Republic of Tunisia Ministry of Higher Education and Scientific Research General Directorate of Technological Studies Higher Institute of Technological Studies of Bizerte

Department of Electrical Engineering

Neural networks and fuzzy logic Project

Prepared by

Andolsi Skander

Title of the project

Fuzzy Logic Tip Calculator

1 Python

1.1 Definition of Python :

Python is a high-level, interpreted programming language that is widely used for web development, data analysis, artificial intelligence, and scientific computing. It was first released in 1991 and has since become a popular language due to its simplicity, readability, and extensive library of pre-built tools and modules that make it easy to develop a wide range of applications. Python is known for its simplicity, which allows developers to express concepts in fewer lines of code compared to other languages, and its flexibility, which makes it well-suited for a variety of tasks. It has a large, active community of developers who contribute to the development of the language and its ecosystem of libraries and tools.

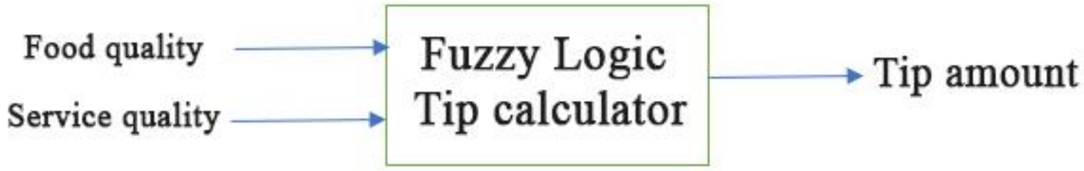
1.2 Python for Fuzzy logic system :

In Python, fuzzy logic systems can be implemented using libraries such as scikit-fuzzy, which provides tools for fuzzy logic modeling, including functions for defining fuzzy sets, membership functions, and fuzzy rules. These libraries allow developers to build and train fuzzy logic models and use them to make decisions based on input data. Fuzzy logic can be applied in Python in a variety of contexts, such as control systems, data analysis, and artificial intelligence.

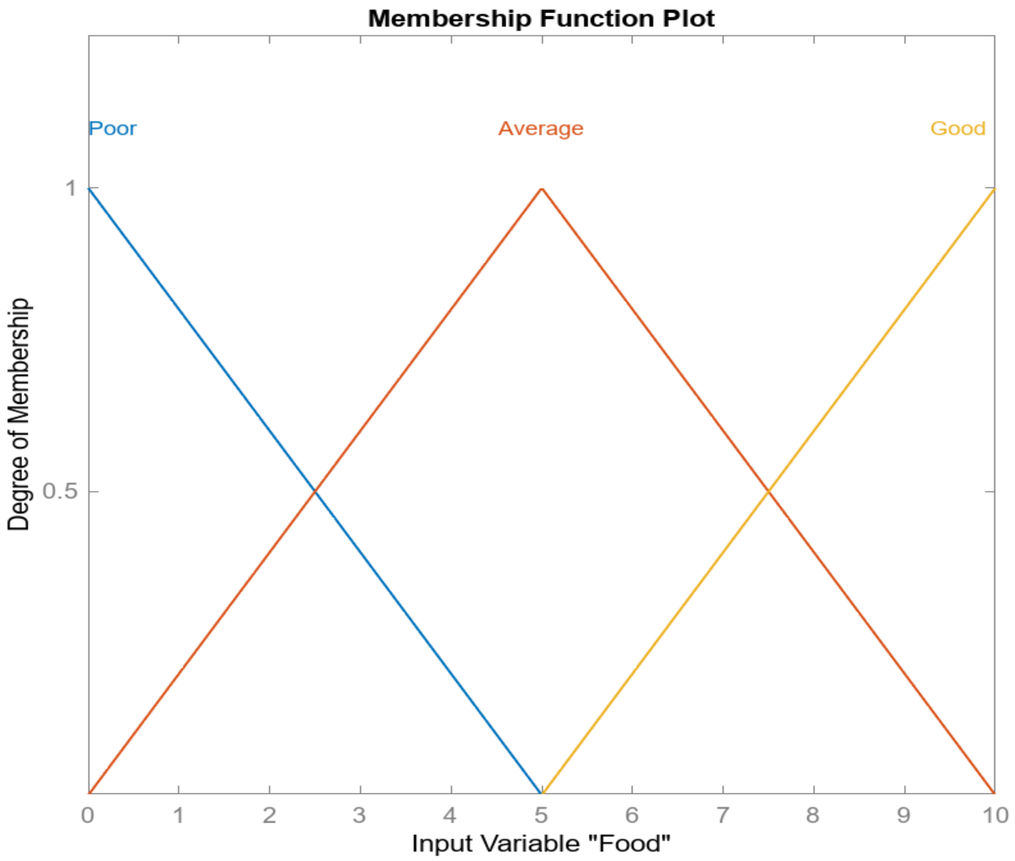
2 Context

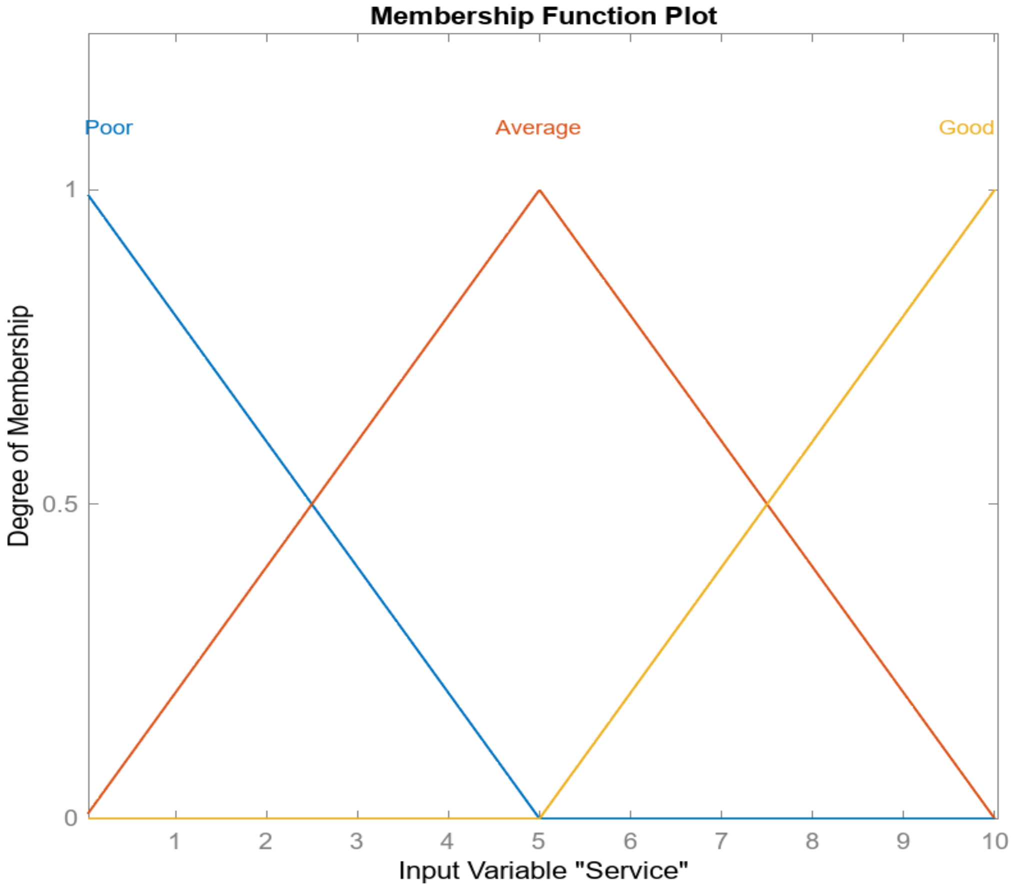
I used Fuzzy logic to build intelligent control systems that can adapt and make decisions based on changing circumstances. This can be useful in a tip calculator, where the appropriate tip amount may vary based on factors.

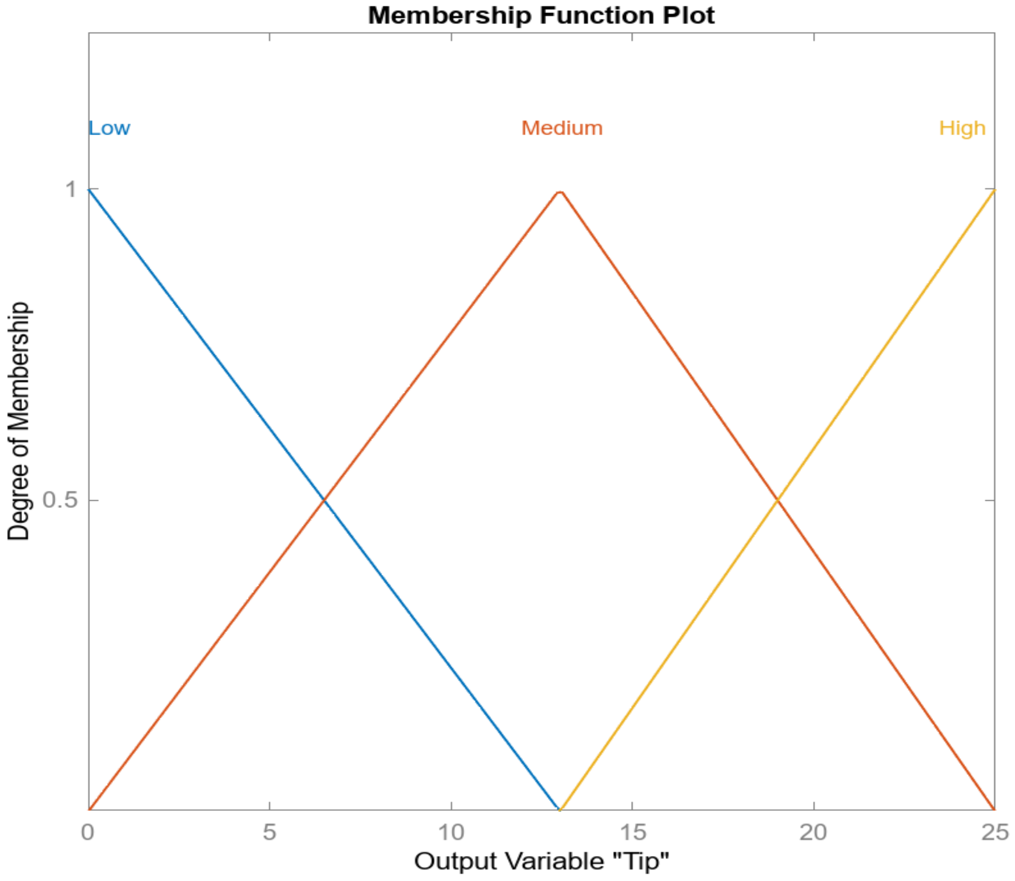
The fuzzy logic system is being used to determine a recommended tip amount based on the inputs of food quality and service quality.



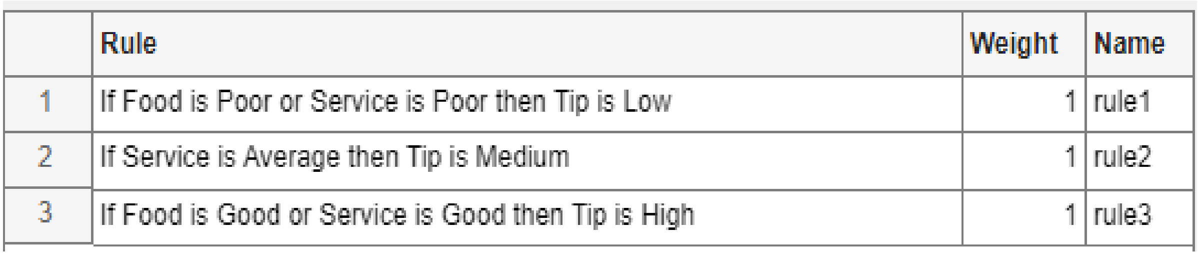
The inputs and output are defined using membership functions, which map the inputs and output to a range of values. In this case, the inputs are food quality and service quality, and the output is the recommended tip amount.




Membership Function Plot
Poor
Average
Good
1
0.5
0
Degree of Membership
1
2
3
4
5
6
7
8
9
10
Input Variable "Service"


Membership Function Plot
Low
Medium
High
1
0.5
Degree of Membership
0
5
10
15
20
25
Output Variable "Tip"

Rules table :



	Rule	Weight	Name
1	If Food is Poor or Service is Poor then Tip is Low	1	rule1
2	If Service is Average then Tip is Medium	1	rule2
3	If Food is Good or Service is Good then Tip is High	1	rule3

3 Motivation

There are several benefits to using fuzzy logic with Python:

- Simplicity: Fuzzy logic systems can be implemented in Python using a variety of libraries, making it relatively easy to develop and test fuzzy models.
- Flexibility: Fuzzy logic systems can be used to model a wide range of complex systems and make decisions based on imprecise or incomplete information.
- Efficient decision-making: Fuzzy logic systems can process large amounts of data and make decisions quickly, making them well-suited for real-time applications.
- Improved accuracy: Fuzzy logic systems can handle uncertainty and imprecision in data, which can improve the accuracy of decision-making in complex systems.
- Strong community support: Python has a large, active community of developers who contribute to the development of the language and its ecosystem of libraries and tools, including those for fuzzy logic. This makes it easier to find help and resources when working with fuzzy logic in Python.

The use of fuzzy logic with Python can help developers build more sophisticated and effective decision-making systems for a wide range of applications.

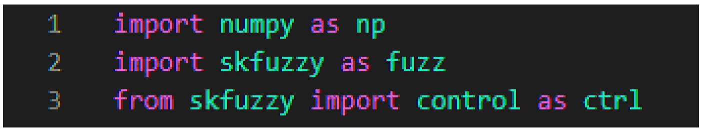
4 Project Code

❖ This code imports three libraries: **Numpy**, **Skfuzzy**, and the **control** module from **Skfuzzy** :

✓ Numpy is a library for working with large, multi-dimensional arrays and matrices of numerical data, and is a fundamental package for scientific computing with Python.

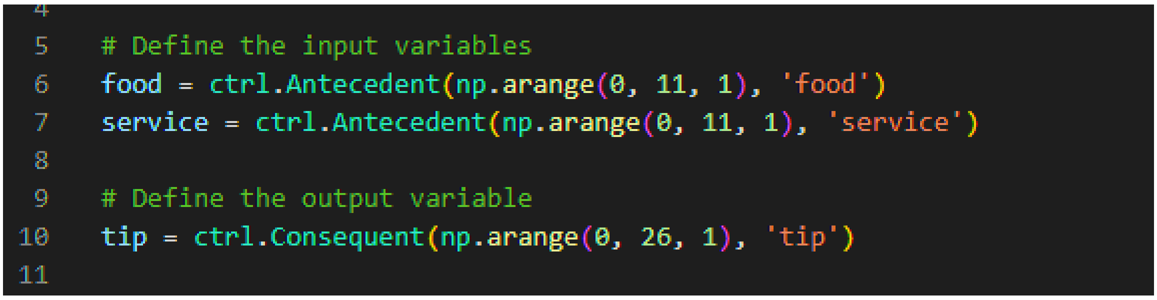
✓ Skfuzzy is a library for fuzzy logic control systems development in Python. Fuzzy logic is a mathematical approach to modeling complex, imprecise, or ill-defined systems. It allows for the representation of uncertainty and imprecision in data, and can be used to build intelligent control systems that can adapt and make decisions based on ambiguous or incomplete information.

✓ The control module from skfuzzy provides tools for creating and manipulating fuzzy control systems. This includes functions for defining fuzzy variables, rules, and membership functions, as well as tools for simulating and evaluating the performance of a fuzzy control system.



```python
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
```

❖ Define of input/output range :

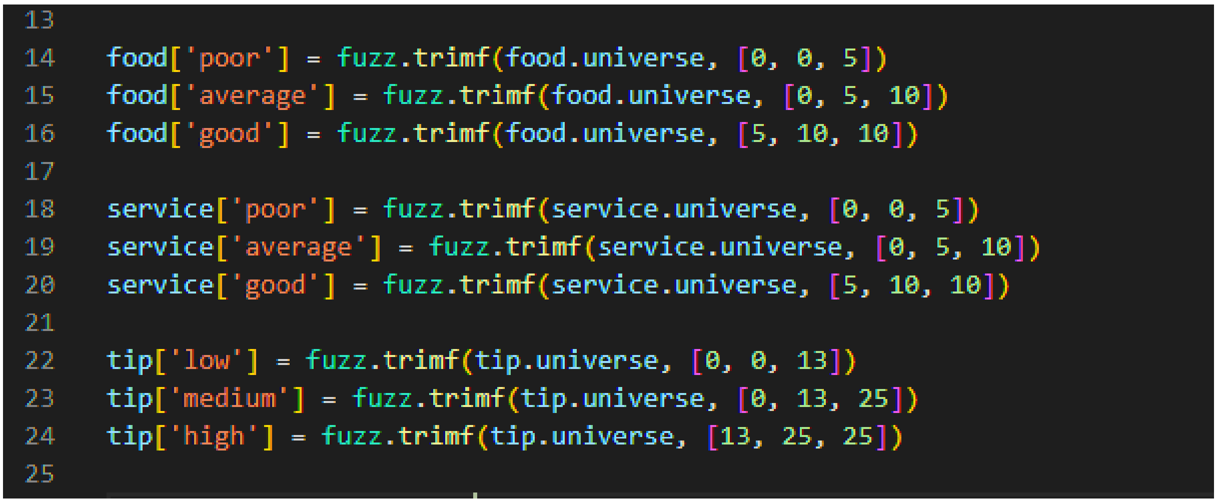


```python

# Define the input variables
food = ctrl.Antecedent(np.arange(0, 11, 1), 'food')
service = ctrl.Antecedent(np.arange(0, 11, 1), 'service')

# Define the output variable
tip = ctrl.Consequent(np.arange(0, 26, 1), 'tip')

```

The **Antecedent** class is used to define input variables, while the **Consequent** class is used to define output variables. These functions take two arguments: a NumPy array specifying the range of the variable, and a string label for the variable.

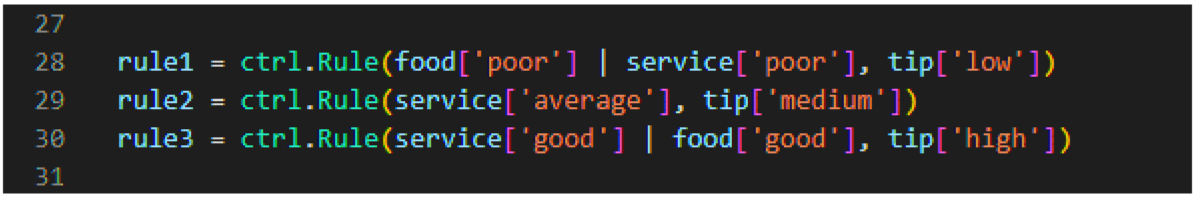
❖ Define of the membership functions of input and output value :

All the membership functions are defined using the **trimf** (triangular membership function) method from the fuzz module



```
food['poor'] = fuzz.trimf(food.universe, [0, 0, 5])
food['average'] = fuzz.trimf(food.universe, [0, 5, 10])
food['good'] = fuzz.trimf(food.universe, [5, 10, 10])

service['poor'] = fuzz.trimf(service.universe, [0, 0, 5])
service['average'] = fuzz.trimf(service.universe, [0, 5, 10])
service['good'] = fuzz.trimf(service.universe, [5, 10, 10])

tip['low'] = fuzz.trimf(tip.universe, [0, 0, 13])
tip['medium'] = fuzz.trimf(tip.universe, [0, 13, 25])
tip['high'] = fuzz.trimf(tip.universe, [13, 25, 25])
```

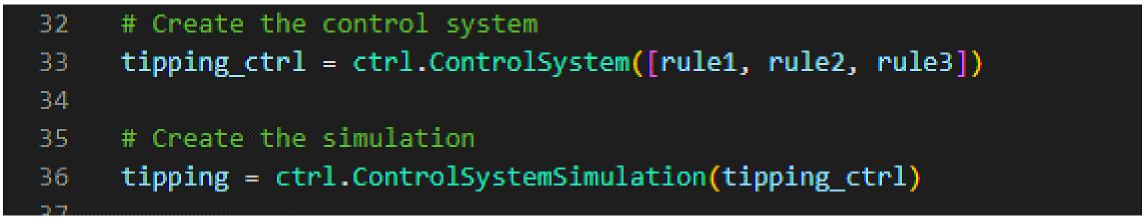
- Define the Fuzzy rules for the output variable :



```
rule1 = ctrl.Rule(food['poor'] | service['poor'], tip['low'])
rule2 = ctrl.Rule(service['average'], tip['medium'])
rule3 = ctrl.Rule(service['good'] | food['good'], tip['high'])
```

Each rule is defined using the ctrl.Rule function, which takes two arguments: an antecedent (a logical condition based on the input variables) and a consequent (the output to be produced if the antecedent is true).

- Creates a fuzzy control system and a simulation of that control system :



```
# Create the control system
tipping_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])

# Create the simulation
tipping = ctrl.ControlSystemSimulation(tipping_ctrl)
```

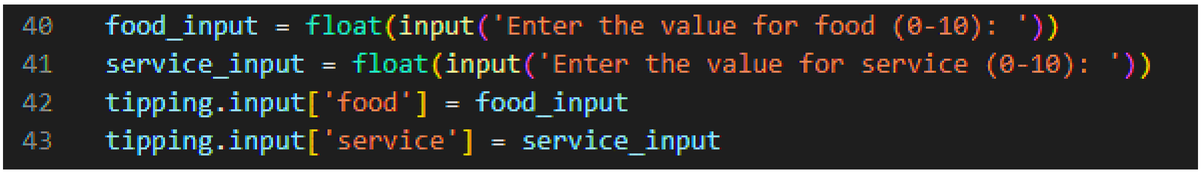
The ctrl.ControlSystem function is used to create a control system object from a list of rules. The control system object is responsible for storing and organizing the rules, and for applying them to determine the output of the system.

The ctrl.ControlSystemSimulation function is used to create a simulation object from a control system. The simulation object can be used to evaluate the performance of the control system under different input conditions.

- Set the input values :

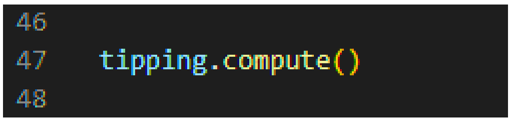
The input function is used to display a prompt to the user and read a value from the user's input. The values are converted to floats using the float function.

The values entered by the user are then assigned to the food and service variables of the simulation object using the input attribute.



```python
food_input = float(input('Enter the value for food (0-10): '))
service_input = float(input('Enter the value for service (0-10): '))
tipping.input['food'] = food_input
tipping.input['service'] = service_input
```

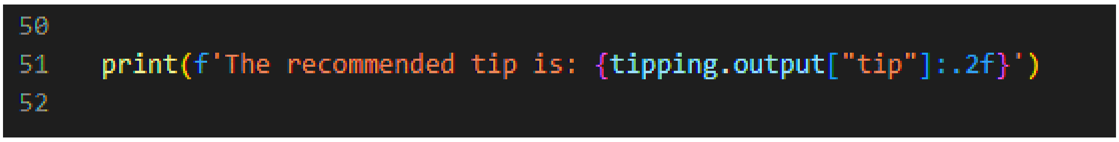
- Calculate the output :



```python

tipping.compute()

```

The compute method of the simulation object applies the rules of the control system to determine the output of the system based on the current values of the input variables.

- Print the Output :



```python

print(f'The recommended tip is: {tipping.output["tip"]:.2f}')

```

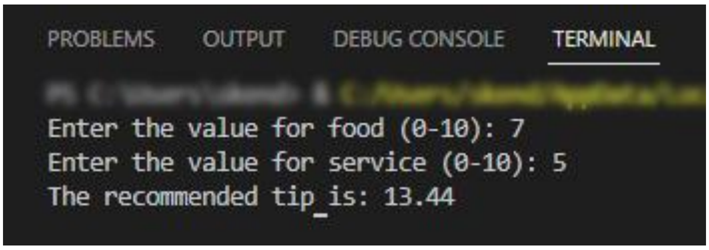
This line of code prints the output of the simulation, which is the value of the tip variable determined by the control system.

The value of the tip variable is formatted as a floating point number with two decimal places using the .2f format specifier, and is included in a string using string interpolation (the f prefix before the string).

The f character indicates that the value being formatted is a floating point number.

The .2 specifies that the value should be rounded to two decimal places.

- Simulation :



```
[redacted]
Enter the value for food (0-10): 7
Enter the value for service (0-10): 5
The recommended tip is: 13.44
```

Conclusion

the fuzzy logic tip calculator project demonstrates the power and flexibility of fuzzy logic and Python for building intelligent and adaptable control systems and solving complex problems. Whether used for research, education, or practical applications, fuzzy logic and Python offer a valuable toolkit for tackling challenging problems and making better decisions.