SCIENTIFIC OASIS

Decision Making: Applications in Management and Engineering

Journal homepage: www.dmame-journal.org
ISSN: 2560-6018, eISSN: 2620-0104

A Machine Learning-Based Decision Analytic Model for Optimal Route Selection in Autonomous Urban Delivery: The ULTIMO Project

Shervin Zakeri[1,*], Dimitri Konstantas[1], Prasenjit Chatterjee[2], Ali Vaez Jalali[3]

[1] Geneva School of Economics and Management, University of Geneva, 1211 Geneva, Switzerland
[2] Department of Mechanical Engineering, MCKV Institute of Engineering, West Bengal- 711204, India.
[3] Computer Engineering Department, Shomal University, Amol, Iran

ARTICLE INFO

Article history:
Received 13 February 2024
Received in revised form 25 May 2024
Accepted 20 June 2024
Available online 25 August 2024

Keywords:
Autonomous vehicles; Decision Support System; Multi-Criteria Decision-Making; ARWEN; WLD; Random Forest classifier.

ABSTRACT

In this research, we introduce a Decision Support System (DSS) that incorporates two multi-criteria decision-making (MCDM) techniques: the Alternatives Ranking with Elected Nominee (ARWEN) and the Win-Loss-Draw (WLD) methods. This system benefits from both methods' advantages to address the challenge of selecting optimal routes for autonomous urban deliveries. The primary objective of this study is to establish a comprehensive framework to assist decision-makers in selection of the optimized strategy. This paper presents not only the implementation codes but also validates the results and conducts a sensitivity analysis. Furthermore, the DSS is applied to a numerical example to demonstrate its practical utility in real-world scenarios.

1. Introduction

This paper focuses on proposing a new Decision Support System (DSS) for solving the problem of optimized route selection within the ULTIMO project. It first introduces the projects, then explores the theories and concepts upon which the proposed DSS is built, and finally delves into the architecture of the DSS, its applications, and the analysis of the obtained results. The introduction section itself is divided into eight sections to cover the aforementioned concepts, including the ULTIMO project, the concept of smart urban mobility, DSSs and their applications, multi-criteria decision-making (MCDM) problems and methods, machine learning, and the paper's motivation, contribution, and structure.

1.1. The ULTIMO project

The transport sector is currently at a pivotal point in introducing automated vehicles (AVs). Despite various projects testing AVs in public transport, the widespread implementation of profitable automated shared fleets is still pending. The ULTIMO project aims to integrate AVs into urban areas,

* *Corresponding author.*
E-mail address: Shervin.Zakeri@etu.unige.ch

https://doi.org/10.31181/dmame7220241114

offering sustainable, accessible, and inclusive on-demand and door-to-door services. Drawing from previous projects, ULTIMO focuses on overcoming economic and other barriers to the large-scale adoption of AVs.

ULTIMO plans to deploy AVs at three European sites, each with over 15 multi-vendor vehicles, operating fully automated and without safety drivers, enhanced by user-centric passenger services. The project's innovative transportation models are expected to have a lasting impact on automated transport in Europe and globally. The consortium's composition promotes interoperability among stakeholders, ensuring new technology adoption is cost-effective and safe. The project also integrates learnings from earlier AV demonstrations to maximize technical and societal benefits, both during and after its completion.

1.2. Smart urban mobility

The fundamental idea of road vehicle automation is defined within the concept of smart urban mobility. The concept of smart urban mobility, as initially framed in the context of the smart cities, is defined by the fusion of sustainable, advanced vehicular technologies and cooperative intelligent transport systems, facilitated through cloud servers and networks that utilize big data [1]. The concept of automated driving emerged simultaneously with the advent of the earliest automobiles.

From the moment the first cars traversed the streets, there has been a persistent dream to create vehicles capable of transporting us to our desired destinations autonomously, without the need for human control. This vision, as old as the automobile itself, reflects a long-standing aspiration for a future where the complexities and efforts of driving are replaced by intelligent, self-guiding machines [2]. Automated Driving Systems (ADSs) are currently in development, holding the promise of enhancing road safety by preventing accidents, lowering emissions, providing transportation for those with mobility impairments, and alleviating stress related to driving [3]. As described by Shladover [4], ADSs involve substituting human driving tasks, either partially or entirely, with electronic and mechanical systems. In recent years, AVs have seen remarkable advancements, thanks to significant efforts from both the research community and industry. This rapid development has garnered considerable interest among researchers [5,6], e.g., the following studies that covered the various aspects of AVs [7-15].

1.3. The decision-support systems

In smart urban mobility domain, specifically for applications like autonomous vehicle route optimization, the use of a DSS is an integral part, since it could efficiently handle complex decision-making by processing vast data sets, including traffic conditions, road quality, and safety ratings, to make well-informed decisions. A DSS's capability for real-time data analysis is highly important for adapting to dynamic urban environments, ensuring route optimization, selection of the most appropriate strategies, taking optimized actions, and operational efficiency. It adeptly balances multiple objectives, like reducing travel time while enhancing safety and minimizing emissions, aligning with broader smart city goals. Safety enhancement is a key benefit, as the DSS considers various risk factors to identify the safest routes. Its adaptability and learning abilities allow for continual improvement in decision-making. Furthermore, a DSS enhances efficiency in resource management and is scalable to meet the evolving needs of urban growth. It also integrates seamlessly with urban infrastructure and can be tailored to user preferences, ensuring a personalized experience. Importantly, it aids in ensuring compliance with local policies and regulatory standards, which is critical for the legal and social acceptance of autonomous vehicles. The role of decision-making and DSSs in smart urban mobility has been investigated by many scholars, such as works done by Caballero *et al*., [16], Gokasar *et al*., [17], Bonab *et al*., [18], Schwarting *et al*, [19], and Li *et al*., [20].

1.4. MCDM applications

Many studies, such as the aforementioned examples [17,18] and the studies conducted by Abdel-Basset *et al*., [21] and Gamal *et al*., [22], employed MCDM methods to architect their proposed DSS to assess the related options, ranking and selection, and aiding the decision-makers. MCDM problems are structured as matrix-based decision problems involving various criteria and options, aimed at attaining one or more objectives [23]. To solve MCDM problems, a process is needed that entails identifying the most suitable option from multiple alternatives, taking into account the viewpoints of decision-makers and all relevant criteria [24]. To address these challenges, a range of MCDM algorithms have been developed, each characterized by distinct underlying philosophies and guiding principles [25]. When a decision-making problem involves criteria, options, and goals, MCDM methods outshine other approaches due to several key advantages: 1. MCDM offers a clear, structured approach for assessing multiple criteria, essential for complex decisions with multiple stakeholders or conflicting goals.2. MCDM methods are typically more transparent than other algorithms, such as the machine learning (ML) algorithms, providing a step-by-step understanding of decision-making processes, crucial in scenarios requiring justification and accountability. 3. MCDM allows the direct inclusion of expert knowledge and subjective judgment in decision-making, especially in criteria weighting, making it vital for scenarios where expertise significantly influences outcomes. 4. MCDM methods are adaptable, allowing easy modification of criteria or weights in response to new information or changing priorities. 5. An MCDM method can operate effectively with limited or qualitative data, making it ideal in data-scarce situations. 6. MCDM is less prone to issues like overfitting and data bias, as it doesn’t rely on historical data patterns, thus reducing the risk of perpetuating past inaccuracies. 7. MCDM is specifically designed for evaluating and ranking options based on a set of criteria, effectively balancing various trade-offs in complex decision scenarios. 8. MCDM handles uncertainty well, whether from decision-maker subjectivity, missing information, or numeric intervals, critical in dynamic decision-making environments. These advantages make MCDM methods ideal for architecting an DSS for solving a decision-making problem.

1.5. The ML application

Recent advancements in artificial intelligence (AI) have naturally led to cohesive integration with AVs, fulfilling various requirements [26]. There is extensive literature on the various applications of AI in AVs. This article’s proposed DSS employs ML as one of the tools for computing the weights of criteria for optimized route selection in Geneva city. Similarly, ML has been employed in many studies as part of their proposed solutions for different aspects of AVs' implementation or integration such as the studies done by Filippou *et al.,* [27], Caesar *et al.,* [28], Qayyum *et al.,* [29], and Fu *et al.,* [30].

1.6. Motivation

The increasing deployment of autonomous delivery vehicles in urban environments introduces a significant challenge in route planning to ensure efficiency and safety. Current methods often lack in effectively balancing multiple decision-making criteria, such as travel time, energy efficiency, and safety considerations. This gap highlights the need for an advanced DSS that can comprehensively evaluate and rank potential routes using the MCDM approach.

In this paper, we aim to integrate two recently developed MCDM methods—the Alternatives Ranking with Elected Nominee (ARWEN) and the Win-Loss-Draw (WLD) methods—to create a robust DSS. ARWEN will be utilized as a ranking method to evaluate route options, while WLD will be employed as a subjective weighting approach for assessing the importance of various criteria. The proposed DSS is designed to identify the most efficient and safest routes for autonomous delivery vehicles operating in urban settings. By doing so, we intend to enhance route decision-making processes, contributing to improved operational reliability and safety of autonomous vehicular technology in complex urban landscapes.

1.7. Contribution of this study

Consistent with existing literature, where only a few studies have proposed DSS for addressing problems related to AVs, the decision-making processes within the ULTIMO project must be based on a systematic approach. Therefore, in this paper, we propose a new combined MCDM framework to aid in making operational decisions for the project. Another gap identified in the literature is the lack of a coded framework for DSS, making them not accessible for researchers and relevant industries. Hence, this paper also introduces software codes that are easy to implement and use.

1.8. The paper's structure

The structure of the paper is as follows: Section two provides a theoretical background on the DSS; section three outlines the proposed theoretical framework of the DSS; section four applies the DSS to a case study; section five focuses on validating the DSS's performance; section six presents the Python codes used for constructing the DSS; and finally, section seven offers conclusions and directions for future research.

2. Theoretical background

In this section, ARWEN and WLD methods are introduced.

2.1. ARWEN

Proposed by Zakeri *et al.,* [31], the ARWEN method is developed to identify the most suitable option based on the smallest rate of change, as opposed to selecting an elected nominee. The method has been further elaborated into four variants, each based on the type and extent of information that decision-makers have at their disposal. The ARWEN's algorithm's fundamental selection process is based on the larger value of Γ_i, as shown in Eq. (1):

$$\Gamma_i = (2n) - \left(\sum_{j=1}^{n} W_j \left(\max_i r_{ij} . \left(r_{ij}\right)^{-1} \right) \right), \qquad i = \{1, \dots, m\}, j = \{1, \dots, n\}, \tag{1}$$

where n and m stand for the number of criteria and alternatives, respectively.

2.2. WLD method

The WLD method is developed based on the assumption that decision-makers have complete information regarding the criteria [32]. This straightforward weighting approach assigns two distinct importance weights for criteria evaluation, mimicking human behavioral patterns in assessing these criteria. The process of the WLD method is outlined in the following steps.

Step 1. The first step is to evaluate the criteria in terms of their importance by employing a scale, in which the upper and lower bounds are 1,10, and the center is 5. Decision-makers can choose any rational numbers between the mentioned numbers.

Step 2. The second step is to establish the pairwise comparison matrix, illustrated in Table 1, using Eq. (2-5).

$$X_{j\ QZ}^{P} = \left(WW_{zq} \vee DD_{zq} \vee LL_{zq}\right), \qquad Q = \{1, \dots, q\}, Z = \{1, \dots, Z\}, c_{n_z} = c_{n_q} \in c_j, j = \{1, \dots, n\}; \tag{2}$$

$$WW_j = \sum_{Q=1}^{q} WW_{zq}\ ; \tag{3}$$

$$DD_j = \langle \sum_{Q=1}^{q} DD_{zq} \rangle - 1; \tag{4}$$

$$LL_j = \sum_{Q=1}^{q} LL_{zq}\ ; \tag{5}$$

In above equations, X_j^P presents the matrix, in which WW_j, DD_j, and LL_jexpress win, lose, and draw, respectively; WW_j refer to the situation where one criterion is more important in achieving the objectives compared to the other one. While LL_j stands for the situation where a criterion is less important than the criterion, DD_j reveals the equal importance between two criteria.

Table 1
The pairwise comparison of criteria

	c_{1_1}		c_{n_q}
c_{1_1}	1	$\cdots$	$\left(WW_{1,n} \vee DD_{1,n} \vee LL_{1,n}\right)$
	$\vdots$	$\ddots$	$\vdots$
c_{n_z}	$\left(WW_{n,1} \vee DD_{n,1} \vee LL_{n,1}\right)$	$\cdots$	1

Step 3. Calculation of the final weights is the third step of the WLD algorithm. The process employs Eq. (6,7), where w_j' stands for the weights determined by the decision-maker and $\mathbb{S}_j$ denotes the scores of criteria.

$$\mathbb{S}_j = WW_j + DD_j + LL_j; \tag{6}$$

$$w_j = w_j' \mathbb{S}_j \langle \sum_j w_j' \mathbb{S}_j \rangle^{-1}\ ; \tag{7}$$

3. Theoretical Framework

The proposed model is architected on three main variables, the weights computed by the WLD methods, the weights computed by the ML random forest employed to analyze the trends and ranks of the routes using ARWEN to generate. The workflow of the DSS is illustrated in Figure 1, in which the inputs, process, and the DSS outputs are displayed.

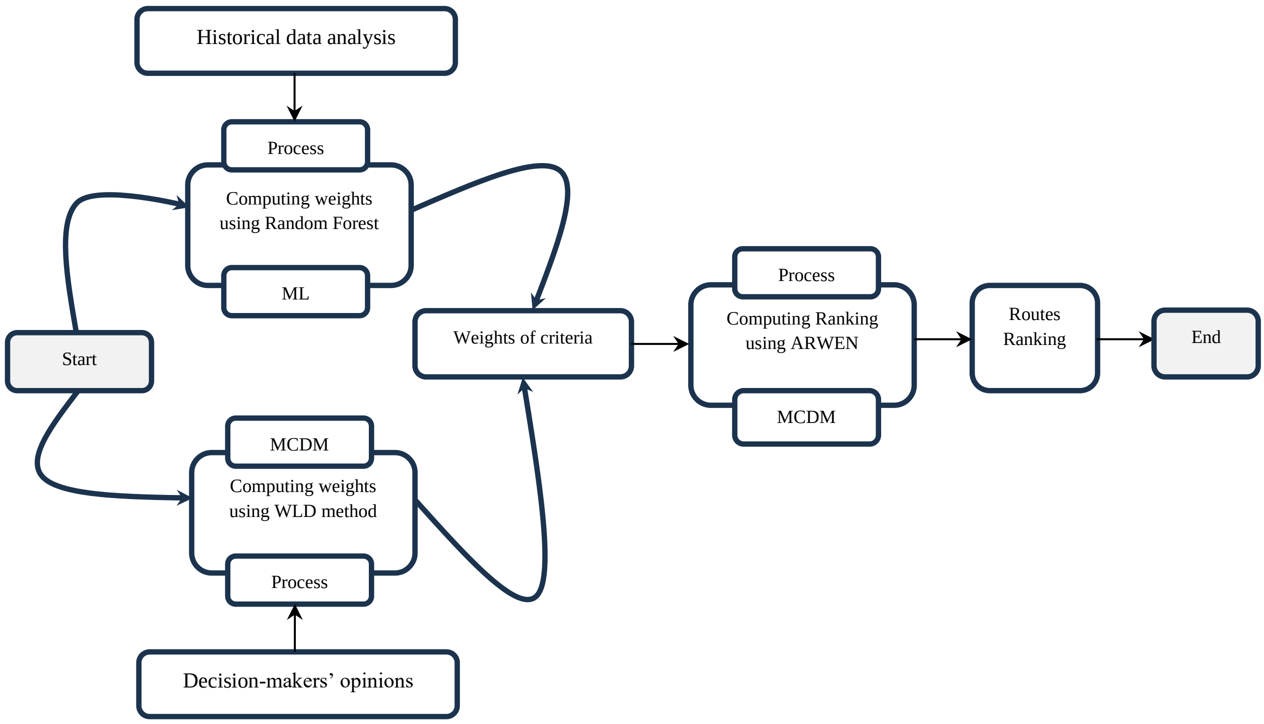

Fig. 1. The workflow of the proposed DSS

Eq. (8,9) show the aggregation of the weights computed by ML and weights computed by WLD method, in which ϑ, θ are the coefficients of the weights determined by decision-makers.

$$W_j = \langle \sum_j \frac{\vartheta w_j^{Ml} + \theta w_j^{WLD}}{2} \rangle^{-1} \frac{\vartheta w_j^{Ml} + \theta w_j^{WLD}}{2}, \qquad 0 < \vartheta, \theta \leq 1, \qquad j = \{1, \dots, n\}; \tag{8}$$

or simply

$$W_j = \vartheta w_j^{Ml} + \theta w_j^{WLD}, \qquad 0 < \vartheta, \theta \leq 1, \qquad j = \{1, \dots, n\}; \tag{9}$$

where

$\vartheta + \theta = 1$

4. The DSS Codes

4.1. Computing criteria weights

Eq. (8) outlines how to aggregate weights from ML and WLD methods. The following pseudo codes (Algorithm 1) display the implementation of the aggregation of the weights using python.

Algorithm 1

The aggregation of the weights using python - pseudo code

```
Program to aggregate weights from ML and MCDM methods using coefficients
Define aggregate_weights function with parameters:
w_j_ML (array of ML weights)
w_j_WLD (array of MCDM weights)
theta (coefficient for ML weights)
vartheta (coefficient for MCDM weights)
Ensure that the sum of theta and vartheta equals 1:
If theta + vartheta is not equal to 1
Raise an error "The coefficients theta and vartheta must sum up to 1."
Compute the aggregated weights using the formula:
W_j = (vartheta * w_j_ML + theta * w_j_WLD) / (vartheta * w_j_ML / 2 + theta * w_j_WLD / 2)
Return W_j
End function
Initialize example weights and coefficients:
w_j_ML = array of [0.2, 0.3, 0.5]
w_j_WLD = array of [0.4, 0.4, 0.2]
theta = 0.6
vartheta = 0.4
Call aggregate_weights with the example weights and coefficients
Print the result as 'Aggregated Weights: followed by the calculated weights'
```

The “aggregate_weights” function combines the ML weights, WLD weights, and their corresponding coefficients, ϑ and θ, to determine the combined weights for each criterion.

Moving forward, we will proceed with the implementation of the ML component, which involves computing the weights through a Random Forest classifier using a ‘imaginary database A.’ Typically, historical data with route criteria measurements and corresponding success or quality labels would be used for this purpose. However, since we do not have access to the data, we will simulate this process using random numbers (see the pseudo code - Algorithm 2).

Algorithm 2

Pseudo code - Program to simulate data and use Random Forest to compute ML weights

```
Import necessary libraries for RandomForestClassifier and data simulation
Simulate a dataset (Imaginary Database A)
    Create a dataset with 1000 samples, each sample having 5 features (criteria)
    Define the target variable as binary, indicating success (1) or failure (0)
Initialize the Random Forest classifier
    Set the number of estimators (trees in the forest) to 100
    Set a fixed random state for reproducibility
Fit the Random Forest classifier on the entire dataset
    Use the simulated feature set (X) and target (y)
Extract feature importances from the trained model
    These importances act as weights from ML
Normalize the ML weights so that their sum equals 1
    Divide each weight by the total sum of weights
Print the computed ML weights
End Program
```

The Random Forest classifier is employed on the simulated dataset. In the above example, the criteria used to evaluate the routes are Traffic Conditions, Road Quality, Safety Rating, Environmental Impact, and Scenic Value. Table 2 shows the machine learning computed random weights for the criteria based on the feature importances.

Table 2
the criteria and their random weights

Criteria	Weights
Traffic Conditions	0.2481
Road Quality	0.3912
Safety Rating	0.0801
Environmental Impact	0.1683
Scenic Value	0.1123

The weights shown in Table 2 reflect the relative importance of each criterion as learned from the simulated historical data. In an actual implementation, the real historical data ought to be used to determine these weights. Now ML computed weights are available, the process proceeds to integrate them with the MCDM computed weights using the aggregation formula and codes provided earlier.

In the code provided above, we simulated an "imaginary database A" by creating a synthetic dataset using the “make_classification” function from scikit-learn[†]. This function generates a random n-class classification problem, which in this context, we used to represent different routes with associated criteria and outcomes. This synthetic dataset stands for real historical data might have be in an actual "database A."

In a real-world application, "database A" must be an actual database containing historical records of route selections and their success or failure outcomes. The features (criteria like Traffic Conditions, Road Quality, etc.) and labels (outcomes of the routes) in "database A" would be used to train the ML model, such as the Random Forest classifier used in the example.

The make_classification function is a placeholder to demonstrate how you would use historical data to extract criteria weights with ML. In practice, the synthetic data must be replaced with the actual dataset, which might be similar to the following Pseudo codes (displayed in Algorithm 3):

Algorithm 3
Pseudo code - Program to load data and train a Random Forest classifier

1. Define placeholders for loading your actual database (Database A)
2. X_db_A would be the feature matrix from your Database A
3. y_db_A would be the outcomes (labels) from your Database A
4. Create a function to load your database
5. Define load_your_database function that:
6. Loads data from Database A
7. Returns features and outcomes
8. Call the function to load the database
9. Store returned features in X_db_A
10. Store returned outcomes in y_db_A
11. Train the Random Forest classifier on your actual data
12. Use the feature matrix (X_db_A) and outcomes (y_db_A) to fit the model
13. End Program

It is worth noting that the “load_your_database()” function is hypothetical and would need to be implemented according to how the actual data is stored and needs to be processed.

[†] *https://scikit-learn.org/stable/*

Decision Making: Applications in Management and Engineering
Volume 7, Issue 2 (2024) 572-590

4.2. Ranking the routes Computing criteria weights

As displayed in Figure 1 and explained in the theoretical framework section, ARWEN algorithm is employed to rank the routes. Below is the Pseudo code (shown in Algorithm 4) for implementing the ARWEN method according to Eq. (1).

Algorithm 4

Pseudo code - Program to select the optimal route using the ARWEN MCDM method

1. Program to select the optimal route using the ARWEN MCDM method
2. Import the necessary library for mathematical operations
3. Define the function arwen_method with parameters:
4. performance_matrix (each row represents a route, each column represents criterion performance)
5. aggregated_weights (weights for each criterion)
6. Procedure within arwen_method:
7. Determine the number of criteria and routes from the performance matrix
8. Initialize Gamma values for each route to zero
9. For each route:
10. Initialize sum of weighted performance to zero
11. For each criterion:
12. Determine the maximum performance for the criterion across all routes
13. Calculate the relative performance ratio for the current route and criterion
14. Accumulate the weighted performance using the ratio and aggregated weight for the criterion
15. Calculate the Gamma value for the current route as twice the number of criteria minus the sum of weighted performances
16. Determine the route with the highest Gamma value as the optimal route
17. Return the index of the optimal route and the Gamma values for all routes
18. End of function definition
19. Example usage:
20. Define a performance matrix for routes and criteria
21. Define aggregated weights for each criterion
22. Call arwen_method with the performance matrix and aggregated weights
23. Output the optimal route index and the Gamma values for all routes

End Program

This code defines the “arwen_method” function, which can be utilized with the data embedded in the decision matrix to select the optimal route. The above codes also include an example of how to call this function using a hypothetical “performance_matrix” and “aggregated_weights”.

5. Numerical Example

Let’s consider a numerical example, in which the objective is to select the most efficient and safe route for an autonomous delivery vehicle in a busy urban environment. The autonomous delivery vehicle must navigate a densely populated city with varying traffic conditions, road types, and safety concerns to deliver packages. The decision involves balancing efficiency, safety, and resource optimization. The vehicle has five potential routes to choose from, and these routes along with their associated attributes are presented in Table 3. To assess these routes, we have defined seven criteria: Distance, Estimated Time, Traffic Conditions, Road Quality, Safety Rating, Environmental Impact, and Scenic Value. You can find a detailed description of these criteria in Table 4 as well. The evaluation of routes' performance for the criteria of Traffic Conditions, Road Quality, Safety Rating, Environmental Impact, and Scenic Value is conducted using a Likert scale. In this scale, a rating of 1 indicates lower performance related to the specific criterion, while a rating of 5 signifies the highest performance achieved by the route in consideration of that particular criterion.

Table 3
The options the vehicle have with their associated attributes

Routes	Attributes
Route A	The shortest in distance but passes through high-traffic areas.
Route B	A longer route but with less traffic and better road conditions.
Route C	An intermediate route in terms of distance and traffic but passes through an area with a lower safety rating.
Route D	A route with moderate distance, known for its scenic views but prone to occasional road closures.
Route E	The longest route, but it bypasses major traffic hotspots and has the best road quality.

Table 4
The criteria and their associated descriptions

Criteria	Description
Distance (km)	The total distance of the route
Estimated Time (min)	The estimated time to complete the delivery
Traffic Conditions	Traffic congestion level
Road Quality	The condition of the roads
Safety Rating	Safety of the area
Environmental Impact	The level of emissions associated with the route
Scenic Value	The aesthetic value of the route, important for brand image

5.1. DSS Application and results

The input used to run the proposed DSS is the decision matrix, mainly the performance of the routes against the seven criteria. Using DSS also provides the weights. Accordingly, the decision matrix, including the weights computed by ML form the historical analysis, weights computed by WLD using the decision-makers opinions, and the performance of the routes against the seven criteria is demonstrated in Table 5. To compute the relative aggregated weights, we considered $\vartheta = 0.6$ and $\theta = 0.4$. The beneficial criteria (+) represents the criteria that higher value is expected and non-beneficial (-) criteria show the criteria that lower values are more favourable.

Table 5
The route selection decision matrix

Beneficial/ non-beneficial	-	-	-	+	+	+	+
w_j^{WLD}	0.15	0.2	0.15	0.1	0.15	0.15	0.1
w_j^{Ml}	0.23	0.175	0.13	0.075	0.22	0.12	0.05
W_j	0.198	0.185	0.138	0.085	0.192	0.132	0.07
	Distance (km)	Estimated Time (min)	Traffic Conditions	Road Quality	Safety Rating	Environmental Impact	Scenic Value
Route A	10	30	4	3	3	2	1
Route B	15	35	2	4	5	3	2
Route C	12	32	3	3	2	2	1
Route D	13	33	2	3	4	3	5
Route E	18	40	1	5	4	4	3

The ranked routes generated by the proposed DSS are exhibited in Table 6, with Route C emerging as the top choice. This selection is based on its balanced performance across efficiency, safety, environmental friendliness, and public image. It ensures the optimal route is chosen for the delivery task within this urban context.

Table 6
The routes ranking

Route	Γ_i	Rank
Route A	13.3363	2
Route B	13.25343	3
Route C	13.40776	1
Route D	13.23691	4
Route E	13.20065	5

5.2. DSS performance

To evaluate the performance of the DSS in terms of reliability of its results, we have conducted the following analysis.

5.2.1. Comparison

As ARWEN and WLD methods play pivotal roles in our DSS, the initial analysis involves comparing the outputs produced by our DSS to the results obtained from other MCDM methods, including TOPSIS and SAW. While there is no definitive superior MCDM method, comparing the results of various MCDM methods is a commonly employed approach for validating the outcomes of an MCDM method. This is done by seeking general consensus in the rankings. The rankings obtained by TOPSIS and SAW are illustrated in Table 7.

Table 7
The results generated by TOPSIS, SAW, and ARWEN

Route	DSS	SAW	TOPSIS
Route A	2	1	1
Route B	3	4	2
Route C	1	3	3
Route D	4	2	4
Route E	5	5	5

The comparative analysis of the results is presented in Figures 2 to 4. As depicted in Figure 2, there is almost no complete consensus among the rankings produced by the various algorithms. Therefore, as illustrated in Figures 3 and 4, the correlation between these rankings is calculated. The correlation between the rankings generated by the proposed DSS and the SAW method is 0.5, whereas the correlation with the TOPSIS method is 0.7. This indicates that our DSS produces results more similar to those of TOPSIS than SAW in selecting the best route.

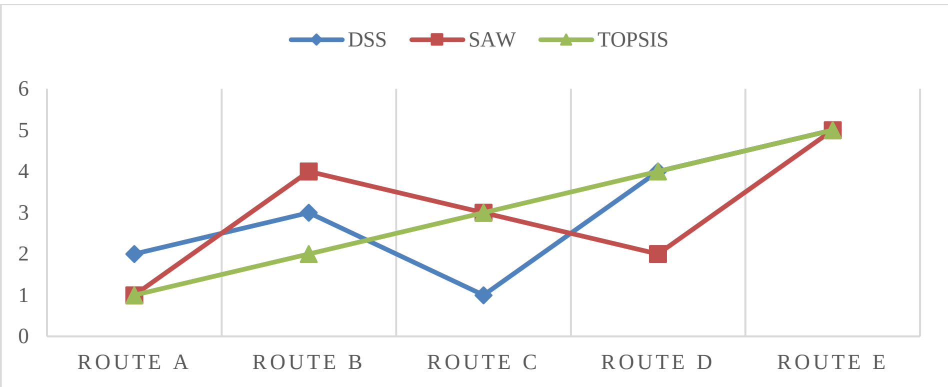

Fig. 2. The comparative analysis of the results generated by TOPSIS, DSS, and SAW

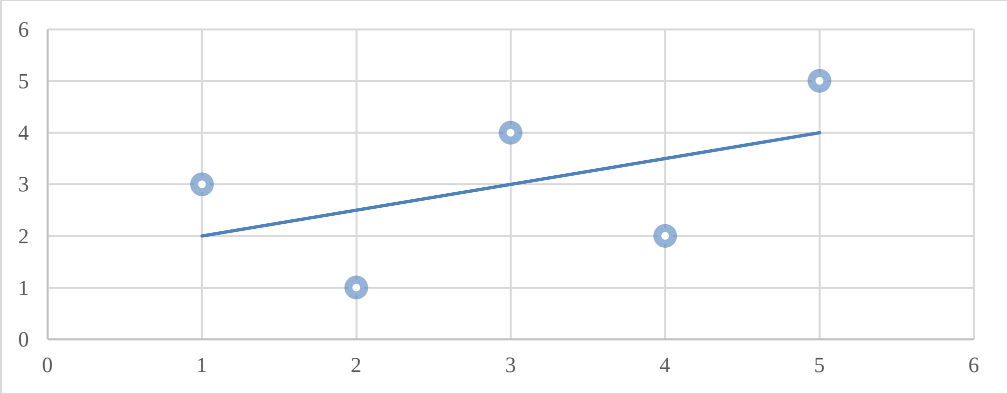

Fig. 3. The correlation between the rankings obtained by SAW and the proposed DSS

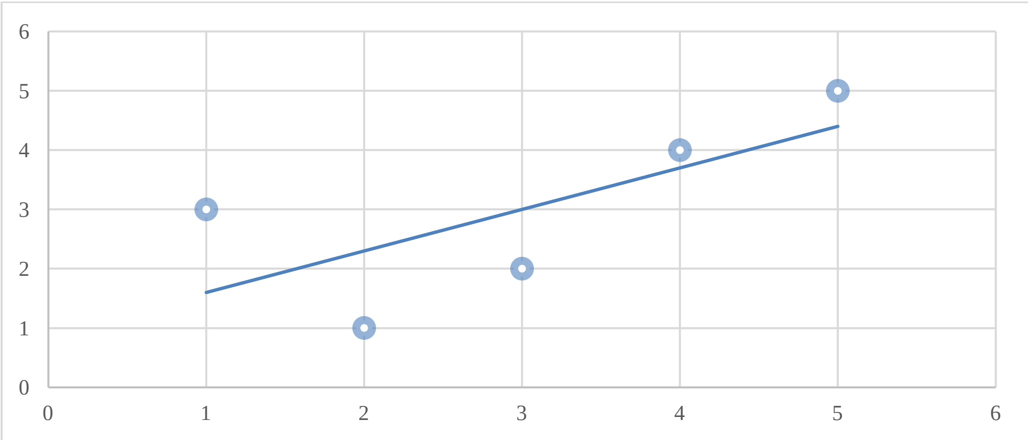

Fig. 4. The correlation between the rankings obtained by TOPSIS and the proposed DSS

5.2.1. Sensitivity analysis

As a second step in analyzing the results of the proposed DSS for selecting the best route, sensitivity analysis was conducted. New weights, as shown in Table 8, were applied to assess the potential changes in the rankings generated by the DSS, where SA stands for sensitivity analysis process, (see Table 7 and Table 8).

Table 8
The different sets of weights used for each analysis process

	Distance	Estimated Time	Traffic Conditions	Road Quality	Safety Rating	Environmental Impact	Scenic Value
SA 1	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429
SA 2	0.5000	0.0833	0.0833	0.0833	0.0833	0.0833	0.0833
SA 3	0.0833	0.0833	0.0833	0.0833	0.0833	0.0833	0.5000
SA 4	0.4000	0.0400	0.0400	0.0400	0.0400	0.0400	0.4000
SA 5	0.1000	0.1000	0.2000	0.2000	0.2000	0.1000	0.1000

The changes in rankings are presented in Table 9, and the comparative analysis is displayed in Figure 5. The comparative analysis clearly shows these changes in rankings. However, due to Route C's superior performance in almost every criterion and an overall balanced performance against all criteria, it has consistently been selected as the best route. The fact that Route C consistently emerges as the best route despite changes in rankings and applying different scenarios indicates stability in the DSS's performance under varying conditions. The fluctuation in rankings during the analysis demonstrates the reliability of our proposed DSS in selecting the best route for the specific case presented in this paper.

Table 9
The changes in rankings generated by the DSS using different sets of weights

Route	SA 1	SA 2	SA 3	SA 4	SA 5
Route A	2	4	2	3	2
Route B	3	2	3	2	4
Route C	1	1	1	1	1
Route D	4	5	5	5	3
Route E	5	3	4	4	5

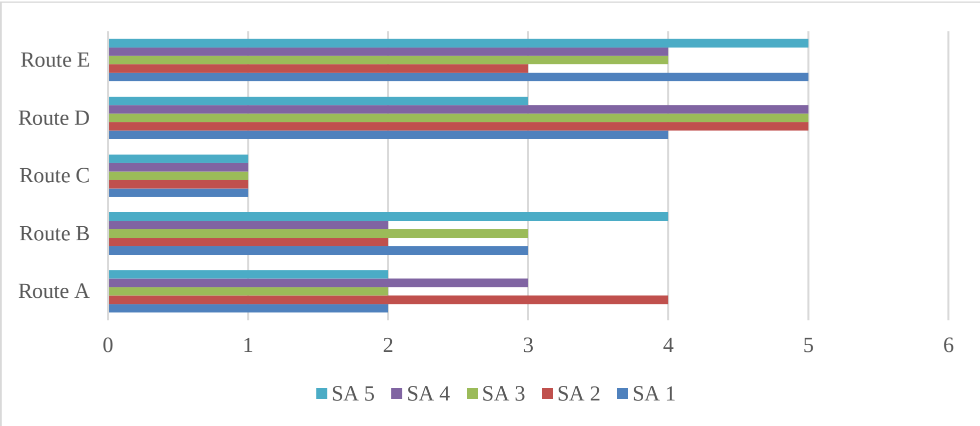

Fig. 5. The sensitivity analysis results

6. Impact

In this paper, we propose a novel DSS to assist decision-makers in selecting the best route for AVs. Constructed on MCDM algorithms, this system considers multiple criteria to provide informed decisions regarding the integration of AVs in the city of Geneva, Switzerland. The use of DSS has been proposed, studied, and applied in many studies in domains related to the integration or implementation of AVs, as demonstrated in the following works conducted by Tu *et al*., [33], Deveci *et al*., [34], Puchongkawarin and Ransikarbum [35], Amudha [36], and Jaoua *et al*., [37], and Li *et al*., [20]. MCDM methods are powerful tools that can be incorporated into DSSs or form the basis of a DSS. Their flexibility and comprehensiveness are particularly beneficial when addressing problems with components that cannot be manipulated or when decision-makers lack the capability to do so. Numerous decision-making problems related to the AVs, specifically in the final stages of the decision-making process where decision-makers must choose an option or a list of options, MCDM methods can create various advantages and bring many values, such as the objectivity of the decision, reliability of the outputs, reduction of error costs, effective risk management, and more. Various studies have employed MCDM methods in solving AVs related problems in their proposed DSS such as the Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE), ELimination and ChoiceExpressingREality (ELECTRE IV), and COmbinative Distance-based ASsessment

(CODAS) that are employed in Akram *et al*., [38], while Altay *et al*., [39] used Best-Worst Method (BWM). Gamal *et al.,* [22] used the method based on the removal effects of criteria (MEREC) and the combined compromise solution (CoCoSo). The Analytic Hierarchy Process (AHP), Multi-Attributive Border Approximation Area Comparison (MABAC) and (PROMETHEE II) used in Abdel-Basset *et al.,* [21] work as main basis of the proposed DSS, while Erdoğan *et al.*, [40] employed DEcision MAking Trial and Evaluation Laboratory (DEMATEL), Analytical Network Process (ANP), and VlseKriterijuska Optimizacija I Komoromisno Resenje (VIKOR) in their proposed DSS.

In general, the application of DSS or decision-making algorithms in general could significantly aid decision-makers in many ways, such as:

i. They can analyze traffic patterns and environmental data to suggest the most efficient routes for AVs, reducing travel time and congestion [41].
ii. By processing real-time data, DSS can identify potential safety hazards, helping in the development of safer AV navigation strategies [42].
iii. DSS predicts the demand for AV services in different areas, allowing for better allocation of vehicles and resources [43].
iv. DSS facilitates coordination between stakeholders and main decision-makers involved in urban transport systems.
v. By analyzing various scenarios and outcomes, DSS helps decision-makers to manage the related risks, i.e., identifying and decreasing risks associated with AV integration [44].
vi. DSS can be employed for the development of strategies that align with environmental sustainability goals [45,46].
vii. An optimized DSS assists in planning adjustments to existing urban infrastructure to accommodate AVs [47,48], such as changes in traffic systems management [49,50].
viii. It enables an assessment of the economic impact of AV integration, including cost-benefit analyses [51,52].
ix. By aggregating and analyzing large datasets, DSS supports informed decision-making based on empirical evidence (see the work done by Wen *et al.,* [53]).

Using AI and considering the traffic conditions, road quality, safety rating, environmental impact, and scenic value as the main criteria, our DSS in particular is designed to benefits the integration of the AVs in the Geneva's transportation system by the following aspects:

i. 1. Traffic Pattern Analysis.
ii. 2. Real-time Data collection and utilization.
iii. 3. Safety Considerations, by identifying routes with lower risk of accidents based on historical data and current conditions.
i. 4. Energy Efficiency, by selecting the most direct or least congested routes.
ii. 5. Infrastructure Compatibility such as consideration of the road quality.
iii. 6. Customization for Specific Needs by the flexibility of the changes of the criteria weights or adding/removing criteria form the problem's framework.

7. Conclusion and future work

In this article, we propose an ML-MCDM-based DSS to address the route selection problem in autonomous urban delivery. Our DSS is built upon two MCDM algorithms and a machine learning algorithm, namely Random Forest. To evaluate and select the best route, we utilized machine learning for historical analysis to determine the importance of criteria defined for route evaluation. Additionally, an MCDM method, WLD, was used for computing weights based on decision-makers' opinions. The ARWEN method is employed to rank the routes based on the aggregated relative weights computed by the ML-WLD model. The proposed DSS was applied to select the best route in a numerical example, and the results were compared with those from other MCDM algorithms. Sensitivity analysis was also conducted to evaluate the DSS's consistency and reliability, with the DSS proving its effectiveness in both aspects for the case study presented in the article. In the article, we also provided the python codes for the development of the proposed DSS.

MCDM methods are powerful tools that assist decision-makers in optimizing their decisions [54-58]. When combined with ML algorithms, they form even more powerful tools and create objective processes that yield reliable outputs. Utilizing the mentioned model aids in making more optimized decisions for real-time data analysis in dynamic urban environments. It effectively optimizes routes and strategies for operational efficiency. The proposed model balances objectives such as reducing travel time, enhancing safety, and minimizing emissions, thereby supporting smart city initiatives. It employs ranking algorithms that consider performance against each criterion. The integration of ML and historical analysis in the proposed DSS model not only ensures safety by identifying the safest routes but also continuously enhances decision-making through adaptability and learning. In general, to summarize the key findings and contributions:

i. This paper proposes a DSS that could be applied to solve decision-making problems.
ii. The DSS has been employed in the case of the ULTIMO project for optimized route selection, addressing not only the project's needs but also the gaps in the literature regarding existing DSS in route planning research.
iii. A code template for the development of the proposed DSS is provided, which aids researchers, experimenters, practitioners, and industries in applying the DSS to their specific problems.
iv. Sensitivity analysis is conducted to ensure the performance of the DSS, proving that it is effective in solving the route selection problem.

As mentioned, the sensitivity analysis conducted on the DSS to assess its consistency and reliability may not fully demonstrate its reliability. While Route C's consistent emergence as the top route despite ranking changes suggests stability under varying conditions, broader reliability typically requires more extensive testing and validation. This would confirm that the DSS performs effectively in diverse scenarios and with various datasets. Hence, the first recommendation for future research is to thoroughly assess the DSS's reliability, providing detailed evidence of its consistent performance and decision-making accuracy under different tests and conditions. Another interesting proposal is to integrate traffic management databases to enhance the ML algorithm's ability to derive criteria weights more reliably and to test the outcomes, following the analysis approach outlined in the first suggestion. Given that the model presented in the article is a conceptual framework for the DSS, our final recommendation is to apply the proposed DSS in real-world cases and gather feedback for further development and improvement.

Author Contributions

Conceptualization, S.Z, D.K, P.C, and A.V.J.; methodology, S.Z, D.K, P.C, and A.V.J.; software, S.Z, D.K, P.C, and A.V.J.; validation, S.Z, D.K, P.C, and A.V.J.; formal analysis, S.Z, D.K, P.C, and A.V.J.; investigation, S.Z, D.K, P.C, and A.V.J.; resources, D.K.; data curation, S.Z, D.K, P.C, and A.V.J.; writing—original draft preparation, S.Z, D.K, P.C, and A.V.J.; writing—review and editing, S.Z, D.K, P.C, and A.V.J.; visualization, S.Z, D.K, P.C, and A.V.J.; supervision, S.Z and D.K.; project administration, D.K.; funding acquisition, D.K. All authors have read and agreed to the published version of the manuscript.

Funding

This research is partly funded by ULTIMO project.

Data Availability Statement

In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. You might choose to exclude this statement if the study did not report any data (this section is mandatory).

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work has been co-funded by the European Union and the Swiss State Secretariat for Education, Research and Innovation (SERI), in the frame of the ULTIMO project. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union, CINEA or SERI. Neither the European Union nor the granting authorities can be held responsible for them.

Appendix

Table A1

The changes

Nr	Code metadata description	
C1	Current code version	*v1.0*
C2	Permanent link to code/repository used for this code version	https://github.com/Dunno4859/Software-Impacts
C4	Legal code license	MIT License
C5	Code versioning system used	
C6	Software code languages, tools and services used	Python, numpy, scikit-learn
C7	Compilation requirements, operating environments and dependencies	Python environment with numpy and scikit-learn libraries installed
C9	Support email for questions	Shervin.Zakeri@unige.ch

References

[1] Faisal, A., Kamruzzaman, M., Yigitcanlar, T., & Currie, G. (2019). Understanding autonomous vehicles. Journal of transport and land use, 12(1), 45-72. https://doi.org/10.5198/jtlu.2019.1405

[2] Konstantas, D., & Fournier, G. (2022). Fully Automated Public Transportation Mobility: A Dream or A Nightmare? https://doi.org/10.20944/preprints202209.0408.v1

[3] Crayton, T. J., & Meier, B. M. (2017). Autonomous vehicles: Developing a public health research agenda to frame the future of transportation policy. Journal of Transport & Health, 6, 245-252. https://doi.org/10.1016/j.jth.2017.04.004

[4] Shladover, S. E. (2018). Connected and automated vehicle systems: Introduction and overview. Journal of Intelligent Transportation Systems, 22(3), 190-200. https://doi.org/10.1080/15472450.2017.1336053

[5] Othman, K. (2022). Exploring the implications of autonomous vehicles: A comprehensive review. Innovative Infrastructure Solutions, 7(2), 165. https://doi.org/10.1007/s41062-022-00763-6

[6] Mallozzi, P., Pelliccione, P., Knauss, A., Berger, C., & Mohammadiha, N. (2019). Autonomous vehicles: state of the art, future trends, and challenges. Automotive systems and software engineering: State of the art and future trends, 347-367. https://doi.org/10.1007/978-3-030-12157-0_16

[7] Hakak, S., Gadekallu, T. R., Maddikunta, P. K. R., Ramu, S. P., Parimala, M., De Alwis, C., & Liyanage, M. (2023). Autonomous Vehicles in 5G and beyond: A Survey. Vehicular Communications, 39, 100551. https://doi.org/10.1016/j.vehcom.2022.100551

[8] Rana, M. M., & Hossain, K. (2023). Connected and autonomous vehicles and infrastructures: A literature review. International Journal of Pavement Research and Technology, 16(2), 264-284. https://doi.org/10.1007/s42947-021-00130-1

[9] Bathla, G., Bhadane, K., Singh, R. K., Kumar, R., Aluvalu, R., Krishnamurthi, R., Kumar, A., Thakur, R. N., & Basheer, S. (2022). Autonomous vehicles and intelligent automation: Applications, challenges, and opportunities. Mobile Information Systems, 2022(1), 7632892. https://doi.org/10.1155/2022/7632892

[10] Jing, P., Hu, H., Zhan, F., Chen, Y., & Shi, Y. (2020). Agent-based simulation of autonomous vehicles: A systematic literature review. IEEE Access, 8, 79089-79103. https://doi.org/10.1109/ACCESS.2020.2990295

[11] Narayanan, S., Chaniotakis, E., & Antoniou, C. (2020). Shared autonomous vehicle services: A comprehensive review. Transportation Research Part C: Emerging Technologies, 111, 255-293. https://doi.org/10.1016/j.trc.2019.12.008

[12] Sohrabi, S., Khreis, H., & Lord, D. (2020). Impacts of autonomous vehicles on public health: A conceptual model and policy recommendations. Sustainable Cities and Society, 63, 102457. https://doi.org/10.1016/j.scs.2020.102457

[13] Kopelias, P., Demiridi, E., Vogiatzis, K., Skabardonis, A., & Zafiropoulou, V. (2020). Connected & autonomous vehicles–Environmental impacts–A review. Science of the total environment, 712, 135237. https://doi.org/10.1016/j.scitotenv.2019.135237

[14] Spence, J. C., Kim, Y. B., Lamboglia, C. G., Lindeman, C., Mangan, A. J., McCurdy, A. P., Stearns, J.A., Wohlers, B., Sivak, A., & Clark, M. I. (2020). Potential impact of autonomous vehicles on movement behavior: a scoping review. American journal of preventive medicine, 58(6), e191-e199. https://doi.org/10.1016/j.amepre.2020.01.010

[15] Gandia, R. M., Antonialli, F., Cavazza, B. H., Neto, A. M., Lima, D. A. D., Sugano, J. Y., Nicolai, I., & Zambalde, A. L. (2019). Autonomous vehicles: scientometric and bibliometric review. Transport reviews, 39(1), 9-28. https://doi.org/10.1080/01441647.2018.1518937

[16] Caballero, W. N., Rios Insua, D., & Banks, D. (2023). Decision support issues in automated driving systems. International Transactions in Operational Research, 30(3), 1216-1244. https://doi.org/10.1111/itor.12936

[17] Gokasar, I., Pamucar, D., Deveci, M., & Ding, W. (2023). A novel rough numbers based extended MACBETH method for the prioritization of the connected autonomous vehicles in real-time traffic management. Expert Systems with Applications, 211, 118445. https://doi.org/10.1016/j.eswa.2022.118445

[18] Bonab, S. R., Ghoushchi, S. J., Deveci, M., & Haseli, G. (2023). Logistic autonomous vehicles assessment using decision support model under spherical fuzzy set integrated Choquet Integral approach. Expert Systems with Applications, 214, 119205. https://doi.org/10.1016/j.eswa.2022.119205

[19] Schwarting, W., Alonso-Mora, J., & Rus, D. (2018). Planning and decision-making for autonomous vehicles. Annual Review of Control, Robotics, and Autonomous Systems, 1, 187-210. https://doi.org/10.1146/annurev-control-060117-105157

[20] Li, J., Dai, J., Issakhov, A., Almojil, S. F., & Souri, A. (2021). Towards decision support systems for energy management in the smart industry and Internet of Things. Computers & Industrial Engineering, 161, 107671. https://doi.org/10.1016/j.cie.2021.107671

[21] Abdel-Basset, M., Gamal, A., Moustafa, N., Abdel-Monem, A., & El-Saber, N. (2021). A security-by-design decision-making model for risk management in autonomous vehicles. IEEE Access, 9, 107657-107679. https://doi.org/10.1109/ACCESS.2021.3098675

[22] Gamal, A., Abdel-Basset, M., Hezam, I. M., Sallam, K. M., & Hameed, I. A. (2023). An Interactive Multi-Criteria Decision-Making Approach for Autonomous Vehicles and Distributed Resources Based on Logistic Systems: Challenges for a Sustainable Future. Sustainability, 15(17), 12844. https://doi.org/10.3390/su151712844

[23] Biswas, A., Gazi, K. H., Sankar, P. M., & Ghosh, A. (2024). A Decision-Making Framework for Sustainable Highway Restaurant Site Selection: AHP-TOPSIS Approach based on the Fuzzy Numbers. Spectrum of Operational Research, 2(1), 1-26. https://doi.org/10.31181/sor2120256

[24] Bouraima, M. B., Jovčić, S., Dobrodolac, M., Pamucar, D., Badi, I., & Maraka, N. D. (2024). Sustainable Healthcare System Devolution Strategy Selection Using the AROMAN MCDM Approach. *Spectrum of Decision Making and Applications*, *1*(1), 46-63. https://doi.org/10.31181/sdmap1120243

[25] Hussain, A., & Ullah, K. (2024). An Intelligent Decision Support System for Spherical Fuzzy Sugeno-Weber Aggregation Operators and Real-Life Applications. *Spectrum of Mechanical Engineering and Operational Research*, *1*(1), 177-188. https://doi.org/10.31181/smeor11202415

[26] Ma, Y., Wang, Z., Yang, H., & Yang, L. (2020). Artificial intelligence applications in the development of autonomous vehicles: A survey. IEEE/CAA Journal of Automatica Sinica, 7(2), 315-329. https://doi.org/10.1109/JAS.2020.1003021

[27] Filippou, S., Achilleos, A., Zukhraf, S. Z., Laoudias, C., Malialis, K., Michael, M. K., & Ellinas, G. (2023). A machine learning approach for detecting gps location spoofing attacks in autonomous vehicles. In 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring) (pp. 1-7). IEEE. https://doi.org/10.1109/VTC2023-Spring57618.2023.10200857

[28] Caesar, H., Kabzan, J., Tan, K. S., Fong, W. K., Wolff, E., Lang, A., Fletcher, L., Beijbom, O., & Omari, S. (2021). nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. arXiv preprint arXiv:2106.11810. https://doi.org/10.48550/arXiv.2106.11810

[29] Qayyum, A., Usama, M., Qadir, J., & Al-Fuqaha, A. (2020). Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward. IEEE Communications Surveys & Tutorials, 22(2), 998-1026. https://doi.org/10.1109/COMST.2020.2975048

[30] Fu, Y., Yu, F. R., Li, C., Luan, T. H., & Zhang, Y. (2020). Vehicular blockchain-based collective learning for connected and autonomous vehicles. Ieee wireless communications, 27(2), 197-203. https://doi.org/10.1109/MNET.001.1900310

[31] Zakeri, S., Chatterjee, P., Konstantas, D., & Farr, A. S. (2023). Introducing alternatives ranking with elected nominee (ARWEN) method: a case study of supplier selection. Technological and Economic Development of Economy, 29(3), 1080-1126. https://doi.org/10.3846/tede.2023.18789

[32] Zakeri, S., Chatterjee, P., Cheikhrouhou, N., & Konstantas, D. (2022). Ranking based on optimal points and win-loss-draw multi-criteria decision-making with application to supplier evaluation problem. Expert Systems with Applications, 191, 116258. https://doi.org/10.1016/j.eswa.2021.116258

[33] Tu, Y. J., Shang, S. S., & Wu, J. (2023). Decision support or decision making? The critical decision roles of IS in autonomous vehicles. International Journal of Services Technology and Management, 28(3-4), 205-222. https://doi.org/10.1504/IJSTM.2023.131714

[34] Deveci, M., Mishra, A. R., Gokasar, I., Rani, P., Pamucar, D., & Özcan, E. (2022). A decision support system for assessing and prioritizing sustainable urban transportation in metaverse. IEEE Transactions on Fuzzy Systems, 31(2), 475-484. https://doi.org/10.1109/TFUZZ.2022.3190613

[35] Puchongkawarin, C., & Ransikarbum, K. (2021). An integrative decision support system for improving tourism logistics and public transportation in Thailand. Tourism Planning & Development, 18(6), 614-629. https://doi.org/10.1080/21568316.2020.1837229

[36] Amudha, G. (2021). ACDS—Assisted Cooperative Decision-Support for reliable interaction based navigation assistance for autonomous vehicles. Microprocessors and Microsystems, 87, 104241. https://doi.org/10.1016/j.micpro.2021.104241

[37] Jaoua, A., Ammar, M. B., & Awasthi, A. (2019). A decision support system for on-demand goods delivery using shared autonomous electric vehicles. International Journal of Decision Support System Technology (IJDSST), 11(2), 72-88. https://doi.org/10.4018/IJDSST.2019040105

[38] Akram, M., Zahid, K., & Kahraman, C. (2024). A new ELECTRE-based decision-making framework with spherical fuzzy information for the implementation of autonomous vehicles project in Istanbul. Knowledge-Based Systems, 283, 111207. https://doi.org/10.1016/j.knosys.2023.111207

[39] Altay, B. C., Boztas, A. E., Okumuş, A., Gul, M., & Çelik, E. (2023). How Will Autonomous Vehicles Decide in Case of an Accident? An Interval Type-2 Fuzzy Best–Worst Method for Weighting the Criteria from Moral Values Point of View. Sustainability, 15(11), 8916. https://doi.org/10.3390/su15118916

[40] Erdoğan, M., Kaya, İ., Karaşan, A., & Çolak, M. (2021). Evaluation of autonomous vehicle driving systems for risk assessment based on three-dimensional uncertain linguistic variables. Applied Soft Computing, 113, 107934. https://doi.org/10.1016/j.asoc.2021.107934

[41] Zhang, M. Y., Yang, S. C., Feng, X. J., Chen, Y. Y., Lu, J. Y., & Cao, Y. G. (2022). Route Planning for Autonomous Driving Based on Traffic Information via Multi-Objective Optimization. Applied Sciences, 12(22), 11817. https://doi.org/10.3390/app122211817

[42] Feng, S., Sun, H., Yan, X., Zhu, H., Zou, Z., Shen, S., & Liu, H. X. (2023). Dense reinforcement learning for safety validation of autonomous vehicles. Nature, 615(7953), 620-627. https://doi.org/10.1038/s41586-023-05732-2

[43] Hyland, M., & Mahmassani, H. S. (2018). Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests. Transportation Research Part C: Emerging Technologies, 92, 278-297. https://doi.org/10.1016/j.trc.2018.05.003

[44] Xie, X. F., & Wang, Z. J. (2018). SIV-DSS: Smart In-Vehicle Decision Support System for driving at signalized intersections with V2I communication. Transportation Research Part C: Emerging Technologies, 90, 181-197. https://doi.org/10.1016/j.trc.2018.03.008

[45] Bechtsis, D., Tsolakis, N., Vlachos, D., & Srai, J. S. (2018). Intelligent Autonomous Vehicles in digital supply chains: A framework for integrating innovations towards sustainable value networks. Journal of cleaner production, 181, 60-71. https://doi.org/10.1016/j.jclepro.2018.01.173

[46] Davidson, R. (2020). The algorithmic governance of connected autonomous vehicles: Data-driven decision support systems and smart sustainable urban mobility behaviors. Contemporary Readings in Law and Social Justice, 12(2), 16-24. https://www.ceeol.com/search/article-detail?id=910387

[47] Sadaf, M., Iqbal, Z., Javed, A. R., Saba, I., Krichen, M., Majeed, S., & Raza, A. (2023). Connected and automated vehicles: Infrastructure, applications, security, critical challenges, and future aspects. Technologies, 11(5), 117. https://doi.org/10.3390/technologies11050117

[48] Bari, D., Bergot, T., & Tardif, R. (2023). Fog Decision Support Systems: A Review of the Current Perspectives. Atmosphere, 14(8), 1314. https://doi.org/10.3390/atmos14081314

[49] Gokasar, I., Simic, V., Deveci, M., & Senapati, T. (2023). Alternative prioritization of freeway incident management using autonomous vehicles in mixed traffic using a type-2 neutrosophic number based decision support system. Engineering Applications of Artificial Intelligence, 123, 106183. https://doi.org/10.1016/j.engappai.2023.106183

[50] Deveci, M., Pamucar, D., & Gokasar, I. (2021). Fuzzy Power Heronian function based CoCoSo method for the advantage prioritization of autonomous vehicles in real-time traffic management. Sustainable Cities and Society, 69, 102846. https://doi.org/10.1016/j.scs.2021.102846

[51] Zavadskas, E. K., Stević, Ž., Turskis, Z., & Tomašević, M. (2019). A novel extended EDAS in Minkowski Space (EDAS-M) method for evaluating autonomous vehicles. Studies in Informatics and Control, 28(3), 255-264. https://doi.org/10.24846/v28i3y201902

[52] Johnson, A. (2020). The smart automation of urban governance: sustainable mobilities and data-driven decision support systems. Contemp. Readings L. & Soc. Just., 12, 51. https://www.ceeol.com/search/article-detail?id=884442

[53] Wen, T., Mihăiţă, A. S., Nguyen, H., Cai, C., & Chen, F. (2018). Integrated incident decision-support using traffic simulation and data-driven models. Transportation research record, 2672(42), 247-256. https://doi.org/10.1177/0361198118782270

[54] Kawecka, E., Perec, A., & Radomska-Zalas, A. (2024). Use of the Simple Multicriteria Decision-Making (MCDM) Method for Optimization of the High-Alloy Steel Cutting Processby the Abrasive Water Jet. *Spectrum of Mechanical Engineering and Operational Research*, *1*(1), 111-120. https://doi.org/10.31181/smeor11202411

[55] Sarfraz, M. (2024). Application of Interval-valued T-spherical Fuzzy Dombi Hamy Mean Operators in the antiviral mask selection against COVID-19. *Journal of Decision Analytics and Intelligent Computing*, *4*(1), 67–98. https://doi.org/10.31181/jdaic10030042024s

[56] Imran, R., Ullah, K., Ali, Z., & Akram, M. (2024). A multi-criteria group decision-making approach for robot selection using interval-valued intuitionistic fuzzy information and aczel-alsina bonferroni means. *Spectrum of Decision Making and Applications*, *1*(1), 1-32. https://doi.org/10.31181/sdmap1120241

[57] Asif, M., Ishtiaq, U., & Argyros, I. K. (2024). Hamacher Aggregation Operators for Pythagorean Fuzzy Set and its Application in Multi-Attribute Decision-Making Problem. *Spectrum of Operational Research*, *2*(1), 27-40. https://doi.org/10.31181/sor2120258

[58] Phulara, S., Kumar, A., Narang, M., & Bisht, K. (2024). A novel hybrid grey-BCM approach in multi-criteria decision making: An application in OTT platform. *Journal of Decision Analytics and Intelligent Computing*, *4*(1), 1–15. https://doi.org/10.31181/jdaic10016012024p