Received November 8, 2018, accepted December 6, 2018, date of publication January 15, 2019, date of current version February 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2893141

Evolving Rule-Based Explainable Artificial Intelligence for Unmanned Aerial Vehicles

BLEN M. KENENI[1], DEVINDER KAUR[1], (Life Member, IEEE), ALI AL BATAINEH[1], VIJAYA K. DEVABHAKTUNI[1], (Senior Member, IEEE), AHMAD Y. JAVAID[1], (Member, IEEE), JACK D. ZAIENTZ[2], (Associate Member, IEEE), AND ROBERT P. MARINIER, III[2]

[1]Electrical Engineering and Computer Science Department, The University of Toledo, Toledo, OH 43606, USA
[2]Soar Technology, Inc., Ann Arbor, MI 48105, USA

Corresponding author: Devinder Kaur (devinder.kaur@utoledo.edu)

This work was supported in part by the Soar Technology, Inc., and in part by the Air Force Research Laboratory under Award FA8650-16-C-6768.

ABSTRACT In this paper, an explainable intelligence model that gives the logic behind the decisions unmanned aerial vehicle (UAV) makes when it is on a predefined mission and chooses to deviate from its designated path is developed. The explainable model is on a visual platform in the format of if-then rules derived from the Sugeno-type fuzzy inference model. The model is tested using the data recorded from three different missions. In each mission, adverse weather, conditions and enemy locations are introduced at random locations along the path of the mission. There are two phases to the model development. In the first phase, the Mamdani fuzzy model is used to create rules to steer the UAV along the designated mission and the rules of engagement when it encounters weather and enemy locations along and near its chosen mission. The data are gathered as UAV traverses on each mission. In the second phase, the data gathered from these missions are used to create a reverse model using a Sugeno-type fuzzy inference system based on the subtractive clustering in the data. The model has seven inputs (time, x-coordinate, y-coordinate, heading direction, engage in attack, continue mission, and steer UAV) and two outputs (weather conditions and distance from the enemy). This model predicts the outputs regarding the weather conditions and enemy positions whenever UAV deviates from the predefined path. The model is optimized with respect to the number of rules and prediction accuracy by adjusting subtractive clustering parameters. The model is then fine-tuned with ANFIS. The final model has six rules and root mean square error value that is less than 0.05. Furthermore, to check the robustness of the model, the Gaussian random noise is added to a UAV path, and the prediction accuracy is validated.

INDEX TERMS Explainable artificial intelligence (XAI), fuzzy logic, ANFIS, unmanned aerial vehicle (UAV), subtractive clustering.

I. INTRODUCTION

Unmanned Air Vehicles(UAVs) are used for many purposes including agriculture, industry, law enforcement, and defense. These autonomous systems have several advantages over manned aerial vehicles as not only they reduce expenses by avoiding human error, but they also save the lives of fighter jet pilots.The incoming generation of artificial intelligence(AI) systems are showing significant success through the use of various machine learning techniques. These systems offer a wide range of benefits when it comes to simplifying the lives of individuals as well as military operations. Continued advances promise to produce autonomous systems that will perceive, learn, decide, and act on their own. However, the effectiveness of today's AI systems is limited by the inability of the machine to explain its decisions and actions to human users [1]–[3]. This is where the concept of Explainable Artificial Intelligence (XAI) comes in to play. XAI aims to create a suite of machine learning techniques that will produce more explainable models while maintaining a high level of learning performance (prediction accuracy). As a rule, XAI enables human users to understand, appropriately trust, and effectively manage [1], [2] the emerging generation of artificially intelligent partners.

The central problem of machine learning models is that they are regarded as black-box models. Meaning, even if we understand the underlying mathematical principles, they lack an explicit declarative knowledge representation [1], [3]. The lack of knowledge representation and explainable features initiated a rising legal, ethical, and privacy concerns. As a result, applying black-box approaches in business, personal, or military operations is becoming unfavorable. Thus, triggering the need for systems that equip machines with transparent and understandable attributes. This does not necessarily imply a ban on automatic learning approaches or an obligation to explain everything at all times. Instead, the goal is to include a possibility to make results re-traceable on demand [3].

There are two distinct methods of achieving explainability i) making the entire decision process transparent and comprehensible (global explainability) ii) explicitly providing an explanation for each decision [4]. This paper discusses methods used to develop a model to achieve explainability using the latter option. Providing an explicit explanation for each decision using the rule-based method was found to be a reasonable approach. For that reason, this paper gives a brief introduction of a Fuzzy Logic, a rule-based method that resembles human reasoning.

II. FUZZY LOGIC

The approach of fuzzy logic imitates the decision making in humans. Unlike two-valued Boolean logic, fuzzy logic is multi-valued. It deals with degrees of membership and degrees of truth [5], [6]. A fuzzy rule can be defined as a conditional statement in the form IF *x is A* THEN *y is B*; where *x* and *y* are linguistic variables; and *A* and *B* are linguistic values determined by fuzzy sets on the universe of discourses X and Y, respectively.

In order for a fuzzy expert system to achieve implementation abilities, a system has to be able to obtain a single crisp solution for the output variable by first aggregating all output fuzzy sets into a single output fuzzy set and then defuzzifying the resulting fuzzy set into a single number [6], [7]. This is achieved through Fuzzy Inference systems (FIS). FIS is a process of mapping from a given input to an output, using the theory of fuzzy sets. The two main types of FIS are Mamdani and Sugeno systems discussed in the next sections.

A. MAMDANI FUZZY INFERENCE

The structure of any Mamadani system looks as follows: IF *x is A* and *y is B* THEN *z is C*; where *x* and *y* are linguistic input variables and is Z is the linguistic output. These input and output variables could be stated in language form. An example of that can be stating temperature measures as being "cold, medium, hot" or distance measures as "far, intermediate, close" and so on.

The Mamdani inference process is performed in four steps:

1) Fuzzification of input variables
2) Rule evaluation
3) Aggregation of the rule output
4) Defuzzification

Mamdani method is widely accepted for capturing expert knowledge. It allows users to describe the expertise in more intuitive more human-like manner. For that reason, in this paper data generation (predefined UAV mission) uses a Mamdani type FIS to capture human knowledge and enable UAV to make autonomous decisions. However, this FIS entails a substantial computational burden. Therefore, for the second and main part of this research (i.e., explaining the decisions taken during a mission) Sugeno FIS is developed.

B. SUGENO FUZZY INFERENCE

In the Sugeno model, the inputs are represented as fuzzy sets, but the output of fuzzy rules is a first-order polynomial of the of the form $f(x, y) = ax + by + c$. Sugeno model can be stated as follows: IF *x is A* and *y is B* THEN *z is f(x,y)*; where *x* and *y* are fuzzy sets on a universe of discourse, and *z* is an output in a form of mathematical function *f(x,y)* [6].

Sugeno method is computationally effective and works well with optimization and adaptive technique, which will be useful particularly in dynamic nonlinear systems [8]. That makes the system a great candidate to create a hybrid system that will not only effectively explain the autonomous decision, but also learn from gathered data (experience). Grammatical Evolution has been used to evolve the structure of fuzzy rules [9], and in [10] the rules are visualized in parallel coordinates. In [11] grammatical evolution is used to evolve the fuzzy rules based on KDD99 intrusion dataset for the classification of both normal and abnormal traffic. In [12], the Grammatical Evolution approach is used for the structure identification of a non-linear response of the laser lap welding process.

The merger of a neural network with a fuzzy logic into one integrated system, therefore, offers a promising approach to building intelligent systems. Integrated neuro-fuzzy systems can combine the parallel computation and learning abilities of neural networks with the human-like knowledge representation and explanation abilities of fuzzy systems [6], [13]. As a result, neural networks become more transparent, while fuzzy systems become capable of learning.

C. PROPOSED EXPLANATION TECHNIQUE

The proposed Explainable Artificial Intelligence (XAI) focuses on improving explainability while maintaining a high level of learning performance for a range of machine learning techniques. There is an inherent tension between machine learning performance (predictive accuracy) and explainability. Often the highest performing methods (e.g., deep learning) are the least explainable, and the most explainable (e.g., decision trees) are less accurate [14]. To cover the performance versus explainability trade-off, this paper introduces the use of a combination of fuzzy logic (highly explainable) with Neural Networks (accurate prediction). To the best of our knowledge, this is the first time ANFIS is proposed to develop an XAI system.

Machine learning algorithm/expert knowledge algorithm is used to train a system from previously collected data, and cause the system to make decisions (make predictions). Consequently, the hypothesis of this research is as follows:

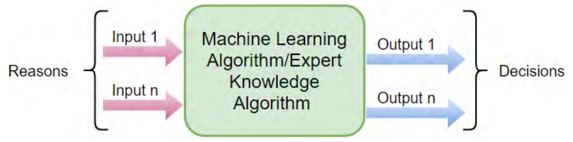


FIGURE 1. **Black-box autonomous system block diagram.**

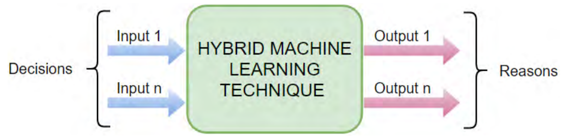


FIGURE 2. **Proposed model block diagram.**

If the output data from the original decision-making model is employed as input in a new explainable model; and the inputs from original decision-making model are used as outputs in the explainable model, then the outputs of the explainable model will serve as reasonings to an action that took place during the decision-making process.

The block diagrams in Figure 1 and Figure 2 summarize the hypothesis:

To prove the hypothesis, outputs 1 and 2 from Figure 1 are used as inputs in Figure 2. Then using ANFIS, the system is trained as seen in Figure 2 to give outputs 1 and 2 (which are the corresponding inputs 1 and 2 from Figure 1). Hence, giving an explanation of why a decision was made in the first place. In other words, the outputs of Figure 2 are the reasons why decisions are made.

To summarize, the goal of this paper is to use a machine learning algorithm that will be able to learn as well as give reasoning to why it is making a prediction.

III. LITERATURE ACQUISITION AND ANALYSIS

This section provides detailed background study of different models that can be used as a development mechanism for XAI.

A. BUILDING EXPLAINABLE ARTIFICIAL INTELLIGENCE

A variety of techniques have been considered to generate a portfolio of methods that will provide future developers with a range of design options covering the performance-versus-explainability trade-off. Recent research is heading in a promising direction when investigating ways to build effective XAI system. Even though the on-going research in XAI does not provide a complete solution, three possible models that are profoundly explored by researches as XAI development methods are discussed below.

1) MODEL INDUCTION

Model Induction can be used to develop techniques that could use any given machine learning model as a black box to infer an approximate explainable model. For instance, LIME (an algorithm that can explain the predictions of any classifier of regressors by approximating it locally with an interpretable model) was developed in order to demonstrate an example of such a model-agnostic explanation system. That was accomplished by observing and analyzing the input-output behavior of the original black-box model [15]. There are also other possible methods to develop opulent model agnostics. These methods include abduction reasoning and story generation to provide rational explanations for the reasonings of a system [16], [17]. Such model induction techniques are being explored because they indicate promising preliminary results as they could work with most machine learning applications.

2) DEEP EXPLANATION

Hybrid learning techniques that have inclusive explainable representation can be implemented using a deep explanation. Parameters such as architectural layers, optimization techniques, and training sequences can affect deep learning. These parameters are complicated and using deep explanation they can be engineered to produce more explainable representations. Recent research is showing progress when it comes to deep explanation. For example, research shows that extending the approaches used to generate image captions and train a second deep network can help generate explanation without explicitly identifying the semantic features of the original work [18], [19]. Some work has also been done where deconvolutional networks are used to visualize convolutional networks by discriminating properties of the visible object, jointly predict a class label, then explain why the predicted label is appropriate for the image [20]. These are many types of research that are trying to deliver XAI to systems using deep learning. Deep learning is one of the most used ML algorithms out there. Hence, one of the most important topics to explore in developing explainable intelligence. However, its the complexity and the large number of the hidden neurons involved in deep learning makes it difficult to integrate explainable features in the system directly.

3) INTERPRETABLE MODELS

Interpretable models should also be considered to be used in XAI as they are more structured, causal, and compact. Studies show promising findings to develop explainable intelligence by using Bayesian Rule Lists, Bayesian Program learning, stochastic grammars, learning models of casual relationships, etc. [21]–[24]. These techniques might show performance less than optimal, but they are essentially more explainable. Therefore, exploring how they can be modified to give reasonable performance is of importance.

After studying the various forms of research that are currently being conducted in areas of XAI, it is decided to focus the methodology of paper solely in using a hybrid method that would combine features from the above three models that can compensate for shortcomings of each model, and therefore give a reasonable explainable intelligent system. Consequently, in this paper, an interpretable model (i.e., fuzzy logic) is combined with a hierarchical learning method (i.e., Artificial Neural Networks). Then utilized a model induction method (i.e., reverse engineering black-box model) by applying a hybrid system to make the explanation more effective.

In the next section, the expert knowledge based UAV mission setup (data generation) methodology used for this research is discussed.

IV. DATA GENERATION

This research entails two significant components; first, UAV makes decisions on its own based on expert knowledge, and second, an explainable intelligence is evolved to justify why a decision was made at a particular time. The data generation (mission defining) is executed by creating a simulation that shows UAV navigating through different conditions to accomplish its mission. For that, a predefined path is formed in MATLAB, and the UAV's mission is to complete navigating through that path while considering external factors. These external factors (input variables) will determine the UAV decision-making process. The selected input variables are (i)weather conditions and (ii) distance between UAV and locations of an enemy. In real-world applications, more parameters need to be considered while UAV is navigating through adverse conditions. However, for this research, these two parameters are decided to be sufficient. By taking the status of input variables, the UAV will consider three different actions. The next sections elaborate more on the technical approach towards defining of UAV missions and data logging process.

A. PROCESS OF DATA GENERATION

By extracting expert knowledge, the UAV makes autonomous decisions. That is accomplished by using a Mamdani fuzzy inference system to create a set of rules. The FIS has two inputs and one output. The inputs are weather zones and distance from an enemy. The first input (weather) has five membership functions and the second input (distance from an enemy) has three membership functions. Similarly, the output (UAV decision) has three membership functions. The step by step process of creating this Mamdani FIS is elaborated below.

1) SPECIFY A PROBLEM AND DEFINE LINGUISTIC VARIABLES

The first step in creating a mamdani model is specifying the problem. As stated above, the problem statement is creating a navigational mission of UAV while it is considering external factors such as weather and distance from enemy. The linguistic variables here are:

1) Input 1 (Weather zones) : Wind, Cloud, Rain, Thunderstorm, Snow
2) Input 2 (Distance from enemy) : Too Close, Moderately Close, Far
3) Output (Decisions/Actions of UAV): Continue, Engage in Attack, Steer UAV

Figure 3 below shows the overall structure of the Mamdani FIS.

The next step is determining fuzzy sets.

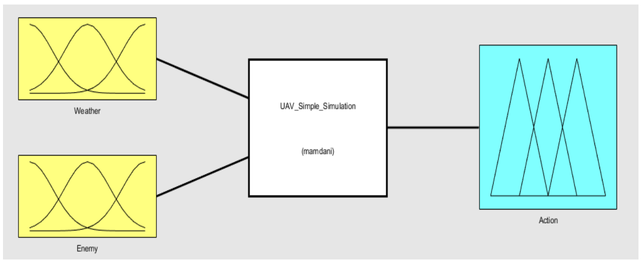


FIGURE 3. Two input one output Mamdani FIS for UAV decisions.

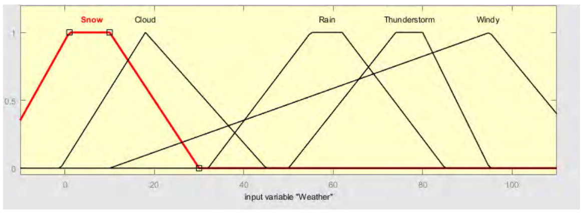


FIGURE 4. Weather zone (input 1) membership functions.

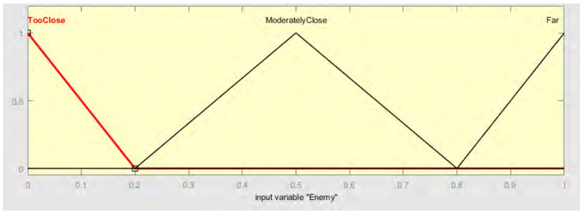


FIGURE 5. Distance from enemy (input 2) membership functions.

2) DETERMINE FUZZY SETS

The fuzzy sets here are values that take different shapes and forms to represent membership functions. Each membership function is assigned a particular shape and a range of values as seen fit by a knowledge engineer. The values given here intend to provide a close representation of conditions (knowledge representations.). However, they can be adjusted accordingly based on the most up to date information gathered by knowledge experts. Triangular functions have parameters [a b c] where a is a value of the left vertex, b is the center, and c is the right side vertex. Trapezoidal shapes have parameters [d e f g] where d is the left vertex, e is the top left edge, f is the top right edge, and g is the right vertex.

Figure 4 below shows the fuzzy sets for input variable ''Weather zone'' with a combination of trapezoidal and triangular membership functions.

As it can be seen above, the fuzzy sets for each membership function is as follows:

Snow [−16 1 10 30], Cloud [−1 18 45], Rain [32 55 62 85], and Thunderstorm [50 74 80 95], Windy [10 95 120]. These values represent temperature in Fahrenheit that represent these weather zones. Similarly, Figure 5 shows the fuzzy sets of for the input variable ''UAV's distance from enemy''.

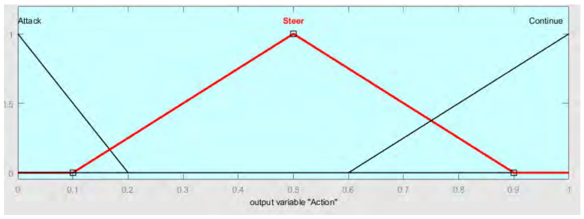


FIGURE 6. Output membership functions.

As it can be seen in Figure 5, the fuzzy set for each membership function is as follows:

Too close [−0.4 0 0.2], Moderately close [0.2 0.5 0.8], and Far [0.8 1 1.4]. These are normalized values that are arbitrarily assigned.

Finally, Figure 6 shows the fuzzy sets of the output (UAV decisions)

As it can be seen above, the fuzzy sets are triangular and are represented as follows:

Engage in attack [−0.4 0 0.2], Steer UAV [0.1 0.5 0.9], and Continue [0.8 1 1.4]. These are normalized values that are assigned for simulation.

After representing the input and output variables as fuzzy sets, the next part is constructing fuzzy rules.

3) EXTRACT FUZZY RULES

Five input membership functions in input 1 and three membership functions in input 2 give 15 total fuzzy rules. The fuzzy rules are listed below.

1) If (Weather is Snow) and (Enemy is too Close) then (Action is Steer)
2) If (Weather is Snow) and (Enemy is Moderately Close) then (Action is Continue)
3) If (Weather is Snow) and (Enemy is Far) then (Action is Continue)
4) If (Weather is Cloud) and (Enemy is Too Close) then (Action is Attack)
5) If (Weather is Cloud) and (Enemy is Moderately Close) then (Action is Continue)
6) If (Weather is Cloud) and (Enemy is Far) then (Action is Continue)
7) If (Weather is Rain) and (Enemy is too Close) then (Action is Attack)
8) If (Weather is Rain) and (Enemy is Moderately Close) then (Action is Steer)
9) If (Weather is Rain) and (Enemy is Far) then (Action is Continue)
10) If (Weather is Thunderstorm) and (Enemy is too Close) then (Action is Steer)
11) If (Weather is Thunderstorm) and (Enemy is Moderately Close) then (Action is Steer)
12) If (Weather is Thunderstorm) and (Enemy is Far) then (Action is Steer)
13) If (Weather is Wind) and (Enemy is too Close) then (Action is Steer)
14) If (Weather is Wind) and (Enemy is Moderately Close) then (Action is Attack)
15) If (Weather is Wind) and (Enemy is Far) then (Action is Continue)

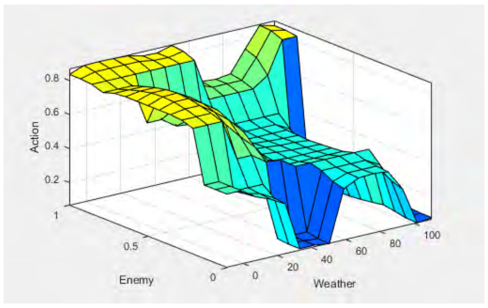


FIGURE 7. Surface view of tuned system.

4) INTEGRATE FUZZY INFERENCE INTO THE EXPERT SYSTEM

Encode the Fuzzy Sets, Fuzzy Rules, and Procedures in fuzzy inference by combining fuzzy rules with fuzzy operator *"AND"*. In the case of operator *"AND"*, the minimum of the two membership functions is the final result. For instance, if rule 1 fires and membership function (μ) of weather zone is snow with a strength of 0.6, and the membership function of distance from the enemy (close) is 0.4. the firing strength of that rule will be $\mu_{Action} = min[\mu_{Weatherzone}, \mu_{Distancefromenemy}] = \min[0.6, 0.4] = 0.4$.

5) EVALUATE AND TUNE THE SYSTEM

The surface view of the system in Figure 7 is used to analyze the input-output relationship.

By using the rules extracted from this Mamdani FIS, a MATLAB program that incorporates these rules is written to run simulations of UAV mission. The data generation process using the MATLAB program is discussed in the next section.

B. MULTI-COMPLEX UAV PATHS

Three different UAV missions with varying complexity are predefined to test the accuracy and robustness of explanation given by the proposed model. A path is set for UAV to navigate through and reach its destination. The UAV navigates in x and y coordinates. For the purposes of the paper, it is assumed that UAV is flying at a constant altitude. Then various adverse weather conditions and enemies are introduced to the simulation. Under certain conditions, the UAV would have to attack the enemy or change its path. It can also detect enemy as well as difficult weather conditions and still decide to continue the mission. When the continue flag is set to high (1), that means something interesting has occurred, but UAV still decided to continue the mission. When UAV is steering (changing direction) the final path of simulation displays that it travels underneath the rectangularly shaped weather conditions. Whereas when no interesting events occur, the UAV follows the top path of the rectangularly shaped weather condition.

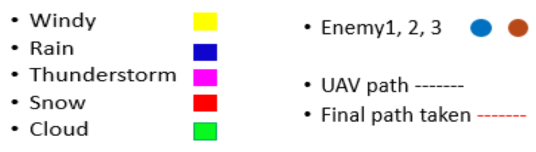


FIGURE 8. Keys for simulation diagrams.

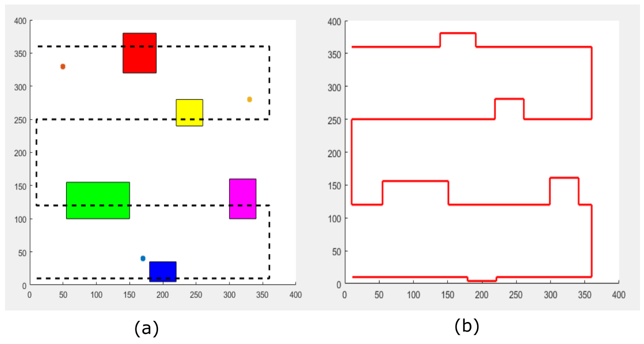


FIGURE 9. Mission 1 (a) mission set up (b) final path taken by UAV.

Note: Use the keys below to interpret what is represented by each color in the mission diagrams

1) FIRST MISSION

In the first mission, the UAV is set to navigate in the environment displayed in Figure 9 (a). On the right, Figure 9 (b) shows the final step UAV has taken. As it can be seen from Figure 9, UAV travels taking into consideration five adverse weather conditions displayed in different sized and colored rectangular shapes. Three enemies are also introduced in the system in forms of small dots. The UAV begins its mission to travel in the predefined path displayed below by dashed black lines. Figure 9 (b) shows that when the UAV enters rainy weather condition (shown in blue) and an enemy is moderately close; it decides to steer (went underneath the rainy zone). That episode occurs around (200, 10) x and y coordinates respectively.

Other decisions the UAV makes cannot be displayed in the images below. Rather they are logged along with other important information that was displayed in Figure 12. The simulation set for the first mission logged data of 1990 rows. That data is used in the next section to evolve explanation.

2) SECOND MISSION

In the second mission, similar procedures are followed to introduce weather conditions and enemies. However, this time the UAV is set to navigate more steps in a more complicated path. This simulation created 13223 rows of data. The Enemies in this mission are placed inside of rainy condition (shown in blue) and snowy condition (shown in red). The path for the simulation is setup is in Figure 10 (a). The final path it has taken can be seen from the Figure 10 (b). Whenever UAV is entering an adverse weather condition, it follows the top section of the rectangularly shaped weather condition. The data collected from this simulation is used to explain all

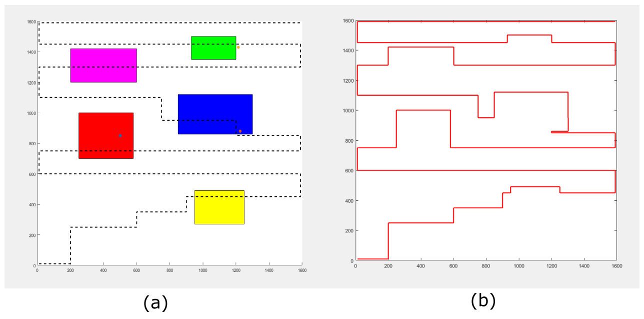


FIGURE 10. Mission 2 (a) mission set up (b) final path taken by UAV.

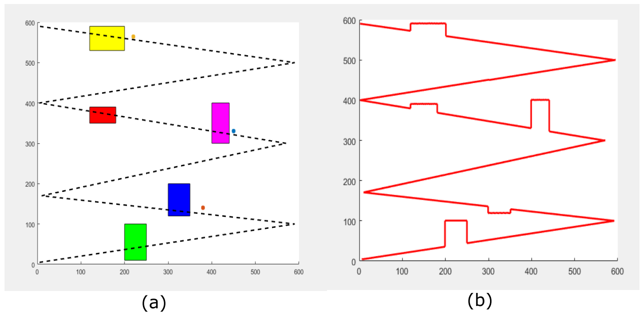


FIGURE 11. Mission 3 (a) mission set up (b) final path taken by UAV.

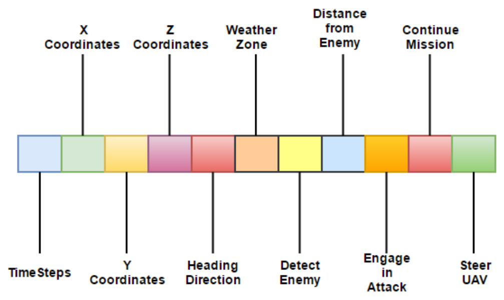


FIGURE 12. Process of data logging.

the interesting episodes that occurred during the time UAV is accomplishing mission two.

3) THIRD MISSION

In the third mission, a unique path is created where UAV has to navigate in a zigzag-shaped path. The simulation logged 3821 rows of data. Similar to the first two missions, the UAV makes the decisions based on the fuzzy rules it is programmed with. Figure 11 (a) and (b) show UAV steers away from the mission (goes under weather condition) when it approaches rainy weather zone, and an enemy is nearby. That is displayed approximately around (350, 150) x and y coordinates respectively.

After completing mission, the most essential parameters are recorded in the order displayed in Figure12. As discussed earlier, the goal of XAI is not to explain everything that happened in autonomous systems. Rather, it is meant to give explanations to interesting (selected) events. In this paper, the 11 parameters displayed in Figure 12 are selected as interesting events that occurred during the missions of the UAV. Then, those parameters are used to develop an XAI system.

Once the data generation process is completed, the next step is selecting a decision, and explaining which episode led to a specific decision. The next section gives a detailed explanation of the results by utilizing the explainable model created.

V. EVOLVING EXPLANATION USING ANFIS

In this section ANFIS models used to give explainable features of UAV decisions are studied. Further, the robustness of the system is tested by introducing noise to the UAV path, and verifying if the system identifies if there is noise in the environment. Then the results from each ANFIS model are analyzed. Finally, using Rule Viewer, an example of explanation is given for each UAV mission set in the previous section.

A. ADAPTIVE NEURO-FUZZY LOGIC

ANFIS performs neural learning using fuzzy typed numbers. Employing a set of input and output data. The adaptation function of ANFIS provides the machine learning system with neuro-fuzzy characters [25]–[28] The structure of a neuro-fuzzy system is similar to a multi-layer neural network [4]. In general, a neuro-fuzzy system has input and output layers, and three hidden layers that represent membership functions, fuzzy rules, normalized rule strengths and defuzzification process. [29].

This paper proposes developing two separate ANFIS models for the two entities that contribute to UAV decision making. Both ANFIS models have seven inputs and one output. While the inputs for both models remain the same, the output, however, varies. The output of the first ANFIS model shows weather zones, whereas the output of the second ANFIS model shows the UAVs distance from the enemy.

B. NEURAL EXPERT SYSTEM TO EXTRACT RULES

Neurons in the network are connected by links, each of which has a numerical weight attached to it [30]. The weights in a trained neural network determine the strength or importance of the associated neuron inputs; this characteristic is used for extracting rules. Neural expert systems still suffer from the limitations of Boolean logic, and any attempt to represent continuous input variables may lead to an infinite increase in the number of rules [31]–[33]. This might significantly limit the area of application for neural expert systems. The natural way of overcoming this limitation is to use fuzzy logic [34].

C. OVERVIEW OF A NEURO-FUZZY SYSTEM

The structure of a neuro-fuzzy system is similar to a multilayer neural network. In general, a neuro-fuzzy system has input and output layers, and three hidden layers that represent membership functions, fuzzy rules and normalized weights of fuzzy rules [29], [34]. Each layer in the neuro-fuzzy system is associated with a particular step in the fuzzy inference process.

1) LAYERS IN NEURO-FUZZY SYSTEMS

Layer 1 is input layer. Each neuron in this layer transmits external crisp signals directly to the next layer.

$$y_1^{(1)} = x_1^{(1)} \tag{1}$$

where $y_1^{(1)}$ and $x_1^{(1)}$ are outputs and inputs of the input layer respectively [31], [32].

Layer 2 is the fuzzification layer. Neurons in this layer perform fuzzification [6], [35]. In Jang's model, fuzzification with regular bell shape is specified as:

$$y_i^{(2)} = \frac{1}{1+(\frac{x_i^{(2)}-a_1}{c_i})^{2bi}} \tag{2}$$

where $x_i^{(2)}$ *and* $y_i^{(2)}$ are input and output of neuron i in layer 2 respectively. Moreover, a_i, b_i, and c_i are parameters of control. The center, width and slope of the bell activation function of neuron i receptively.

Layer 3 is the rule layer. Each neuron in this layer corresponds to a single Sugeno-type fuzzy rule [31]–[33]. A rule neuron receives inputs from the respective fuzzification neurons and calculates the firing strength of the rule it represents [6], [35]. In ANFIS, the conjunction of the rule antecedents is evaluated by the operator product. Thus, the output of neuron i in Layer 3 is obtained as:

$$y_i^{(3)} = \prod_{j=1}^{k} x_{ji}^{(3)} \tag{3}$$

where $y_i^{(3)}$ and $x_{ji}^{(3)}$ are the output and input of neuron i in Layer 3, (e.g. $y_{\Pi 1}^{(3)} = \mu A1 \times \mu B1 = \mu 1$), where $\mu 1$ represents the firing strength (truth value) of Rule 1.

Layer 4 is the normalization layer. Each neuron in this layer receives inputs from all neurons in the rule layer, and calculates the normalized firing strength of a given rule. The normalized firing strength is the ratio of the firing strength of a given rule to the sum of firing strengths of all rules. It represents the contribution of a given rule to the final result. Thus, the output of neuron i in Layer 4 is determined as:

$$y_i^{(4)} = \frac{x_i^{(4)}}{\sum_{j=1}^{n} x_{ji}^{(4)}} = \frac{\mu_i}{\sum_{j=i}^{n} \mu_j} = \bar{\mu}_i \tag{4}$$

where $x_{ji}^{(4)}$ is the input from neuron j located in Layer 3 to neuron i in Layer 4, and n is the total number of rule neurons (e.g. $y_{N1}^{(4)} = \frac{\mu_1}{\mu_1+\mu_2+\mu_3+\mu_4} = \bar{\mu}_1$).

Layer 5 is the defuzzification layer. Each neuron in this layer is connected to the respective normalization neuron, and also receives initial inputs, x_1 and x_2. A defuzzification neuron calculates the weighted consequent value of a given rule as,

$$y_i^{(5)} = x_i^{(5)}[k_{i0} + k_{i1}x_1 + k_{i2}x_2] = \bar{\mu}_i[k_{i0} + k_{i1}x_1 + k_{i2}x_2] \tag{5}$$

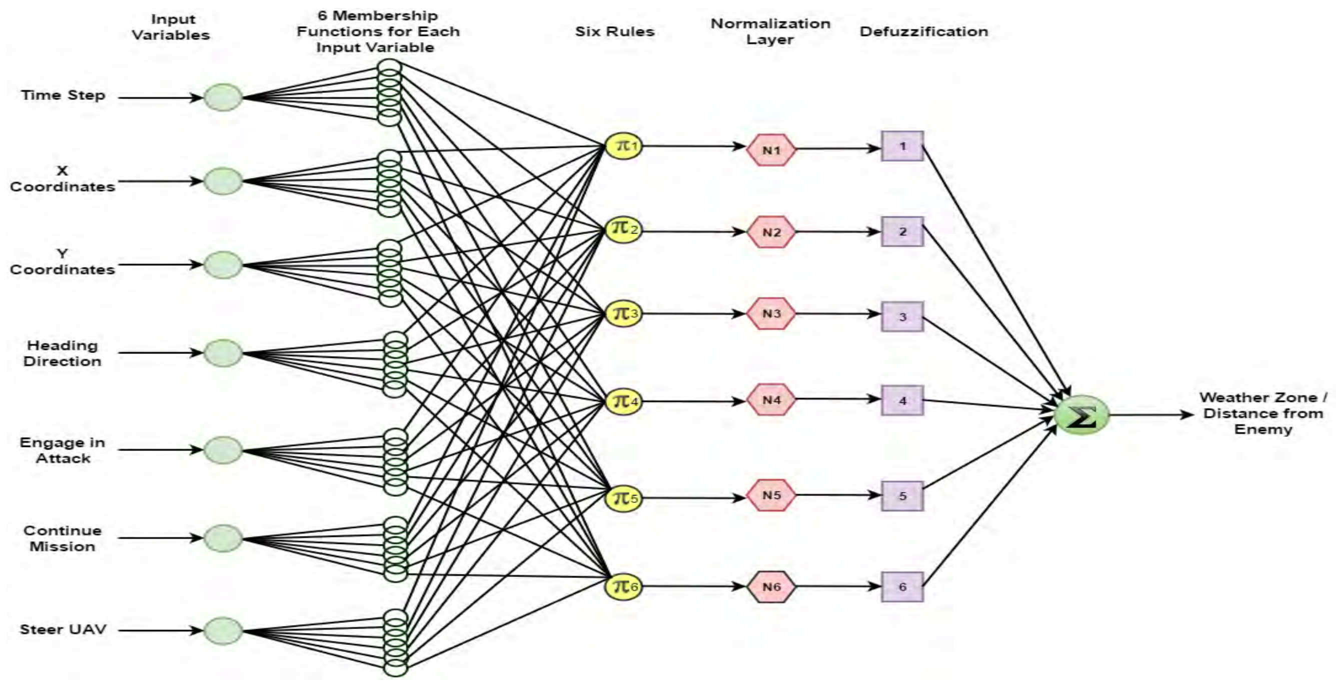


FIGURE 13. **ANFIS model structure for six rules.**

where $y_i^{(5)}$ and $x_i^{(5)}$ are the output and input of defuzzification neuron i in Layer 5, and $[k_{i0}, k_{i1}, k_{i2}]$ are a set of consequent parameters of rule i.

Layer 6 is represented by a single summation neuron. This neuron calculates the sum of outputs of all defuzzification neurons and produces the overall ANFIS output, y,

$$y = \sum_{i=1}^{n} x_i^{(6)} = \sum_{i=1}^{n} \bar{\mu}_i [k_{i0} + k_{i1}x1 + k_{i2}x2] \tag{6}$$

First-order Sugeno FIS, does not require to have any prior knowledge of rule consequent parameters of an ANFIS model. That eases the difficulty of specifying a rule consequent in a polynomial form. ANFIS learns the parameters discussed in Equations 1-6 and tunes membership functions.

D. HYBRID OPTIMIZATION TECHNIQUE

The learning algorithm for ANFIS developed in this paper is a hybrid algorithm that is a combination of gradient descent and least square methods. In the forward pass of the hybrid learning algorithm, node outputs go forwards until Layer 4, and the consequent parameters are determined by the least squares. In the backward pass, the error signals propagate backward, and premise parameters are updated using gradient descent [28], [36]–[38].

E. SUBTRACTIVE CLUSTERING

The purpose of clustering is to identify natural groupings of data from a large data set, such that a concise representation of system's behavior is produced [39]. A subtractive clustering method with improved computational effort in which data points themselves are considered as cluster center candidates has been proposed by Hájek [35], Keshavarzi *et al.* [38], and Chiu *et al.* [40]. By using this method, the computation is simply proportional to the number of data points and is independent of the dimension of the problem. The potential of data point is estimated by the following equation:

$$P_i = \sum_{j=1}^{n} e^{-\alpha |x^i - x^j|^2}, \tag{7}$$

$$\alpha = \frac{\gamma}{r_a^2}, \tag{8}$$

Here, P_i is the potential of i^{th} data point, n is the total number of data points, x^i and x^j are data vectors in data space including both input and output dimensions, γ is a positive contant and is selected as 4, and r_a is a positive constant defining the neighborhood range of the cluster or simply the radius of hypersphere cluster in data space. Each time a cluster center is obtained, the data points that are close to new cluster center are penalized in order to facilitate the emergence of new cluster centers [39]. The revising of the potential is done by subtraction as shown in the following equation:

$$P_i = P_i - P_k^* \zeta, \tag{9}$$

$$\zeta = e^{-\beta |x^i - c^k|^2}, \tag{10}$$

$$r_b = \eta * r_a, \tag{11}$$

In this case, P_k^* is the potential of k^{th} cluster center, x^i is the i^{th} data point being subtracted and c^k is k^{th} cluster center. A positive constant, r_b defines the efficient subtractive range. Squash factor η is a positive constant greater than 1.

The positive constant r_b is somewhat greater than r_a, and it helps to avoid closely spaced cluster centers.

The process of acquiring a new cluster center is based on the potential value in relation to an acceptance threshold $\bar{\epsilon}$, rejection threshold $\underline{\epsilon}$ and the relative distance criterion. A data point with the potential greater than the acceptance threshold is directly accepted as a cluster center. The acceptance of a data point with a potential between the upper and the lower thresholds depends on the relative distance equation, defined as

$$\frac{d_{min}}{r_a} + \frac{P_k^*}{P_1^*} \geq 1, \tag{12}$$

where d_{min} is the shortest distance between the candidate cluster center and all previously found cluster centers.

Once the clusters are formed in the input and output space, they are projected into each dimension. The Chiu's [40] proposal of defining fuzzy sets in input space is extended to the individual input dimensions. The exponential type membership degree in the input space is defined by the following equation:

$$\mu_j^{ik} = e^{-\alpha |x_j^i - c_j^k|^2} \tag{13}$$

$$\alpha = \frac{\gamma}{r_a^2} \tag{14}$$

Here, $|x_j^i - c_j^k|$ is the distance measure in dimension j between the i^{th} data point and the k^{th} cluster center (represented by a data point, i.e., $c_j^k = x_j^k$), and γ is a positive constant determining fuzziness in the cluster. Subtractive clustering has four parameters, namely, accept ratio $\bar{\epsilon}$, reject ratio $\underline{\epsilon}$, cluster radius r_a, and squash factor η (or r_b). There parameters have influence on the number of rules and error performance measures. Large values of $\bar{\epsilon}$ and $\underline{\epsilon}$ will result in small number of rules. Conversely, small values of $\bar{\epsilon}$ and $\underline{\epsilon}$ will increase the number of rules. A large value of r_a generally results in fewer clusters that lead to a coarse model. A small value of r_a can produce excessive number of rules that may result in an over-defined system Chiu [35], [38], [40]. The suggested values for η and r_a are $1.25 \leq \eta \leq 1.5$ and $0.15 \leq r_a \leq 0.30$.

VI. DEVELOPMENT TOOL AND PERFORMANCE CRITERIA

To implement and test the proposed architecture, a development tool is required. MATLAB Fuzzy Logic Toolbox (FLT) from MathWorks was selected as the development tool. This tool provides an environment to build and evaluate fuzzy systems using a graphical user interface. The rule viewer allows users to interpret the entire fuzzy inference process at once. The ANFIS editor GUI menu bar can be used to load a FIS training initialization, save the trained FIS, and open a new Sugeno system to interpret the trained FIS model [28]. To test the performance criteria, the root mean square error (RMSE) between the actual values and predicted values of ANFIS output are compared. RMSE can be calculated using the following equation:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Actual_i - Predicted_i)^2} \tag{15}$$

Another criteria that is used to measure performance is the accuracy percentage. For accuracy, an RMSE threshold value is set, and a simple percentage calculation is executed using the question:

$$Accuracy\ Percentage = \frac{Threshold - RMSE}{Threshold} \times 100\% \tag{16}$$

Now that the theory behind the explainable model in this paper is studied, the results are discussed in the next section.

VII. RESULTS AND DISCUSSION

The results in this section are presented by analyzing the performance of ANFIS model by comparing the RMSE values of the actual and predicted values from ANFIS models created for each UAV mission. The findings show that ANFIS can be used in multiple scenarios to evolve effective explanation features to autonomous systems.

As shown in Table 1, the sub-clustering parameters to create a Sugeno model are varied until six fuzzy rules are created. The reason six rules are determined to be sufficient is that there are five weather zone membership function in the simulation. In addition, there is the condition where UAV navigates freely, which is considered to be the sunny zone. Those weather states added together give six zones. Therefore, having six clusters is comprehensible by users. The performance of the model is then tested to see how much the RMSE will vary based on the changes in sub-clustering parameters.

In Tables 1 and 2, for each UAV mission, three attempts of varying subclustering variables are displayed, and the RMSE for each attempt are compared. As the data is scaled between 0 and 1, the acceptable threshold of RMSE for the purposes of this paper is selected to 0.05. As long as the RMSE is below this threshold, the RMSE is considered to be acceptable. The results are discussed in detail in the following subsections.

A. ANFIS MODEL ONE: WEATHER ZONE AS OUTPUT

The first ANFIS model has seven inputs and one output. The output in this model gives values for different weather zones. Each number that is given as a crisp output from the Sugeno FIS gives a number that indicates what weather conditions caused UAV to make a decision to either engage in attack with an enemy, steer (change the course of the path), or continue mission at that specific time, location and heading direction.

Table 1 shows three attempts to generate six rules by adjusting subclustering parameters without compromising the persistence of keeping low RMSE values. As it can be seen, in Mission 1, for six rules, the RMSE of checking and the predicted value is 0.000206. Whereas for 19 rules it

TABLE 1. Effects of subclustering parameters on evaluating parameters for weather output.

		Mission 1			Mission 2			Mission 3		
		1st Attempt	2nd Attempt	3rd Attempt	1st Attempt	2nd Attempt	3rd Attempt	1st Attempt	2nd Attempt	3rd Attempt
Subclustering Parameters	Range of Influence	0.5	0.6	0.7	0.5	0.6	0.6	0.5	0.6	0.65
	Squash Factor	1.25	1.5	1.55	1.25	1.4	1.5	1.25	1.35	1.35
	Accept Ratio	0.5	0.6	0.65	0.5	0.5	0.5	0.5	0.5	0.6
	Reject Ratio	0.15	0.3	0.45	0.15	0.15	0.25	0.15	0.15	0.3
Evaluating Parameters	**Number of Rules**	**19**	**10**	**6**	**21**	**10**	**6**	**11**	**10**	**6**
	Error after 10 Epoch	**0.000113**	**0.000145**	**0.000160**	**0.0000247**	**0.0000508**	**0.0000637**	**0.0000247**	**0.0000721**	**0.000110**
	Training RMSE	**0.000113**	**0.000175**	**0.000160**	**0.0000247**	**0.0000508**	**0.0000637**	**0.0000247**	**0.0000722**	**0.000110**
	Checking RMSE	**0.000134**	**0.000170**	**0.000206**	**0.0000574**	**0.0000527**	**0.0000643**	**0.0000574**	**0.0000686**	**0.000121**

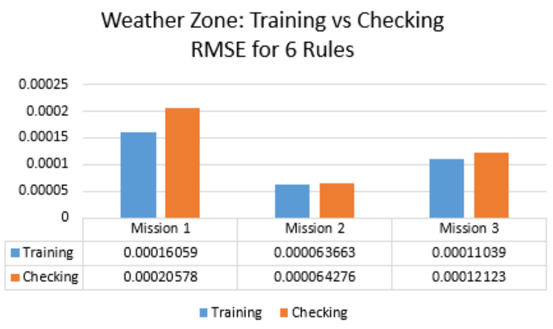
TABLE 2. Effects of subclustering parameters on evaluating parameters for distance output.

		Mission 1			Mission 2			Mission 3		
		1st Attempt	2nd Attempt	3rd Attempt	1st Attempt	2nd Attempt	3rd Attempt	1st Attempt	2nd Attempt	3rd Attempt
Subclustering Parameters	Range of Influence	0.5	0.7	0.8	0.5	0.65	0.6	0.5	0.6	0.65
	Squash Factor	1.25	1.6	1.75	1.25	1.25	1.5	1.25	1.45	1.55
	Accept Ratio	0.5	0.5	0.6	0.5	0.5	0.6	0.5	0.55	0.6
	Reject Ratio	0.15	0.25	0.3	0.15	0.15	0.3	0.15	0.25	0.3
Evaluating Parameters	**Number of Rules**	**21**	**9**	**6**	**19**	**9**	**6**	**17**	**9**	**6**
	Error after 10 Epoch	**0.00349**	**0.00438**	**0.00656**	**0.00108**	**0.00162**	**0.00171**	**0.00288**	**0.00517**	**0.00718**
	Training RMSE	**0.00352**	**0.00438**	**0.00656**	**0.00108**	**0.00162**	**0.00171**	**0.00302**	**0.00517**	**0.0718**
	Checking RMSE	**0.00428**	**0.00499**	**0.00734**	**0.00102**	**0.00154**	**0.00160**	**0.00505**	**0.00547**	**0.00733**

is 0.000134. Both these values are way below the threshold set giving an accuracy of 99.58% for six rules and 99.73% for 19 rules. There is only a percentage difference of 0.15% to reduce the number of rules by 13. Therefore, for the first mission, it can be concluded that ANFIS 1 gives excellent results for generating Sugeno FIS of only six rules.

In mission 2, generated data points are seven times more than mission one. That gives more training and checking points for the learning process. After adjusting subclustering parameters to generate six rules, ANFIS 1 gives an accuracy of 99.87% for six rules and 99.88% for 21 rules. An increase in 0.01% is very insignificant when we were able to reduce rules by 15 between checking RMSE of FIS of 21 rules to that of six rules.

In Mission 3 after adjusting subclustering parameters to generate six rules, ANFIS has 99.76% accuracy, whereas for 11 rules it gives 99.88% accuracy. The accuracy decreased by 0.12% to remove five rules, which again is not very significant. Therefore, just like in previous missions, ANFIS 1 gives excellent results for mission 3.

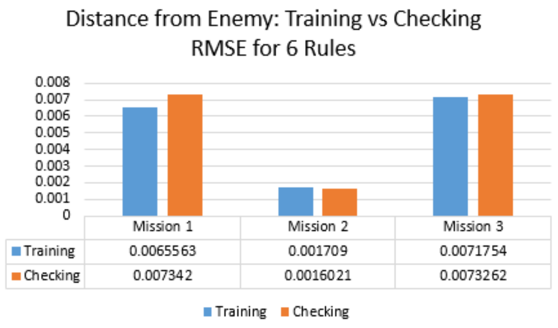


	Mission 1	Mission 2	Mission 3
Training	0.00016059	0.000063663	0.00011039
Checking	0.00020578	0.000064276	0.00012123

FIGURE 14. Comparison of ANFIS performance for missions 1, 2 and 3.

This section presented three mission with different results. A summary of the performance of ANFIS 1 for each mission is given in Figure 14.

Figure 14 shows mission 2 has the smallest RMSE for training and checking data. As expected, the model performs better when it has more data to work with. Similarly, it also gives great results for relatively small data as ANFIS 1 performed well for missions 1 and 3.



	Mission 1	Mission 2	Mission 3
🟦 Training	0.0065563	0.001709	0.0071754
🟧 Checking	0.007342	0.0016021	0.0073262

FIGURE 15. Comparison of ANFIS 2 performance for missions 1, 2 and 3.

B. ANFIS MODEL TWO: DISTANCE FROM ENEMY AS OUTPUT

ANFIS model 2 has the exact same input variables as ANFIS 1. It also follows the same methodology as ANFIS 1. For this ANFIS model, subclustering parameters are adjusted to give six rules. Six rules are selected to have a consistent number of rules generated because both entities (weather and distance) are the causes of UAV decision making.

In Table 2, after adjusting subclustering parameters to generate six rules, the RMSE of checking data set vs predicted value is 0.00734. For 21 rules RMSE is of checking data set vs predicted value 0.00428. This gives an accuracy of 85.32% for six rules and 91.44% for 21 rules. The accuracy decreased by 6.12% to reduce number of rules by 15.

Similarly, for Mission 2, after adjusting subclustering parameters to generate 6 rules, ANFIS 2 gives an accuracy of 96.8% for six rules and 97.96% for 19 rules. A decrease in 1.16% accuracy is very insignificant when the number of rules is reduced rules by 13. As stated previously, the trade-off between number of rules and accuracy has been known to be one of the limitation in fuzzy systems. However, in this ANFIS model, there is only a minimum decrease in accuracy while adjusting subclustering parameters to reduce a significant number of rules. As a result, it can be concluded that ANFIS 2 gives excellent results to model XAI for mission 2.

As it can been seen from Table 2, after adjusting subclustering parameters to generate six rules, ANFIS for six rules has 85.34% accuracy, whereas for 17 rules it gives 89.9% accuracy. The accuracy decreased by 4.56% to reduce 11 rules. Therefore, for the purpose of this research, generating six Sugeno rules is found to be paramount.

A summary of the performance of ANFIS 2 for each mission is given in Figure 15.

As it can be seen in the Figure 15 above, mission 2 has the smallest RMSE for training and checking. Just like expected, the model performs better when it has more data to work with. Similarly, it also gives great results for relatively small data as ANFIS 1 performed well for missions 1 and 2.

Next uncertainty will be added to the path of mission one and the robustness of the system will be tested by comparing the RMSE of mission 1 with and without error.

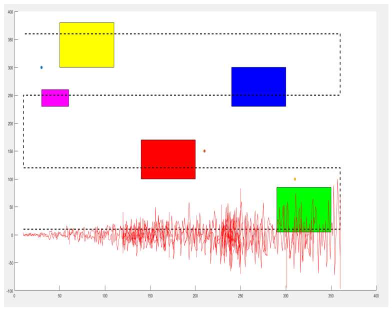
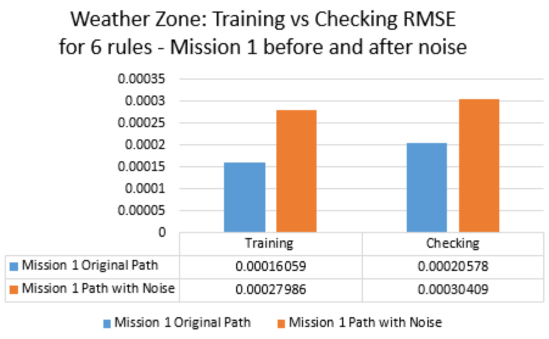


FIGURE 16. Mission 1 with random Gaussian noise added on path1.



	Training	Checking
🟦 Mission 1 Original Path	0.00016059	0.00020578
🟧 Mission 1 Path with Noise	0.00027986	0.00030409

FIGURE 17. Mission 1 for weather output with and without random noise RMSE comparison for training and checking data.

VIII. UNCERTAINTY IN PATHTAKEN

As Figure 16 shows a noise of the following equation is added to the path

$$P(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{z-\lambda^2}{2\sigma^2}} \tag{17}$$

Here $P(z)$ is the probability density function P of Gaussian random variable z, λ, and σ represent mean value and standard deviation respectively.

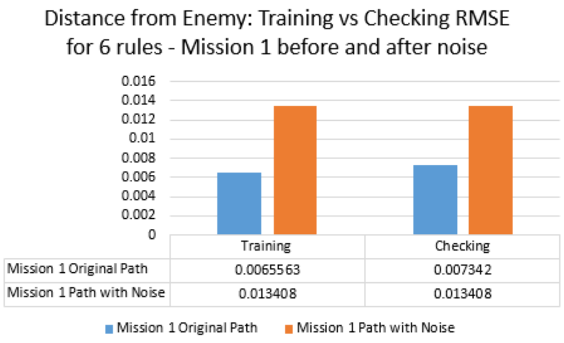
A. RESULTS AFTER UNCERTAINTY

Figure 17 and Figure 18 show a comparison of the RMSE value of mission 1 with and without uncertainty added in the system, in ANFIS model 1 (Weather zone) and ANFIS model 2(distance) respectively. As it can be observed from Figure 17 and Figure 18 when Gaussian noise is introduced to the system, the RMSE value increases in both training and checking results. This is beneficial because the system can detect that this is a glitch in the environment when the same mission shows such a significant increase in RMSE values. As a result, it provides the opportunity to correct the error.

In the next section, three examples of episodes that occurred during each mission will be discussed. As a result, the paper will show the use of Rule View for XAI purposes.

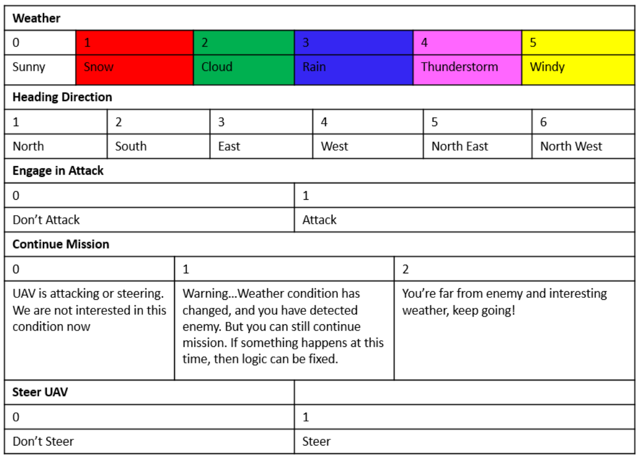
IX. EXPLAINABLE UAV DECISIONS

For every decision UAV has taken in the simulation created, the explanation can be provided by referring to the rule view window. An example of an episode from each mission is



	Training	Checking
Mission 1 Original Path	0.0065563	0.007342
Mission 1 Path with Noise	0.013408	0.013408

FIGURE 18. **Mission 1 for distance output with and without random noise RMSE comparison for training and checking data.**



Weather					
0	1	2	3	4	5
Sunny	Snow	Cloud	Rain	Thunderstorm	Windy
Heading Direction					
1	2	3	4	5	6
North	South	East	West	North East	North West
Engage in Attack					
0	1				
Don't Attack	Attack				
Continue Mission					
0	1	2			
UAV is attacking or steering. We are not interested in this condition now	Warning...Weather condition has changed, and you have detected enemy. But you can still continue mission. If something happens at this time, then logic can be fixed.	You're far from enemy and interesting weather, keep going!			
Steer UAV					
0	1				
Don't Steer	Steer				

FIGURE 19. **Rule view numerical output to English interpretation.**

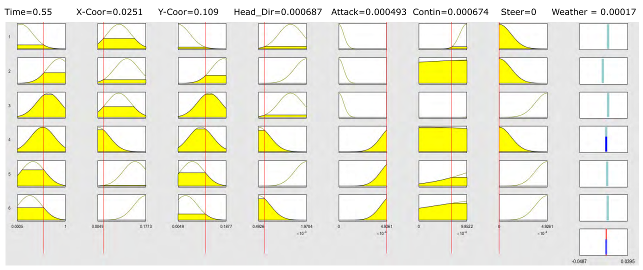


FIGURE 20. **Rule viewer for XAI weather output for mission 1.**

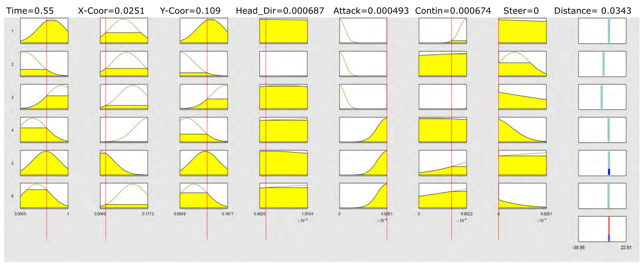


FIGURE 21. **Rule viewer for XAI distance output for mission 1 Explanation.**

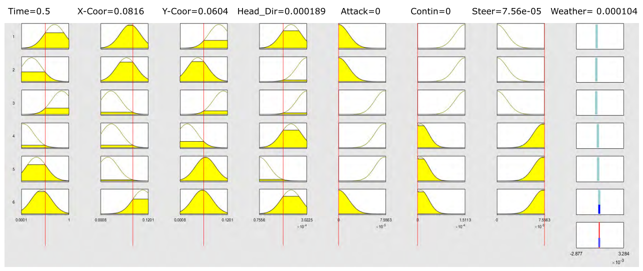


FIGURE 22. **Rule viewer for XAI weather output for mission 2.**

given. For every event, the input variables are listed as the first seven columns in rule view, whereas the output is the last column in the rule view. The rules that fired are studied to give an explanation as to why an event has occurred. If multiple rules fired for one event, the rule with the highest degree of membership is used for explanation purposes. On top of the rule view window, there are numerical representations of input variables. Those values are representations from scaled data. Therefore, they would have to be converted to their original form. To do that, multiply each numerical value of variables in rule view window by the number of data points generated in the mission (basically, multiply with the amount of time UAV takes to complete the mission). After the conversion (unscaling) is completed, the numerical values can be interpreted to linguistic explanations based on the respective values defined during UAV data generation process.

Note: use the following values to interpret numerical results to linguistic values.

The results in the next sections give three examples of explainable intelligence.

1) XAI FOR AN EVENT IN MISSION 1

In mission 1, an example to explain what caused the UAV to engage in attack with an enemy is given. The Figure 20 shows a rule view window for Sugeno model that has weather zone as output (ANFIS 1). Whereas in Figure 21 a rule view window for Sugeno model that has UAVs distance from the enemy (ANFIS 2) as output is given.

In Figure 20, rule 4 has fired, and in Figure 21, rule 5 has fired The English equivalent of this explanation is as follows:

Explanation 1: At time step 1094, x-coordinate 49.95, y-Coordinate 0.109, UAV was headed North, and it decided to engage in attack with enemy **because it was in a sunny zone and moderately close to the enemy.**

2) XAI FOR AN EVENT IN MISSION 2

This example gives explanation to what caused the UAV in mission 2 steer (change course).

In Figure 22, rule 5 has fired, and in Figure 23, rule 5 has fired.

The English Explanation is as follows:

Explanation 2: At time step 6611.5, x-coordinate 1079, y-Coordinate 789.669, UAV was headed south, and it decided to change course **because it was in a snow zone and moderately far from the enemy.**

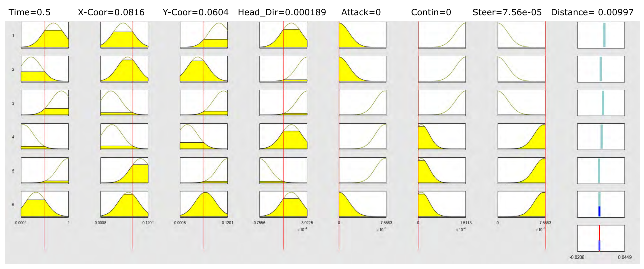


FIGURE 23. Rule viewer for XAI distance output for mission 2.

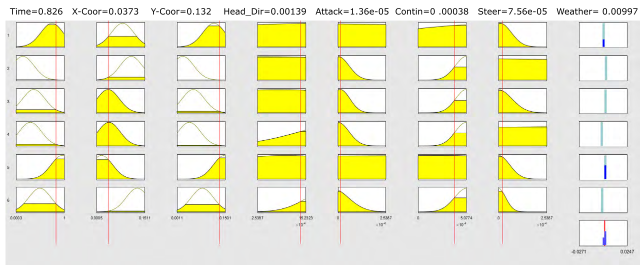


FIGURE 24. Rule viewer for XAI weather output for mission 3.

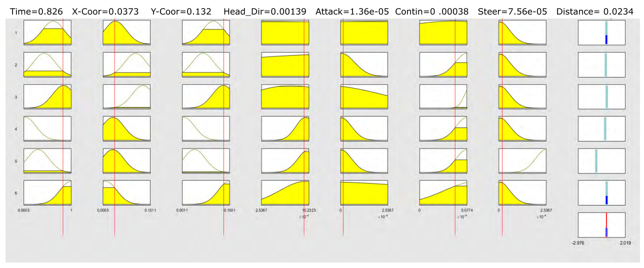


FIGURE 25. Rule viewer for XAI distance output for mission 3.

3) XAI FOR AN EVENT IN MISSION 3

This example gives explanation to what caused the UAV in mission 3 to continue mission even though it has detected enemy. In Figure 24, rules 1 and 5 have fired. The rule with largest membership function will cause the decision. That in this case, is rule 5. In Figure 25, rules 1 and 6 have fired. The rule with largest membership function will cause the decision. That in this case is rule 6. The linguistic explanation can be given as follows:

Explanation 3: At time step 3156, X-coordinate 142.523 Y-Coordinate 504.372, UAV was headed North East, and it decided to continue mission even though it detected a nearby enemy **because it was in a rainy zone and moderately close to the enemy.**

X. DEVELOP AN EASILY ACCESSIBLE RULE VIEW HCI INTERFACE

Currently, in the XAI model studied in this paper, users have to manually interpret the numerical values that are displayed in the Rule View window to give explanations to decisions. Therefore, developing an interface that allows easy human-computer interaction for this explainable model is vital for the implications of the use of ANFIS for XAI. Moreover, we have designed and implemented an interface to interpret the values of the ANFIS model to linguistic values. It allows

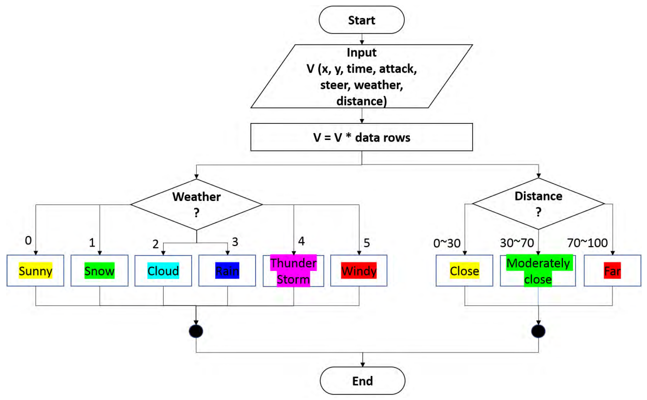
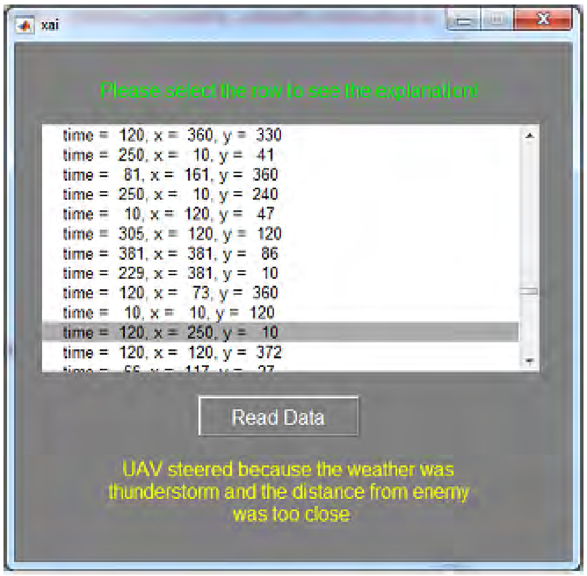


FIGURE 26. Flow chart diagram, describing stages design of the interface.



Please select the row to see the explanation!

```
time =  120, x =  360, y =  330
time =  250, x =   10, y =   41
time =   81, x =  161, y =  360
time =  250, x =   10, y =  240
time =   10, x =  120, y =   47
time =  305, x =  120, y =  120
time =  381, x =  381, y =   86
time =  229, x =  381, y =   10
time =  120, x =   73, y =  360
time =   10, x =   10, y =  120
time =  120, x =  250, y =   10
time =  120, x =  120, y =  372
```

UAV steered because the weather was thunderstorm and the distance from enemy was too close

FIGURE 27. Interface design and outputs/UAV steer.

the user to understand the explanation of the UAV behavior in human-reasoning logic. The interface has been designed and developed in the MATLAB platform that is capable of explaining the decision of the UAV at each point of time and location. The workflow diagram of the proposed interface is shown in Figure 26. Figure 27 shows an example of the interface output; here the user can easily choose any raw data to see the explanation of the UAV at a given location and time.

XI. CONCLUSION

The goal of this research is to investigate a method to evolve an explanation of the decisions taken by autonomous systems. The autonomous system that is selected in this research is Unmanned Aerial Vehicle. By using Mamdani FIS inspired method, a simulation that incorporated 15 fuzzy rules is generated. During each mission, different weather conditions and enemies are placed at random, and the UAV navigates its predefined path by taking those entities into consideration and following the 15 fuzzy rules. Two ANFIS models of seven input and one output are created. The first model has output of weather zone and the second model has an output of distance from the enemy. By adjusting subtractive clustering parameters to extract a different number of rules with different RMSE values between actual data and predictive data are generated.

The best optimization was found to be generating six fuzzy rules for the Sugeno model. In all of the missions having six rules is found to be easily comprehensible and the RMSE stays within a limit of acceptable value. The highest checking RMSE found (the worst case) is 0.00734 during mission 1 distance output model. Based on the RMSE threshold value (0.05) set for this research, that model gives an accuracy of 85.32%. The lowest checking RMSE (the best case) is 0.000064 which was achieved in mission 2 weather zone model. Based on the RMSE threshold set for this paper, this model gives an accuracy of 99.87%. The explanation is given using three examples for three missions.

In the paper, it is shown how ANFIS can be used to develop XAI. As the findings are showing promising results, models as such should be considered to be integrated with upcoming UAV technologies in order to make the UAVs more transparent, easily understandable, and trustworthy.

XII. FUTURE WORK

As demonstrated in this paper, the finds from applying ANFIS to develop explainable artificial intelligence show promising results. The research thereof has the potential to diverge into many possible future directions.

A. REAL-TIME APPLICATION

Acquiring real-time data and analyzing the performance of this ANFIS model with that data can bring the research towards XAI a step closer to acquiring the necessary tools to achieve not only smart and self sustaining machines that can learn from experience, but also smart systems that can explain their autonomous decisions to human users adequately.

B. EXPLAINABILITY METRICS

It will be very useful to develop a metric that defines explainability in a quantitative manner. If future research can find quantifiable metrics for XAI, it will facilitate the development of a faster, reliable, and accurate explainable artificial intelligence systems.

REFERENCES

[1] D. Gunning, "Explainable artificial intelligence (XAI)," in *Proc. Defense Adv. Res. Projects Agency (DARPA)*, 2017, pp. 2–6.

[2] *Explainable Artificial Intelligence (XAI)*, Defense Adv. Res. Projects Agency, Arlington, VA, USA, Aug. 2016.

[3] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell. (2017). "What do we need to build explainable AI systems for the medical domain?" [Online]. Available: https://arxiv.org/abs/1712.09923

[4] H. K. Dam, T. Tran, and A. Ghose, "Explainable software analytics," in *Proc. 40th Int. Conf. Softw. Eng., New Ideas and Emerg. Results (ICSE-NIER)* Feb. 2018, pp. 53–56.

[5] N. J. Nilsson, *Principles of Artificial Intelligence*. San Mateo, CA, USA: Morgan Kaufmann, 2014.

[6] M. Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems*. London, U.K.: Pearson, 2005.

[7] E. Wenger, *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. San Mateo, CA, USA: Morgan Kaufmann, 2014.

[8] J. Yen and R. Langari, *Fuzzy Logic: Intelligence, Control, and Information*, vol. 1. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.

[9] D. A. Wilson and D. Kaur, "Fuzzy classification using grammatical evolution for structure identification," in *Proc. IEEE Annu. Meeting North Amer. Fuzzy Inf. Process. Soc. (NAFIPS)*, Jun. 2006, pp. 80–84.

[10] M. Abdulgader and D. Kaur, "Parallel coordinates for visualization of rules developed using grammatical evolution," in *Proc. Int. Conf. Modeling, Simulation Vis. Methods (MSV), Steering Committee World Congr. Comput. Sci., Comput. Eng. Appl. Comput. (WorldComp)*, 2014, p. 1.

[11] D. Wilson and D. Kaur, "Using grammatical evolution for evolving intrusion detection rules," *WSEAS Trans. Syst.*, vol. 6, no. 2, p. 346, 2007.

[12] D. Wilson, D. Kaur, M. Forrest, and F. Lu, "A grammatical evolution approach to system identification of laser lap welding," SAE Tech. Paper 2006-01-1614, 2006.

[13] L. A. Zadeh, "From computing with numbers to computing with words: From manipulation of measurements to manipulation of perceptions," in *The Dynamics of Judicial Proof*. Warsaw, Poland: Springer, 2002, pp. 81–117.

[14] A. Holzinger, R. Goebel, V. Palade, and M. Ferri, *Towards Integrative Machine Learning and Knowledge Extraction*. Banff, AB, Canada: Springer, 2017, pp. 1–12.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Aug. 2016, pp. 1135–1144.

[16] S. Park, B. X. Nie, and S.-C. Zhu. (Jul. 2016). "Attribute and-or grammar for joint parsing of human attributes, part and pose." [Online]. Available: https://arxiv.org/abs/1605.02112

[17] T. Lombrozo, "Explanation and abductive inference," in *The Oxford Handbook of Thinking and Reasoning*. Oxford, U.K.: Princeton Univ. Library, Oxford Univ. Press, 2017.

[18] Q. Yu, J. Liu, H. Cheng, A. Divakaran, and H. Sawhney, "Multimedia event recounting with concept based representation," in *Proc. 20th ACM Int. Conf. Multimedia (MM)*, New York, NY, USA, 2012, pp. 1073–1076.

[19] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. (Mar. 2016). "Generating visual explanations." [Online]. Available: https://arxiv.org/abs/1603.08507

[20] M. D. Zeiler and F. Rob, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, Nov. 2013, pp. 818–833.

[21] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model," *Ann. Appl. Statist.*, vol. 9, no. 3, pp. 1350–1371, Nov. 2015.

[22] B. M. Lake, R. Salakhutdinov, and J. B. Tenebaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[23] S. Lee and V. Honavar, "On learning causal models from relational data," in *Proc. 13th AAAI Conf. Artif. Intell.* Phoenix, AZ, USA: AAAI Press, 2016, pp. 3263–3270.

[24] W. Brendel and S. Todorovic, "Learning spatiotemporal graphs of human activities," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 778–785.

[25] N. Walia, H. Singh, and A. Sharma, "ANFIS: Adaptive neuro-fuzzy inference system—A survey," *Int. J. Comput. Appl.*, vol. 123, no. 13, pp. 35–37, 2015.

[26] M.-Y. Chen, "A hybrid ANFIS model for business failure prediction utilizing particle swarm optimization and subtractive clustering," *Inf. Sci.*, vol. 220, pp. 180–195, Jan. 2013.

[27] A. Bagheri, H. M. Peyhani, and M. Akbari, "Financial forecasting using ANFIS networks with quantum-behaved particle swarm optimization," *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6235–6250, 2014.

[28] A. Al-Hmouz, J. Shen, R. Al-Hmouz, and J. Yan, "Modeling and simulation of an adaptive neuro-fuzzy inference system (ANFIS) for mobile learning," *IEEE Trans. Learn. Technol.*, vol. 5, no. 3, pp. 226–237, Jul./Sep. 2012.

[29] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, May/Jun. 1993.

[30] M. S. A. Aziz, M. A. M. Hassan, and E. A. Zahab, "Applications of ANFIS in high impedance faults detection and classification in distribution networks," in *Proc. IEEE Int. Symp. Diag. Elect. Mach., Power Electron. Drives (SDEMPED)*, Sep. 2011, pp. 612–619.

[31] S. I. Gallant, *Neural Network Learning and Expert Systems*. Cambridge, MA, USA: MIT Press, 1993.

[32] C. Nikolopoulos, *Expert Systems: Introduction to First and Second Generation and Hybrid Knowledge Based Systems*. New York, NY, USA: Marcel Dekker, 1997.

[33] S. Sestito and T. Dillon, "Using single-layered neural networks for the extraction of conjunctive rules and hierarchical classifications," *Appl. Intell.*, vol. 1, no. 2, pp. 157–173, 1991.

[34] I. Güler and E. D. Übeyli, "Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients," *J. Neurosci. Methods*, vol. 148, no. 2, pp. 113–121, 2005.

[35] P. Hájek, *Metamathematics of Fuzzy Logic*, vol. 4. Prague, Czech Republic: Springer, 1998.

[36] A. M. Abdulshahed, A. P. Longstaff, and S. Fletcher, "The application of ANFIS prediction models for thermal error compensation on CNC machine tools," *Appl. Soft Comput.*, vol. 27, pp. 158–168, Feb. 2015.

[37] D. Karaboga and E. Kaya, "Adaptive network based fuzzy inference system (ANFIS) training approaches: A comprehensive survey," *Artif. Intell. Rev.*, vol. 27, pp. 1–31, Jan. 2018.

[38] A. Keshavarzi, F. Sarmadian, J. Shiri, M. Iqbal, R. Tirado-Corbalá, and E.-S. E. Omran, "Application of ANFIS-based subtractive clustering algorithm in soil cation exchange capacity estimation using soil and remotely sensed data," *Measurement*, vol. 95, pp. 173–180, Jan. 2017.

[39] K. Demirli, S. Cheng, and P. Muthukumaran, "Subtractive clustering based modeling of job sequencing with parametric search," *Fuzzy Sets Syst.*, vol. 137, no. 2, pp. 235–270, 2003.

[40] S. Chiu, D. Dubois, H. Prade, and R. Yager, "Fuzzy model identification based on cluster estimation," *J. Intell. Fuzzy Syst.*, vol. 137, no. 2, pp. 235–270, 1997.

BLEN M. KENENI received the B.S. degree in electrical engineering and the M.S. degree in electrical engineering, with a focus on machine learning, from The University of Toledo, in 2016 and 2018, respectively. She co-oped at Diebold Nixdorf for two semesters, as a Systems Engineer, where she developed skills of circuit schematic and PCB layout design. She also co-oped at ZF TRW for two semesters, as a Product Engineer, where she was with ADAS Team, worked on spatial frequency and electromagnetic compatibility tests. She was an Undergraduate Researcher under NSF REU Grant, where she developed an educational prototype comprising of a set of communication nodes with call priorities to help educate students on cellular technology and future implications of spectrum sharing. She also held multiple leadership positions in campus organizations, such as NSBE, UTESA, ISA, and the IEEE during both her graduate and undergraduate studies. She is currently a Graphics Hardware Engineer with Intel, CA, USA. She has three conference paper publications in IEEE Xplore.

DEVINDER KAUR (M'89–SM'00–LM'19) received the B.Sc. and M.Sc. degrees (Hons.) in physics, majoring in electronics, from Panjab University, in 1969 and 1970, respectively, the M.Sc. degree in medical physics from The University of Aberdeen, U.K., in 1976, under the Commonwealth Scholarship Award, and the M.Sc. and Ph.D. degrees in computer engineering from Wayne State University, USA, in 1985 and 1989, respectively. She was Scientist with the Central Scientific Instruments Organization, a national laboratory, under the Ministry of Science and Technology, Chandigarh, India, from 1971 to 1981. In 1989, she joined The University of Toledo, as a Faculty Member, where she is currently a Full Professor with the Department of EECS. She visited Nippon Institute of Technology, Japan, in that capacity. She has published over 100 papers in refereed journals and proceedings of the international conferences. She has worked on projects funded by NSF, AFRL, Daimler Chrysler, and ROMAN Engineering. Her research interests include developing intelligent applications based on hybrid computational models using biologically inspired computing and fuzzy systems. She received the University Medal for obtaining the first rank. She was a recipient of the IIT Delhi Fellowship (1970–1971) and the Fulbright Senior Specialist Award, in 2004.

ALI AL BATAINEH received the B.S. degree in computer engineering from Yarmouk University, Jordan, in 2010, and the M.S. degree (Hons.) in computer engineering from the University of Bridgeport, Bridgeport, CT, USA, in 2016. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, The University of Toledo, Toledo, OH, USA. His current research interests include the areas of artificial intelligence, machine learning, computer vision, metaheuristic optimization, and fuzzy logic.

VIJAYA K. DEVABHAKTUNI (S'97–M'03–SM'09) received the B.Eng. degree in EEE and the M.Sc. degree in physics from BITS, Pilani, in 1996, and the Ph.D. degree in electronics from Carleton University, Canada, in 2003. From 2003 to 2004, he held the Canada Post-Doctoral Fellowship with the Natural Sciences and Engineering Research Council and spent the tenure researching with the University of Calgary. In 2005, he taught at Penn State Erie Behrend. From 2005 to 2008, he held the internationally prestigious Canada Research Chair in computer-aided high-frequency modeling and design with Concordia University, Montreal, QC, Canada. In 2008, he joined the EECS Department, The University of Toledo, as an Associate Professor. Since 2012, he has been the Director of the College of Engineering for Interdisciplinary Research Initiatives, where he has been recently promoted to a Full Professor. He has co-authored around 190 peer-reviewed papers and is advising 13 M.S./Ph.D. students. His interests include applied electromagnetics, biomedical applications of wireless networks, computer-aided design, device modeling, image processing, infrastructure monitoring, neural networks, optimization methods, power theft modeling and education, RF/microwave optimization, and virtual reality. In these areas, he secured external funding close to 5 million (sponsoring agencies include AFOSR, CFI, ODOT, NASA, NSERC, NSF, and industry). He is a Registered Member of the Association of Professional Engineers, Geologists, and Geophysicists of Alberta. He was a recipient of the Carleton University Senate Medal for outstanding scholastic accomplishments at the Ph.D. level. He received several teaching excellence awards in Canada and USA. He serves as an Associate Editor for the *International Journal of RF and Microwave Computer-Aided Engineering*.

AHMAD Y. JAVAID (GS'12–M'15) received the B.Tech. degree (Hons.) in computer engineering from Aligarh Muslim University, India, in 2008, and the Ph.D. degree from The University of Toledo, in 2015. He was a Scientist Fellow with the Ministry of Science and Technology, Government of India, for two years. In 2015, he joined the EECS Department, as an Assistant Professor, and is the Founding Director of the Paul A. Hotmer Cybersecurity and Teaming Research Laboratory. Since 2018, he has been the Cyber Education and Threat Mitigation Faculty Fellow with the Division of Technology and Advanced Solutions, The University of Toledo. He has published more than 50 peer-reviewed journals, conferences, and poster papers along with several book chapters. His research expertise is in the areas of cybersecurity of drone networks, smartphones, wireless sensor networks, and other systems. He is also conducting extensive research on human–machine teams and applications of AI and machine learning to attack detection and mitigation. During his time at UT, he has participated in several collaborative research proposals that have been funded by agencies, including the National Science Foundation, the Air Force Research Lab, and the State of Ohio. He has as a member of the technical program committee for several conferences. He received the prestigious University Fellowship Award. He has served as a reviewer for several high-impact IEEE journals.

JACK D. ZAIENTZ received the master's degree in human–computer interaction from Carnegie Mellon University, in 2001. His capstone project focused on visualizing Cognitive Percept Motor–Goals Operators Methods Selection Rules cognitive model behavior. In 2001, he joined Soar Technology to develop interactive visualizations for AI systems and cognitive modeling frameworks. Since 2004, he has been leading DoD applied research projects, focusing on adaptive user interfaces, cognitive visualization, and AI system explanation for a variety of domains, including command and control communications computers intelligence surveillance reconnaissance (C4ISR), autonomous platform control, and satellite operations. Since 2011, he has been a Judge of the annual IEEE Visual Analytics Software and Technology competition. He serves on the Review Board of the journal *Theoretical Issues in Ergonomics Science*. Since 2001, he has been the member of the HCI.

ROBERT P. MARINIER, III, received the Ph.D. degree in CS and engineering from the University of Michigan, in 2008. His Ph.D. thesis was on the integration of emotion and learning in the soar cognitive architecture. He has been doing DoD research for over nine years at Soar Technology. His work at SoarTech has included exploring applications of soar and episodic memory to a variety of domains, including ground robot navigation, navy MCM planning, and intelligent training. He has designed enhancements to the Soar Cognitive Architecture's Episodic Memory, many of which have been incorporated in recent projects, including ONR's SUMET, DARPA's SSIM, and ONR's TRUFAST. In SUMET, he designed and implemented mission level behaviors in Soar, including explanations of why the agent selected various actions. He also participated in RDECOM's MAGIC 2010 robotics competition as part of first-place Team Michigan, under which he was the primary implementer of the explanation user interface.