Course name	Module name	Coordinator	Date
Intelligent Systems and Robotics, MSc	Fuzzy Logic	Dr Sarah Greenfield	17/12/2020
Title	Practical Assignment - FIS		
Author	Lukasz Pszonak (P2619730)		
Type	University report		

ABSTRACT

The report concerns the task that is specified in The Fuzzy Logic Module Content's handout which is the implementation of Fuzzy Inference System using MATLAB. The work contains many sections and is made according to a pattern that meets all University standards. The first part of the report is called Introduction, where the author explains what one understands by Fuzzy Logic, Fuzzy Logic Sets, Membership Functions, a Fuzzy Inference System and how is it helpful in resolving many problems. The author also describes the problem for which he creates the Fuzzy Inference System. Next section – Overview of the System focuses on presenting the GUI and FIS. In this part of the report, the author focuses on fine tuning, discusses other membership functions that could be used and defuzzification methods. In System Evaluation the author explains the chosen methodology for a working system that pleases the author and could be used as a support for volleyball coaches before making the decision of signing a player for their team. Reference section lists down all used sources that have an annotation in the report. Final part – Appendices contains screenshots, that show various defuzzification methods, different aggregation approaches and their impact on the system.

TABLE OF CONTENTS

INTRODUCTION

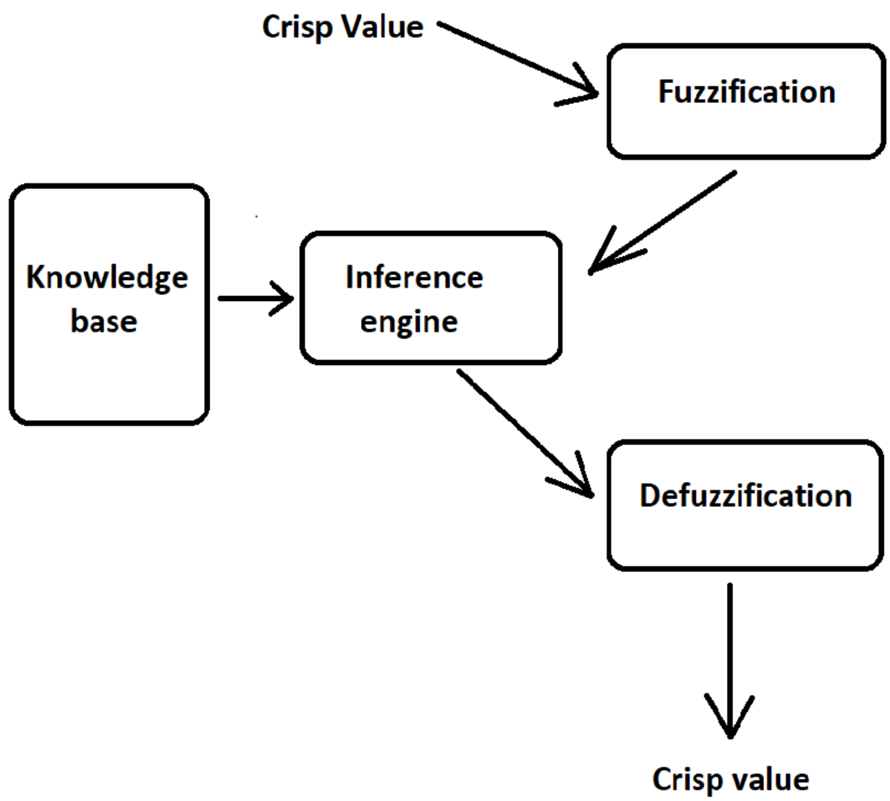
Fuzzy Logic is a concept that intends to model human logical reasoning using imprecise and/or vague statements. Instead of using "truth and false" or "yes and no" it uses a component known as the degree of membership (degree of truth)[1]. The degree of membership is a numerical value from 0 to 1 which is a degree to which an element matches or belongs to a certain fuzzy set. By a fuzzy set one understands a class of objects with a continuum of degrees of membership. Such a set has a membership function which assigns a degree of membership to each object in the set[2]. Fuzzy Inference System or FIS works as a way of mapping from input to output with the usage of fuzzy logic. It attempts to formalize the reasoning process of human language by creating fuzzy "if-then" rules. A general structure of FIS is displayed on Picture 1. System inputs are crisp numbers, after applying a fuzzification function they are transformed into fuzzy sets and passed on to the inference engine which can be described as a module that simulates the human reasoning process, it makes fuzzy inference on the inputs and "if-then" that were passed on from the Knowledge Base. Lastly, the defuzzification module takes the obtained fuzzy set and transforms it into a crisp value[3].

---

[1] P. Cintula, R. Horčík, C. Noguera. *The quest for the basic fuzzy logic*. In Franco Montagna, editor, *Petr Hájek on Mathematical Fuzzy Logic*, volume 6 of Outstanding Contributions to Logic, pp. 245–290, Springer 2015

[2] L. A. Zadeh, *Fuzzy Sets**, Department of Electrical Engineering and Electronics Research Laboratory, University of California, Berkeley, California, 1965.

[3] http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/fuzzy_inferencia/introfis_en.htm#:~:text=Practice%3A%20FIS%20Creation-,FIS%20INTRODUCTION,fuzzy%20IF%2DTHEN%20rules, date and time of access: 16.12.2020 | 19:30.



Picture 1. FIS structure.

Fuzzy Inference Systems are used to help solve decision problems. Any mathematical concepts within fuzzy reasoning are relatively easy. FIS is flexible so there is no need to create new ones from the beginning, all it takes is adding, deleting rules or changing defuzzification methods. It does not provide a “yes or no” answer. It should be considered as a guide in one’s decision making. For example, a system created to tell how heavy someone is, does not give “A person is heavy” as an answer. It can say “This person is heavy to the degree of 0.7” as a substitute to “This person weighs 100 kg”. It shows that fuzzy logic is not used to solve new problems but to use new and unique methods to solve everyday problems[4].

The question that the author needs answered is “How good of a volleyball player one can be”. Good tuning of the system could help volleyball coaches in making the right decisions whether to sign a player for a team or not by acting accordingly to the result of the system’s output.

[4] *Fuzzy Logic*, Stanford Encyclopedia of Philosophy, https://plato.stanford.edu/entries/logic-fuzzy/, date of access: 17.12.2020 | 13:55.

In brief, a high value of the system's output should encourage the coach to consider a player to be a good choice for the team, low output's value should tell him to keep looking and a mediocre output should be taken as "worth considering provided there are no better options". The author decides that the system should have 4 inputs (each with 3 membership functions of the same type) as 4 player skills that should matter for volleyball coaches. These are: agility, reception, attack power and attack accuracy. As agility the author understands player's overall fitness capabilities, speed, jumping power and flexibility. Reception is the ability to make the correct ball deflection that results in a proper pass of the ball to another teammate. Attack power and accuracy are self-explanatory, more power enables the player to break through a block and results in less time needed for the ball to hit the ground, whereas accuracy means a higher chance for the ball to hit in the playing field area[5]. What is obvious to the author is that most volleyball players are tall, so tallness is omitted, assuming that above mentioned traits are only considered for players above a certain height. Table 1 displays inputs, their membership function types and their measure with justification.

Table 1. Inputs

Input number	**Input name**	**Membership functions type**	**Measure**	**Justification**
1	Agility	Triangular x 3 (Low, Mediocre, High)	0 to 100 [%]	This skill should be scored as a percentage from 0 to 100. As 0 the author understands being unable to perform any agile movement or maneuver while 100 means quick and agile movement, quick maneuvers and high jumps.
2	Reception	Trapezoidal x 3 (Poor, Average, Good)	0 to 100 [%]	This skill should be graded as a percentage from 0 to 100 according to

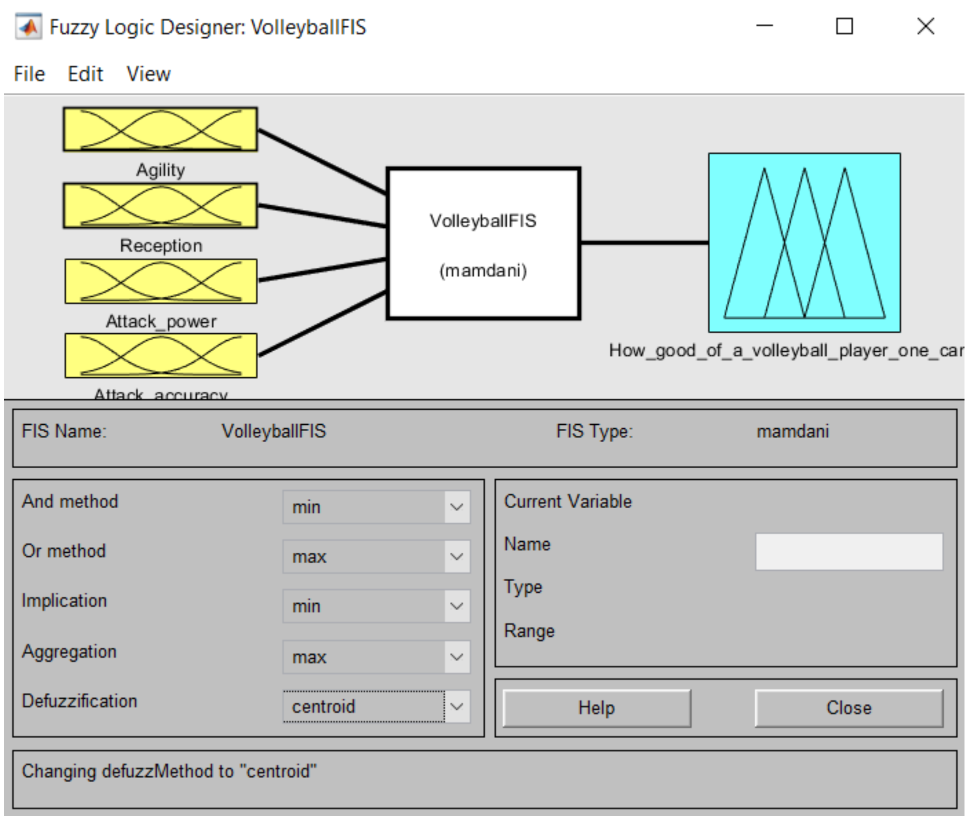
[5] Coleman, J. (1992). *Evalution of blocking*. *Coaching Volleyball,* 15: p. 6 -10.

				common sense and not being able to be assessed on any other scale. The only substitute the author could think of might be a crunchy "yes and no", but this is against the assumptions of the system. As 0 the author understands being unable to properly deflect any incoming balls while 100 means receiving every single one correctly.
3	Attack power	Triangular x 3 (Weak, Decent, Powerful)	0 to 100 [%]	This skill should be graded as a percentage from 0 to 100 according to common sense. The scale for it could also be an actual kinetic force of the ball in [J] but that could bring discordance to the system. Values from 0 to 100 are a percentage of how much attacks would break through a block.

4	Attack accuracy	Gaussian x 3 (Low, Mediocre, High)	0 to 100 [%]	This skill should be graded as a percentage from 0 to 100 according to common sense. Values from 0 to 100 means the percentage of attacks that hit the playing field on the other side of the net. (There is no block.)

SYSTEM OVERVIEW

Fuzzy logic toolbox or Fuzzy GUI (Graphical User Interface) is a quick and efficient way to create a Fuzzy Inference System. It can be used to quickly add/change input membership functions, add new inputs or change defuzzification methods[6]. Picture 2 presents FIS created by the author for this assignment.



Picture 2. FIS.

[6] The MathWorks, Inc., "*Fuzzy Logic Toolbox 2.2.4*", Sept. 2006.

Above is a Mamdani Fuzzy Inference System, what means that it uses a Mamdani method to obtain the output value. It is used because it is the most intuitive and well-suited for human input with widespread acceptance[7]. Mamdani contains of 4 steps[8] which are:

Step 1 – Evaluating the antecedent for each rule (Obtaining inputs' membership values - "input fuzzification". If the antecedent of the rule has more than one part, a fuzzy operator (t-norm or t-conorm) is applied to obtain a single membership value.)

Step 2 – Obtaining each rule's conclusion (Applying a fuzzy implication operator to obtain a new fuzzy set.)

Step 3 – Aggregating conclusions (Combining the outputs obtained for each rule into one fuzzy set with the use of aggregation operator – max.)

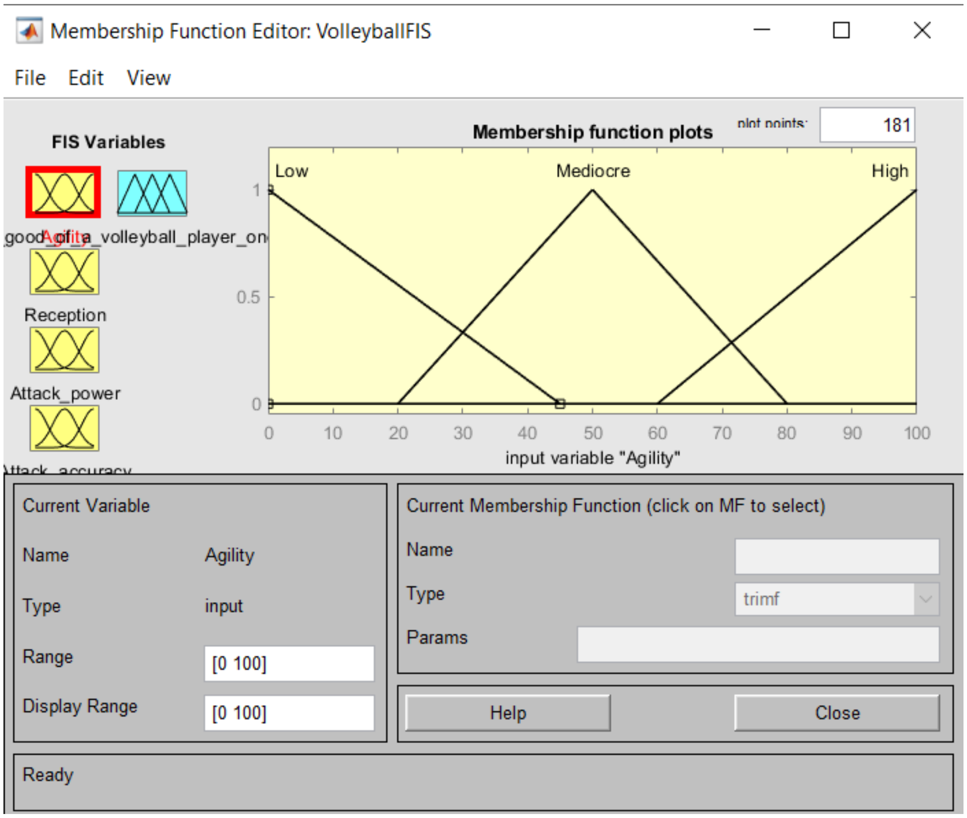
Step 4 – Defuzzification (In order for the output to be a number instead of a fuzzy set it must be transformed using one of defuzzification methods[9]. The centroid method proves to be the best choice for this system. The author has more experience with this method and understands it the most. It returns the center of the area under the fuzzy set.)

As mentioned, the system contains 4 inputs and each of them has 3 membership functions. The first one is called "Agility" and its membership functions are triangular (Low, Mediocre and High). Input 2 – "Reception" has trapezoidal membership functions (Poor, Average, Good). Input 3 called "Attack power" has triangular membership functions (Weak, Decent, Powerful) and lastly, input 4 - "Attack accuracy" has gaussian membership functions (Low, Mediocre, High). Different membership function types were implemented for different inputs to ensure diversity and to add complexity to the system. Further explanation provided below.

[7] https://www.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html, date of access: 16.12.2020 | 21.19

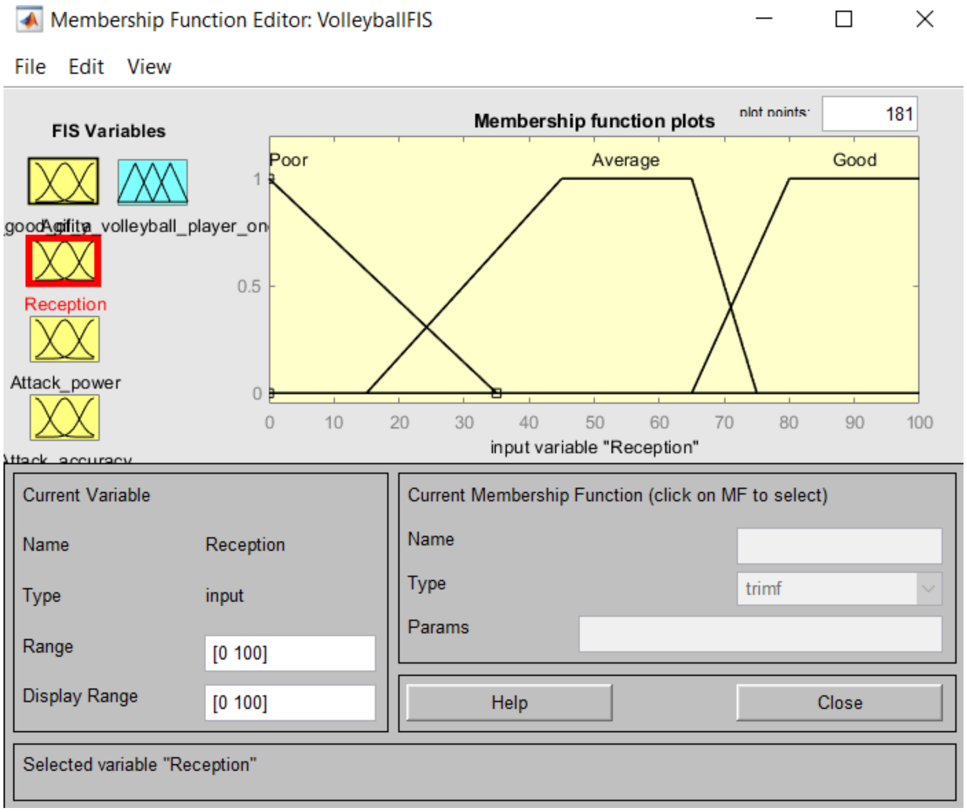
[8] http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/fuzzy_inferencia/mamdani_en.htm, date of access: 16.12.2020 | 21:25

[9] H. J. Zimmerman, *Fuzzy Set Theory and its* Applications, Kluwer Academic Publishers, 1991.



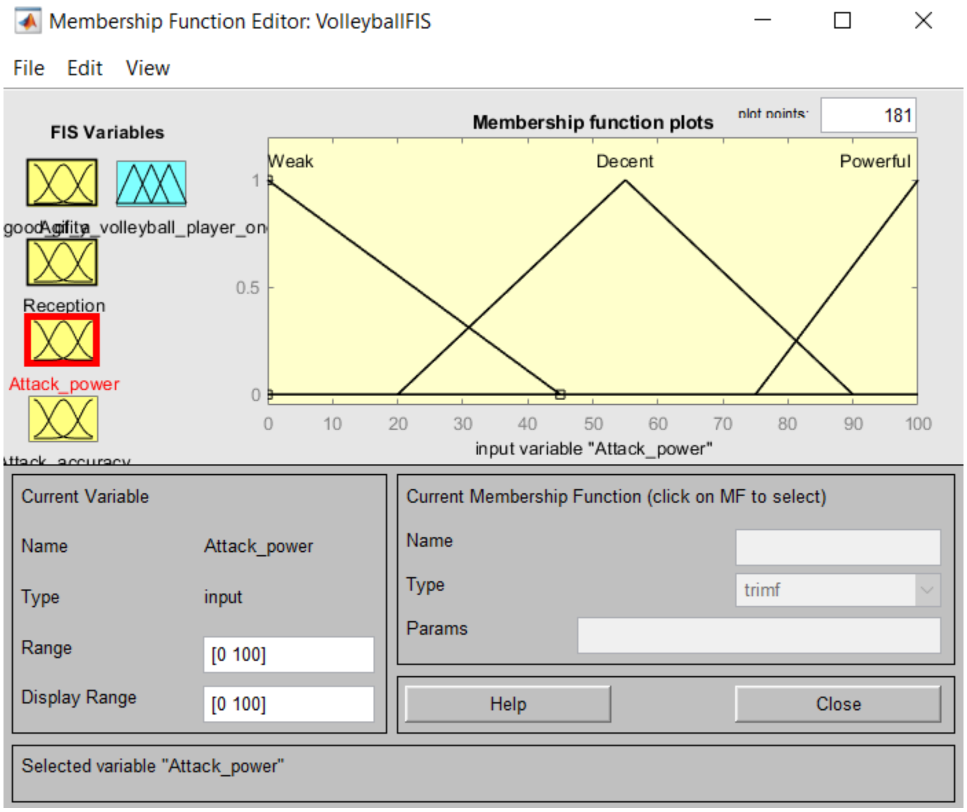
Picture 3. Input 1 – Agility.

The author chose triangular type for membership functions of Input 1 because it enables displaying linear changes which best present his idea of how this skill fluctuates.



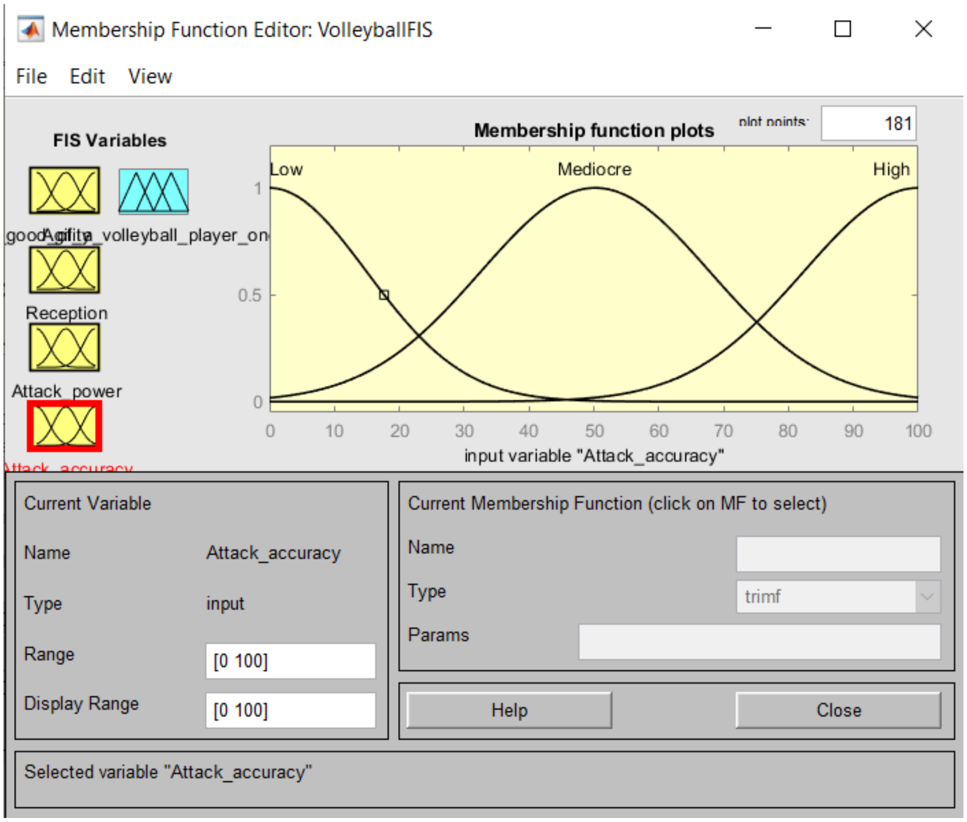
Picture 4. Input 2 – Reception.

The author chose trapezoidal type for membership functions of Input 2 because it simplifies the fact that a broad spectrum of values should be considered “Average” and another wide area of values should be considered “Good”. This way the author’s idea for presenting changes in this skill is met.



Picture 5. Input 3 – Attack power.

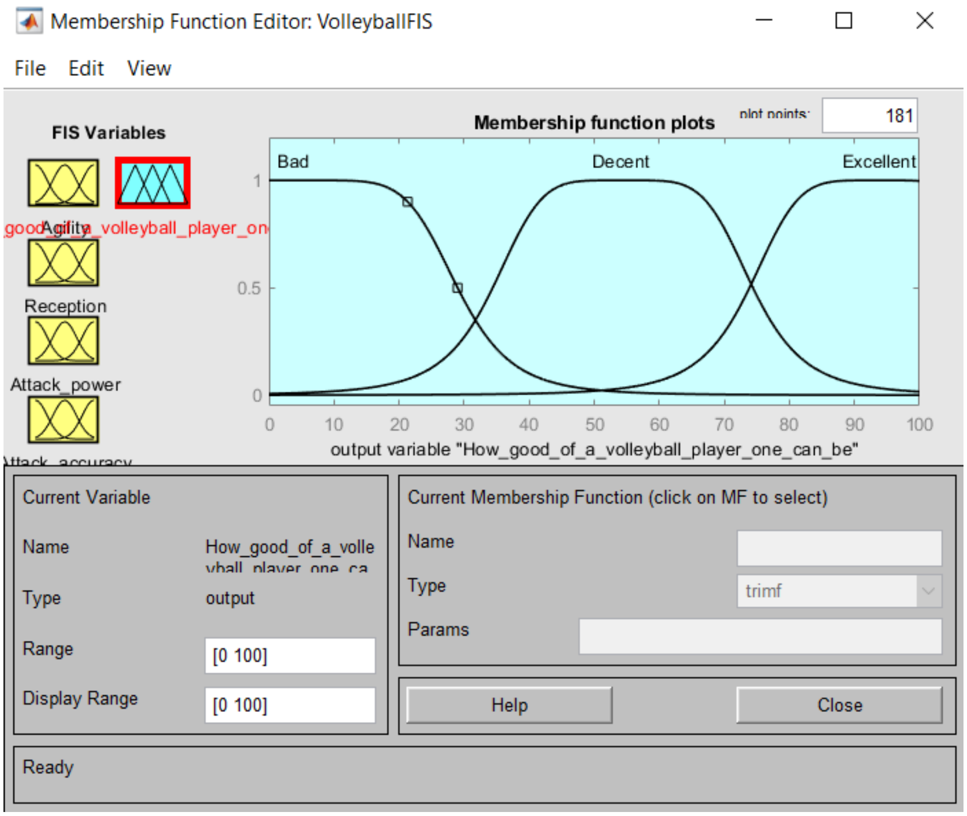
The author chose triangular type for membership functions of Input 3 because it enables displaying linear changes which best present his idea of how this skill fluctuates.



Picture 6. Input 4 – Attack accuracy.

The author chose gaussian type for membership functions of Input 4, because it enables displaying non-linear changes which best present changes in one's attack accuracy, because, according to the author's experience in volleyball, this skill could never fluctuate in a linear manner.

The output of the system is the question: "How good of a volleyball player one can be?" and its value provides the answer. It contains 3 membership functions, each of a generalized bell-shape type (Bad, Decent, Excellent). Using this type of membership function allows to generate a believable look for the output of the system. In other words, it is reasonable for the output of the system, where at least one input has non-linear changes, to have values fluctuating in a non-linear manner. Using this type of membership functions also enables to give a broad spectrum of small values the degree of membership equal to 1 as "Bad", a large spectrum of values the degree of membership equal to 1 as "Decent" and it is also sensible to have a wide area of values with the degree of membership equal to 1 as "Excellent" because as a coach one does not look for perfect players (that task would be impossible). A result above a certain point would be satisfactory and just, considering all the rules included in the system. The output and its membership functions are visible on Picture 7.



Picture 7. The output of the system.

SYSTEM EVALUATION

To evaluate the system, one must put himself in a volleyball coach's role. A FIS is introduced to help make fair decisions about players[10]. Many consider themselves a good fit for the team, however it is the coach who is responsible for organizing a group of skilled athletes. Putting teamwork aside in the elimination stage the coach must investigate the individual skills of a contender and be able to measure them in some sort of scale[11]. A scale from 0 to 100 % explained in the previous section of the report seem adequate for it. Same scales. The author starts with developing rules for every possible combination – 4 inputs, each with 3 membership functions, give 4^3 = 64 rules, however it is noticed that some of them are unnecessary because they stand on the contrary to common sense, and it is a good practice to have a relatively small number of rules, therefore created Fuzzy Inference System is limited to 42 rules[12]. It enables complex consideration of various combinations of the traits and appears to be reasonable[13]. Because it is difficult to determine whether one particular trait dominates over the other, the author decides to give each rule the same weight[14]. The list of rules is displayed below on Table 2. Surface views that consider 2 out of 4 inputs for various defuzzification methods are visible in the Appendices section of the report. They provide some extra data and a graphical representation that could be used by a coach in the decision making.

[10] https://www.efdeportes.com/efd83/fuzzy.htm, date of access: 17.12.2020 | 23:27.

[11] Marques, Mário C; Tillaar, Roland van den[2,3]; Gabbett, Tim J; Reis, Victor M; González-Badillo, Juan J *Physical Fitness Qualities of Professional Volleyball Players: Determination of Positional Differences*, Journal of Strength and Conditioning Research: July 2009 - Volume 23 - Issue 4 - p 1106-1111 doi: 10.1519/JSC.0b013e31819b78c4

[12] Zadeh, L. A. (1992). *The calculus of fuzzy if-then rules*. *AI Expert,* ch. 7: p. 23 - 27.

[13] Jerry M. Mendel, *Introduction to Rule-Based Fuzzy Logic Systems*, University of Southern California, p. 1-23.

[14] J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, NJ: Prentice-Hall, 2001.

Table 2. List of rules.

Rule's number	Rule's name	Rule's weight
1	If Agility is (low) and Reception is (poor) and Attack power is (weak) and Attack accuracy is (low), then the player is (bad).	1
2	If Agility is (mediocre) and reception is (poor) and attack power is (weak) and attack accuracy is (low), then the player is (bad).	1
3	If Agility is (high) and Reception is (poor) and Attack power is (weak) and attack accuracy is (low), then the player is (bad).	1
4	If Agility is (low) and Reception is (average) and Attack power is (weak) and attack accuracy is (low), then the player is (bad).	1
5	If Agility is (low) and Reception is (good) and Attack power is (weak) and attack accuracy is (low), then the player is (bad).	1
6	If Agility is (low) and Reception is (poor) and Attack power is (decent) and attack accuracy is (low), then the player is (bad).	1
7	If Agility is (low) and Reception is (poor) and Attack power is (powerful) and attack accuracy is (low), then the player is (bad).	1
8	If Agility is (low) and Reception is (poor) and Attack power is (weak) and attack accuracy is (mediocre), then the player is (bad).	1
9	If Agility is (low) and Reception is (poor) and Attack power is (weak) and attack accuracy is (high), then the player is (bad).	1
10	If Agility is (mediocre) and Reception is (average) and Attack power is (weak) and attack accuracy is (low), then the player is (bad).	1
11	If Agility is (high) and Reception is (average) and Attack power is (weak) and attack accuracy is (low), then the player is (decent).	1
12	If Agility is (mediocre) and Reception is (good) and Attack power is (weak) and attack accuracy is (low), then the player is (decent).	1
13	If Agility is (high) and Reception is (good) and Attack power is (weak) and attack accuracy is (low), then the player is (decent).	1

14	If Agility is (low) and Reception is (good) and Attack power is (decent) and attack accuracy is (low), then the player is (decent).	1
15	If Agility is (low) and Reception is (average) and Attack power is (powerful) and attack accuracy is (low), then the player is (decent).	1
16	If Agility is (low) and Reception is (good) and Attack power is (powerful) and attack accuracy is (low), then the player is (decent).	1
17	If Agility is (low) and Reception is (poor) and Attack power is (decent) and attack accuracy is (mediocre), then the player is (bad).	1
18	If Agility is (low) and Reception is (poor) and Attack power is (powerful) and attack accuracy is (mediocre), then the player is (decent).	1
19	If Agility is (low) and Reception is (poor) and Attack power is (decent) and attack accuracy is (high), then the player is (decent).	1
20	If Agility is (low) and Reception is (poor) and Attack power is (powerful) and attack accuracy is (high), then the player is (decent).	1
21	If Agility is (mediocre) and Reception is (poor) and Attack power is (weak) and attack accuracy is (mediocre), then the player is (bad).	1
22	If Agility is (high) and Reception is (poor) and Attack power is (weak) and attack accuracy is (mediocre), then the player is (decent).	1
23	If Agility is (mediocre) and Reception is (poor) and Attack power is (weak) and attack accuracy is (high), then the player is (decent).	1
24	If Agility is (high) and Reception is (poor) and Attack power is (weak) and attack accuracy is (high), then the player is (decent).	1
25	If Agility is (mediocre) and Reception is (poor) and Attack power is (decent) and attack accuracy is (low), then the player is (decent).	1
26	If Agility is (high) and Reception is (poor) and Attack power is (decent) and attack accuracy is (low), then the player is (decent).	1
27	If Agility is (mediocre) and Reception is (poor) and Attack power is (powerful) and attack accuracy is (low), then the player is (decent).	1

28	If Agility is (high) and Reception is (poor) and Attack power is (powerful) and attack accuracy is (low), then the player is (decent).	1
29	If Agility is (low) and Reception is (average) and Attack power is (weak) and attack accuracy is (mediocre), then the player is (bad).	1
30	If Agility is (low) and Reception is (good) and Attack power is (weak) and attack accuracy is (mediocre), then the player is (decent).	1
31	If Agility is (low) and Reception is (average) and Attack power is (weak) and attack accuracy is (high), then the player is (decent).	1
32	If Agility is (low) and Reception is (good) and Attack power is (weak) and attack accuracy is (high), then the player is (decent).	1
33	If Agility is (mediocre) and Reception is (average) and Attack power is (decent) and attack accuracy is (mediocre), then the player is (decent).	1
34	If Agility is (high) and Reception is (good) and Attack power is (powerful) and attack accuracy is (low), then the player is (decent).	1
35	If Agility is (high) and Reception is (good) and Attack power is (powerful) and attack accuracy is (mediocre), then the player is (decent).	1
36	If Agility is (high) and Reception is (good) and Attack power is (powerful) and attack accuracy is (high), then the player is (excellent).	1
37	If Agility is (high) and Reception is (good) and Attack power is (weak) and attack accuracy is (mediocre), then the player is (decent).	1
38	If Agility is (high) and Reception is (good) and Attack power is (decent) and attack accuracy is (high), then the player is (excellent).	1
39	If Agility is (high) and Reception is (poor) and Attack power is (powerful) and attack accuracy is (high), then the player is (decent).	1
40	If Agility is (high) and Reception is (average) and Attack power is (powerful) and attack accuracy is (high), then the player is (excellent).	1

41	If Agility is (low) and Reception is (good) and Attack power is (powerful) and attack accuracy is (high), then the player is (decent).	1
42	If Agility is (mediocre) and Reception is (good) and Attack power is (powerful) and attack accuracy is (high), then the player is (excellent).	1

It is difficult for the author to determine which defuzzification method is the best for this system, and he resigns from the idea of choosing one basing only on his experience with it. Instead, the author proceeds to evaluate the system even more and decides to take numerous readings from the Rule Viewer section of the GUI for different defuzzification methods and include them in this report as Tables 3-7. The author then tries to find the most suitable system configuration that would be both most readable for him and would provide most satisfactory results, where as "satisfactory" he understands – most useful for the coach.

Table 3. Rule Viewer readings for the centroid defuzzification method:

Agility	Reception	Attack Power	Attack Accuracy	Output Value
25	25	25	25	**35.3**
50	50	50	50	**54.3**
100	50	50	100	**50**
100	90	80	70	**80**
75	75	75	75	**81.6**
100	100	100	100	**85.3**

Table 4. Rule Viewer readings for the bisector defuzzification method:

Agility	Reception	Attack Power	Attack Accuracy	Output Value
25	25	25	25	**29**
50	50	50	50	**54**
100	50	50	100	**50**
100	90	80	70	**81**
75	75	75	75	**83**
100	100	100	100	**86**

Table 5. Rule Viewer readings for the mom defuzzification method:

Agility	Reception	Attack Power	Attack Accuracy	Output Value

25	25	25	25	**16**
50	50	50	50	**54.5**
100	50	50	100	**50**
100	90	80	70	**84.5**
75	75	75	75	**86**
100	100	100	100	**94**

Table 6. Rule Viewer readings for the lom defuzzification method:

Agility	Reception	Attack Power	Attack Accuracy	Output Value
25	25	25	25	**32**
50	50	50	50	**68**
100	50	50	100	**50**
100	90	80	70	**100**
75	75	75	75	**100**
100	100	100	100	**94**

Table 7. Rule Viewer readings for the som defuzzification method:

Agility	Reception	Attack Power	Attack Accuracy	Output Value
25	25	25	25	**0**
50	50	50	50	**41**
100	50	50	100	**50**
100	90	80	70	**69**
75	75	75	75	**72**
100	100	100	100	**94**

In order to use different defuzzification methods, it is necessary to explain each one of them:

a) Centroid method returns the center of gravity of the fuzzy set. It uses the following formula: $x_c = \frac{\sum_i \mu(x_i) x_i}{\sum_i \mu(x_i)}$, where μ(xi) is the membership degree for point x_i.

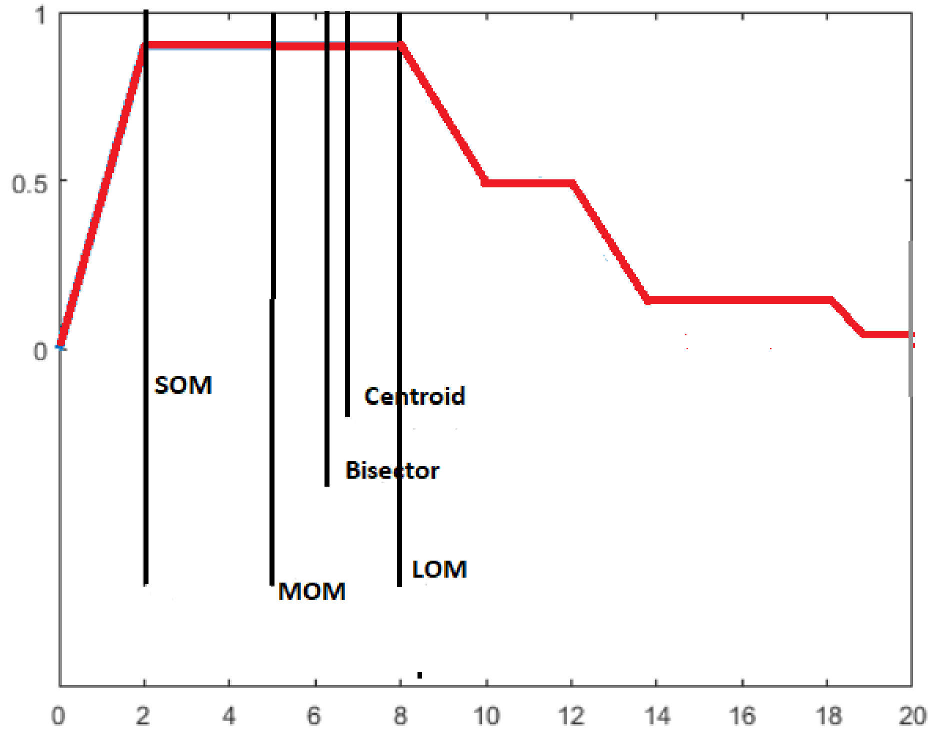
b) Bisector method finds the vertical line that would split the considered fuzzy set into two sub-sets of equal area. It is often, however not always, coincident with the centroid value.

c) MON stands for Middle of Maximum

d) LOM stands for Largest of Maximum

e) SOM stands for Smallest of Maximum[15]

Below, on Picture 11, the differences between mentioned methods are presented on an example.



Picture 8. Differences between defuzzification methods available in MATLAB.

Table 6 presents a direct comparison between output values for different defuzzification methods. It is used to choose the best system configuration. It is worth mentioning that the author decides not to change default aggregation (max), implication (min), "or" method (max) and "and" method (min) in order not to introduce too many complications to the system and because he believes the default configuration of mentioned FIS parameters suits his goal for the system, which is to provide as much sensible data as possible for the coach, without any confusion.

[15] https://www.mathworks.com/help/fuzzy/defuzzification-methods.html, date of access: 17.12.2020 | 23:42.

Table 6. Output comparison for different defuzzification methods:

Agility	Reception	Attack Power	Attack Accuracy	Centroid	Bisector	MON	LOM	SOM
25	25	25	25	**35.3**	**29**	**16**	**32**	**0**
50	50	50	50	**54.3**	**54**	**54.5**	**68**	**41**
100	50	50	100	**50**	**50**	**50**	**50**	**50**
100	90	80	70	**80**	**81**	**84.5**	**100**	**69**
75	75	75	75	**81.6**	**83**	**86**	**100**	**72**
100	100	100	100	**85.3**	**86**	**94**	**94**	**94**

Where:

Green colour – data compliance between different methods,

Red colour - data would not be useful for the coach, value is not sensible,

Orange colour– doubtful data usefulness for the coach, it is difficult to determine whether the value is rational,

Blue colour – data would be useful for the coach, value is sensible.

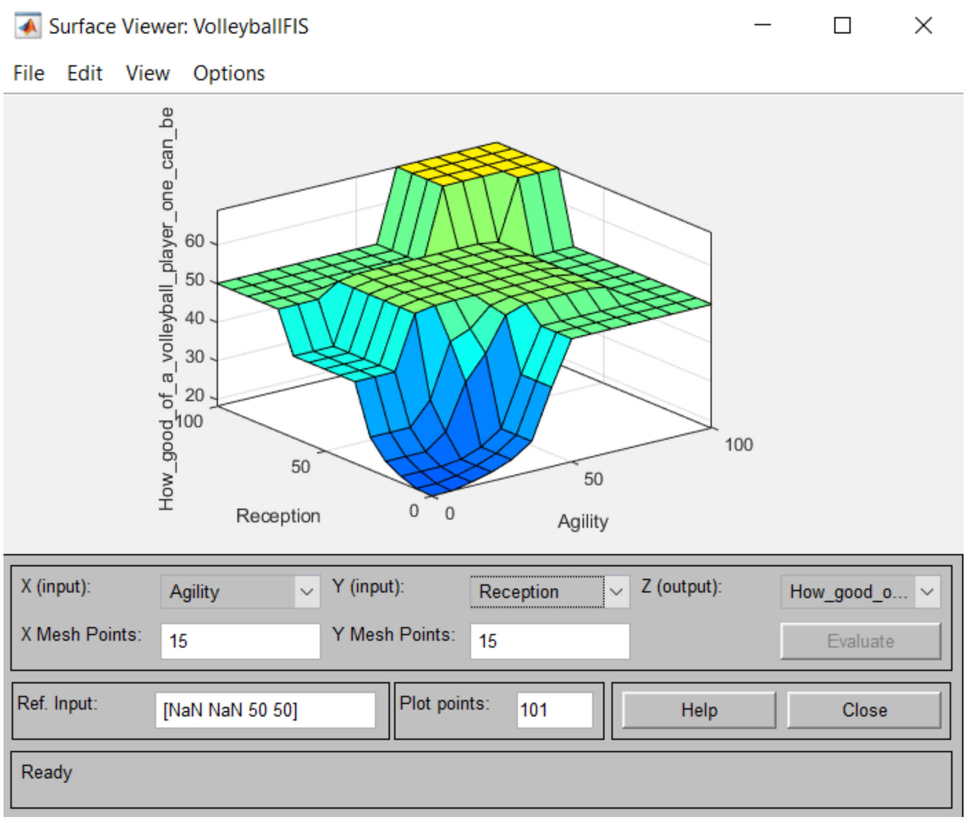
According to the above readings with support of Surface view from the Appendices section, the author decides that the best configuration for the system is to use the bisector defuzzification methods because it provides the most reasonable output values which could be used by the coach in order to make a just decision regarding a player.

REFERENCES

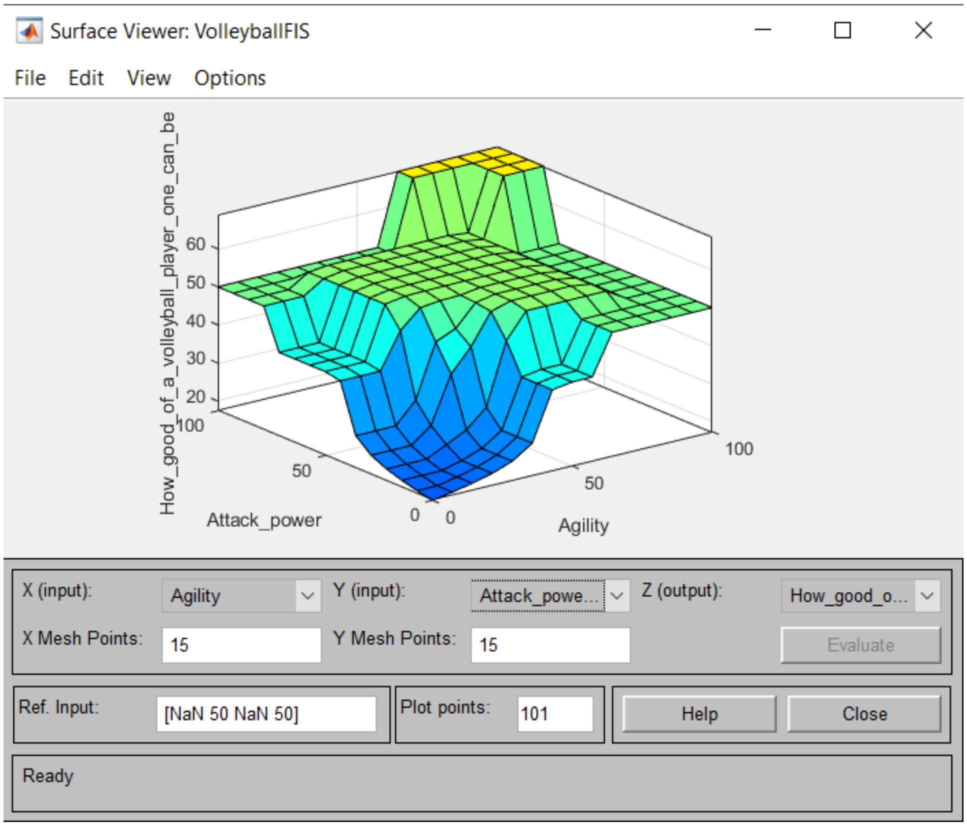
1. P. Cintula, R. Horčík, C. Noguera. *The quest for the basic fuzzy logic*. In Franco Montagna, editor, *Petr Hájek on Mathematical Fuzzy Logic*, volume 6 of Outstanding Contributions to Logic, pp. 245–290, Springer 2015
2. L. A. Zadeh, *Fuzzy Sets**, Department of Electrical Engineering and Electronics Research Laboratory, University of California, Berkeley, California, 1965.
3. http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/fuzzy_inferencia/introfis_en.htm#:~:text=Practice%3A%20FIS%20Creation-,FIS%20INTRODUCTION,fuzzy%20IF%2DTHEN%20rules, date and time of access: 16.12.2020 | 19:30.
4. *Fuzzy Logic*, Stanford Encyclopedia of Philosophy, https://plato.stanford.edu/entries/logic-fuzzy/, date of access: 17.12.2020 | 13:55.
5. Coleman, J. (1992). *Evalution of blocking*. *Coaching Volleyball,* 15: p. 6 -10.
6. The MathWorks, Inc., "*Fuzzy Logic Toolbox 2.2.4*", Sept. 2006.
7. https://www.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html, date of access: 16.12.2020 | 21.19
8. http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/fuzzy_inferencia/mamdani_en.htm, date of access: 16.12.2020 | 21:25
9. H. J. Zimmerman, *Fuzzy Set Theory and its Applications*, Kluwer Academic Publishers, 1991.
10. https://www.efdeportes.com/efd83/fuzzy.htm, date of access: 17.12.2020 | 23:27.
11. Marques, Mário C; Tillaar, Roland van den2,3; Gabbett, Tim J; Reis, Victor M; González-Badillo, Juan J *Physical Fitness Qualities of Professional Volleyball Players: Determination of Positional Differences*, Journal of Strength and Conditioning Research: July 2009 - Volume 23 - Issue 4 - p 1106-1111 doi: 10.1519/JSC.0b013e31819b78c4
12. Zadeh, L. A. (1992). *The calculus of fuzzy if-then rules. AI Expert*, ch. 7: p. 23 - 27.
13. Jerry M. Mendel, *Introduction to Rule-Based Fuzzy Logic Systems*, University of Southern California, p. 1-23.
14. J. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, NJ: Prentice-Hall, 2001.
15. https://www.mathworks.com/help/fuzzy/defuzzification-methods.html, date of access: 17.12.2020 | 23:42.

APPENDICES

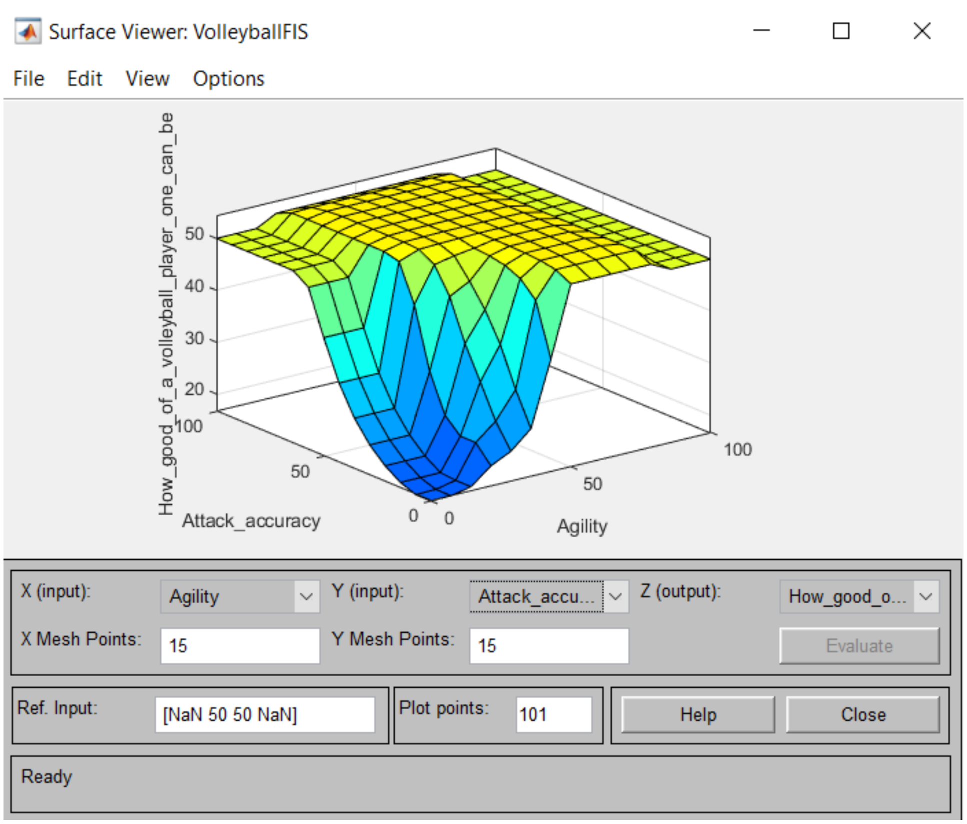
Centroid method:



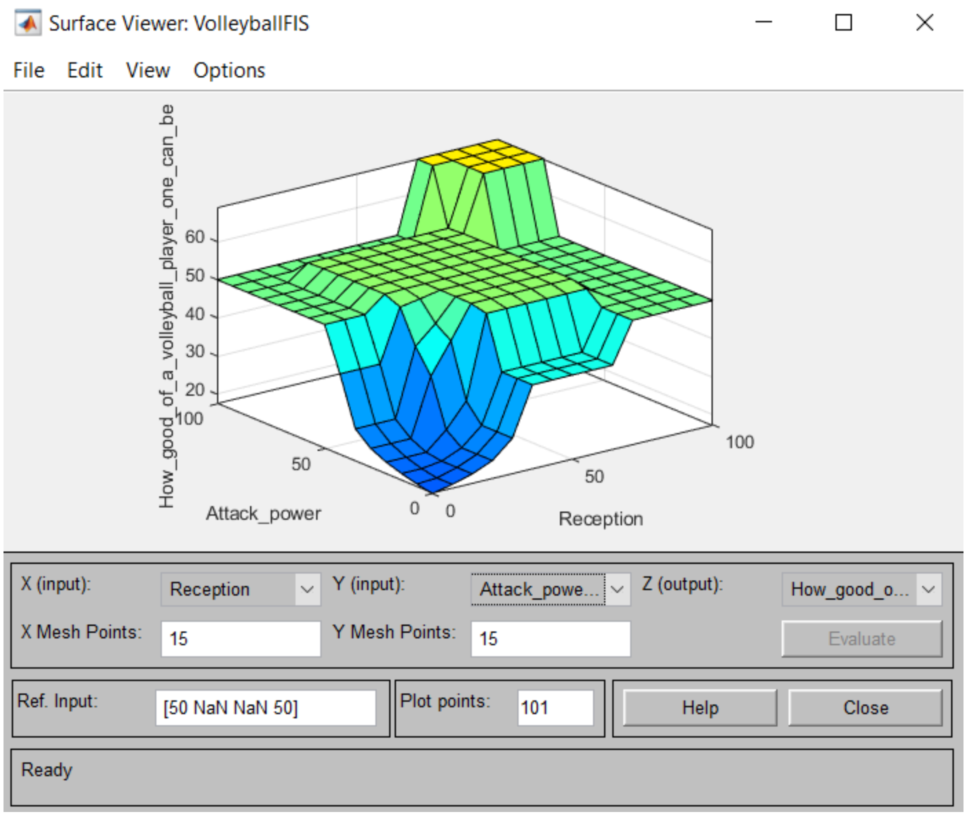
Picture 9. Surface view considering Agility and Reception.



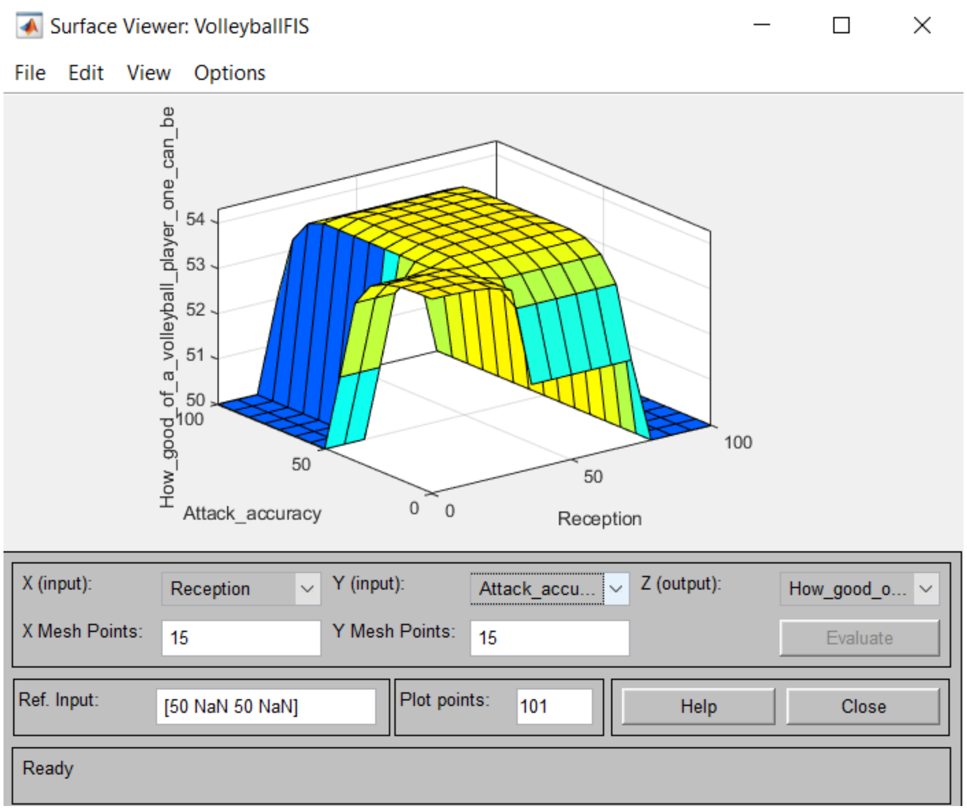
Picture 10. Surface view considering Agility and Attack power.



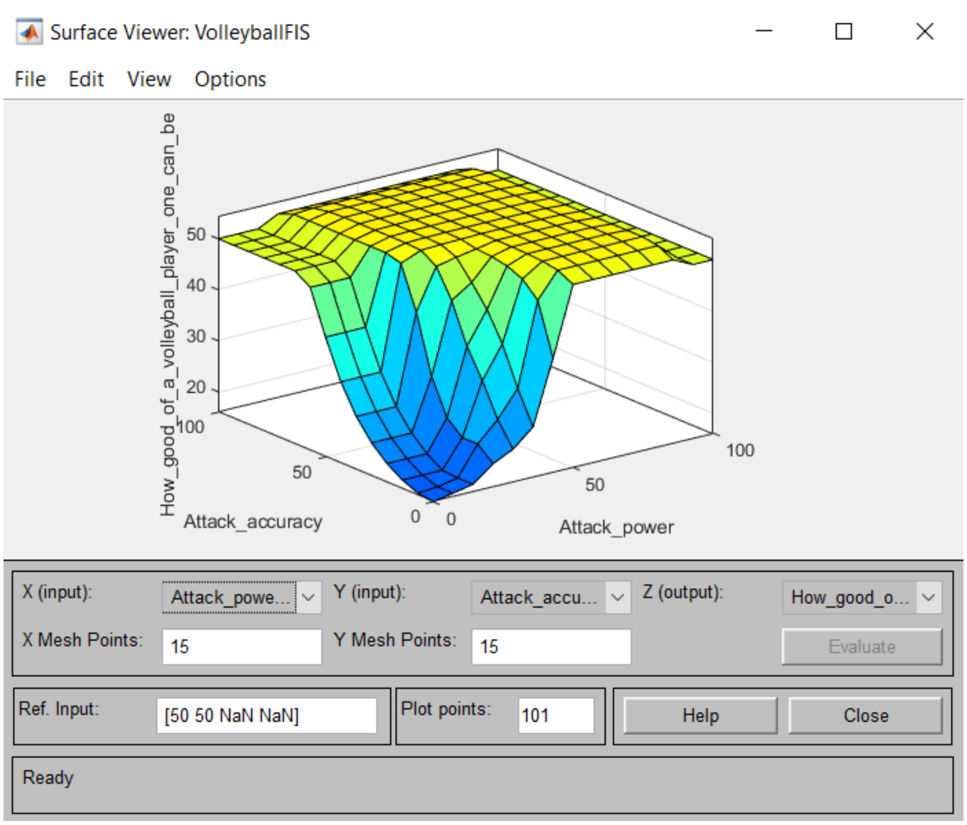
Picture 11. Surface view considering Agility and Attack accuracy.



Picture 12. Surface view considering Reception and Attack power.

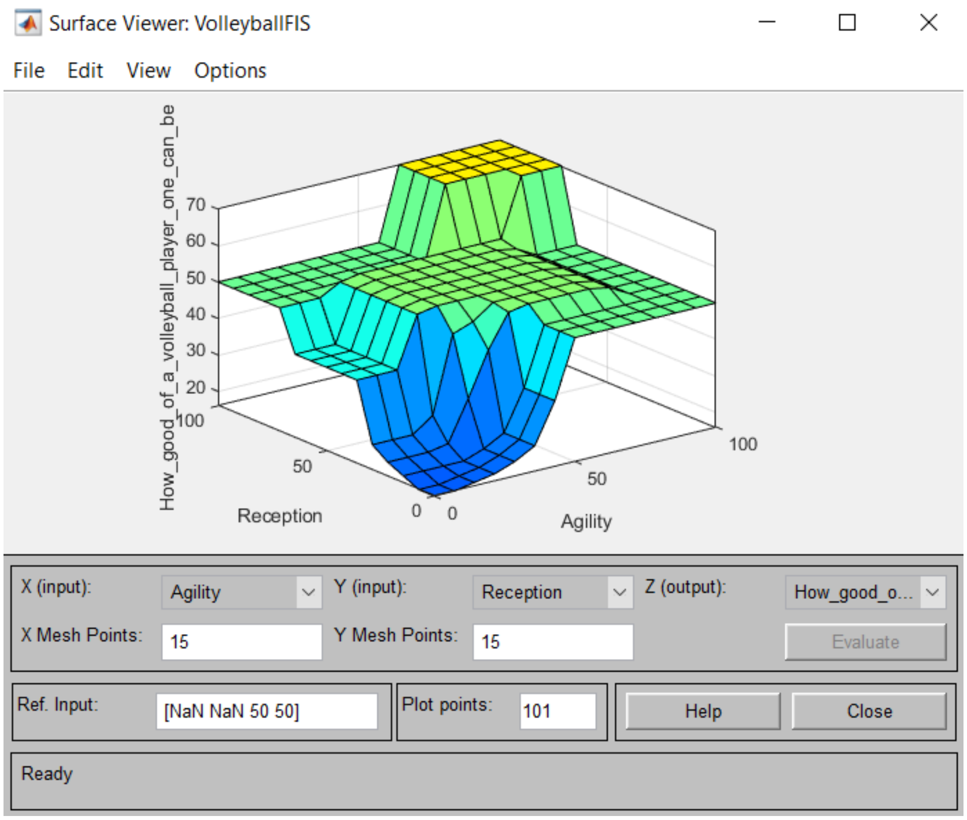


Picture 13. Surface view considering Reception and Attack accuracy.

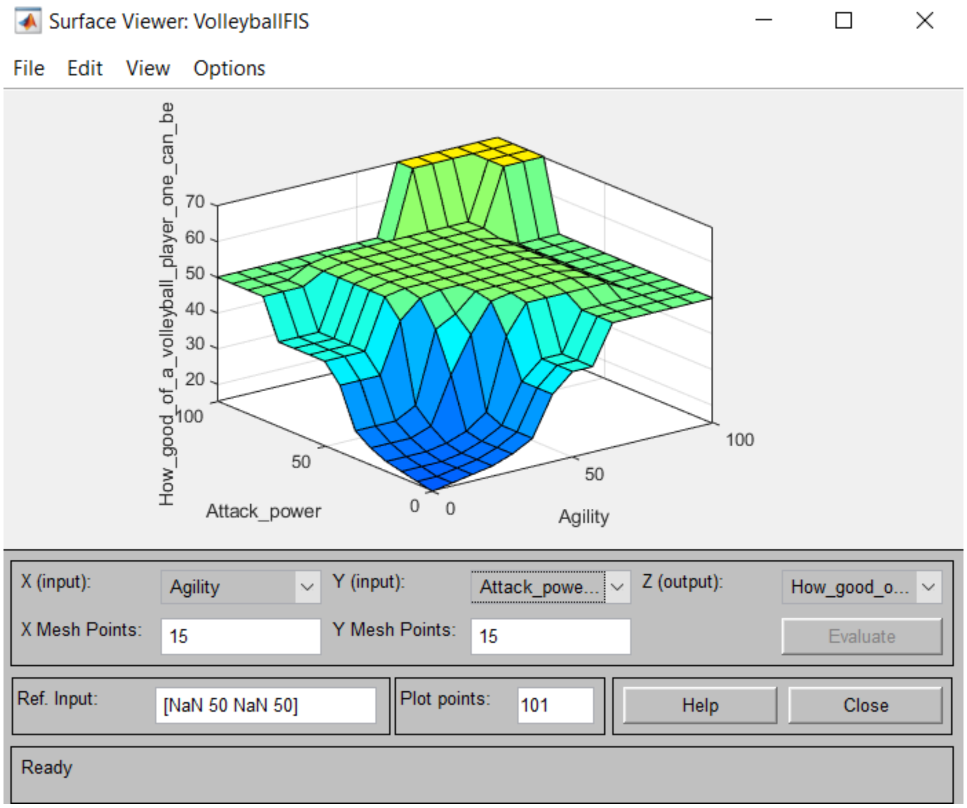


Picture 14. Surface view considering Attack power and Attack accuracy.

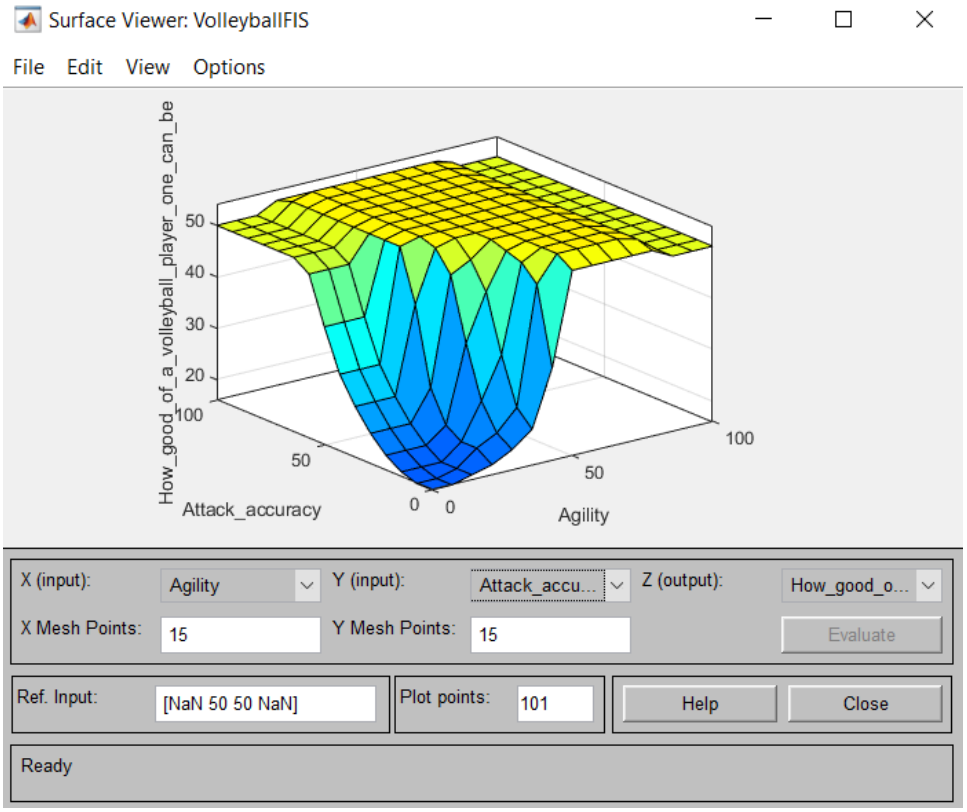
Bisector method:



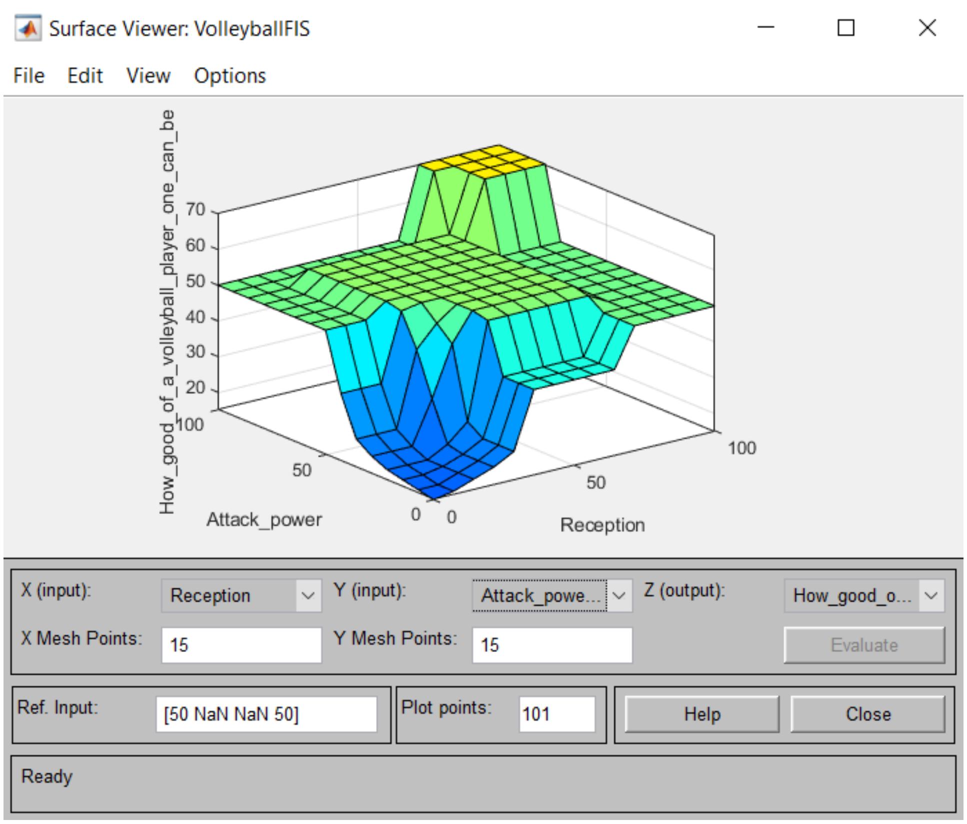
Picture 15. Surface view considering Agility and Reception.



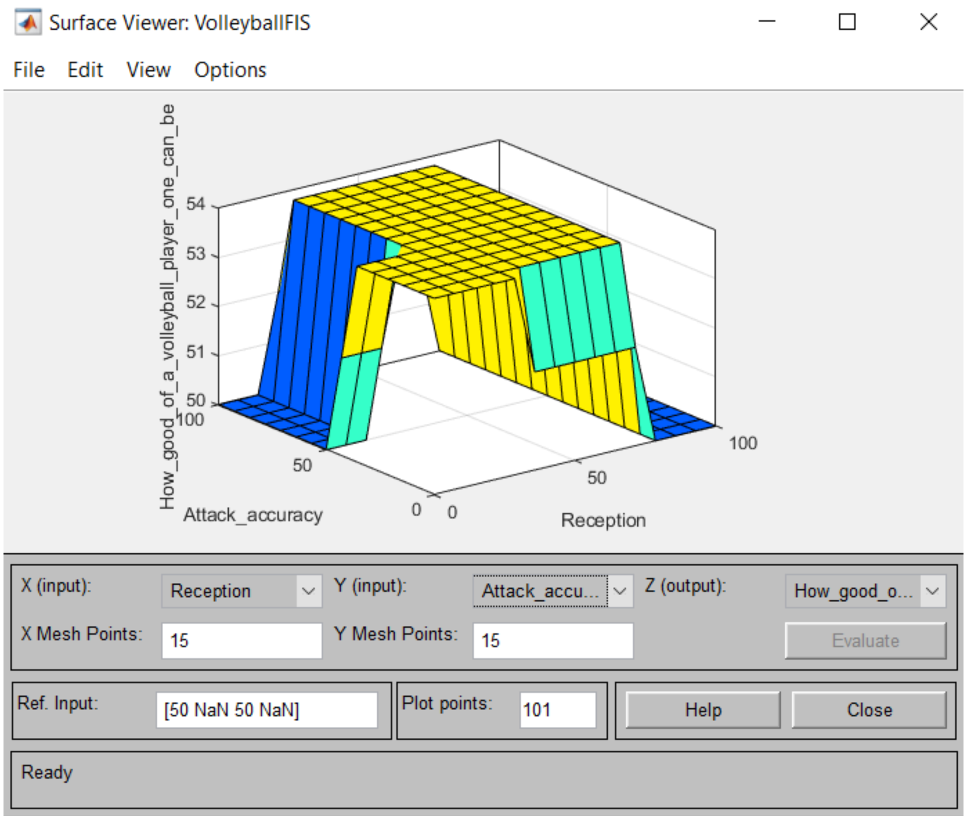
Picture 16. Surface view considering Agility and Attack power.



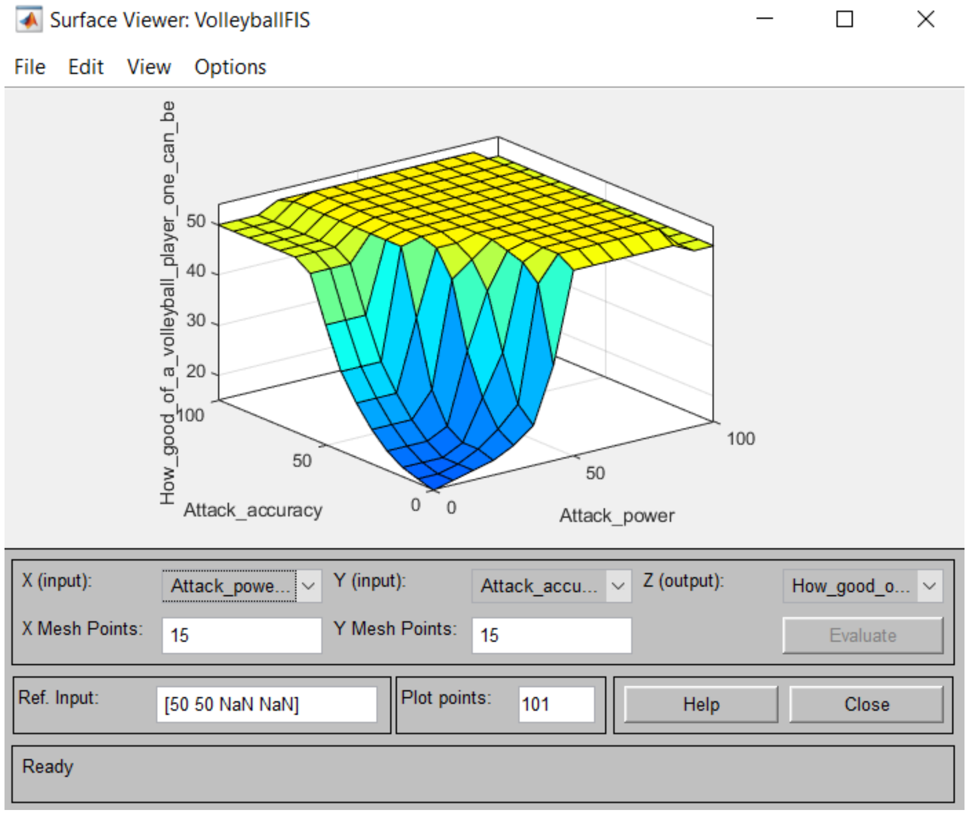
Picture 17. Surface view considering Agility and Attack accuracy.



Picture 18. Surface view considering Reception and Attack power.

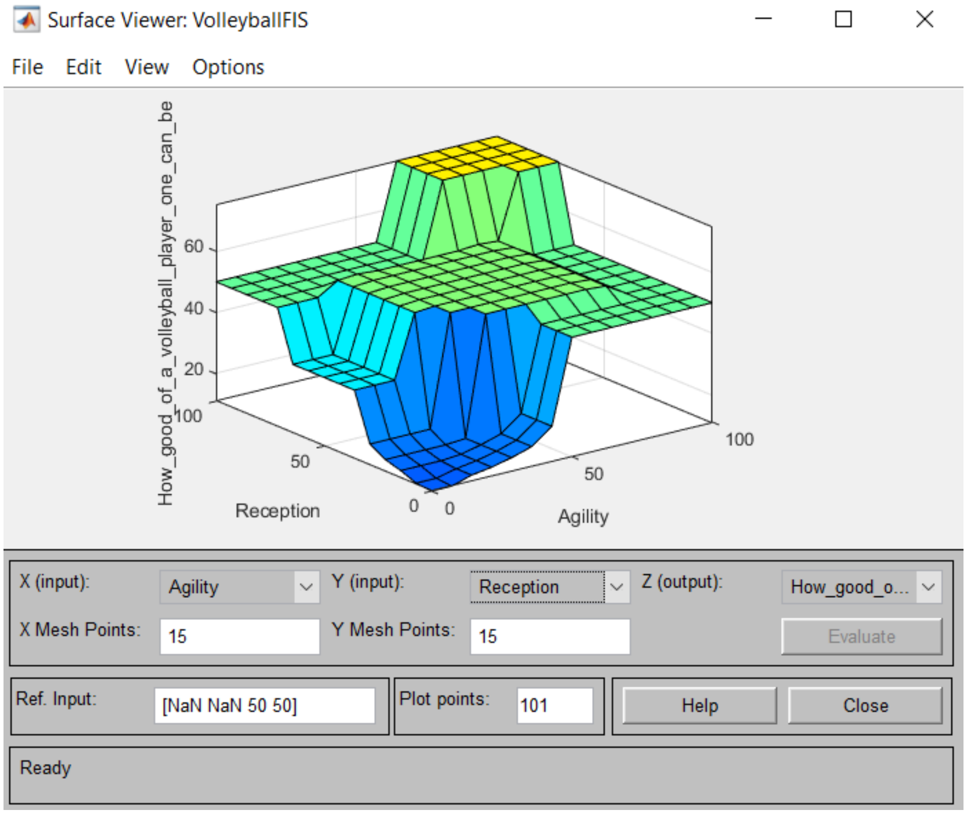


Picture 19. Surface view considering Reception and Attack accuracy.

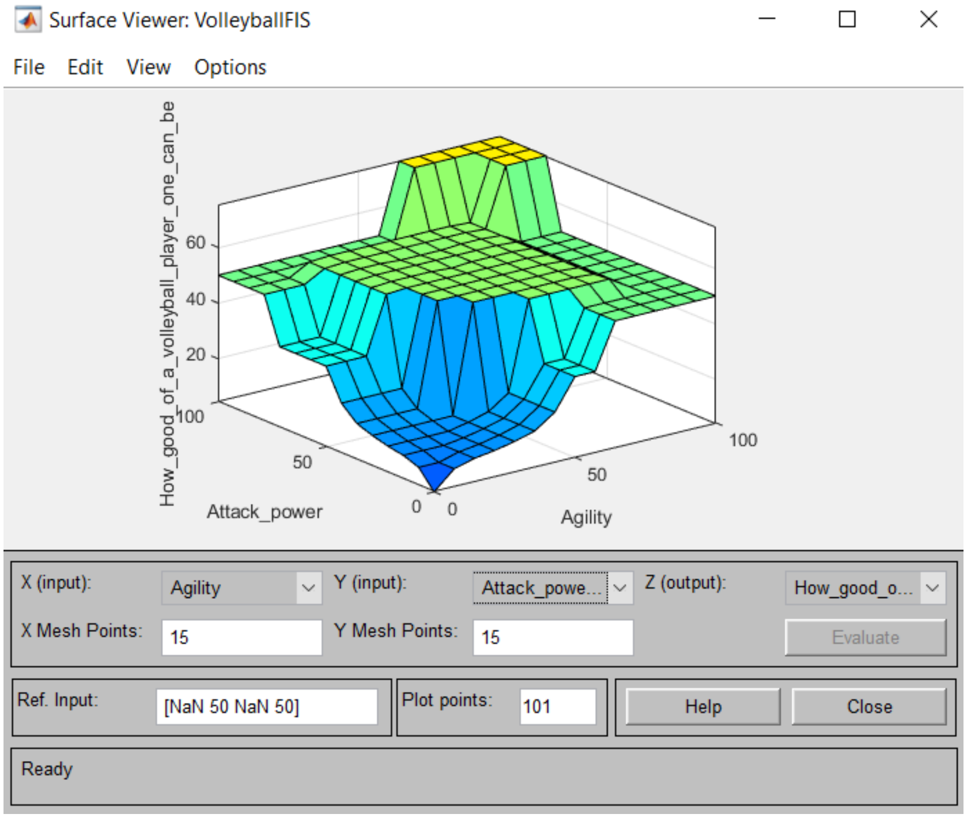


Picture 20. Surface view considering Attack power and Attack accuracy.

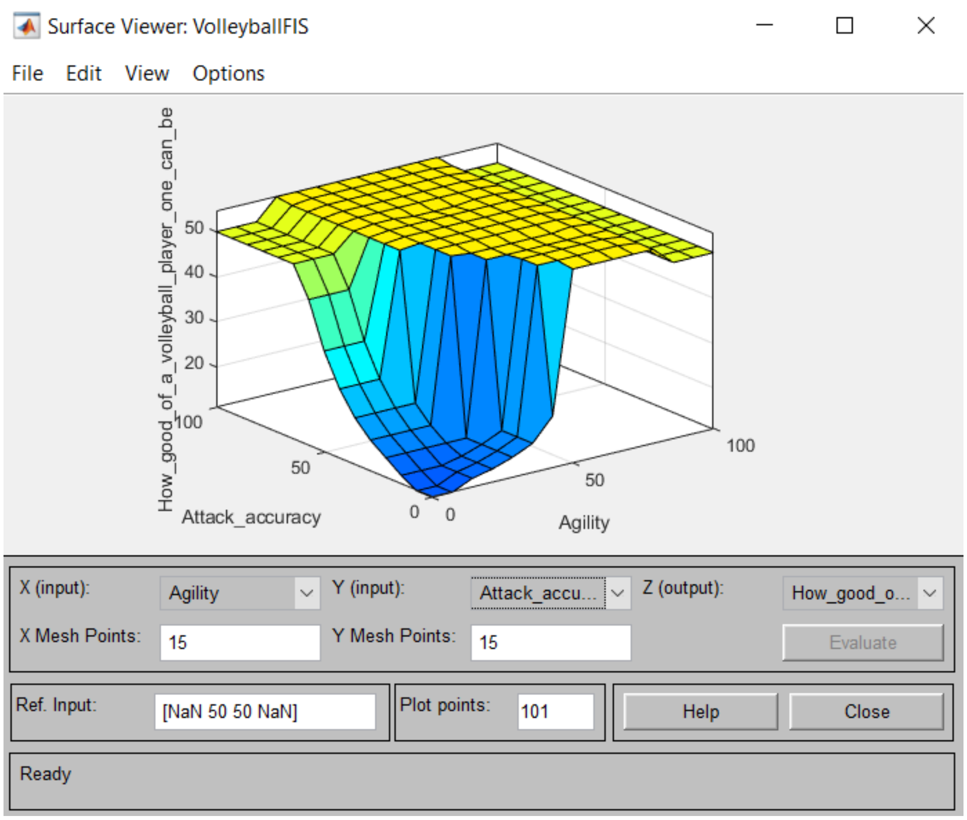
MOM method:



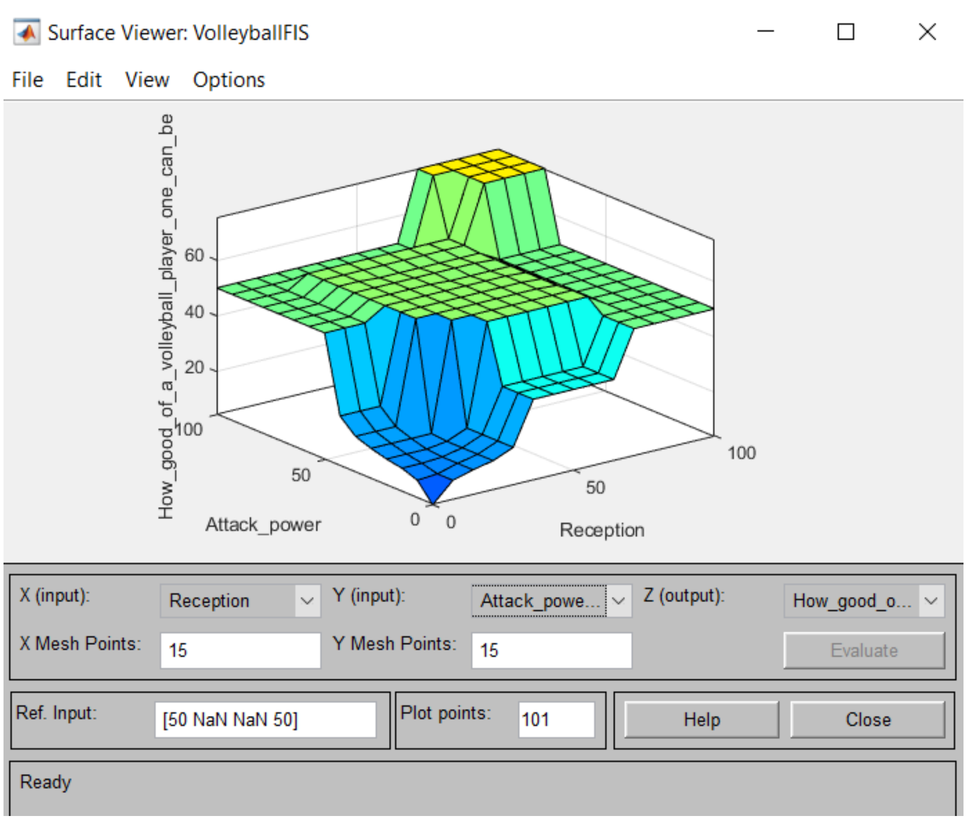
Picture 21. Surface view considering Agility and Reception.



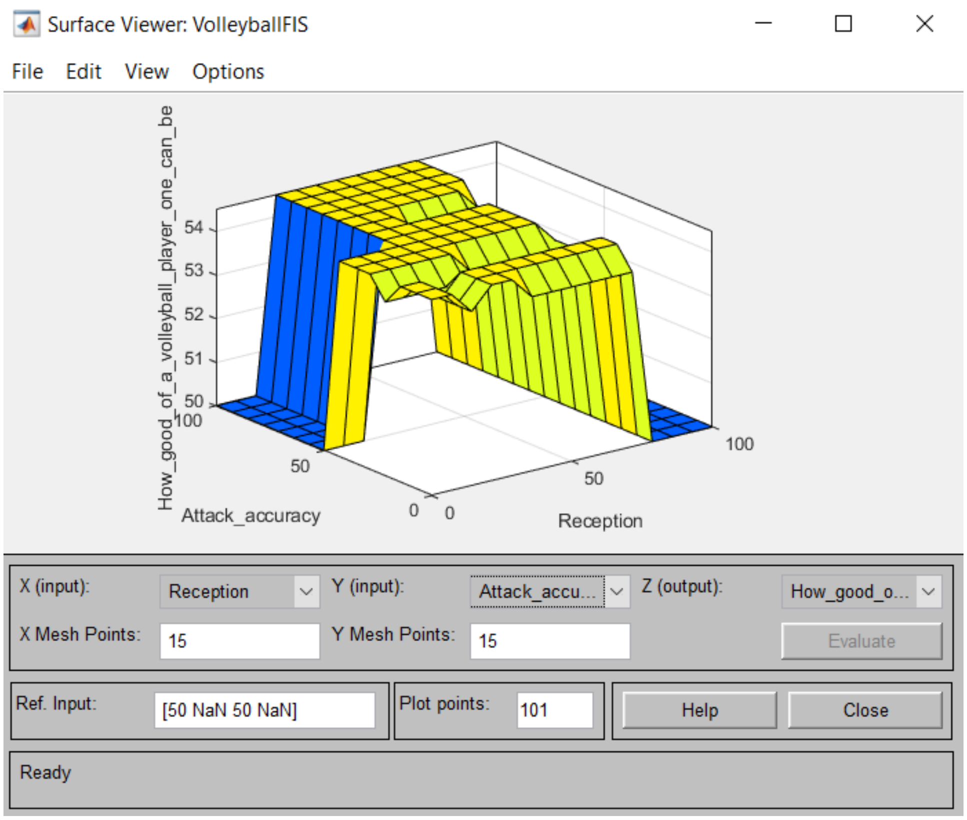
Picture 22. Surface view considering Agility and Attack power.



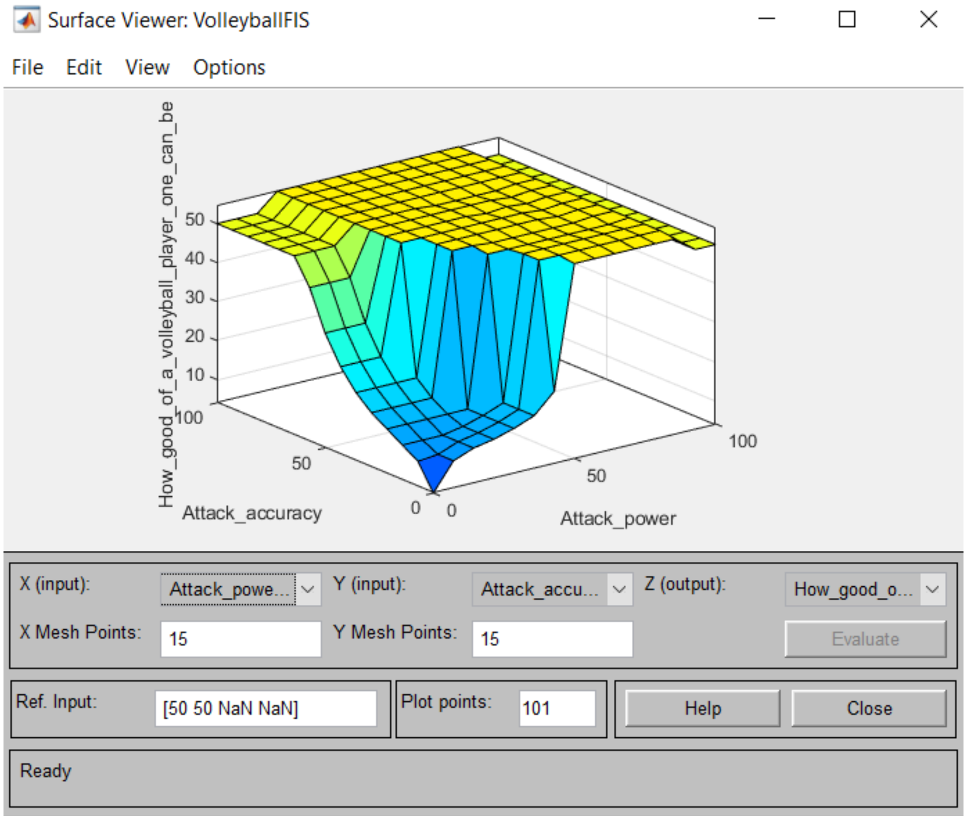
Picture 23. Surface view considering Agility and Attack accuracy.



Picture 24. Surface view considering Reception and Attack power.

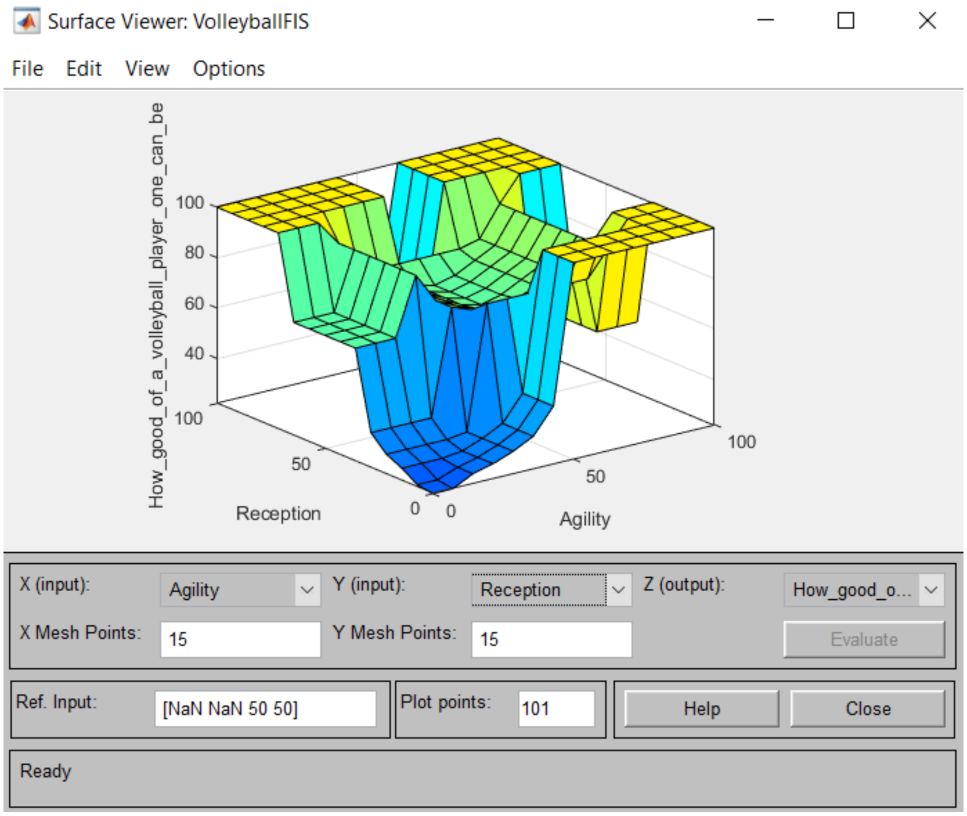


Picture 25. Surface view considering Reception and Attack accuracy.

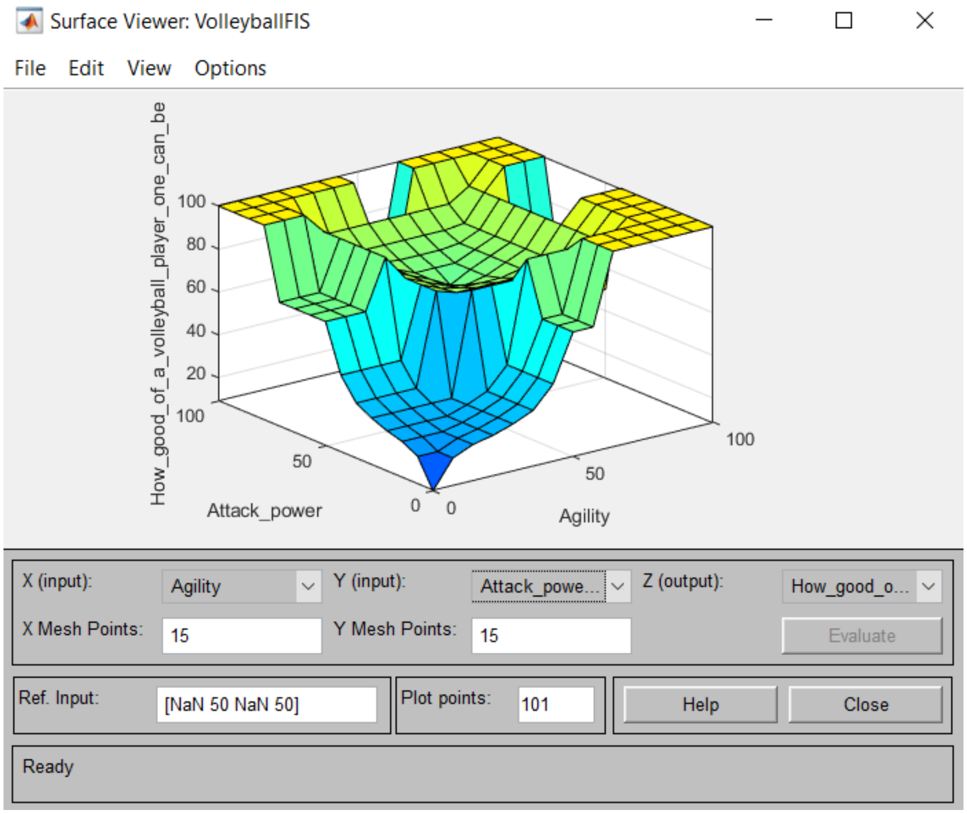


Picture 26. Surface view considering Attack power and Attack accuracy.

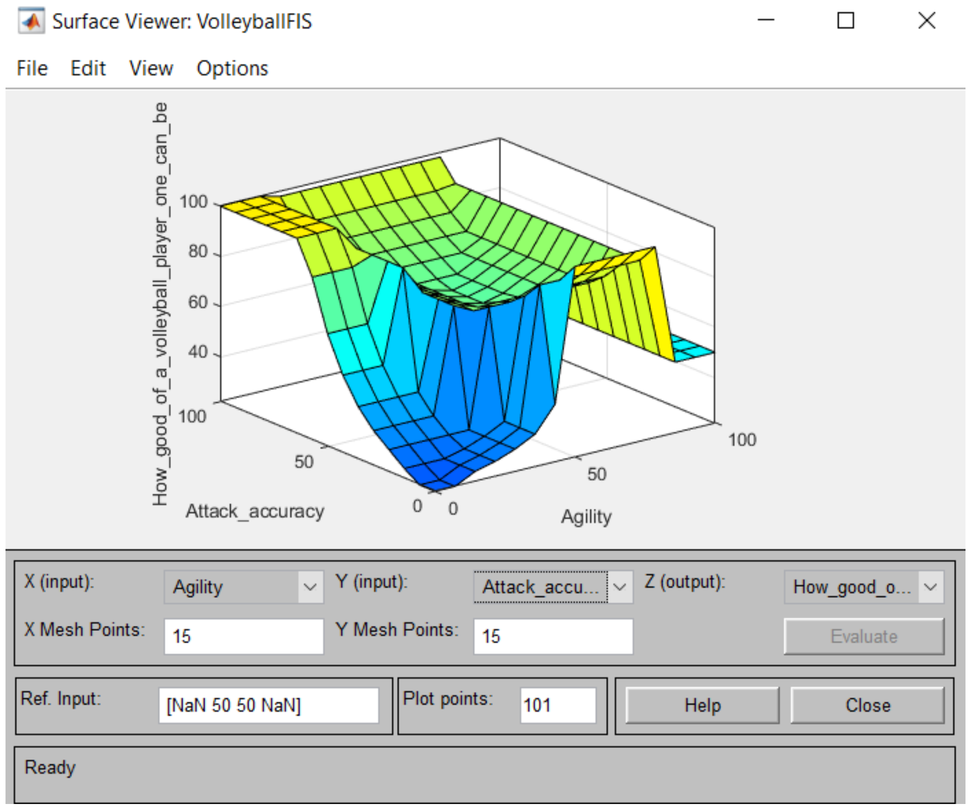
LOM method:



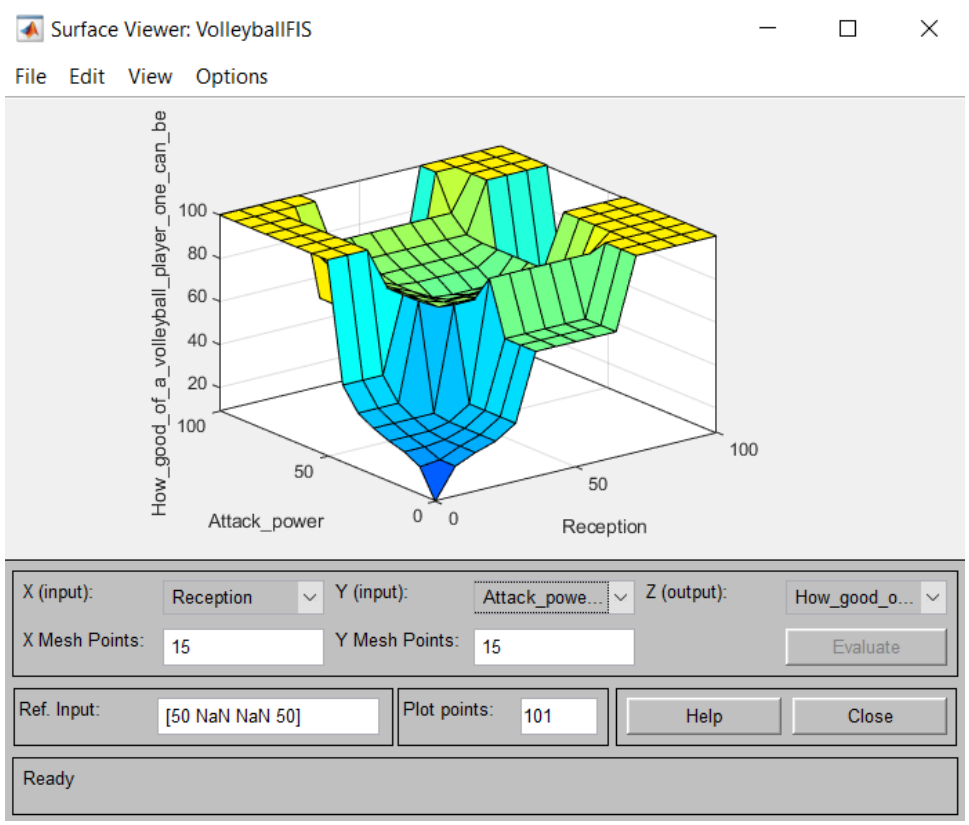
Picture 27. Surface view considering Agility and Reception.



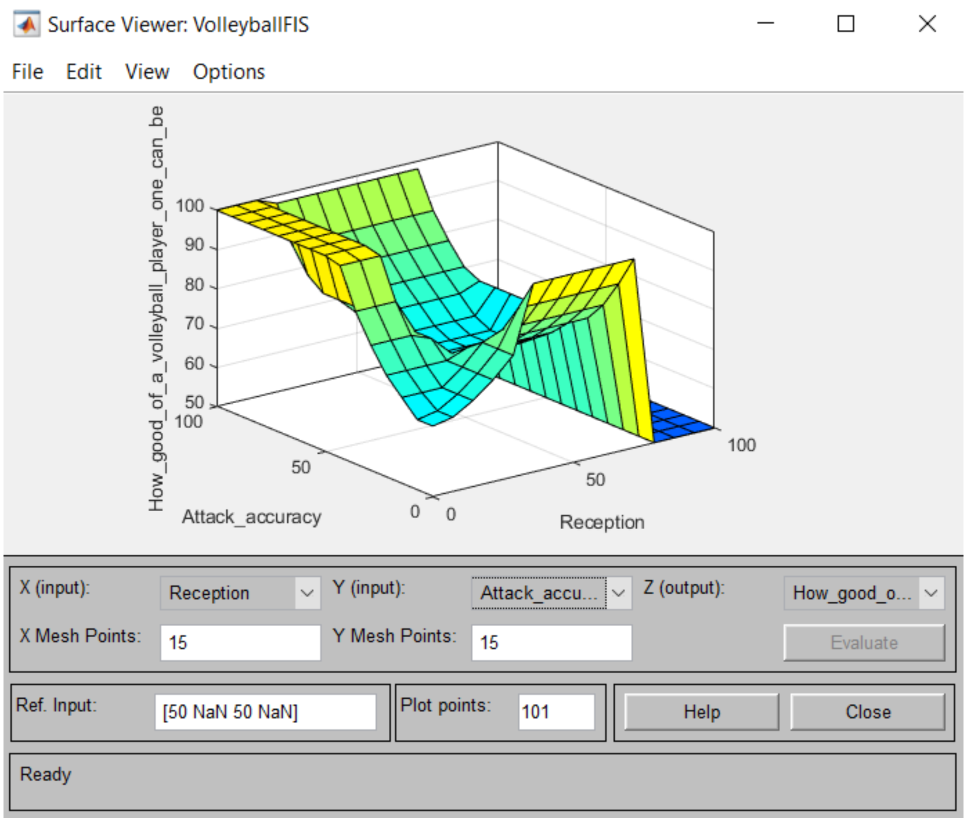
Picture 28. Surface view considering Agility and Attack power.



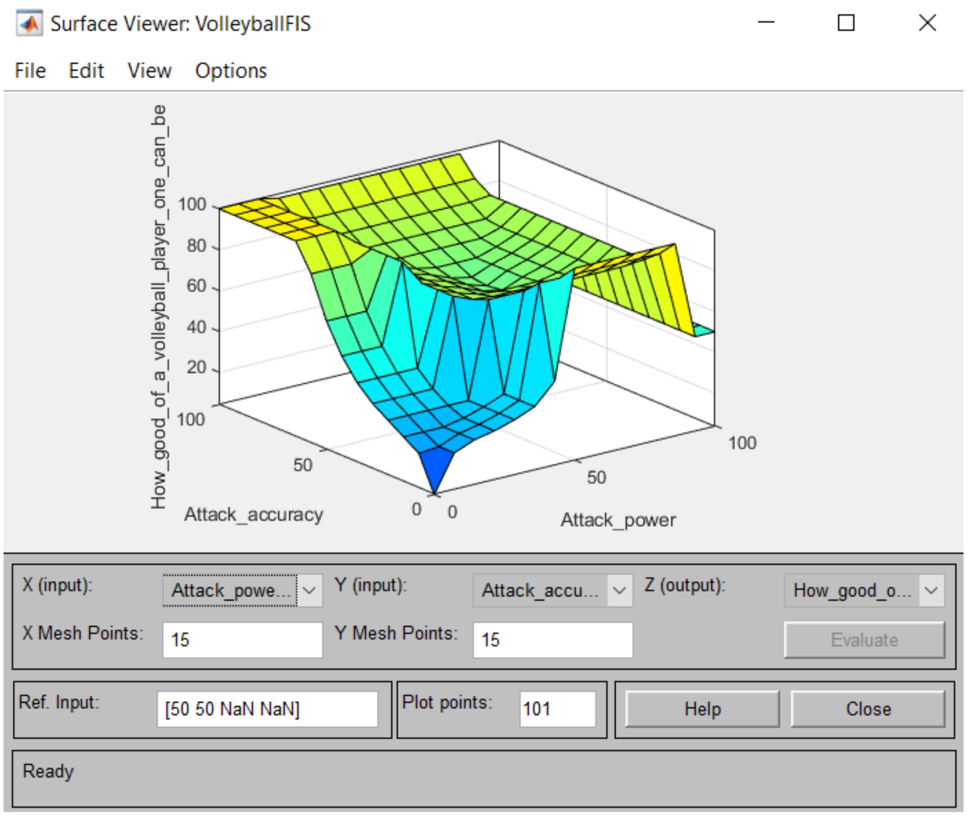
Picture 29. Surface view considering Agility and Attack accuracy.



Picture 30. Surface view considering Reception and Attack power.

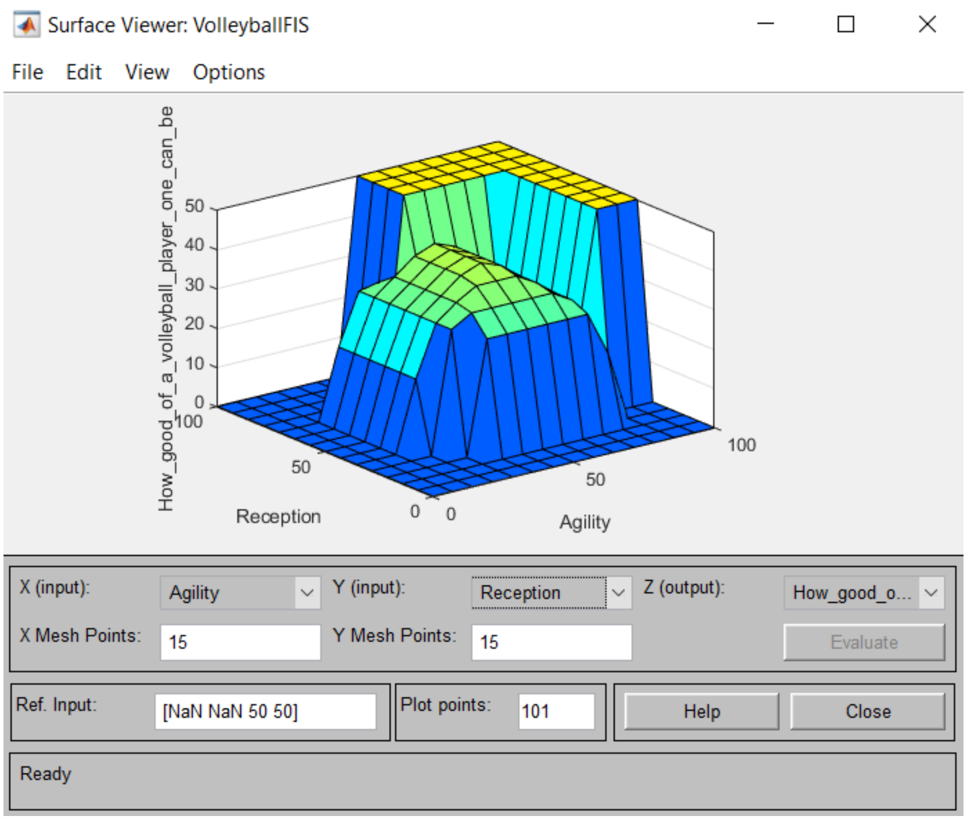


Picture 31. Surface view considering Reception and Attack accuracy.

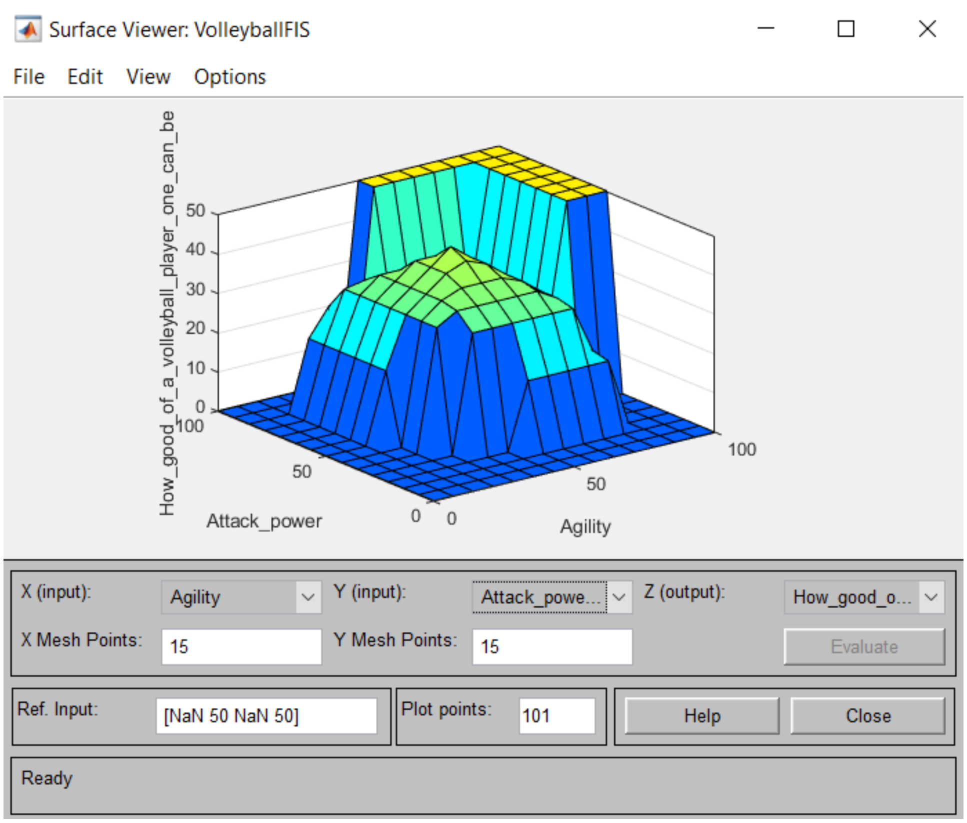


Picture 32. Surface view considering Attack power and Attack accuracy.

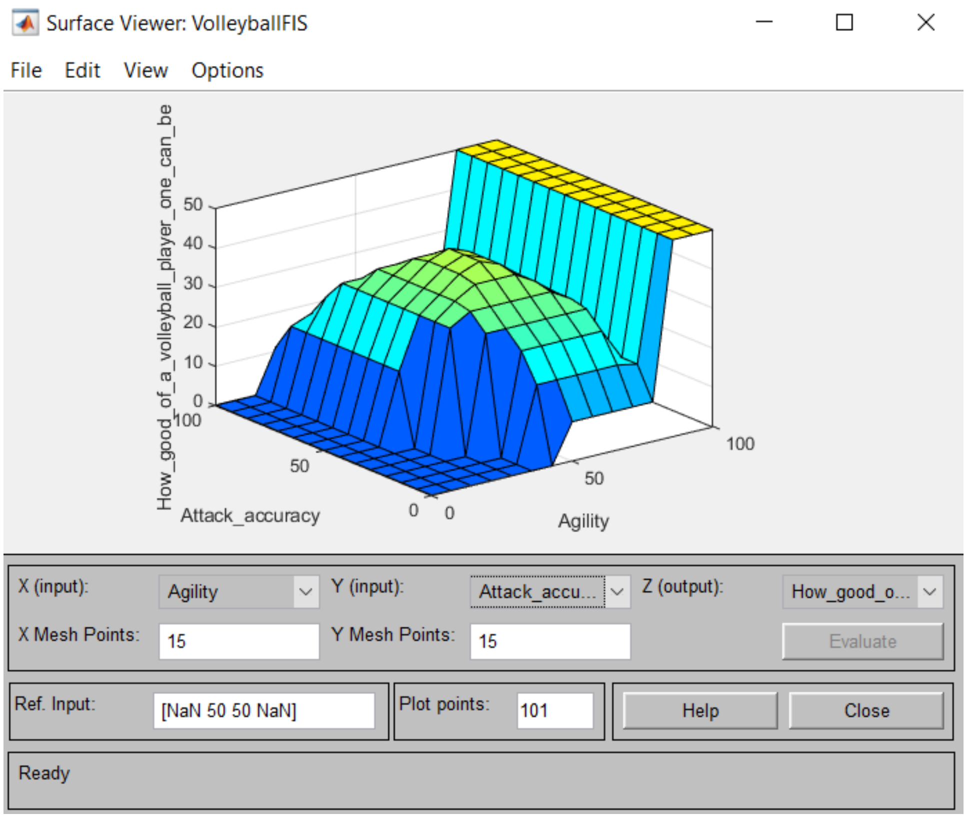
SOM method:



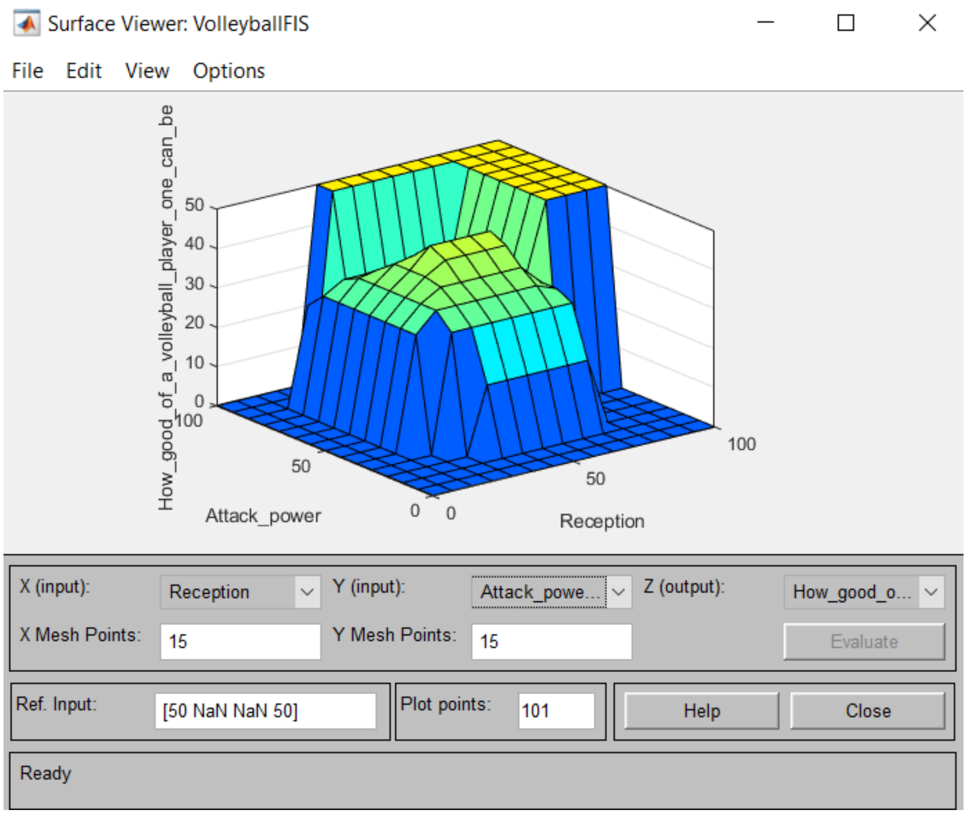
Picture 33. Surface view considering Agility and Reception.



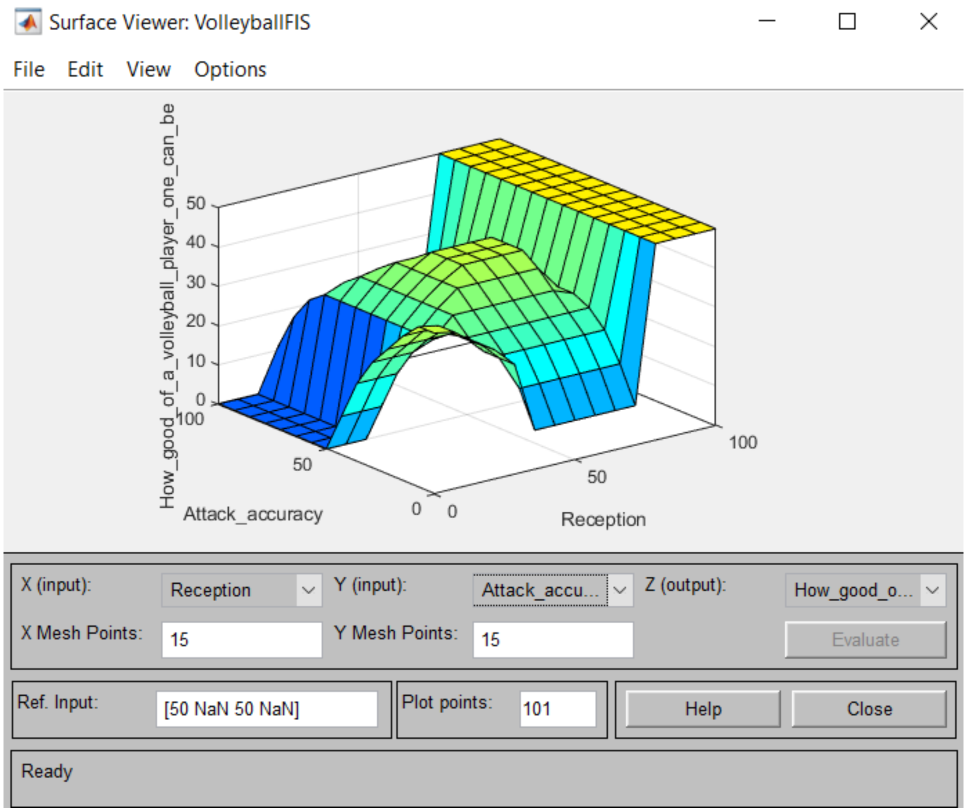
Picture 34. Surface view considering Agility and Attack power.



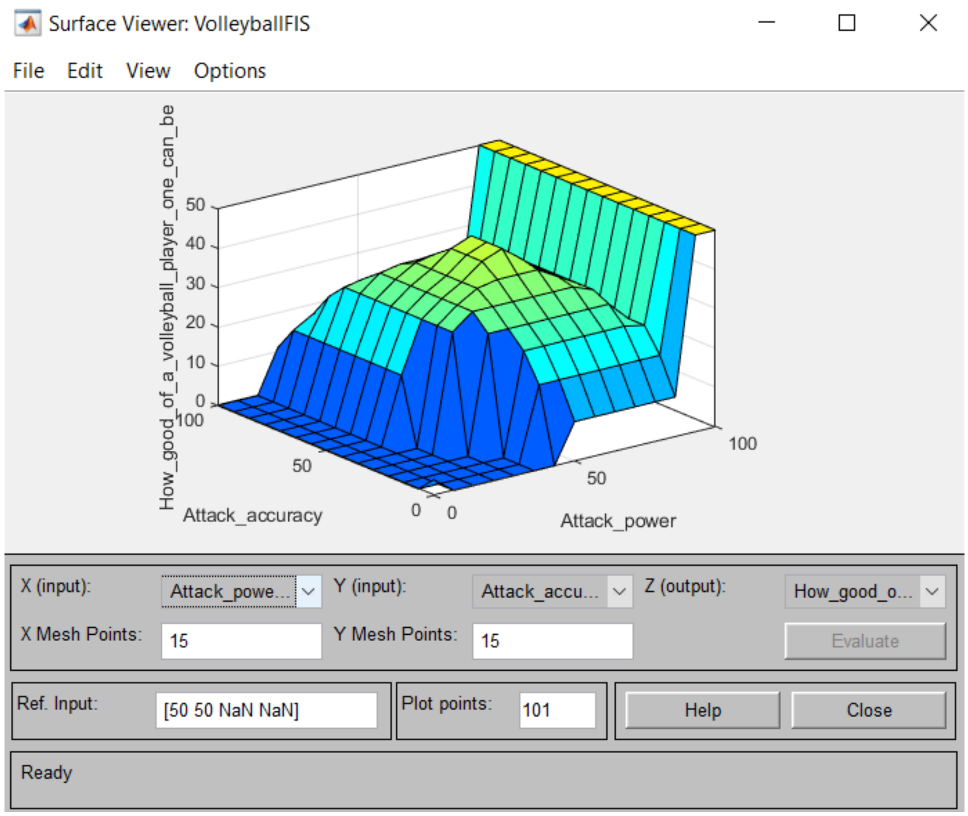
Picture 35. Surface view considering Agility and Attack accuracy.



Picture 36. Surface view considering Reception and Attack power.



Picture 37. Surface view considering Reception and Attack accuracy.



Picture 38. Surface view considering Attack power and Attack accuracy.