Trust in real-time decision support: A case study in drone assisted maritime operations

Dominic J. Duxbury, Norman W. Paton, John A. Keane

Abstract

Decision makers increasingly operate in information-rich environments, where static or mobile sensing devices can provide numerous data points on which to base decisions. One prominent tool in this space is drones, which can be deployed to glean targeted information. However, deciding how best to deploy available drones is nontrivial, and stands to benefit from decision support aids that plan routes taking into account user preferences. This in turn raises the question of the features that a decision support system should provide to ensure that it is trusted by users. This study demonstrates a methodology applying partial least squares structural equation modelling (PLS-SEM) to assess decision support systems and their features in the absence of clear success criteria. We propose a theoretical framework to model the antecedents of trust in real-time decision support. We conducted a user study with a drone-assisted maritime operations scenario to evaluate the effectiveness of user preferences, explanation, and dynamic updates. The results show that transparency has a strong impact on trust, and that transparency can be improved through the inclusion of user preferences.

Keywords: Drones; Trust; Decision support systems

1. Introduction

Sea ports around the world face difficult challenges, including monitoring sea and ground traffic, port safety and emergency response. The amount of traffic these ports see has increased considerably, along with the need for more rigorous safety and prompt emergency response [39, 29]. Drones have been identified as a technology for tackling these challenges, by providing a means for aerial situational awareness beyond the shoreline [12, 25]. The decision making surrounding the management of these drones is just one aspect of the expanding role of managing a harbour. As such, there is an increasing need for decision support for the jobs comprising harbour management.

One task with the potential to improve harbour safety is automatic identification of ships approaching the harbour. Drones could be employed to take photos of ships for identification of traffic and potential threats. Selecting an appropriate route for these drones is a difficult task that could be simplified through a decision support system. Ships are always moving and can quickly change direction, therefore it is important for routes to be dynamically updated as the scenario unfolds. In this kind of high-stakes decision making domain, experts are relied upon to make a final decision, supported by decision support systems (DSS). Together they form a human-computer team, performing better than either the human or computer alone [43, 42].

An important aspect underpinning the effectiveness of human-computer teams is trust. If a decision maker does not trust the system they are working with, then useful outputs can be discarded or ignored. On the other hand if a user has too much trust in a system this can lead to over-reliance [4, 22]. Therefore, an effective team requires a decision maker to be aware of what the system does and does not know. This highlights the importance of *interpretability* in DSS. Interpretability is the ability to explain or to present in understandable terms to a human [10]. This includes giving decision makers the ability to understand what aspects a system has taken into consideration, and how it has used these factors to arrive at a solution. We refer to this capability as *transparency* [44]. Transparency allows a decision maker to make use of their expert knowledge, supplemented by the systems' ability to process large amounts of data. For a system to be interpretable, it should provide enough information for its decision process to be understood, without overloading a user. Therefore, a system should be designed to be transparent, without inducing high *cognitive load*. Unfortunately, any features added to a system are likely to come with additional cognitive load, therefore when building DSS it is important to scrutinise the cost-benefit of features. We have identified decision maker preferences [34] and explanation [18] as two system features that could help improve transparency in DSS.

DSS are most often employed to deal with multi-objective optimisation problems. This means selecting a solution requires a decision maker to make trade-offs between multiple objectives [41]. A system recommendation automates the procedure of recognising worthwhile trade-offs between objectives. Preferences allow a decision maker to input their own knowledge into this procedure, guiding the system in selecting an appropriate solution [27]. This improves transparency by giving decision makers an understanding of how the system values each objective [34]. The other feature, explanation, is the provision of reasoning describing how an output was reached. A well executed explanation directly addresses the problem of a decision maker recognising the limits of a system, providing transparency [18]. Our harbour management task occurs in real-time, so an explanation must also be quick to understand due to the time limits placed on the decision making process. A third feature, relevant to the real-time nature of our system, is dynamic updates. This is the ability to provide updated recommendations, whilst an action is being undertaken.

In this paper, we assess the transparency, cognitive load, trust and satisfaction of a decision support system for a real-time single unmanned aerial vehicle (UAV) task assignment problem. We measure the effect on these concepts of the decision support features: preferences, explanation and dynamic updates.

This paper provides two contributions. Firstly this study puts forward a trust-based model for interactions with real-time DSS. We have validated this model by applying the Partial Least Squares Structural Equation Modeling (PLS-SEM) technique [45], providing empirical evidence that transparency is a strong determinant of a decision makers trust and satisfaction with a system. This study also assessed the effect of explanation, preferences and dynamic updates on our model. This gives designers of DSS an indication of which features are likely to assist decision making in a dynamic environment. The second contribution of the paper is a methodology for the assessment of features in DSS without a clear metric for success. By allowing users to each engage with different versions of the system, having different features enabled or disabled, we can measure their impact on trust and its antecedents.

Section 2 presents a background for the Vehicle Routing Problem, the UAV Task Assignment problem and PLS-SEM. The architecture of the DSS is given in Section 3. Section 4 proposes our research model and our hypotheses. Section 5 follows with a description of our research methodology and data collection. This section also includes an analysis of results. Section 6 provides a discussion of the findings, and we conclude with limitations and implications of the study.

2. Literature Review

In this section, an overview of the literature surrounding the problems solved and techniques used in this paper is presented. Firstly, an introduction to the vehicle routing problem (VRP) and some studies producing VRP DSS are described. Next, we introduce the UAV task assignment problem (UAVTAP) and the application of multi-objective evolutionary algorithms (MOEA) to solve them. The final section introduces PLS-SEM and previous studies applying this statistical method to examine the antecedents of trust in systems.

2.1. Vehicle Routing Problem

VRP is a well known problem in operational research and combinatorial optimisation, in VRP, routes must be assigned to a set of vehicles that must visit a set of customers such that the total cost of the operation is minimised. VRP has been tackled in a wide array of real-world systems.

Santos *et al.* [38] introduce a web-based spatial decision support system for waste collections. The system provides static solutions with inputs in terms of constraints: shift time limit, vehicle types, capacities and an attribute to maximise/minimise. The system is therefore limited by its ability to optimise towards multiple objectives, requiring the user to specify a single attribute to optimise towards.

Addressing this issue in the Safe Route Planner, a decision support system designed to provide drivers with recommendations for the safest route between two locations, Sarraf *et al.* [40] integrate Multi-Criteria Decision Analysis (MCDA) methods. The integration of user preferences allows the system to optimise towards multiple objectives at once and trade-off between the shortest, fastest and safest routes. They assess the Analytic Hierarchy Process (AHP), Fuzzy AHP, TOPSIS, Fuzzy TOPSIS and PROMETHEE for suitability, concluding that AHP is most appropriate due to its simplicity and robustness.

Abbatecola *et al.* [3] introduce a decision support approach for postal delivery and waste collection. This system provides static routes for offline optimal planning of vehicle assignments. They apply a two-phase heuristic algorithm based on a clustering strategy and a fast insertion heuristic for solving a travelling salesman problem. This approach scales very well compared to a mixed integer linear programming approach, allowing results to be quickly recalculated. This improvement paves the way for future work which includes plans to modify in real-time the planning of routes.

These papers highlight user preferences as a desired feature for future VRP decision support systems. This is because the suitability of routes in the real-world depend upon a wide array of criteria. To complement the addition of preferences, explanation is put forward as another useful feature for decision support. Explanation refers to the addition of capabilities allowing the user to distinguish the trade-offs made between solutions according to the user preferences. We also identify dynamic updates as a desired feature of VRP decision support systems.

2.2. *UAV Task Assignment Problem*

UAVTAP consists of finding an optimal assignment of UAVs to a set of tasks [9]. UAVTAP shares many characteristics with VRP, with the difference being that UAVTAP allows multiple visits and sub-tours.

Ramerirez *et al.* first formulated UAVTAP as a Constraint Satisfaction Problem (CSP), with the mission being modelled and solved using constraint satisfaction techniques [30]. In a later paper UAVTAP is formulated as a multi-objective optimisation. Their objective consists of minimising the number of drones employed, the total flight time, the total fuel, the total distance, total cost and the time taken. The original approach was combined with a MOEA [32], this algorithm provides an estimate of the Pareto Optimal Front (POF), i.e. the set of all non-dominated solutions of the problem. The paper concludes that, as the complexity of the mission increases the number of solutions in the POF becomes huge, and therefore the time needed to calculate the complete POF becomes intractable.

Addressing this issue, Ramirez *et al.* [31] introduce a Knee-Point MOEA intending to reduce the POF to a set of the most likely best solutions. This reduces the size of the POF from hundreds to tens of solutions. This however still leaves a difficult task for decision makers who must select the most appropriate mission plan. In another paper, the authors rank the outputted POF using a selection of MCDA algorithms according to user preferences [33]. The work assumes that the decision maker cannot provide *a priori* information regarding preferences on criteria. This approach can therefore become costly in a dynamic setting, as the POF must be recalculated each time the mission scenario is updated.

Coelho *et al.* [8] also proposed a multi-objective UAVTAP. Taking inspiration from a multi-criteria view of real systems, the approach considers seven different objective functions which it seeks to minimise using a Mixed-Integer Linear Programming (MILP) model solved by a matheuristic algorithm. This produces an estimate of the POF but the paper does not make an attempt to rank the non-dominated solutions. This approach has also been employed by Beheshifar *et al.* [6] for the location-allocation of clinics. In their paper they produced a complete POF then applied TOPSIS as a *posteriori* preference method to select the most appropriate solution.

In our work, we apply the Analytic Hierarchy Process as a fitness function, rather than applying a MOEA. This approach is enabled by the fact that the decision makers preferences are available prior to route calculation. The approach allows us to maintain a solution set ranked according to preferences, via a continuously running evolutionary algorithm, as the situation and therefore solutions, evolve over time.

The similarities between VRP and UAVTAP suggest that a decision support approach would be appropriate for UAVTAP problems. We have highlighted features from the decision support systems for VRP, including: dynamic updates, user preferences, and explanation.

2.3. *PLS-SEM*

The Swedish econometrician Herman Wold [45] developed the statistical methods underpinning PLS-SEM. PLS-SEM estimates partial model structures by combining principal components analysis with ordinary least squares regressions [13]. This statistical method allows us to analyse complex interrelationships between observed and latent variables. A latent variable is a variable not observed, but inferred from the observed variables. Often, latent variables are aggregated observable variables, created to represent an underlying concept.

In the case of decision support, we can use latent variables to represent user trust and its antecedents in a system. Kim *et al.* [19] used PLS-SEM to assess the role of trust, perceived risk, and their antecedents in consumer decision making. The paper finds that user trust both directly and indirectly affects their intention to purchase online. PLS-SEM allows the authors to show that trust is a critical facet of the decision making process.

In another paper, Hsu *et al.* [15] use PLS-SEM to examine factors affecting intention to repurchase in online group-buying, including trust. The authors find that satisfaction with the website and satisfaction with sellers exert significant influences on user intention to repurchase. In addition, the results indicate that trust in a website is a strong predictor of satisfaction with a website, and trust in a seller is a strong predictor of satisfaction with the seller.

The literature indicates that both trust and satisfaction are key predictors for user intention to engage with a system or new technology. It is therefore important to gain an understanding of which features of decision support systems influence trust and satisfaction. In our case, we observe the features enabled for each user: dynamic updates, user preferences and explanation, and assess the effect each feature has on our latent variables: transparency, cognitive load, trust and satisfaction.

3. Harbour Management Decision Support System and Features

To evaluate our research model, a decision support system for harbour management has been developed. In this section, we outline the harbour management task, then the architecture and its components are described. The final subsections describe the experiment set-up and the implementation of explanation, preferences and dynamic updates. The system utilises a cloud architecture shown in Figure 1.

3.1. Harbour Management Task

In this task, a decision maker takes the role of a harbour master managing a harbour. The harbour master controls a single drone to identify ships close to the harbour zone. The job of the user is to select a route for the drone, visiting as many ships as possible before they reach the harbour. The length of these routes is limited by the fuel of the drone. Once the drone has run out of fuel, it must return to the refuel point, located within the harbour zone. We assume that the most suitable route may depend on different criteria: *ship distance*, *ship direction* and *identification speed*.

- **Ship Distance** - A measure of the distance of ships from the harbour.
- **Ship Direction** - A measure of how directly ships are heading towards the harbour.
- **Identification Speed** - A measure of the time taken to identify each ship over the route.

The lower the *Ship Distance*, the closer it is to the harbour, and the more important it may be to identify. It may be more useful to identify a vessel if its *Ship Direction* is towards the harbour rather than away. Finally, the *Identification Speed* indicates if a ship can be identified quickly, leaving more time to identify other ships.

3.2. Architecture

The decision maker operates the decision support system through a web interface. To access this interface, a get request is made to the *Static File Server* (1) and the static files are returned. From here the user can input their credentials and start a scenario. When a user starts a scenario, a scenario request is sent to the *User Server* (2) and the user is assigned a set of features. The *User Database* (3) contains the id for a user, their enabled/disabled features and their score for a session.

When a scenario request is authenticated it is placed onto the *Message Queue* (4). Once the *Track Processor* (5) receives a scenario request it retrieves scenario data from the *Scenario Database* (3). This database holds a set of scenarios defined by a set of ships and their position at each point in time called tracks. The *Track Processor* processes the stream of tracks, tracking the position of the drone and which tracks have been identified. The resultant stream of drone and ship tracks is passed to the front-end via the *Message Queue*.

This stream is also processed further by the *Route Processor* (6). This component calculates a ranking of routes for each time-step and passes it to the front-end via the *Message Queue*. Once a scenario has begun, a user can activate a route from the ranked recommendations or by drawing a path between tracks. The drone will then identify ships along the active route. If the recommendations are not appropriate, a user can input preferences to refine the selection of routes. This alters the weighting of criteria in the fitness function of the evolutionary algorithm.

3.3. Evolutionary Algorithm

In this section we describe the algorithms used to generate routes within the *Route Processor*. To calculate a dynamically updated ranking of routes we apply a continuously running, streaming evolutionary algorithm.

To optimise towards multiple objectives we apply the Analytic Hierarchy Process (AHP) [36] as a fitness function. AHP is a structured technique for organising and analysing complex decisions. AHP consists of an overall goal, a group of options or alternatives for reaching the goal and a group of factors or criteria that relate the alternatives to the goal; the criteria can be further broken down. These criteria generally have different

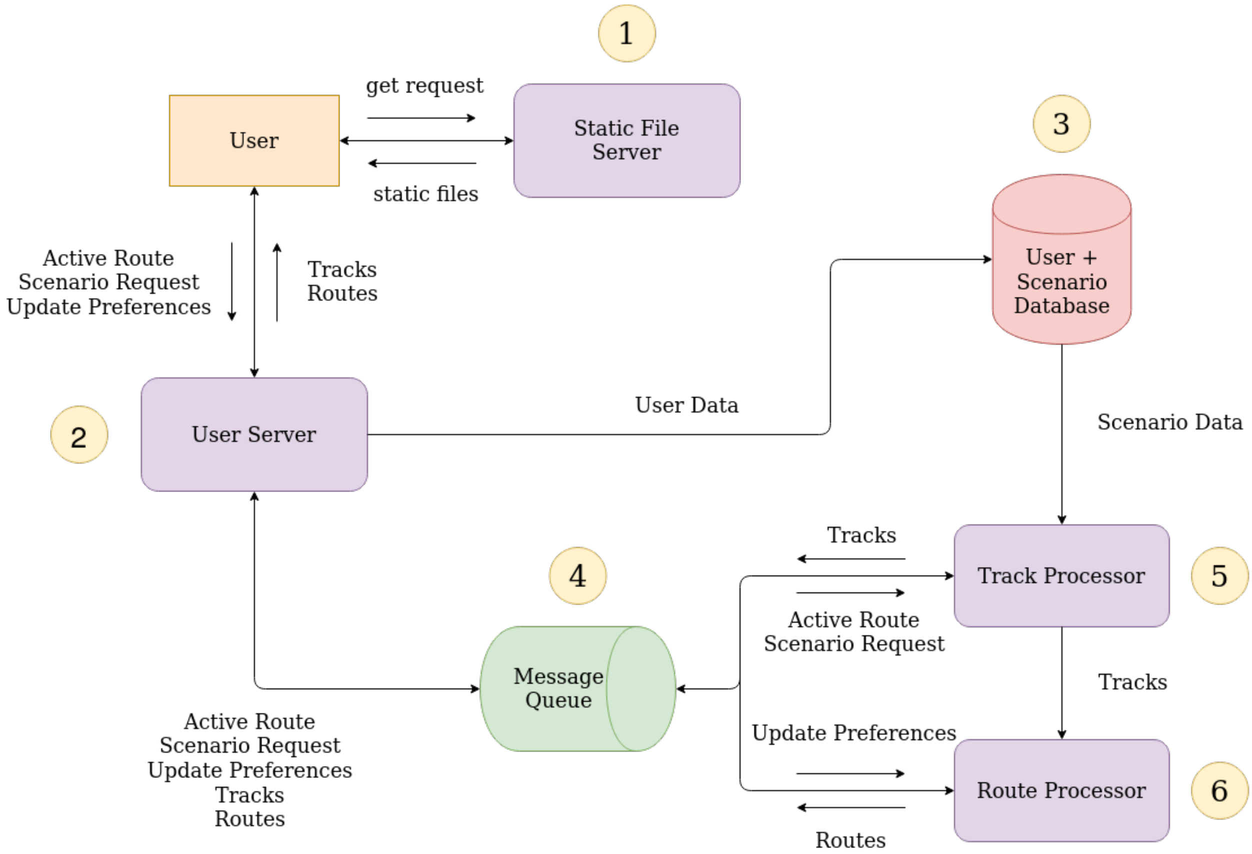


Figure 1: The system architecture

values for different decision makers and so the algorithm requires users to express their preferences.

The user preferences are expressed in the form of pairwise comparisons. For instance, a decision maker could express that "Ship Distance is more important than Ship Direction". Pairwise comparisons are easy for a user to express and model the users' knowledge within the system. The comparisons are then used to generate weightings for each criterion. If no comparisons are provided then we set weightings equally.

Figure 2 shows the flow of data through the *Route Processor*. The *Solution Creator* takes the current state of the scenario and the previous ranking, applying mutate(x) and crossover(x,y) operations to create new solutions. For the first generation, it creates a random set of routes. The *Criteria Calculator* then calculates the criteria values for each route. In streaming, it is common to perform set-based aggregations over subsets of events that fall within a period of time. This set of elements is referred to as a window. We take a window of recently validated routes in the *Solution Ranker*, and aggregate them into a ranking. To produce a ranking, criteria values must also be scored. To do this the values are first normalised according to the range of values across all solutions using the following formula:

$$Norm(x) = \frac{x - minX}{maxX - minX}$$

Here $minX$ and $maxX$ are the smallest and largest criteria values respectively. The values are then compared pairwise to generate a comparison matrix. For three solutions S_1, S_2 and S_3 and a criterion X with normalised criteria values x_1, x_2, x_3, we would generate a comparison matrix C.

$$\mathbf{C} = \begin{array}{c c} & \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{array}{c} S_1 \\ S_2 \\ S_3 \end{array} & \begin{bmatrix} 1 & f(x_1, x_2) & f(x_1, x_3) \\ f(x_2, x_1) & 1 & f(x_2, x_3) \\ f(x_3, x_1) & f(x_3, x_2) & 1 \end{bmatrix} \end{array}$$

We tested two separate formulas for comparing criteria values, depending on whether the values fall along a linear scale (1) or an exponential scale (2).

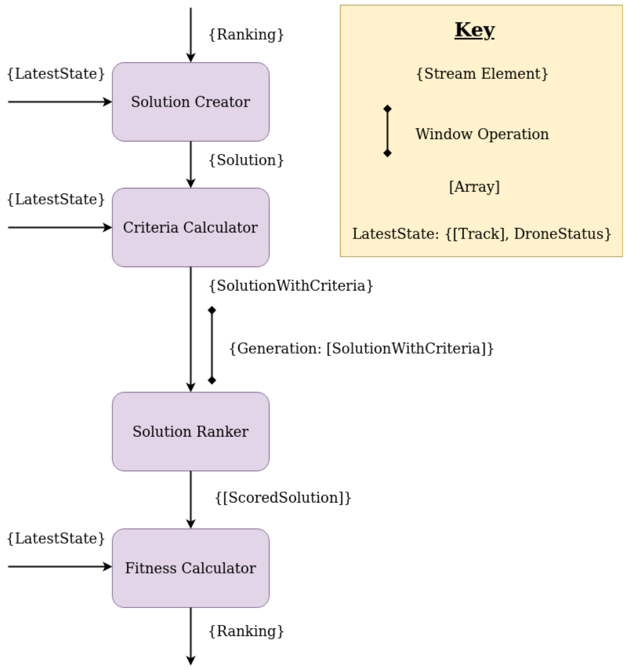


Figure 2: The flow of data through the continuous genetic algorithm

$$g(x, y) = |(x - y) \times 8| + 1 \tag{1}$$

$$f(x, y) = e^{x-y} \tag{2}$$

These formulas map two normalised values (x, y) to the fundamental scale proposed by Saaty [36]. In formula (1) $|x|$ refers to the absolute value of x. We call a matrix $C = (c_{ij})$ *consistent* [5] if

$$c_{ij} \cdot c_{jk} = c_{ik} \ \forall\ i, j, k = 1, 2, ..., n$$

We chose to apply formula (2) rather than (1) as it produces comparison matrices that satisfy this equation for all values of i, j, k as shown below.

$$f(i, j) \cdot f(j, k) = f(i, k)$$
$$e^{i-j} \cdot e^{j-k} = e^{i-k}$$
$$e^{i-k} = e^{i-k}$$

The principle eigenvector of the comparison matrix for each criterion represents the priority vector from which we derive the scores for the respective criteria value of each solution. The eigenvector of a comparison matrix can only be considered meaningful as a priority vector in a consistent or near consistent matrix [37]. Therefore, a formula creating consistent matrices is crucial to produce stable rankings in a dynamic environment.

Within the *Fitness Calculator*, criteria value scores are then multiplied by the relevant criteria weightings, and summed across each solution. This process produces the scores that are used to derive a global ranking.

When selecting routes for the next generation of the algorithm, we first chose the route with the highest score. We then applied a fitness penalty to routes that overlap with the set of routes chosen for the next generation. This was intended to promote a diverse set of routes for recommendation. The scores are discounted according to the formula (3) with overlap ($O(x, R)$) being calculated using formula (4). For formula (4), x refers to the route, R refers to the set of selected routes, $|R|$ refers to the number of routes selected for the next generation so far, $|x|$ refers to the numbers of ships in a route and $N(s)$ refers to the number of routes in R which visit ship s.

$$(3)\quad discount(x) = Score(x) * e^{2O(x,R)}$$

$$(4)\quad O(x, R) = 1 - \sum_{s}^{R} \frac{N(s)}{|R||x|}$$

This diversified set of routes is then used for the next generation of the genetic algorithm within the *Solution Creator*. The interface is updated with the latest set of ranked diversified routes in one-second intervals.

3.4. Interface

In this section we explain how the user interacts with the system. The user interface for the harbour management task is shown in Figure 3. The window includes a *map* (1), the *route selection popover* (2) in the bottom left and a *scenario information popover* (3) in the top right. To render the map, Leaflet™ [2] was chosen as an open-source alternative to Google Maps™.

The *map* shows the current condition of the scenario, including: tracks, the drone, any visible routes, the harbour zone and the refuel point. The tracks are indicated by a ship icon coloured yellow or green for unidentified or identified respectively. The *harbour zone* (4) is displayed as a blue polygon and the position of the refuel point is shown as a fuel icon within.

The *route selection popover* shows the routes recommended by the system. These routes are ranked according to their fitness and similarity to the best possible route. The user can toggle each route as visible/hidden on the map. Any visible routes will be displayed within the *scenario information popover*. When a route is visible and has been deemed acceptable, the user can activate it by clicking *activate* in the *scenario information popover*. This popover also includes the current stage,

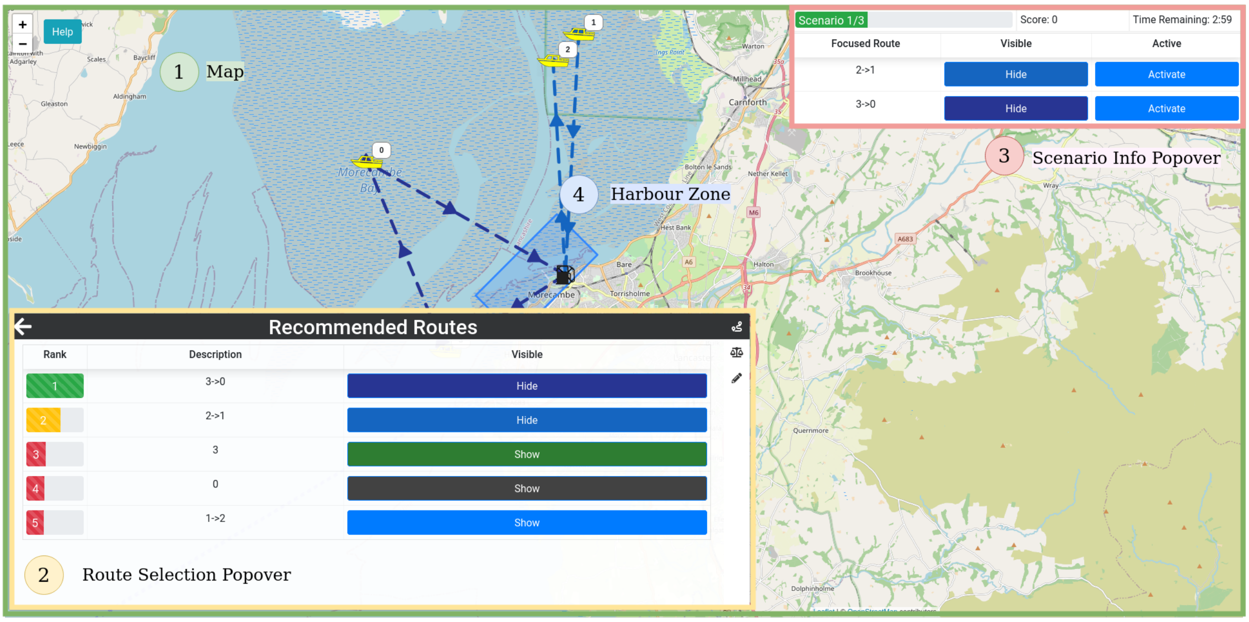


Figure 3: The user interface with no features enabled *colour print*

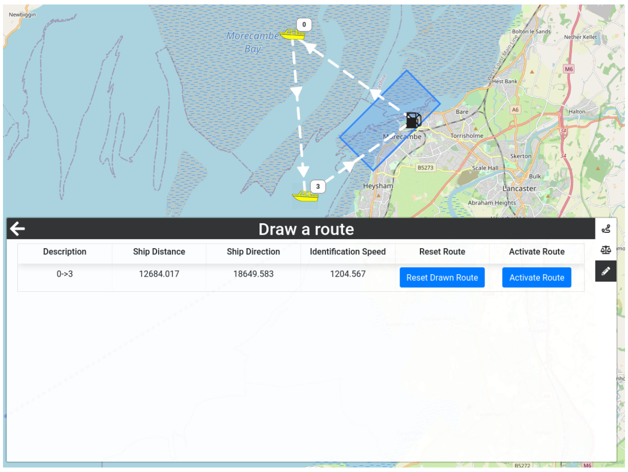


Figure 4: The interface for drawing a route *colour print*

the score for the current stage, and the time remaining. The recommended routes are dynamically updated until a route is activated, at which point the route selection popover is hidden.

If all routes recommended by the system are deemed unsuitable, the user can select the pencil icon to access the *draw a route* interface shown in Figure 4. Once this tab is open, a route can be drawn by clicking on ships in order. In this tab the user can view the criteria values of the drawn route, reset the drawn route, or activate the drawn route.

3.5. Experiment Design

In this section we outline the method for collecting data for analysis by PLS-SEM using our harbour management DSS. We describe how these experiment users are recruited in 5.1.

Users began the experiment by receiving a link to the *Static File Server* and credentials to authorise with the *User Server*. When authorised, the users were presented with the harbour management DSS. Each user was provided with a random selection of features enabled/disabled, recorded within the *User Database*. They were required to complete a tutorial tailored to the features enabled within their interface. Once they had completed the tutorial, users were given three five minute scenarios of increasing difficulty, presented as three stages to complete. To incentivise and measure performance for this task, a score is recorded in the *User Database*. The user is awarded one point for each identified ship, and deducted one point for each unidentified ship, that enter the harbour zone. The score was recorded, and reset to zero between stages. The users were incentivised to maximise their score across all three stages through a bonus payment, paid to the best performing harbour manager.

After all three stages were complete, users were presented with the questionnaire, comprising questions relating to *trust*, *satisfaction*, *transparency* and *cognitive load* (as shown in Table 3). The questionnaire used 7-point Likert scales [16], with responses ranging from

Figure 5: Routes with the explanation feature enabled *colour print*

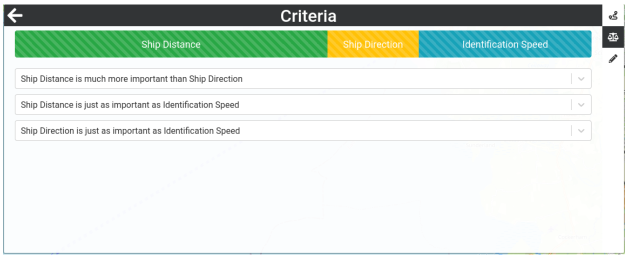


Figure 6: The interface for setting criteria preferences *colour print*

one (strongly disagree) to seven (strongly agree). The features and questionnaire answers for experiment users were then compiled for analyses.

3.6. *Features*

To investigate which features are useful for real-time decision support, we provided each experiment user with a set of features. In this section we will describe the features that can be enabled or disabled for different users.

3.6.1. *Explanation*

Explanation is the provision of reasoning describing how an output was reached. To provide explanation of the recommendations in the harbour management scenario we added coloured bar charts, which show how a solution performed over a specific criterion, to the route selection popover, as shown in Figure 5. A visual form of explanation was designed to quickly give decision makers an idea of how each criterion contributed towards the ranking of solutions. The size of the bar indicates how well a criteria value compares to the optimal value of the respective criterion. A bar is coloured green if it is greater than 75% of the optimal. A yellow bar for values between 50% and 75% and a red bar is used for values less than 50%. If the feature is disabled the user will see only the show/hide button, as shown in Figure 3.

3.6.2. *Preferences*

The preferences feature allows decision makers to state their criteria preferences in the form of pairwise comparisons. For instance, a decision maker could express that "Ship Distance is more important than Ship Direction". The interface for setting preferences is shown in Figure 6. This menu is accessed by clicking on the scales icon on the route selection popover. Here the user can set their preferences through a drop down menu for each pair. AHP is applied to transform these pairwise comparisons into criteria weightings. The bars at the top of the interface provide visual feedback of the resultant weightings. If this feature is disabled the criteria tab is hidden and criteria weights are fixed evenly.

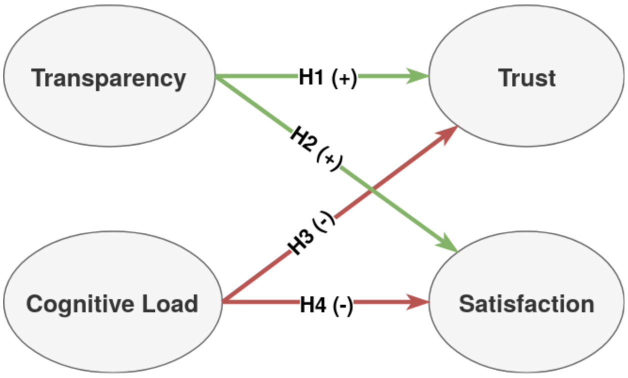


Figure 7: The basic theoretical framework *colour print*

3.6.3. *Dynamic Updates*

The dynamic updates feature refers to the system's capacity to make new recommendations whilst the drone is in motion. If this feature is enabled, after a route is activated the route selection popover will not be hidden, and the system will continue to recommend updates to the plan. These updates could be small adjustments, or a complete change of plan, depending on how the scenario unfolds. If this feature is disabled when a route is active, the route selection popover is hidden and no routes will be recommended until the drone returns to refuel.

4. Research model and hypotheses

4.1. *Theoretical Model*

Trust and satisfaction are two key measures of success for a DSS. The research model (Figure 7) shows how these measures are driven by two more practical targets, transparency and cognitive load. The model suggests that trust and satisfaction can both be improved by increasing the transparency of a system and lowering cognitive load.

4.1.1. Transparency

For our study, we define transparency as a measure of how well the user understands what actions are being performed. The understanding of how the recommendations are calculated, how an activated route will perform, and the current status of a task, are therefore all aspects of transparency. A system with greater transparency allows a decision maker to make more accurate judgements of the limits of a system. This means that transparency should improve a users' confidence in their decisions, and lead to greater trust in the system [44]. Understanding when to trust a recommendation creates a more competent human-computer team. This improvement in efficacy should create a higher user satisfaction with the system. Therefore we hypothesise:

Hypothesis 1. *Transparency increases trust.*

Hypothesis 2. *Transparency increases satisfaction.*

4.1.2. Cognitive Load

We define cognitive load as a measure of the amount of mental effort invested in operating a system. This mental effort detracts from the thought a user can put into selecting a solution. This gives the user less confidence in their decisions, undermining trust in the system. The added difficulty in correctly estimating the limits of the system also can cause lower user satisfaction. Therefore we hypothesise:

Hypothesis 3. *Cognitive load decreases trust.*

Hypothesis 4. *Cognitive load decreases satisfaction.*

4.2. Extended Research Model

Improving transparency can be specifically targeted by features of DSS. Architects of these systems should consider that all features are likely to increase cognitive load. Figure 8 shows how we hypothesise *explanation*, *dynamic updates* and *preferences* affect transparency and cognitive load.

4.2.1. Explanation

Explanation aims to provide reasoning describing how an output was reached. Our explanation feature gives this in the form of bar charts, shown in Figure 5, with detail given in Subsection 3.6.1. These charts show how the solutions perform over each criterion. This should give the user an understanding of how a ranking of solutions was reached, improving transparency [18]. This also gives the decision maker more information to consider; increasing cognitive load. We hypothesise:

Hypothesis 5. *Explanation increases transparency.*

Hypothesis 6. *Explanation increases cognitive load.*

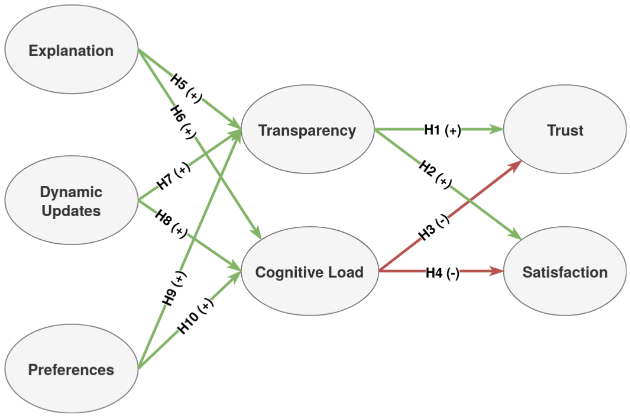


Figure 8: Research Model *colour print*

4.2.2. Dynamic Updates

Dynamic updates is the ability to provide updated rankings of routes whilst the drone is in flight. Details for this feature are given in Subsection 3.6.3. This feature should increase transparency, as the user is provided with more information regarding the ranking of routes. This extra information comes at the cost of increased cognitive load whilst a route is active. Therefore we hypothesise:

Hypothesis 7. *Updates increase transparency.*

Hypothesis 8. *Updates increase cognitive load.*

4.2.3. User Preferences

The user preferences feature allows decision makers to state their criteria preferences in the form of pairwise comparisons. The details for our implementation of this feature are given in Subsection 3.6.2 and the interface is shown in Figure 6. This feature gives the user a lever to tailor recommendations. Another benefit is it gives the decision makers insight into the criteria under consideration, and trade-offs made by the system, providing transparency [34]. At the same time, this creates an extra tab of input which must be read and understood, increasing cognitive load. We therefore hypothesise:

Hypothesis 9. *Preferences increase transparency.*

Hypothesis 10. *Preferences increase cognitive load.*

5. Methodology

5.1. Data Collection

To test our theoretical framework, we surveyed users of our harbour management system. The users were

Amazon Mechanical Turk (MTurk) [1] workers that attained our harbour management qualification in exchange for compensation at the rate of the UK minimum wage. MTurk is a crowdsourcing website for businesses (known as Requesters) to hire remotely located "crowdworkers" to perform discrete on-demand tasks. One issue was that these workers are not specifically trained for harbour management. To address this issue we created a specific harbour management qualification, allowing us to test user efficacy and understanding of the harbour management task, before allowing them into the study.

As a further test for workers, the questionnaire also included two repeated questions. Workers giving different answers to the same question were marked as inappropriate. A total of 58 workers completed the task. After eliminating inappropriate responses and system errors, a total of 43 usable responses were included for construct validation and hypothesis testing.

5.2. Results

To test our proposed research model we performed data analysis using Partial Least Squares (PLS). Figure 9 shows the results calculated using the python package, PSLPM version 0.5.5 with bootstrapping.

5.2.1. Reliability

To calculate internal consistency we used both Cronbach's alpha (Alpha) and composite reliability (CR). Table 2 shows that the CR values are above 0.7 [11] and the Alpha values are all above 0.65 [21], satisfying the standard requirements for internal consistency. We also show that all Average Variance Extracted (AVE) values were higher than 0.50, the suggested minimum. An AVE greater than 0.5 indicates that more than 50% of the variance of the measurement items can be accounted for by the constructs.

5.2.2. Construct Validity

Construct validity was assessed via convergent validity and discriminant validity. Convergent validity is shown to be acceptable in Table 3, with all item loadings greater than 0.50, and all items for each construct loading onto only one factor with an eigenvalue greater than 1.0. To evaluate discriminant validity we applied the Heterotrait-monotrait (HTMT) ratio of correlation. Henseler *et al.* [14] proposed HTMT, providing evidence for its superior performance by means of Monte Carlo simulation study, that showed that HTMT is able to achieve higher specificity and sensitivity rates (97% - 99%) compared with the Fornell-Lacker (20.82%). HTMT values close to 1 imply a lack of discriminant validity. Table 1 shows that all values are below the accepted threshold value of 0.9. This indicates discriminant validity among variables.

Table 1: HTMT ratios of correlation for constructs

Construct	TRAN	CL	TRUST
CL	0.892		
TRUST	0.872	0.619	
SAT	0.880	0.734	0.828

5.2.3. Structural model assessment

To assess the structural model we assessed both path coefficients and R^2. Both R^2 and path coefficients give us an indication of the model fit. Figure 9 shows the results. Transparency (TRAN) had a strong positive effect on trust (TRUST) and satisfaction (SAT). The path coefficients TRAN → TRUST and TRAN → SAT were both significant at the 0.01 level. Therefore Hypotheses 1 and 2 were supported. Cognitive Load (CL) was shown to have no effect on trust, with a small negative effect on satisfaction. The CL → SAT path coefficient fell short of the 0.05 level, therefore we failed to prove Hypotheses 3 and 4.

For the paths from features, contrary to our hypothesis, explanation showed a small negative effect on transparency. This path and the hypothesised path from explanation to transparency were not significant, therefore not supporting Hypotheses 5 and 6. The dynamic updates feature had a similar outcome, with neither Dynamic Updates → TRAN or Dynamic Updates → CL showing significance at $p < 0.05$. So hypotheses 7 and 8 were not supported. The hypothesised paths from preferences to transparency and preferences to cognitive load showed weak positive and negative effects respectively, significant at $p < 0.05$. This validates Hypotheses 9 and 10.

The R^2 for trust and satisfaction were both 0.79, showing that transparency and Cognitive Load provide a strong explanation for trust and satisfaction in a system. On the other hand, transparency and cognitive load had an R^2 value of 0.27 and 0.21 respectively, implying that these variables were largely driven by factors outside the scope of the study.

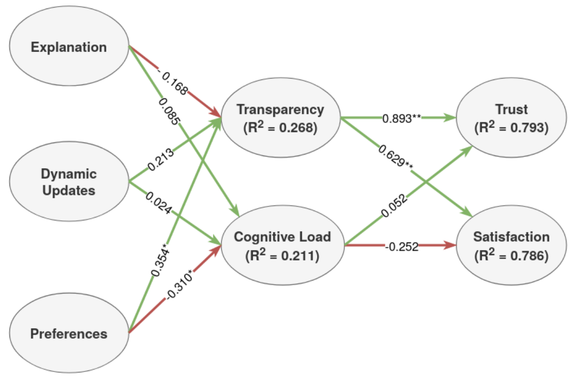
6. Discussion and Conclusion

PLS-SEM has been used extensively to investigate the antecedents of trust in electronic commerce [19, 7,

Table 2: Descriptive statistics and reliability indices for constructs

Construct	Item	Loading	AVE	CR	Alpha
TRAN	TRAN1	0.879	0.573	0.815	0.693
	TRAN2	0.774			
	TRAN3	0.442			
	TRAN4	0.851			
CL	CL1	0.866	0.758	0.926	0.879
	CL2	0.923			
	CL3	0.819			
TRUST	TRUST1	0.826	0.600	0.885	0.836
	TRUST2	0.831			
	TRUST3	0.795			
	TRUST4	0.772			
	TRUST5	0.631			
SAT	SAT1	0.832	0.676	0.873	0.802
	SAT2	0.882			
	SAT3	0.763			
	SAT4	0.808			

The questions corresponding to each item are given in Table 3



Note: * Significant at the 0.05, ** Significant at the 0.01, *** significant at the 0.001 level

Figure 9: Results of PLS analysis *colour print*

20] and the adoption of emerging technologies [24, 23, 26]. In these papers, the authors survey participants that have used pre-existing systems: online retailers, in the case of electronic commerce; or mobile banking services, as an example of an emerging technology. These systems are assessed for features and the effect of these features is measured on the research model. Our study differs from this standard PLS-SEM approach as we assess participants engaging with a purpose-built system, controlling which features are enabled.

A past study by Kawamoto *et al.* in clinical decision support has investigated validity of decision support features by assessing the improvement in clinical practice. This assessment was enabled through the measurement of patient outcomes or process measures [17]. The study summarised 82 relevant comparisons of which 71 compared a clinical DSS with a control group (control-system comparisons) and 11 directly compared a system with the same system plus extra features (system-system comparisons). Another study conducted a similar assessment with 162 randomised control trials [35].

These studies collate data from a multitude of papers, assessing features for purpose-built clinical DSS. This type of study is enabled in clinical decision support by the presence of clear success criteria (patient outcomes), to compare against the presence of decision support features. In lieu of this, our study instead assesses the effect of the presence of features on the constructs *trust*, *satisfaction*, *transparency* and *cognitive load* through the application of PLS-SEM.

The study by Kawamoto *et al.* investigated 15 decision support features. These were mostly specific to the clinical setting, but there was one feature that had significant overlap with our study, justification of decision support via provision of reasoning. We view this as a clinical domain specific implementation of explanation. The study found that systems with this feature had a 12% uplift in success rate, this corresponds with Hypothesis 5. Unfortunately we did not find evidence to support this hypothesis. Instead our implementation of explanation caused reduced transparency and increased cognitive load, therefore reducing trust and satisfaction. This could have been caused by our specific implementation of explanation for route selection.

6.1. Research Findings

The empirical results suggest that the transparency of a system positively impacted the trust and satisfaction a user associated with a DSS. This is consistent with previous research, finding that transparency of a leader significantly affected trust [28]. While different, human teams share many similarities with human-computer teams. In both contexts, the leader/DSS prescribes a plan of action, with trust and transparency being significant factors shaping the outcomes of the interaction. Furthermore, a large portion of the variance within trust and satisfaction can be explained through the constructs of transparency and cognitive load. This validates the argument that architects should target transparency as an imperative aspect of real-time decision support.

The study also provides evidence that the ability to set user preferences improve transparency, indirectly improving trust and satisfaction. Our implementation of preferences was also found to reduce the cognitive load induced by a system. This feature provided necessary insight into the criteria considered for recommending routes. We hypothesise that the presence of explanation without preferences induced additional cognitive load, as users did not have adequate information to understand the system reasoning. This suggests that the ability to set preferences provides the user with a better understanding of the system and recommendations, enabling them to follow the reasoning provided by an explanation. Therefore reducing cognitive load.

6.2. Theoretical and Practical Contributions

Our primary contribution is a methodology for the assessment of DSS and their features in the absence of clear success criteria. We also contribute a theoretical framework for the antecedents of trust in a real-time DSS. This model implies that transparency is a strong predictor of trust and satisfaction in a system.

From a practical standpoint, our research highlights several features that an architect should consider when building a real-time DSS. The findings imply that the implementation of these features should be carefully considered. For instance, we found that our implementation of explanation had an effect contrary to our hypotheses. This suggests that architects should be mindful that reasoning provided is concise and clear, because it is possible for an explanation to reduce transparency.

6.3. Limitations and directions for future research

Future work is needed to assess the generalisability of this model to other VRP and UAVTAP applications. We find that preferences improve transparency, but it is worth noting that our results are specific to the implementations within the harbour management system. Another limitation is that Low R^2 for transparency and cognitive load imply they were largely driven by factors outside the scope of the study. Therefore a future study taking into account more features could provide an explanation of the drivers of these constructs.

We failed to achieve statistical significance regarding the effect of explanation and dynamic updates on cognitive load and transparency. This is likely caused by the small impact of these two features. It would therefore be beneficial for a study to be conducted at a larger scale, to provide conclusive results of the effect of these features.

Acknowledgements. Dominic Duxbury is supported by an EPSRC iCASE award in association with BAE Systems.

References

[1] Amazon mechanical turk. https://www.mturk.com/.

[2] Leaflet. https://leafletjs.com/.

[3] Lorenzo Abbatecola, MariPiaa Fanti, Agostino Marcello Mangini, and Walter Ukovich. A Decision Support Approach for Postal Delivery and Waste Collection Services. *IEEE Transactions on Automation Science and Engineering*, 13(4):1458–1470, oct 2016.

[4] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. Beyond Accuracy: The Role of Mental Models in Human-AI Team Performance. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 7(1):19, 2019.

[5] L. Basile and Livia D'Apuzzo. Transitive matrices, strict preference order and ordinal evaluation operators. *Soft Computing*, 10(10):933–940, 2006.

[6] Sara Beheshtifar and Abbas Alimoahmmadi. A multiobjective optimization approach for location-allocation of clinics. *International Transactions in Operational Research*, 22(2):313–328, mar 2015.

[7] Chao Min Chiu, Meng Hsiang Hsu, Hsiangchu Lai, and Chun Ming Chang. Re-examining the influence of trust on online repeat purchase intention: The moderating role of habit and its antecedents. In *Decision Support Systems*, volume 53, pages 835–845. North-Holland, nov 2012.

[8] Bruno N. Coelho, Vitor N. Coelho, Igor M. Coelho, Luiz S. Ochi, Roozbeh Haghnazar K., Demetrius Zuidema, Milton S.F. Lima, and Adilson R. da Costa. A multi-objective green UAV routing problem. *Computers and Operations Research*, 88:306–315, dec 2017.

[9] Walton Pereira Coutinho, Maria Battarra, and Jörg Fliege. The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Computers and Industrial Engineering*, 120(June 2017):116–128, 2018.

[10] Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. feb 2017.

[11] Claes Fornell and David F. Larcker. Evaluating Structural Equation Models with Unobservable Variables and Measurement Error. *Journal of Marketing Research*, 18(1):39, feb 1981.

[12] Marianne Harbo Frederiksen and Mette Præst Knudsen. Drones for offshore and maritime missions: Opportunities and barriers. *SDU Centre for Integrative Innovation Management*, (April), 2018.

[13] Joseph F. Hair, Jeffrey J. Risher, Marko Sarstedt, and Christian M. Ringle. When to use and how to report the results of PLS-SEM. *European Business Review*, 31(1):2–24, 2019.

[14] Jörg Henseler, Christian M Ringle, and Marko Sarstedt. A new criterion for assessing discriminant validity in variance-based structural equation modeling. *Journal of the Academy of Marketing Science*, 43(1):115–135, 2014.

[15] Meng Hsiang Hsu, Chun Ming Chang, Kuo Kuang Chu, and Yi Jung Lee. Determinants of repurchase intention in online group-buying: The perspectives of DeLone and McLean is success model and trust. *Computers in Human Behavior*, 36:234–245, 2014.

[16] Ankur Joshi, Saket Kale, Satish Chandel, and D. Pal. Likert Scale: Explored and Explained. *British Journal of Applied Science and Technology*, 7(4):396–403, jan 2015.

[17] Kensaku Kawamoto, Caitlin A. Houlihan, E. Andrew Balas, and David F. Lobach. Improving clinical practice using clinical decision support systems: A systematic review of trials to identify features critical to success. *British Medical Journal*, 330(7494):765–768, 2005.

[18] Buomsoo Kim, Jinsoo Park, and Jihae Suh. Transparency and

Table 3: Proposed Measurement Items for Constructs

Constructs	Question	Loading
TRAN1	I understood why the recommended routes were provided over alternatives.	0.879
TRAN2	The recommended routes were clearly justified.	0.774
TRAN3	I was aware of trade-offs when I was choosing routes.	0.442
TRAN4	I felt I was kept up to date with events in the scenario.	0.851
		Eigenvalue : 2.18
CL1	The interface of the exercise was complex.	0.866
CL2	The way the interface presented information was distracting.	0.923
CL3	It was difficult to find relevant information for selecting routes.	0.819
		Eigenvalue : 2.42
TRUST1	I felt that I could trust the application	0.826
TRUST2	The recommended routes were good	0.831
TRUST3	The recommended routes helped me to make decisions.	0.795
TRUST4	The recommended routes often turned out as expected.	0.772
TRUST5	I believe the application has been designed to enhance my decision making.	0.631
		Eigenvalue : 3.05
SAT1	I feel satisfied with the overall experience of using the application.	0.832
SAT2	I enjoyed using the application.	0.882
SAT3	I think route recommendations are a good idea.	0.763
SAT4	I feel good about the decisions I made with the support of the application.	0.808
		Eigenvalue : 2.55

accountability in AI decision support: Explaining and visualizing convolutional neural networks for text information. *Decision Support Systems*, 134:113302, jul 2020.

[19] Dan J. Kim, Donald L. Ferrin, and H. Raghav Rao. A trust-based consumer decision-making model in electronic commerce: The role of trust, perceived risk, and their antecedents. *Decision Support Systems*, 44(2):544–564, jan 2008.

[20] Sanghyun Kim and Hyunsun Park. Effects of various characteristics of social commerce (s-commerce) on consumers' trust and trust performance. *International Journal of Information Management*, 33(2):318–332, 2013.

[21] Jae Nam Lee and Young Gul Kim. Effect of partnership quality on IS outsourcing success: Conceptual framework and empirical validation. *Journal of Management Information Systems*, 15(4):29–61, 1998.

[22] John D. Lee and Katrina A. See. Trust in automation: Designing for appropriate reliance, jan 2004.

[23] Francisco Liébana-Cabanillas, Juan Sánchez-Fernández, and Francisco Muñoz-Leiva. Antecedents of the adoption of the new mobile payment systems: The moderating effect of age. *Computers in Human Behavior*, 35:464–478, 2014.

[24] Xin Luo, Han Li, Jie Zhang, and J. P. Shim. Examining multidimensional trust and multi-faceted risk in initial acceptance of emerging technologies: An empirical study of mobile banking services. *Decision Support Systems*, 49(2):222–234, 2010.

[25] Giusy Macrina, Luigi Di Puglia Pugliese, Francesca Guerriero, and Gilbert Laporte. Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies*, 120:102762, nov 2020.

[26] Azizbek Marakhimov and Jaehun Joo. Consumer adaptation and infusion of wearable devices for healthcare. *Computers in Human Behavior*, 76:135–148, 2017.

[27] R T Marler and J S Arora. Survey of multi-objective optimization methods for engineering, 2004.

[28] Steven M Norman, Bruce J. Avolio, and Fred Luthans. The impact of positivity and transparency on trust in leaders and their perceived effectiveness. *Leadership Quarterly*, 21(3):350–364, 2010.

[29] Vanessa Pirotta, Alana Grech, Ian D Jonsen, William F Laurance, and Robert G Harcourt. Consequences of global shipping traffic for marine giants, feb 2019.

[30] Cristian Ramirez-Atencia, Gema Bello-Orgaz, Maria D. R-Moreno, and David Camacho. A simple CSP-based model for Unmanned Air Vehicle Mission Planning. *INISTA 2014 - IEEE International Symposium on Innovations in Intelligent Systems and Applications, Proceedings*, pages 146–153, 2014.

[31] Cristian Ramirez-Atencia, Gema Bello-Orgaz, María D. R-Moreno, and David Camacho. Solving complex multi-UAV mission planning problems using multi-objective genetic algorithms. *Soft Computing*, 21(17):4883–4900, 2017.

[32] Cristian Ramirez-Atencia, Sanaz Mostaghim, and David Camacho. A knee point based evolutionary multi-objective optimization for mission planning problems. *GECCO 2017 - Proceedings of the 2017 Genetic and Evolutionary Computation Conference*, pages 1216–1223, 2017.

[33] Cristian Ramirez-Atencia, Victor Rodriguez-Fernandez, and David Camacho. A revision on multi-criteria decision making methods for multi-UAV mission planning support. *Expert Systems with Applications*, 160:113708, 2020.

[34] P. Reichert, M. Borsuk, M. Hostmann, S. Schweizer, C. Spörri, K. Tockner, and B. Truffer. Concepts of decision support for river rehabilitation. *Environmental Modelling and Software*, 22(2):188–201, feb 2007.

[35] Pavel S. Roshanov, Natasha Fernandes, Jeff M. Wilczynski, Brian J. Hemens, John J. You, Steven M. Handler, Robby Nieuwlaat, Nathan M. Souza, Joseph Beyene, Harriette G.C.

Van Spall, Amit X. Garg, and R. Brian Haynes. Features of effective computerised clinical decision support systems: Meta-regression of 162 randomised trials. *BMJ (Online)*, 346(7899):1–12, 2013.

[36] R. W. Saaty. The analytic hierarchy process-what it is and how it is used. *Mathematical Modelling*, 9(3-5):161–176, 1987.

[37] Thomas L. Saaty. Decision-making with the AHP: Why is the principal eigenvector necessary. *European Journal of Operational Research*, 145(1):85–91, 2003.

[38] Luís Santos, João Coutinho-Rodrigues, and Carlos Henggeler Antunes. A web spatial decision support system for vehicle routing using Google Maps. *Decision Support Systems*, 51(1):1–9, 2011.

[39] Anthony Sardain, Erik Sardain, and Brian Leung. Global forecasts of shipping traffic and biological invasions to 2050. *Nature Sustainability*, 2(4):274–282, apr 2019.

[40] Reza Sarraf and Michael P. McGuire. Integration and Comparison of Multi-Criteria Decision Making Methods in Safe Route Planner. *Expert Systems with Applications*, 154:113399, 2020.

[41] Dragan Savic. Single-objective vs Multiobjective Optimisation for Integrated Decision Support. *Proceedings of the First Biennial Meeting of the International Environmental Modelling and Software Society*, pages 7–12, jul 2002.

[42] David F. Steiner, Robert Macdonald, Yun Liu, Peter Truszkowski, Jason D. Hipp, Christopher Gammage, Florence Thng, Lily Peng, and Martin C. Stumpe. Impact of Deep Learning Assistance on the Histopathologic Review of Lymph Nodes for Metastatic Breast Cancer. *American Journal of Surgical Pathology*, 42(12):1636–1646, dec 2018.

[43] Richard Tomsett, Alun Preece, Dave Braines, Federico Cerutti, Supriyo Chakraborty, Mani Srivastava, Gavin Pearson, and Lance Kaplan. Rapid Trust Calibration through Interpretable. *Patterns*, 1(4):100049, 2020.

[44] Carl Westin, Clark Borst, and Brian Hilburn. Automation Transparency and Personalized Decision Support: Air Traffic Controller Interaction with a Resolution Advisory System. *IFAC-PapersOnLine*, 49(19):201–206, jan 2016.

[45] Herman Wold. Model Construction and Evaluation When Theoretical Knowledge Is Scarce. In *Evaluation of Econometric Models*, pages 47–74. Elsevier, jan 1980.