

# Selecting an Enterprise Development Strategy Using Machine Learning Methods

Mikhail Leizerovich Krichevsky, Julia Anatolevna Martynova

**Abstract:** It is particularly difficult to choose from enterprise strategies amid a changing economic situation, inaccurate definitions of variables affecting the company's actions, and incomplete information about the competitors' behavior. It is preferable to use methods that are part of machine learning (ML) when choosing a strategy in such a situation. The purpose of the study is to develop a method for selecting a strategy using ML instruments, which should include ways to select the most important indicators of an enterprise, estimate a market share of an enterprise among competitors, and test the efficiency of the created method on simulated or real data. Neural and fuzzy systems are the most suitable ML instruments for solving the problem. The Statistica software package implements a mechanism for selecting an appropriate enterprise strategy using the neural network (NN) system. A trained NN as a perceptron allows to choose a procedure of the organization's actions that suits the current situation best, depending on a set of selected variables that influence the strategy. A hybrid neuro-fuzzy system (NFS) like ANFIS, which is a combination of NNs and fuzzy logic (FL), is used to estimate the market share of the company. It is advisable to reduce the number of input variables when working with this module, which is done using the Feature Selection tool that allows to rank parameters and select the most significant ones.

**Index Terms:** enterprise strategy, machine learning, feature selection, strategy selection, market share estimation.

## I. INTRODUCTION

Selecting a strategy is an important task for any enterprise, the solution of which influences its further operation. In fact, a strategy is a long-term direction of the organization development in terms of the system of relationships within an organization and the enterprise position among competitors. In other words, a strategy is a set of rules for the organization's decision making in its operation [1,2]. Each of the strategies chosen by an enterprise leads to different consequences; therefore, strategies should be chosen taking all available information into account at the decision-making stage. It is noted in [1] that the strategy is a subtle and abstract concept. However, it is also a tool that can help a firm in the context of instability. As a management tool, the strategy therefore serves as an important guideline for enterprises in various fields. When choosing a strategy type, a list of indicators should be substantiated by which the strategy

selection is evaluated, the most suitable method for solving this task should be identified, the performance of the proposed methodology should be verified, and the accuracy of the method used should be determined [3,4].

The article is further arranged as follows. First, the methods of ML are reviewed and analyzed, which can be used to solve the problem of choosing a strategy. Then, the principles of NNs used in this article are briefly described, a method for reducing the dimensionality of the attribute space is described, the operation of the hybrid NFS is described, and the operation of the system for estimating the market share using the Simulink module is demonstrated. The results are discussed and ways for further research are indicated in the final part of the article.

## II. METHODS

Analysis of any data and their processing implies building a model from observations and its further use – for example, in classification, forecasting, etc. Methods that are used in artificial intelligence (AI) to process data are part of ML, which is a subset of AI. By definition provided in [5], ML is a method that generates a model from data. In this case, the data mean the following information: documents, audio, images, etc. As such, the model is the final ML product, which is suitable for tasks related to intelligence – in particular, in situations where physical laws or equations make it impossible to build a model. The process of building a model from training data is shown in Figure 1.

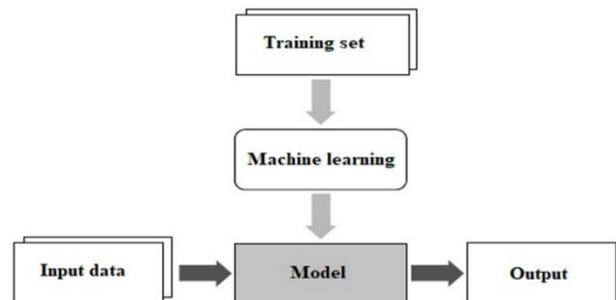


Fig. 1: Model definition from training data [5]

The vertical arrows in Figure 1 define the learning process, and the horizontal arrows indicate the use of the model. It must be mentioned that different data are used for building the model, the ways of its use also vary. Some of the ML methods used in this article are described below [5, 6].

### A. Neural networks

The NN is a "black box" reflecting the situation with a completely unknown process, where observations (examples) are available.

Manuscript published on 30 April 2019.

\* Correspondence Author (s)

**Kavin. R.**, EEE, Sri Krishna College of Engineering and Technology, Coimbatore, India.

**Elamcheren. S.**, EEE, Sri Krishna College of Engineering and Technology, Coimbatore, India.

**Dr. S. Sheebarani Gnanamalar**, EEE, Sri Krishna College of Engineering and Technology, Coimbatore, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

The inputs and the output are known in this case, but a database of examples is required by which the network is trained. The NN mimics the brain mechanism to some extent. In general, the NN is a machine simulating a method of processing a specific task by the brain [7]. Such a network is realized through the electronic components or modeled by a software running on a computer.

Due to their learning and generalization capabilities, NNs can be expressed as a mathematical representation of the human brain architecture.

Let us explain the principle of the NN technology on a single-layer network with  $R$  inputs and  $S$  neurons in the layer (Figure 2). In this network, each element of the input vector  $\mathbf{p}$  is connected to each input neuron through the weights' matrix  $W$ . The neuron with index  $i$  has a summer, where weighted inputs and offsets arrive to form the  $i$ -th scalar output  $n(i)$ . Various  $n(i)$  are combined together to form the  $S$ -th element of the input vector  $\mathbf{n}$ , which are the arguments of the activation function  $f$ . As a result, the output vector column  $\mathbf{a}$  is formed at the output of the network.

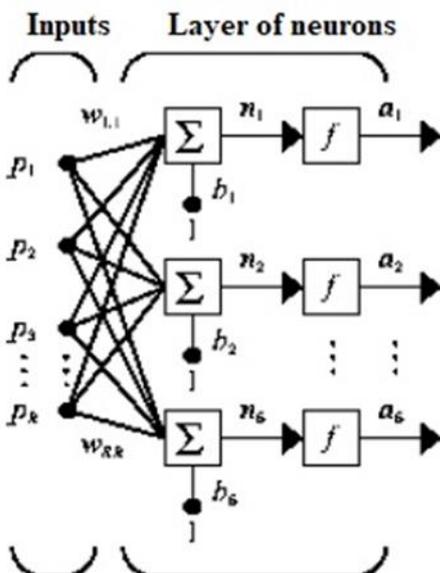


Fig. 2: One-layer network[8]

The principle of the network operation remains unchanged except that the outputs of the hidden layer are the inputs of the output layer. After the data are collected and a database of examples is formed, the network can be chosen quite simply, since the multilayered direct propagation NN is a "workhorse" in the NN methods. The NN with two layers (hidden and output) is used in this case. The NN operation is determined by how well it generalizes: in other words, how correctly it predicts the results of the classification on the data that were not used in the learning process.

Let us summarize the most characteristic positive and negative characteristics of the NN technology in Table 2 [9].

Table 2: Advantages and disadvantages of NNs

Advantages	Disadvantages
1. They have the property to learn by example.	1. They take much time to train in tasks with local minima.
2. They approximate any multidimensional nonlinear function.	2. They don't reveal the relationships between

3. They don't require a deep understanding of the process under study.	variables and don't increase knowledge of the process.
4. They are resistant to noise.	3. They have poor generalization when presenting new images in some tasks.

## B. Feature selection

One of the common tasks of the ML is the predictor selection from a large list of candidates. The NN training is difficult with a large number of input parameters, and time costs increase.

The "curse of dimensionality" term usually refers to the difficulties associated with fitting a model, evaluating its parameters, or optimizing a multidimensional function with a large sample. This happens quite often when NNs are used. As the space dimension of the input data increases, it becomes more difficult to find global optima for this space. As such, there is a relevant practical need to select input variables from a large set that are most useful for predicting the outputs of the dependent output variables.

The methods that are implemented in the *Feature Selection and Variable Screening* module of *Statistica* software are designed to handle large sets of continuous and/or categorical predictors in regression or classification tasks. A subset of predictors can be selected from a large list of candidates without assuming the degree of relationship between predictors and dependent variables in this case. This module can serve as an ideal preprocessor in ML, allowing to select smaller sets of predictors for further analysis.

One of the potential solutions to the problem of selecting predictors from a large set of candidates is to find correlations between each predictor and the dependent variable. The software can find correlation coefficients for regression problems (with a continuous dependent variable) and then select predictors that demonstrate the highest correlation with the dependent variable. The problem with this approach is that the correlation coefficient determines linear relationships. However, such relationships are most often nonlinear in real situations, and many data processing algorithms in ML do not imply a linear (or even monotonous) relationship between the predictors and the dependent variable in question. As a result, the use of such a strategy will distort the results for further analysis, since the selection process itself contributes to specific models and results (linear models, monotonous dependence). The solution of the parameters selection in this module does not imply any particular form of connection between the predictors and dependent variables (classes) in question. Instead, a generalized concept of relationship is applied in the software with simultaneous selection of predictors for regression (classification). As a result, the list of predictors selected using this technique can be further expanded by linear or nonlinear regression (classification) algorithms. The method used in the *Feature Selection and Variable Screening* module is optimized for large data sets and usually requires only two passes through the data.



The variable selection algorithm can be applied to regression (continuous dependent variable) or classification tasks (categorical dependent variable).

**C. Hybrid neuro-fuzzy network**

A hybrid network that combines the principles of NNs and FL is a multilayered NN of a special structure without feedbacks, which uses common (not fuzzy) signals, weights, and activation functions.

The main idea used in the hybrid network model is to apply the existing data sample to determine the parameters of the membership functions that best fit the specific fuzzy inference system. The known NN learning procedures are used for this purpose. The NFS concept is to determine the parameters of fuzzy systems by training methods adopted in NNs [10]. Combining NNs and FL into one hybrid system allows to eliminate the disadvantages of individual technologies and create such NFS as ANFIS (*Adaptive Network-Based Fuzzy Inference System*) [11].

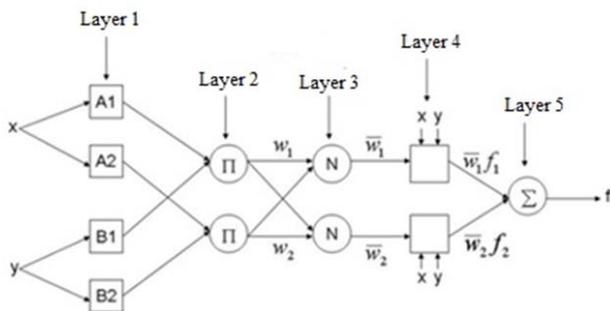
The FL mechanism can be implemented using an algorithm such as *Mamdani* or *Sugeno*. The main difference between the Mamdani and Sugeno interference systems is the way to get a clear interference. In the Mamdani algorithm, a clear interference value is obtained by defuzzification of the interference fuzzy set. In the Sugeno algorithm, a weighted average is formed at the output.

ANFIS implements the Sugeno fuzzy inference system as a five-layer NN of direct signal propagation.

The layer purpose is as follows:

- first layer – terms of input variables;
- second layer – antecedents (premises) of fuzzy rules;
- third layer – normalization of compliance with the rules;
- fourth layer – conclusion of the rules; and
- fifth layer – output variable.

The ANFIS network with two input variables and two fuzzy rules is shown in Figure 3.



**Fig. 3: ANFIS network architecture with two inputs[10]**

**D. Network performance evaluation**

The *Simulink* simulation tool, which is an application to the *MatLab* package, can be used for the network performance evaluation. A principle of visual programming is used in this simulation, according to which the user creates a device model and performs calculations on the screen from the library of standard blocks. Unlike the classical simulation methods, the user does not need to learn a programming language but can have general knowledge in this field.

The *Simulink* interactive environment allows using ready-made libraries of blocks for modeling electric power, mechanical and hydraulic systems, and applying an advanced model-focused approach in the development of control

systems, digital communications and real-time devices. *Simulink* allows to model, simulate and analyze systems whose outputs vary over time. This package can be used to study and analyze a wide class of systems.

Simulation of the systems is a two-step process:

1. A user creates a block diagram using the *Simulink* editor, which graphically displays the mathematical relationships among inputs, states, and outputs of the system.
2. Then the user gives a command to simulate the system for a predetermined time.

**III. RESULTS**

A base of examples (training set) is needed to solve the problem of choosing strategies for training the NN. Such a base can be collected in two ways:

- use the real indicators of companies on which the choice of strategy depends, according to the developer of the method;
- apply the drawn data of indicators using the Monte Carlo method, for example.

The second approach will be used in this article, emphasizing that the purpose of the article is to create a method for selected a strategy, where it not so important what data were used to obtain it.

For concreteness of the solution, let us assume that an enterprise manufactures high-tech products – in particular, a set of controls for the technical equipment operation. Based on the experience of solving such problems in the field of strategic management, the following indicators are selected, on which the enterprise strategy depends: *X1* is the level of prices for high-tech products; *X2* is the level of product quality; *X3* is the level of technical support for products; *X4* is the degree of compliance of the product range with the needs of consumers; *X5* is the frequency of product innovation; *X6* is the support for long-term relationships with consumers; *X7* is the level of costs for the products update; *X8* is the control over distribution channels; and *X9* is the level of capacity utilization.

Besides, it must be assumed that there are three strategic groups of competitors in this area, which vary by the above indicators.

Let us form the base of examples for nine indicators using the Monte Carlo method, where the range of changes is assumed to be from zero to ten points. The set size is set as 30 implementations, 10 objects in each of the three selected groups. The set fragment is provided in Table 3.

**Table 3: Fragment of the set**

X1	X2	X3	X4	X5	X6	X7	X8	X9
7.38	3.55	4.94	3.07	5	9	2	5	6
7.10	3.36	5.41	3.13	3	6	5	7	1
7.60	3.99	3.40	3.93	7	9	7	3	7

## Selecting an Enterprise Development Strategy Using Machine Learning Methods

7.90	3.21	5.92	3.36	3.7	3.1	3.4	7.3	5.1
				9	2	1	8	1
7.88	3.88	5.16	3.90	3.5	3.7	3.4	6.7	5.1
				4	9	4	0	4

Let us emphasize that the complete set of the performed data consists of 30 rows and nine columns, i.e., each object is described by a nine-dimensional vector.

Let us apply the principal component analysis (PCA) to this sample in order to represent all objects in a space of a lower dimension – for example, on a plane. All 30 objects on the plane of the first two PCAs are shown in Figure 4.

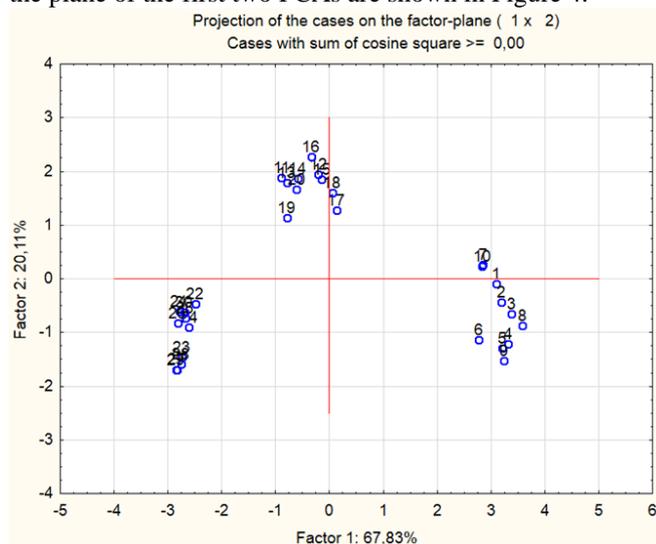


Fig. 4: Objects in the plane of the first two principal

Table 4. Characteristics of the chosen network

Net. ID	Net. name	Training perf.	Test perf.	Validation perf.	Algorithm	Error funct.	Hidden act.	Output act.
3	MLP 9-9-3	100,000000	100,000...	100,000000	BFGS 4	SOS	Identity	Exponen...

It can be seen from Table 4 that the chosen NN is a perceptron with a 9-9-3 formula, i.e., a network with nine neurons in the input and hidden layers and three neurons in the output layer. The network demonstrates very good learning results: no failures on the training, test, and validation sets.

The *BFGS* (*Broyden-Fletcher-Goldfarb-Shanno*) algorithm was used to train the network, which is an iterative method for solving optimization problems. *SOS* (*Sum of Squares*) was used as a failure function. Activation between the input and hidden layers was taken as an identical function; activation between hidden and output layers was taken as an exponential function.

The confusion classification matrix for this network is shown in Table 5.

Table 5: Confusion matrix

Var10 (Classification summary) (Spreadsheet26)				
Samples: Train				
	Var10-1	Var10-2	Var10-3	Var10-All
Total	7,0000	7,0000	8,0000	22,0000
Correct	7,0000	7,0000	8,0000	22,0000
Incorrect	0,0000	0,0000	0,0000	0,0000
Correct (%)	100,0000	100,0000	100,0000	100,0000
Incorrect (%)	0,0000	0,0000	0,0000	0,0000

### components

As can be seen from Figure 4, all objects are well separated (due to the method of data formation through simulation) and form three explicit clusters. Please note that PCA is an aggregated indicator that includes all nine of the original weighted features.

In the transition to the NN technology, the training set is supplemented with another column that defines the number of the chosen strategy. As such, the source data represent a matrix consisting of 30 rows and ten columns in this case.

Once the base of examples is formed, the basic steps are performed by the *Neural Network* module of the *Statistica* software, where the developer first needs to specify only the number of network layers and the number of neurons in each layer.

The problem can be solved using a perceptron type of network with input, one hidden and output layers. The number of neurons in the input layer is determined by the number of features (nine), and the number of neurons in the output layer is determined by the number of groups (three) into which the initial data set was divided. The composition of the hidden layer neurons is formed by the software, preserving the network with the best architecture by learning failure. Following the analysis of results, the best network was selected, and its characteristics are provided in Table 4.

As can be seen from Table 5, there are no failures in training.

Before using the NN in the network operation mode, it was alternately presented with objects that it had already "seen" during training, determining their belonging to the first, second or third groups of selected strategies. For example, the values of the input parameters *Var1* – *Var9* with a known belonging to groups are indicated in the first three lines of Table 6, starting with *Var1* indicator.

Table 6: Test results

#	3.Var10	Var1	Var2	Var3	Var4	Var5
1	1	7,380000	3,550000	4,940000	3,070000	3,050000
2	2	7,050000	5,510000	5,950000	3,930000	7,330000
3	3	7,360000	7,410000	9,310000	9,460000	9,010000
4	3	7,000000	5,000000	4,000000	7,000000	5,000000

It follows from column 2 of this table, designated as 3.Var10, that a trained NN correctly recognizes these objects. The fourth line of Table 6 contains values of the new object, and the network classifies this enterprise, which it has not seen before, in the third group. Next, let us turn to the problem of estimating the market share that a given enterprise can capture with the chosen strategy. This problem will be solved using the hybrid NFS *ANFIS*.

Let us preliminarily reduce the number of input variables, since the solution of the problem becomes more complicated with a large number of these parameters.

The *Feature Selection* module from the *Statistica* software will be used, which ranks nine input variables by their importance, as shown in Figure 5.

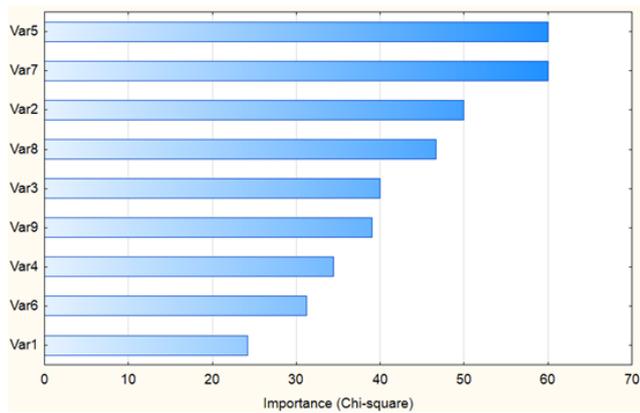


Fig. 5: Ranking input variables

Let us confine to the four most significant variables: *Var5* is the frequency of product innovation; *Var7* is the level of costs for the products update; *Var2* is the level of the product quality; and *Var8* is the control over product distribution channels. In this case, the training set consists of the initial 30 rows and five columns. The first four columns correspond to the selected parameters, and the last column provides the values of a market share on a 100-point scale, estimated by an expert. The matrix rows were randomly mixed, after which two sets have been formed: a training one (21 rows) and a test one (nine rows). These sets loaded into the *Matlab* software are shown in Figure 6.

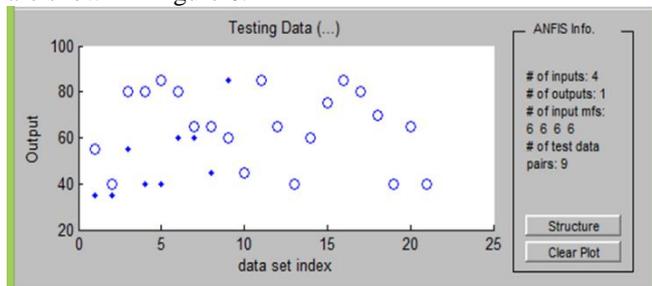


Fig. 6: Training and test sets

The fuzzy inference system can be generated in two ways:

- through the *grid partition*. In this case, the membership functions of fuzzy terms are evenly distributed inside the data universe;
- through *subtractive clustering*. In this case, the number of clusters is determined at the time of processing the initial data distribution.

The option where the test result shows a smaller error is preferable. The errors turned out to be approximately similar in this article, therefore the second option was chosen due to the simpler NFS structure. The structure of the formed *ANFIS* system is shown in Figure 7, from which it follows that the generated system has five layers: the first layer determines the number of input parameters; the second layer determines the number of membership functions for each variable; the third layer determines the established number of rules; the fourth layer determines output values corresponding to each rule; and the fifth layer determines the system interference.

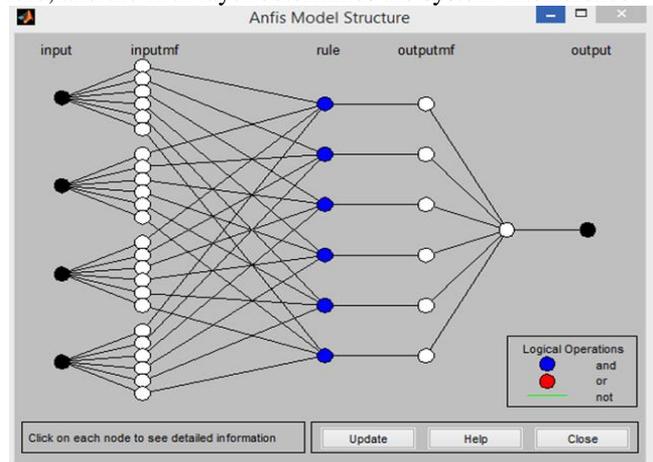


Fig. 7: Structure of the formed system

Figure 8 shows the results of testing the *ANFIS* system with nine pairs of points: the test data and the results obtained by the *ANFIS* system. The average error on the test set was 7.5.

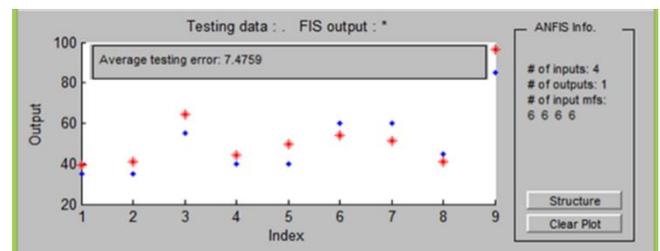


Fig. 8: Testing results for the ANFIS system

The formed fuzzy inference system, which is part of the *ANFIS* system, is shown in Figure 9.

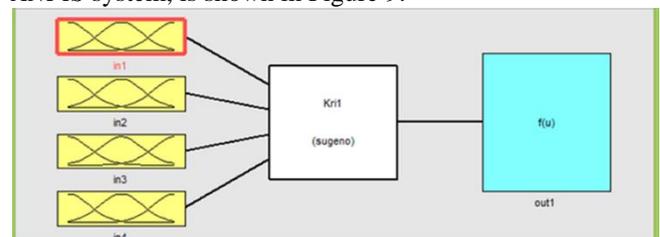


Fig. 9: Fuzzy inference system

In order to switch to *Simulink*, where the system is simulated, this system should be exported to the *MatLab* workspace first, and then the *fuzblock* command should be used to create a model for solving the problem demonstrated in Figure 10.

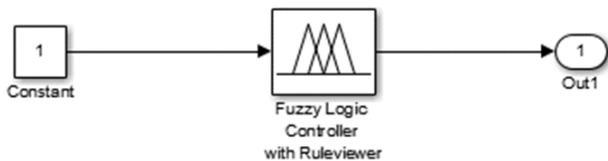


Fig. 10: Model of solving the problem in *Simulink*

Having omitted some intermediate manipulations in *Simulink*, the simulation results in viewing the rules with the input values of the enterprise parameters *Var5*, *Var7*, *Var2*, and *Var8* (Figure 11).

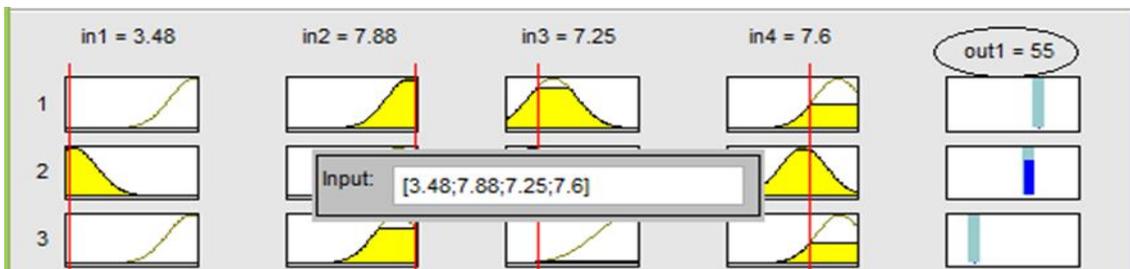


Fig. 11: Result of the system simulation in *Simulink*

A result of the problem solving, i.e., estimate of the market share that an enterprise can capture with specific values of its parameters, is shown above the last column in Figure 11 (circled in oval). Since the selected scale for the output parameter of the system (market share) is a 100-point scale, the resulting estimate of the market share can be interpreted as follows. The lower, zero, border of such a scale is considered as the absence of the given enterprise on the market; while the top 100-point border determines the

dominance of the enterprise in the market.

*Simulink* allows to view the structure of the system. A fragment of this structure, including membership functions for the first input, rule base, and relations between different blocks, is presented in Figure 12. The *Simulink* software allows to analyze the system further, considering each block of the system, its connections, inputs and outputs, but this is beyond the scope of this article.

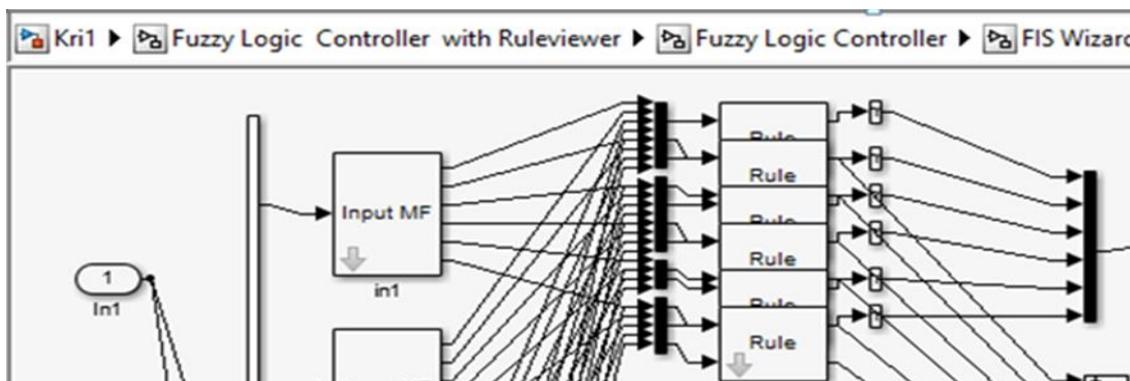


Fig. 12: Fragment of the system structure

## IV. DISCUSSION

The results allow to choose an enterprise strategy using the NN technology, estimate the market share occupied by an enterprise using the NFS, and simulate the behavior of a fuzzy interference system using the *Simulink* software. Solving the problem of choosing strategies with a reduced number of features by using only the main components is of interest in this case. Besides, further analysis of the fuzzy inference system analysis using *Simulink* tools is possible.

## V. CONCLUSION

As such, it is demonstrated in the article that the problem of the strategic position of an enterprise, including the choice of a specific strategy and estimation of the market share, can be solved using technologies based on NNs and FL. These methods are part of ML, which, in turn, is part of AI.

## REFERENCES

1. I. Ansoff, "Strategic management": Classic edition, NY, Palgrave Macmillan, 2007, pp: 272 .
2. H. Mintzberg, B. Ahlstrand, J. Lampel, "Strategy safari: a guided tour through the wilds of strategic management", NY: The Free Press, 1998, pp. 406.
3. M. Porter, "Competitive Strategy: Techniques for Analyzing Industries and Competitors", M.: Alpina Publisher, 2019, pp. 454.
4. J. Kotter, "Accelerate: Building Strategic Agility for a Faster-Moving World", Harvard Business Review Press, 2014, pp. 224
5. P. Kim, "MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence", Seoul, Soul-t'ukpyolsi, Korea, 2017, pp. 162 .
6. H. Brink, J. Richards, M. Fetherolf, "Real-World Machine Learning", Shelter Island, Manning, 2017, pp. 427.

7. S. Haykin, "Neural Networks and Learning Machines", NY, Pearson Education, 2009, pp. 937.
8. Neural Network Toolbox™. User's Guide. The MathWorks, Inc., MA, 2015, pp. 410.
9. V. Kecman, "Learning and Soft Computing – Support Vector Machines, Neural Networks, and Fuzzy Logic Models", London, MIT Press, 2001, pp. 576.
10. R. Jang, C-T. Sun, E. Mizutani, "Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence", Prentice-Hall, 1997, pp. 640.
11. R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 1993, pp. 665-685.