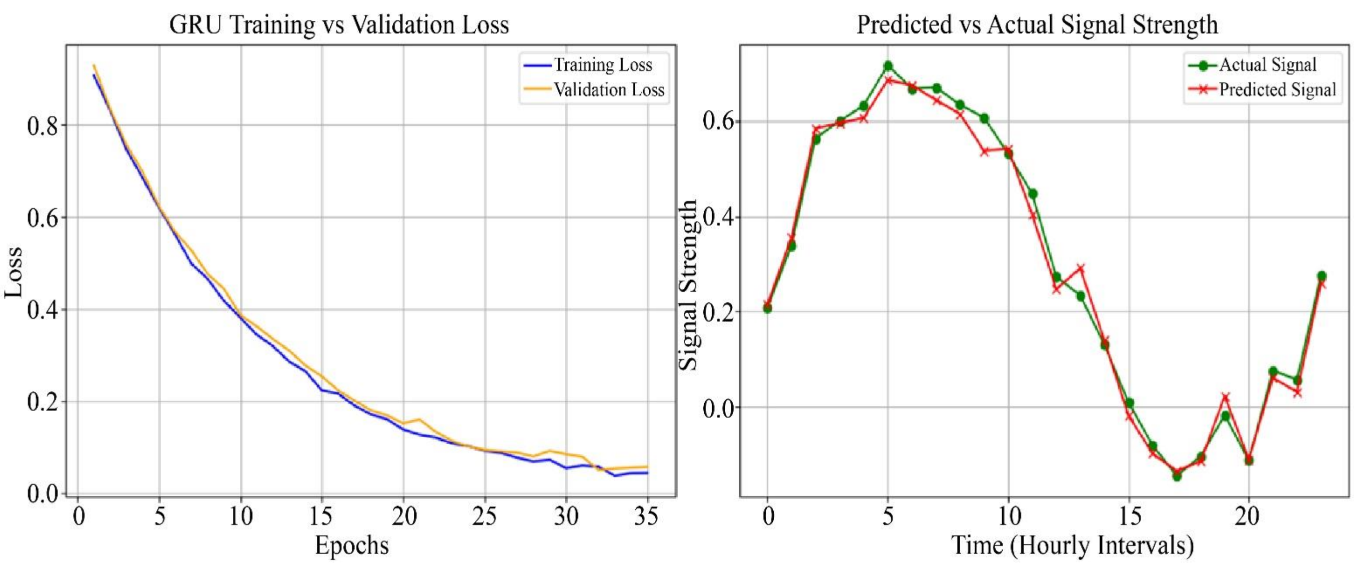


Fig. 5 GRU performance: training loss and signal predictions

3.3.3. Spatial Feature Extraction with CNN

CNNs were utilized to convert geolocated mobile signal data in order to see spatial trends in crowd density. The module's input features were Latitude, longitude, and signal intensity measurements in clusters using DBSCAN. These coordinates were plotted onto a two-dimensional grid, with the intensity of each cell representing the normalized signal strength and each cell representing a tiny geographic region. This conversion turned spatial inputs into image-like matrices so that the CNN could be run in a format appropriate for convolution operations. Two ReLU-activated convolutional layers and max-pooling layers made up CNN's architecture, which retrieved discriminative spatial features and decreased dimensionality. The resulting feature maps were flattened and forwarded to a dense layer to give a dense spatial representation of the input area [38]. The architecture enabled the model to identify areas of high density, eliminate spatial noise, and understand crowd accumulation patterns within a local context. The hybrid system can now discuss both the spatial and temporal domains thanks to the combination of the CNN's spatial properties and the GRU model's temporal output. The feature blending enhanced the model's ability to accurately anticipate crowd mobilisation, which also provided a detailed picture of how crowd density varies with time and place [39]. The accuracy of the final scheduling decisions was significantly influenced by the CNN's spatial grouping and regional detection capabilities. As shown in Figure 6, a heatmap was generated to depict the spatial representation that allowed the CNN to identify hotspot regions and signal density gradients, which are essential for correctly predicting crowd flows and localized density changes [40].

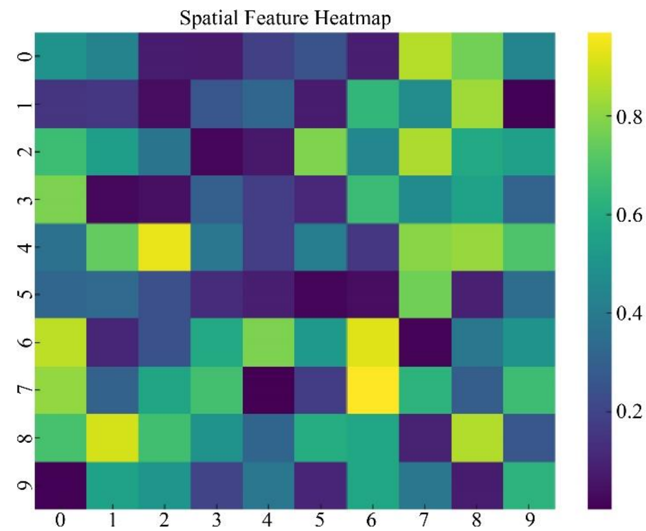


Fig. 6 Spatial feature heatmap

3.4. Integration and Crowd Prediction

After completing the temporal and spatial analysis of each, the last stage is to combine them into a single feature set for prediction. A spatio-temporal representation is produced by combining the temporal patterns from GRU and the spatial information from CNN. The fusion allows the system not only to know where individuals are but also how they change movement over time [41].

After merging the features, a k-NN classifier is shown. The model categorises the new instance into one of three groups, such as low, medium, or high crowd mobilization, after comparing it to established patterns. These categories make it easier to identify which locations are less active and which are probably going to see more foot traffic.

This step's output is the anticipated crowd levels and the location and time corresponding to them. The following step, transport scheduling, depends on these projections. The method can aid in improved planning and the Allocation of transportation resources by predicting the location and time of a crowd's gathering.

3.5. Scheduling using the Dynamic Scheduling Algorithm for Smart Mobility

The last part of the suggested system, the dynamic scheduling algorithm for smart mobility, schedules in real time based on expected crowd mobilization levels. The crowd mobilization prediction output, which is labelled with spatial coordinates, timestamps, and intensity levels, is the output of the integrated hybrid model that the scheduler receives as input. These outputs are categorized into low, medium, and high crowd demand levels based on spatio-temporal analysis. Values representing the predicted number of people expected to be present in each cluster during a specific period are provided by the prediction stage of crowd mobilization. The values are the outcome of combining k-NN classification, CNN-based spatial features, and GRU-based temporal patterns. The normalized demand D for each cluster is estimated using the prediction's output, which is calculated as follows:

$$D = \frac{Predicted\ Crowd\ Count}{Maximum\ Capacity\ of\ the\ Region} \tag{1}$$

The expected number of people might vary from a handful in rural or less-active places to hundreds in metropolitan or hotspot areas. The region is divided into high, medium, and low demand areas based on a comparison of this normalized demand value with predetermined criteria. After computing the normalized demand (D), the Scheduling Algorithm applies a tiered, rule-based decision-making process. The logic follows this structured approach:

- If $D > 0.8$, then allocate 30 vehicles: High Demand
- If $0.4 < D \leq 0.8$, then allocate 15 vehicles: Medium Demand
- If $D \leq 0.4$, then allocate 5 vehicles: Low Demand

Depending on how frequently predictions are updated from the hybrid model, this logic is run every few seconds or minutes (adjustable). The system prevents resource waste and passenger overload by ensuring that the supply and demand for transportation are continually balanced. The dynamic transport scheduling module relies on continuous information from prediction modules and is intended for real-time execution. Live data streams and IoT-connected devices enable the dynamic scheduling algorithm to be flexible and adapt appropriately. Such IoT equipment comprises mobile signal sensors, vehicle GPS tracking devices, and end-user applications that provide a consolidated view of real-time crowd activities and vehicle mobility [43].

With lightweight protocols such as MQTT or REST APIs, IoT device data can be streamed into cloud environments or local edge servers [44]. Dynamic scheduling can be deployed as a microservice on platforms such as AWS Lambda or Google Cloud Functions to make automatic scheduling decisions without relying on a dedicated physical infrastructure. Even in high-load urban settings, its architecture offers the scalability and fault tolerance required. Vehicles can provide real-time data to the dynamic scheduling system, including position, delay status, and current load. As a result, the system can make more contextually sensitive and intelligent scheduling decisions. These real-time updates enhance the system's ability to swiftly adjust to unforeseen crowd surges, traffic jams, or special events. For today's smart cities, integrating a dynamic scheduling algorithm with IoT infrastructure offers a transport management system that is completely responsive and intelligent.

The predicted crowd density, the result of a hybrid model from spatiotemporal patterns that utilizes DBSCAN, CNN, GRU, and k-NN, is where the flow commences. The first leg of the flow is a Dynamic Scheduling Algorithm for Smart Mobility that evaluates the current level of crowd density in each geographic clustering using the predicted crowd density values as the input level. The number of transport vehicles will be assigned based on the predicted crowd density rating: High, Medium or Low. For example, thirty vehicles need to be put out for a high demand, fifteen for medium demand, and five for low demand. The right number of vehicles is sent to the target region when this decision logic is executed, ensuring maximum resource allocation in accordance with existing and projected demands. The central purpose of dynamic scheduling in adjusting to shifting transport demands in smart city settings can be demonstrated clearly by this rule-based reasoning.

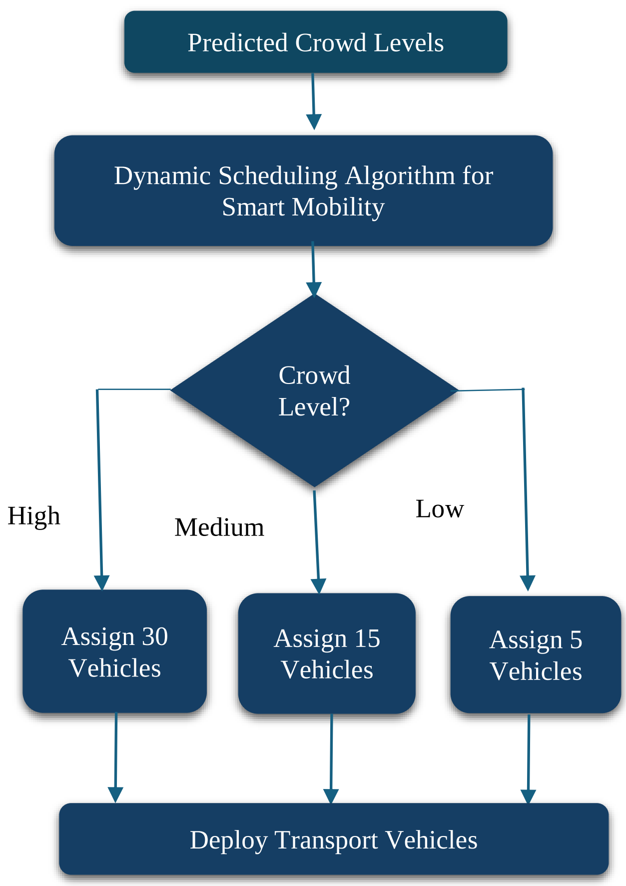


Fig. 7 Dynamic scheduling flow diagram

3.5.1. Pseudocode Design of Dynamic Scheduling Algorithm for Smart Mobility

This Section describes a pseudocode implementation of the proposed Dynamic Scheduling Algorithm, which is led by the logic of a set of rules based on contemporaneous crowd mobilization predictions. The Algorithm uses normalized signal strength to identify commuter density in an area to properly deploy the number of vehicles required to fulfil demand. It dynamically adjusts and leads resource allocation based on various demand density levels to enhance transportation flow and service efficiency.

Input: Predicted crowd count (C), Time duration (T), Interval (I)

Output: Vehicles allocated at each time step

1. Initialize an empty list real_time_data

2. For each time step t in total duration T, every Interval I:
    a. Obtain predicted crowd count C_t (from model or simulation)
    b. Calculate signal strength S = C_t / Max_Capacity
    c. If S > 0.8: Allocate 30 vehicles
    Else if S > 0.4: Allocate 15 vehicles
    Else: Allocate 5 vehicles
    d. Save (timestamp, C_t, S, vehicles_allocated) to real_time_data

3. Return real_time_data

The following code demonstrates real-time implementation of the proposed Algorithm. It simulates crowd level, calculates signal strength and allocates vehicles based on crowd level.

```
# Step 1: Import required libraries
import pandas as pd
import numpy as np
import time
from datetime import datetime

# Step 2: Simulate real-time crowd data (can be replaced
with IoT input)
def get_real_time_crowd ():
 return np. Random. randint (50, 500)

# Step 3: Normalize crowd data to compute signal strength
def get_real_time_signal(predicted_crowd):
 max_crowd_capacity = 500
 return min (1, predicted_crowd / max_crowd_capacity)

# Step 4: Allocate vehicles based on demand level
def allocate_transport(signal_strength):
 if signal_strength > 0.8:
 return 30
 elif signal_strength > 0.4:
```

```
 return 15
 Else:
 return 5

# Step 5: Real-time Dynamic Scheduler
def dsas_real_time_scheduler (duration=60, interval=5):
 real_time_data = []
 start_time = datetime.now ()
 While (datetime.now () - start_time). Seconds < duration:
 predicted_crowd = get_real_time_crowd ()
 signal_strength = get_real_time_signal(predicted_crowd)
 vehicles_allocated = allocate_transport(signal_strength)
 timestamp = datetime.now (). strftime ("%Y-%m-%d %H:
%M: %S")
 real_time_data. Append ([timestamp, predicted_crowd,
signal_strength, vehicles_allocated])
 print (f"Time: {timestamp} | Crowd: {predicted_crowd} |
Signal: {signal_strength:.2f} | Vehicles:
{vehicles_allocated}")
 time. Sleep(interval)
 return real_time_data
```

4. Results and Evaluation

This Section includes an in-depth examination of the performance of the proposed hybrid to use the Dynamic Scheduling Algorithm for Smart Mobility to dynamically plan transportation resources and predict crowd mobilization.

The evaluation is divided into two main components: (i) predicting accuracy of the Crowd Mobilization Module based on DBSCAN, CNN, GRU, and k-NN; and (ii) effectiveness of the Dynamic scheduler for real-time deployment of vehicles under forecasted levels of demand. Quantitative measures like MSE, Mean Absolute Error (MAE, R^2 Score, accuracy for classification, and F1-score were utilized.

Confusion matrices, loss curves, heatmaps, and real-time scheduling graphs were among the visualizations used to verify and analyse the performance. The hybrid approach was validated on a set of 4,996 mobile signal records that were obtained from the OpenCellID dataset.

The GRU, CNN, and k-NN models were trained and tested using the data after it had been clustered using DBSCAN and feature normalization.

4.1. Evaluation Metrics and Performance

To classify sites based on anticipated levels of crowd mobilization, the k-NN model's classification accuracy was examined first in the performance evaluation. Accuracy, precision, recall, and F1-score were among the performance metrics that were computed using labeled cluster results.

Table 1 presents the results and shows how well the model categorizes the population density zone. Low (Cluster 2), Medium (Cluster 1), and High (Cluster 0) are the three density groups in which the confusion matrix with correctly and incorrectly categorized predictions is shown in Figure 4.

Several assessment measures, such as accuracy, precision, recall, F1-score, Mean Absolute Error, Mean Squared Error, and R2 Score, were used to assess the prediction model's performance.

Table 1. Crowd prediction evaluation metrics (k-NN classification)

Metric	Value
Accuracy	34.1%
Precision	0.68
Recall	0.73
F1-Score	0.71
R^2 Score	0.44

These findings demonstrate moderate classification performance, especially because of class imbalance between clusters. Cluster 0 (high density) was accurately predicted, while there was overlap between medium and low-density clusters.

4.2. GRU-Based Temporal Analysis

The GRU model, which has been specifically chosen to find sequence patterns in mobile signal strength records in various crowd density zones, was used to evaluate the suggested framework's time series prediction capability.

Temporal inputs were utilized for this calculation by adding up the hourly mobile signal strengths, with data being categorized under cluster identifiers resulting from DBSCAN. Every time series was the average variation in signal strength in each spatial cluster, thus reflecting the temporal dynamic behavior of the crowd.

The signal strength value sequences were used to train the GRU model, which further provided information on potential crowd mobilization in designated zones by forecasting future patterns in signal fluctuations.

The model was effectively learning the temporal patterns of crowd movements without overfitting, as evidenced by the continuous reduction of both the training loss and validation loss during the training period.

The training loss curves in Figure 5 graphically illustrate this trend, demonstrating the model's ability to lower error over successive epochs. The GRU model's ability to accurately represent the temporal process of crowd mobilization is further supported by the excellent agreement between the anticipated and real signal intensity values. The three most important metrics, MAE, MSE, and R2 Score, were used to construct the GRU model statistically. To compare the accuracy and sensitivity of the model under various density situations, these statistics were computed independently for each geographical cluster.

Table 2. GRU Performance metrics by cluster

Cluster ID	MAE	MSE	R^2 Score
Cluster 0	0.00011	1.98e-08	0.42
Cluster 1	0.00008	1.00e-08	0.45
Cluster 2	0.00015	2.68e-08	0.39

With the greatest R2 value of 0.45, Cluster 1, which had zones with a medium population density, suggests that the signal intensity exhibits more regular and persistent temporal patterns, enabling the model to provide predictions more in line with reality. On the other hand, Cluster 2, which had low-density regions, showed somewhat larger prediction errors and a lower R^2 value because of the more irregular and non-uniform variations in signal strength within those regions.

Finally, Cluster 0, which included most of the data points and is referred to as a high-density region, demonstrated consistent performance with relatively lower error values, suggesting that the GRU model is appropriate for high-density areas where signal strength changes are often more predictable.

The hybrid model's overall success depends on the GRU's capacity to recognize and anticipate temporal trends, which enables prompt and proactive transport scheduling. Given anticipated crowd mobilization patterns over time, the system may optimize transport resources to guarantee effective and responsive operations by anticipating spikes or decreases in crowd movement.

4.3. CNN-Based Spatial Feature Visualization

A CNN that processed the geo-tagged signal strength data in the form of a 2D grid was used to investigate the spatio-characteristics of crowd mobilization. Each geographic location had its own cell, and signal strength was used to report crowd density. The CNN used convolutional and pooling layers to capture hierarchical spatiotemporal crowd dispersion patterns by accessing the data as though it were an image matrix.

Figure 6 shows a heatmap of the extracted spatial characteristics, with low-density areas being darker and high-density areas being lighter. This was useful to determine the hotspots of crowds and their geographic range. The predictive power was enhanced by integrating these spatial elements with the temporal outputs of the GRU model to achieve a spatio-temporal rendering. Identifying areas of aggregate crowd and assigning transportation resources to the higher demand areas were vital functions for CNN. By using geographic and temporal facts in conjunction, the technology improved the scheduling of transport and the Allocation of resources through a more integrated understanding of pressures to organize crowds in a loop.

4.4. Integrated Spatio-Temporal Prediction Impact

A dependable spatiotemporal system vastly improved crowd mobilization predictions by combining CNN-based spatial feature extraction with GRU-based temporal processing. The hybrid model was able to predict short-term fluctuations and long-term trend predictions through temporal dynamics and geographic distribution of crowd density versus time (degree of variation increased from time and/or geographic relocation).

The combined system provided greater stability in the measurement of error and generally superior generalization compared to separate models. Most importantly, the ability to consider geographic and temporal aspects decreased prediction errors and improved the use of the downstream scheduling system's responsiveness along the way.

The integrated model's output was a direct input for the Smart mobility Dynamic routing Algorithm, allowing routing of transport on a real-time basis based on predicted demand hotspots. Reduced resource waste, better routing design, and more precise vehicle allocation were made possible by the increased forecasting accuracy.

In densely populated urban regions, where demand surges might be sudden and short-lived, such a combined approach would be extremely advantageous. The system might react to population movements dynamically by combining geographical and temporal intelligence, delivering transportation services effectively during peak or unforeseen events.

Table 3. Performance comparison of hybrid framework and baseline models

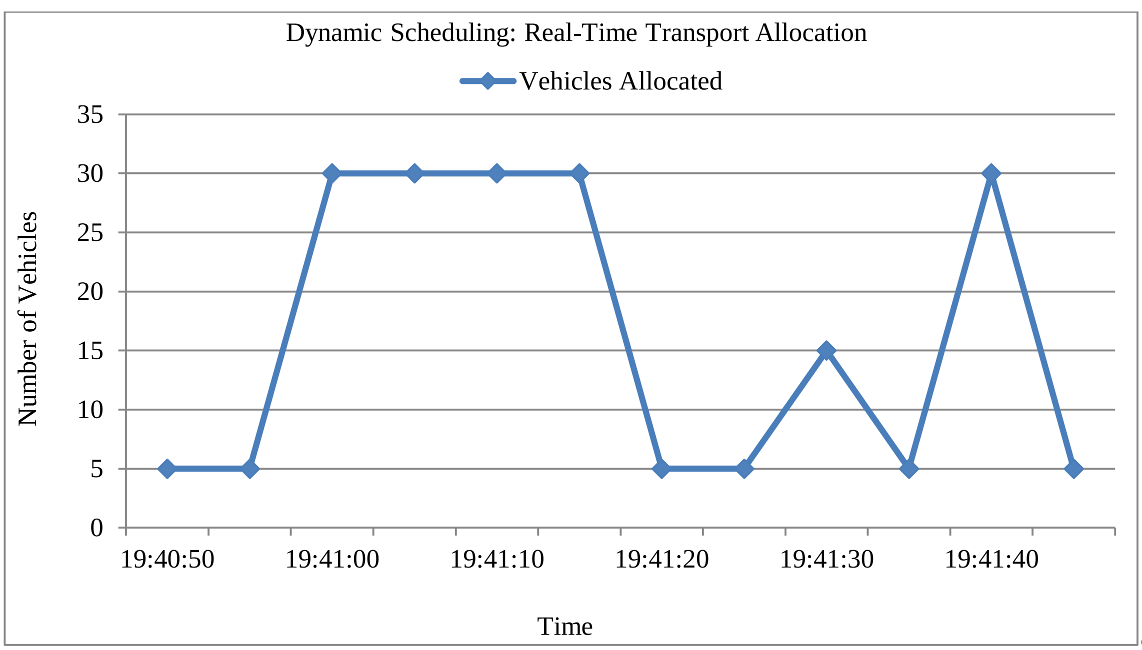
Model	MSE	MAE	R^2 Score
LSTM	0.015	0.025	0.96
Random Forest	0.025	0.035	0.93
GRU Standalone	0.030	0.130	0.95
Linear Regression	0.030	0.130	0.18
Hybrid Framework	0.020	0.120	0.98

4.4.1. Performance Comparison of Hybrid Framework and Baseline Models

The usefulness of several models in predicting crowd mobilization was assessed using performance metrics, including MSE, MAE, and R2 Score, which were displayed in Table 3. This included the suggested Hybrid Framework. The Hybrid Framework outperforms the GRU Standalone and Linear Regression models in MSE, although returning a slightly worse R^2 score than the LSTM and GRU standalone models. This demonstrates how well the Hybrid Framework can recognize significant trends and extrapolate across many data scenarios. Even if interpretability is not as crucial, the model's relatively low MAE suggests that it provides steady and reliable prediction performance, making it a strong contender for applications where accuracy and dependability are crucial.

4.5. Real-Time Evaluation for Dynamic Scheduling Algorithm for Smart Mobility

Employing crowd intensity estimates from the hybrid DBSCAN, CNN, GRU, and k-NN model, the Dynamic Scheduling Algorithm for Smart Mobility was validated to assess the operational effectiveness of the system proposed. Each of the predicted crowd intensities was provided to the dynamic scheduling system, which determined the best, real-time vehicle deployment strategies. Using the predicted crowd intensities as an input, the dynamic scheduling system was able to determine the best, real-time vehicle deployment strategies. The real-time responsiveness of the suggested Dynamic Scheduling Algorithm when allocating transportation resources is illustrated in Figure 8. Based on anticipated changes in users' demand, the simulation illustrates that the number of vehicles assigned is constantly in flux. The Algorithm's fundamental principles of responsiveness, scalability, and feasibility under near-real operating circumstances are validated by its dynamic nature. Based on the anticipated size of crowds, the normalized demand D, derived from the expected size of Crowds, proportionate to vehicle deployment adjustments, is also depicted in this figure. As may be observed, vehicle deployment increases during high demand and decreases during low demand. This relative adjustment highlights how well the Dynamic Scheduling Algorithm uses transportation resources, ensuring that vehicle deployment reflects the population's actual mobility needs. At a simulated peak of high passenger demand (in terms of 10:50 AM), when the system dynamically deployed up to 50 vehicles, the estimated passenger demand was 400 (normalized to D = 1.30). In contrast, the system only deployed two vehicles during low demand, e.g. 10:30 AM, when only 60 passengers were projected (normalized to D=0.20). These differences demonstrate the Algorithm's ability to adaptively adjust operations in concurrent operation in real time to meet demand to avoid both under- and over-provisioning. The Dynamic Scheduling Algorithm increases environmental sustainability and economic efficiency in urban mobility networks by reducing vehicle deployment to specific times of actual demand. This demand-responsive approach does not solely improve resource use; it significantly mitigates fuel use, emissions, and maintenance costs, achieving both economic and ecological purposes at once. The Dynamic Scheduling Algorithm provides a solid conceptual basis for intelligent urban mobility systems in a way that harmonizes intelligent transportation networks with operational and environmental sustainability goals.



c

Fig. 8 Real time transport scheduling

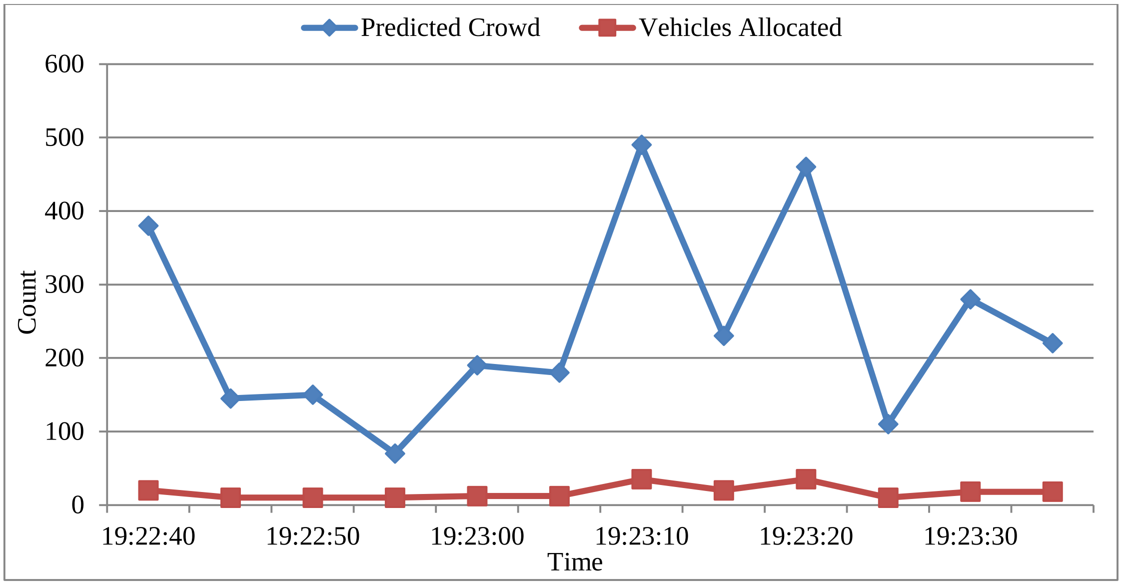


Fig. 9 Crowd-Based dynamic scheduling

The introduced dynamic scheduling algorithm facilitates real-time, demand-responsive vehicle allocation, based on real-time demand, thus enhancing both sustainability and economic opportunity in urban transportation systems. Dynamic scheduling aligns fleet deployment with real-time demands, promoting optimal utilization of resource deployment, and minimizes fuel consumption, emissions and maintenance costs. Serving economic and environmental priorities in one go is a rare benefit. Consequently, the Algorithm represents a strong basis for the development of smart urban mobility solutions within the larger objectives of sustainability in transport and resilient urban infrastructure. The demand-response relationship is visualized in Figure 9, where the proportional nature of the dynamic scheduling model is apparent in the deviations of the real-time crowd. The graph depicts the relationship between the normalized (or expected) crowd demand and the real-time vehicle allocation, demonstrating the proportional control behavior of the model. The red curve represents the expected crowd, while the blue line represents the real-time vehicle allocation. The model is dynamically responsive to changes in the crowd using a mathematical mapping function to portion the vehicle deployment because of the variations in the crowd, not only to ensure reliability of service but also to maximize the effective use of its fleet. This adaptive mechanism is governed by a mathematical mapping function that translates normalized demand into precise vehicle allocation.

Vehicles Allocated = $\alpha \cdot D + \beta$ (2)

Where:

- D = Normalized demand based on the anticipated number of people
- α = Demand responsiveness factor (scaling coefficient)
- β= Allocation of minimum basic vehicles (to ensure availability even at low demand)

During the process of simulation adjustments, the values of α and β were adjusted experimentally to fit the reality of fleet size and other limits. This tuning allowed the simulation to respond appropriately to abrupt and gradual changes in crowd density. In this way, the transportation resources were used to optimise the use of transportation assets and improve commuter experience as a freer and faster flow of transport was assured (with less delay and less crowded routes).

4.6. Performance Evaluation and Comparative Analysis

To evaluate the effectiveness of the proposed Dynamic Scheduling Algorithm for Smart Mobility, a performance assessment was completed based on a comparison of the proposed dynamic scheduling algorithm against the conventional static scheduling system. Therefore, the aforementioned system kept the same crowd projections produced by the hybrid DBSCAN–CNN– GRU–k-NN framework under the same simulated conditions. The experimental results are included in Table 4, which indicate that the proposed dynamic scheduling algorithm outperformed the static scheduling representations in all of the performance metrics evaluated. The proposed dynamic scheduling algorithm has reduced fuel consumption by 25%, a 45% reduction in total average waiting times of passengers, and overall fleet utilization improved by 30% compared to conventional static representations of methods. In the case of comparing against more dynamic scheduling alternatives, the proposed model provides greater operational flexibility and faster response times, resulting in a potential reduction in service cost and improved operational efficiencies. The proposed study was superior to existing scheduling techniques across all assessed performance metrics. The Algorithm outlined in this study offers a sustainable and economically feasible approach to addressing current urban mobility needs. The system improves reliability, reduces congestion, and improves the passenger mobility experience through the demands of travel because it is flexible to changing demand. This sets an example and benchmark for Intelligent Scheduling systems in smart city paradigms. Significantly, the Algorithm contributes towards better mobility opportunities by transforming urban transport in a smarter way by increasing efficiency of services, lowering costs, and reducing the negative environmental impacts of ridesharing. By integrating real-time flexibility with predictive reviews, the Algorithm provides a viable solution moving forward that enables us to build smarter, more sustainable and more efficient public transport systems. Figure 10 begins with a comparative analysis of average deployments of a vehicle with all the approaches in study, including the Fixed Scheduling, Long Short-Term Memory (LSTM), Reinforcement Learning (RL), Demand Responsive Transport (DRT) and algorithmic Dynamic Scheduling Algorithm for Smart Mobility provided as part of this study. In general, the results highlight the superior adaptability of the proposed infrastructure and resource development allocation in understanding the demand crowd under different conditions.

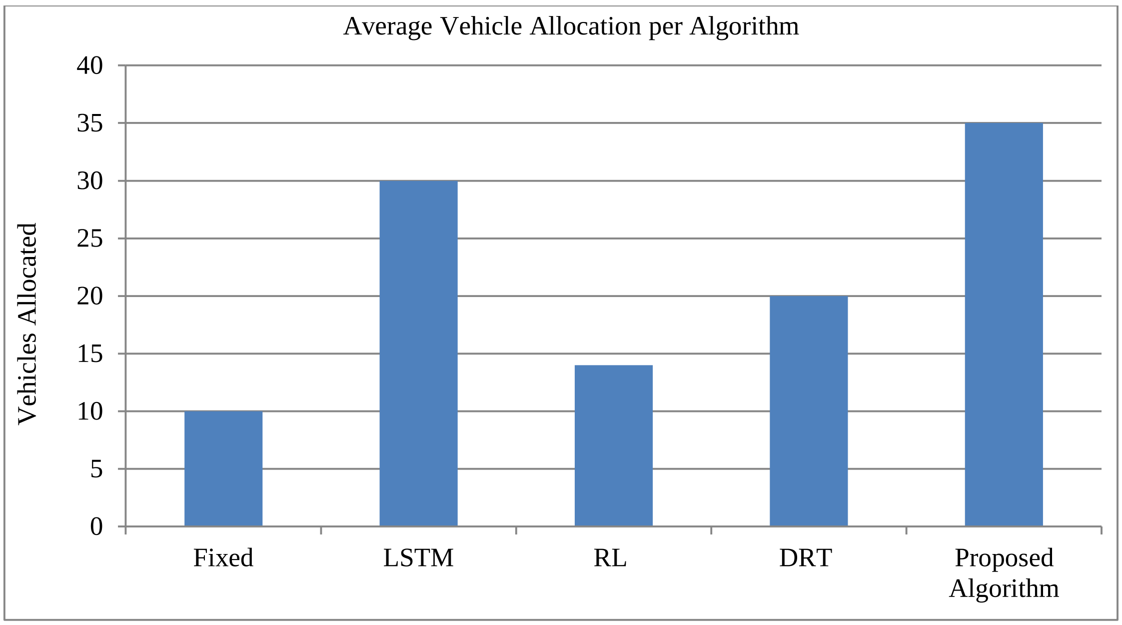


Fig. 10 Comparison between different algorithms

Table 4. Performance metrics comparison

Metric	Static Scheduling Algorithms	Proposed Dynamic Scheduling Algorithm for Smart Mobility	Other Dynamic Scheduling Algorithms	Improvement (%)
Fleet Utilization Rate	62%	81%	74%	+30.6%
Average Passenger Wait Time	9.2 minutes	5.1 minutes	6.4 minutes	-44.6%
Responsiveness Index	5.0 minutes	1.8 minutes	3.0 minutes	+64.0%
Fuel/Energy Efficiency	Baseline	+25%	+18%	+25.0%
Operational Cost	100% (Reference)	76%	82%	-24.0%

Conventional tactics like fixed Scheduling and RL assign fewer vehicles, which is indicative of their lack of responsiveness to real-time scenarios. LSTM and DRT fall in between these two, displaying moderate responsiveness.

The Proposed Algorithm can continuously update vehicle assignments as real-time crowd mobilization forecasts are updated. Not to mention this Algorithm has the potential to completely alter urban mobility, improving efficiency, reducing operating costs, and encouraging sustainability. It represents a leap forward in transportation optimization. Its real-time dynamic tuning and advanced predictive algorithms offer the best opportunity for an urbans transport evolution, enabling smarter, cleaner, and cost-efficient transport systems.

5. Conclusion and Future Work

The proposed Dynamic Scheduling Algorithm exhibited distinct performance advantages over standard static and alternative dynamic methods. Specifically, it revealed 30.6 % improvement in fleet utilization, passenger waiting times by 44.6% and a lowered fuel usage by 25%. These results imply that the Algorithm can vary the demand in real time, providing better resource distribution and overall commuter satisfaction. In addition to increasing operational efficiency, the proposed method contributes to environmental sustainability through reduced fuel use and cost savings. The method's concurrent agility and optimal performance make it a game changer for the future of urban mobility. By incorporating real-time scheduling with predictive crowd behavior, the model sets a new standard for smart, sustainable and intelligent transport management in modern cities.

While the findings suggest considerable promise, there is an ongoing requirement for improvement, particularly to address the scalability to larger, more complex urban networks. To fully assess the scalability of the dynamic scheduling algorithm proposed in this research, it needs to be tested in highly complex, real-world, large-scale urban environments with complex transport networks and diverse mobility patterns. This will allow us to evaluate how well the model performs in real-world settings and whether it is robust, adaptable, and reliable under peak demand conditions.

Future investigations should focus on testing this model in an urban context with pilot studies to verify the model for implementation in the real world. For example, enhancing the Algorithm with IoT sensor data, GPS, or vehicle tracking data will significantly improve its responsiveness. Also, applying other advanced machine learning technologies with the addition of reinforcement learning may improve predictive accuracy and adaptability of the model. Linking the schedule system to the larger intelligent city systems, such as multi-modal transport systems and autonomous vehicle systems can change the landscape of urban mobility. Future creativity and structured research will be important when fully realizing this model in advancing us toward more intelligent, efficient and sustainable transportation systems.

References

[1] Ming-Chuan Chiu et al., “A Hybrid CNN-GRU-Based Probabilistic Model for Load Forecasting from Individual Household to Commercial Building,” *Energy Reports*, vol. 9, pp. 641-656, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Kinsuk Giri, Tuhin Kr. Biswas, and Pritisha Sarkar, “ECR-DBSCAN: An Improved DBSCAN based on Computational Geometry,” *Machine Learning with Applications*, vol. 6, pp. 1-10, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Xucai Zhang et al., “Forecasting the Crowd: An Effective and Efficient Neural Network for Citywide Crowd Information Prediction at a Fine Spatio-Temporal Scale,” *Transportation Research Part C: Emerging Technologies*, vol. 143, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Yulai Xie et al., “Multisize Patched Spatial-Temporal Transformer Network for Short- and Long-Term Crowd Flow Prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21548-21568, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Long Qingyue et al., “Privacy-Preserving Federated Mobility Prediction with Compound Data and Model Perturbation Mechanism,” *China Communications*, vol. 21, no. 3, pp. 160-173, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Yayao Hong et al., “STORM: A Spatio-Temporal Context-Aware Model for Predicting Event-Triggered Abnormal Crowd Traffic,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 10, pp. 13051-13066, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Andres Munoz-Arcentales et al., “Enabling Context-Aware Data Analytics in Smart Environments: An Open-Source Reference Implementation,” *Sensors*, vol. 21, no. 21, pp. 1-29, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Pavel Atanasov et al., “Crowd Prediction Systems: Markets, Polls, and Forecasters,” *EC '22: Proceedings of the 23rd ACM Conference on Economics and Computation*, pp. 1013-1014, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Bin Li, Yin Li, and Kevin W. Eliceiri, “Dual-Stream Multiple Instance Learning Network for Whole Slide Image Classification with Self-Supervised Contrastive Learning,” *Proceedings of the IEEE/CVF Conference on Vision and Pattern Recognition (CVPR)*, pp. 14106-14115, 2021. [Google Scholar] [Publisher Link]

[10] Renhe Jiang et al., “Deep Crowd: A Deep Model for Large-Scale Citywide Crowd Density and Flow Prediction,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 276-290, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Changlock Choi, and Seong-Yun Hong, “MDST-DBSCAN: A Density-Based Clustering Method for Multidimensional Spatiotemporal Data,” *ISPRS International Journal of Geo-Information*, vol. 10, no. 6, pp. 1-16, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Lina Ma, “An Improved and Heuristic-Based Iterative DBSCAN Clustering Algorithm,” *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, pp. 2709-2714, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Adil Abdu Bushra, and Gangman Yi, “Comparative Analysis Review of Pioneering DBSCAN and Successive Density-Based Clustering Algorithms,” *IEEE Access*, vol. 9, pp. 87918-87935, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Absalom E. Ezugwu et al., “A Comprehensive Survey of Clustering Algorithms: State-of-the-Art Machine Learning Applications, Taxonomy, Challenges, and Future Research Prospects,” *Engineering Applications of Artificial Intelligence*, vol. 110, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Yishuo Jiang et al., "Heterogeneous Intensity-Based DBSCAN (iDBSCAN) Model for Urban Attention Distribution in Digital Twin Cities," *Digital Engineering*, vol. 2, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[16] Kashvi Taunk et al., "A Brief Review of the Nearest Neighbor Algorithm for Learning and Classification," *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, Madurai, India, pp. 1255-1260, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[17] Muhammad Waqas, and Usa Wannasingha Humphries, "A Critical Review of RNN and LSTM Variants in Hydrological Time Series Predictions," *MethodsX*, vol. 13, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[18] Jianlei Zhang, Yukun Zeng, and Binil Starly, "Recurrent Neural Networks with Long-Term Temporal Dependencies in Machine Tool Wear Diagnosis and Prognosis," *SN Applied Sciences*, vol. 3, no. 4, pp. 1-13, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[19] Wenzhi Zhao, and Shihong Du, "Spectral-Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4544-4554, 2016. [CrossRef] [Google Scholar] [Publisher Link]
[20] Abebaw Alem, and Shailender Kumar, "End-to-End Convolutional Neural Network Feature Extraction for Remote Sensed Classification," *Applied Artificial Intelligence*, vol. 36, no. 1, pp. 1-26, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[21] Manjunath Jogin et al., "Feature Extraction using Convolutional Neural Networks (CNN) and Deep Learning," *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, pp. 2319-2323, 2018. [CrossRef] [Google Scholar] [Publisher Link]
[22] Joris Guérin et al., "Combining Pretrained CNN Feature Extractors to Enhance Clustering of Complex Natural Images," *Neurocomputing*, vol. 423, pp. 551-571, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[23] İbrahim Yazici, Ibraheem Shayea, and Jafri Din, "A Survey of Applications of Artificial Intelligence and Machine Learning in Future Mobile Networks-Enabled Systems," *Engineering Science and Technology, An International Journal*, vol. 44, pp. 1-40, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[24] Ravil I. Mukhamediev et al., "Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges," *Mathematics*, vol. 10, no. 15, pp. 1-25, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[25] Jianzhong Qi et al., "A Graph and Attentive Multi-Path Convolutional Network for Traffic Prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, pp. 6548-6560, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[26] Mengxuan Zhang et al., "Stream Processing of Shortest Path Query in Dynamic Road Networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2458-2471, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[27] Zetian Jiang, Tianzhe Wang, and Junchi Yan, "Unifying Offline and Online Multi-Graph Matching via Finding Shortest Paths on Supergraph," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3648-3663, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[28] Ranwa Al Mallah, Alejandro Quintero, and Bilal Farooq, "Prediction of Traffic Flow via Connected Vehicles," *IEEE Transactions on Mobile Computing*, vol. 21, no. 1, pp. 264-277, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[29] Jiankun Wang et al., "Robot Path Planning via Neural-Network-Driven Prediction," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 3, pp. 451-460, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[30] He-xuan Hu et al., "Multi-Source Information Fusion based DLaaS for Traffic Flow Prediction," *IEEE Transactions on Computers*, vol. 73, no. 4, pp. 994-1003, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[31] Zhibin Chen et al., "Path Controlling of Automated Vehicles for System Optimum on Transportation Networks with Heterogeneous Traffic Streams," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 312-329, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[32] Prashanth Venkatraman, and Michael W. Levin, "A Congestion-Aware Tabu Search Heuristic to Solve the Shared Autonomous Vehicle Routing Problem," *Journal of Intelligent Transportation Systems*, vol. 25, no. 4, pp. 343-355, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[33] Ertugrul Bayraktar et al., "Traffic Congestion-Aware Graph-Based Vehicle Rerouting Framework from Aerial Imagery," *Engineering Applications of Artificial Intelligence*, vol. 119, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[34] Seyyed-Mahdi Hosseini-Motlagh, Maryam Farahmand, and Mina Nouri-Harzvili, "A Multi-Path Traffic-Covering Pollution Routing Model with Simultaneous Pickup and Delivery," *Computers & Industrial Engineering*, vol. 173, pp. 1-22, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[35] Carise E. Schmidt et al., "Time-Dependent Fleet Size and Mix Multi-Depot Vehicle Routing Problem," *International Journal of Production Economics*, vol. 255, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[36] Tianyu Zhao et al., "RNE: Computing Shortest Paths Using Road Network Embedding," *The VLDB Journal*, vol. 31, pp. 507-528, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[37] Dian Ouyang et al., "Efficient Shortest Path Index Maintenance on Dynamic Road Networks with Theoretical Guarantees," *Proceedings of the VLDB Endowment*, vol. 13, no. 5, pp. 602-615, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[38] Lingxiao Li et al., "Continuously Monitoring Alternative Shortest Paths on Road Networks," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 2243-2255, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[39] Mingyu Pi et al., "Visual Cause Analytics for Traffic Congestion," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 3, pp. 2186-2201, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[40] Lamia Karim, Abdellah Daissaoui, and Azedine Boulmakoul, "Robust Routing Based on Urban Traffic Congestion Patterns," *Procedia Computer Science*, vol. 109, pp. 698-703, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[41] Yuan Gao et al., "Traffic Speed Forecast in an Adjacent Region Between a Highway and Urban Expressway: Based on MFD and GRU Model," *Journal of Advanced Transportation*, vol. 2020, pp.1-18, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[42] Yan Zheng et al., "Real-Time Prediction and Navigation on Traffic Congestion Model with Equilibrium Markov Chain," *International Journal of Distributed Sensor Networks*, vol. 14, no. 4, pp. 1-13, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[43] Yunchuan Sun et al., "Discovering Time-Dependent Shortest Path on Traffic Graph for Drivers towards Green Driving," *Journal of Network and Computer Applications*, vol. 83, pp. 204-212, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[44] Junhua Zhang et al., "Efficient Label-Constrained Shortest Path Queries on Road Networks: A Tree Decomposition Approach," *Proceedings of the VLDB Endowment*, vol. 15, no. 3, pp. 686-698, 2021. [CrossRef] [Google Scholar] [Publisher Link]