



## Artificial Neural Network Model for Prediction of Students' Success in Learning Programming

Nebojša Ljubomir Stanković<sup>1\*</sup>, Marija Dragovan Blagojević<sup>1</sup>, Miloš Željko Papić<sup>1</sup> and Dijana Ivan Karuović<sup>2</sup>

<sup>1</sup>University of Kragujevac, Faculty of Technical Sciences, Čačak 32000, Serbia

<sup>2</sup>University of Novi Sad, Technical Faculty, Zrenjanin 23101, Serbia

\*Author for Correspondence E-mail: [nebojsa.stankovic@ftn.kg.ac.rs](mailto:nebojsa.stankovic@ftn.kg.ac.rs)

*Received 07 September 2020; revised 23 December 2020; accepted 09 February 2021*

The model for predicting students' success in acquiring programming knowledge and skills is presented in this paper. In order to collect the data needed for development of the model, 159 undergraduate IT students from Faculty of Technical Sciences in Čačak were analyzed. Besides the score on programming knowledge test, the following data were also gathered for each student: high school, the subject he/she took at the entrance exam, size of student's birthplace, average high school grade, points from high school, gender, previous education, existence of IT educational profile in high school, study year, percentage of attendance on classes, reason for enrolment, subjective assessment of preparedness for programming, solving sequential tasks, type of programming student prefers, subjective assessment of preparedness for working in industry, solving tasks with branching and cycle, solving complex tasks, knowledge level, formal education, informal education, Kolb's learning style. In order to predict students' success in learning programming multilayer perceptron was used with backpropagation learning algorithm. The cross-validation methodology was used for the training and testing of the classifiers. Transformation process is performed on the points students achieved on the test in order to get three categories related to success. Based on the results about the relevance of the parameters, the model reached an accuracy of 92.3%. In order to facilitate the use of the model, a Web-based application for displaying the results was created. It is primarily intended for teachers with no experience in working with neural networks, who can use it for planning the teaching.

**Keywords:** ANN, Knowledge acquisition, Knowledge test, Programming skills, Web-based application

### Introduction

The digital world of today demands programming competences. Programming has become an essential tool and crucial skill in many spheres of human activity. Thus, there is a growing need for prediction of behaviour patterns and future achievement of students in programming, especially ones attending IT study programs. The existing solutions and models for predicting students' achievement are not sufficiently adapted to the specificities of IT, either regarding the whole field, or its different subfields (e.g. programming, computer networks, data security, network security, data bases, etc.). Moreover, the end users who do not possess specific IT competences related to artificial neural networks should also be able to determine the values of the input parameters and obtain the result for the predicted parameter. The subject of this research is the application of artificial neural networks for predicting the achievement level

in the field of programming, as well as development of a Web based application for working with the proposed model of artificial neural networks. The research goals include: 1) creation and evaluation of the neural network model for predicting the success in learning programming; 2) creation of a Web based application for working with the created model of artificial neural networks. General hypothesis: It is possible to develop an artificial neural network model in order to accurately predict students' success in programming, taking into account specific characteristics of its learning. The evaluation measures of the model's accuracy are in accordance with the results of real-time validation. The practical goal of the paper refers to the adaptation of teaching process according to the predicted values of the parameter "success in learning programming". A great deal of related research is concerned with the analysis of success in programming as well as the use of neural networks in the process. Bahadir<sup>1</sup> uses artificial neural networks and regression analysis to predict the academic success of prospective teachers

\*Author for Correspondence  
E-mail: [nebojsa.stankovic@ftn.kg.ac.rs](mailto:nebojsa.stankovic@ftn.kg.ac.rs)

of mathematics. Data mining techniques are often compared to traditional techniques in order to identify possibilities in predicting success. Such an approach was used by Hardgrave *et al.*<sup>2</sup> That study concluded that artificial neural networks provide satisfactory results in prediction of success. In addition to comparing with traditional techniques, the authors also compare different data mining techniques in order to find the most accurate model for success prediction. Ibrahim and Rusli<sup>3</sup> compared artificial neural networks, decision tree and linear regression and found that the best result was obtained by artificial neural networks. Oladokun *et al.*<sup>4</sup> used artificial neural networks in order to predict the success of students applying to university. The starting point for this research was the fact that graduates from some Nigerian universities do not possess satisfactory knowledge and skills. Because of the limitations of classic statistical techniques, artificial neural networks were used by Lau<sup>5</sup> to predict students' performance. That model was created with 11 input parameters and two hidden layers. This paper also confirmed satisfactory accuracy of neural network models.

### Materials and Methods

In order to achieve the goals, the artificial neural network is used, together with the methodology related to this technique. The methodology includes the following specific research tasks which are related to data mining process: data collection, pre-processing and transformation.

#### Data Collection, Pre-processing and Transformation

For the purpose of collecting, processing and analyzing data and presenting the results on which the development of the model is based, the following research techniques will be used: 1) for data collection (most instruments were developed by the authors):

- a) for primary data: questionnaires, knowledge tests on programming, Kolb's Learning Style Inventory, self-assessment scales;
- b) for secondary data: data on students' performance from internal records in higher education institution and external data on programs of high school subjects within the IT field (which is publicly available data of the Ministry of Education, Science and Technological Development).

The sample consists of 159 students of four-year study program Information Technologies at the

Faculty of Technical Sciences Čačak, University of Kragujevac. The number of students who participated in the research is shown in Table 1, including year of study and gender.

Two generations of students (2017 and 2018) were used for the first year of the research. The 2018 generation students represented a sample that was used for real-time validation. The data was collected at the Faculty of Technical Sciences, in the student services' database and in direct contact with students. All relevant data were collected: high school, city of previous education, number of points from high school and subject chosen for the entrance exam. The data pre-processing is a necessary step in data mining process. Pre-processing is a phase that involves removing the entries which contain errors. In this phase, anything that is not relevant to this specific research is considered an error and it is removed. For example, the database contains the names of students, which is irrelevant for the research. There were no empty cells so the problem with missing data did not exist. The collected data are in the appropriate form for the research and majority of parameters do not demand further transformation. In the research transformation is performed on the points students achieved on the test. Students had a test with questions and tasks from programming. All the answers were recorded in a database. For the purpose of the research composite measure of success was made. It included: the test results, percent of passed subjects related to programming and average grade in these subjects.

At the beginning levels of success in programming were defined as:

- 1 – very successful students (achievement >70%)
- 2 – students with average (>40% achievement  $\leq$ 70%)
- 3 – students with small success or unsuccessful (achievement  $\leq$ 40%)

#### Neural Network Model Creation and Training

A so-called multilayer perceptron is used in this research which can be trained by many algorithms.

Table 1 — Number of students by gender and year of study

Year of study	Female	Male	Total
1	14	15	29
1	5	29	34
2	16	18	34
3	9	23	32
4	7	23	30
Total	51	108	159

The neural network algorithm is used to create a network that can contain three layers of neurons: an input layer, a hidden layer (which is optional), and an output layer. Neural network model is presented in Fig 1. The input layer contains the following parameters: high school, the subject he/she took at the entrance exam, size of student's birthplace, average high school grade, points from high school, gender, previous education, existence of IT educational profile in high school, study year, percentage of attendance on classes, reason for enrolment, subjective assessment of preparedness for programming, solving sequential tasks, solving tasks with branching and cycle, solving complex tasks, type of programming student prefers, subjective assessment of preparedness for working in industry, knowledge level, formal education, informal education, Kolb's learning style. The back-propagation training algorithm was applied. Weights are set for individual elements of initialized neural network. They are randomly assigned after which the optimisation process starts. The weights are applied to the activation function in order to determine each neuron's output. An error function is defined and the function captures the delta between the correct output and the actual output of the model (given the current model weights). The main objective is discovering the weights that can generate the most accurate output.

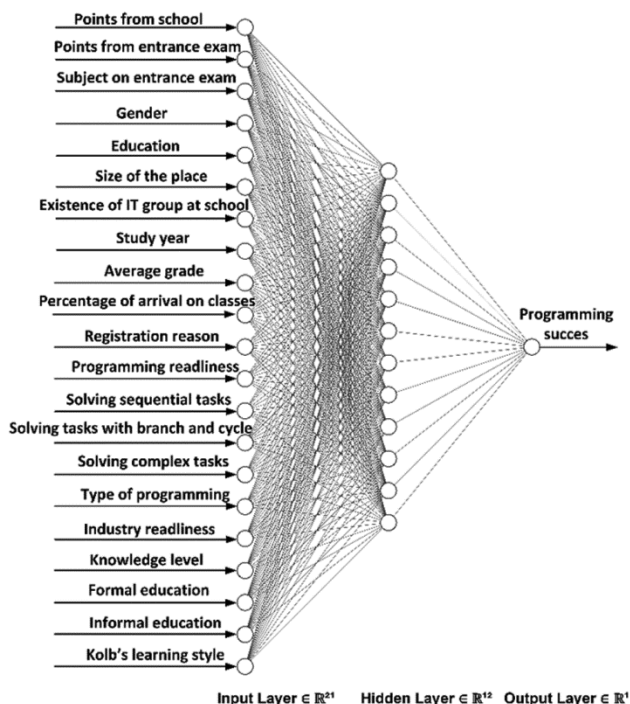


Fig. 1 — Artificial neural network model<sup>6</sup>

### Neural Network Evaluation

As mentioned earlier, neural network model contains three layers (Fig 1):

- Input layer (standardized term, 34.02.07 in ISO/IEC 2382–34:1999, 1999);
- Hidden layer;
- Output layer (34.02.08 in ISO/IEC 2382–34:1999, 1999).

Neurons in the hidden layer use a sigmoid function:  $(f(x) = \frac{1}{1+e^x})$  which converts the input in the interval  $(-\infty, +\infty)$  to interval  $(0, 1)$ . Output neuron represents the attribute values that were predicted. In this study, the output refers to the number of students. Neurons in the output layer use linear activation function. The back-propagation algorithm was used to train the neural network. 30% of data for testing and 70% of data for training the neural network are used for model evaluation in this research. Similar approach is presented in<sup>7-9</sup> but in other fields. Besides that, the root mean square error is calculated according to equation (1).<sup>(10)</sup> Confusion matrix is presented in Table 4.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (t_i - o_i)^2}$$

where  $t_i$  refers to the calculated output given by the network,  $o_i$  stands for the real output for case  $i$ , and  $n$  is the number of cases in the sample. The model is useful when RMSE is lower than 1. Lower RMSE is a sign of a model which is more accurate than unintelligent predictor. RMSE presents the relation between total error of the created model and unintelligent predictor (which always predicts mean value of the output). In order to get the satisfying performance, the occurrence of two main problems (overfitting and underfitting) should be reduced. When machine learning model cannot capture the underlying trend of the data underfitting occurs. Overfitting occurs when machine learning model tries to detect all the data points. The overfitting is the often problem in supervised learning. There are several ways to avoid this issue, like cross-validation, training with more data, removing features, early stopping the training, regularization and ensembling. In order to avoid overfitting in this research, cross-validation was used. The cross-validation methodology was used for the training and testing of the classifiers. The data set is randomly divided into a set of  $K$  distinct sets. Training is carried on  $K-1$  sets and the remaining set is tested. The process is

repeated for all of the possible K training and test sets. The average of all K results represents the classification results.

**Results and Discussion**

This section presents the research results presented within Weka program.<sup>11</sup> An artificial neural network with the following parameters is selected:

- Learning rate: 0,3
- Momentum: 0,2

The results have shown that the prediction accuracy of the model is exactly 77, 69% (Table 2).

Accuracy measures are presented for each class separately (Table 3).

Moreover, the Confusion Matrix displays accurately and inaccurately classified cases by classes (Table 4).

In addition to the performed analysis, an evaluation of the attributes was done in order to optimize the neural network model and increase its accuracy. All eleven criteria were used in order to select attributes in Weka. The best result was obtained using Relief Attribute Eval.

It evaluates the value of an attribute by repeatedly sampling an event and considering the value of the famous attribute for the nearest sample of the same and different class.

Table 2 — ANN model’s accuracy

Correctly Classified Instances	101	77.6923 %
Incorrectly Classified Instances	29	22.3077 %
Kappa statistic		0.6603
Mean absolute error		0.1521
Root mean squared error		0.3553
Relative absolute error		34.605 %
Root relative squared error		75.7955 %
Total Number of Instances	130	

Table 3 — Detailed Accuracy By Class

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.659	0.163	0.674	0.659	0.667	0.499	0.755	0.680	average success
	0.800	0.053	0.848	0.800	0.824	0.762	0.956	0.876	very successful
	0.863	0.127	0.815	0.863	0.838	0.729	0.916	0.849	unsuccessful
Weighted Avg.	0.777	0.119	0.776	0.777	0.776	0.660	0.872	0.799	

Table 7 — Detailed Accuracy By Class

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.909	0.047	0.909	0.909	0.909	0.863	0.929	0.949	average success
	1.000	0.042	0.897	1.000	0.946	0.927	0.991	0.973	very successful
	0.882	0.025	0.957	0.882	0.918	0.871	0.930	0.890	unsuccessful
Weighted Avg.	0.923	0.037	0.925	0.923	0.923	0.883	0.946	0.932	

With the mentioned algorithm for attribute selection, the following parameters with a rank above 0,5 were selected as input parameters (Table 5). In this way, the neural network was optimized with the following inputs:

After the creation of a new model of neuron networks with the mentioned six input parameters, the following accuracy result was obtained (Table 6, Table 7 and Table 8):

Table 4 — Confusion Matrix

a	b	c	Average success
29	5	10	a = average success
7	28	0	b = very successful
7	0	24	c = unsuccessful

Table 5 — Display of the top 6 parameters

Rank	Input parameters
0.1093	Average grade
0.09591	Gender
0.06862	Solving sequential tasks
0.06339	Students’ subjective assessment of whether they are ready to work in the field of programming
0.06169	Size of the town in which students finished highschool
0.05719	Kolb

Table 6 — Accuracy after variable selection

Correctly Classified Instances	121	92.3077 %
Incorrectly Classified Instances	9	7.6923 %
Kappa statistic		0.8839
Mean absolute error		0.0667
Root mean squared error		0.227
Relative absolute error		15.1676 %
Root relative squared error		48.426 %
Total Number of Instances	130	

Table 8 — Confusion Matrix

a	b	c	Average success
40	2	2	a = average success
0	35	0	b = very successful
4	2	45	c = unsuccessful

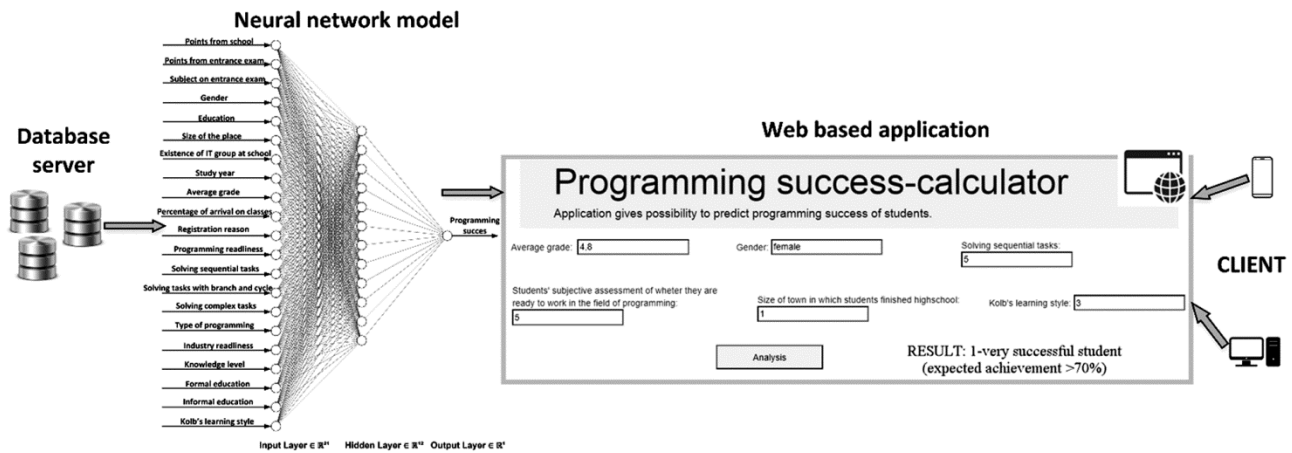


Fig. 3 — Architecture of proposed system

The screenshot shows the **Programming success-calculator** web application interface. It features a title bar and a subtitle: "Application gives possibility to predict programming success of students." Below this, there are several input fields: "Average grade: 4.8", "Gender: female", "Solving sequential tasks: 5", "Students' subjective assessment of whether they are ready to work in the field of programming: 5", "Size of town in which students finished highschool: 1", and "Kolb's learning style: 3". An **Analysis** button is positioned below the inputs. The result displayed is **RESULT: 1-very successful student (expected achievement >70%)**.

Fig. 4 — Application's graphical user interface

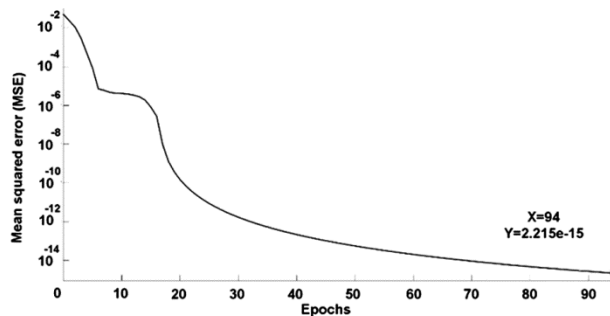


Fig. 2 — Convergence curve of neural network

As it can be seen from the obtained results, the accuracy was increased to 92.30%. Comparison of the results with the related research shows that the models used in this field have satisfactory accuracy in general. In other words, with adequately used data mining methods, this field is suitable for analysis and prediction. The convergence curve for the summed squared error between the displacements from numerical simulation and those from backpropagation network. Thus, it is not too difficult to fall into local convergence during the process of training.

In order to assure providing results to teachers, the web based application was developed. Its architecture is presented in Fig 3.

After entering the parameters through application's interface (Fig 4), the end user gets the results related to the category for the chosen student. One test parameter for new student along with expected achievement is shown as an example in Fig 4.

The main advantages of proposed application are:

- its simplicity and
- easy adaptation to other educational questions.

## Conclusions

Taking into consideration the methodology presented and the obtained results, several conclusions can be drawn:

- Years of experience in working with students in the field of programming have resulted in an original artificial neural network model with satisfactory prediction accuracy;
- Possibilities of applying artificial neural networks in relation to the proposed model and accuracy of prediction make it possible to identify potentially successful and unsuccessful students which

further influences the course planning and individual approach to extremely successful and unsuccessful students in order to meet their specific learning needs.

Advantages of the proposed solution are related to the availability of the results from neural networks to people without specific IT skills through the use of developed Web based application. Its simplicity and friendly user interface allow users to use advanced techniques and consequently adapt their teaching according to the results. Limitations of the study are related to the size of the sample – larger sample would result in smaller error of prediction model and more precise results. This limitation could easily be resolved with a greater number of participants in future research. Three classes of success in this model could also be observed as a limitation but it should be noted that more classes (eg. 6 classes which depict students grades) would diminish precision of predictions.

### Acknowledgment

This study was supported by the Ministry of Education, Science and Technological development of the Republic of Serbia, and these results are parts of the Grant No. 451–03–68/2020–14/200132 with University of Kragujevac – Faculty of Technical Sciences Čačak.

### References

- 1 Bahadır E, Using neural network and logistic regression analysis to predict prospective mathematics teachers' academic success upon entering graduate education, *Educ Pract Theory*, **16(3)** (2016) 943–964.
- 2 Hardgrave C B, Wilson L R & Walstrom K A, Predicting graduate student success: A comparison of neural networks and traditional techniques, *Comput Oper Res*, **21(3)** (1994) 249–263.
- 3 Ibrahim Z & Rusli D, Predicting students' academic performance: comparing artificial neural network, decision tree and linear regression, *21<sup>st</sup> Annual SAS Malaysia Forum*, Shangri-La Hotel, Kuala Lumpur (2007).
- 4 Oladokun O V, Adebajo T A & Charles-Owaba O E, Predicting Students academic performance using artificial neural network: a case study of an engineering course, *Pac J Sci Technol*, **9(1)** (2008) 72–79.
- 5 Lau E T, Sun L & Yang Q, Modelling, prediction and classification of student academic performance using artificial neural networks, *SN Appl Sci*, **1** 982 (2019), <https://doi.org/10.1007/s42452-019-0884-7>
- 6 Stanković N & Blagojević M, Artificial neural networks in predicting student performance in programming, *Proc. Development trends Innovations in contemporary education* (Kopaonik, Serbia) 2020, 371–375.
- 7 Blagojević M, Papić M, Vujičić M & Šučurović M, Artificial neural network model for predicting air pollution. Case study of the Moravica district, Serbia, *Environ prot eng*, **44 (1)** (2018) 129–139.
- 8 Jovanović Ž, Blagojević M, Peulić A & Janković D, Patient comfort level prediction during transport using artificial neural network, *Turk J Elec Eng & Comp Sci*, **27(4)** (2019) 2817–2832.
- 9 Blagojević M, Blagojević M & Ličina V, Web-based intelligent system for predicting apricot yields using artificial neural networks, *Sci Hortic*, **213** (2016) 125–131.
- 10 Draper C, Reichle R, Jeu R, Naemi V, Parinussa R & Wagner W, Estimating root mean square errors in remotely sensed soil moisture over continental scale domains, *Remote Sens Environ*, **137** (2013) 288–298.
- 11 See Weka software, <https://www.cs.waikato.ac.nz/ml/weka/> (15 April 2020)