JOHANNES KEPLER
UNIVERSITÄT LINZ

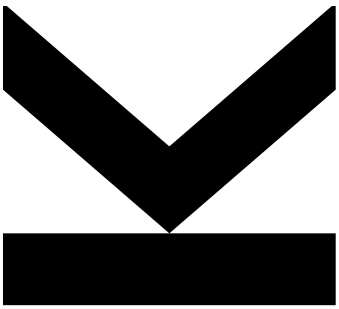
Submitted by
Martin Dallinger

Submitted at
LIT Secure and Correct Systems Lab,
Secure Systems Group

Supervisor
Univ.-Prof. DI DI Dr. Stefan Rass

July 2024

Fuzzy Rule-Based Regression for Explainable AI



Seminar Thesis

in the Program

Artificial Intelligence

JOHANNES KEPLER
UNIVERSITY LINZ
Altenbergerstraße 69
4040 Linz, Austria
www.jku.at
DVR 0093696

Contents

1 Introduction to eXplainable Artificial Intelligence (XAI)

Over the last decades, Artificial Intelligence (AI) Systems have revolutionized the way in which classical computer science problems are approached which lead to equalling or even outperforming humans in more and more tasks [1]. These recent successes were mainly achieved in the subfields Machine Learning (ML) and Deep Learning (DL) where modern approaches are often too complex to be interpretable by humans. Such AI models are often called *black boxes* as without additional explanation methods it is unknown what reasoning emerges in the model after training.

One of the fundamental problems of such complex, obscure, but often well-performing systems is the lack of trust - even more so in domains like disease diagnosis as was highlighted by Cao et al. [2]. One major goal of the research area eXplainable Artificial Intelligence (XAI) is to bring light into these black boxes. However, these techniques are not only useful to comprehend the inner-workings of AI models but also to explain human (mis-)behavior in creating datasets as we shall see later.

1.1 Overview of XAI

Before we accustom ourselves with the intricate details of concrete models or methods, let us see bigger picture of XAI to better understand the problems these models solve. According to Adadi et al. [3], XAI has four primary motives:

1. **Explain to justify** (the decisions): Particularly crucial in high-stakes fields like healthcare or finance for trust and accountability. Reasons for the justification should be provided and checked to avoid erroneous outcomes.

2. **Explain to control**: Helps identify when AI operates outside intended parameters, and aids in understanding the capabilities of the system. Furthermore, this lucidity of XAI aids in finding errors or showing potential vulnerabilities.

3. **Explain to improve**: Understanding AI decisions aids in model refinement by highlighting shortcomings or biases.

4. **Explain to discover**: AI uncovers new knowledge, then explanations of these discoveries may lead to new insights and advancements.

Considering these different goals, it is apparent that different architectures will be more or less effective in achieving them. In Section 3 we will notice that later introduced fuzzy models and linear combinations of them can fulfill many of these requirements.

To see how these models fit into the bigger picture, let us consider a basic taxonomy of such XAI models in Figure 1.1. It provides a simplified combination of the XAI-taxonomies from Singh et al. [4] and Ding et al. [1].

In categorizing explainability based on scope, three levels can be distinguished. The first, termed *local*, involves inferences leading to a specific outcome, such as selecting a single leaf or branch in a decision tree. This demonstrates the decision tree's practical application for narrow, targeted decisions. Continuing with the example of decision trees, the second level, *semi-local*, encompasses broader but still limited sections of the tree, involving multiple branches or outcomes. This scope is used when a subset of the decision tree needs to be considered for analyzing a decision-making process. The third, *global* scope, involves the analysis of an entire model and particularly its reasoning for all possible outputs (in this example the entire decision tree). Contrary to merely presenting the

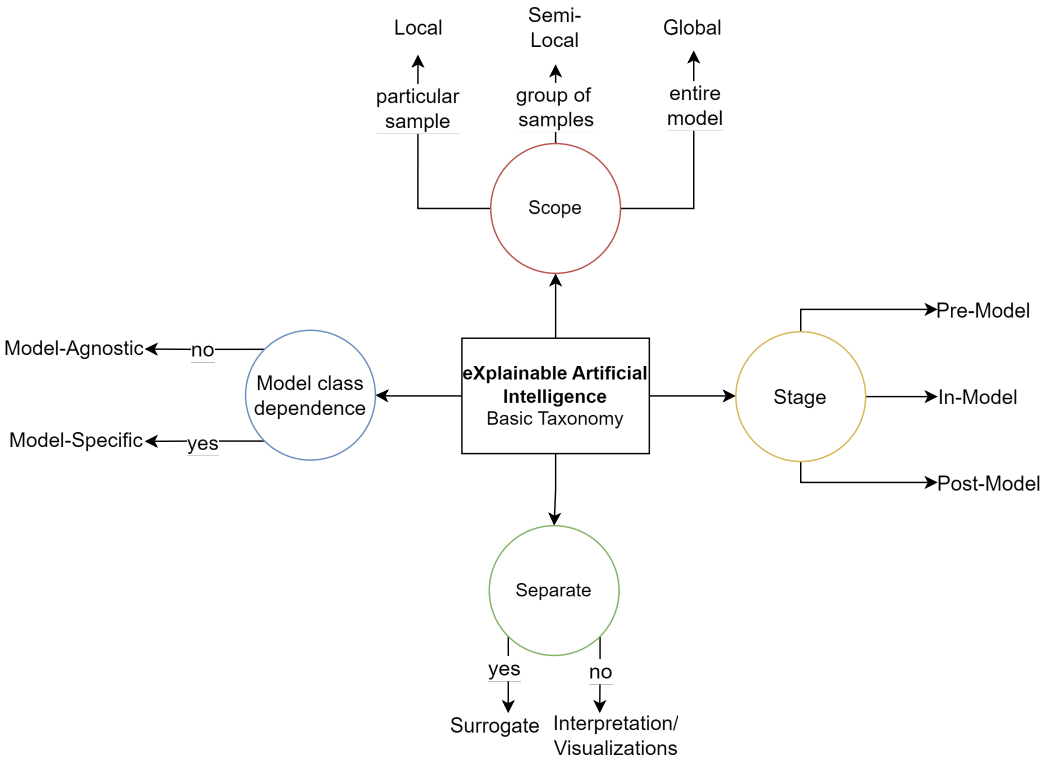


Figure 1.1: A non-exhaustive basic taxonomy of XAI methods (adapted from [4] and [1])

decision tree as an output, the *global* approach implies utilizing the complete structure to overview or analyze all potential decisions and outcomes.

If the applicability of the explanation method is very restricted to certain model classes, then the explanation method is referred to as being *model-specific*. If the method can work on various model-classes, we refer to it as being *model-agnostic*.

The categorization can also be made with regard to the stage of model development or deployment: Model-agnostic methods, like downprojection of the data, which are typically applied to the data before using the model, are referred to *Pre-Model* approaches. If the model comes with intrinsic properties giving interpretable meaning to its outputs (like fuzzy approaches in Section 2 and Section 3), then we speak of *In-Model* systems and if the explanations come from techniques applied after the model was trained, the authors refer to *Post-Model* systems, which are usually model-agnostic. A popular and simple example of such Post-Model systems was highlighted by L. Breiman [5] where one of the input-features of a dataset is randomly permuted to then assess its importance by decrease in predictive performance.

If possible, it is always preferable to analyze the model of interest directly by examining and visualizing parameters or inherent aspects. Unfortunately, certain architectures demand different analysis methods. Therefore, another common technique, called *surrogate models*, involves using other, more transparent models to attempt to (often only locally) describe the inner workings of the black-box model. However, for the sake of clarity, this work will concentrate solely on direct model interpretations, leaving the intriguing realm of surrogate models for another exploration.

To witness the practical use-cases of the above definitions, let us examine one concrete technique in

more detail.

1.2 Significance of Fuzzy Logic Within XAI

One common application for fuzzy logic in XAI is to use generated rules (either automatically or by human labor) and then verify how well they explain the given data. A *rule* is commonly seen as an If-Then-Statement. For example: If the temperature is low then the heating power should be increased. In Section 2.2.2 we will note that, due to the definition of fuzzy systems, fuzzy rules should more correctly be viewed as a correlation as shown by Mendel et al. [6]. Nonetheless, for now it suffices to see a rule as an If-Then-Statement.

Since these rules are fully transparent and can even be stated in natural language, they are fully explainable descriptions of what happens on the technical level. We can use different methods (e.g. linear combinations of them) to describe a dataset, and then evaluate the contributions of the rules to see how important they are and if these significances deviate from the expectations. A prime example of this process is provided by Cichy et al. [7, 8] in the data quality management domain, where such rules are defined based on domain-specific metrics, which are later regressed over. As a result, one could quickly estimate which rules are important, see if there is contradicting information and even use the model to generate new predictions. Thus, we can see that such systems are promising with respect to all the four explainability motives we listed in the beginning of this section, but the example especially shows how such systems excel in explaining to discover.

In the categorization with the simplified taxonomy of Section 1.1, such a fuzzy-based rule-regression-system can be viewed as *interpretations* of a *model-specific in-model global* method. This piece will focus on such and similar tasks. It should, however be noted, that recent work by Zhu et al. [9] has shown that fuzzy models can also be used as model-agnostic surrogate models for local interpretability of more complex DL models.

2 Fundamentals of Fuzzy Logic

Traditional logic is based on crisp binary values, which can only be *true* or *false*. The essence of fuzzy logic is based on L.A. Zadeh's *fuzzy sets* [10] where the degree of membership of an arbitrary element of the fuzzy set a in the set X is not denoted by 0 (i.e. false: a is not an element of X) or 1 (true: a is an element of X) like it would be in the traditional set-theoretic sense. Instead, it is expressed by a membership degree $\mu_X(a) \in [0,1]$, where $\mu_X : \mathbb{R} \to [0,1]$ is then commonly referred to as being the *Membership Function (MF)* of the fuzzy set X.

To illustrate the necessity of this approach for accurately describing the real world, L.A. Zadeh [10] provided a prominent example: The class of animals is not as well-defined as we think, since for most humans it might be clear that clouds and buildings are not animals but cats and dogs are. Looking at the boundary, however, like bacteria or starfish, it becomes clear that also this distinction should perhaps not be modeled as being binary (=*crisp*) but in the opposite way with a degree of membership (=*fuzzy*). This observation becomes even more pronounced when we consider that the challenge of classification is not solely limited to individuals. It rather extends to the variability in perspectives among different people, leading to potential discrepancies in setting boundaries for identical natural language descriptions.

Amongst others, E. H. Mamdani transfers this concept to approximate reasoning and even argued:

> ... vagueness is not a defect of language, but rather an important source of creativity. Analogies are extremely important to creative thinking and vagueness surely plays a dominant role in such thought processes.[1]

The definition of *fuzzy logic* as given by Swathi et al. [12] is "A type of mathematical logic ...", that "... makes it easier to reason with incomplete or contradictory material". This logic then uses fuzzy variables, which are derived by the respective set MFs and their operations as opposed to simple binary variables used in traditional logic. Let us choose the AND-Gate for comparing an example operation of traditional with fuzzy logic:

For traditional logic it holds that if we have two boolean variables m and n, the expression m AND n is exactly *true* if and only if (iff) m is *true* and n is *true*. In fuzzy logic the AND-operation can be expressed in multiple ways. A common way to define this relationship between the fuzzy variables was also shown by E. H. Mamdani [11] as follows: Let $\mu_A, \mu_B : \mathbb{R} \to [0,1]$ be arbitrary MFs for some fuzzy sets A and B. Moreover, let x be the degree of membership of the crisp value a in the fuzzy set A, specified as $x := \mu_A(a)$ and similarly define $y := \mu_B(b)$. Then a AND $b = \min(x, y)$.

Before proceeding further into the intricacies of operations involving fuzzy sets and their correspondence with fuzzy logic, we shall gain familiarity with the types of fuzzy sets commonly employed in the literature.

2.1 Types of Fuzzy Sets

In the literature, authors commonly distinguish between two (sometimes three) forms of fuzzy sets:

- **Type 0**: Less commonly referred to explicitly, but still used in publications like Liang et al. [13]. It commonly refers to the crisp output of a fuzzy Takagi-Sugeno-Kang (TSK) system where the inherent uncertainty has been resolved into a definitive value. TSK will be introduced in the beginning of Section 2.2 and more concretely explained in Section 2.2.4.

[1] Page 1182, [11]

- **Type 1**: As described in the introduction of Section 2. Each element has a degree of membership ranging between 0 and 1.

- **Type 2**: Mendel et al. [14] claim that "There are (at least) four different sources of uncertainties in type-1 FLSs", where FLS stands for Fuzzy Logic Systems. They identify the first two sources as stemming from the variability in how individuals, potentially experts, interpret the meanings of fuzzy rule outputs, suggesting that these interpretations should be considered as distributions. The latter two sources of uncertainty arise from noise affecting the inference data and the training data used for model fitting. The authors convey the concept of fuzzy MFs, where a new layer of fuzziness is introduced: The uncertainties of the MFs are added on top of the usual notion of fuzzy sets, yielding more expressive power at the cost of complexity.

For the sake of clarity, this work will focus exclusively on Type-1 fuzzy sets, with the possibility of extension to broader applications in subsequent research.

2.2 Fuzzy Inference Systems (FIS)

The computational models built upon the principles of fuzzy logic are called Fuzzy Inference Systems (FIS). What makes these models special is that the inference is built on linguistic variables with their inherent uncertainties. Thus, every part of the inference (usually If-Then-Statements) is comprehensible with natural language. Such FIS commonly consist of three parts *Fuzzification*, *Fuzzy Inference* and *Defuzzification* described in the following subsections.

There are two popular types of FIS: Mamdani and TSK systems as depicted in Figure 2.2. In this context *linguistic* variables can be seen as fuzzy variables - the wording should emphasize that no AI-expertise is required for creating and interpreting inference rules. The key difference lies in the output of the fuzzy inference and third step where Hamam et al. [15] show that both systems have their (dis-)advantages over the other.

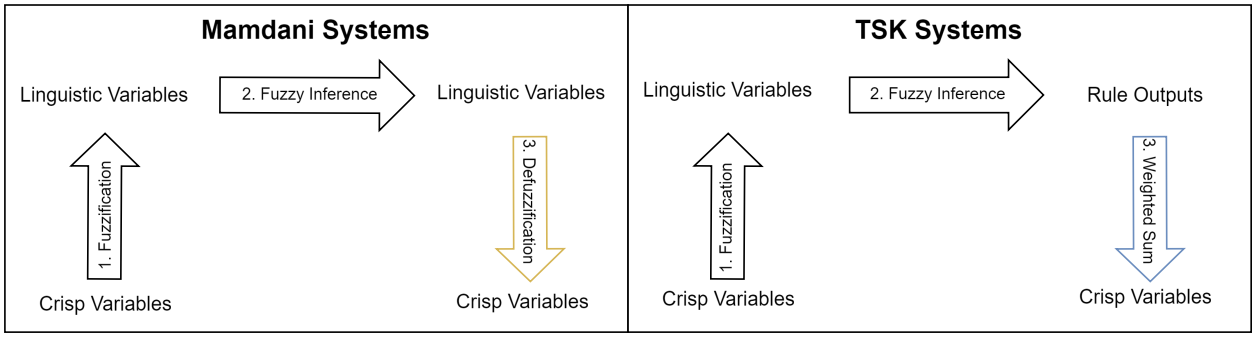


Figure 2.2: A high-level comparison between Mamdani and TSK systems adapted from [16]

Most importantly, the rules of fuzzy inference systems can always be specified in natural language and hence remain intrinsically interpretable. This will become more clear after examining it in greater detail with the following example:

Say we want to model a predictor with the task to estimate the final competition-placement of a contestant. This predictor considers the two factors:

- The foating-point number of hours (x) the contestant has spent preparing for the competition.

- The foating-point number of weeks (y) it takes the contestant to complete one level of a

specified course.

A thorough visualization of the entire Mamdani FIS with respect to this example is given in Figure 2.4. Let us now devote our attention to the three specified parts of a Mamdani system and examine the image in greater detail:

2.2.1 Fuzzification

The first step in the default pipeline of a FIS, called fuzzification, is perhaps the most standardized one. For Type-1 fuzzy logic it simply means to apply the respective MF μ_X of the parameter-domain X to the crisp input values which will yield the degrees of membership with certain sets. Note that these MFs can be arbitrarily complex and have no restrictions regarding their definition except for the range $\mu_X(x) \in [0, 1]$. As shown in Figure 2.4, this example uses trapezoidal and linear MFs for simplicity, but as L.A. Zadeh already introduced in the original paper concerning fuzzy sets [10], these could be very complex non-convex structures if the system designers choose so.

In the context of our example, this could translate to belonging equally to the set "Beginner" and to "Intermediate" with a degree of 0.45, since 115 hours of practice are assumed for the concrete input. With more formal notation we express this as $\mu_{\text{Beginner}}(115) = 0.45$ and $\mu_{\text{Intermediate}}(115) = 0.45$. Similarly, since the input for the duration a user would take to finish a level was assumed to be supplied 1.5 weeks, we can read off that the remaining set memberships are: $\mu_{\text{Medium}}(1.5) = .0.65$ and $\mu_{\text{Slow}}(1.5) = 0.35$. The memberships for the classes "Fast" and "Professional" are 0.

We can even extend the descriptive capabilities by adding certain linguistic operators called *fuzzy hedges* as later shown by L.A. Zadeh [17]. Simple Type 1 Fuzzy hedges typically include the words "slightly, highly, very, more or less, much"[17]. For instance, the hedge "very" was defined by the author by taking the default MF of the class and squaring it, which will reduce almost all values as the co-domain has to be in $[0, 1]$. This can also be seen in Figure 2.3. *Type 2 Fuzzy hedges on the other hand require more descriptive context and are out of scope for this thesis.*

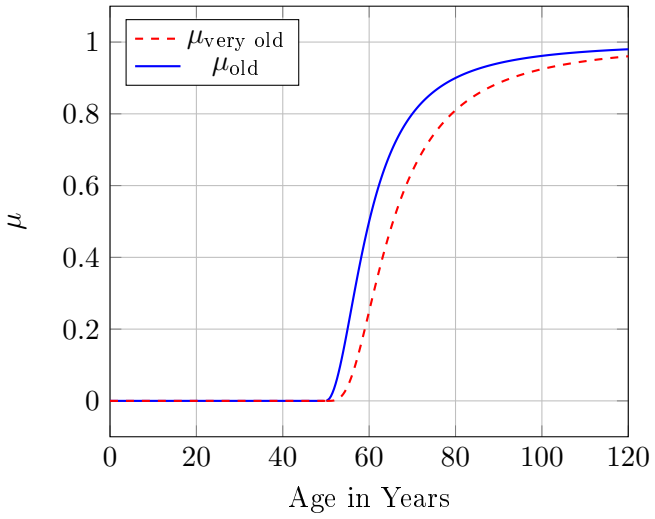


Figure 2.3: The *very*-hedge comparing the MFs "old" and "very old" (adapted from L.A. Zadeh [17])

2.2.2 Fuzzy Inference (Mamdani)

After successfully obtaining the degrees of membership, we can apply pre-defined or generated fuzzy rules as follows: We use the natural language description of If-Then-Rules but actually apply something more complex and similar to correlation as was shown by Mendel et al. [6]. In traditional logic, we have that for the boolean variables x and y. If x is false, then $x \implies y$ (read x implies y, so y is true if x is true) gives no indication with regard to the truth value of y. In fuzzy logic there exist (at least) 14 different methods to encode the implication operation as Wedding et al. [18] have outlined.

However, for this work we will adhere to the original FIS approach outlined by Mamdani et al. [19] which involves limiting the membership value of the consequent by the computed value of the antecedent. Here, the *antecedent* refers to the If-Part of the implication, while the *consequent* denotes the Then-Part. In the context of fuzzy inference, the consequent describes the desired output for later defuzzification.

For computing the value of the rule's antecedent we can apply the original ideas behind manipulating fuzzy sets from L.A. Zadeh [10]. Let a and b be certain domain-specific values, which are fuzzified by some MF μ_A and μ_B respectively. The *logical operations* between these variables in fuzzy logic are commonly realized as follows:

- **AND** $\mu_{A\cap B}$: Multiple possibilities to encode this relationship are commonly applied. Popular and simple variants are $\min(\mu_A(a), \mu_B(b))$ and $\mu_A(a) \cdot \mu_B(b)$, where K. Wang [16] claims that the product gives a smoother output (perhaps with respect to differentiability) which is said to be desirable in systems modeling.
- **OR** $\mu_{A\cup B}$: $\max(\mu_A(a), \mu_B(b))$.
- **NOT** $\mu_{\neg A}$: Recall that $\mu_A \in [0, 1]$. Thus, we can define negation as $1 - \mu_A(a)$

Using what we learned before about hedges, we can now see that the rather natural sounding sentence "If person is professional and very intelligent then performance is good" could quite easily be encoded into fuzzy logic by software.

2.2.3 Defuzzification (Mamdani)

In fuzzy logic, the concluding step entails transforming the fuzzy output memberships into a singular, precise value. This crucial process, termed defuzzification, offers various methods for execution, including centroid, center-of-sets, and height, as detailed by Saadaoui et al. [20]. For the purpose of this discussion, we will focus on the centroid approach, which is also the default method employed in MATLAB. Essentially, it computes the center of mass beneath the amalgamated membership function curve, depicted in Figure 2.4.

During fuzzy inference, each rule contributes a fuzzy value of a respective output-set X. The centroid method considers these contributions c_X along with the original fuzzy set definitions for the output sets via the respective MF μ_X. Mathematically, the minimum between the original MF and the consequent value is taken, where c_X is treated as a constant function: $\min\{\mu_X, c_X\}$. This effectively creates horizontal slices at the height of each consequent, and the centroid is calculated based on this modified MF which can be seen in Figure 2.4 as the white cross.

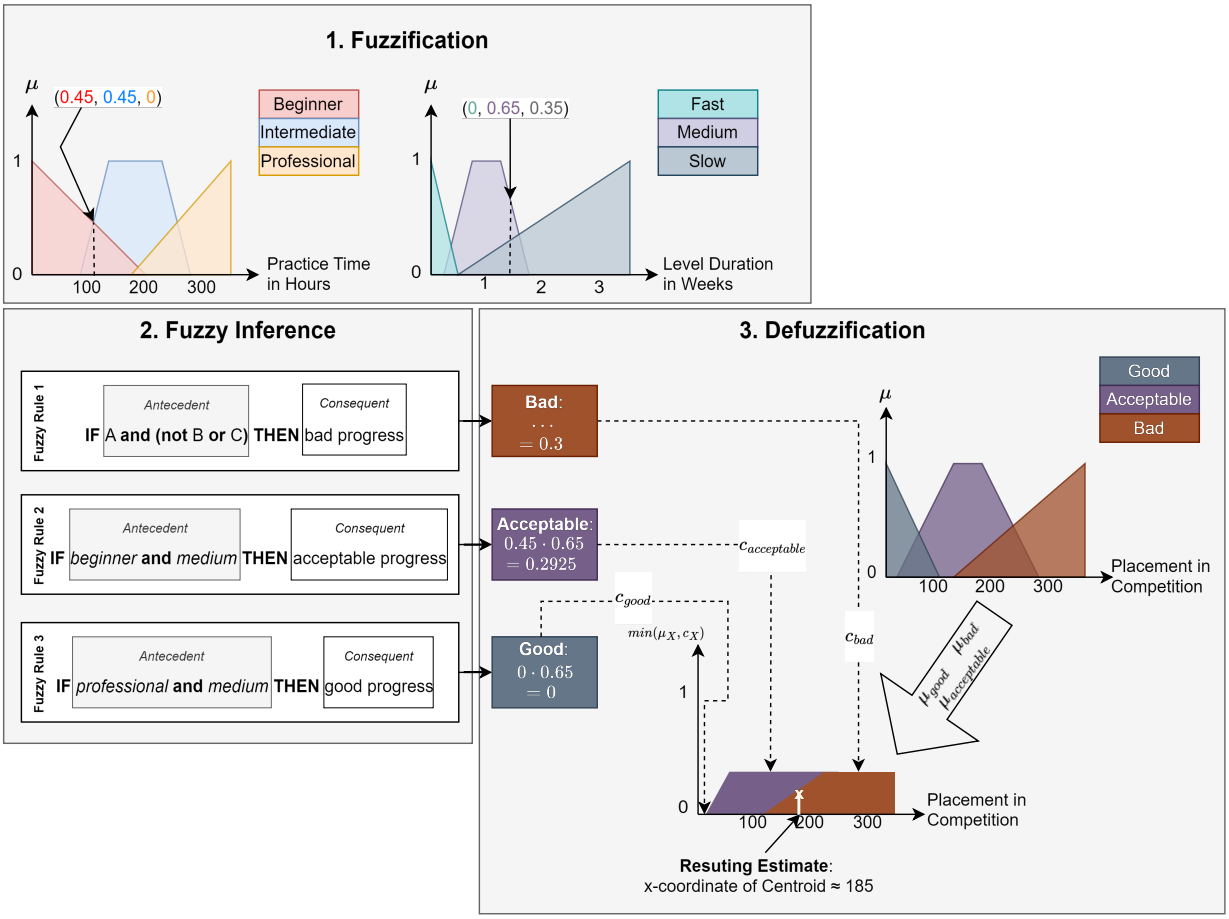


Figure 2.4: Mamdani FIS visualized with center-of-mass defuzzification.

2.2.4 Difference to TSK Systems

While the fuzzification step in TSK systems remains consistent, as elucidated by Takagi et al. [21], the primary distinction lies in the structure of the consequents within the rules. These consequents typically manifest as polynomial functions, yet they can also be simpler forms such as constants or linear combinations. Subsequently, these outputs undergo weighting by their corresponding antecedents.

Let the exemplary inputs "Practice Time in Hours" be denoted by h and the input "Level Duration in Weeks" be denoted by d. An example set of primitive TSK Rules which could replace step 2 and 3 of Figure 2.4 could be structured as follows:

- IF beginner and fast THEN placement $= 150 - h + 20 \cdot d$
- IF professional or medium THEN placement $= 30 - h$
- ...

Since the antecedent of the i-th rule will have the value $a_i \in [0, 1]$, TSK systems use this as the weight for the output value (i.e. consequent) of the i-th rule denoted as c_{output_i}.

To compute the output for placement is straightforward, since it is the only output of the TSK System but in theory there could be multiple, so let us define a set I that contains all indices of

rules that contain the output `placement` as $I = \{1, 2\}$ since we defined the first two rules to contain it as output above. Then the value of placement can be computed as follows:

$$\text{placement} = \frac{\sum_{i \in I} a_i \cdot c_{\text{placement}_i}}{\sum_{i \in I} a_i}$$

Note that the inference step here outputs crisp values and the defuzzification is more about combining these values fairly. Nonetheless, this additive expansion was proven to be powerful as B. Kosko [22] has shown that such fuzzy additive systems can

> ... uniformly approximate any real continuous function on a compact domain to any degree of accuracy.[2]

This is a strong theoretical capability, but as Mendel et al. [6] have highlighted, one limited factor certainly is the explainability, as humans cannot perform p-fold correlations for combining $p \in \mathbb{N}$ respective antecedents well enough to comprehend the possibly vast resulting rules. Besides, this assertion aligns with findings demonstrated across various architectures, such as Multi-Layer Perceptrons, as illustrated by G. Cybenko [23] emphasizing its status as a theoretical advantage rather than a pivotal property.

2.3 Explainable Representations of FIS

In a recent article, Cao et al. [2] have demonstrated four different visually engaging ways of displaying fuzzy rules and their results:

1. The simple and default way of listing the fuzzy rules in a linguistic manner as If-Then-Relationships.

2. For TSK-Systems the required antecedents with the MF and consequents can directly be listed similar to Table 2.1 where a zero-order TSK System (zero-order meaning the rules have constant outputs) with only one antecedent has its Gaussian MF defined by the stated mean μ and variance σ^2. This representation is also suitable for multiple independent consequents/outputs, where the b-th output is referred to as p^b in the table.

3. Visualizing the MF activations, the inference results and the defuzzification process for a concrete input example which can be performed similar to Figure 2.4.

4. Plotting a surface view for a subset of up to two input dimensions. This is exemplified in Figure 2.5 and comes with the disadvantages of a three-dimensional plot (occlusion, distortion from the perspective, etc.) but gives a clear and intuitive overview.

Table 2.1: Example TSK with Gaussian MF, p^b as b-th consequent (adapted from [2])

	Antecedent parameter 1	Consequent parameters
Rule 1	$\mu = 0.61; \sigma^2 = 0.5$	$p^0 = 0.589$
Rule 2	$\mu = 0; \sigma^2 = 1$	$p^0 = 0.195$
...	...	...
Rule k	$\mu = 12; \sigma^2 = 4$	$p^3 = 0.613$

[2] Abstract, [22]

[3] Image source: `https://github.com/tenaci-hand-grip/fuzzy-visualization`, accessed on April 3rd, 2024

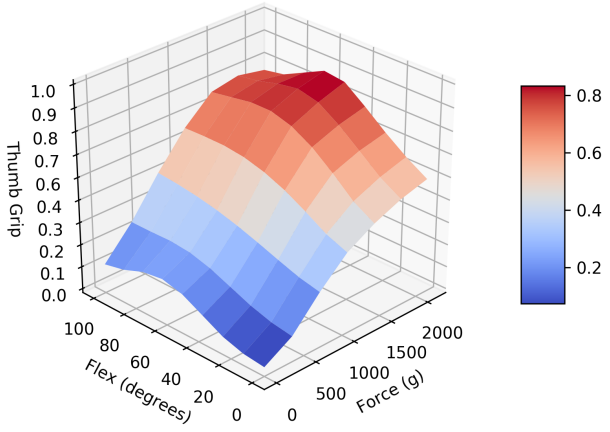


Figure 2.5: Example surface plot from an open-source fuzzy inference system[3]

Certainly, this compilation of potential visualizations is not comprehensive, and there exist a myriad of methods for generating domain-specific plots using the data derived from fuzzy inference systems. The above listed methods were provided solely to exemplify popular visualization techniques and are by no means exhaustive.

3 Fuzzy Rule-Based Regression

Now the focus transitions from having a single inference system with the goal of correctly predicting certain values to having a set of inference systems, each with certain rules to evaluate which (set of) rules are the most important for explaining a given dataset.

This implementation was adapted from the Section "Regression for Rule Selection" from Cichy et al. [7]. Let $g_1, \ldots, g_k : \mathbb{R}^d \to \mathbb{R}$ for some fixed number of inference systems $k \in \mathbb{N}$ and a fixed number of input dimensions $d \in \mathbb{N}$ denote a set of fuzzy inference systems as shown in Section 2.2 where each arbitrary g_i can be as simple as desired and the type of inference system is arbitrary as long as the final output is a crisp value. There might even only be one rule with one variable as antecedent in this inference system. In the context of this task, we call each g_i a *basis function* for the linear system we will specify shortly. Then we can form a linear combination of these individual basis functions as follows:

Let $\mathbf{x_i} \in \mathbb{R}^d$ be the i-th input vector (data sample) with d dimensions, assume there are n such samples to train on and let $\alpha_0, \ldots, \alpha_k \in \mathbb{R}$ be the coefficients to learn. The output for the data sample $\mathbf{x_i}$ can be modeled with a linear system as follows:

$$f(\mathbf{x_i}, \alpha_0, \alpha_1, \ldots, \alpha_n) = \alpha_0 + \alpha_1 g_1(\mathbf{x_i}) + \cdots + \alpha_k g_k(\mathbf{x_i}) \quad \forall i \in 1, \ldots, n \tag{3.1}$$

Note that the coefficients $\alpha_0, \ldots, \alpha_k$ are independent of the i-th input, although the input vector x changes. We call this a linear combination as the model is linear in its parameters which are the coefficients. All inference systems g_i are assumed to be constant in solving this problem and only depend on the input.

To find the best coefficients, the authors used ordinary least squares as the optimization criterion which sums the squared deviation of the model output to the respective ground truth label y_i. This sum for all data samples will be denoted as Sum of Squared Errors (SSE). So the following expression should be minimized for all data points:

$$SSE = \sum_{i=1}^{n} (y_i - f(\mathbf{x_i}, \alpha_0, \alpha_1, \ldots, \alpha_n))^2 \tag{3.2}$$

After performing significance testing by leaving out one basis function and checking if the performance deviation is statistically significant (see Section 3.2), the above coefficients can be interpreted as the importance of the corresponding rules.

The above minimization problem is easy to state but solving it requires some more intricate thoughts. Let us now discover how such a linear regression least-squares problem can be solved.

3.1 Analyzing Resolution Methodologies for General Linear Least Squares

Press et al. [24] provided clear approaches for getting potential solutions for such and more sophisticated optimization problems. Hence, for minimizing Equation (3.2), we will first introduce the standard method of using the so-called *normal equations*. Before we explore this method, it is vital to notice that the described method only sets the first derivative to zero, which is only a necessary condition for an optimum but not sufficient. If (few) finitely many potential minimizers are returned from this method, the corresponding function values can be compared to find the best one. However, in more complex settings with intricate non-linear basis functions, finding a global minimum might require the use of more advanced methods.

3.1.1 Solving With Normal Equations

We will first extend our notation for convenience in the proceeding parts: g_{ab} will now denote the output of the a-th basis function for sample with index b. This way we can set the derivative of the overall SSE to zero in a more compact expression, which is shown the following derivations.

To get the potential solutions for this optimization problem with calculus, we take the derivative with respect to α_j for all $j \in \{0, 1, \ldots, k\}$ and since there are $k+1$ coefficients of this linear problem to be found, we will find $k+1$ constraints. We handle this in two distinct cases: If $j \neq 0$ we get:

$$\frac{\partial SSE}{\partial \alpha_j} = -2 \sum_{i=1}^{n} ((y_i - \alpha_0 - \alpha_1 g_{1i} - \cdots - \alpha_k g_{ki}) g_{ji}) = 0$$

Which can be rewritten into the following constraint for $j \in \{1, 2, \ldots, k\}$:

$$\sum_{i=1}^{n} y_i g_{ji} = \alpha_0 \sum_{i=1}^{n} g_{ji} + \alpha_1 \sum_{i=1}^{n} g_{1i} g_{ji} + \cdots + \alpha_k \sum_{i=1}^{n} g_{ki} g_{ji}$$

For $j = 0$, that means we want to compute the intercept a_0, we get the slightly simpler necessary condition:

$$\frac{\partial SSE}{\partial a_0} = -2 \sum_{i=1}^{n} (y_i - \alpha_0 - \alpha_1 g_{1i} - \cdots - \alpha_k g_{ki}) = 0$$

Similarly, we get the constraints:

$$\sum_{i=1}^{n} y_i = n\alpha_0 + \alpha_1 \sum_{i=1}^{n} g_{1i} + \cdots + \alpha_k \sum_{i=1}^{n} g_{ki}$$

These equations can be nicely assembled into a matrix equation, which in this context is known as the *normal equation*

$$\mathbf{G}^T \mathbf{G} \mathbf{a} = \mathbf{G}^T \mathbf{y}$$

where

- $\mathbf{G}$ is the matrix with the i-th row of the form $(1, g_{1i}, \ldots, g_{ki})$,
- $\mathbf{a}$ is the vector of coefficients $(\alpha_0, \alpha_1, \ldots, \alpha_k)^T$,
- $\mathbf{y}$ is the vector of target values $(y_1, y_2, \ldots, y_n)^T$.

Note: One should keep in mind that the matrix $\mathbf{G}$ can be fully pre-computed, since we evaluate the fuzzy inference j at the sample i which has to be a real number by our definition of fuzzy inference systems.

So if we can find $\mathbf{a}$ we have achieved our goal. To perform this task, one can re-arrange the above equation and solve the following system:

$$\mathbf{a} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{y}$$

This assumes that the resulting square matrix ($\mathbf{G}^T\mathbf{G}$) is invertible. If this is the case, we will obtain a unique solution for $\mathbf{a}$ and if we construct this problem in software nicely, we can ensure that the matrix is invertible: If we force $\mathbf{G}$ to be full-rank, that means that, depending on the shape of $\mathbf{G}$, all the columns or all the rows have to be linearly independent, then $\mathbf{G}^T\mathbf{G}$ will have linearly independent columns as well by the definition of matrix multiplication as being the inner-product of rows and columns.

Since we commonly expect $\mathbf{G}$ to have more rows than columns (more samples than fuzzy inference systems), one can check for redundancies/copies as a pre-processing step in the columns and drop such inference systems before applying the optimization. Nonetheless, many solutions might exist (as described in Section 3.1) and should be tried to find the true global optimum.

3.1.2 Solving With QR-Decomposition

While the matrix inversion in Section 3.1.1 provides a more direct method to find the least squares solution to the said optimization problem, this approach may not always be numerically stable, especially when the matrix $\mathbf{G}^T\mathbf{G}$ is (close to) singular or ill-conditioned. An alternative, more robust method involves the use of the QR-Decomposition as was also highlighted by Press et al. [24].

QR-Decomposition is a matrix factorization technique that decomposes a matrix into a product of an orthogonal matrix $\mathbf{Q}$ and an upper triangular matrix $\mathbf{R}$. Specifically, for our matrix $\mathbf{G} \in \mathbb{R}^{n \times m}$ we can write:

$$\mathbf{G} = \mathbf{QR}$$

where $\mathbf{Q}$ is an $n \times n$ orthogonal matrix (by definition $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, the identity matrix) and $\mathbf{R}$ is an $n \times m$ upper triangular matrix. Here, n is the number of observations and $m = k + 1$ is the number of coefficients including the intercept. Furthermore, we assume that $n \gg m$ since more samples than basis functions are expected for most use-cases.

The similar system we aim to solve here (with the same variables defined in Section 3.1.1), $\mathbf{Ga} = \mathbf{y}$, can be rewritten using QR-Decomposition as:

$$\mathbf{QRa} = \mathbf{y}$$

By multiplying both sides by $\mathbf{Q}^T$, we leverage the orthogonality of $\mathbf{Q}$ ($\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$) to get:

$$\mathbf{Ra} = \mathbf{Q}^T\mathbf{y}$$

Since $\mathbf{R}$ is an upper triangular matrix, this system can be solved efficiently through back substitution, avoiding the numerical instability issues that can arise from directly inverting $\mathbf{G}^T\mathbf{G}$. Nonetheless, we still have to be careful to choose a numerically stable algorithm for computing the QR-Decomposition, where a prominent example would be so-called Householder-Transformations as outlined by Golub et al. [25].

3.2 Testing Significance Levels of the Coefficients

In the last step of the fuzzy-approximation publication by Cichy et al. [7] a statistical test was applied to find which basis functions are not statistically significant in their contribution and could be omitted without impacting the model's predictive power. As was confirmed by one of the authors after a personal inquiry, this test was most likely a Likelihood Ratio Test (LRT).

The LRT offers a statistical metric for the said evaluation, where we will compare two models: the full model, which includes all predictors (basis functions), and a reduced model, from which one or more predictors are excluded. The null hypothesis (H_0) can be phrased as a certain basis function g_i not being necessary to maintain the descriptive power. If the test statistic Λ from Equation 3.3 exceeds a critical value Λ_{crit}, we reject H_0, which certainly means that g_i is necessary from a statistical perspective (up to a certain significance level). If we cannot reject H_0, we obtain no formal statistical knowledge for being able to remove g_i without reducing the predictive power of the model from this test, but we will use it as an informal indication in this scenario.

The LRT statistic for this ordinary least squares scenario is defined as follows:

$$\Lambda = 2(\log(\mathcal{L}_{\text{full}}) - \log(\mathcal{L}_{\text{reduced}})) \tag{3.3}$$

where $\mathcal{L}_{\text{full}}$ and $\mathcal{L}_{\text{reduced}}$ are the likelihoods of the full and reduced models, respectively. The value of the test statistic Λ approximately follows a χ^2-distribution as shown by Chvosteková et al. [26] where we adopt the assumption that the errors are normally distributed. The degrees of freedom for this distribution are determined by the difference in the number of parameters between the full and reduced models. For instance, if the reduced model has one fewer parameter than the full model, the degree of freedom would equal one.

It should be noted that popular regression libraries like *statsmodels* in Python typically make assumptions for the estimation of the likelihoods, as they frequently assume Gaussian distributions on the error distributions as an approximation for the underlying true error distribution. This would then result in a maximum likelihood estimation for the variance of $\sigma^2 = SSE/n$, where SSE is the sum of the squared errors and n is the number of tested samples as can be examined in more detail in the work by Lehmann et al. [27] in Chapter 12.4.1. Let us quickly derive the Ordinary Least Squares Likelihood with the assumed Gaussian distribution by using the tools provided by Lehmann et al. [27]:

Let $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ be the vector containing all labels, $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_k)$ be the vector containing all parameters and $\boldsymbol{X}$ be the matrix containing the rows as $(1, g_{1i}, g_{2i}, \ldots, g_{ki})$ where g_{ai} represents the a-th fuzzy inference system applied on sample i. Then for some assumed variance σ^2 under the assumption of homoscedasticity, we get the following likelihood for the linear regression model parameterized by $\boldsymbol{\alpha}$:

$$\mathcal{L}(\boldsymbol{\alpha}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\alpha})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\alpha})\right)$$

Taking the natural logarithm of the above expression and applying the corresponding calculation rules:

$$\log \mathcal{L}(\boldsymbol{\alpha}, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\alpha})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\alpha})$$

Now, we notice that the last term contains the SSE and simplify:

$$\log \mathcal{L}(SSE, \sigma^2) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}SSE$$

Finally, we substitute the above MLE of the variance $\sigma^2 = SSE/n$ and see that the resulting formula for the log likelihood of a model under the given assumptions is:

$$\log \mathcal{L}(SSE, n) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\left(\frac{SSE}{n}\right) - \frac{n}{2}$$

The computed Λ value is then compared to a critical value from the χ^2-distribution, given a significance level (usually 0.05) and as stated above, an indication of being able to prune g_i is given. This also improves the interpretability of the underlying model by being more overseeable as fewer basis functions are used.

4 Conclusion and Acknowledgements

In this exploration of fuzzy logic's ability to supply explanations, we started with its simple mathematical concepts to address vagueness and imprecision. This lead us to unveiling the inner-workings behind the linguistic formulations provided by FIS to improve interpretability of AI systems and data generation processes. We have observed that FIS are inherently explainable models which can mimic human reasoning regarding certain tasks on a high level by careful (partially manual) modeling of such processes. Therefore, fuzzy logic is a promising candidate for providing models in critical domains like healthcare, finance or cybersecurity.

It is vital to recognize that our exploration has merely scratched the surface of the possibilities which modern fuzzy-based systems provide. Different established techniques such as Forward Step-Wise Regression (as demonstrated by Mendel et al. [6]) and Adaptive Neuro-Fuzzy Inference Systems (as outlined by J.R. Jang [28]) exemplify two popular of many methods and the high degree of sophistication in this field of study. By their intrinsic structure, such systems also have outstanding capabilities in explaining/unveiling certain relationships or concrete biases in large datasets.

Acknowledgements

Last but not least, sincere gratitude is owed to Univ.-Prof. Priv.Doz. DDI Dr. Rass for his willingness to promply address inquiries, furnish clarifications, and contribute invaluable insights, which have greatly enhanced this academic endeavor.

5 Acronyms

AI Artificial Intelligence

DL Deep Learning

FIS Fuzzy Inference Systems

LRT Likelihood Ratio Test

MF Membership Function

ML Machine Learning

SSE Sum of Squared Errors

TSK Takagi-Sugeno-Kang

XAI eXplainable Artificial Intelligence

References

[1] W. Ding, M. Abdel-Basset, H. Hawash, and A. M. Ali, "Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey," *Information Sciences*, vol. 615, pp. 238–292, Nov. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S002002552201132X

[2] J. Cao, T. Zhou, S. Zhi, S. Lam, G. Ren, Y. Zhang, Y. Wang, Y. Dong, and J. Cai, "Fuzzy inference system with interpretable fuzzy rules: Advancing explainable artificial intelligence for disease diagnosis—A comprehensive review," *Information Sciences*, vol. 662, p. 120212, Mar. 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025524001257

[3] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8466590/

[4] A. Singh, S. Sengupta, and V. Lakshminarayanan, "Explainable Deep Learning Models in Medical Image Analysis," *Journal of Imaging*, vol. 6, no. 6, p. 52, Jun. 2020. [Online]. Available: https://www.mdpi.com/2313-433X/6/6/52

[5] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: http://link.springer.com/10.1023/A:1010933404324

[6] J. M. Mendel and P. P. Bonissone, "Critical Thinking About Explainable AI (XAI) for Rule-Based Fuzzy Systems," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 12, pp. 3579–3593, Dec. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9430516/

[7] C. Cichy and S. Rass, *A Fuzzy-Approximation-Approach to Explainable Information Quality Assessment.* International Business Information Management Association (IBIMA), 2019, pp. 3919–3931.

[8] ——, "Fuzzy expert systems for automated data quality assessment and improvement processes," in *Proceedings of the CEUR Workshop*, vol. 2751, 2020, pp. 7–11. [Online]. Available: https://ceur-ws.org/Vol-2751/short2.pdf

[9] X. Zhu, D. Wang, W. Pedrycz, and Z. Li, "Fuzzy Rule-Based Local Surrogate Models for Black-Box Model Explanation," *IEEE Transactions on Fuzzy Systems*, vol. 31, no. 6, pp. 2056–2064, Jun. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9933617/

[10] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S001999586590241X

[11] Mamdani, "Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis," *IEEE Transactions on Computers*, vol. C-26, no. 12, pp. 1182–1191, Dec. 1977. [Online]. Available: http://ieeexplore.ieee.org/document/1674779/

[12] S. Y. C, J. Ebienazer, S. M, and S. S, "Fuzzy logic," *International Journal of Innovative Research in Information Security*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259691169

[13] Qilian Liang and J. Mendel, "An introduction to type-2 TSK fuzzy logic systems," in *FUZZ-IEEE'99. 1999 IEEE International Fuzzy Systems. Conference Proceedings (Cat.*

No.99CH36315). Seoul, South Korea: IEEE, 1999, pp. 1534–1539 vol.3. [Online]. Available: http://ieeexplore.ieee.org/document/790132/

[14] J. Mendel and R. John, "Type-2 fuzzy sets made simple," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 117–127, Apr. 2002. [Online]. Available: http://ieeexplore.ieee.org/document/995115/

[15] A. Hamam and N. D. Georganas, "A comparison of Mamdani and Sugeno fuzzy inference systems for evaluating the quality of experience of Hapto-Audio-Visual applications," in *2008 IEEE International Workshop on Haptic Audio visual Environments and Games*. Ottawa, ON, Canada: IEEE, Oct. 2008, pp. 87–92. [Online]. Available: http://ieeexplore.ieee.org/document/4685304/

[16] K. Wang, "Computational Intelligence in Agile Manufacturing Engineering," in *Agile Manufacturing: The 21st Century Competitive Strategy*. Elsevier, 2001, pp. 297–315. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9780080435671500164

[17] L. A. Zadeh, "A Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges," *Journal of Cybernetics*, vol. 2, no. 3, pp. 4–34, Jan. 1972. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01969727208542910

[18] D. K. Wedding, "Fuzzy sets and fuzzy logic: Theory and applications: George j. klir and bo yuan, prentice hall, 1995, isbn 0-13-101171-5, pp. 574," 1997. [Online]. Available: https://api.semanticscholar.org/CorpusID:62557335

[19] E. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020737375800022

[20] K. Saadaoui, B. Bouderah, O. Assas, and M. Khodja, "Type-1 and Type-2 fuzzy Sets to Control a Nonlinear Dynamic System," *Revue d'Intelligence Artificielle*, vol. 33, no. 1, pp. 1–7, May 2019. [Online]. Available: http://www.iieta.org/journals/ria/paper/10.18280/ria.330101

[21] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985. [Online]. Available: http://ieeexplore.ieee.org/document/6313399/

[22] B. Kosko, "Fuzzy systems as universal approximators," *IEEE Transactions on Computers*, vol. 43, no. 11, pp. 1329–1333, Nov. 1994. [Online]. Available: http://ieeexplore.ieee.org/document/324566/

[23] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, Dec. 1989. [Online]. Available: http://link.springer.com/10.1007/BF02551274

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge, USA: Cambridge University Press, 1992. [Online]. Available: https://dl.acm.org/doi/10.5555/148286

[25] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[26] M. Chvosteková and V. Witkovský, "Exact likelihood ratio test for the parameters of the linear regression model with normal errors," 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:25195916

[27] E. L. Lehmann and J. P. Romano, *Testing statistical hypotheses*, 3rd ed., ser. Springer texts in statistics. New York: Springer, 2005.

[28] J.-S. R. Jang, "Anfis: adaptive-network-based fuzzy inference system," *IEEE Trans. Syst. Man Cybern.*, vol. 23, pp. 665–685, 1993. [Online]. Available: https://api.semanticscholar.org/CorpusID:14345934