Advances in Human Factors of Transportation, Vol. 224, 2026, 149–159
https://doi.org/10.54941/ahfe1007842

Explainable AI for Emergency Landing Decisions: A Comparative Study of Learning Classifier Systems and Neural Networks

Evelyn Yamilet Quintero Guzman[1], Jakob Suchan[1], and Boris Djartov[2]

[1]Constructor University Bremen, Bremen, 28759, Germany
[2]German Aerospace Center (DLR), Braunschweig, 38108, Germany

ABSTRACT

During mid-flight emergencies, pilots must rapidly assess multiple operational and environmental factors to select a safe alternate landing airport. This Dynamic Alternate Airport Selection (DAAS) process requires fast, reliable decision-making under high cognitive load. Although established cockpit procedures such as those in the QRH provide essential guidance, additional data-driven support tools could further help pilots manage complex information under time pressure. Different Artificial Intelligence (AI) methods offer promising opportunities in this regard, however, for aviation applications, it is necessary that the applied methods are both, accurate and transparent, and that their decision logic is explainable to pilots. Accordingly, this paper investigates which AI methods are most suitable for modelling pilot behaviour in emergency airport-selection tasks while maintaining a high degree of explainability to foster trust in the system. Using a dataset derived from an online survey of professional pilots capturing their preferences across emergency diversion scenarios, and expanded through structured data augmentation to generate 7,140 labelled decision scenarios, the study evaluates two variants of interpretable Learning Classifier Systems (LCS), using Hyperellipsoid and Hyperrectangle conditions, whose decision-making is encoded in explicit, human-readable IF–THEN rules that enable direct inspection of how inputs lead to decisions. These models were contrasted with a more modern, non-interpretable baseline: a Feedforward Neural Network (FNN). The models were designed for single-instance classification using a scoring framework. The scores were used to label the augmented dataset by combining the scenario scores with the Euclidean distance between the original decision scenarios and the unique airport combinations generated within each scenario. Model performance was evaluated using accuracy and interpretability considerations, key factors for integration into cockpit decision-support systems. The Hyperellipsoid LCS achieved the highest accuracy (86.34%), demonstrating strong adaptation to multidimensional feature interactions. The Hyperrectangular LCS offered greater rule-level transparency but lower accuracy (78.33%), while the FNN achieved intermediate accuracy (82.20%) with limited inherent interpretability. Results show that the Hyperellipsoid LCS provides the best overall balance between predictive performance and transparency, outperforming the Hyperrectangle LCS and the FNN. These findings indicate that ellipsoid-based LCS models offer a promising foundation for trustworthy AI components in future pilot decision-support systems.

Keywords: Human-centered artificial intelligence, Aviation decision support, Emergency decision-making, Explainable AI in aviation, Dynamic alternate airport selection

Received February 7, 2026; Revised April 5, 2026; Accepted April 19, 2026; Available online July 20, 2026

INTRODUCTION

The aviation industry has increasingly explored Artificial Intelligence (AI) and Machine Learning (ML) to support critical decision-making and enhance operational safety (Mian et al., 2024). AI has shown promise for enhancing decision support in high-stakes scenarios, where traditional procedures and statistical methods may be insufficient. Machine Learning techniques such as Deep Learning (DL) can process large volumes of operational data, detect patterns, and provide real-time insights that improve situational awareness and operational safety (Demir, Moslem and Duleba, 2024). For instance, AI enabled systems can assist in anticipating potential hazards, optimizing flight operations, and supporting pilot decisions during abnormal or emergency situations, where timely and reliable responses are critical. However, in aviation contexts, such systems must also be explainable, allowing pilots and certifying authorities to understand and trust how recommendations are generated. One concrete application of AI in aviation is the Intelligent Pilot Advisory System (IPAS), currently under development at the German Aerospace Center (DLR) (Würfel et al., 2023). IPAS is a human-centered decision support system designed to provide pilots with recommendations by integrating AI algorithms with real-time flight and environmental data. The central design principle of IPAS focuses on interpretability, transparency, and explainability, enabling pilots to understand how AI recommendations are derived and thereby fostering trust in the system. Such characteristics are important in emergency situations, where the consequences of errors can be severe. Within IPAS, one critical scenario is the Dynamic Alternate Airport Selection (DAAS), a time-sensitive decision-making task in which pilots must evaluate multiple variables to select a safe diversion airport. Intelligent decision support systems can help reduce the cognitive workload in such scenarios by analyzing complex data and providing pilots with interpretable, actionable guidance.

This paper focuses on evaluating AI models for simulating pilot decision-making in emergency landing situations, focusing on interpretability. Specifically, it compares hyperellipsoid and hyperrectangle Learning Classifier Systems (LCS) with a Feedforward Neural Network (FNN) baseline, examining how these models balance predictive performance with transparency. By investigating the human-centered aspects of AI in emergency decision-making, this study contributes to the ongoing development of IPAS and the broader goal of enhancing trustworthy, effective decision support in aviation.

DATA SET

The data was collected through an anonymous online survey with forty-six participants recruited via the Institute of Flight Guidance mailing list (Djartov, Papenfuß and Wies, 2025; Keul, 2023). The survey included the following sections:

1. A closed-ended question regarding whether the pilot has active type approval for the Airbus A320 family as criteria for inclusion.

2. A set of questions to gather information about the social and demographic characteristics of the participants.
3. Questions to help evaluate a pilot's proficiency and experience within the aviation field, such as total flight hours, flight hours last year, flight hours on the A320 type approval, and pilot rank.
4. Twelve decision-making scenarios, each with three destination airports with different landing conditions where the reason for selecting the airport is not specified.

The aircraft used for all scenarios was a fully operational Airbus A320 at full capacity, using the maximum landing weight. The scenarios were described as having normal weather conditions according to ISA (International Standard Atmosphere), without wind, until reaching the point of diversion to the alternate location. Participants ranked airports in each scenario based on their preferred alternate landing destination mid-flight. The scenarios represented extreme cases to identify which characteristics pilots consider most important when selecting a new airport. The following factors were provided as decision criteria:

1. Distance alternate to original airport: distance from the planned airport to the alternate airport.
2. Related Landing Performance: single value summarizing runway conditions and expected aircraft performance.
3. Margin landing distance: gap between runway length and aircraft's braking distance.
4. Crosswind: wind perpendicular to the flight direction.
5. Tailwind: wind aligned with the flight direction.
6. Fuel remaining: amount of fuel left before landing.
7. Number of runways at the airport.

Resulting data showed the number of participants who had ranked each airport as rank 1, rank 2, and rank 3, where the rank indicates the pilot's preferred alternate airport in each scenario (Djartov, Papenfuß and Wies, 2025).

DATA PREPROCESSING AND AUGMENTATION

To enable model training, the original dataset was transformed into a single-instance classification format, with the target variable representing the pilot-selected airport in each scenario. A custom objective function combined the three airport rankings into a single score, giving the highest weight to Rank 1 while allowing Ranks 2 and 3 to contribute proportionally less. This approach increased separation between similarly ranked airports, reducing ties and providing a more precise representation of pilot preferences. The scoring function is defined as:

$$SC(a_i) = R_1 + \sqrt{R_2(a_i)} + \sqrt[3]{R_3(a_i)}$$

where a_i represents the airport within scenario i. Feature correlations were then analysed using Pearson correlation with Fisher's z-transform to assess the impact of operational variables on the new scores. Results indicated that factors such as Fuel Remaining and Margin in Landing Distance were more influential than previously observed, highlighting their importance for pilot decision-making.

Given the limited size of the original dataset (12 scenarios), data augmentation was performed to expand the training set. Features were first normalized using Min-Max scaling to ensure comparability across variables with different units:

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

New scenarios were generated by creating unique combinations of airports. Each new scenario (N) was compared to the original scenarios (O) using Euclidean distances, stored in matrix form for efficiency. For each matrix, the three smallest Euclidean distances for each airport in N were identified and matched to an airport in O ensuring all airports were paired. After selecting these three distances, their average value was calculated and denominated d_{min}, resulting in 12 values per new scenario. Subsequently, the new scenario with the smallest d_{min} value was matched to the original scenario, considering these scenarios the closest. Scores for the new scenario were assigned using a weighted inverse-distance approach, ensuring that airports closer to highly ranked originals received higher scores:

$$w_{i,j} = \frac{1}{\left(\frac{d\left(O_{i,j}, N_{i,j}\right)}{d_{min}}\right)^2}$$

where $d(O_{i,j}, N_{i,j})$ refers to the Euclidean distance between the selected airports j within scenario i. The final score for each airport in a new scenario was calculated as a weighted combination of the scores from the matched original scenario:

$$SC_i = \frac{\sum_{j=1}^{3} w_{i,j} \cdot SC_j}{\sum_{j=1}^{3} w_{i,j}}$$

This process created 7,140 labelled scenarios, each representing a single airport selection as a class (0, 1, or 2). The resulting dataset provides both sufficient size and structure for training AI models to simulate pilot decision-making in emergency landing situations, while preserving the relative importance of scenario features and maintaining interpretability for evaluation purposes.

MODEL CONFIGURATION

We evaluate three models for predicting pilot decisions in emergency airport selection: two variants of LCS and a FNN. The three model configurations represent distinct trade-offs between flexibility, interpretability, and predictive performance. The FNN provides strong nonlinear modelling capacity but operates as a black-box model. The hyperellipsoid LCS offers flexible and expressive decision regions with moderate interpretability, while the hyperrectangle LCS prioritizes transparency and rule simplicity at the expense of representational power. These complementary characteristics make the models well suited for comparative analysis in safety-critical aviation decision-support contexts.

Feed Forward Neural Network

The Feedforward Neural Network (FNN) follows a standard multilayer architecture in which information propagates unidirectionally from the input layer through hidden layers to the output layer (Sazlı, 2006). Each neuron computes a weighted sum of its inputs, adds a bias, and applies a non-linear activation function (Svozil, Kvasnicka and Pospichal, 1997). The hidden layers transform the input features into higher-level representations, capturing non-linear interactions between features, while the output layer applies the SoftMax function to generate probability distributions over the three alternative airports:

$$\widehat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{m} e^{z_j}}, \quad \text{for } i = 1, 2, 3$$

where z_i is the pre-activation output of neuron i, and m = 3 is the number of classes. The specific architecture used is as follows:

1. Input layer: 21 neurons (scenario features)
2. Hidden layer 1: 64 neurons → 128 neurons
3. Hidden layer 2: 128 neurons → 32 neurons
4. Output layer: 32 → 3 neurons (classes)

Rectified Linear Unit (ReLU) activation functions were applied to the hidden layers:

$$\mathrm{ReLU}(x) = \max(0, x)$$

The model was trained using the cross-entropy loss function for multi-class classification, optimized via the Adam optimizer. To prevent overfitting, early stopping was applied based on validation loss, and a ReduceLROnPlateau scheduler dynamically adjusted the learning rate. This design balances model flexibility for capturing complex relationships and interpretability relative to more complex architectures like recurrent or convolutional networks (Bebis and Georgiopoulos, 1994).

Learning Classifier Systems (LCS)

LCS is a population-based, rule-learning model that combines reinforcement learning with evolutionary computation to generate interpretable IF–THEN rules that drive the learning process of the model (Bull, 2004; Lanzi, Stolzmann and Wilson, 2000; Preen and Pätzel, 2024). Each rule, referred to as a classifier, maps input conditions to actions or predictions and evolves over time based on performance feedback. We employ XCSF (Preen and Pätzel, 2024), an extended Michigan-style LCS. Unlike traditional reinforcement-learning LCS formulations, XCSF is well suited for supervised learning tasks, as it directly approximates target outputs using regression while preserving an evolving population of interpretable rules. XCSF emphasizes prediction accuracy as the primary fitness criterion, encouraging the formation of classifiers that accurately represent the underlying decision space rather than those that simply maximize reward. Each classifier in XCSF consists of three main components:

1. **Condition** (*cl.C)*: region of the input space for which the rule is applicable.
2. **Action** (*cl.A)*: predicted class or decision outcome.
3. **Prediction function** (*cl.P)*: expected output given the input state.

Both LCS variants operate on the same input representation: a 21-dimensional feature vector describing the emergency scenario. The task is formulated as a supervised multi-class classification problem with three one-hot encoded output classes corresponding to the candidate diversion airports. During inference, classifiers whose conditions match the current input form a match set. From this set, classifiers are selected based on their fitness values to propose an action. The system then updates classifier parameters using supervised feedback derived from prediction error, allowing both rule conditions and predictions to adapt over time.

The first configuration uses the **Hyperellipsoid condition**, represented mathematically as:

$$C = (m, \Sigma) = \left(\begin{pmatrix} m_1 \\ m_2 \\ \vdots m_n \end{pmatrix}, \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \sigma_n \end{pmatrix} \right)$$

where m is the center of the ellipsoid, and Σ stands for the Mahalanobis distance, which is the fully weighted Euclidean distance metric of the ellipsoid. Each classifier defines a smooth, multidimensional decision region that can be rotated and stretched to align with the structure of the data. The condition is represented by a center vector and a shape matrix:

$$C = (\mu, A)$$

where μ denotes the center of the ellipsoid and A is a positive-definite matrix governing its orientation and scale. A classifier matches an input vector x if:

$$(x - \mu)^{\top} A^{-1} (x - \mu) \leq 1$$

This formulation allows classifiers to capture oblique and curved decision boundaries, enabling flexible generalization across correlated features. As a result, hyperellipsoid conditions are well suited for complex, high-dimensional decision spaces, though they are difficult to interpret due to their geometric complexity.

The second configuration uses the **Hyperrectangle center-spread condition** component. This condition is determined using the following formula:

$$C = (l, u) = \left(\begin{pmatrix} l_1 \\ l_2 \\ \vdots l_n \end{pmatrix}, \begin{pmatrix} u_1 \\ u_2 \\ \vdots u_n \end{pmatrix} \right)$$

where l represents the lower bound in the dimension i, meaning the minimum value of the input x_i for which the condition is satisfied, and u represents the upper bound vector (Butz, Lanzi and Wilson, 2008). Each classifier uses a condition based on hyperrectangle Compressed Sparse Representation (CSR), rather than a hyperellipsoid. This condition model simplifies input matching by defining axis-aligned bounds per feature dimension. An input vector x matches a classifier if the following condition is satisfied:

$$l_i \leq x_i \leq u_i \quad \forall \mathrm{i}$$

This representation constrains decision boundaries to be orthogonal to the feature axes, resulting in simpler, more localized rules. While this limits the model's ability to capture complex feature interactions, it significantly improves interpretability, as each rule can be directly translated into explicit threshold-based IF–THEN statements.

EXPERIMENTAL DESIGN AND EVALUATION

All experiments were conducted using Python 3.11.9. The LCS variants were implemented with the XCSF framework (Preen and Pätzel, 2024), while the FNN was implemented using the PyTorch deep learning library (Paszke et al., 2017).

Experimental Design

The experimental design aimed to enable a fair and statistically robust comparison between the three modelling approaches under identical data conditions. The dataset of 7,140 labelled emergency airport selection scenarios was randomly split into training (70%), validation (10%), and test sets (20%), where the validation set was used for hyperparameter tuning and monitoring generalization during training, while the test set was reserved for final evaluation on unseen data. To reduce sensitivity to data partitioning and improve result reliability, five-fold cross-validation was applied. In each iteration, four folds were used for training and one fold for validation.

Training

Model training was monitored using validation loss to detect convergence behaviour and potential overfitting. Early stopping was applied uniformly across models, halting training when validation performance no longer improved and restoring the best-performing model state. Hyperparameter optimization was conducted using Bayesian optimization with Optuna framework, with the validation set serving as the objective evaluation metric (Nogueira, 2014). This strategy enabled efficient exploration of the hyperparameter space while minimizing unnecessary training runs.

Evaluation

Model performance was evaluated using multiple complementary metrics to capture both predictive accuracy and class-level behaviour. Accuracy was used as the primary performance metric to assess the proportion of correctly classified scenarios, defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where the numerator represents all correct predictions, while the denominator represents the total number of predictions made. The three models show clear differences in accuracy and stability (see Figure 1). The hyperellipsoid-based LCS performs best, achieving the highest median accuracy (≈86.3%) with low variance, indicating strong reliability and generalization across runs. The hyperrectangle-based LCS yields the lowest accuracy (≈78.3%) and greater variability, reflecting limited representational capacity despite its high interpretability. The FNN attains intermediate accuracy (≈82.2%) with stable performance, offering greater modelling flexibility than hyperrectangles but at the cost of reduced interpretability compared to rule-based approaches.

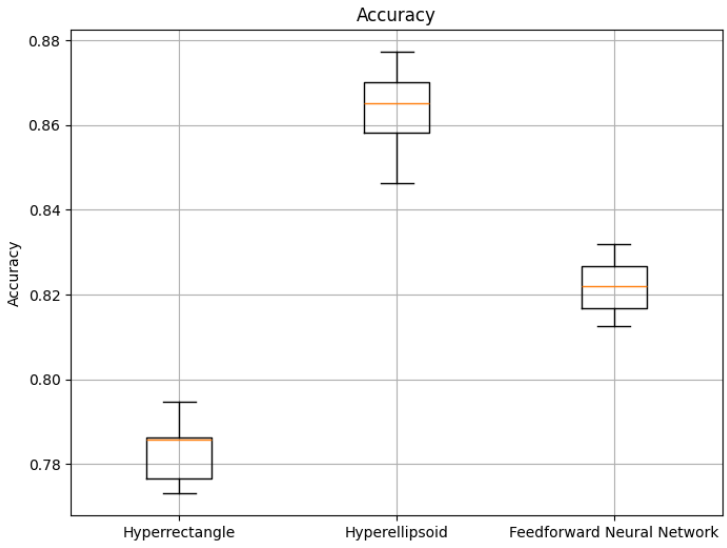

Figure 1: Accuracy of the models.

Furthermore, confusion matrices were analysed to identify systematic misclassification patterns and to evaluate how reliably each model distinguished between the three alternative airport choices, which is particularly relevant in safety-critical decision-support contexts. The results, summarized in Table 1, show that the hyperellipsoid-based LCS achieves the most balanced performance across all three classes, with accuracy differences of approximately 1%, indicating high stability and reliability for the DASS problem. The hyperrectangle-based LCS performs adequately on class 0 but shows notably lower accuracy for class 2, revealing a class imbalance that could negatively affect safety-critical decisions. The FNN also classifies class 0 effectively but exhibits reduced accuracy for classes 1 and 2, further highlighting the superior consistency of the hyperellipsoid-based LCS across decision scenarios.

Table 1: Percentage of correctly labelled scenarios per class.

Model	Class 0	Class 1	Class 2
Hyperellipsoid	86.94	87.42	85.12
Hyperrectangle	83.58	82.65	75.77
FNN	85.69	81.72	80.82

To further understand model behaviour, permutation feature importance was applied to all architectures. Results showed consistent patterns, with landing performance among the most influential predictors, confirming its central role in emergency airport selection. The hyperellipsoidal LCS distributed importance more evenly across features, reflecting its ability to capture multidimensional interactions. In contrast, the hyperrectangle-based LCS emphasized a smaller subset of features, consistent with its localized, rule-based structure, while the FNN highlighted dominant structural features while retaining moderate sensitivity to secondary variables through its nonlinear transformations.

Overall, the results demonstrate clear trade-offs between predictive accuracy, robustness, and interpretability. The hyperellipsoidal LCS offers the best balance between performance and stability, the hyperrectangle-based LCS provides greater transparency at the cost of accuracy, and the FNN delivers flexible modelling capacity with limited interpretability. These findings underscore the importance of aligning model selection with domain-specific requirements, particularly in safety-critical, human-centered aviation decision-support applications.

CONCLUSION

This paper examined the use of explainable AI models for supporting emergency landing decisions within the Dynamic Alternate Airport Selection (DASS) problem, a time-critical and safety-sensitive task in aviation. As part of the broader Intelligent Pilot Advisory System (IPAS), three modelling approaches were evaluated: an LCS with hyperellipsoidal conditions, an LCS

with hyperrectangular conditions, and a FNN. The models were compared using simulated emergency scenarios, with evaluation focused on predictive accuracy, robustness, and explainability, key requirements for human-centered decision-support systems. The results demonstrate clear trade-offs between performance and interpretability. Among the evaluated models, the hyperellipsoid-based LCS achieved the best overall classification performance and the most consistent generalization across airport alternatives. While it is less simple and immediately interpretable than the hyperrectangle-based LCS, its decision structure remains substantially more transparent than that of a FNN, offering a balance between predictive performance and explainability for safety-critical decision support.

Overall, the findings indicate that the hyperellipsoidal LCS provides the most effective balance between predictive capability and practical applicability for emergency landing decision support. While not fully transparent, its robustness and balanced feature sensitivity make it a strong candidate for integration into IPAS, particularly when complemented by targeted explanation mechanisms. This study underscores the importance of model selection in explainable AI systems for aviation and highlights that performance alone is insufficient, interpretability and human trust must be integral design considerations. Future work will focus on validating these findings using real-world flight data, assessing pilot interaction and trust through human-in-the-loop evaluations, and extending the approach to dynamic, time-dependent decision-making scenarios. Together, these steps are essential for advancing explainable, reliable, and operationally safe AI-based decision-support systems for aviation emergencies.

REFERENCES

Bull, L. (2004) 'Learning classifier systems: a brief introduction', in Bull, L. (ed.) Applications of Learning Classifier Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–12. https://doi.org/10.1007/978-3-540-39925-4_1.

Butz, M., Lanzi, P. L. and Wilson, S. (2008) 'Function approximation with XCS: hyperellipsoidal conditions, recursive least squares, and compaction', IEEE Transactions on Evolutionary Computation, 12(6), pp. 355–376. https://doi.org/10.1109/TEVC.2007.903551.

Demir, G., Moslem, S. & Duleba, S. Artificial Intelligence in Aviation Safety: Systematic Review and Biometric Analysis. Int J Comput Intell Syst 17, 279 (2024). https://doi.org/10.1007/s44196-024-00671-w.

Djartov, B., Papenfuß, A. and Wies, M. (2025) 'Wings of wisdom: learning from pilot decision data with interpretable AI models', in Harris, D., Li, W.-C. and Krömker, H. (eds.) HCI International 2024 – Late Breaking Papers. Cham: Springer Nature Switzerland, pp. 241–256. ISBN 978-3-031-76824-8.

G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," in IEEE Potentials, vol. 13, no. 4, pp. 27–31, Oct.-Nov. 1994, doi: 10.1109/45.329294.

Keul, M. (2023) Are there any factors that make the pilot's decision predictable? An analysis about the influence of the conditions on the choice of alternate airports. MA thesis. Frankfurt am Main: Hochschule Fresenius.

Lanzi, P. L., Stolzmann, W. and Wilson, S. W. (eds.) (2000) Learning classifier systems: from foundations to applications. Vol. 1813, Lecture Notes in Artificial Intelligence. Berlin, Heidelberg: Springer. https://doi.org/10.1007/3-540-45027-0.

Nogueira, F. (2014) Bayesian Optimization: Open source constrained global optimization tool for Python. Available at: https://github.com/fmfn/BayesianOptimization.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. and Chintala, S. (2017) 'Automatic differentiation in PyTorch'. Available at: https://pytorch.org.

Preen, R. J. and Pätzel, D. (2024) XCSF. Available at: https://github.com/xcsf-dev/xcsf/wiki. DOI: https://doi.org/10.5281/zenodo.10699246.

S. M. Mian, M. S. Khan, M. Shawez and A. Kaur, "Artificial Intelligence (AI), Machine Learning (ML) & Deep Learning (DL): A Comprehensive Overview on Techniques, Applications and Research Directions," 2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2024, pp. 1404–1409, doi: 10.1109/ICSCSS60660.2024.10625198.

Svozil, D., Kvasnicka, V. and Pospichal, J. (1997) 'Introduction to multi-layer feed-forward neural networks', Chemometrics and Intelligent Laboratory Systems, 39(1), pp. 43–62. https://doi.org/10.1016/S0169-7439(97)00061-0.

Urbanowicz, R. and Moore, J. (2009) 'Learning classifier systems: a complete introduction, review, and roadmap', Journal of Artificial Evolution and Applications, 2009. https://doi.org/10.1155/2009/736398.

Würfel, J., Djartov, B., Papenfuß, A., Wies, M. (2023). Intelligent Pilot Advisory System: The journey from ideation to an early system design of an AI-based decision support system for airline flight decks. In: Gesa Praetorius, Charlott Sellberg and Riccardo Patriarca (eds) Human Factors in Transportation. AHFE (2023) International Conference. AHFE Open Access, vol 95. AHFE International, USA. http://doi.org/10.54941/ahfe1003844.