JOURNAL OF PHYSICS AND TECHNOLOGY
Volume 6 (2025) Number 1, pp. 15-21

DESIGN OF A DECISION-MAKING SYSTEM WITH FUZZY LOGIC FOR THE SELECTION OF CANDIDATES FOR A GIVEN JOB POSITION

Krasimir Slavyanov and Elena Karadzhova

"Vasil Levski" National Military University, 1 "Karel Shkorpiul" str., Shumen, Bulgaria
Corresponding authors: koslavyanov@aadcf.nvu.bg

Keywords:
fuzzy logic
fuzzy rules
membership function
job position

Submitted on: 31.12.2024

Accepted on: 01.08.2025

Abstract:

With the introduction of artificial intelligence to automate more and more processes in the economy, in this report a variant of applying fuzzy logic in the selection of candidates for a given position is proposed. The designed fuzzy inferential decision-making system was simulated using the Fuzzy Logic tool of MATLAB. After a certain number of simulation experiments, it has been statistically proven that the proposed algorithm of the system allows for justified and adequate decisions to be made in the selection of candidates.

1. INTRODUCTION

The use of MATLAB for creating and testing fuzzy inference systems has been studied. MATLAB provides a variety of tools for modeling with fuzzy logic and allows easy integration into real-world applications [1, 2]. The special thing about this report is that it emphasizes the importance of evaluating and visualizing fuzzy systems, as well as the potential for future research and development in the field. A specific application of a fuzzy inference system has been studied. The created program code represents a fuzzy inference system that can be applied in various fields with the necessary modify-cation. The applications of fuzzy inference systems are in very diverse fields such as traffic and transportation systems management, energy systems management, climate systems, financial analysis and investments. In predicting success in education - the system can be applied in education to predict student success depending on their skills and experience. In the lending process and risk analysis - in the banking industry this system can be used for analyzing credit risks, setting credit limits and other similar tasks. In robotics and automation - fuzzy inference systems are used in robotics to make decisions about the movement and navigation of robots. In the energy industry, fuzzy inference systems can be applied to optimize energy consumption and manage distribution. For healthcare and diagnostics - in medicine, they can be used to diagnose diseases and make decisions about appropriate treatment.

These are some of the many areas in which fuzzy inference systems can be successfully applied. They can be used where there is uncertainty in the data like in aerial vehicle management [3] and flight stability improvement [4], and it is necessary to make decisions based on this type of information. Recognition of partly visible objects in images [5] can also be improved with fuzzy logical approaches like in ISAR systems [6]. Some possible application of this approaches can be done in The IT sector is very well developing management with Artificial Intelligence too [7].

The specific problem studied is the selection of candidates for jobs or study programs also well examined in several researches [8-10].

The proposed approach aims to evaluate candidates based on their education, experience, language skills and other criteria. This can be used in human resources departments to support the personnel selection process. Here are the main stages in designing a fuzzy logic inference system of Mamdani type [11] in MATLAB will be described.

2. STARTING FUZZY LOGIC DESIGNER

Creating a fuzzy set: To create a fuzzy set, the fuzzy.set function is used. This function accepts a name for the set and the basic shapes for the membership function specified by the creator (e.g. traingular, trapezoidal, Gaussian, etc.).

For this case, the application for decision-making based on four input parameters was studied:

1. Education: Reflects the level of education of the candidate.

2. Experience: Measures the number of years of experience in the relevant field.

3. Programming Languages: Assesses the number and complexity of programming languages with which the candidate works.

4. Recommendations: Includes the quality of recommendations provided by previous employers.

The output variables for the system are:

- Sysadmin: Probability that the candidate is suitable for a system administrator.
- Soft_Developer: Probability that the candidate is suitable for a software developer.

3. CRITERIA WITH THEIR BASIC SETTINGS

3. 1. EDUCATION

Criterion for the level of education of the candidate and its focus.

Four membership functions are seen, representting different levels of education:

- Higher_IT: Represents higher education in the field of information technology.
- Higher_INF: Indicates higher education in the field of informatics.
- High_Sc_IT: Secondary education with a focus on information technology.
- Not_Applicable: Lack of applicable education.

The range of the variable is [0, 4], with each of the functions having specified parameters for the shape of the curve that forms the membership function for each of the fuzzy sets using a combination of two Gaussian membership functions (**Fig. 1**).

1. Higher_IT
2. Higher_INF
3. High_Sc_IT
4. Not_Applicable

Fuzzy intervals are ranges of values used in fuzzy logic to describe the membership of a value to a certain category. They are defined by Membership Functions, which describe to what extent a given input or output variable belongs to a particular linguistic label (e.g. “Excellent”, “Good”, “Not_Good”).

The main idea is that instead of strictly defined boundaries (as in classical logic), fuzzy intervals allow a value to partially belong to more than one category. This is the basis of the flexibility of fuzzy logic systems.

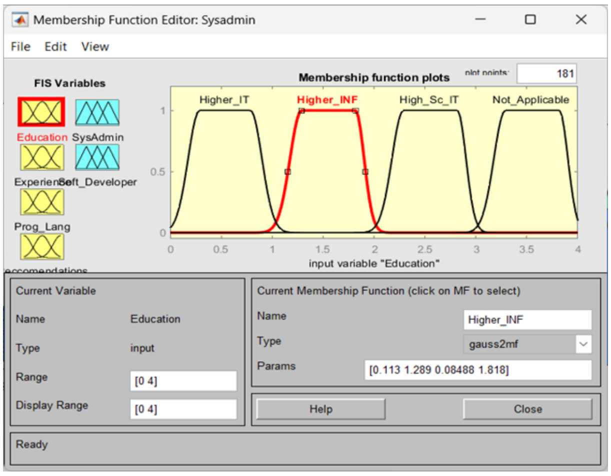

Fig. 1 Membership functions for Education.

3. 2. EXPERIENCE

A criterion for assessing the level of experience of the candidate.

There are four membership functions on the graph that define different levels of professional experience:

- Sys_Admin: Specific experience in systems administration.
- Sys_Sec_Dev: Experience in developing and administering systems.
- Sys_ADMIN: Again, experience in systems administration, but with a different degree of parameterization.
- No_Experience: Lack of professional experience.

The range of the criterion is [0, 5], and the specified parameters are for a triangular form of a function (**Fig. 2**).

Volume 6 (2025) Number 1, pp. 15-21

3. 3. PROG_LANG (PROGRAMMING LANGUAGE)

A criterion for the candidate's knowledge of different programming languages and technologies.

The graph shows four membership functions, each of which denotes knowledge in a different programming language:

- Python_C#_JAVA: Knowledge in Python, C# or Java.
- C++_Ruby_Deploy: Knowledge in C++, Ruby or DevOps Deployment.
- Assembler: Knowledge in low-level programming languages such as Assembler.
- NO_Language: No knowledge in programming languages.

The range of the variable is [0, 4], with each function defined with specific parameters is shown on **Fig. 3**.

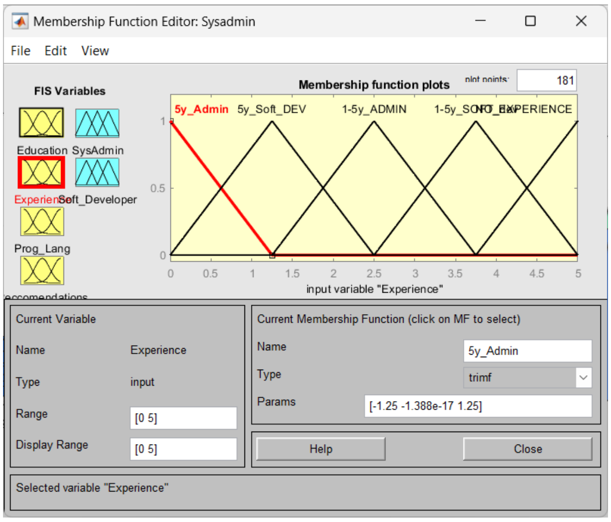

Fig. 2 Membership functions for Experience.

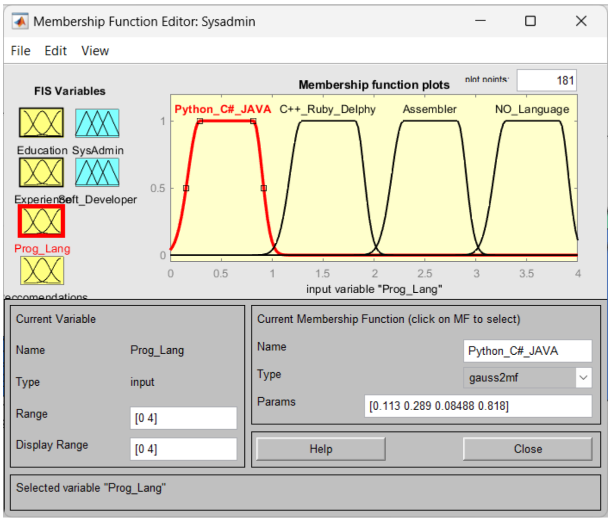

Fig. 3 Membership functions for Programming Language

3. 4. RECOMMENDATIONS

A criterion for the quality of the recommendations provided for the candidate, which is an indicator of his reputation and past achievements.

Three membership functions have been created:

- Excellent: Excellent recommendations.
- Good: Good recommendations.
- Not_Good: Bad recommendations.

The range of the criterion is [0, 9], with the functions being linear and their parameters being different (**Fig. 4**)

Each of the pictures visualizes the process of setting membership functions for different criteria: education, experience, programming knowledge and recommendations. These criteria are an important part of the fuzzy logic system and determine how the input data will be processed and classified.

The graphs help to clarify and visualize the boundaries between the different categories.

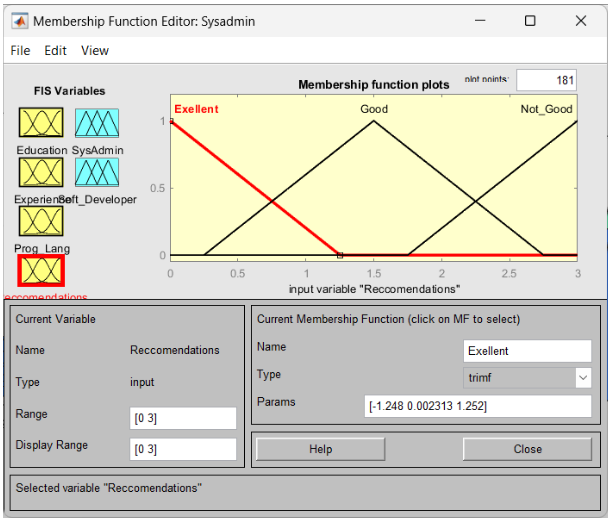

Fig. 4 Membership functions for Recommendations.

4. SETTING RULES FOR MAKING DECISIONS WITH FUZZY LOGIC

Fuzzy Logic decision-making is based on the combination of predefined rules and membership functions that process the input data and determine the output. The main goal of setting rules is to define the logical relationships between different input criteria and the desired result. These rules reflect expert knowledge and allow working with uncertainty and intermediate values.

Components of setting rules

1. Input criteria:

Each criterion or input variable (for example, education, experience, programming knowledge and recommendations) is divided into categories using membership functions. They reflect different degrees of compliance, such as “higher education”, “secondary education”, “lack of experience”, etc.

2. Logical rules:

Rules are built in an “IF-THEN” style, combining different input criteria:

- Example: “If education is Higher_IT AND experience is Sys_Admin, then the recommendation is Excellent.”

These rules can be constructed using operators such as AND, OR, and NOT, which govern logical relationships.

3. Output variables:

The output of the system is the result of the combination of the rules, for example, classifying the candidate as “Excellent”, “Good”, or “Not good”.

4. Defuzzification:

After applying the rules, the system performs defuzzification to convert the fuzzy output values into a precise (defined) value. This allows for a specific decision to be made and the candidates to be ranked.

Rule Setting Process

1. Identifying the Relationships Between Criteria and Outcome:

By analyzing existing data or expert experience, the logical relationships between inputs and desired outputs are determined.

2. Defining specific rules:

Formulations such as:

- “If the candidate is proficient in Python_C#_JAVA and has Excellent recommendations, then the score is Excellent.”
- “If the experience is No_Experience and lacks programming skills, then the score is Poor.”

3. Verification and Tuning:

The rules are tested with sample data to ensure that the system makes logical and correct decisions.

5. CREATING A SYSTEM OF LOGICAL RULES

- “If education is Higher_IT AND experience is Sys_Sec_Dev, then the candidate is Good.”
- “If no programming knowledge AND recommendations are Not_Good, then the result is Not Good.”
- “If recommendations are Excellent AND programming skills are Python_C#_JAVA, then the result is Excellent.”

Rules in fuzzy logic provide structured and transparent decision-making based on expert knowledge and logic. This approach allows for the evaluation of complex cases that cannot be solved by traditional methods with fixed boundaries. Through properly defined input criteria, logical rules and outputs, the system guarantees flexibility and efficiency in decision-making (**Fig. 5**).

An example of a rule in the system could be:

An example of a rule in a fuzzy logic decision-making system:

If:

- The candidate's education is Higher_IT
- The candidate's experience is Sys_Admin
- The candidate is proficient in programming languages Python_C#_JAVA
- The candidate's recommendations are Excellent

Then:

- The result is Excellent

JOURNAL OF PHYSICS AND TECHNOLOGY
Volume 6 (2025) Number 1, pp. 15-21

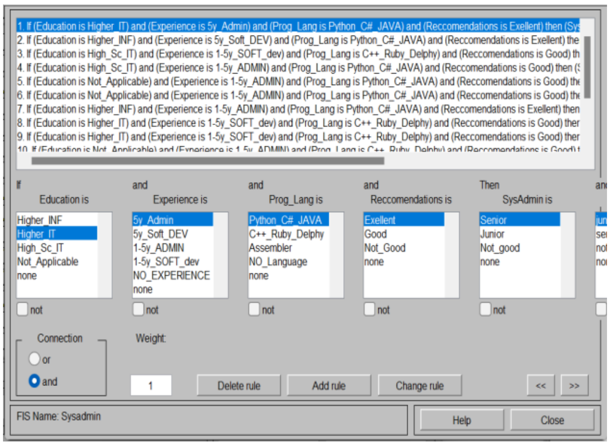

Fig. 5 Fuzzy rules creation.

This logic combines the four main criteria and uses fuzzy logic to interpret their values in the system.

Under these conditions, the selected rules are 15 and some of them are designed as follows:

Rule 1:

If the education is Higher_IT, the experience is 5 years as an administrator, the programming languages used are Python, C#, or JAVA, and the recommendations are excellent, then the SysAdmin is classified as Senior, and the Soft_Developer is classified as Junior. This rule reflects a combination of high qualifications and significant experience to determine a senior position.

Rule 3:

When the education is High_Sc_IT, the experience is 1-5 years in software development, the languages used are C++, Ruby, or Delphi, and the recommendations are excellent, then the SysAdmin is classified as Junior, and the Soft_Developer is classified as Senior. This emphasizes the importance of specific programming languages for growth in development.

Rule 9:

When the education is Not_Applicable, the experience is 1-5 years in software development, knowledge of C++ and Ruby, and good references, SysAdmin and Soft_Developer are rated as Not_good. This emphasizes the lack of significant education and experience for key roles.

Rule 11:

If the education is Higher_INF, the experience is 1-5 years in software development, languages such as C++, Ruby or Delphi are used, and the recommendations are good, then the SysAdmin is classified as Not_good, and the Soft_Developer is classified as Junior. This suggests that although the education and recommendations are present, insufficient experience limits the opportunities for advancement.

Rule 15:

If the education is Not_Applicable, the experience is 5 years in software development, the knowledge of C++ and Ruby is good, then SysAdmin is classified as Not_good, and Soft_Developer is classified as Junior. Here, development experience is significant, but the lack of education limits the chances of a higher role.

6. CONNECTING THE CRITERIA AND RULES TO RANKSCORE

This is done by using the fuzzy control system, which includes input variables (criteria), rules and an output variable (Rankscore).

In the fuzzy control system, the input variables are the criteria that describe the characteristics of the candidates.

The rules in the system are logical expressions that connect the input variables with the output variable (Rankscore). Each line in the [Rules] section represents a rule and expresses the conditions under which a certain conclusion must be fulfilled.

The conclusion of the rules determines what value of Rankscore to assign to the candidate when the conditions of the rule are met. These Rankscore values are set as membership functions.

The input values of the variables are evaluated by the system based on the rules, which contain the logic for assessing the candidate's suitability. After executing all the rules, the system calculates the final Rankscore value, which reflects the candidate's suitability for the job position according to the specified criteria.

Thus, linking the criteria and rules to Rankscore allows the system to evaluate candidates based on fuzzy logic and provide final results that reflect the degree of suitability of each candidate for the given job position, depicted on **Fig. 6**.

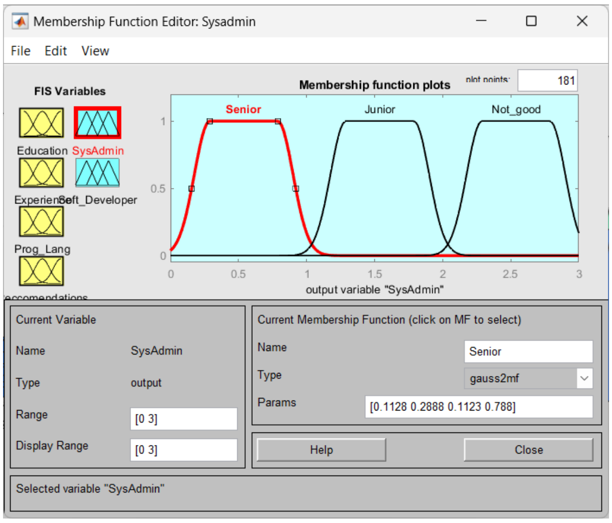

Fig. 6 Output variable System Administrator.

SysAdmin:

1. Senior:

- Type: gauss2mf (Gaussian function with two curves).

2. Junior:

- Type: gauss2mf.
- Parameters: (not visible at the moment, but it is clear that they cover values in the middle part of the range).

3. Not_good:

- Type: gauss2mf.

Soft_Developer (Software Developer)

In Fig. is a picture visualized the output variable Soft_Developer, which evaluates the suitability of the candidate for the position of software developer. The membership function is also divided into three categories:

- Senior_SofD: The candidate is qualified as a senior developer.
- Junior_SofD: The candidate is qualified as a junior developer.
- Not_good_SofD: The candidate is not suitable for the position.

The system uses fuzzy logic to classify candidates into three main categories, with membership function settings specific to each role.

The range of output values [0, 3] allows for smooth transitions between categories.

Membership functions are defined using Gaussian curves (gauss2mf), which provide smooth transitions between categories.

7. VISUALIZATION OF THE RULES FOR THE INPUT VARIABLES.

Main functionalities used with the visualization tools are described:

Visualize rules: Each row in the Rule Viewer represents a single rule that contains conditions and conclusions. Users can see how the conditions are formulated and what the conclusions are for each rule (**Fig. 7**).

Modify rules: Users can edit rules directly in the Rule Viewer and change the logic of the system.

Test input values: The Rule Viewer allows users to enter different input values and see how the output variable (e.g. Rankscore) changes with those values.

Explore rule interactions: Users can analyze how rules interact with each other and how input variables affect the output variable.

Fig. 7 Visualization of the rules for the input variables.

8. CONCLUSIONS

Through simulation experiments with the thus created fuzzy logic decision-making system, its qualities have been empirically proven:

- Precision and adaptability: Evaluates multiple criteria for different positions.
- Transparency: Provides clear evaluation rules.

- Flexibility: Can be adapted to any organization and position.

This approach can transform personnel selection processes, making them more efficient and objective.

REFERENCES

[1] Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning—I. Information Sciences, 8, pp. 199-249.

[2] Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transactions on Systems, Man, and Cybernetics, 3 (1), pp. 28-44.

[3] Genoff, V. M. (2024). Modern technologies with artificial intelligence for unmanned aerial vehicle management. Proceedings of International Scientific Conference - Defense Technologies, DefTech 2024, Shumen, Bulgaria.

[4] Georgieva, V. S., Petrov, N. P. & Kirov, M. K. (2024). Flight stability methods of unmanned aircraft. Proceedings of International Scientific Conference - Defense Technologies, DefTech 2024, Shumen, Bulgaria.

[5] Manov, L. & Kulev, N. (2023). Detection and recognition of partially visible objects in optical images using artificial intelligence techniques - mini review from 1970 to 2000. Proceedings of International Scientific Conference “Defense Technologies”, DefTech 2023, Shumen, Bulgaria, pp. 444-450.

[6] Slavyanov, K. & Minchev, C. (2017). An Algorithm of Fuzzy Inference System for ISAR Image Classification, Environment. Technology. Resources, Rezekne, Latvia, Proceedings of the 11th International Scientific and Practical Conference, II, pp. 154-158.

[7] Vasylkiv, N., Dubchak, L., Sachenko, A., Lendyuk, T. & Sachenko, O. (2020). Fuzzy Logic System for IT Project Management, ICT&ES-2020: Information-Communication Technologies & Embedded Systems, 12 November 2020, Mykolaiv, Ukraine.

[8] Doctor, F., Hagras, H., Roberts, D. & Callaghan, V. (2009). A neuro-fuzzy based agent for group decision support in applicant ranking within human resources systems, Fuzzy Systems, FUZZ-IEEE 2009.

[9] Doctor, F., Hagras, H., Roberts, D. & Callaghan, V. (2008). A type-2 fuzzy based system for handling the uncertainties in group decisions for ranking job applicants within Human Resources systems. Fuzzy Systems, FUZZ-IEEE 2008.

[10] Doctor, F., Hagras, H., Roberts, D. & Callaghan, V. (2009). A fuzzy based agent for group decision support of applicants ranking within recruitment systems, Intelligent Agents, IA ‘09. IEEE Symposium.

[11] Mamdani, E. H. & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies, 7 (1), pp. 1-13.