Fuzzy Logic Concepts, Developments and Implementation

SAATCHI, Reza <http://orcid.org/0000-0002-2266-0187>

Published version

SAATCHI, Reza (2024). Fuzzy Logic Concepts, Developments and Implementation. Information, 15 (10): 656. [Article] information

MDPI

Article

Fuzzy Logic Concepts, Developments and Implementation

Reza Saatchi

School of Engineering and Built Environment, City Campus, Sheffield Hallam University, Sheaf Building, Howard Street, Sheffield S1 1WB, UK; r.saatchi@shu.ac.uk

Abstract: Over the past few decades, the field of fuzzy logic has evolved significantly, leading to the development of diverse techniques and applications. Fuzzy logic has been successfully combined with other artificial intelligence techniques such as artificial neural networks, deep learning, robotics, and genetic algorithms, creating powerful tools for complex problem-solving applications. This article provides an informative description of some of the main concepts in the field of fuzzy logic. These include the types and roles of membership functions, fuzzy inference system (FIS), adaptive neuro-fuzzy inference system and fuzzy c-means clustering. The processes of fuzzification, defuzzification, implication, and determining fuzzy rules' firing strengths are described. The article outlines some recent developments in the field of fuzzy logic, including its applications for decision support, industrial processes and control, data and telecommunication, and image and signal processing. Approaches to implementing fuzzy logic models are explained and, as an illustration, Matlab (version R2024b) is used to demonstrate implementation of a FIS. The prospects for future fuzzy logic developments are explored and example applications of hybrid fuzzy logic systems are provided. There remain extensive opportunities in further developing fuzzy logic-based techniques, including their further integration with various machine learning algorithms, and their adaptation into consumer products and industrial processes.

Keywords: fuzzy logic; fuzzy inference system; adaptive neuro-fuzzy inference system; fuzzy c-means clustering; fuzzy logic applications; fuzzy logic implementation; hybrid fuzzy logic models

Citation: Saatchi, R. Fuzzy Logic Concepts, Developments and Implementation. *Information* **2024**, *15*, 656. https://doi.org/10.3390/info15100656

Academic Editor: Maria Carmen Carnero

Received: 11 September 2024
Revised: 11 October 2024
Accepted: 17 October 2024
Published: 19 October 2024

1. Introduction

The term logic could be described as the study of correct reasoning [1] while reasoning could be defined as the process of drawing conclusions from the available information. The term fuzzy is associated with uncertainty in a process or data and fuzzy logic is an artificial intelligence technique that uses linguistic terms to perform reasoning and thus facilitates analysis and interpretation of imprecise information [2–4]. Using linguistic terms in fuzzy logic may reduce the complexities of system modelling as there could be less need for extensive mathematical formulations. The field of fuzzy logic and its applications have evolved greatly in the last few decades with numerous reported publications and applications [5]. Fuzzy logic is distinct from probability theory as the latter is generally applied to problems associated with random characteristics. Probability theory can have limitations in scenarios where the problem is termed in vague linguistic terms or cases where the available information could be imprecise.

Crisp or classical sets can be considered as special cases of fuzzy sets. If A is a crisp set and U is a universal set (or universe of discourse set or a set that contains all elements of other sets, including its own elements), then for any element x in U, x is either a full member of A (membership = 1) or not a member at all (membership = 0). In general, a crisp set A can be described by its characteristic function $\mu_A(x)$ as

$$\mu_A(x) = 1\ if\ x \in A$$

$$\mu_A(x) = 0\ if\ x \notin A$$

where the symbols ∈ and ∉ represent "is a member" and "is not a member", respectively. For instance, if the speed of the cars on a road is limited to 60 mph, then three crisp sets could be defined representing low, medium and high speeds as shown in Figure 1.

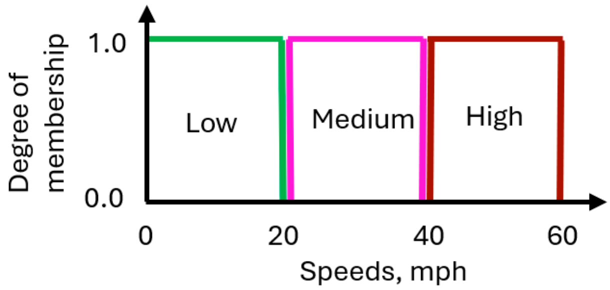

Figure 1. Representations of Low, Medium and High speeds crisp (classical) sets.

For the crip sets in Figure 1, a car speed of 39.9 mph is a member of the medium-speed set even though it is not significantly different from 41.1 mph which is a member of the high-speed set. In fuzzy sets, the membership of x to a set is no longer binary or true or false, as x can simultaneously be a member of multiple sets with associated degrees of memberships. Zadeh, in 1965, presented a seminal paper about fuzzy sets [2] that provided an important foundation of the field. Zadah's work and those of others have resulted in the evolution of the field, extensive publications and several applications. As shown in Figure 2, fuzzy sets typically overlap; thus, a speed of 37 mph can simultaneously be a member of medium- and high-speed sets with degrees of memberships of 0.50 and 0.25, respectively. The degree of membership indicates the extent that an element x belongs to a set. It ranges between 0 and 1, where 0 represents not a member and 1 a full member.

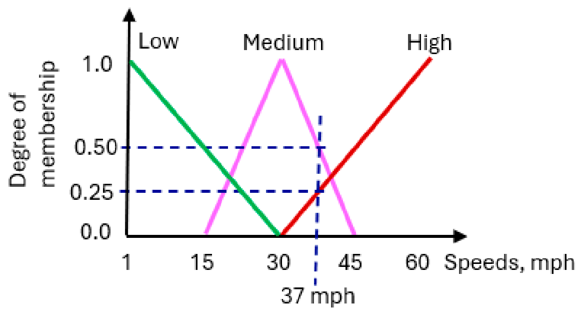

Figure 2. Representations of Low, Medium and High speeds fuzzy sets.

In general, if x is a member of fuzzy set A, its mapping can be expressed as

$$\mu_A(x) \in [0,1]$$

$$A = \ (x, \mu_A(x) | x \in U)$$

This mapping for a finite discrete fuzzy set can be represented as

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \cdots = \sum \frac{\mu_A(x_i)}{x_i} \quad (1)$$

where the symbol "+" in expression (1) is not a mathematical addition or logical OR but it represents an aggregation or collection operation. For a finite continuous U, the fuzzy set A can be represented as

$$A = \int_X \frac{\mu_A(x)}{x}\, d(x) \quad (2)$$

The contribution of this article is informative explorations of the main fuzzy logic concepts, combined with discussions of recent developments in the field, and system modelling implementation coverage. The manner of preparing this article was to ensure prior information about the field was not required to be able to follow up and understand its contents. Membership functions play a central role in the field of fuzzy logic as they facilitate conversion between crip and fuzzy data. Therefore, the main membership functions are presented, and their features are explained. Several fuzzy logic applications were based on fuzzy inference system (FIS), adaptive neuro-fuzzy inference system (ANFIS) and fuzzy c-means techniques. Therefore, the operations of these techniques are explained. A discussion of recent fuzzy logic applications associated with industrial processes and control, decision support, data and telecommunication, and signal and image processing are provided. Although the coverage of all publications in the fuzzy logic fields was not practical, the aim was to provide a representative coverage of the developments. In the following sections, a description of the main fuzzy logic concepts is provided and some of the developments in the field are outlined. A section on implementing fuzzy logic systems is included and the manner FIS could be implemented in Matlab [6] is illustrated through an example.

2. Fuzzy Logic Concepts

In this section, the main fuzzy logic concepts are explained.

2.1. Membership Functions

A design consideration in developing a fuzzy logic system is the types and parameters of the membership functions as they characterise fuzzy sets. A study provided a review of issues associated with membership functions [7]. Some popular membership functions are described in this section. In Figure 2, triangular membership functions are used to represent the fuzzy sets for different speeds. As indicated in Figure 3, a triangular membership function is specified by the parameters a, b and c, where $a < b < c$.

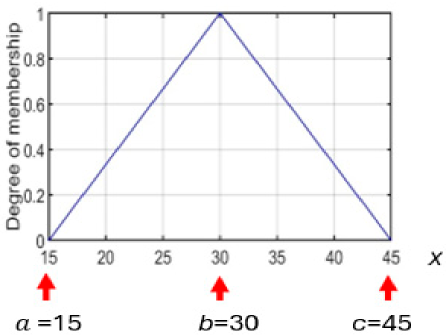

Figure 3. A triangular membership function.

The parameters a and c indicate the starting and end points of base of the triangle and b corresponds to the point on the base of the triangle associated with its peak. The triangular membership function can be symmetric (as in Figure 3) or it can be asymmetric. It can be expressed as

$$\mu_{Triangular}(x) = max\left(min\left(\frac{x-a}{b-a}, 1, \frac{c-x}{c-b}\right), 0\right) \quad (3)$$

where *max* and *min* operators represent the maximum and minimum values, respectively. For example, to determine the degree of membership of $x = 20$ to the triangular membership function (i.e., $\mu_{Triangular}(20)$), shown in Figure 3, the following calculations can be performed.

$$\mu_{Triangular}(20) = max\left(min\left(\frac{20-15}{30-15}, 1, \frac{45-20}{45-30}\right), 0\right)$$

$$\mu_{Triangular}(20) = max\left(min\left(\frac{5}{15}, 1, \frac{25}{15}\right), 0\right)$$

$$\mu_{Triangular}(20) = max\left(min\left(\frac{1}{3}, 1, \frac{5}{3}\right), 0\right)$$

$$\mu_{Triangular}(20) = max\left(\frac{1}{3}, 0\right) = \frac{1}{3}$$

The trapezoidal membership function, shown in Figure 4, can provide greater flexibility for some applications as compared to the triangular membership function.

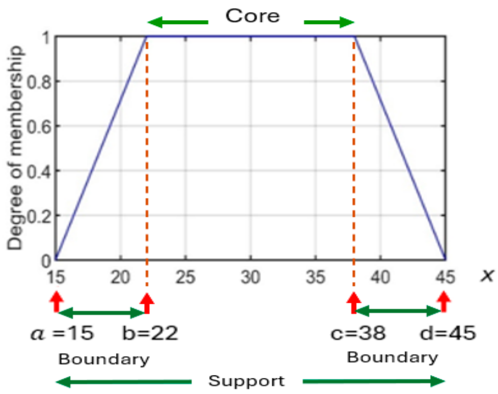

Figure 4. A trapezoidal membership function.

The trapezoidal membership function is characterised by its parameters a, b, c and d, ($a < b < c < d$). The values of a and d indicate the points on the x-axis for the starting and end points of its base and b and c indicate the x-values for the starting point and end point of its core (shown in Figure 4). For a fuzzy set A, its core, denoted as *Core(A)*, is defined as

$$Core(A) = \{x|\ \mu_A(x) = 1\ and\ x \in U\} \tag{4}$$

The support of the trapezoidal membership function, denoted as *Support(A),* is defined as

$$Support(A) = \{x|\ \mu_A(x) > 0\ and\ x \in U\} \tag{5}$$

Like the triangular membership function, the trapezoidal membership function can also be symmetric (as in Figure 4) or asymmetric. It is expressed as

$$\mu_{Trapezoidal}(x) = max\left(min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \tag{6}$$

The Gaussian membership function (shown in Figure 5) is characterised by its centre (c) and width (σ) and is expressed as

$$\mu_{Gaussian}(x) = e^{-0.5\left(\frac{x-c}{\sigma}\right)^2} \tag{7}$$

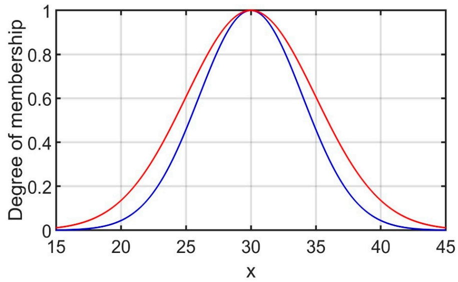

Figure 5. Gaussian membership functions for $\sigma = 4$ (blue plot) and for $\sigma = 5$ (red plot).

The generalised bell membership function, shown in Figure 6, is characterised by the parameters a, b and c and is expressed as

$$\mu_{Bell}(x) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}} \tag{8}$$

The parameter c determines its centre, a is its half width, and b together with a, controls the slope at the crossover points (slope = $-\frac{b}{2a}$).

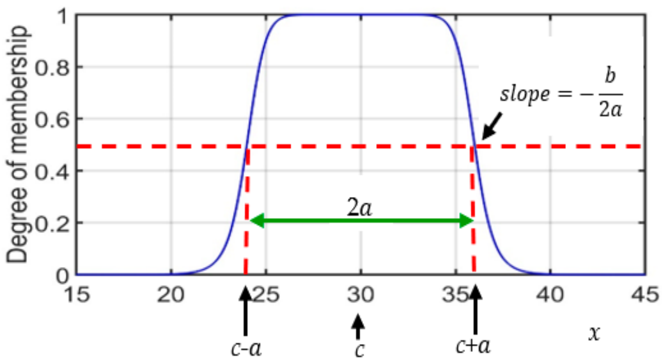

Figure 6. Generalised bell membership function with $a = b = 6$ and $c = 30$ (the red horizontal dashed line is at $\mu_{Bell}(x) = 0.5$).

As indicated in Figure 7, an increase in a widens the generalised bell membership function.

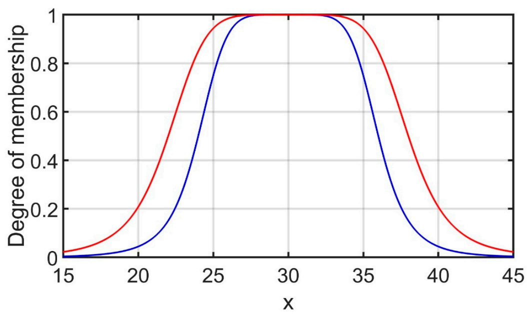

Figure 7. Generalised bell membership functions with $b = 3$ and c = 30, blue: $a = 6$, red: a=8.

The sigmoid membership function, shown in Figure 8, is controlled by parameters a and b, where a defines the slope at $x = b$, where b is the inflection point. The symbol e is a mathematical constant, approximately equal to 2.71828. The sigmoid is expressed as

$$\mu_{Sigmoid}(x) = \frac{1}{1+e^{-a(x-b)}} \tag{9}$$

As shown in Figures 8 and 9, the sign of a determines whether this membership function opens to the right (when a is positive) or left (when a is negative).

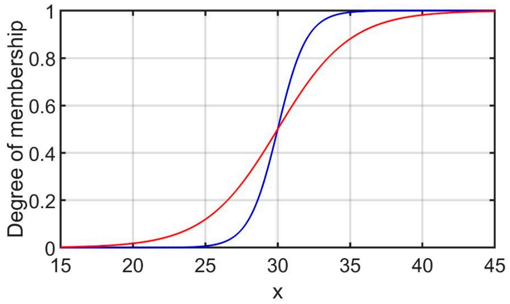

Figure 8. Sigmoid membership functions with b = 30, blue plot a =1, red plot a=0.4.

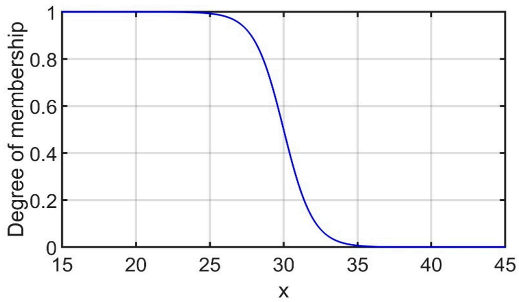

Figure 9. Sigmoid membership functions with b = 30 and a = −1.

Although this section focused on the main membership functions, there were also studies reporting the design of dynamic membership functions [8]. The dynamism could for example be achieved through automatic updates in the degrees of memberships, the ranges of membership functions, and the extent of their overlaps. It is also possible to design membership functions by considering the data being analysed [9,10]. The approaches for these could be based on heuristics, the probability to possibility transformations, histograms, nearest neighbour, artificial neural networks, clustering, and mixture decomposition [9]. Approaches such as fuzzy clustering [11] and maximum entropy [12] were also devised to design membership functions.

2.2. Fuzzy Rules and Operators

A fuzzy logic model usually has multiple inputs. For example, if the fuzzy logic model is assessing the severity (output of the model) of a car accident with another car or a pedestrian, it may have as its input the car's speed and its distance to the other car or the pedestrian as inputs, as shown in Figure 10.

Figure 10. Fuzzy logic model to assess a car accident severity.

The operators AND, OR and NOT are used to combine the associated logic combinations relating the input(s) and output(s) fuzzy sets as shown in Figure 11.

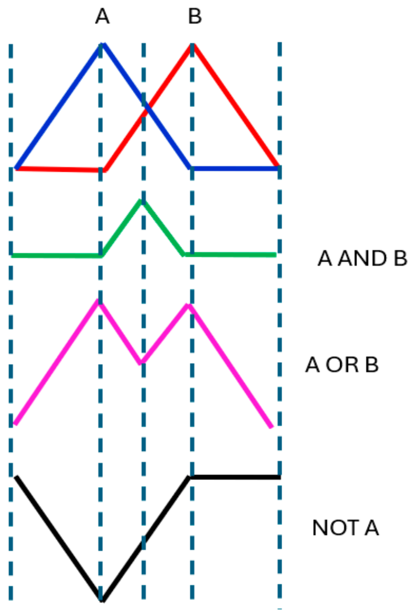

Figure 11. Illustration of the fuzzy logical operators, AND, OR and NOT on fuzzy sets *A* and *B*.

The AND logic operator corresponds to the intersection of the sets. The degree of truth "*A* AND *B*" is the minimum (min) value of the degrees of truth of the sets *A* and *B*, i.e.,

$$\mu(A \text{ AND } B) = \min(\mu(\mathrm{A}), \mu(\mathrm{B})) \tag{10}$$

The logic operator OR corresponds to the union of the sets. Therefore, the degree of truth of "*A* OR *B*" is the maximum (max) value of the degrees of truth of *A* and *B*, i.e.,

$$(A \text{ OR } B) = \max(\mu(A), \mu(B)) \tag{11}$$

The complement or NOT operator of a set indicates its negation, i.e.,

$$\mu(\text{NOT } A) = 1 - \mu(A) \tag{12}$$

A list of the main set operations and their properties is included in Appendix A. The inferencing operation in a fuzzy logic model requires adaptation of the domain's knowledge that is typically formulated by a series of IF-THEN structured rules. The IF-THEN rules allow the formulation of conditional statements for a fuzzy logic model. The domain knowledge can be gained from an expert or by experimentation. A rule is "activated" (or "fired") when its inputs conditions, i.e., its IF part (or the antecedent) are satisfied, resulting in the implementation of the rule's THEN part (i.e., the consequence or conclusion part). Given the fuzzy sets *A*, *B* and *C*, a fuzzy rule could be expressed as:

IF $x \in A$ and $y \in B$ THEN $z \in C$.

For example,

IF Speed High AND Distance Small THEN Severity High.

For the above rule, for the IF part, "High" represents a fuzzy set for Speed and "Small" represents a fuzzy set for Distance and in the THEN part, "High" represents a fuzzy set for the car accident Severity.

2.3. Fuzzy Inference System (Mamdani-Type)

In this section, the focus is on the Mamdani-type fuzzy inference system (FIS) due to its popularity. A FIS has four main elements as shown in Figure 12.

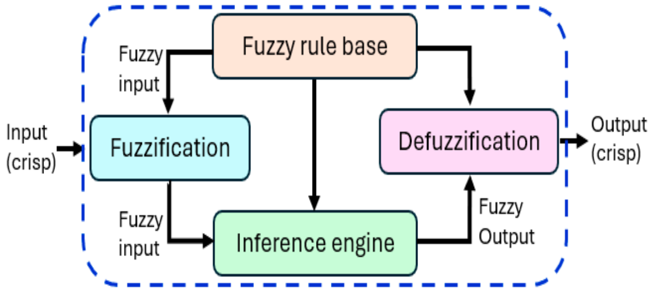

Figure 12. The structure of a fuzzy inference system (FIS).

Fuzzification element: The input(s) to the FIS is crip, i.e., numerical values obtained from some sensors or various forms of measured data. The fuzzification element uses the inputs membership functions to convert crip inputs into linguistic terms characterised by their associated fuzzy sets. This would indicate the degrees of membership of each input to the associated input fuzzy sets. For example, in Figure 2, when the speed of a car is 37 mph, this speed belongs to both the medium-speed (degree of membership 0.50) and high-speed sets (degree of membership 0.25).

Fuzzy rule base element: The fuzzy rule base or the knowledge base element interprets its fuzzified input(s) to facilitate inferencing. It consists of linguistic rules associated with the domain knowledge represented in the form of "IF conditions (premise) are satisfied THEN the consequences are inferred". The number of rules incorporated depends on the nature and complexity of the model.

Inference engine element: The inference engine element uses the fuzzified inputs and the information from the fuzzy rule base to draw conclusions and infer fuzzy control actions. This process initially involves determining the strength of the premise of each rule (this is also known as the firing strength) [13] and applying the rule's implication. The associated operations are illustrated using the car accident severity example, where the inputs to the model are the car's speed represented by the fuzzy sets shown in Figure 2 and the car distance (to another car or a pedestrian), represented by the fuzzy sets shown in Figure 13. The output of the model is the car accident severity, represented by the fuzzy sets shown in Figure 14.

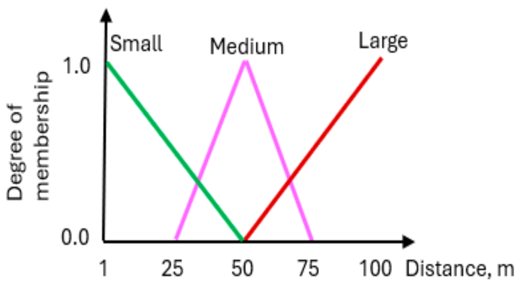

Figure 13. The fuzzy sets for the distance input to the FIS.

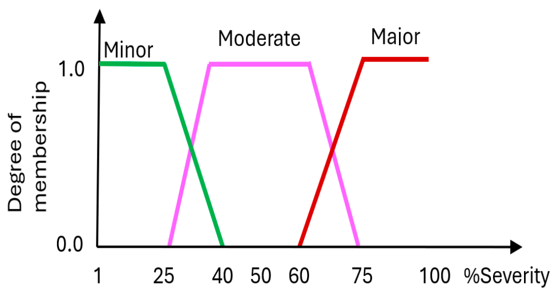

Figure 14. The fuzzy sets for the severity of car accident.

For an illustrative purpose, only two rules are considered:

Rule 1: IF Speed Low AND Distance Large THEN Severity Minor.
Rule 2: IF Speed High OR Distance Small THEN Severity Major.

The minimum (min) and maximum (max) descriptors are used to implement the AND and OR operators, respectively. It is possible to give a unique weighing to each rule to indicate their significance to the overall FIS output(s) relative to the other rules. In this example, both rules have an equal weighting (i.e., weighting = 1). If a rule has a firing strength of 0, it does not affect the FIS output(s). Assuming the inputs to the FIS are Speed = 15 mph and Distance = 80 m, as shown in Figure 15, when rule 1 is executed, $\mu_{Low}(15) = 0.5$ and $\mu_{Large}(80) = 0.8$. As the operator in the IF part is AND, the minimum value between 0.5 and 0.8, i.e., 0.5 is selected. The value of 0.5 is then used to reshape the output fuzzy set for rule 1, i.e., the Minor fuzzy set. This operation is called implication. The reshaped output fuzzy set is shown as a dashed read area in Figure 15.

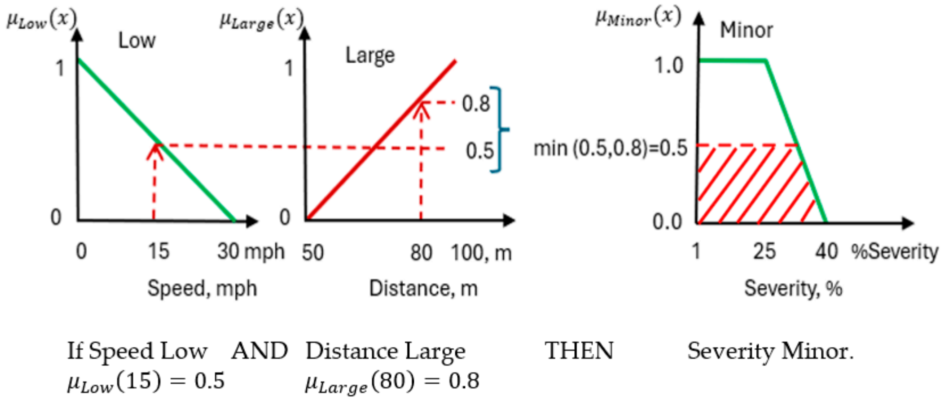

Figure 15. Determining rule 1 firing strength through the implication operation.

To perform implication for rule 2, the operator in its IF part is OR. For speed = 15 mph, $\mu_{High}(15) = 0$ and for distance=80 m, the $\mu_{Small}(80) = 0$. As shown in Figure 16, the implication process results in the fuzzy set Major, associated with Severity of the car accident, to be reshaped to zero along the horizontal axis. Therefore, rule 2 does not affect the output of the FIS as the rule's firing strength is zero.

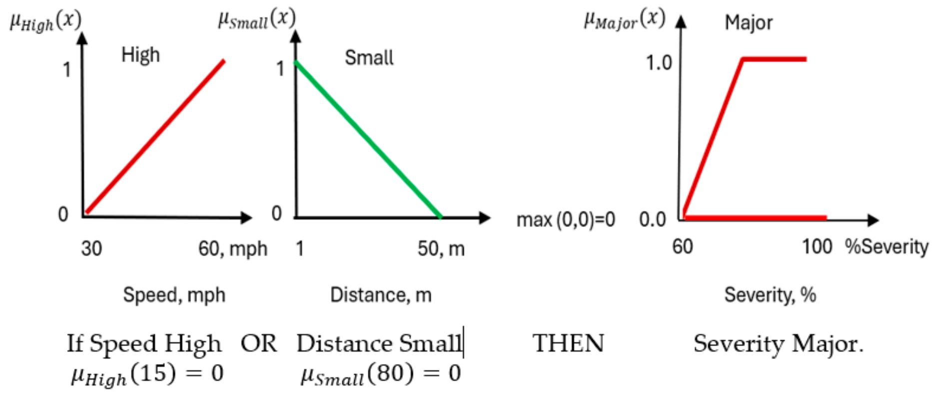

Figure 16. Determining rule 2 firing strength through the implication operation.

The FIS decisions require consideration of all rules. This involves a process called aggregation whereby the output fuzzy sets following their implication process are combined into a single fuzzy set. Thus, the inputs of the aggregation process are the reshaped output membership functions returned after the implication process for each rule. Three possible aggregation methods are [14]:

- Maximum (max)
- Probabilistic OR (probor)
- Summation (sum, the sum of the rules aggregated sets).

As shown in Figure 17, the aggregation of the rules' output sets using the summation method is the shaded area of the Minor fuzzy set associated with the Severity of the accident as rule 2 does not affect the FIS output.

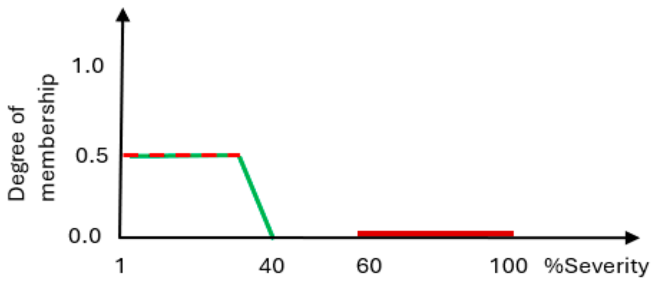

Figure 17. Aggregated output membership functions following the implication operation (the green plot from 1% to 40% is associated with rule 1, the red line from 60% to 100% is associated with rule 2).

Defuzzification element: The output(s) of the FIS is crip because the value(s) may need to be used to control a device or be interpreted numerically. The defuzzification process converts the aggregated fuzzy set associated with each output to a crisp (numerical) value. There are multiple defuzzification methods that include [15]:

- Centroid (centre of gravity)
- Bisector
- Middle of maximum (the average of the maximum value of the output set),
- Largest of maximum
- Smallest of maximum.

The centroid defuzzification method is widely used in numerous applications. It determines the centre of the area under the aggregated fuzzy set. Its output (y, i.e., the FIS output) for a non-smooth aggregated set is

$$y = \frac{\sum_i \mu(x_i) x_i}{\sum_i \mu(x_i)} \tag{13}$$

where $\mu(x_i)$ is the degree of membership value for point x_i in U.

2.4. Adaptive Neuro-Fuzzy Inference System

Although FIS is a valuable tool for performing reasoning in a linguistic form, it does not have the ability to learn from examples and thus automatically adapt its parameters. On the other hand, artificial neural networks (ANNs) have a learning capability through training but have a limitation that can behave like a black box, i.e., they do not provide transparent reasoning for their decisions. By combining the FIS and ANNs, the strengths of the two artificial intelligence techniques complement, resulting in a powerful tool called adaptive neuro-fuzzy inference system (ANFIS) [16]. For completeness, a brief introduction to ANNs is included to help understanding of ANFIS. A basic ANN architecture is the perceptron [17]. Like other ANNs, a perceptron consists of several interconnected nodes also known as processing elements or neurons. A node making up the perceptron is shown in Figure 18.

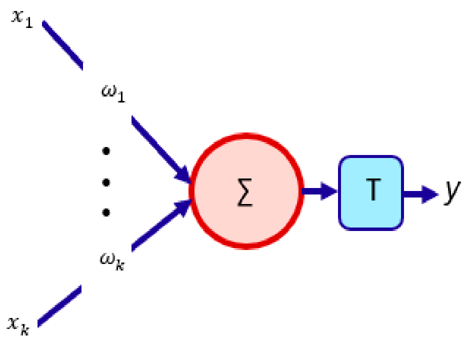

Figure 18. A node forming a part of perceptron artificial neural network, x_i (i = 1, …, k) are the inputs, ω_i (i = 1, …, k) are the connections' weights, $\sum$ is the summing function, and T is the transfer function (also known as the activation function).

Its k inputs are x_1, x_2, …, x_k, and there is a connection weight (ω_i) associated with each input (i = 1, …, k). The output of the node is determined by initially combining its inputs with the associated connection weights (that are initially chosen as random values), providing a sum (s), i.e.,

$$s = \sum_{i=1}^{k} \omega_i x_i \tag{14}$$

The output of the node (y) is determined by mapping s using a transfer function (activation function). For a sigmoid transfer function

$$y = \frac{1}{1+e^{-a(s-b)}} \tag{15}$$

where a determines the sigmoid's slope at the crosspoint $x = b$ and e is a mathematical constant approximately equal to 2.71828. A sigmoid activation function is shown in Figure 8.

The perceptron is a supervised learning ANN, i.e., it requires its inputs to be associated with provided labels (i.e., the identity of the inputs, also known as desired values or targets) during its training. To convert a node to a perceptron, a learning mechanism and an ability to determine how well it is learning during its training (i.e., a performance measurement element, P) are integrated resulting in a structure shown in Figure 19.

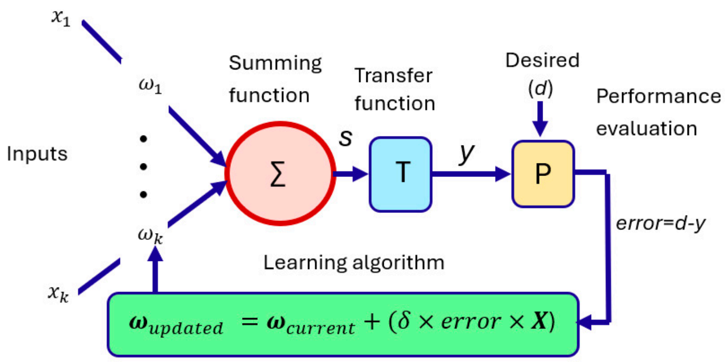

Figure 19. A perceptron artificial perceptron neural network.

As an example, for a problem involving differentiating between objects of two types, the desired values for objects from the categories could be 0 and 1. During training, the performance measure (represented as P in Figure 19) determines the difference between the desired output (d) for the provided input, and the actual output of the perceptron (y) resulting in an error

$$error = d - y \tag{16}$$

The learning algorithm updates the weight by [18]

$$\boldsymbol{\omega}_{updated} = \boldsymbol{\omega}_{current} + (\delta \times error \times \boldsymbol{X}) \tag{17}$$

where $\boldsymbol{\omega}_{updated}$ is the vector of a new set of connection weights, $\boldsymbol{X}$ is the vector of inputs and δ is the learning rate controlling the learning convergence speed. The weights' update is iteratively continued until the error is close to zero.

ANFIS is a hybrid of an ANN structure facilitating learning and the FIS accommodating a means of handing imprecise inputs. ANFIS proved effective in numerous applications [19]. Its structure is shown in Figure 20 [16].

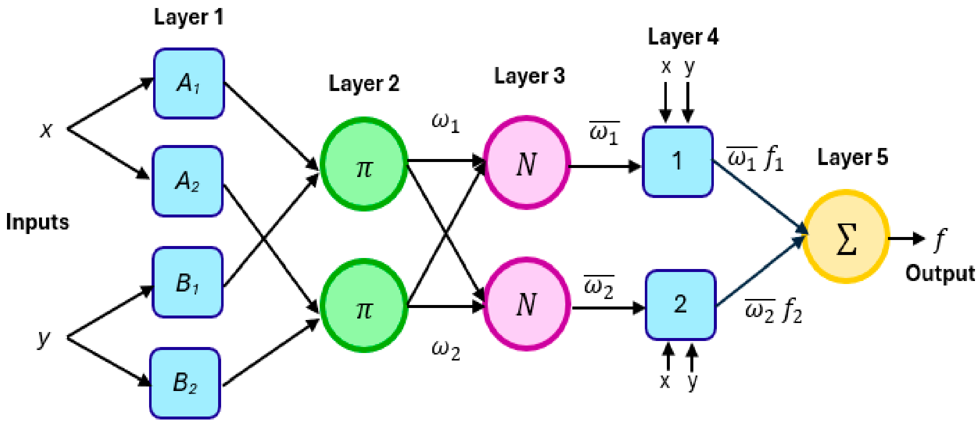

Figure 20. Adaptive neuro-fuzzy inference system (ANFIS) [16].

ANFIS is a feedforward network consisting of five layers [20]. The ANFIS architecture in Figure 20 has two inputs (x and y) and an output (f). Its layers are either adaptive with adjustable parameters (shown as squares in Figure 20) or fixed nodes (represented by circles in Figure 20) with fixed parameters. The ANFIS operation could be illustrated by considering two fuzzy IF-THEN rules based on the first-order Sugeno fuzzy inference [16,20]:

Rule 1: IF x is A_1 AND y is B_1 THEN $f_1 = p_1x + q_1y + r_1$

Rule 2: IF x is A_2 AND y is B_2, THEN $f_2 = p_2x + q_2y + r_2$

where A_i and B_i are fuzzy sets (i = 1, 2 is the fuzzy rule index) and f_i (i = 1,2) are the outputs of the fuzzy rules, p_i, q_i, and r_i are the ANFIS adaptive parameters that are determined during its training. Each layer of the ANFIS has a unique function as described next.

Layer 1: The adaptive nodes in layer 1 perform fuzzification of the ANFIS crip inputs to provide outputs (O_i^1) representing degrees of memberships for the fuzzy sets A_i (i = 1, 2), i.e., $\mu A_i(x)$ and for the fuzzy sets B_{i-2} (i = 3, 4), i.e., $\mu B_{i-2}(y)$, where i is the node's index. For the Gaussian membership function with adjustable parameters a_i and b_i, the outputs are

$$O_i^1 = \mu A_i(x) = e^{-0.5\left(\frac{x-b_i}{a_i}\right)^2} \quad i = 1, 2 \tag{18}$$

$$O_i^1 = \mu B_{i-2}(y) = e^{-0.5\left(\frac{y-b_i}{a_i}\right)^2} \quad i = 3, 4 \tag{19}$$

Layer 2: This layer uses fixed parameters (represented by circles π in Figure 20) to implement AND operation and computes the firing strengths of the rules. The outputs of this layer ($\omega_i, i = 1, 2$) are the products of the layer's inputs, i.e.,

$$O_i^2 = \omega_i = \mu A_i(x) \times \mu B_i(y) \quad i = 1, 2 \tag{20}$$

Layer 3: This layer consists of fixed nodes to normalise the data. The i^{th} node output (O_i^3, i = 1, 2) indicates the ratio of i^{th} rules firing strength (ω_i) to the sum of the firing strengths of all rules.

$$O_i^3 = \overline{\omega_i} = \frac{\omega_i}{\sum_{k=1}^{2} \omega_k} \quad i = 1, 2 \tag{21}$$

Layer 4: The adaptive nodes in layer 4 provide outputs ($O_i^4, i = 1, 2$) that are the product of the normalised firing strength ($\overline{\omega_i}$) and the first order polynomial of its inputs, i.e.,

$$O_i^4 = \overline{\omega_i} f_i = \overline{\omega_i}\,[p_i(x) + q_i(y) + r_i] \quad i = 1, 2 \tag{22}$$

Layer 5: The final layer consists of a fixed node that provides the sum of its incoming data. The output of this node is

$$f = O_i^5 = \sum_{i=1}^{2} \overline{\omega_i} f_i \tag{23}$$

ANFIS uses a hybrid learning algorithm based on the combination of gradient descent and least squares methods [20] and involves multiple iterations consisting of forward passes and backward passes. During the forward pass, the premise parameters ($\text{i.e.}, a_i, b_i$) of the second layer are kept constant while the consequent parameters p_i, q_i and r_j associated with the fourth layer are updated using the least square method. During the backward pass, the consequent parameters determined from the previous iteration are kept constant and the premise parameters are updated using the gradient descent method [20].

2.5. Fuzzy c-Means Clustering

The fuzzy c-means (FCM) clustering algorithm has been applied extensively to analyse and interpret data from various sources. It was proposed by Dunn in 1973 and further improved by Bezdek in 1981 [21–23]. Defining,

- $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$: A matrix of N observations in D-dimensional Euclidean space.
- C: The number of clusters.
- $\boldsymbol{V} = \boldsymbol{v}_j$: The centres of the identified clusters, $1 \leq j \leq C$.
- $J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} \mu_{ij}^m \left\|\boldsymbol{x}_i - \boldsymbol{v}_j\right\|^2, 1 \leq m < \infty$: The objective function is minimised using the generalised form of the least-square errors to partition $\boldsymbol{X}$ into C clusters. The ‖ ‖ symbol represents any norm indicating the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{v}_j$.
- m: The fuzziness weighting, a positive value, typically 2.
- $\mu_{ij} \in [0, 1], 1 \leq i \leq N, 1 \leq j \leq C$: The degree of membership of i^{th} observation to j^{th} cluster. It is in the range 0 (not a member) to 1 (full member). The sum of the degrees membership of the i^{th} observation to all clusters is 1, i.e., $\sum_{j=1}^{C} \mu_{ij} = 1, 1 \leq i \leq N$, and the sum of all degrees of membership in a single cluster is less than N, i.e., $0 < \sum_{i=1}^{N} \mu_{ij} < N, 1 \leq j \leq C$.

The FCM algorithm requires an initialisation of its parameters and then an iterative process to determine the centres of the clusters and the degrees of membership of each observation to each cluster. The parameters requiring initialisation are:

- C (between 2 and $N - 1$)
- m (m larger than 0, typically 2.)
- ε, the algorithm's iteration termination criteria ($\varepsilon > 0$)
- u_{ij}, the degrees of membership initially randomised between 0 and 1.
- t, iteration counter, initially $t = 1$

The iterative stage of the FCM algorithm involves the following steps:

i. Compute the centres of the clusters ($\boldsymbol{v}_j$)

$$\boldsymbol{v}_j = \frac{\sum_{i=1}^{N} \mu_{ij}^{m} \times x_i}{\sum_{i=1}^{N} \mu_{ij}^{m}} \quad j = 1, 2, \ldots, C \tag{24}$$

ii. Update μ_{ij} with $\boldsymbol{v}_j$

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\| x_i - v_j \|}{\| x_i - v_k \|} \right)^{\frac{2}{m-1}}} \quad i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, C \tag{25}$$

iii. Determine $\| \mu^t - \mu^{t-1} \|$, i.e., the magnitude of change in the degrees of membership between the current and previous iteration.

iv. If $\| \mu^t - \mu^{t-1} \| < \varepsilon$, the algorithm iteration is terminated (i.e., training is completed) otherwise t is incremented by 1 and the iteration is continuous from step (i).

3. Results

Fuzzy logic has been applied to a broad range of problems associated with process control, object recognition, image and signal processing, prediction, classification, decision-making, optimisation and time series analysis [13]. In this section, an overview of some fuzzy logic developments is provided. The section also includes the implementation aspects of fuzzy logic models.

3.1. Fuzzy Logic Developments in Decision Support

An article has provided a review of the developments in fuzzy logic applications for decision support [24]. The topics included in the article covered the roles of fuzzy logic in decision-making, evaluation methods, prediction methods and decision-support algorithms. Fuzzy logic has been applied to assess ships for search and rescue operations [25]. This proved helpful for the application as several parameters had to be compared to rank the ships according to the coordinator's preferences and as some related data could be missing, or incomplete. A framework that utilised FIS was devised to assist decision-makers in partitioning an urban area based on the presence and characteristics of the greenery, related to the built and social fabric [26].

A fuzzy logic decision support system that integrated bridge information modelling and cost estimation was devised [27]. The model could assist the relevant stakeholders to conceptually plan for concrete box–bridge construction projects. A study considered the ergonomic risks that modular construction workers are exposed to and devised a decision support system that could help practitioners automatically assess the risks [28]. A decision support system that adapted four fuzzy inference systems was implemented to monitor the renal function by the level of proteinuria and the glomerular filtration rate [29]. Fuzzy logic was utilised to deal with the risk that could arise from a lack of information leading to uncertainty in auditing [30]. A fuzzy logic approach assisted in optimising display windows in a clothes shop [31]. An intelligent strategy with Q-learning (i.e., a model-free reinforcement learning algorithm) and a fuzzy control algorithm was proposed to improve decision-making in enterprise innovation [32]. A hybrid fuzzy multi-criteria decision-making model to open innovation partner evaluation was devised [33]. The fuzzy set theory proved effective in measuring the commercial potential of new product ideas [34].

A multi-criteria fuzzy logic decision support approach to assess possible alternatives for power generation in non-interconnected areas of Colombia was reported [35]. The effectiveness of this approach was demonstrated using a real case study on the San Andrés energy-planning problem. A fuzzy inference model to facilitate decision support for sustainable production planning has been developed and its effectiveness demonstrated in a case study [36]. A fuzzy logic-based tool to support decision-making in planning transport development was devised [37]. The tool allowed assessment of the infrastructure development projects in road and rail transportation. An ANFIS was applied in a decision support system to determine the dew point temperature [38]. Customer services were optimised using fuzzy logic models [39].

An ANFIS model was developed to predict the free fatty acid (FFA) content in bleached and deodorized palm oil [40]. The ANFIS was effective in estimating the FFA quality for the palm oil refining process. Adaptivity in learning and facilitating mobility in the manner of learning was achieved by incorporating an ANFIS [41].

3.2. Fuzzy Logic Developments in Industrial Processes and Control

Fuzzy logic applications in oil and gas industries included the development of high-resolution geological models for flow simulations, pressure control systems, ranking hydrocarbon reservoirs, estimating the rate of drill bit penetration, estimating rock strength, and chock size control [13]. A fuzzy logic model was integrated with a wireless sensor network to monitor essential soil parameters, including PH, temperature, humidity, electrical conductivity, nitrogen, phosphorus, and potassium [42]. The model proved effective in helping farmers to gather information about the soil quality of stevia crops. A review of applications of fuzzy logic control for refrigeration and air conditioning systems showed that fuzzy controllers can provide an improved thermal efficiency as compared to classic controllers such as the proportional–integral–derivative (PID) and they reduce energy consumption [43]. A fuzzy logic controller was devised for a small-scale solar plant. The controller recognised the type of users according to their energy consumption and optimised the thermal load by switching among different operational modes [44]. The controller increased the overall electrical and thermal production.

A fuzzy control was used as part of the scheduling of feed rate for computer numerical control machine tools [45]. The controller shortened the processing time, improved the cutting precision and provided a more stable machining. A methodology to adjust the parameters of a fuzzy logic controller of an energy management system was devised [46]. The approach aimed to minimise the power peaks and fluctuations of a residential microgrid. A risk assessment and management model to improve understanding of safety in railway stations was devised using an ANFIS [47]. A hybrid ANFIS and genetic algorithm system managed to optimise the operation of a piezoelectric cantilever–oscillator–spring energy harvester [48]. Adsorption of reactive orange 16 by hydrochar was assisted by an ANFIS [49]. There has been a recent detailed survey of the applications of ANFIS to photovoltaic applications [50]. ANFIS has proved effective in multiple applications related to the construction industry [51,52].

3.3. Fuzzy Logic Developments in Data Communication, Telecommunication and Internet of Things

Wireless mobile ad-hoc networks (MANETs) communicate without reliance on a fixed infrastructure as the computer data can move through the network by hopping intermediate nodes till it reaches the destination. A fuzzy logic system was devised as part of a MANET to determine the transmission parameters resulting in an improved operational performance [53]. The manner the operational load is shared across the elements of a computer network is a factor in the overall performance of the network. A sender-initiated fuzzy-logic-based protocol was devised to improve the performance of computer networks by appropriately balancing the operational load across the nodes [54]. A fuzzy neural network-based model was designed to deal with some of the problems associated with the Internet industry, e.g., runtime ambiguity, instability and large data volume [55]. The performance of computer networks can be examined by considering their quality of service (QoS). However, to determine QoS, computer traffic information based on the communicated data packets needs to be gathered. As examining every transmitted packet is not practical due to their quantity, sampling of the data packets is required. A study utilised an adaptive fuzzy logic data packet sampling approach that proved more effective than the conventional non-adaptive approaches in representing the communicated traffic [56].

Handover (handoff) in cellular networks allows a call or data session to move from one cell to another as a user moves between the network's coverage areas. The process is critical in wireless communication to ensure a continuous connection for the users. A fuzzy logic-based adaptive handover decision for 5G ultra-dense networks was devised that could dynamically adjust its handover parameters [57]. A fuzzy control system was used to improve operations in edge computing [58]. In video transmission, QoS is reflective of the communication traffic (e.g., the time taken for a data packet to arrive at a destination and the proportion of packet lost during transmission) while quality of experience (QoE) is indicative of the user's perception of the quality of the received images (e.g., the extent of noise and image distortion). A modular fuzzy logic system that determined the QoS and QoE of videos in wireless computer networks was developed and its performance was successfully evaluated [59]. Fuzzy control has proved effective in dealing with non-periodic denial of service attacks [60].

The Internet of Things (IoT) represents a network of devices with technologies that allow communication between themselves as well as the cloud. IoT has benefited from adapting fuzzy logic techniques. These included a fuzzy model for IoT information security evaluation [61], a secure intelligent fuzzy blockchain framework for effective threat detection in IoT networks [62], a fuzzy description logic-based IoT framework that allows users to build their IoT applications according to their needs [63], an activity recognition for IoT devices using fuzzy spatio-temporal features [64] and a fuzzy logic controller for distributed IoT gateway to manage input uncertainties [65]. ANFIS models were also found effective for IoT-related applications. For example, an ANFIS was used to regulate multipath congestion in IoT [66], an intelligent fire warning application was devised using a combination of IoT and ANFIS [67], a lung cancer detection and prediction model was reported that uses an ANFIS and Internet of Medical Things (IoMT) [68], and an ANFIS was devised to minimize higher-order harmonics due to nonlinear load disturbances in smart IoT devices [69].

3.4. Fuzzy Logic Developments in Image and Signal Processing

Fuzzy logic has been applied to image processing tasks such as image segmentation, image filtering, image classification and edge detection [70,71]. A combination of fuzzy logic and convolutional neural networks was devised to recognise ambient sounds of daily events with the fuzzy logic part performing filtering of the raw audio events [72]. Fuzzy approaches proved valuable for defining spatial relationships (e.g., adjacency) and metrical relations (distances, directional relative position) [73]. A fuzzy filter was devised for additive noise reduction, distinguishing between local variations due to the noise and image features [74]. Lung radiography images were enhanced by developing fuzzy logic techniques with the aim of assisting physicians in interpreting the images [75]. A fuzzy logic-based method that matched characters using their edge corners as part of Captcha recognition was reported [76]. An FIS model was devised to process electroencephalogram signals as part of improving understanding of saccade [77]. Fuzzy logic has been utilised to automatically analyse x-ray images of industrial products for defect detection [78].

ANFIS has been applied in numerous medical image processing-related work. A review of some of these applications was conducted [79]. These included its applications for diagnosing prostate cancer, eye diseases, brain tumours and breast cancers [79].

FCM proved effective for segmenting breast tumours in mammograms [80]. FCM was applied to classify oxidation products into different groups, based on their properties [81] and was integrated with Markov random field as part of remote sensing classification tasks [82]. Metabolomics data were interpreted and clustered with the aid of FCM [83].

3.5. Fuzzy Logic Implementation Methods

Fuzzy logic systems and models were implemented in hardware [84,85], in VHDL (Very High-Speed Integrated Circuit Hardware Description Language) [86] and in software, e.g., in Python [87]. They were also implemented on integrated circuit chips [88–90]. There were numerous implementations of fuzzy logic systems and models based on Matlab [91]. Matlab has a Fuzzy Logic Toolbox™ with a comprehensive user's guide document [92] providing detailed instructions on developing fuzzy logic systems and applications. In this section, Matlab is used to illustrate implementations of a basic Mamdani FIS-based model to determine a car accident's severity (i.e., Figure 10). The example is kept deliberately simple to be general and focus on the implementation aspects rather than on a specific application. The rules are:

Rule 1: If Speed High AND Distance Small THEN Severity Major.

Rule 2: If Speed Low AND Distance Large THEN Severity Minor.

Rule 3: If Speed Medium AND Distance Medium THEN Severity Moderate.

For this purpose, the Matlab Fuzzy Logic Designer [92] was used as it allowed a graphic user interface (GUI) method of FIS implementation; however, it is also possible to implement fuzzy logic models using Matlab's scripting and Simulink [92]. The Matlab Fuzzy Logic Designer has five GUI widows [92]:

- *Fuzzy Logic Designer's Main Window*: This window allows the overall FIS model to be viewed, its input(s) and output(s) to be named, their ranges specified, and the FIS operational parameters such as implication, aggregation, and defuzzification methods to be selected from a range of possible options.
- *Membership Function Editor*: This window allows the types and parameters of the input(s) and output(s) fuzzy sets to be defined by selecting amongst 13 different membership functions.
- *Rule Editor*: This window provides an easy approach to defining the rules associated with the FIS model. The operators 'AND' and 'OR' are available to construct complex rules. It also allows the rules to have a weighting to control the level of their significance to the FIS output in relation to other rules.
- *Rule Viewer*: This window allows the user to select the values of the inputs to the FIS and determine and observe the FIS output(s).
- *Surface View*: This window provides a 3-dimensional surface view relating any two selected inputs to the FIS and one of its outputs.

The FIS can be implemented by typing '*fuzzy*' from the Matlab [6] command line. This results in the display of the *Fuzzy Logic Designer* GUI window showing a FIS with a single input (called input1) and single output (called output1). From this window

- A single click on input1 allows its name to be changed to Speed.
- From the window's *Edit* menu, followed by *Add variables*, then *Input*, a second input can be added to the FIS model, and its name can be changed to Distance.
- By clicking on output1, its name can be changed to Severity.
- The FIS parameters can be set as:
- Operators for the rules: AND (Minimum, min), OR (Maximum, max)
- Implication method: (Minimum, min)
- Aggregation method: Sum
- Defuzzification method: Centroid
- FIS type: Mamdani

From the *Fuzzy Logic Designer* window, a double click on the input Speed opens the related *Membership Function Editor* window. The default membership functions are triangular. The range can be set as 1 to 60 (mph). By clicking the individual membership functions, their names and parameters can be set to those shown in Figure 2. It is also possible to move the membership functions to the desired locations by pointing a mouse at them. The second input (Distance) and output (Severity) parameters can similarly be defined (following Figures 13 and 14, respectively) except for the output trapezoidal membership functions should be selected.

From the *Fuzzy Logic Designer* window, by clicking on the *Edit* icon followed by *Rules*, the *Rule Editor* window is displayed, and the associated rules can be entered. From the

Fuzzy Logic Designer window, selecting the *View* icon allows the *Rule Viewer* window to be displayed as shown in Figure 21. The two columns on the left of the figure are the inputs to the FIS, i.e., Speed and Distance, and the column to the right (shown in blue) of the figure is the output of the FIS, i.e., the car accident severity (Severity). The red line on the two inputs can be moved to the user's desired values. For the desired inputs, the car accident severity in percentage is determined and displayed. For example, when the Speed = 40 mph and the Distance = 40 m, the severity of the car accident is indicated as 60.2%

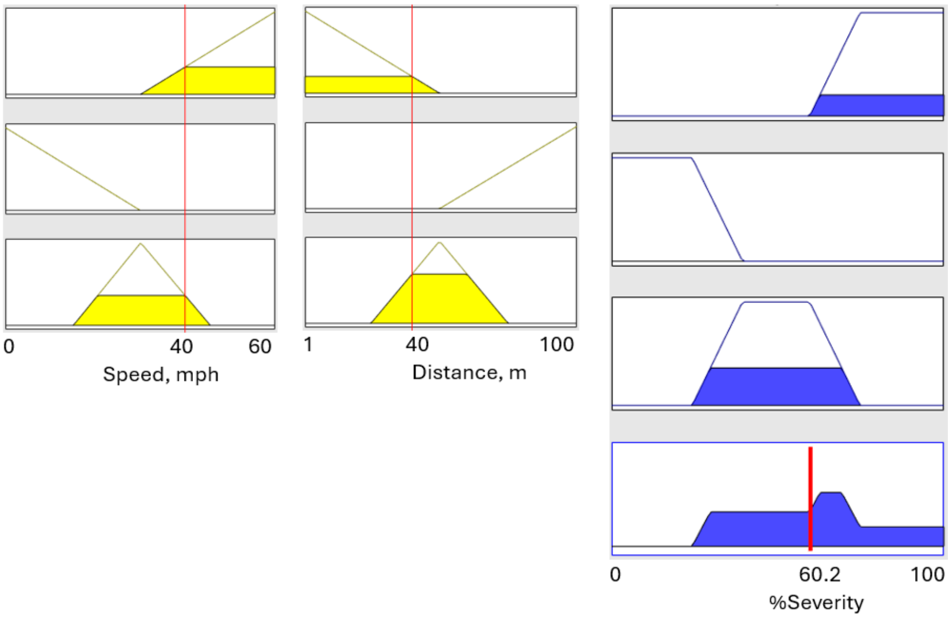

Figure 21. Matlab rule viewer window for the 3 rules. The two columns on the left of the figure (shown in yellow) are associated with the inputs (i.e. Speed and Distance) to the FIS and the column on the right (shown in blue) is associated with the output (Severity) of the FIS model.

From the *Fuzzy Logic Designer* window, by clicking on the *View* icon followed by *Surface*, the 3-dimensional plot shown in Figure 22, relating the two inputs and output, is displayed. The figure indicates that as speed increases and distance decreases, the severity of car accidents increases.

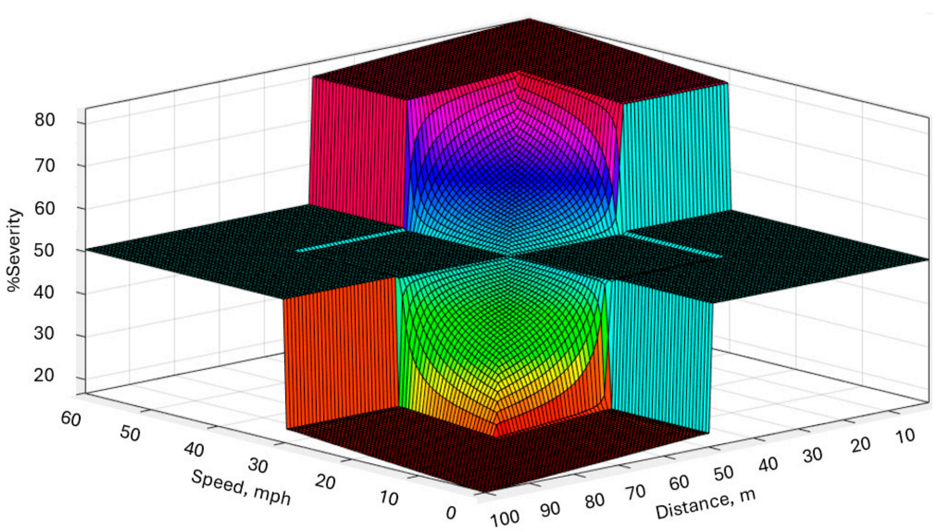

Figure 22. Matlab surface viewer window for the three rules.

4. Discussion

The field of fuzzy logic has greatly progressed in theories, concepts and industrial applicability. Its evolution has been further accelerated by the realization of capabilities and innovations in hybrid techniques. Integration of deep learning and fuzzy logic can facilitate imprecision handling while allowing complex learning capabilities [93]. For example, hybrid fuzzy logic and deep learning allowed remote English translation [94], diagnosis of skin cancer [95] and image classifications [96]. Fuzzy logic has been integrated with genetic algorithms (GAs) providing an improved optimization capability. For example, the integration of fuzzy logic and GAs proved effective in engineering applications [97]. Hybrid metaheuristic optimisation methods were explored to forecast energy carbon dioxide emissions [98]. A self-organising fuzzy classifier proved effective in diagnosing Alzheimer's disease [99]. Fuzzy logic controllers' stability analysis and design can be improved by the introduction of model-based control. These proved valuable in the field of networked control systems [100]. A fuzzy adaptive fixed-time sliding mode control technique was reported for trajectory tracking of a class of high-order non-linear systems [101].

A convenient manner to implement fuzzy logic models is by using the Matlab Fuzzy Logic Toolbox [92]. Its user guide [92] provides descriptions of various means of implementing fuzzy logic models, including models based on its Simulink. An FIS-based water level control is implemented by the toolbox allowing the user to indicate a desired water level. There is also an example of fuzzy logic-based automatic car parking. The toolbox has several applications of ANFIS. A typical application is the modelling of inverse kinematics in a robotic arm (Figure 23). Kinematics is the study of motion of objects without considering the forces that result in their movement while inverse kinematics is the process of determining joint angles from known coordinates of an end effector.

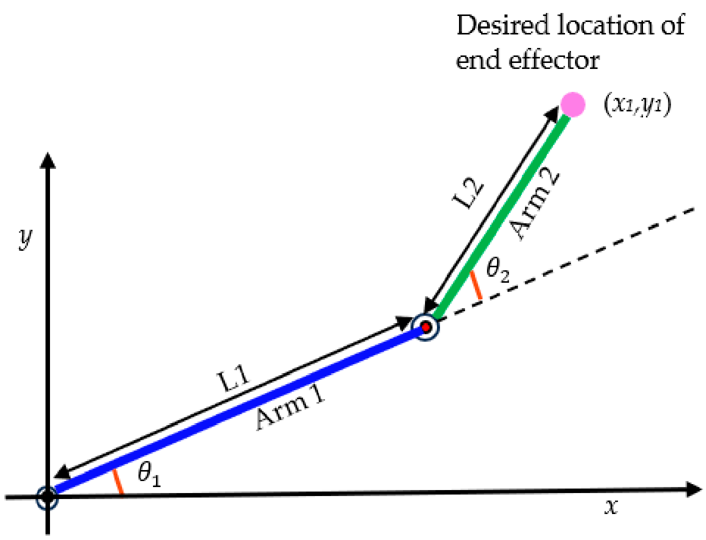

Figure 23. Use of an ANFIS to control a robot arm [92].

An ANFIS was used to determine the joint's angles θ_1 (i.e., the angle between the position of the arm 1 and the ground) and the joint's angle θ_2 (i.e., the angle between arm 2 and arm 1) for the end effector in the two-joint robotic arm, shown in Figure 23, to move to a pre-defined x-y coordinate [92]. To achieve this, initially, forward kinematics formulae were used to determine the end effector coordinate (x_1, y_1) for different values of θ_1 and θ_2. The resulting dataset was then used as part of reverse kinematics to train an ANFIS (using the Matlab function *anfis* [92]) to learn to determine θ_1 and θ_2 for a desired end effector x-y coordinate. The associated Matlab code and further explanations can be found in [92]. Other ANFIS applications implemented in [92] include a chaotic time-series prediction and an adaptive noise cancellation method. There are also several implemented applications of fuzzy c-means such as clustering quasi-random data.

Developments in deep learning techniques [102–104] have created opportunities for their integration with fuzzy logic [96], facilitating complex problems to be tackled effectively and opening opportunities for further research and development.

5. Conclusions

The operations of main fuzzy logic techniques, i.e., fuzzy inference system, adaptive neuro-fuzzy inference system and fuzzy c-means, were explained and an exploration of developments in the field was carried out. Approaches to implement fuzzy logic models were outlined. The article was prepared in such a way as to allow a broad readership.

Although fuzzy logic is a well-established field, there remain numerous development and application opportunities. An exciting area is hybrid systems, where fuzzy logic techniques are combined with one or more other artificial intelligence techniques such as deep learning, robotics, and genetic algorithms, thus providing a greater problem-solving capability. There are also numerous opportunities to use fuzzy logic as part of embedded systems and devise solutions for their optimum adaptation to integrated circuits.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A. Set Properties

Given the definitions:

- A, B and C: sets
- $\emptyset$: Null set, i.e., a set without any member
- U: Universal set (a set that has all elements of other sets including its own elements)
- c: Complement
- $\cup$: Union
- $\cap$: Intersection
- $\subseteq$: subset or equal ($\subset$: subset)
- $\backslash$: Complement
- Min: minimum
- Max: maximum

The main operations and properties are:

Union: $A \cup B = \mu_A(x) \cup \mu_B(x) = max\big(\mu_A(x), \mu_B(x)\big)$

Intersection: $A \cap B = \mu_A(x) \cap \mu_B(x) = min\big(\mu_A(x), \mu_B(x)\big)$

Complement: $A^C = \backslash A$

Commutativity $A \cup B = B \cup A$

$A \cap B = B \cap A$

Associativity $A \cup (B \cup C) = (A \cup B) \cup C$

$A \cap (B \cap C) = (A \cap B) \cap C$

Distributivity: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Impotency: $A \cup A = A$

$A \cap A = A$

Identity: $A \cup \emptyset = A$

$$A \cap U = A$$
$$A \cap \emptyset = \emptyset$$
$$A \cup U = U$$

Transitivity: $A \subseteq B \subseteq C \;\; then\; A \subseteq C$

Involution: $(A^c)^c = A$

De Morgan laws: $(A \cup B)^c = A^c \cap B^c$

$$(A \cap B)^c = A^c \cup B^c$$

References

1. Mossakowski, T.; Goguen, J.; Diaconescu, R.; Tarlecki, A. *What is Logic?* Beziau, J.-Y., Ed.; Logica Universalis, Birkhäuser Verlag: Basel, Switzerland, 2006; pp. 113–135.
2. Zadah, L.A. Fuzzy sets. *Inf. Control.* **1965**, *8*, 338–353.
3. Zadah, L.A. Is there a need for fuzzy logic? *Inf. Sci.* **2008**, *178*, 2751–2779.
4. Zadeh, L.A. Fuzzy logic, neural networks, and soft computing, *Commun. ACM* **1994**, *37*, 77–84.
5. Chen, T.; Karimov, I.; Chen, J.; Constantinovitc, A. Computer and fuzzy theory application: Review in home appliances. *J. Fuzzy Ext. Appl.* **2020**, *1*, 133–138.
6. Matlab, Mathworks®, Version R2024a. Available online: https://uk.mathworks.com/help/ (accessed on 4 October 2024).
7. Jain, A.; Sharma, A. Membership function formulation methods for fuzzy logic systems: A comprehensive review. *J. Crit. Rev.* **2020**, *7*, 8717–8733.
8. Pancardo, P.; Hernández-Nolasco, J.A.; Wister, M.A.; Garcia-Constantino, M. Dynamic membership functions for context-based fuzzy systems. *IEEE Access* **2021**, *9*, 29665–29676.
9. Medasani, S.; Kim, J.; Krishnapuram, R. An overview of membership function generation techniques for pattern recognition. *Int. J. Approx. Reason.* **1998**, *19*, 391–417.
10. Schwaab, A.A.D.S.; Nassar, S.M.; Filho, P.J.D.F. Automatic methods for generation of type-1 and interval type-2 fuzzy membership functions. *J. Comput. Sci.* **2015**, *11*, 976–987.
11. Chen, M.-S.; Wang, S.-W. Fuzzy clustering analysis for optimizing fuzzy membership functions, *Fuzzy Sets Syst.* **1999**, *103*, 239–254.
12. Cheng, H.D.; Chen, J.-R. Automatically determine the membership function based on the maximum entropy principle. *Inf. Sci.* **1997**, *96*, 163–182.
13. Belyadi, H.; Haghighat, A. *Machine Learning Guide for Oil and Gas Using Python: A Step-By-Step Breakdown with Data, Algorithms, Codes, and Applications*; Elsevier Inc: Amsterdam, The Netherlands, 2021.
14. Pham, D.T.; Castellani, M. Action aggregation and defuzzification in Mamdani-type fuzzy systems. *Proc. Inst. Mech. Eng. Vol. 216 Part C J. Mech. Eng. Sci.* **2002**, *216*, 747–759.
15. Jager, R.; Verbruggen, H.B.; Bruijn, P.M. The role of defuzzification methods in the application of fuzzy control. *IFAC Intell. Compon. Instrum. Control. Appl.* **1992**, *25*, 75–80.
16. Jang, J.-S.R. ANFIS adaptive-network-based fuzzy inference system, *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685.
17. Gallant, S.I. Perceptron-based learning algorithms, *IEEE Trans. Neural Netw.* **1990**, *1*, 179–191.
18. Du, K.-L.; Leung, C.-S.; Mow, W.H.; Swamy, M.N.S. Perceptron: Learning, generalization, model selection, fault tolerance, and role in the deep learning era. *Mathematics* **2022**, *10*, 4730.
19. Kar, S.; Das, S.; Ghosh, P.K. Applications of neuro fuzzy systems: A brief review and future outline. *Appl. Soft Comput.* **2014**, *15*, 243–259.
20. Lingxiao, L.; Pang, S. An implementation of the adaptive neuro-fuzzy inference system (ANFIS) for odor source localization. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020.
21. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Plenum Press: New York, NY, USA, 1981.
22. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203.
23. Kesemen, O.; Tezel, Ö.; Özkul, E. Fuzzy c-means clustering algorithm for directional data (FCM4DD). *Expert Syst. Appl.* **2016**, *58*, 76–82.
24. Wu, H.; Xu, Z.S. Fuzzy logic in decision support: Methods, applications and future trends. *Int. J. Comput. Commun. Control.* **2021**, *16*, 4044. https://doi.org/10.15837/ijccc.2021.1.4044.
25. Malyszko, M. Fuzzy logic in selection of maritime search and rescue units. *Appl. Sci.* **2022**, *12*, 21.
26. Cardone, B.; Di Martino, F. A fuzzy rule-based GIS framework to partition an urban system based on characteristics of urban greenery in relation to the urban context. *Appl. Sci.* **2020**, *10*, 8781.
27. Markiz, N.; Jrade, A. Integrating a fuzzy-logic decision support system with bridge information modelling and cost estimation at conceptual design stage of concrete box-girder bridges. *Int. J. Sustain. Built Environ.* **2014**, *3*, 135–152.
28. Govindan, A.R.; Li, X. Fuzzy logic-based decision support system for automating ergonomics risk assessments. *Int. J. Ind. Ergon.* **2023**, *96*, 103459.

29. Improta, G.; Mazzella, V.; Vecchione, D.; Santini, S. Fuzzy logic–based clinical decision support system for the evaluation of renal function in post-transplant patients. *J. Eval. Clin. Pract.* **2020**, *26*, 1224–1234.
30. Friedlo, G.T.; Schleifer, L.L.F. Fuzzy logic: Application for audit risk and uncertainty. *Manag. Audit. J.* **1999**, *14*, 127–135.
31. Lashin, M.M.A.; Khan, M.I.; Khedher, N.B.; Eldin, S.M. Optimization of display window design for females' clothes for fashion stores through artificial intelligence and fuzzy System. *Appl. Sci.* **2022**, *12*, 11594.
32. Jia, Y.; Wang, Z. Application of artificial intelligence based on the fuzzy control algorithm in enterprise innovation. *Heliyon* **2024**, *10*, e28116.
33. Puzović, S.; Vasović, V.J.; Milanović, D.D.; Paunović, V. A hybrid fuzzy MCDM approach to open innovation partner evaluation. *Mathematics* **2023**, *11*, 3168.
34. Sitnicki, M.W.; Balan, V.; Tymchenko, I.; Sviatnenko, V.; Sychova, A. Measuring the commercial potential of new product ideas using fuzzy set theory. *Innov. Mark.* **2021**, *17*, 149–163.
35. Cerón, A.M.R.; Kafarova, V.; Latorre-Bayona, G. A fuzzy logic decision support system for assessing sustainable alternative for power generation in non-Interconnected areas of Colombia- case of study. *Chem. Eng. Trans.* **2017**, *57*, 421–426.
36. Zarte, M.; Pechmann, A.; Nunes, I.L. Fuzzy inference model for decision Support in sustainable production planning processes—A case study. *Sustainability* **2021**, *13*, 1355.
37. Kaczorek, M.; Jacyna, M. Fuzzy logic as a decision-making support tool in planning transport development. *Arch. Transp.* **2022**, *61*, 51–70.
38. Zhang, G.; Band, S.S.; Ardabili, S.; Chau, K.-W.; Mosavi, A. Integration of neural network and fuzzy logic decision making compared with bilayered neural network in the simulation of daily dew point temperature. *Eng. Appl. Comput. Fluid Mech.* **2022**, *16*, 713–723.
39. Díaz, G.M.; González, R.A.C. Fuzzy logic and decision making applied to customer service optimization. *Axioms* **2023**, *12*, 448.
40. Ali, N.S.; Mohd-Yusof, K.; Othman, M.F.; Latip, R.A.; Ismail, M.S.N. Adaptive Neuro Fuzzy Inference System (ANFIS) modelling for quality estimation in palm oil refining process. *J. Mech. Eng.* **2019**, *8*, 36–47.
41. Al-Hmouz, A.; Shen, J.; Al-Hmouz, R.; Yan, J. Modeling and simulation of an adaptive neuro-fuzzy inference system (ANFIS) for mobile learning. *IEEE Trans. Learn. Technol.* **2012**, *5*, 226–237.
42. Vejar-Cortés, A.-P.; García-Díaz, N.; Soriano-Equigua, L.; Ruiz-Tadeo, A.-C.; Álvarez-Flores, J.-L. Determination of crop soil quality for stevia rebaudiana bertoni morita II using a fuzzy logic model and a wireless sensor network. *Appl. Sci.* **2023**, *13*, 9507.
43. Belman-Flores, J.M.; Rodríguez-Valderrama, D.A.; Ledesma, S.; García-Pabón, J.J.; Hernández, D.; Pardo-Cely, D.M. A review on applications of fuzzy logic control for refrigeration systems. *Appl. Sci.* **2022**, *12*, 1302.
44. Cioccolanti, L.; De Grandis, S.; Tascioni, R.; Pirro, M.; Freddi, A. Development of a fuzzy logic controller for small-scale solar organic Rankine cycle cogeneration plants. *Appl. Sci.* **2021**, *11*, 5491.
45. Lin, C.-J.; Lin, C.-H.; Wang, S.-H. Using fuzzy control for feed rate scheduling of computer numerical control machine tools. *Appl. Sci.* **2021**, *11*, 4701.
46. Arcos-Aviles, D.; Pacheco, D.; Pereira, D.; Garcia-Gutierrez, G.; Carrera, E.V.; Ibarra, A.; Ayala, P.; Martínez, W.; Guinjoan, F. A comparison of fuzzy-based energy management systems adjusted by nature-inspired algorithms. *Appl. Sci.* **2021**, *11*, 1663.
47. Alawad, H.; An, M.; Kaewunruen, S. Utilizing an adaptive neuro-fuzzy inference system (ANFIS) for overcrowding level risk assessment in railway stations. *Appl. Sci.* **2020**, *10*, 5156.
48. Babaei, A.; Parker, J.; Moshave, P. Adaptive neuro-fuzzy inference system (ANFIS) integrated with genetic algorithm to optimize piezoelectric cantilever-oscillator-spring energy Harvester: Verification with Closed-Form solution. *Comput. Eng. Phys. Model.* **2022**, *5*, 1–22.
49. Nayagam, O.J.P.; Prasanna, K. Response surface methodology and adaptive neuro-fuzzy inference system for adsorption of reactive orange 16 by hydrochar. *Glob. J. Environ. Sci. Manag.* **2023**, *9*, 373–388.
50. Guerra, M.I.S.; de Araújo, M.F.U.; de Carvalho Neto, J.T.; Vieira, R.G. Survey on adaptative neural fuzzy inference system (ANFIS) architecture applied to photovoltaic systems. *Energy Syst.* **2024**, *15*, 505–541.
51. Obianyo, J.I.; Udeala, R.C.; Alaneme, G.U. Application of neural networks and neuro-fuzzy models in construction scheduling. *Sci. Rep.* **2023**, *13*, 8199.
52. Nguyen, P.H.D.; Fayek, A.R.F. Applications of fuzzy hybrid techniques in construction engineering and management research. *Autom. Constr.* **2022**, *134*, 104064.
53. Yuste, A.J.; Triviño, A.; Trujillo, F.D.; Casilari, E. Using fuzzy logic in hybrid multihop wireless networks. *Int. J. Wirel. Mob. Netw.* **2010**, *2*, 96–108.
54. Huang, M.-C. A sender-initiated fuzzy logic control method form network load balancing. *J. Comput. Commun.* **2024**, *12*, 110–122.
55. Yu, J. Application of improved CSA algorithm-based fuzzy logic in computer network control systems. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *15*, 1084–1094.
56. Salama, A.; Saatchi, R.; Burke, D. Fuzzy logic and regression approaches for adaptive sampling of multimedia traffic in wireless computer networks. *Technologies* **2018**, *6*, 24.
57. Hwang, W.-S.; Cheng, T.-Y.; Wu, Y.-J.; Cheng, M.-H. Adaptive handover decision using fuzzy logic for 5G ultra-dense networks. *Electronics* **2022**, *11*, 3278.
58. Silva, S.N.; Goldbarg, M.A.S.d.S.; Silva, L.M.D.d.; Fernandes, M.A.C. Application of fuzzy logic for horizontal scaling in Kubernetes environments within the context of edge computing. *Future Internet* **2024**, *16*, 316.

59. Salama, A.; Saatchi, R. Evaluation of wirelessly transmitted video quality using a modular fuzzy logic system. *Technologies* **2019**, *7*, 67.
60. Pan, Y.; Wu, Y.; Lam, H.-K.; Security-based fuzzy control for nonlinear networked control systems with DoS attacks via a resilient event-triggered scheme. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 4359–4368.
61. de Mello, F.L. A fuzzy model for knowledge base IoT information security evaluation. *J. Inf. Secur. Cryptogr.* **2018**, *5*, 20–26.
62. Yazdinejad, A.; Dehghantanha, A.; Parizi, R.M.; Srivastava, G.; Karimipour, H. Secure intelligent fuzzy blockchain framework: Effective threat detection in IoT networks. *Comput. Ind.* **2023**, *144*, 103801.
63. Pérez-Gaspar, M.; Gomez, J.; Bárcenas, E.; Garcia, F. A fuzzy description logic based IoT framework: Formal verification and end user programming. *PLoS ONE* **2024**, *19*, e0296655.
64. Medina, M.Á.L.; Espinilla, M.; Paggeti, C.; Quero, J.M. Activity recognition for IoT devices using fuzzy spatio-temporal features as environmental sensor fusion. *Sensors* **2019**, *19*, 3512. https://doi.org/10.3390/s19163512.
65. Firouzia, R.; Rahmania, R.; Kanter, T. An autonomic IoT gateway for smart home using fuzzy logic reasoner. *Procedia Comput. Sci.* **2020**, *177*, 102–111.
66. Aalsalem, M.Y. An intelligent adaptive neuro-fuzzy for solving the multipath congestion in Internet of Things. *J. Inf. Syst. Eng. Manag.* **2023**, *8*, 23845.
67. Sarwar, B.; Bajwa, I.S.; Jamil, N.; Ramzan, S.; Sarwar, N. An intelligent fire warning application using IoT and an adaptive neuro-fuzzy inference system. *Sensors* **2019**, *19*, 3150.
68. Shabu, S.L.J.; Refonaa, J.; Mallik, S.; Dhamodaran, D.; Grace, L.K.J.; Ksibi, A.; Ayadi, M.; Alshalali, T.A.N. An Improved Adaptive neuro-fuzzy inference framework for lung cancer detection and prediction on Internet of Medical Things platform. *Int. J. Comput. Intell. Syst.* **2024**, *17*, 228.
69. Gupta, U.K.; Sethi, D.; Goswami, P.K. Adaptive TS-ANFIS neuro-fuzzy controller based single phase shunt active power filter to mitigate sensitive power quality issues in IoT devices. *Adv. Electr. Eng. Electron. Energy* **2024**, *8*, 100542.
70. Castillo, O.; Sanchez, M.A.; Gonzalez, C.I.; Martinez, G.E. Review of recent type-2 fuzzy image processing applications. *Information* **2017**, *8*, 97. https://doi.org/10.3390/info8030097.
71. Bloch, I. Fuzzy sets for image processing and understanding. *Elsevier Fuzzy Sets Syst.* **2015**, *281*, 280–291.
72. Polo-Rodriguez, A.; Vilchez Chiachio, J.M.; Paggetti, C.; Medina-Quero, J. Ambient sound recognition of daily events by means of convolutional neural networks and fuzzy temporal restrictions. *Appl. Sci.* **2021**, *11*, 6978. http://doi.org/10.3390/app11156978.
73. Bloch, I. Fuzzy spatial relationships for image processing and interpretation: A review. *Elsevier Image Vis. Comput.* **2005**, *23*, 89–110.
74. Van De Ville, D.; Nachtegael, M.; Van der Weken, D.; Kerre, E.E.; Philips, W.; Lemahieu, I. Noise reduction by fuzzy image filtering. *IEEE Trans. Fuzzy Syst.* **2003**, *11*, 429–436.
75. Sousa, W.P.; Cruz, C.C.P.; LanzillottI, R.S. Fuzzy divergence for lung radiography image enhancement. *Trends Comput. Appl. Math.* **2023**, *24*, 699–716.
76. Nachar, R.A.; Inaty, E.; Bonnin, P.J.; Alayli, Y. Breaking down Captcha using edge corners and fuzzy logic segmentation/recognition technique. *Security Commun. Netw.* **2015**, *8*, 3995–4012.
77. Saatchi, R. Single-trial lambda wave identification using a fuzzy inference system and predictive statistical diagnosis. *J. Neural Eng.* **2004**, *1*, 21–31.
78. Amza, C.G.; Cicic, D.T. Industrial image processing using fuzzy-logic. *Procedia Eng.* **2015**, *100*, 492–498.
79. Sheikh Hosseini, M.; Zekri, M. Review of medical image classification using the adaptive neuro-fuzzy inference system. *J. Med. Signals Sens.* **2012**, *2*, 49–60.
80. Krasnov, D.; Davis, D.; Malott, K.; Chen, Y.; Shi, X.; Wong, A. Fuzzy c-means clustering: A review of applications in breast cancer detection. *Entropy* **2023**, *25*, 1021.
81. Wu, R.; Zorn, S.R.; Kang, S.; Kiendler-Scharr, A.; Wahner, A.; Mentel, T.F. Application of fuzzy c-means clustering for analysis of chemical ionization mass spectra: Insights into the gas phase chemistry of NO3-initiated oxidation of isoprene. *Atmos. Meas. Tech.* **2024**, *17*, 1811–1835.
82. HongLei, Y.; JunHuan, P.; BaiRu, X.; DingXuan, Z. Remote sensing classification using fuzzy c-means clustering with spatial constraints based on Markov random field. *Eur. J. Remote Sens.* **2013**, *46*, 305–316.
83. Li, X.; Lu, X.; Tian, J.; Gao, P.; Kong, H.; Xu, G. Application of fuzzy c-means clustering in data analysis of metabolomics. *Anal. Chem.* **2009**, *81*, 4468–4475.
84. Ibrahim, A.M. Hardware implementation. In *Fuzzy Logic for Embedded Systems Applications*; Elsevier (Newnes): Amsterdam, The Netherlands, 2004; Chapter 8.
85. Yamakawa, T. Electronic circuits dedicated to fuzzy logic controller. *Sci. Iran. D* **2011**, *18*, 528–538.
86. Barriga, A.; Sánchez-Solanoa, S.; Brox, P.; Cabrera, A.; Baturone, I. Modelling and implementation of fuzzy systems based on VHDL. *Int. J. Approx. Reason.* **2006**, *41*, 164–178.
87. Spolaor, S.; Fuchs, C.; Cazzaniga, P.; Kaymak, U.; Besozzi, D.; Nobile, M.S. Simpful: A user-friendly Python library for fuzzy logic. *Int. J. Comput. Intell. Syst.* **2020**, *13*, 1687–1698.
88. Peyravi, H.; Khoei, A.; Hadidi, K. Design of an analog CMOS fuzzy logic controller chip. *Fuzzy Sets Syst.* **2002**, *132*, 245–260.
89. Azimi, S.M.; Miar-Naimi, H. Designing an analog CMOS fuzzy logic controller for the inverted pendulum with a novel triangular membership function. *Sci. Iran. D* **2019**, *26*, 1736–1748.

90. Gheysari, K.; Mashouf, B. Implementation of CMOS flexible fuzzy logic controller chip in current mode. *Fuzzy Sets Syst.* **2011**, *185*, 125–137.
91. Sivanandam, S.N.; Sumathi, S.; Deepa, S.N. *Introduction to Fuzzy Logic Using Matlab*; Springer: Berlin/Heidelberg, Germany, 2007.
92. Matlab Fuzzy Logic Toolbox User Guide. 2024. Available online: https://uk.mathworks.com/help/fuzzy (accessed on 1 October 2024).
93. Das, R.; Sen, S.; Maulik, U. A Survey on fuzzy deep neural networks. *ACM Comput. Surv.* **2020**, *53*, 1–25.
94. Han, X. Analyzing the impact of deep learning algorithms and fuzzy logic approach for remote English translation. *Sci. Rep.* **2024**, *14*, 14556.
95. Singh, S.K.; Abolghasemi, V.; Anisi, M.H. Fuzzy logic with deep learning for detection of skin cancer. *Appl. Sci.* **2023**, *13*, 8927.
96. Kamthan, S.; Singh, H.; Meitzler, T. Hierarchical fuzzy deep learning for image classification. *Mem.-Mater. Devices Circuits Syst.* **2022**, *2*, 100016.
97. Plerou, A.P.; Vlamou, E.; Papadopoulos, V. Fuzzy Genetic Algorithms: Fuzzy Logic Controllers and Genetics Algorithms. *Glob. J. Res. Anal.* **2016**, *5*, 497–500.
98. Moayedi, H.; Mukhtar, A.; Khedherd, N.B.; Elbadawi, I.; Ben Amara, M.; TT, Q.; Khalilpoor, N. Forecasting of energy-related carbon dioxide emission using ANN combined with hybrid metaheuristic optimization algorithms. *Eng. Appl. Comput. Fluid Mech.* **2024**, *18*, 2322509.
99. Stirling, J.; Chen, T.; Bucholc, M. Diagnosing Alzheimer's disease Using a self-organising fuzzy classifier. In *Fuzzy Logic Recent Applications and Developments*; Carter, J., Chiclana, F., Khuman, A.S., Chen, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; pp. 69–82.
100. Precup, R.-M.; Preitl, S.; Petriu, E.M.; Bojan-Dragos, C.-A.; Szedlak-Stinean, A.-I.; Roman, R.-C.; Hedrea, E.L. Model-based fuzzy control results for networked control systems. *Rep. Mech. Eng.* **2020**, *1*, 10–25.
101. Abadi, A.S.S.; Hosseinabadi, P.A.; Mekhilef, S. Fuzzy adaptive fixed-time sliding mode control with state observer for a class of high-order mismatched uncertain systems. *Int. J. Control. Autom. Syst.* **2020**, *18*, 2492–2508.
102. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117.
103. Sarker, I.S. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *Spring Nat. Comput. Sci.* **2021**, *2*, 420.
104. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53.