applied sciences

MDPI

Article

Designing and Implementing a Public Urban Transport Scheduling System Based on Artificial Intelligence for Smart Cities

Cosmina-Mihaela Rosca [1], Adrian Stancu [2,*], Cosmin-Florinel Neculaiu [1] and Ionuț-Adrian Gortoescu [1]

[1] Department of Automatic Control, Computers, and Electronics, Faculty of Mechanical and Electrical Engineering, Petroleum-Gas University of Ploiesti, 100680 Ploiesti, Romania; cosmina.rosca@upg-ploiesti.ro (C.-M.R.)
[2] Department of Business Administration, Faculty of Economic Sciences, Petroleum-Gas University of Ploiesti, 100680 Ploiesti, Romania
* Correspondence: astancu@upg-ploiesti.ro

Abstract: Many countries encourage their populations to use public urban transport to decrease pollution and traffic congestion. However, this can generate overcrowded routes at certain times and low economic efficiency for public urban transport companies when buses carry few passengers. This article proposes a Public Urban Transport Scheduling System (PUTSS) algorithm for allocating a public urban transport fleet based on the number of passengers waiting for a bus and considering the efficiency of public urban transport companies. The PUTSS algorithm integrates artificial intelligence (AI) methods to identify the number of people waiting at each station through real-time image acquisition. The technique presented is Azure Computer Vision. In a case study, the accuracy of correctly identifying the number of persons in an image was computed using the Microsoft Azure Computer Vision service. The proposed PUTSS algorithm also uses Google Maps Service for congestion-level identification. Employing these modern tools in the algorithm makes improving public urban transport services possible. The algorithm is integrated into a software application developed in C#, simulating a real-world scenario involving two public urban transport vehicles. The global accuracy rate of 89.81% demonstrates the practical applicability of the software product.

Keywords: AI methods; public urban transport; optimization algorithm; smart cities; Microsoft Azure Computer Vision; Azure service; Google Maps Service; C#

check for updates

Citation: Rosca, C.-M.; Stancu, A.; Neculaiu, C.-F.; Gortoescu, I.-A. Designing and Implementing a Public Urban Transport Scheduling System Based on Artificial Intelligence for Smart Cities. *Appl. Sci.* **2024**, *14*, 8861. https://doi.org/10.3390/app14198861

Academic Editor: Christos Bouras

Received: 7 September 2024
Revised: 20 September 2024
Accepted: 28 September 2024
Published: 2 October 2024

1. Introduction

Public urban transportation aims to transport several passengers close to a vehicle's capacity simultaneously. This reduces urban traffic congestion, lowers transportation costs for passengers, and decreases pollution by reducing the number of cars on the road. Consequently, many countries encourage the population to use this type of transport. However, there are some disadvantages, mainly the discomfort caused by waiting times and often poor conditions. Generally, public urban transport is very crowded during peak hours, while during off-peak hours, the frequency of vehicles is reduced, resulting in longer waiting times. Given these considerations, there is a clear need to propose new methods to improve public urban transportation conditions, encouraging the population to opt for public urban transport instead of personal vehicles.

1.1. Public Urban Transport

To benefit from the advantages offered by public urban transport use among a population, it is necessary to identify why people consider this type of transport a last resort. Kita et al. [1] mention improving the quality of life by increasing opportunities for the population when public transport uses optimized routes and schedules. Focusing on user satisfaction, Khademi-Vidra et al. [2] reveal that users prioritize cleanliness and equipment of vehicles and stops, service frequency, timetable adherence, passenger information, and accessibility. By investigating transport delays, Nichols et al. [3] reveal that delays reduce job access by 4–9%, depending on travel time thresholds. Using a binary logistic regression model, their research finds that individuals experiencing frequent delays are less likely to opt for public transport over other motorized modes. The analysis incorporates travel survey data from approximately 6000 origin–destination pairs, measuring travel times with scheduled and actual data to capture the variability in commuting experiences.

Wong and Yap [4] analyze service changes and investigate the impact of public transport service adjustments on passenger demand by examining over 25 million individual journeys in London. Their research shows that a 1% increase in journey time leads to a 0.61% decrease in public transport demand. Thus, demand is more sensitive to service degradations than improvements, suggesting a slower recovery after reductions.

A method to improve public transportation services by using multi-modal social rerouting strategies, which involve asking a portion of passengers to change their departure time, line, or service, is presented by Luan et al. [5]. This approach balances the load on the network and improves the level of service by addressing demand-side issues in addition to the traditional supply-side measures. The strategies consider trains, buses, and trams with predefined timetables and use a centrally coordinated information system to guide passengers. Their research highlights the potential of social rerouting strategies to enhance public transport efficiency by dynamically managing passenger demand and reducing the impact of overcrowding and failed boardings [5].

Sharma et al. [6] explore a bi-modal public transport system that combines rail services with on-demand shuttles. Through agent-based simulations and analytical frameworks, their research demonstrates that bi-modal transit can significantly lower energy consumption while maintaining reasonable service quality. Their key findings reveal that adding more intermediate stops on rail lines can slow train speeds without affecting overall energy use or service quality performance. The analysis suggests that reducing the number of stops could be beneficial, allowing for faster transit.

Another method to improve the service quality of public transport is the correct time of arrival predictions. Traditional Expected Time of Arrival (ETA) estimators use data from bus service providers, including vehicle positions and traffic conditions. The proposed method improves these estimators by utilizing historical ETA records accessed through public transport application programming interfaces. By applying a machine learning (ML) scheme, the method predicts and corrects systematic errors in the existing ETA estimators. The results demonstrate significant improvements in error mean and standard deviation for bus fleets in Madrid and Paris [7].

Wang et al. [8] present another methodology that involves preprocessing input data, defining two optimization models—one aimed at minimizing waiting times and the other at reducing operational costs—and iteratively refining solutions until satisfactory results are achieved. Notably, a 10% reduction in fuel consumption could lead to significant cost savings and lower environmental pollution while extending vehicle lifespan.

Modern transport systems include the control center (CC), data centers, operators, and communication infrastructure to connect vehicles to the CC. Data transfer utilizes Global System for Mobile Communication (GSM) networks, General Packet Radio Service (GPRS), Short Message Service (SMS), and Internet communications. In this way, the system monitors transport process parameters in real time, improving service quality and user satisfaction [9].

Yakimov [10] proposes a new algorithm for developing initial routes, leveraging intermediate data on operational modes. The study employs genetic algorithms (GAs) to iteratively refine route configurations, using operations like crossover, mutation, and selection to evolve solutions across generations. This approach integrates traditional and heuristic optimization techniques, addressing multifactorial challenges in urban transport planning. High-quality initial data, including transport demand, supply, and passenger flow distributions, are essential parameters for this process.

A new proposed model focuses on defining operational areas and dynamic transport lines in low-demand integrated zones. It introduces the concept of "obliquity", referring to route deviations from main lines, and calculates coefficients to assess their impact. A case study in Radotín and Černošice near Prague demonstrates the model's feasibility, suggesting that optimizing routes and timetables can effectively address local transportation challenges [11].

By detecting crowdedness in public transport stations, the study by Gheorghiu et al. [12] presents a computer vision solution to assist local authorities and operators in making informed decisions regarding different measures and system responses. Advanced computer vision technology extracts information from images and video streams, improving traditional detection algorithms. Beyond crowdedness detection, applications include people and vehicle count, lost object identification, rail safety monitoring, and illegal crossing detection. The proposed solution addresses the immediate need to manage crowdedness and demonstrates the broader potential of computer vision technology in public transport systems decisions.

Kasatkina et al. [13] integrate linear algebra, graph theory, and statistical information processing to address route duplication within the Izhevsk, Udmurt Republic, public transport system. The analysis of 323 stopping points and 55 urban transport routes identifies duplicate routes through correlation analysis. Their findings suggest route synchronization, partially merging routes, and streamlining the network. These improvements contribute to better efficiency in Izhevsk's public transport and offer a model applicable to similar urban systems facing route duplication challenges.

Saddal et al. [14] develop a data processing pipeline to acquire and refine transportation data, structuring it as a nonlinear graph representing Islamabad's network. Schedules are converted into the General Transit Feed Specification format, enhancing travel time estimation. The dataset identifies mobility patterns and improves intra-city connectivity, supporting urban development efforts focused on route optimization and intelligent journey planning.

Centering on sustainable urban mobility, Douskova et al. [15] assess city transport conditions, specifically how car traffic flow affects public transport. A pilot study utilizing Google traffic data analyzes traffic characteristics and compares them with actual travel speeds of public transport on specific routes. Their findings aim to inform a broader research initiative to create a method for evaluating public transport operations across urban networks. By leveraging modern data sources like Google Traffic, their study reveals traffic dynamics and their impact on public transport, contributing to improved urban mobility solutions.

1.2. Traffic Management Systems

Some studies collectively contribute to advancing traffic management systems through innovative sensor technologies, real-time data acquisition, and intelligent algorithms. Guo et al. [16] introduce a new algorithm for optimizing traffic sensor networks. The algorithm considers multiple types of sensors and constructs an output matrix to enhance system observability. By integrating different sensor types, the system can achieve optimal sensor placement. The method was validated using an urban freeway model, demonstrating comprehensive coverage and strong representativeness. A dual-subsystem framework for traffic management was proposed by Elmrini and Amrani [17], collecting real-time traffic data, including vehicle count, speed, classification, and environmental data. The Control Station processes these data, enabling macroscopic and microscopic traffic flow evaluation and automated response to traffic issues. The system offers a cost-effective solution for real-time traffic monitoring.

Furthermore, integrating the Internet of Things (IoT) into traffic management systems aims to address issues such as traffic congestion, rule violations, and accidents. Jha et al. [18] develop an IoT-based Smart Traffic Controlling System to monitor and resolve traffic problems by collecting real-time data and communicating it to authorities. The system enhances traffic signal operation, reduces idle times at red lights, and improves overall traffic safety and management.

Syum et al. [19] propose a new model to predict traffic patterns and fill data gaps to make accurate traffic predictions even with limited data, enhancing intelligent transportation systems. Their study also integrates GPT-4 for user interaction, providing personalized and contextual responses. Another work [20] extensively reviews current traffic flow sensor technologies by detailing various sensors' operational strengths and limitations, aiding in informed decision-making for specific applications.

Furthermore, it discusses integrating alternative data sources like Wi-Fi, Bluetooth sensing, and connected vehicle data, highlighting their growing role in modern traffic management systems. Advances in Light Detection and Ranging (LiDAR) technology have enabled it to detect vehicles, cyclists, and pedestrians at intersections. A methodology to transform LiDAR data into Purdue Probe Diagrams is presented by Saldivar-Carranza and Bullock [21], allowing for performance measures such as arrivals on green, split failures, downstream blockage, and control delay level of service. A case study demonstrates the reliability of these measures, showing close agreement with connected vehicle data, thus validating the methodology.

Artificial intelligence (AI) is revolutionizing traffic transport, mainly through advancements in computer vision and image processing. To address these challenges, advancements in intelligent vehicle technologies, such as sensors and communication networks, offer transformative potential [22]. For instance, platooning for autonomous vehicles, which travel in groups at high speeds with minimal distances, aims to enhance road traffic efficiency. This requires precise vehicle positioning, which can be improved using images captured by onboard cameras and Global Positioning System (GPS) data [22]. Innovative systems utilize ML algorithms for route optimization, cargo volume forecasting, predictive fleet maintenance, and real-time vehicle tracking [23].

AI methodologies are anticipated to assist aviation personnel in traffic, potentially leading to single-pilot operations and AI-based air traffic management [24]. AI-based traffic surveillance technologies already assist in real-time traffic incident detection and alert systems using computer vision [25]. AI simulation techniques have been applied to assess pedestrian flow at intersections, helping predict traffic capacity and optimize signal control. These methods use neural networks and fuzzy logic to model and visualize traffic flows, aiding in better traffic management decisions [26].

AI's application in traffic management aims to overcome congestion, safety concerns, and environmental impacts. Deep Reinforcement Learning (DRL) is one approach explored to optimize traffic signal configurations, improving traffic throughput and alleviating congestion [27,28]. AI's potential to revolutionize urban traffic management lies in its ability to provide real-time, adaptive solutions to complex traffic issues [29,30].

Another approach used in the literature is autonomous vehicles (AVs), which aim to reduce traffic congestion, air pollution, and fuel consumption [31]. One innovative approach to managing AVs is converting their movement into a simulated train model on a virtual track. This method, employing a "particle swarm" optimization algorithm, aims to improve road safety and the integration of AVs with human-driven vehicles, ensuring efficient traffic management [32]. Additionally, a proportional–integral–derivative controller with an adaptive slide surface was proposed for trajectory tracking of fully actuated vessels, showcasing the potential of advanced control algorithms in AV management [33].

In combining AVs and human-driven vehicles, mixed traffic flow remains a significant research focus. A new cellular automata model revealed that a 30% AV penetration rate significantly increases traffic flow, though benefits plateau between 40% and 60%. Equipping human-driven vehicles with communication devices enhances traffic efficiency at lower AV penetration rates [34]. Introducing AVs as bus bays can improve these metrics, especially in heavy traffic conditions [35]. Optimized route selection and traffic control using deep neural networks, aided by Golden Levy optimization, can further enhance AV traffic management [36].

Simulation studies using actual traffic data in mixed traffic conditions indicate that AVs can adapt to different driving styles, improving road safety and efficiency [37]. Employing real-time data, ML-based traffic management strategies show promise in reducing congestion and emissions, offering significant benefits for urban transportation systems [38]. Despite the potential of AVs to transform urban traffic, low AV adoption rates pose challenges, highlighting the need for further research to address these limitations and ensure seamless integration into existing traffic systems [39].

Another proposal for public traffic management is the IoT approach. Mondal and Rehena [40] use stationary sensors and the k-means clustering algorithm to classify road segments based on traffic density and speed. Efficient traffic management in smart cities can be achieved by pre-estimating vehicle numbers at crowded junctions using image processing and IoT technology, which conveys real-time vehicle counts to control centers [41].

A dynamic IoT-based traffic management system using ultrasonic sensors regulates traffic light durations based on real-time congestion levels. This system was developed in the following three phases: simulation, IoT system development, and hardware implementation [42]. The rise of IoT offers new opportunities for inventions, such as an energy-saving electrical device surveillance and control system. This system uses wireless technology for small areas and wired configuration for larger areas to improve fault detection in lighting appliances [43]. An image processing system can calculate fines and connect drivers with the traffic system via IoT technology [44].

A smart traffic management system using IoT and solar-powered streetlights can reduce energy wastage in traffic monitoring. This system adjusts signal timings based on traffic densities and activates streetlights only when needed [45]. Intelligent traffic management systems using IoT and Long Range (LoRa) technology can monitor and control traffic flow, reducing congestion and emissions while providing real-time air quality data [46].

Integrating ultrasonic sensors with Raspberry Pi and IoT allows for smart security and road safety enhancements, gradually improving the overall traffic system [47]. To address frequent traffic congestion and uncontrollable commuting times, a collaborative intelligent traffic planning algorithm uses AI and traffic sensors to optimize signal durations and driving paths dynamically. This method improves on-time arrival ratios and meets commuting delay requirements [48].

To improve the effectiveness of the Internet of Vehicles (IoV), the study [49] proposes a Multi-Criteria Decision-Making (MCDM) approach, utilizing CRITIC and TOPSIS methodologies to select optimal choices based on qualitative and quantitative factors. By analyzing data from sensors, GPS, and communication networks, this research seeks to optimize IoV architecture, enhancing system performance and supporting the development of smart, human-centric transportation solutions.

The main contributions of this paper are as follows:

- Designing a Public Urban Transport Scheduling System (PUTSS) algorithm that has dynamic decision-making capabilities in real-time urban transportation management by adjusting bus schedules, optimizing service for passengers while minimizing operational costs for the transport provider;
- Integrating Google Maps Service data that identify congestion levels with AI techniques that estimate the number of people at transit stations;
- Implementing Microsoft Azure Computer Vision services to monitor and count passengers at each station in real time and address sudden changes in passenger numbers.

This paper is structured into five sections. Section 2 presents the materials and methods employed to design the proposed algorithm. Section 3 provides insight into the results of the tests of the proposed algorithm modules. The discussion, limitations, and future research are described in Section 4. Section 5 reveals the conclusions.

2. Materials and Methods

The proposed PUTSS algorithm employs two components to identify the congestion levels on the route. The first component uses statistical data, whereas the second applies an intelligent computer vision service to identify the number of people at each station in real time. Both components are equally important and are described as follows:

- The Statistical Data Component (SDC) uses historical and real-time statistical data to predict passenger numbers at various stations. The SDC prevents scenarios where passengers arrive at a station just before the bus, which would not allow the algorithm to adjust the fleet frequency in real time. It identifies patterns and trends in passenger flow based on time of day, day of the week, and special events. The advantages are as follows:
    - Helps in planning and deploying resources efficiently based on historical data;
    - Allows for preemptive adjustments in fleet frequency to manage expected congestion.
- The Real-Time Computer Vision Component (RTCVC) employs intelligent computer vision services to monitor and count passengers at each station in real time. The RTCVC allows for real-time adjustments to fleet frequency, addressing sudden changes in passenger numbers. It handles unexpected situations such as heavy rain, concerts, or other events that cause deviations from the usual statistical patterns. The RTCVC's advantages are as follows:
    - Provides immediate insights into the current number of passengers waiting at each station;
    - Enables the system to adjust the fleet dynamically, ensuring optimal service even during unpredicted spikes in demand.

By visualizing data in real time, the algorithm can accurately adjust the vehicle fleet, ensuring optimal transport conditions.

2.1. SDC Module

Google Maps Service was used to identify each station to implement the SDC module. Subsequently, Selenium WebDriver was employed to retrieve data related to congestion levels at each hour. Google Maps Service provides real-time congestion levels and compares them to the estimated daily average, theoretically known to the public urban transport provider. This allows the proposed algorithm to adjust the number of buses in real time based on the varying congestion levels.

To implement the SDC module, the following steps were taken:

1. Station identification with Google Maps Service. Each public urban transport station was identified using Google Maps Service for the analyzed route;
2. Data retrieval using Selenium WebDriver to scrape data related to congestion levels in each hour of the day from Google Maps Service. Selenium WebDriver's automated browsing capabilities allowed for systematic data collection;
3. Real-time congestion levels were computed to compare current level congestion against the estimated daily averages. The estimated daily average congestion levels are known to the public urban transport provider and serve as a baseline for comparison;
4. Algorithm adjustment based on real-time data. By comparing real-time congestion data with historical daily averages, the algorithm can make informed decisions about adjusting the frequency and deployment of public urban transport vehicles.

The SDC calculates the congestion factor (*SDCF*) across the route using relation (1). Google Maps Service describes various levels of traffic congestion and conditions, including light, moderate, heavier-than-usual, heavy, peak hours, and standstill. All these statuses are included in relation (1).

$$SDCF = \frac{\sum_{i=1}^{n} cf_i}{n}, \tag{1}$$

where:

SDCF ranges between 1 and 5 and is a generic mathematical tool for measuring the congestion levels on a route;

n—the total number of stations along a route;

cf_i—congestion factor corresponding to station i, using the correlation from Table 1.

Table 1. Correlation between the congestion factor and Google Maps Service status.

Google Maps Service Status	Congestion Factor
Light	1
Moderate	2
Heavier-than-usual	3
Heavy	4
Peak hours	5
Standstill	6

The congestion factor is assigned numeric values based on the observed traffic status from Google Maps Service, ranging from 1 for light traffic to 6 for standstill conditions. This relationship is further integrated into a generic decision-making formula.

2.2. RTCVC Module

Although the algorithm previously presented allows for collecting information related to station occupancy percentages, it does not provide a comprehensive perspective on the actual number of passengers in the stations. Therefore, it was decided to implement a complementary algorithm based on object recognition in images using the Azure platform and computer vision AI service. The service utilizes real-time images acquired from bus stations.

The analysis performed on the station images yielded a list of coordinates corresponding to identified individuals. Each identified person was assigned a probability score, upon which relation (2) was applied. This approach approximates the number of people present in the stations.

$$n_{pers} = \sum_{i=1}^{m} \partial(c_i), \tag{2}$$

where:

n_{pers} is the factor representing the number of identified objects as a person;

m—the total number of objects identified;

c_i—the probability that the identified object is a person;

∂—the defined function as follows:

$$\partial(c_i) = \begin{cases} 1, \ if\ c_i \geq 0.5 \\ 0, \ if\ c_i < 0.5' \end{cases} \tag{3}$$

The following illustrates two examples of Azure service usage with images. In Figure 1, one person is waiting at a station. Since the entire body is visible and the light exposure is favorable in the picture, one person is identified after applying relation (2).

Figure 2 illustrates a scenario involving seven individuals. Of these seven individuals, only five are correctly identified by the Computer Vision service, as two blend into the background because of low lighting conditions.

To assess the accuracy of the person detection service using Azure Computer Vision, tests were conducted on 100 real-world images captured from various bus stations and different angles. The goal was to evaluate the service's confidence level in identifying individuals within different scenarios.

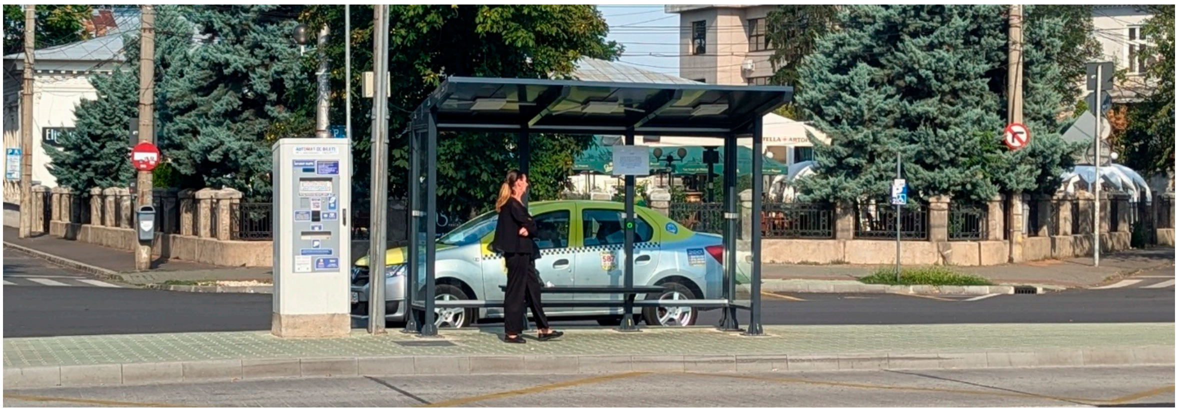


Figure 1. Correct identification scenario.

Figure 2. False identification scenario.

2.3. Person Identification Accuracy with Azure Computer Vision

Azure Computer Vision is a Microsoft cloud service for image recognition tasks. The service has pre-trained models to recognize objects, including people, and can classify if an object is a person associated with the recognition trusted factor. The programmer can adjust the trusted factor according to the developed application.

Computer vision may face challenges in accurately identifying people in specific scenarios. For instance, in crowded environments where individuals may overlap or obstruct each other, the model might struggle to distinguish between multiple individuals. Similarly, poor lighting conditions or blurry images can also impact the model's accuracy.

Another indicator proposed to evaluate the system's performance in correctly identifying the desired image features is the global accuracy rate (*GAR*). We computed the accuracy rate as the aggregate percentage of correct recognition relative to the total number of tests, as seen in relation (4).

The *GAR* is a measure of the precision of a recognition system, expressed as a percentage. It indicates the proportion of tests or images in which the system successfully identifies the desired objects or features from the tests conducted.

$$GAR = \frac{\sum_{i=1}^{n} LAR_i}{n} [\%], \quad (4)$$

where:

GAR represents the global accuracy rate;

n—total number of tests;

LAR_i—local accuracy rate for test i.

The LAR is the percentage of the identified persons related to the total number of persons in the image. It is computed with relation (5):

$$LAR_i = \frac{identified_i}{total_i} \times 100[\%], \quad (5)$$

LAR_i—local accuracy rate for test i;

$identified_i$—the number of persons identified with the Microsoft Azure Custom Vision in the test i;

$total_i$—the real number of persons existing in test i.

2.4. Proposed PUTSS Algorithm

Public urban transportation operates on a standard schedule at equal time intervals. The challenge is to devise an algorithm that adjusts these intervals to ensure the following:

- Public urban transport vehicles do not run empty;
- Passengers do not wait excessively long at stations.

To meet these constraints, the PUTSS algorithm was developed based on imposing a maximum waiting time of 15 min. Assuming a route provides departures every 5 min, the issue of skipping departures arises if there are fewer passengers on the road (*RTCVCF*) compared with a statistical value (*SDCF*).

The PUTSS algorithm utilizes the calculation formula from relation (6), where *SDCF* is less than 2.5 and the RTCVC factor (*RTCVCF*) exceeds the total bus capacity minus 5, then the bus can depart. If the *SDCF* exceeds 2.5, a safety margin of 5 seats is maintained, ensuring passenger comfort and preventing situations where additional passengers may arrive after the bus departs from the terminal.

$$Decision = \begin{cases} -1,\ SDCF < 2.5\ \ and\ RTCVCF < (k-5) \\ 0,\ (SDCF < 2.5\ and\ RTCVCF > (k-5)) \\ or\ (SDCF\ \geq 2.5\ \ and\ RTCVCF < (k-5)) \\ 1,\ SDCF\ \geq\ 2.5\ \ and\ RTCVCF \geq (k-5) \end{cases}, \quad (6)$$

where:

Decision is a flag variable that can take on the following three values:

- A value of -1 if a scheduled run can be skipped;
- A value of 0 if the bus should depart on its scheduled run;
- A value of 1 if an additional run should be added because of high identified congestion levels.

SDCF—the Statistical Data Component factor;

RTCVCF—the Real-Time Computer Vision Component factor;

k—the capacity of the bus scheduled for departure on its run, measured in the maximum number of passengers.

Figure 3 depicts the block diagram for the PUTSS algorithm, highlighting its components, decision points, and overall functionality. The first step involves utilizing Google Maps Service to identify public urban transport stations along the analyzed route. This identification provides accurate data regarding congestion levels at each station. This component ensures that the algorithm clearly understands the geographic layout of the transport system.

The second component is responsible for scraping congestion data from Google Maps Service for each hour of the day. Selenium automates the browsing process, allowing systematic data collection. The data retrieved provides a real-time view of congestion levels, which is necessary for the *SDCF* calculation and subsequent decision-making in the algorithm.

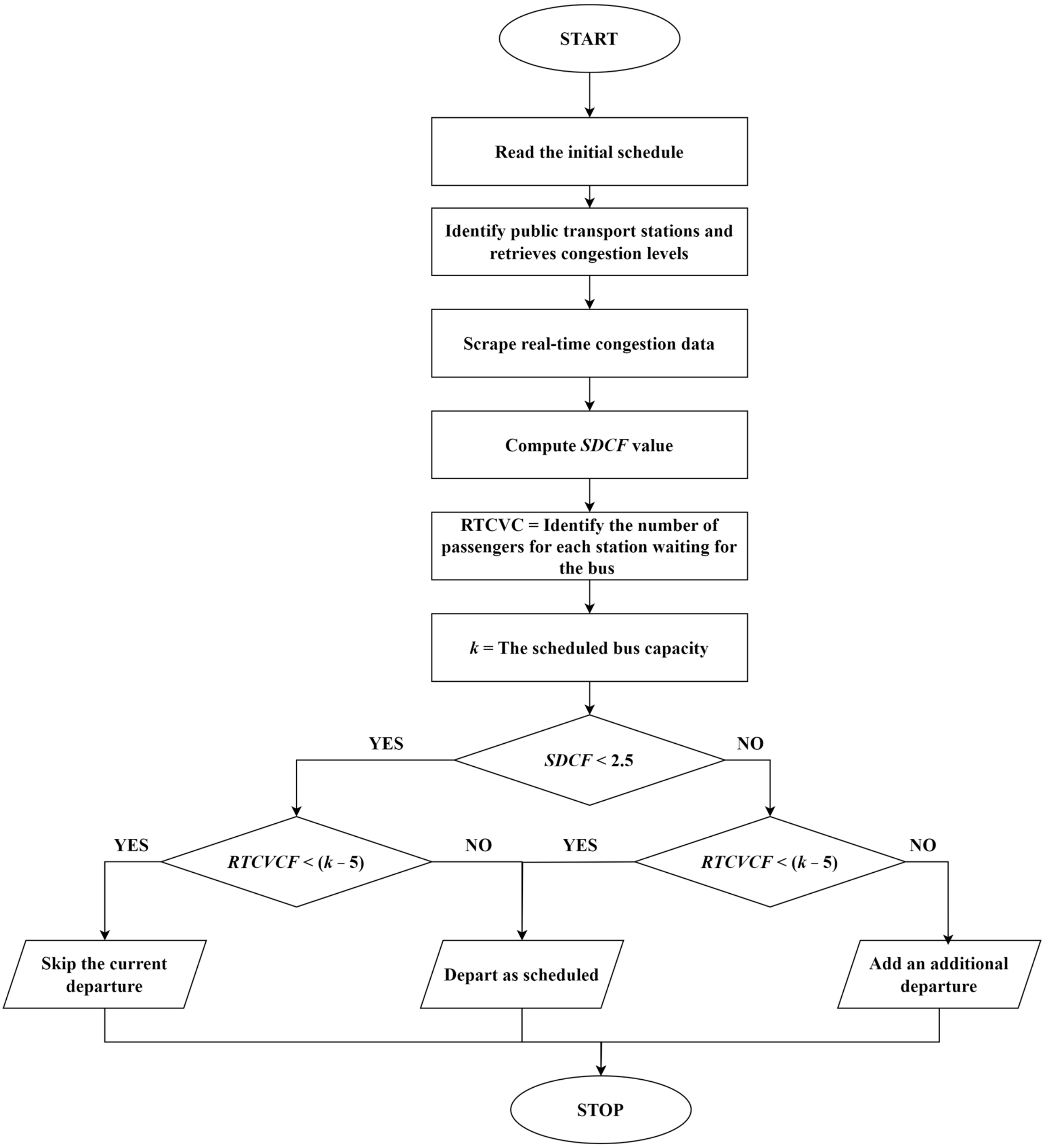


Figure 3. Block diagram of the PUTSS algorithm.

The RTCVC employs Azure Computer Vision to identify the number of station passengers based on real-time images. This module provides real-time passenger data, allowing for a more informed assessment of whether to adjust bus schedules based on actual occupancy levels. This decision-making process dynamically adjusts the bus schedule based on real-time data, which helps to minimize overcrowding and avoid running empty buses. This final component translates the decisions made by the PUTSS algorithm into actionable outputs, such as adjusting the bus schedule.

The Public Urban Transport Scheduling System (PUTSS) algorithm operates through a series of steps that combine real-time data and historical patterns to optimize public transportation schedules. Below, we outline the core processes involved that support the decision-making framework.

1. The algorithm receives two primary inputs including historical congestion data (from the Statistical Data Component, SDC) and real-time visual information about passengers at bus stops (from the Real-Time Computer Vision Component, RTCVC). Google Maps provides real-time traffic data, while Microsoft Azure's Computer Vision service identifies the number of passengers at bus stops in real time. These two components feed data into the system continuously;
2. The SDC component relies on historical data to predict future congestion levels. For a given station i at time t, the congestion level is classified into one of six categories as follows: light, moderate, heavier-than-usual, heavy, peak hours, or standstill;
3. The RTCVC component processes the visual data to count passengers at a given station i. The $RTCVCF$ is calculated based on the number of passengers detected and the bus capacity k. If the number of detected passengers exceeds the capacity, the system triggers an alert for additional resources;
4. The decision to add, cancel, or modify a bus departure is based on the combined factors from the $SDCF$ and $RTCVCF$. The dynamic nature of this decision-making process allows the system to respond flexibly to fluctuations in real-time demand;
5. The algorithm's core optimization goal is to prove the possibility of minimizing passenger waiting time and transportation service operational costs;
6. The final schedule adjustments are made based on a pre-defined maximum waiting time of 15 min. If the calculated decision output indicates that passenger demand exceeds the bus capacity at any station, the system adjusts the schedule dynamically, adding or canceling trips accordingly. In the Result Section, these adjustments showcase the algorithm's decision-making for a sample route.

This PUTSS algorithm was implemented in C#. A simulator was also developed in C# to facilitate multiple testing scenarios, enabling it to retrieve various pictures from a folder of images. Through numerous runs, the simulator fetched different sets of photographs and buses with varying capacities, generating diverse situations that map onto different scenarios of the algorithm. This approach ensured stable testing and validation across a spectrum of conditions, with high adaptability in real-world applications.

The PUTSS algorithm prevents congestion in urban transportation systems by leveraging historical data and real-time contextual information. The goal is to provide accurate congestion predictions and actionable insights to improve the management of public transit systems, optimize schedules, and enhance the overall commuter experience.

3. Results

This section assesses the effectiveness of the components utilized by the proposed algorithm through a detailed case study. Thus, the SDC and RTCVC integration is outlined, highlighting their roles in predicting and managing congestion levels on public urban transport routes. Subsequently, a case study that evaluates the accuracy of the Microsoft Azure Computer Vision service is presented, providing insights into its implications for the algorithm's overall efficiency.

The SDC component monitors the congestion levels along the route and weather conditions, while the RTCVC component is responsible for the actual congestion at the bus stops. Considering that ordinary persons cannot install cameras in public places, including bus stations, for capturing images or videos because of privacy rights, data protection laws, surveillance regulations, security and safety concerns, and liability issues, a simulator developed in C# was used, which dynamically processed various images captured by the authors. If the PUTSS algorithm is implemented by a public urban transport company, it will have access to the traffic surveillance cameras that are already placed in any big city that faces traffic congestion or buses' onboard cameras.

While the current case study is based on simulated data generated through a C# simulator, the following challenges must be addressed to ensure the successful implementation of PUTSS with real urban transportation data:

- Integrating real-time GPS data from buses and traffic monitoring systems into the PUTSS algorithm may involve dealing with inconsistent data formats, communication delays, and network outages;
- In real-world scenarios, the precision of data collected through camera sensors for detecting passenger counts might vary because of environmental factors such as poor lighting, weather conditions, or obstructions;
- The system's performance needs to be tested under varying traffic conditions and with different fleet sizes to ensure that the PUTSS algorithm can handle peak urban traffic efficiently without increasing operational costs.

To bridge the gap between simulation and real-world implementation, we must collaborate with a local urban transportation authority to implement PUTSS on a small subset of bus routes. This pilot project would collect real-time data from traffic sensors, buses, and bus stops. However, convincing local authorities to adopt and integrate the system could be challenging because of regulatory hurdles, budget constraints, and the complexity of retrofitting existing infrastructure. Additionally, ensuring the consistent availability and reliability of the necessary data sources could be difficult, as urban environments often present unpredictable variables that may hinder the seamless operation of the system.

Based on historical and real-time data, the SDC is applied to predict congestion levels at stations or along a route. This component helps estimate congestion trends using past data (like average crowding at different times or days) combined with current conditions (e.g., real-time traffic data, weather conditions, or nearby events).

The application of the SDC is as follows:

1. Data Collection: The system collects historical data about crowd levels at each station. This includes the following:
   - Time of day and day of the week.
   - Special events (e.g., sports games, concerts).
   - Weather conditions, traffic disruptions, etc.
2. Classification: The algorithm classifies each station's congestion status based on historical patterns. Congestion statuses include the following six stages: light, moderate, heavier-than-usual, heavy, peak hours, and standstill.
3. Average Congestion Factor Calculation: The algorithm calculates an average congestion factor for the entire route or a section using relation (1).

The SDC component relies on statistical data to predict the level of station congestion based on historical data and real-time information. The algorithm uses data such as time of day, day of the week, and special events.

We analyzed a bus route with six stations using historical and real-time data from Google Maps Service. The congestion statuses for each station are in Table 2.

Table 2. The SDC component example.

Station No	Congestion Status	Congestion Factor
Station 1	Light	1
Station 2	Moderate	2
Station 3	Heavier-than-usual	3
Station 4	Moderate	2
Station 5	Heavy	4
Station 6	Light	1

Of the six bus stations, two exhibit "light congestion", two show "moderate congestion", one experiences "heavier-than-usual congestion", and one encounters "heavy congestion". Hence, the *SDCF* computes to 2.16, which denotes the comprehensive congestion level across the route. This factor offers a statistical overview regarding congestion levels relative to typical public usage patterns of the respective transportation modes, contextualized by the time of day.

Table 3 presents all 100 tests conducted, showing the number of identified versus actual individuals in the images and the *LAR*. The trusted factor was set to 0.5 for the tests presented in Table 3.

Table 3. Comparison of identified versus actual individuals in images.

Test No.	Result	*LAR*	Test No.	Result	*LAR*	Test No.	Result	*LAR*	Test No.	Result	*LAR*
1.	3/5	60%	26.	4/4	100%	51.	4/5	80%	76.	1/1	100%
2.	2/4	50%	27.	4/4	100%	52.	1/1	100%	77.	7/10	70%
3.	5/5	100%	28.	4/4	100%	53.	1/1	100%	78.	10/10	100%
4.	6/7	85%	29.	4/4	100%	54.	2/2	100%	79.	3/5	60%
5.	6/7	85%	30.	5/5	100%	55.	1/1	100%	80.	4/5	80%
6.	6/7	85%	31.	4/4	100%	56.	1/1	100%	81.	1/1	100%
7.	4/5	80%	32.	2/4	50%	57.	1/1	100%	82.	2/2	100%
8.	3/3	100%	33.	5/5	100%	58.	3/3	100%	83.	1/1	100%
9.	3/3	100%	34.	5/5	100%	59.	3/3	100%	84.	1/1	100%
10.	2/2	100%	35.	4/5	80%	60.	5/5	100%	85.	6/6	100%
11.	2/2	100%	36.	2/5	40%	61.	5/5	100%	86.	3/3	100%
12.	2/2	100%	37.	7/7	100%	62.	4/5	80%	87.	4/4	100%
13	7/7	100%	38.	5/7	71%	63.	5/5	100%	88.	5/5	100%
14.	5/5	100%	39.	6/6	100%	64.	4/5	100%	89.	5/5	100%
15.	7/7	100%	40.	3/3	100%	65.	4/5	100%	90.	5/5	100%
16.	8/8	100%	41.	2/3	66%	66.	2/5	40%	91.	4/4	100%
17.	7/7	100%	42.	1/3	33%	67.	6/6	100%	92.	5/5	100%
18.	4/4	100%	43.	2/2	100%	68.	3/3	100%	93.	5/6	83%
19.	5/5	100%	44.	3/3	100%	69.	2/6	33%	94.	4/4	100%
20.	77	100%	45.	2/2	100%	70.	2/5	40%	95.	2/2	100%
21.	2/2	100%	46.	8/8	100%	71.	2/5	40%	96.	5/5	100%
22.	4/4	100%	47.	3/3	100%	72.	3/3	100%	97.	4/5	80%
23.	4/4	100%	48.	3/3	100%	73.	1/1	100%	98.	9/10	90%
24.	3/3	100%	49.	1/5	20%	74.	2/2	100%	99.	4/4	100%
25.	4/4	100%	50.	1/5	20%	75.	2/2	100%	100.	2/2	100%

For instance, the first test indicates that the Computer Vision service correctly recognized three persons out of five. Therefore, the *LAR* value is 60% (LAR_1 = 60%). This way, the recognition percentage was calculated for each test. Finally, by summing all the percentages and dividing by the total number of tests (100), the overall recognition rate (*GAR*) was obtained. For the tests presented in Table 3, the *GAR* is 89.81%. This value indicates a general algorithm's precision in identifying individuals in various image scenarios and conditions.

The Computer Vision service correctly recognizes the number of people in images with an accuracy of 73%. This value was calculated as the number of images where it correctly identified the number of people divided by the total number of images used in the tests. Therefore, the service correctly recognized the number of people in 73 images out of a total of 100 images.

Next, we analyzed the possible combinations of scenarios for the PUTSS algorithm and explanations for each scenario.

Scenario 1. Normal Traffic Conditions

SDCF: 2 (moderate congestion).

RTCVCF: 20 (20 passengers at the station).

Bus Capacity (k): up to 50 passengers.

Decision Output: 0 (bus departs as scheduled).

In this scenario, the *SDCF* indicates moderate congestion, while the *RTCVCF* shows an expected number of passengers at the station. Since the number of passengers is below the bus capacity, the bus departs as scheduled, maximizing transport efficiency without making passengers wait excessively.

Scenario 2. Additional Bus Needed
SDCF: 3 (heavier-than-usual congestion).
RTCVCF: 30 (30 passengers at the station).
Bus Capacity (k): up to 50 passengers.
Decision Output: 1 (an additional bus is added).

The *SDCF* reflects heavier congestion, indicating a need for increased transport capacity. The *RTCVCF* of 30 suggests that a significant number of passengers waiting. An additional bus is scheduled to accommodate the demand, ensuring the passengers do not waiting too long.

Scenario 3. Skipping a Scheduled Bus
SDCF: 1 (light congestion).
RTCVCF: 10 (10 passengers at the station).
Bus Capacity (k): up to 50 passengers.
Decision Output: -1 (scheduled bus is skipped).

In this case, the *SDCF* indicates light congestion, and the *RTCVCF* shows only 10 passengers. Since the number of passengers is far below the bus capacity, the decision is made to skip the scheduled bus to avoid running an empty service, thus optimizing operational costs.

Scenario 4. High Congestion, Few Passengers
SDCF: 4 (heavy congestion).
RTCVCF: 20 (20 passengers at the station).
Bus Capacity (k): up to 50 passengers.
Decision Output: 0 (bus departs as scheduled).

Despite the heavy congestion indicated by the *SDCF*, the number of passengers is still manageable. The bus can depart on schedule to prevent passengers from experiencing delays due to traffic conditions, even if it is not fully occupied.

Scenario 5. Very High Congestion, Full Capacity
SDCF: 5 (peak hours).
RTCVCF: 50 (50 passengers at the station).
Bus Capacity (k): up to 50 passengers.
Decision Output: 1 (an additional bus is added).

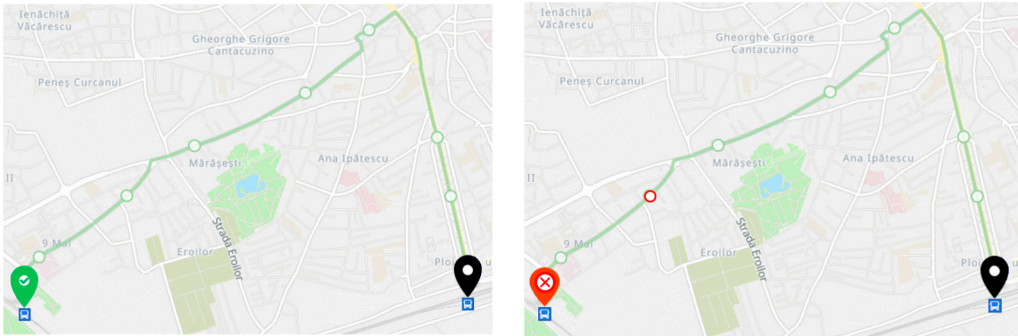
In this scenario, the congestion level and the number of waiting passengers are at their peaks. An additional bus is scheduled to address the overwhelming demand and ensure all passengers can board without significant delays.

Each of these scenarios illustrates the dynamic decision-making capabilities of the PUTSS algorithm in real-time urban transportation management. By analyzing congestion levels (*SDCF*) and passenger counts (*RTCVCF*), the algorithm can adjust bus schedules, optimizing service for passengers while minimizing operational costs for the transport provider.

Table 4 shows the programmed scheduled departure times of a bus, along with the applied scheduled changes by the *SDCF* and *RTCVCF*. The sixth column of Table 4 identifies the actions taken regarding the bus service, such as the bus departing as scheduled or the bus being canceled (Figure 4).

Table 4. Bus departure scheduling and capacity decision analysis.

Scheduled Departure	k	*SDCF*	*RTCVCF*	Decision	Action Taken
08:00	40	3	15	0	No changes; the bus departs as scheduled
08:15	40	5	36	-1	The programmed and the next scheduled bus go
08:30	40	2	10	0	The next bus is canceled
08:45	40	6	36	1	The programmed and the next scheduled bus go
09:00	40	2	20	0	The next bus is canceled
09:15	40	3	10	0	No changes; the bus departs as scheduled
09:30	40	5	30	-1	No changes; the bus departs as scheduled



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

Figure 4. Actions taken: (**a**) the bus departs as scheduled and (**b**) the next bus is canceled.

Table 4 outlines a bus schedule operating at 15 min intervals, detailing each departure's status regarding bus capacity and congestion factors.

The proposed algorithm is a demonstrative model for potentially integrating AI techniques to optimize public urban transportation fleet allocation. Municipalities can implement and improve it regardless of their size and bus fleet so that public transportation systems become more adaptive and responsive to real-time needs, reduce costs, and enhance sustainability, contributing to a greener and more environmentally friendly future.

4. Discussion

The case study shows that the Microsoft Azure Computer Vision service achieved recognition accuracy of only 73% in identifying the correct number of passengers in 100 images, resulting in a global accuracy rate of 89.81%. This indicates significant room for improvement in real-time passenger counting.

4.1. The PUTSS Algorithm Scenarios

The results of the five scenarios show that the PUTSS algorithm has dynamic decision-making capabilities in real-time urban transportation management. Furthermore, the algorithm can adjust bus schedules by optimizing service for passengers while minimizing operational costs for the transport provider.

The benefits of employing the PUTSS algorithm are as follows:

1. Fewer empty buses on the road. One of the leading environmental issues with traditional bus systems is the inefficiency of buses running on fixed schedules, even when there is little or no demand. This results in buses traveling with few or no passengers, wasting fuel and resources. The PUTSS algorithm adapts to real-time demand, ensuring that buses run only when and where needed. This reduces the number of empty or underutilized buses on the road, directly cutting down fuel consumption.
2. Lower CO_2 emissions. By optimizing the schedule, the PUTSS algorithm reduces the time buses spend idling in traffic or running off schedule. Efficient routes mean less fuel is burned per trip, which translates to lower CO_2 emissions, an essential requirement in smart cities. Given that, nowadays, in most cities, buses are often diesel-powered (and their electrification is just beginning), which produces higher levels of pollution than gasoline, even small reductions in bus trips or fuel usage can have a significant impact on overall emissions.
3. Less traffic congestion: When buses run more efficiently, with fewer vehicles on the road at unnecessary times, there is a reduction in overall traffic congestion. This, in turn, means that other vehicles (cars, trucks, etc.) can move more smoothly, leading to less stop-and-go traffic. Vehicles burn more fuel when constantly stopping and starting, so reducing congestion by the PUTSS algorithm helps improve fuel efficiency and lowers emissions across the entire transportation network in a smart city.
4. Encouraging public transportation use: A more reliable and optimized bus schedule by the PUTSS algorithm improves the overall experience for passengers, making

public transportation a more attractive option. As more people choose buses over personal cars, the number of individual vehicles on the road decreases. This shift from private vehicles to public transport can significantly reduce the total emissions of a city, as fewer cars are needed to move the same number of people.

5. Supporting sustainable urban growth. Smart cities need to optimize public transportation systems to grow sustainably. Reduced emissions and better resource use contribute to cleaner air quality and lower noise pollution, enhancing the overall environmental health of urban areas.

A drawback of the PUTSS algorithm lies in the empirically set reserve value of five for the passenger margin reserve, which implies that the algorithm may not adjust dynamically to minor passenger fluctuations, potentially leading to underutilization or overburdening of buses. This static approach could limit responding to real-time changes in passenger demand across different times of the day or in response to unforeseen events like weather changes or special events.

Another identified limitation is when a trip is skipped based on the algorithm's decision. Buses often wait unnecessarily at the terminal until their next scheduled departure time. This waiting time incurs operational costs and may lead to inefficiencies in resource utilization, as buses could have been deployed more on alternative routes or schedules.

4.2. Handling Uncertainty in the PUTSS Algorithm

The PUTSS algorithm addresses uncertainty by incorporating real-time data from multiple sources, such as Google Maps and Microsoft Azure Computer Vision, to adjust bus schedules dynamically. External factors like weather (rain or snow) are integrated through real-time traffic updates, allowing the system to make informed decisions about bus departure times and additional fleet requirements.

While traditional statistical approaches to predicting congestion can be practical in stable conditions, they often fail to account for unpredictable external factors such as sudden weather changes, public events, or accidents. These factors can significantly impact congestion levels, making it difficult for classical methods to provide accurate real-time predictions. In contrast, AI-based algorithms, such as the one used in the PUTSS model, offer greater flexibility by incorporating real-time data and dynamically adjusting to changing conditions. This makes them more suited for handling complex and fluid urban transportation systems. Moreover, by combining historical data (*SDC*) and real-time computer vision (*RTCVC*), the proposed model addresses these challenges more effectively, ensuring that public transportation services remain responsive.

Fuzzy logic is widely recognized as a robust framework for managing uncertainty in path planning, particularly in scenarios with ambiguous constraints like adverse weather conditions. The PUTSS algorithm uses a more deterministic approach. By relying on real-time data and statistical models, the algorithm directly responds to observable conditions rather than fuzzy memberships, which could be a more appropriate approach in situations with precise, quantifiable data.

However, it is acknowledged that fuzzy-based approaches might offer additional flexibility in scenarios where uncertainty is high or data are more imprecise. Future research could compare the performance of the PUTSS algorithm against fuzzy logic systems, particularly in extreme weather conditions, to evaluate which framework better handles dynamic and uncertain environments in urban transport systems.

By refining these aspects, the PUTSS algorithm could achieve higher operational use in practice, improving urban mobility and public transit management.

Future work should incorporate additional data sources, such as weather forecasts, event schedules, and social media trends, to enhance the predictive accuracy of the *SDCF* and the responsiveness of the *RTCVCF*. This multi-faceted data integration could help better anticipate and manage sudden surges in passenger numbers.

Secondly, future research should investigate methods for dynamically adjusting the empirically set parameters, such as the passenger number margin. ML techniques could optimize these parameters based on real-time data and changing conditions.

Although Google Maps Service provides real-time traffic and congestion data, it relies on another model's estimations rather than raw data directly collected from buses. This approach introduces potential inaccuracies, as Google Maps' data are derived from a combination of user data and various prediction models, which may not always reflect the exact traffic conditions relevant to a specific public transport system.

In an ideal scenario, buses could be equipped with sensors to deliver raw, real-time data about traffic and passenger congestion levels. Such sensors would offer more accurate and reliable insights, allowing for precise scheduling and congestion management adjustments. However, the cost and complexity of implementing this infrastructure might not be feasible for many public transportation systems.

4.3. Future Work

Future work could explore integrating sensor-based systems in buses, potentially using IoT (Internet of Things) technology to directly track bus movements, passenger counts, and environmental conditions. This would offer a more robust solution for real-time data collection, improving the accuracy of congestion predictions and enhancing the overall efficiency of the PUTSS algorithm.

We plan to extend the PUTSS algorithm to address scalability by optimizing its computational efficiency and incorporating distributed computing. This will allow it to handle larger, more complex urban environments with numerous variables and stations.

Regarding privacy concerns, the system complies with data protection regulations like GDPR by anonymizing all collected data and limiting the use of real-time images for traffic analysis without storing or identifying passengers' personal information.

To improve passenger detection accuracy, we plan to explore alternative computer vision models and complementary techniques, such as advanced deep learning (DL) algorithms that perform better in challenging conditions like low lighting or high congestion. The historical congestion data (*SDC*) used in our algorithm is sourced from public transit databases and traffic management systems; however, we acknowledge the need for further validation and are considering integrating additional real-time data sources to enhance prediction accuracy.

This article considers reducing CO_2 emissions by decreasing the number of vehicles in traffic. However, it does not focus on conducting a complete environmental impact study, which falls outside the scope of this research.

Moreover, forthcoming work will consider researching alternative technologies for real-time monitoring and congestion management, such as IoT sensors, wearable technology, or mobile app integration, to complement the capabilities of the current algorithm.

5. Conclusions

This article proposes a new algorithm to optimize public urban transportation and encourage its use among the population. It enhances the efficiency of public urban transport systems by integrating advanced technological components, such as statistical data analysis and real-time computer vision services.

The PUTSS algorithm integrates the following components to identify congestion levels on public transport routes: a Statistical Data Component (SDC) and a Real-Time Computer Vision Component (RTCVC).

The SDC leverages historical and real-time statistical data to predict passenger numbers at various stations, allowing for preemptive adjustments in fleet frequency to manage expected congestion.

The RTCVC employs Microsoft Azure Computer Vision services to monitor and count passengers at each station in real time. This component enables the system to address sudden changes in passenger numbers, such as those caused by heavy rain, concerts, or other unexpected events.

This article includes a case study evaluating the accuracy of the Microsoft Azure Computer Vision service. Analyzing 100 images, the recognition accuracy was 73%, correctly identifying the number of people in only 73 out of 100 tests. The global accuracy rate was computed at 89.81%.

Using a proposed mathematical tool, the PUTSS algorithm computes the congestion factor across the route, assigning numeric values based on traffic status from Google Maps Service. The algorithm adjusts public urban transport schedules to avoid empty runs and excessive passenger wait times by setting a maximum waiting time of 15 min. It makes decisions based on the *SDCF* and *RTCVCF* values, ensuring that buses depart when they meet specific congestion criteria.

The five scenarios underline the dynamic decision-making capabilities of the PUTSS algorithm in real-time urban transportation management. The PUTSS algorithm analyzes the congestion levels and passenger counts to adjust bus schedules, optimizing passenger service while minimizing operational costs for the transport provider.

Implemented in C#, the algorithm was tested using a simulator that generates various scenarios. However, the empirically set reserve value of five for the passenger number margin and the potential inefficiency of buses waiting at terminals when trips are skipped highlight areas for refinement.

Author Contributions: Conceptualization, C.-M.R.; methodology, C.-M.R. and A.S.; software, C.-M.R.; validation, C.-M.R. and A.S.; formal analysis, C.-M.R. and A.S.; investigation, C.-F.N. and I.-A.G.; resources, C.-F.N. and I.-A.G.; data curation, C.-M.R.; writing—original draft preparation, A.S., C.-F.N. and I.-A.G.; writing—review and editing, A.S., C.-F.N. and I.-A.G.; visualization, C.-M.R. and A.S.; supervision, C.-M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Petroleum-Gas University of Ploiesti, Romania.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kita, H.; Komoda, S.; Ozaki, R. A Quantified Planning Method of Local Public Transport Services for Expanding Residents' Activity Opportunities. *Transp. Policy* **2024**, *in press*. [CrossRef]
2. Khademi-Vidra, A.; Nemecz, G.; Mária Bakos, I. Satisfaction measurement in the sustainable public transport of Budapest. *Transp. Res. Interdiscip. Perspect.* **2024**, *23*, 100989. [CrossRef]
3. Nichols, A.; Ryan, J.; Palmqvist, C.-W. The importance of recurring public transport delays for accessibility and mode choice. *J. Transp. Geogr.* **2024**, *115*, 103796. [CrossRef]
4. Wong, H.; Yap, M. A data driven approach to update public transport service elasticities. *J. Public Transp.* **2023**, *25*, 100066. [CrossRef]
5. Luan, X.; Eikenbroek, O.A.L.; Corman, F.; Van Berkum, E.C. Passenger social rerouting strategies in capacitated public transport systems. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *188*, 103598. [CrossRef]
6. Sharma, P.; Herminghaus, S.; Heuer, H.; Heidemann, K.M. Impact of the density of line service stations on overall performance in Bi-modal public transport settings. *Multimodal Transp.* **2024**, *3*, 100118. [CrossRef]
7. García-Mauriño, C.; Zufiria, P.J.; Jarabo-Peñas, A. Improving bus arrival time predictors using only public transport API data. *Transp. Lett.* **2023**, *16*, 804–813. [CrossRef]
8. Wang, Z.; Huang, K.; Massobrio, R.; Bombelli, A.; Cats, O. Quantification and comparison of hierarchy in Public Transport Networks. *Phys. A Stat. Mech. Its Appl.* **2024**, *634*, 129479. [CrossRef]
9. Bajcetic, S.A.; Zivanovic, P.V.; Tica, S.M.; Petrovic, M.M.; Dorojevic, A.M.; Milovanovic, B.M. Implementation of the new public transport management system in Belgrade. In Proceedings of the 11th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS), Nis, Serbia, 16–19 October 2013; pp. 643–646. [CrossRef]

10. Yakimov, M.R. The Use of Various Input Data in the Design of Public Transport Route Networks. In Proceedings of the Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russia, 19–20 March 2020; pp. 1–5. [CrossRef]
11. Horazdovsky, P.; Svitek, M. Implementation of Smart Dynamic Public Transport into a Real Operation. In Proceedings of the Smart City Symposium Prague (SCSP), Prague, Czech Republic, 27–28 May 2021; pp. 1–6. [CrossRef]
12. Gheorghiu, R.A.; Iordache, V.; Stan, V.A. Computer vision application to determine crowdedness in public transport stations. In Proceedings of the 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 1–3 July 2021; pp. 1–4. [CrossRef]
13. Kasatkina, E.; Vavilova, D.; Ketova, K. Optimization of the Public Transport System Using Data Analysis Methods. In Proceedings of the 4th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russian, 9–11 November 2022; pp. 174–177. [CrossRef]
14. Saddal, M.; Rasheed, A.; Qureshi, I.-U.-H.; Irfan, M.; Rehman, A.; Abbasi, R.A.; Sabir, I.; Irfan, H.; Zaman, M. Public Transport Network Data; A Case-Study of Islamabad. In Proceedings of the 18th International Conference on Emerging Technologies (ICET), Peshawar, Pakistan, 6–7 November 2023; pp. 269–274. [CrossRef]
15. Douskova, N.; Novotny, V.; Richter, P.; Jirmanova, M. Pilot Study of Using Google Traffic Data to Assess Conditions for Public Transport Operation on Urban Roads. In Proceedings of the Smart City Symposium Prague (SCSP), Prague, Czech Republic, 26–27 May 2022; pp. 1–5. [CrossRef]
16. Guo, Y.; Li, H.; Li, X.; Gu, L.; Zhang, J. System Observability Analysis Based on Multiple Source Mixed Traffic Sensors. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 7516–7520. [CrossRef]
17. Elmrini, A.; Amrani, A.G. Wireless Sensors Network for Traffic Surveillance and Management in Smart Cities. In Proceedings of the 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), Marrakech, Morocco, 21–27 October 2018; pp. 562–566. [CrossRef]
18. Jha, A.; Agrawal, P.; Kaur, R.; Agrawal, A.; Bhattacharya, M. IoT Based Smart Traffic Controller System. In Proceedings of the 2023 Second International Conference on Informatics (ICI), Noida, India, 23–25 November 2023; pp. 1–5. [CrossRef]
19. Syum Gebre, T.; Beni, L.; Tsehaye Wasehun, E.; Elikem Dorbu, F. AI-Integrated Traffic Information System: A Synergistic Approach of Physics Informed Neural Network and GPT-4 for Traffic Estimation and Real-Time Assistance. *IEEE Access* **2024**, *12*, 65869–65882. [CrossRef]
20. Klein, L.A. Roadside Sensors for Traffic Management. *IEEE Intell. Transp. Syst. Mag.* **2024**, *16*, 21–44. [CrossRef]
21. Saldivar-Carranza, E.D.; Bullock, D.M. Deriving Verified Vehicle Trajectories from LiDAR Sensor Data to Evaluate Traffic Signal Performance. *Future Transp.* **2024**, *4*, 765–779. [CrossRef]
22. Beti, I.-A.; Herghelegiu, P.-C.; Caruntu, C.-F. Architectural Framework to Enhance Image-Based Vehicle Positioning for Advanced Functionalities. *Information* **2024**, *15*, 323. [CrossRef]
23. Olugbade, S.; Ojo, S.; Imoize, A.L.; Isabona, J.; Alaba, M.O. A Review of Artificial Intelligence and Machine Learning for Incident Detectors in Road Transport Systems. *Math. Comput. Appl.* **2022**, *27*, 77. [CrossRef]
24. Kirwan, B. The Impact of Artificial Intelligence on Future Aviation Safety Culture. *Future Transp.* **2024**, *4*, 349–379. [CrossRef]
25. Basheer Ahmed, M.I.; Zaghdoud, R.; Ahmed, M.S.; Sendi, R.; Alsharif, S.; Alabdulkarim, J.; Albin Saad, B.A.; Alsabt, R.; Rahman, A.; Krishnasamy, G. A Real-Time Computer Vision Based Approach to Detection and Classification of Traffic Incidents. *Big Data Cogn. Comput.* **2023**, *7*, 22. [CrossRef]
26. Shepelev, V.; Glushkov, A.; Bedych, T.; Gluchshenko, T.; Almetova, Z. Predicting the Traffic Capacity of an Intersection Using Fuzzy Logic and Computer Vision. *Mathematics* **2021**, *9*, 2631. [CrossRef]
27. Jaiswal, M.; Gupta, N.; Rana, A. Real-time Traffic Management in Emergency using Artificial Intelligence. In Proceedings of the 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 4–5 June 2020; pp. 699–702. [CrossRef]
28. Wu, B. Research on a Traffic Signal Control Method Based on Deep Reinforcement Learning. In Proceedings of the 24 IEEE 4th International Conference on Power, Electronics and Computer Applications (ICPECA), Shenyang, China, 26–28 January 2024; pp. 53–57. [CrossRef]
29. Nechoska, D.K.; Nechkoska, R.P.; Duma, R. Application of Artificial Intelligence for Traffic Data Analysis, Simulations and Adaptation. In Proceedings of the 2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), Ohrid, North Macedonia, 16–18 June 2022; pp. 1–4. [CrossRef]
30. Sony, A.K.; Bhushan, B.; Astya, R. Revolutionizing Transportation System Using Artificial Intelligence Technique. In Proceedings of the 2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 3–4 November 2023; pp. 1171–1176. [CrossRef]
31. Makahleh, H.Y.; Ferranti, E.J.S.; Dissanayake, D. Assessing the Role of Autonomous Vehicles in Urban Areas: A Systematic Review of Literature. *Future Transp.* **2024**, *4*, 321–348. [CrossRef]
32. Hou, K.; Giannopoulos, G. Modeling the Deployment and Management of Large-Scale Autonomous Vehicle Circulation in Mixed Road Traffic Conditions Considering Virtual Track Theory. *Future Transp.* **2024**, *4*, 215–235. [CrossRef]
33. Aguilar-Ibanez, C.; Suarez-Castanon, M.S.; García-Canseco, E.; Rubio, J.D.J.; Barron-Fernandez, R.; Martinez, J.C. Trajectory Tracking Control of an Autonomous Vessel in the Presence of Unknown Dynamics and Disturbances. *Mathematics* **2024**, *12*, 2239. [CrossRef]

34. Guo, X.-Y.; Jia, A.-F. Study on traffic flow mixed with autonomous vehicles and human-driven vehicles. In Proceedings of the 2022 3rd International Conference on Intelligent Design (ICID), Xi'an, China, 21–23 October 2022; pp. 166–171. [CrossRef]
35. Zhou, W.; Liu, M.; Liu, T.; Wang, Y. Influence of Bus Bay on Heterogeneous Traffic Flow Consisting of Human Driving and Autonomous Vehicles. *IEEE Access* **2023**, *11*, 89455–89468. [CrossRef]
36. Sarath, P. Deep NN-Based Autonomous Vehicle Framework for Route Selection and Traffic Control Using Golden Levy Optimization. In Proceedings of the 2023 7th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 22–24 November 2023; pp. 517–522. [CrossRef]
37. Andreotti, E.; Selpi; Boyraz, P. Potential impact of autonomous vehicles in mixed traffic from simulation using real traffic flow. *J. Intell. Connect. Veh.* **2023**, *6*, 1–15. [CrossRef]
38. Gupta, N.; Saawan, S.; Kumar, S.; Khan, M.U. Enhancing Traffic Flow Efficiency and Mitigating Congestion through ML-Based Traffic Management Strategies. In Proceedings of the 2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT), Faridabad, India, 23–24 November 2023; pp. 737–743. [CrossRef]
39. Mao, X. The Mixed Impact of Autonomous Vehicles on Urban Traffic. In Proceedings of the 2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST), Guangzhou, China, 10–12 December 2021; pp. 1471–1476. [CrossRef]
40. Mondal, M.A.; Rehena, Z. Identifying Traffic Congestion Pattern using K-means Clustering Technique. In Proceedings of the 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 18–19 April 2019; pp. 1–5. [CrossRef]
41. Singh, S.; Singh, B.; Ramandeep; Singh, B.; Das, A. Automatic Vehicle Counting for IoT based Smart Traffic Management System for Indian urban Settings. In Proceedings of the 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 18–19 April 2019; pp. 1–6. [CrossRef]
42. Akhtar, M.; Raffeh, M.; Ul Zaman, F.; Ramzan, A.; Aslam, S.; Usman, F. Development of congestion level based dynamic traffic management system using IoT. In Proceedings of the 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Istanbul, Turkey, 12–13 June 2020; pp. 1–6. [CrossRef]
43. Gupta, A.K.; Johari, R. IOT based Electrical Device Surveillance and Control System. In Proceedings of the 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 18–19 April 2019; pp. 1–5. [CrossRef]
44. Amer, M.; Maher, R. PV/IoT based Smart Traffic Management System. In Proceedings of the 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Maldives, Maldives, 16–18 November 2022; pp. 1–6. [CrossRef]
45. Shobana, S.; Shakunthala, M.; Vimala, J.C.; Kingslin, M.T.; Sivarajan, S.; Lakshmi, K.R.C. IoT based on Smart Traffic Lights and Streetlight System. In Proceedings of the 2023 2nd International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, 19–21 July 2023; pp. 1311–1316. [CrossRef]
46. Usmonov, S.; Pradeep, A.; Fakhriddinov, Z.; Sanjar, T.; Abdurakhim, A.; Khusniddinova, M. Intelligent Traffic Management System: AI-Enabled IoT Traffic Lights to Mitigate Accidents and Minimize Environmental Pollution. In Proceedings of the 2023 3rd International Conference on Intelligent Technologies (CONIT), Hubli, India, 23–25 June 2023; pp. 1–6. [CrossRef]
47. Talukder, M.Z.; Towqir, S.S.; Remon, A.R.; Zaman, H.U. An IoT based automated traffic control system with real-time update capability. In Proceedings of the 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, India, 3–5 July 2017; pp. 1–6. [CrossRef]
48. Zhu, Y. Collaborative Intelligent Traffic Planning in the Internet of Vehicles. *Sensors* **2024**, *24*, 1303. [CrossRef]
49. Ullah, I.; Ali, F.; Khan, H.; Khan, F.; Bai, X. Ubiquitous computation in internet of vehicles for human-centric transport systems. *Comput. Hum. Behav.* **2024**, *161*, 108394. [CrossRef]