Determining Feature Importance in Self-Enforcing Networks to achieve Explainable AI (xAI)

Anneliesa Greisbach[1], Christina Klüver[2]

[1]Universität Duisburg-Essen
E-Mail: anneliesa.greisbach@stud.uni-due.de

[2]Universität Duisburg-Essen
Universitätsstr. 12, 45117 Essen
E-Mail: christina.kluever@uni-due.de

1 Introduction

Due to the increasing number of Artificial Intelligence (AI) and Machine Learning (ML) methods and their applications there is a growing demand to understand the results of the respective method (e.g. [1] with reference to the demands in industry 4.0; [2]). While an "explanation component" as in expert systems would be desirable, for many methods this is not applicable.

Explainability is a major challenge, and accordingly, various concepts and methods for achieving Explainable AI (xAI) have been developed in recent years [3], which can be categorized in the taxonomy shown in Fig. 1.

In the stage of explainability, a distinction is made between post-hoc and ante-hoc procedures [2]. In a *post-hoc* procedure, an explanation for a solution occurs after training. This is classified as a *model-agnostic* xAI that applies methods to make the learning process understandable without the actual model passing information to the explanation component. One model-agnostic method is to create a second duplicated model that learns the same data and, upon reaching the same final result, provides information about the basic composition and influence of different features [4][5][6].

DOI: 10.58895/ksp/1000151141-16 erschienen in:
Proceedings – 32. Workshop Computational Intelligence: Berlin, 1. - 2. Dezember 2022
DOI: 10.58895/ksp/1000151141 | https://www.ksp.kit.edu/site/books/m/10.58895/ksp/1000151141/

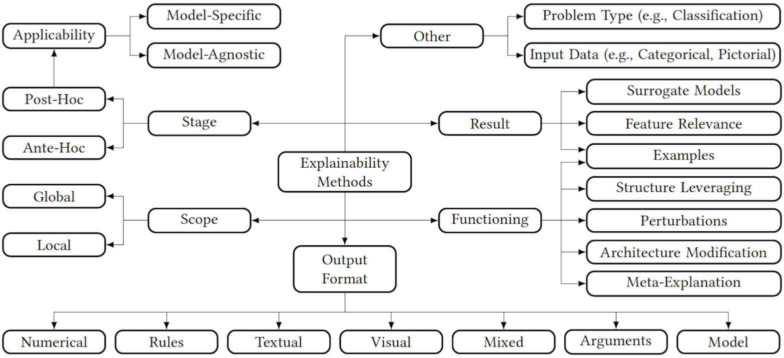

Figure 1: Taxonomy around the topic area of Explainable AI [3, p. 2246]

If the explanation is part of the algorithm, the explainability is designated as *ante-hoc* and is given by design of the algorithm. It is especially suitable for classical machine learning methods, provided that the models are small and manageable. This method is also called *intrinsic xAI*, which is *model specific* and already explainable due to its internal structure. The way of learning can either be transparent (algorithmic transparency), the technical functionality can be clear (simulability), or the algorithm can be decomposed into its individual parts (decomposability) [5][7][8].

Explainable AI algorithms also differ in their *scope*. The scope describes the area to which the explanation refers. It can refer to the functioning of the entire model (global xAI) or to the explanation of the background of a specific decision (local xAI).

In *global xAI*, the explanation component provides an understanding of the general functioning, e.g., the influence of different features on the totality of all decision recommendations. This is used to understand the model and to validate that the features have the expected impact on the decision.

Local xAI provides explanations of a specific decision. This can be used to validate the recommendation or to provide decision support for humans [4][6][9].

The taxonomy of explanatory methods distinguishes whether the outcome of an explanation (*result*) or the operation of an explanatory method is essential (*functioning*). The explanations (*output*) themselves are in turn dependent on the method used and the objective [3].

In this contribution, the meaning of xAI follows the definition of Arrieta et al. [5] according to which an xAI can provide details about a model or explanations of its operation that are understandable to a user. An Explainable AI method for self-organized learning, namely *Self-Enforcing Networks* (SEN) is presented. This method can be used to explain and visualize the impact of a feature on the result. We demonstrate the application of this method to a real case, namely for the decision on the direction of operations at Frankfurt Airport, based on weather data. [10].

AI methods in the broadest sense are used in various areas of air traffic management. Explainability for the methods used is considered by Degas et al. [11] to be necessary for a) describing the algorithms or the results (*descriptive xAI*), b) predicting the behavior of an algorithm (*predictive xAI*), or c) detecting errors or an undesirable behavior of an AI method (*prescriptive xAI*). Accordingly, the solution we present is classified in this case to a) by clarifying the result with a visualization.

The organization of the contribution is as follows: In the next sections, Feature Importance, the concept of Shapley Values, as well as the application of Feature Importance at Self-Enforcing Networks are explained in more detail. In section four, the model of a Self-Enforcing Network as a decision support system regarding the mode of operation at Frankfurt Airport is presented. Recommendations by SEN are shown, especially the benefits resulting from xAI.

2 Feature Importance as a concept of Explainable AI

Within the topic area of Explainable AI, various methods and practices exist that increase the explainability of an algorithm to the user. One of these is explaining the impact of a model's features on the outcome. The impact and relevance of a feature on the overall model can be referred to as *Feature Importance*.

Determining Feature Importance can help the user understand the output of the model and provides him with a basis of information that can be used to analyze and evaluate a concrete decision recommendation. It can also help the developer of the model to validate and optimize the functioning of the entire model.

2.1 The concept of Shapley Values as the basis of Feature Importance

The calculation of the Feature Importance is based on the concept of Shapley Values from the cooperative game theory of Lord Shapley [12], according to which the influence of a player can be computed considering effects by cooperation and individual performance on the game outcome.

The value that a player contributes to the payoff is called the Shapley Value. Shapley Values can be attributed four characterizing properties. They follow the "null player property", are efficient, symmetric, and linear [13].

Null player property. The „null player property“ means that a player with no share in the final outcome will also get no share in the payoff.

Efficiency. The property of efficiency describes that by mapping the actual influence of an actor on the result, conclusions can also be made about this actor.

Symmetry. The property of symmetry indicates that if two players have the same influence on the outcome, they will have the same Shapley Value. Accordingly, a player with a larger contribution to it also has a higher Shapley Value.

Linearity. The property of linearity states that the sum of all Shapley Values makes up the total influence on the outcome.

Only if these four properties are true, a value can be called Shapley Value. The Shapley Value of a player i on the game result can be usually calculated with the following formula [14]:

$$\Phi_i(v) = \sum_{S:i\in S} \frac{|S|!(n-1-|S|)!}{n!}(v(S\cup\{i\}) - v(S)) \tag{1}$$

The overall contribution of a player i to a game outcome is calculated by the sum of all partial influences resulting from coalitions with other players of the player set S, given n players.

The concept of Shapley Values is already applied to artificial intelligence methods. A Shapley Value thus explains the contribution of features on the output of a model and can therefore also be called Feature Importance. This allows a statement to be made about the contribution of a feature in interaction with other features to the final outcome, e.g., a prediction or classification. [13][15][16].

3 Feature Importance at Self-Enforcing Networks

The idea of computing Feature Importance based on the concept of Shapley Values is already used in the context of supervised learning models [15][16][17] and has now been applied to self-organized learning Self-Enforcing Networks (SEN).

The identification of Feature Importance for Self-Enforcing Networks can be categorized as intrinsically explainable Artificial Intelligence due to the structure and operation of SEN, as shall be demonstrated.

3.1 Self-Enforcing Networks

A Self-Enforcing Network (SEN) is a non-supervised and self-organized learning network that acquires and orders knowledge according to cognitive theory learning models [18][19].

SEN consists of three components: the semantic matrix, the actual neural network, and the visualizations [cf. 20].

The semantic matrix is the basis of the learning process. It contains the essential attributes (features) and their degree of membership to an object [18].

As SEN is used as both Machine Learning and Artificial Intelligence method, the data sets are transferred to the semantic matrix according to the problem. In the first case, sensor data, image data, business data, etc. are imported into the semantic matrix, which are learned and ordered according to their similarity.

If SEN is used as a method of AI, frequently so-called *reference types* are formed in addition by experts, based on data containing the specifics of the use case and expert knowledge. This method is used in the following.

The neural network itself is often two-layered and can be designed as a feed-forward, feed-back, or recurrent network depending on the defined connections between attributes and reference types [20].

The data can be normalized in SEN binary in the interval of (0,1) or bipolar in the interval of (-1,1). For each feature, a cue validity factor (cvf) can be set when building the model. The cue validity factor, following the concept of cognitive psychologist Eleanor Rosch [21], influences the strength of an attribute's effect on activation by the network. The cvf can be determined based on the relevance of the attribute for the use case [18].

The peculiarity of SEN is that the weight matrix is not randomly generated but consists of zeros at the beginning.

The Self-Enforcing Rule (SER) i.e. the learning rule used in SEN transforms the values of the semantic matrix v_{sm} into a weight value between object and attribute w_{oa} with the learning rate c [20]:

$$w_{oa} = c * v_{sm} \tag{2}$$

The learning rate describes how forcefully the network adjusts the weight values after each learning process. If the cue validity factor of the respective attribute cvf_a is to be taken into account, the learning rule is extended as follows:

$$w(t+1) = w(t) + \Delta w \text{ und} \tag{3}$$
$$\Delta w = c * w_{oa} * cvf_a$$

This learning procedure makes it possible to reconstruct the results at any time [cf. 18].

Seven activation functions are available in SEN; for the model shown here, the Enforcing Activation Function (EAF) has been proven to be effective [24]:

$$a_j = \sum_{i=1}^{n} \frac{w_{ij} * a_j}{1 + |w_{ij} * a_j|} \tag{4}$$

SEN transfers the information of the semantic matrix to the network via the learning rule. After the learning process, the data are ordered according to their similarity: The more alike the data are, the closer they are represented to one another in the so-called map visualization. When new data are presented after the learning process, the activation values indicate the strength of the similarity of the input data to the reference types. The higher the final activations are, the more similar the input data are to the reference types. In addition, the Euclidean distances of the input data to the reference types are also displayed, indicating their similarity by the smallest distance.

3.2 Intrinsic Explainability at Self-Enforcing Networks

The individual activation values of the attributes are extracted from SEN to directly obtain the Feature Importance, or Shapley Values, which are subsequently visualized. The retrieved values are the individual activations of the attributes and indicate the final activation of the input vector set to a reference type when added together.

The properties "null player property", efficiency, symmetry and linearity of Shapley Values are given, since on the one hand the individual values are extracted and on the other hand, the added values constitute the total activation value. Thus, these can be defined and considered as Feature Importance.

This results from the SEN's mode of operation, which is that the weight values directly reflect the importance of the individual attributes in the reference types, and thus the activation values of the individual attributes have the corresponding effects on the final result.

3.3 Visual representation of Feature Importance as Local Explainable AI

The Feature Importance of Self-Enforcing Networks can be displayed visually to provide the end user with easy access to the values. Either the Feature Importance of the entire model (global xAI) or the Feature Importance of a specific decision (local xAI) can be displayed.

In outlining the influences of a specific recommendation by SEN, the representations listed below are suitable.

The absolute Feature Importance of the input dataset and the classified reference type can be displayed side by side. This allows a quick view of the differences in the activation values of the individual features between the input vector and the reference type (Fig. 2).

In SEN, the output is the reference type that has the highest activation, or the smallest distance, with respect to the new input vector. By applying Feature Importance, the activation is calculated for each individual attribute. Afterwards these can be compared.

In this representation, the bars indicate the absolute Feature Importance of the input vector (gray) and reference type (red). The size of the bars describes the influence of the feature on the total activation value. A bar below the x-axis means negative activation, a bar close to the x-axis means weak activation, a high bar means strong activation.

Another possible representation is the difference between input vector and classified reference type (Fig. 3). For this purpose, the difference in Feature

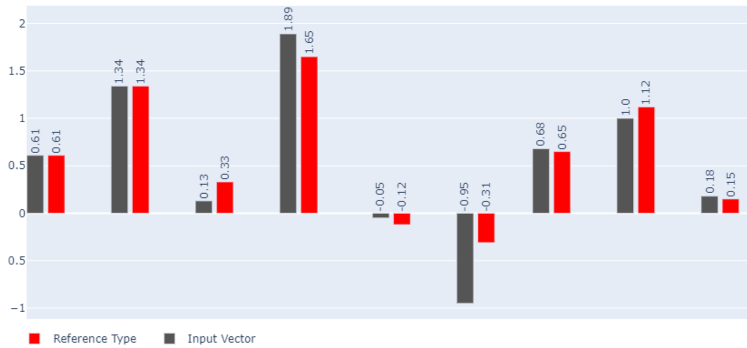

Figure 2: Exemplary absolute feature values of the activation of an input vector and of a reference type by a bipolar normalization.

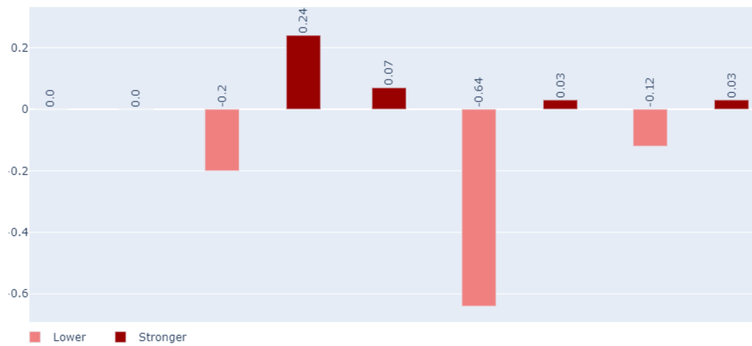

Figure 3: Exemplary difference between input vector and classified reference type.

Importance between the input vector and the classified reference type is calculated and displayed. This representation allows an even easier comparison of the differences and deviations between the Feature Importance of the input vector and the classified reference type.

In this plot, the classified reference type is located on the x-axis. The bars describe the differences between Feature Importance of the input vector and the classified reference type with respect to the individual activation values. A bar below the x-axis means a weaker feature importance, a bar close to the axis of the classified reference type means a small difference to it, a high bar means a large deviation between input vector and the classified reference type.

In addition to the visualizations presented, further visualizations related to reference types that are not classified but preferred by experts can also be created in case the expert would have made a different decision than the SEN. Both visualizations provide the expert with a basis for decision-making.

4 Use case of the selection of the operating direction at Frankfurt Airport

The determination of the Feature Importance is presented as an example for the recommendation of the operating direction at Frankfurt Airport.

4.1 Description of the use case

Frankfurt Airport has four runways (Fig. 4). Three of them are on a parallel runway system aligned with the compass heading of 250° and 70°, with operating directions designated 25 and 07. Runway West, with a compass heading of 180° and the designation 18, lies nearly orthogonal to the parallel runway system. This runway can only be used for takeoff in one given direction [22].

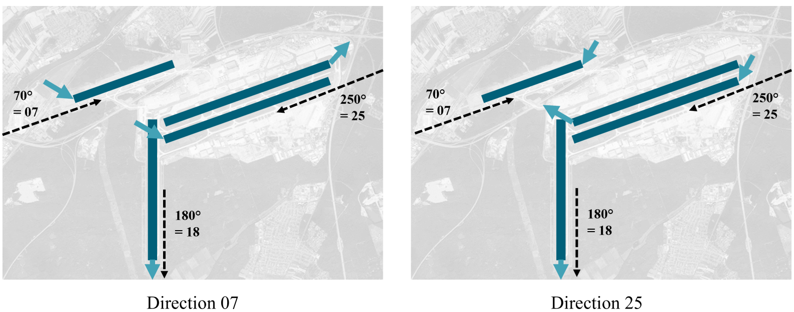

Figure 4: Own representation of the runways at Frankfurt Airport according to [25, p. 376].

The runways on the parallel runway system are each used in the direction specified by air traffic control [23].

The decision on the direction of operations is made by air traffic control on the basis of the current conditions, in particular on the basis of the prevailing wind direction and strength. For this purpose, air traffic control has, among other things, weather forecasts on the wind development, measurements of the wind conditions at the airport near the ground and information from the pilots on the wind conditions during landing [24].

The design of the weather forecast is defined worldwide by the regulations of Annex 3 of the International Civil Aviation Organization (2018). As Terminal Aerodrome Forecast (TAF), the forecasts for wind direction and wind speed, visibility, cloud cover and significant weather phenomena are provided for international airports [24].

Since wind conditions in Central Europe can change rapidly within a short period of time, air traffic control must monitor weather forecasts for the next few hours at all times and continuously review decisions made regarding the direction of the runway [25].

The choice of operating direction is a major challenge for air traffic control because, on the one hand, the safety of crew and passengers must be ensured at all times and, on the other hand, the decision to change the operating direction involves considerable effort and delays, so it must be very well justified [23].

Furthermore, such a decision to change the direction of operations requires a certain lead time in order to be able to coordinate the actual implementation at the airport. Especially in situations with weak winds, the available information is not very meaningful, which complicates the decision of air traffic control [23][24].

The use of a computer-based decision support system can remedy this situation and validate or automate the decision through a second assessment.

In cooperation between the department of aviation meteorology of Deutscher Wetterdienst (the National Weather Service of Germany), which among other things provides the weather forecasts for Frankfurt Airport, and the research

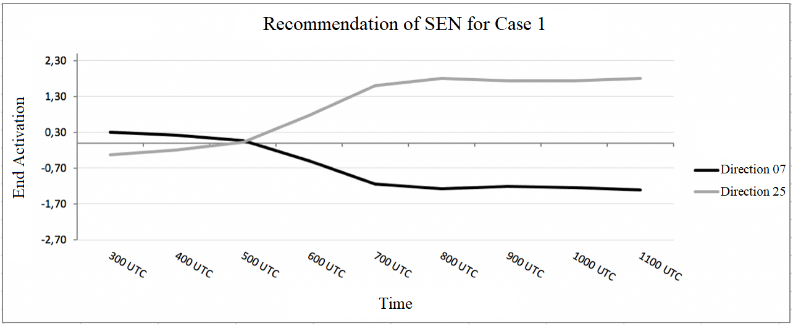

Figure 5: Recommendation on the direction of operation by SEN [23]

group CoBASC, such a decision support system for Frankfurt Airport was developed. The SEN has as parameter settings the Enforcing Activation Function (EAF) a learning rate of 0.5 and 4 learning steps.

Based on the predicted wind conditions, it gives a recommendation for the optimal operating direction to the available weather forecasts. The SEN thus provides a recommendation of whether and when the operating direction should be changed [23].

To simplify and test the first model, the focus was initially placed on the parallel runway system with runway directions 07 and 25. The decision for or against Runway West with runway direction 18 was not initially integrated into the model. To briefly illustrate how SEN works, an example of a recommendation by SEN based on the weather forecast is shown in Fig. 5. At 05:00 UTC, SEN recommends to change the operating direction to direction 25.

Since the recommendations by SEN follow the same trend as those of the controllers, SEN is suitable to serve as a decision support system for Air Traffic Control [23].

4.2 Data of the use case

Air traffic control at Frankfurt Airport uses weather forecasts from the COSMO-DE Ensemble Prediction System (COSMO-DE-EPS), which is provided by the

German Weather Service. As an ensemble forecast model, it provides a total of 20 possible weather forecasts. Uncertainties and probabilities for the forecasts can be expressed by the 20 ensemble members.

For the model in SEN, the path-parallel wind components were calculated at 11 measurement points, meaning the wind strengths and directions aligned with compass degrees 70° and 250°, respectively (Fig. 6). The calculated path-parallel wind component thus provides information on the strength of the headwind and tailwind in knots for aircraft taking off and landing on the glide path. The glide path is the path that aircrafts follow when using the runway.

To represent the uncertainty of the forecasts in SEN, the path-parallel wind component at the 11 measurement points was calculated for all 20 ensemble members. The ensemble members were then used to calculate the quantiles at each of the points with thresholds of 10%, 25%, 50%, 75%, and 90% [24]. Thus, each dataset has five wind components for each of the 11 measurement points, making a total of 55 features in SEN [23]. The cue validity factor was set in SEN per feature as a function of the quantiles and the distance to the runway.

The geographic location of the reference points is shown in Fig. 7.

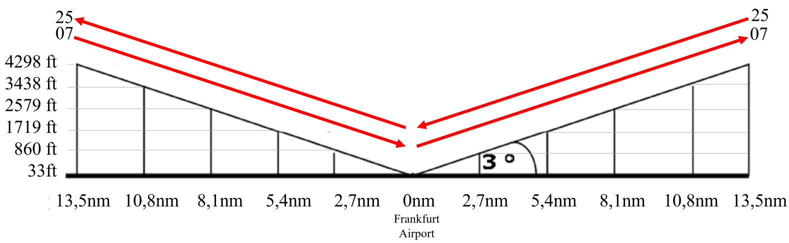

Figure 6: Own model, not to scale, representation of the glide path and the 11 measuring points of runway direction 07 and 25 at Frankfurt Airport based on [24, p. 232].

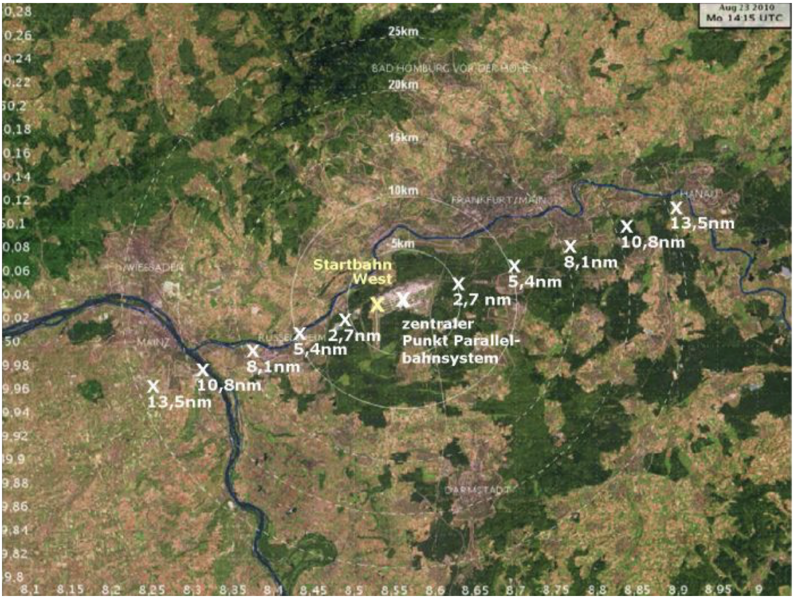

Figure 7: Geographical location of the measurement points [24, p.232].

4.3 Determination of the Local Feature Importance of the use case

The decision to change the direction of operation involves considerable effort as well as delays and must be well justified. An explanation component is of substantial benefit, especially when wind conditions do not permit an immediate and unambiguous decision.

Since a representation of the 55 features in a visualization would appear unmanageable, it was decided specifically for this use case to add the individual activation values of the five calculated quantiles per measuring point and thus to obtain an activation at the 11 measuring points.

Fig. 8 shows the absolute Feature Importance of reference type ’Direction 07’. The Feature Importance of the measurement points is arranged according to the positioning on the glide path for greater clarity and, when viewed together,

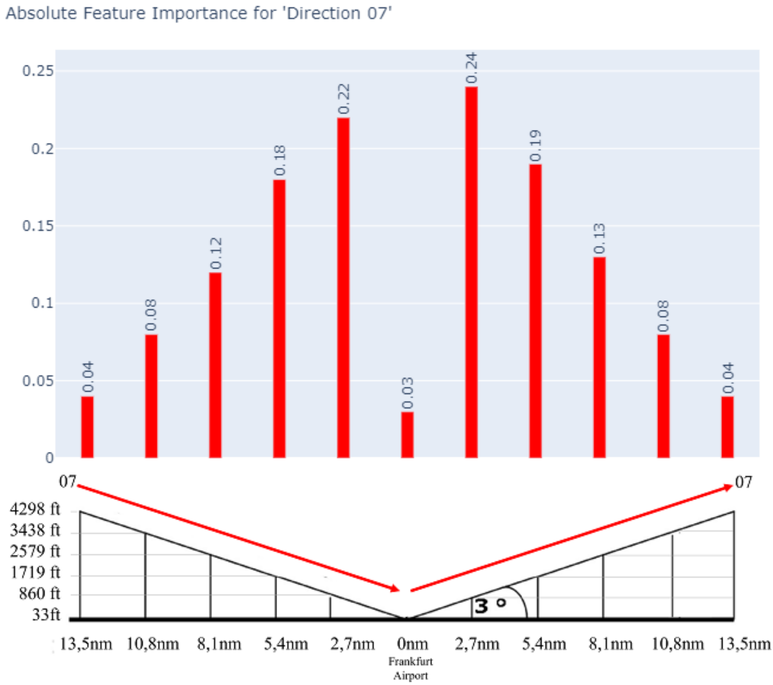

Figure 8: Visualization of the Feature Importance for reference type 'Direction 07' with a model-like, not to scale representation of the measuring points on the glide path at Frankfurt Airport following Zinkhan [24].

allows direct conclusions to be drawn about the wind conditions at the various heights of the glide path.

To visualize the Feature Importance for the use case, the data of "case 1" from [23] is used and analyzed with respect to their Feature Importance. For each of these 9 recommendations between 03:00 UTC and 11:00 UTC, the Feature Importance can be read out to analyze the decision recommendation in more detail. The Feature Importance of the decision at 09:00 UTC is analyzed in depth below.

In Fig. 9 the input vector 09:00 UTC was classified as 'Direction 25'. The Feature Importance with respect to this classification is shown in Fig. 10 as

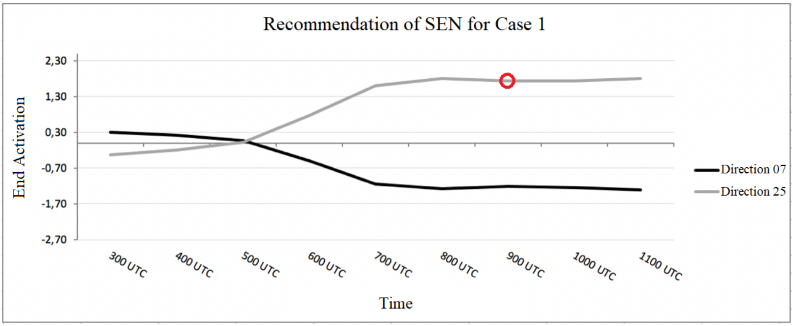

Figure 9: Visualization of the recommendation on the direction of operation by SEN from case 1 at 09:00 UTC [24].

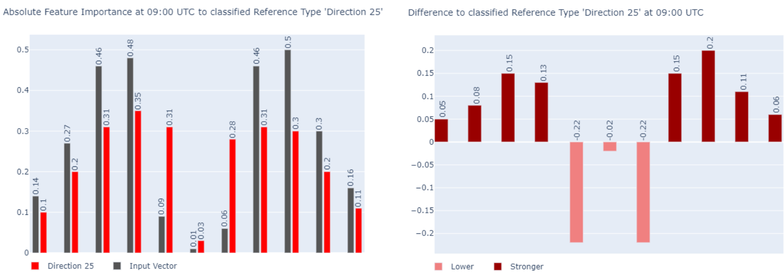

Figure 10: Feature Importance of 09:00 UTC regarding ‘Direction 25’

absolute value and as difference to the Feature Importance of the reference type ’Direction 25’.

Both plots indicate that for input vector 09:00 UTC, the high-altitude winds have a higher activation than those of the reference type. The ground winds, on the other hand, have a lower influence on the final activation.

The final activation of 2.93 of the input vector can be broken down by this visualization and gives experts the opportunity to analyze the classification and the differences to the feature influences on the final result in more detail. Decision support systems, such as this one, thus take on greater meaningfulness.

5 Conclusion and further work

In this contribution, we have shown how Feature Importance can be applied to self-organized learning networks. The architecture and learning procedures of SEN fulfill the properties of Shapley Values. Feature Importance can be read directly by decomposing a vector into its individual components (individual attributes) without additional calculations; thus, SEN can be classified as intrinsic xAI.

The knowledge gained will be integrated into a follow-up project in which weather forecasts are to be improved by using large amounts of data. Since a reference type consists of almost 2,000,000 data points, it is of great importance for decision makers to know which features are decisive for the decision of a runway.

With the developed method, there is no loss of performance in determining Feature Importance, but the number of features must be reduced. One possibility is for the experts themselves to determine which features are relevant to them and should be displayed.

References

[1] I. Ahmed, J. Gwanggil, F. Piccialli. "From artificial intelligence to explainable artificial intelligence in industry 4.0: a survey on what, how, and where". In: *IEEE Transactions on Industrial Informatics* 18.8. pp. 5031-5042. 2022.

[2] M.R. Islam, M.U. Ahmed, S. Barua, S. Begum. "A Systematic Review of Explainable Artificial Intelligence in Terms of Different Application Domains and Tasks". In: *Applied Science* 12, 1353. 2022.

[3] T. Speith. "A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods". In: *ACM Conference on Fairness, Accountability, and Transparency.* pp. 2239-225012. 2022.

[4] A. Das, P. Rad: “Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey”. In: *arXiv e-prints (2020): arXiv-2006*. 2020.

[5] A.B. Arrieta, N. Díaz Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado et al. “Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI”. In: *Information Fusion* 58. pp. 82-115 2020.

[6] G. Schwalbe, B. Finzel: “A Comprehensive Taxonomy for Explainable Artificial Intelligence: A Systematic Survey of Surveys on Methods and Concepts” In: *arXiv e-prints, arXiv-2105*. 2021.

[7] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi: “A Survey of Methods for Explaining Black Box Models”. In: *ACM Computing Surveys* 51.5. 1019.

[8] B. Mittelstadt, C. Russell, S. Wachter: “Explaining Explanations in AI”. In: *Proceedings of the conference on fairness, accountability, and transparency* pp. 279-288. 2019.

[9] A. Adadi, M. Berrada. “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)”. In: *IEEE Access* 6. pp. 52138–52160. 2018.

[10] D. Zinkhan, S. Eiermann, C. Klüver, J. Klüver. “Decision Support Systems for Air Traffic Control with Self-enforcing Networks Based on Weather Forecast and reference types for the Direction of Operation”. In: *Advances in Computational Intelligence. IWANN 2021* (Rojas I., Joya, G., Catala, A. eds.), pp. 404-415. Springer, Cham. 2021.

[11] A. Degas, M.R. Islam, C. Hurter, S. Barua, H. Rahman, M. Poudel, D. Ruscio, M.U. Ahmed, S. Begum, M.A. Rahman, et al. “A Survey on Artificial Intelligence (AI) and eXplainable AI in Air Traffic Management: Current Trends and Development with Future Research Trajectory”. In: *Applied Science* 12. 1295. 2022.

[12] L.S. Shapley "A Value for n-Person Games". In: *Contributions to the Theory of Games (AM-28)* (Kuhn, H.W., Tucker A.W. eds.), 2. pp. 307–318. Princeton: Princeton University Press. 1953.

[13] B. Rozemberczki, L. Watson, P. Bayer, H.-T. Yang, O. Kiss, S. Nilsson, R. Sarkar. "The Shapley Value in Machine Learning". In: *Proceedings of the 31st International Joint Conference on Artifical Intelligence* (L. De Raedt, L., ed.) pp. 5572-5579. IJCAI-ECAI 2022.

[14] L. Bokati, O. Kosheleva, V. Kreinovich, N.N. Thach. "Why Shapley Value and Its Variants Are Useful in Machine Learning (and in Other Applications)". In: *Proc., 15. Workshop Computational Intelligence*. Departmental Technical Reports (CS). 1729. 2022.

[15] S. Lundberg, S.-I. Lee. "An unexpected unity among methods for interpreting model predictions". In: *Proceedings of NIPS 2016 Workshop on Interpretable Machine Learning for Complex Systems* (Wilson, A.G., Kim, B. Herlands, W. eds.) arXiv preprint arXiv:1611.07478 (2016). 2016.

[16] S. Lundberg, S.-I. Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in neural information processing systems* 30. pp. 4768-4777. 2017.

[17] H. Chen, S. Lundberg, S.-I. Lee. "Explaining a series of models by propagating Shapley values". In: *Nat Commun* 13. 4512. 2022.

[18] C. Klüver, J. Klüver "KI - Das Self-Enforcing Network (SEN)". In: *Neue Algorithmen für praktische Probleme. Variationen zu Künstlicher Intelligenz und Künstlichem Leben* (Klüver, C. J. Klüver, J. eds.). pp. 9-20. Wiesbaden: Springer Fachmedien. 2021.

[19] C. Klüver, J. Klüver "Chancen und Herausforderungen beim Einsatz neuronaler Netzwerke als Methoden der Künstlichen Intelligenz oder des Maschinellen Lernens in KMU.". In: *Digitalisierung und Nachhaltigkeit – Transformation von Geschäftsmodellen und Unternehmenspraxis. Organisationskompetenz Zukunftsfähigkeit* (M. Bodemann, W.

Fellner, V. Just eds.). pp. 121-148. Berlin, Heidelberg: Springer Gabler. 2022.

[20] C. Klüver, J. Klüver, J. Schmidt. “Modellierung komplexer Prozesse durch naturanaloge Verfahren. Künstliche Intelligenz und Künstliches Leben”. Wiesbaden: Springer Fachmedien. 2021.

[21] E. Rosch. “Natural Categories”. In: *Cognitive Psychology* 4.3. pp. 328-350. 1973.

[22] H. Mensen. “Planung, Anlage und Betrieb von Flugplätzen”. Berlin, Heidelberg: Springer. 2013.

[23] C. Klüver, J. Klüver, D. Zinkhan. “A Self-Enforcing Neural Network as Decision Support System for Air Traffic Control based on probabilistic Weather Forecasts”. In: *International Joint Conference on Neural Networks (IJCNN)*. pp. 729-736. 2017.

[24] D. Zinkhan. “Entscheidungsunterstützungssystem zur Interpretation probabilistischer Wettervorhersagen für den Flughafen Frankfurt”. In: *Neue Algorithmen für praktische Probleme. Variationen zu Künstlicher Intelligenz und Künstlichem Leben* (Klüver, C. J. Klüver, J. eds.). pp. 9-20. Wiesbaden: Springer Fachmedien. 2021.

[25] J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha, D. Abramson. “Displacement Problem and Dynamically Scheduling Aircraft Landings”. In: *Journal of the Operational Research Society* 55. pp. 54-64. 2004.