

Prediction of students' performance based on heterogeneous and ambiguous data

Goran P. Šimić^a, Ana Miletić^b, Jelena Mitić^c

^a Academy of Technical and Art Applied Studies, School of Electrical and Computer Engineering, Belgrade, Republic of Serbia, e-mail: goran.simic@viser.edu.rs, **corresponding author**, ORCID iD: <https://orcid.org/0000-0002-7563-699X>

^b Academy of Technical and Art Applied Studies, School of Electrical and Computer Engineering, Belgrade, Republic of Serbia, e-mail: ana.miletic@viser.edu.rs, ORCID iD: <https://orcid.org/0009-0001-7586-057X>

^c Academy of Technical and Art Applied Studies, School of Electrical and Computer Engineering, Belgrade, Republic of Serbia, e-mail: jelena.mitic@viser.edu.rs, ORCID iD: <https://orcid.org/0009-0000-0015-0840>

 <https://doi.org/10.5937/vojtehg74-58137>

FIELD: computer sciences

ARTICLE TYPE: original scientific paper

Abstract:

Introduction/purpose: The presented research describes the application of artificial neural networks for prediction of students' performance based on heterogeneous and ambiguous data.

Methods: The students' data collected during three school years on subject "Object-Oriented Design" at the School of Electrical and Computer Engineering, Academy of Technical and Art Applied Studies in Belgrade represent the basis of the research. The sample size is one hundred fifty-nine students. The initial dataset consists of thirty-six different features from three different sources of collected data. The presented data and results are completely anonymized. The research describes the complexity of data preparation, as well as the processes of modeling, training, testing and evaluation of an artificial neural network, applied for the prediction of students' performance.

Results: The main results of the research are the handling the ambiguity and heterogeneity of data and the design of the prediction model. In addition, the research includes discussions about data preprocessing and machine learning techniques applied on prediction models.

Conclusion: The development methods, prediction models and prediction results, their critical review, the limitations of the research, and the possibilities for improvements represent the main conclusions of the research.

Key words: artificial neural networks, students' performance prediction, Web service architecture, data analysis.

Introduction

The presented research describes the application of artificial neural networks (ANN) for the prediction of students' performance based on heterogeneous and ambiguous data. The students' data collected during three school years on the subject Object-Oriented Design (OOD) at the School of Electrical and Computer Engineering (Academy of Technical and Art Applied Studies in Belgrade) represent the basis of the research. The sample size is one hundred and fifty-nine students. The research includes thirty-six different features from three different sources of collected data.

Contemporary educational management requires fast responses to questions related to students' enrollments and academic progression by year. On the other hand, the data sources necessary for this purpose are often distributed and stored in different formats. For instance, the student service collects historical data from the students' previous education (e.g. annual results, finished schools, etc.) and final examination results. In contrast, intermediate results and information about students' educational activities are commonly stored in e-learning systems and maintained by teachers and assistants. Moreover, data from different sources are processed manually (sometimes separately), which provides unsatisfactory results. It is obvious that improvements are necessary, and the mentioned facts provide the primary motivation for the development of the prediction model presented in the manuscript.

The next section gives a review of research related to the topic, as well as the arguments for the selection of an appropriate machine learning (ML) model for the prediction of students' performance. The third section describes the new method used for designing prediction models. The complex nature of the data requires a combined consideration of data preparation and prediction model design. This iterative process enables improved accuracy of prediction models through modified and redesigned datasets. The fourth section presents the results of the research. For utilizing the model and proving the feasibility of the solution, *Python* Web service was developed. This *early-stage* module enables users to conduct experiments and use the results as support for decision-making. The section also presents the results of applying the proposed model to real and synthesized datasets. Finally, the critical review and the possibilities for improvements are given in the *Conclusion* section.

Literature review

Artificial intelligence (AI) is used in almost all spheres of human life. One of the specific ways of using it is in education. In the last decades, many researchers in this area have tried to incorporate different techniques of AI into existing educational systems (Virtanen et al, 2017; Al-Theeb et al, 2024; Kosareva et al, 2023). More precisely, they have tried to improve e-learning systems to make them adaptive in accordance with the educational needs of individual students (Essa & Human-Hendricks, 2023). Such systems facilitate personalized learning through adaptation of content and navigation based on an actual student profile (Tsai et al, 2019).

In addition, there are e-learning systems designed for learning specific content (Mitrovic, 2016; Johns, 2022). These systems, commonly named Intelligent Tutoring Systems (Wang et al, 2023), significantly increase learning productivity due to AI technologies that enable them to learn about students' needs, interests, motivation, habits and behavior. All these data enrich the student model and provide sufficient information for delivering appropriate learning content.

The student model represents a common component for both general-purpose e-learning systems and intelligent tutoring systems. It has been recognized as a significant part of the intelligent behavior of e-learning system (Woolf, 2010). For that reason, many different standards have been designed for user modelling (Böck, 2024).

Over the last decade, there have been intentions to integrate different systems to collect data about students from different sources and to use the data for obtaining a better student profile and extending the information about their competences (Ocheja et al, 2018; Labba et al, 2020). Moreover, new standards have been developed for this purpose. The most cited ones are xAPI (Manso-Vázquez et al, 2018) and CMI5 (Bakhouyi et al, 2022). The former provides the collection of more detailed information about students' activities during their interactions with e-learning systems. The latter enables the exchange of this information among systems.

There are already developed and well-formed process models for data mining and data preparation for further processing in AI models. One of the frequently mentioned models is CRISP-DM (Dwivedi & Rawat, 2015). It is the *de facto* industrial standard that proposes a process framework for data mining from any kind of data source. However, in the educational process, there is a variety of information that e-learning systems, as well as teachers, collect about students. Although the standards mentioned above enable automatization of information

exchange, teachers usually use their individual forms and formats for collecting student results and information about their activities and progress during the learning process. For that reason, the data preparation method presented in the research has a more detailed design than proposed by the standards found in the literature.

The selection of models for students' profile classification and results prediction represents another challenge in contemporary educational systems. The questions *How many students would pass from one school year to the next?* and *What are the predictions about students' results?* are commonly asked by the institutional management, who need this information for planning enrollment quotas and selecting the method of scholarship distribution. For a similar purpose, Benkhalfallah & Laouar (2023) propose linear and polynomial regression models, as well as models based on tree algorithms (e.g. random forest, tree ensemble, gradient boosted trees, and simple regression trees), emphasizing the high performance obtained. On the other hand, a review paper by Rastrollo-Guerrero, Gómez-Pulido & Durán-Domínguez (2020) focused on the general use of different machine learning (ML) techniques and algorithms highlights ANN and support vector machine (SVM) models for students' classification and results prediction. Another research that compares different ML models for students' classification (Ahammad et al, 2021) favors the feed-forward multilayer perceptron (hereafter MLP) as the basic ANN classification model compared with Naive Bayes, K-nearest neighbors, SVM and Gradient Boost Tree models. The presented results provide strong support for choosing ANN as a classification model.

Methods

For building reliable prediction models, it is important to consider available data and potential prediction models together. The resulting model should depend on the size, complexity and variance of the data. Further, the complexity of the data depends on the number of data sources as well as the number of features and their mutual correlation. On the other hand, there is a variety of models suitable for prediction purposes. The selection of appropriate models cannot be performed without experimenting with the data collected and prepared for prediction purposes. The method used in the research treats data preparation together with the design of the prediction model (Figure 1). A UML activity model (Ahmad et al, 2019) is used as a template for the design. The data preparation activities (presented in blue) are in accordance with the CRISP-DM process model (Schröer et al, 2021), while the activities of the

prediction model design (presented in yellow) are used to support the development and evaluation of the prediction model.

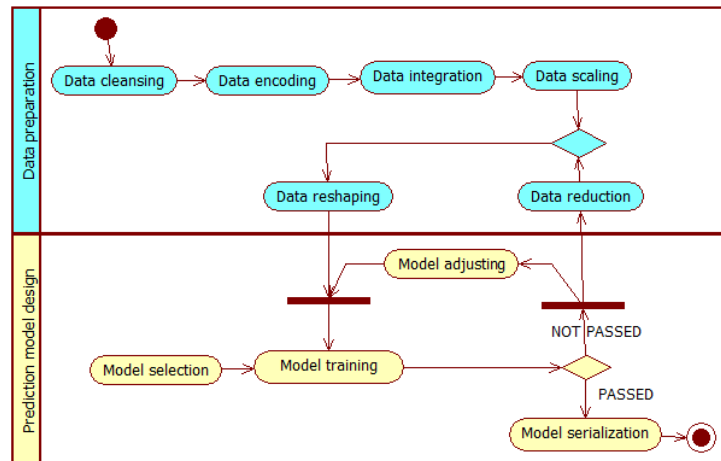


Figure 1 – Method used for data preparation and prediction model design

Data preparation activities

Firstly, the data preparation includes sequential stages: cleansing, encoding, integration and scaling. Data cleansing removes records from datasets that contain unreliable or incomplete data. For instance, the records of students that have paused or discontinued their studies are irrelevant and are therefore excluded from the dataset.

brind	ime	prezime	prisustvo	testovi	kol1	kol2	projekat
ELITE-2020	Душан	Милошевић	7	7	17	24	20
ИС-1405	Томислав	Милошевић	9	8.2	0	0	22
ИС-1405	Јован	Милошевић	10	8	18	26	20
ИС-1405	Милош	Милошевић	10	9.4	0	0	0
ИС-1405	Душан	Милошевић	9	7.6	18	27	21
ИС-1405	Милош	Милошевић	9	7.6	18	27	21

Teacher's records

Records from faculty information system

brind	ime	prezime	prosek_1g	prosek_2g	prosek_3g	prosek_4g	prijemni_bd	srednja_skola
ELITE-2020	Душан	Милошевић	3.8	3.73	3.92	4.69	49	Смедерево; Гимназија
ИС-1405	Томислав	Милошевић	5	5	5	5	48	Лаковац; 17. септембар; Средња
ИС-1405	Јован	Милошевић	4.87	4.81	4.93	5	48	Београд; Шеста београдска гимназија
ИС-1405	Милош	Милошевић	3.87	3.8	3.77	4	57	Београд; Пета београдска гимназија
ИС-1405	Душан	Милошевић	4	4.08	4.71	4.14	52	Београд; Раде Кончар; Електротехника
ИС-1405	Милош	Милошевић	3.92	3.91	4.47	4.58	54	Смедерево; Техничка школа

Records from e-learning system

d	datumvreme	korisnik	po...	kontekstdogadjaja	komp...	nazivdogadjaja	opis
341,287	28/05/2022 21:14	Душа...	-	Датотека: Projektni zadatak 202...	Дато...	Модул курса приказан	The user with id '4803' viewed the 'resource' act
341,288	28/05/2022 21:13	Душа...	-	Курс: Објектно оријентисано про...	Сист...	Курс приказан	The user with id '4803' viewed the course with id
341,313	28/05/2022 19:03	Душа...	Д...	Датотека: osobe	Сист...	Завршетак активности ...	The user with id '4803' updated the completion s
341,314	28/05/2022 19:03	Душа...	Д...	Датотека: osobe	Сист...	Завршетак активности ...	The user with id '4803' updated the completion s
341,315	28/05/2022 19:03	Душа...	-	Датотека: osobe	Дато...	Модул курса приказан	The user with id '4803' viewed the 'resource' act
341,316	28/05/2022 18:59	Душа...	-	Датотека: Vežba 9 - Rad sa dato...	Дато...	Модул курса приказан	The user with id '4803' viewed the 'resource' act

Figure 2 – Samples of students' data from different sources

Data encoding provides a common format for all data. For instance, there are names of students, schools, exam terms' names, as well as

descriptive results originally recorded in text format. The categories for prediction are also in textual format. Therefore, encoding them into numerical format is necessary for further processing. The target categories defined by management (with meanings adapted to the needs of decision-making) are encoded by integers. In this way, the categories (*critical*, *less_critical*, *in_the_middle*, *scholarship_candidate* and *best_candidate*) are encoded with number values (0,1, 2, 3 and 4).

Data integration provides the fusion of data from different sources. The raw data used in the research consists of thirty-six different features collected from three different sources: teachers' spreadsheets, e-learning platform and faculty information system (Figure 2). The teachers' spreadsheets contain information about students' attendance and results (e.g. from pretests, colloquiums, essays and projects). The data collected from the e-learning platform consists of records with a uniform structure. Each record contains information about the user, activity type, context of the activity, event description, and date and time of the student's interaction with the e-learning system. The information from the faculty information system consists of the name of the secondary school the student finished, the student's results from four years of high school, the entering exam results (before enrollment) and the subject's final exam results.

Table 1 – 36 features collected from three different sources

raw_id	average_HS1	file_downloads
student_id	average_HS2	access_url
first_name	average_HS3	exercise1
surname	average_HS4	exercise2
attendance	entering_exam	exercise3
pretest	high_school_id	exercise4
colloquium1	high_school_name	exercise5
colloquium1	final_exam_term	exercise6
project	course_attendance	exercise7
activity	forum_attendance	exercise8
num_of_final_exam_tries	task_attendance	exercise9
school_year	test_attendance	exercise10

The whole set of the collected features is presented in Table 1. Some features are eliminated as they hold irrelevant information: *row id*, *student id*, *first name*, *surname* and *URL address*. Other features are mostly

numerical, except for *high school name* and *final exam term* (e.g. *January term* or *June term*). As high schools are represented with an additional identifier (*high school id*), the feature named *high school name* is unnecessary, and it is also eliminated. Further, through (exploratory) analysis, three features with the information redundancy are found – *number of final exams tries* on the one hand, and *school year* and *final exam term* (mentioned above) on the other. The feature *number of final exams* is left in the candidate set, while *school year* and *final exam term* are eliminated.

After elimination of unnecessary features, there are still 28 features left in the candidate set. As the number of features is significantly high for an ANN classifier (and for most other classification models generally), the principal component analysis (PCA) is performed on the remaining feature set. The following diagram shows the distribution of variance among the components (Figure 3). Based on the Kaiser-Guttman rule (Kaiser, 1960) presented by *eigenvalues* (Frost, 2022), the most variance in the dataset is captured by six principal components (*eigenvalues* > 1). The diagram shows the components in descending order to emphasize the recommended number of candidates for further processing.

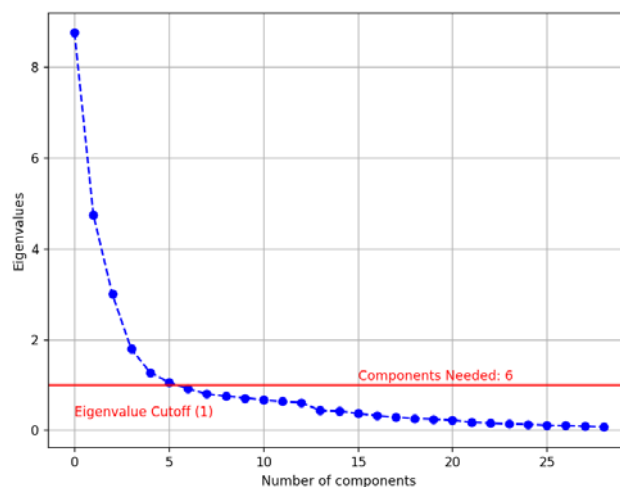


Figure 3 – PCA Eigen-values distribution

On the other hand, the cumulative variance plot (Figure 4) suggests 12 components for building a consistent prediction model in the case of 90% variance threshold. The suggested values should be between 80% and 95%, depending on the domain of use. Empirically, if N represents the full number of features, their optimal number for ML tasks (within the

limited dataset) should be between $\sqrt{N}$ and $2\sqrt{N}$ (Yongmin et al, 2025). In the concrete case, if the total number of features is 36, then the optimal number of features included in ML can vary from 6 to 12.

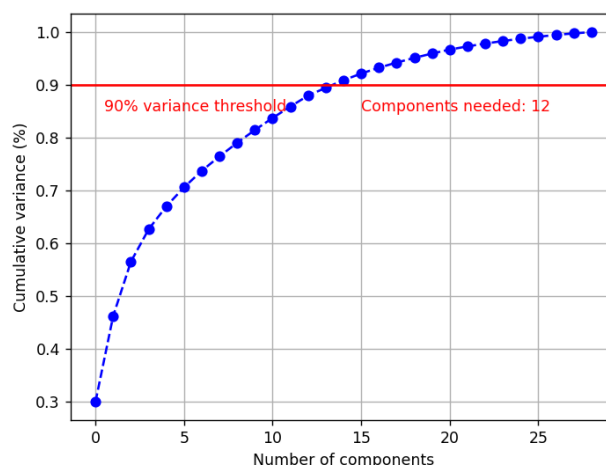


Figure 4 – PCA cumulative variance of components

To select the exact features for further processing, a supervised PCA that uses ANOVA F-value as a score function is performed. Seven subsets of features that represent the supervised PCA result are presented in three columns in Table 2. The parameter k defines the number of features to be selected, and k -selected features are on the right-hand side. For clarity, the subsets are presented in different colors. The most informative features (*attendance*, *pretest*, *colloquium1*, *colloquium2*, *project* and *num_of_final_exam_tries*) are common to all subsets and they represent the content of $k=6$ subset. Other subsets are extended by one to six other features, as shown in the table.

Data scaling is necessary as different features have data in different ranges. The contemporary machine learning models expect standardized input datasets. Without scaling, the joint processing of different features' data will not be reliable. Moreover, the model can favor the features with greater number values and wider data ranges. For instance, students produce hundreds and thousands of records during interactions with the e-learning system. On the other hand, there are features encoded with small integer values from narrow intervals (for instance, descriptive students' ratings encoded by numbers from zero to four, as mentioned above). Therefore, data scaling transforms the features' data to be in the same ranges. This way, the datasets become more comparable and further processing becomes more reliable for prediction purposes. Since

property values vary within expected ranges, *StandardScaler* is used for this purpose (*sklearn.preprocessing* module).

Table 2 – Results of features selection by supervised PCA with ANOVA F-value as a score

k	features	k	features	k	features
6	attendance	9	attendance	11	attendance
	pretest		pretest		pretest
	colloquium1		colloquium1		colloquium1
	colloquium1		colloquium1		colloquium1
	project		project		project
	num_of_final_exam_tries		num_of_final_exam_tries		num_of_final_exam_tries
7	attendance		average_HS2		average_HS1
	pretest		average_HS3		average_HS2
	colloquium1		entering_exam		average_HS3
	colloquium1	10	attendance		entering_exam
	project		pretest		exercise10
	num_of_final_exam_tries		colloquium1	12	attendance
8	average_HS3		colloquium1		pretest
	attendance		project		colloquium1
	pretest		num_of_final_exam_tries		colloquium1
	colloquium1		average_HS1		project
	colloquium1		average_HS2		num_of_final_exam_tries
	project		average_HS3		average_HS1
	num_of_final_exam_tries		entering_exam		average_HS2
	average_HS3				average_HS3
	entering_exam				average_HS4
					entering_exam
					exercise10

Besides data preprocessing, the proposed method contains two additional activities in the data preparation process. Data reshaping is the first one in the sequence. As machine learning models support different data shapes, this activity adjusts the actual data shape to match the one expected by the model. Experimentation with different models needs reshaping of the input data. Consequently, this activity can be repeated.

Data reduction represents the second additional activity. If the prediction model is not satisfactory, before changing it (e.g. its type,

structure and/or parameters), one of the techniques to improve the model is to change the input dataset. This boils down to the reduction of feature data included in model training (already described in the previous section). As the used dataset consists of 159 records, the completion of missing data is performed instead of removing incomplete records. The *Python* class *KNNImputer* (K-Nearest Neighbors Imputer) was used for this purpose (*sklearn.impute* module). It fills in missing data based on the similarity of neighboring samples in the dataset. *KNNImputer* selects the K nearest neighbors (samples that are most similar to the one to be completed). Further, it uses the mean value of the missing feature to fulfill the incomplete data.

Activities of prediction model design

Model preparation represents the other subprocess of the proposed method. It consists of four activities (the yellow part of Figure 1). The first one is model selection. As the prediction of students' performance based on heterogeneous and ambiguous data represents the research problem, it is essentially a classification problem type. The data complexity (a lot of uncorrelated features and data records from different sources) led to the selection of ANN as a prediction model.

Model training represents the next activity. The ANN model uses prepared datasets for the machine learning process. The training happens in epochs. An epoch consists of forward and backward passes. In the forward pass, ANN model processes the training dataset by propagating it through its structure (the layers). Then it uses the resulting (output) values and expected (test) values for loss calculation. Finally, it uses the loss function values for adjusting ANN parameters (weights of synapses between neurons of the neighboring layers) through the backward pass. If the loss function is sufficiently low and accuracy is sufficiently high, the ANN prediction model is trained and prepared for exploitation. The next activity is model serialization. It includes saving the model byte code in an appropriate file format that can be reloaded and reused.

Otherwise, if the model is not trained at a satisfactory level, this leads to loopback and changes to the model as well as to the data. This way, the next activities are model adjusting and data reduction. The model adjusting implies changes to the model parameters: number of epochs, types of layers' activity functions, number of neurons in the layers and number of layers. Empirically, minor changes are better than major ones. In experimentation, one should select one parameter at a time, slightly changing it and monitoring the consequential changes.

The design and implementation of the prediction model

The design and implementation of the overall system was done by using a Python environment. The *keras* library (<https://keras.io/>) is used for building an ANN prediction model and its serialization, while the *sklearn* library (<https://scikit-learn.org/>) is used for data preparation, and the *pandas* library (<https://pandas.pydata.org/>) is used for loading and formatting data from different sources. The resulting model for prediction is sequential and consists of an input layer of twelve neurons, a hidden layer of eight neurons, and an output layer of five neurons – one for each category/class (Figure 5). Generally, for a small number of features and records in the dataset, an empirical rule of thumb is to keep the number of neurons in the layers as small as possible. *Dense* class (*keras.layers* module) is used for constructing the layers. It provides full connectivity between neurons in adjacent layers. Initially, there were eight neurons in the input layer (same as the number of features). To avoid overfitting, the *Dropout* layer (*keras.layers* module) was added after the input layer. It is suitable for small-sized models. The *Dropout* layer drops out a fraction of neurons from the previous (input) layer in each training iteration. This way, the *Dropout* layer forces the model to learn more from the data patterns of consistent features than from noisy ones. For this reason, the number of neurons in the input layer increased to twelve.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 12)	108
dropout (Dropout)	(None, 12)	0
dense_1 (Dense)	(None, 8)	104
dense_2 (Dense)	(None, 5)	45

Figure 5 – Model of ANN used for prediction

Number of parameters for each layer can be calculated as presented in Equation 1. Each neuron has one parameter for each input neuron and one parameter for its *bias*. For instance, since the input layer consists of 12 neurons, each neuron has eight inputs (for each feature) and one bias. The full number is 108 parameters ($12 * 8 + 12 = 96 + 12 = 108$). This formula is not applicable to the *Dropout* layer as it doesn't have any parameters.

$$N_{params} = N_{neurons} * N_{inputs} + N_{bias} \quad (1)$$

Besides the dropout layer, another technique is implemented for avoiding overfitting problems of the model. This is L2 regularization that adds additional value to the loss function. It is calculated as a sum of squared weights of all synapses in the model. The resulting loss function (Equation 2) consists of two parts: categorical cross entropy (CCE) loss (left-hand side) and L2 regulator (right-hand side). In CCE (Mao et al, 2023), constant N represents the number of samples in the dataset (in the concrete case, it is 159 samples), C stands for the number of categories, factor y_{ij} determines if the i^{th} sample belongs to class j (its values can be only 1, or 0), while p_{ij} represents the probability of prediction that i^{th} sample belongs to class j . As probability values are in the $[0,1]$ interval, the logarithm produces negative values. To make the loss function positive, CCE has the minus prefix. The right-hand side of the equation is the L2 regulator, where λ represents the hyperparameter for regularization strength. Constant P represents the full number of the model's parameters (in the concrete case, it is 257 parameters) and w_k represents the weight (parameter) of k^{th} synapse. The L2 regulator has negligibly small values if the weights are close to zero. In contrast, the L2 penalty value is significant if the model's weights are closer to one. Consequently, the L2 regulator gains the influence of new samples (with smaller weights) on the learning process.

$$F_{loss} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log(\hat{p}_{ij}) + \lambda \sum_{k=1}^P w_k^2 \quad (2)$$

Except for the output layer, hyperbolic tangent (*tanh*) (Equation 3) represents the activation function for all neurons in the model (Apicella et al, 2021). Regarding linear activation functions (e.g. ReLU activation function), *tanh* has a greater sensitivity for smaller input values and a reduced sensitivity for greater input values, producing neuron output in the $[-1, 1]$ interval. Empirically, *tanh* better fits shallow ANN.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3)$$

As the output layer consists of five neurons (one output for one category), the *softmax* function (Ren et al, 2023) represents the activation function for the model's output layer. This function (Equation 4) is the extension of the sigmoid (logistic) function, with the difference in additional parameter K that represents the number of output categories. Basically, sigmoidal functions produce probabilities for input values. Unlike the sigmoid function, the *softmax* function produces the probability distribution over the predicted categories.

$$\text{softmax}(x) = \frac{e^{x_j}}{\sum_{j=1}^K e^{x_j}} \quad (4)$$

The limitation of the sigmoid function is that it fits only for binary classification. On the other hand, the *softmax* function is suitable in the case of more than two categories. The outcome of prediction is a vector of K values that represents the probability of a sample belonging to a particular category. As a single value is expected as the result of the prediction, the *argmax* (*argument maxima*) function is used for this purpose. It simply returns the maximum value(s) from the input dataset. In other words, the predicted category will be the one with the highest probability.

Results

For training and evaluation purposes, the dataset was split into 127 records (80%) for training and 32 records (20%) for evaluation of the training process. The dataset was split by stratification to obtain the same category distribution in the training and evaluation subsets. The target labels (categories) were encoded automatically by using the *LabelEncoder* class (*sklearn.preprocessing* module). It results in an encoded categories vector, which is converted into an encoded categories matrix necessary for CCE loss calculations. For control of the training process, *EarlyStopping* class (*keras* module) was used with the purpose of stopping the training when the validation loss function stagnates (Figure 6).

Although the maximum number of epochs was set to 200, the diagram shows that the training was stopped earlier (after 114 epochs). During these iterations, both training and evaluation losses decreased steadily. The validation loss curve does not vary at all while the training loss curve has minor variations caused by small inconsistencies in the dataset. Through experiments, the obtained training accuracy varies as expected – between 0.59 to 0.65 (due to the small dataset).

The confusion matrix (Figure 7a) of validation subset (32 records) points out satisfactory results obtained for labels 0 (*critical*), 1 (*less_critical*), 3 (*scholarship_candidate*) and 4 (*best_candidate*). The results for label 2 (*in_the_middle*) are not satisfactory, as the model failed to correctly predict this category for any sample in the validation dataset. For analysis, the confusion matrix of the ANN model is compared with the confusion matrices produced by other models: random forest classifier, logistic regression model and support vector machine model. The same pattern was found – the results are satisfactory for each label except for label 2. The conclusion is that the models cannot be well trained for this

category as there is just 14% of category *in_the_middle* samples in the whole dataset. The problem is resolved by using SMOTE (*Synthetic Minority Over-sampling Technique*) that slightly extends the original dataset with synthetic records (Elreedy et al, 2024). The confusion matrix after SMOTE processing of the dataset (Figure 7b) shows improvements in all categories.

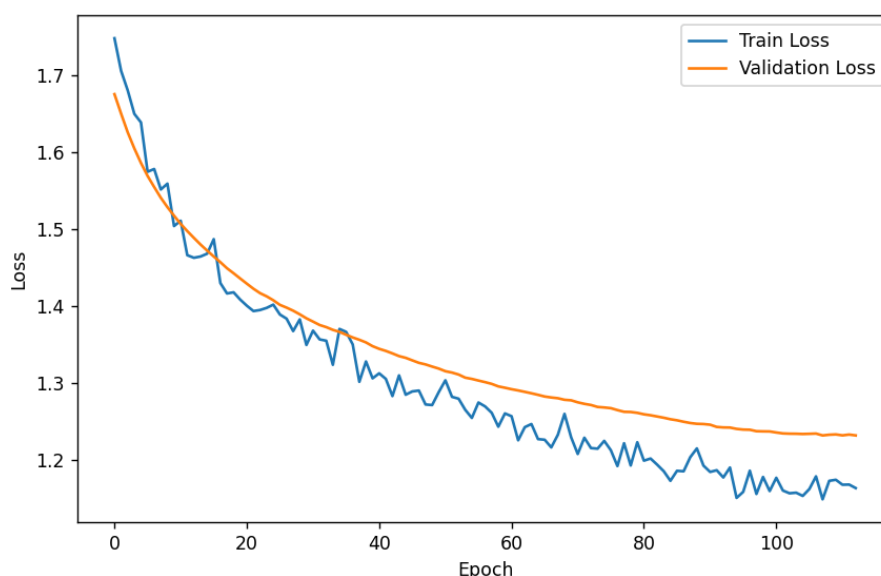


Figure 6 – Loss function descending during the training process

SMOTE technique is designed to reduce the imbalance of samples between categories. The original training set (consisting of 127 samples and category distribution 0: 27, 1: 26, 2: 18, 3: 30 and 4: 26), is extended with 23 new samples (the SMOTE class from *imblearn.over_sampling* module is used for this purpose) and the same category distribution among 150 samples (each category 30 samples). Consequently, the model accuracy is improved (~ 0.67).

Once the model is trained, it should be saved in a format appropriate for further use. Serialization into *keras* binary format (https://keras.io/api/models/model_saving_apis/) provides model reusability. Otherwise, if the prediction model with the same structure is trained with the different data sets (e.g. prediction models for different education subjects), the trained models should be serialized separately. This resulted in multiple *keras* files. This way, the prediction of students' performance for subject *x* needs both the subject *x* prediction model and subject *x* students' results. For exploitation purposes, the *Python HTTP*

server (embedded module *http.server*) is added as a frontend of the system (Figure 8).

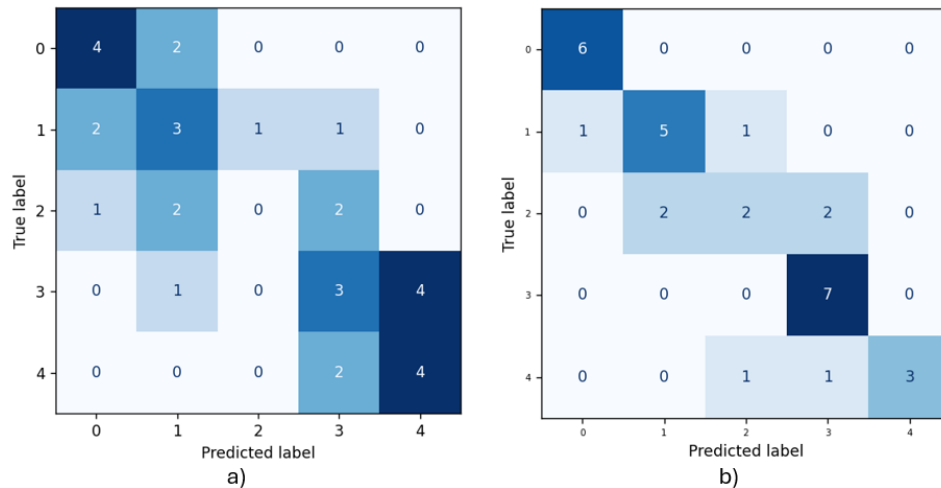


Figure 7 – Confusion matrix of validation subset of data before (a) and after SMOTE improvement (b)

During the server initialization, the system loads the *keras* models and *Python* scripts into the working memory and starts to listen on a specified server port, expecting Web client requests. The system design is flexible and scalable. The client request consists of one or more student records. For instance, it can be a dataset for one study program or dataset for first-year students, as well as an individual student record. The server responds with the category predictions.

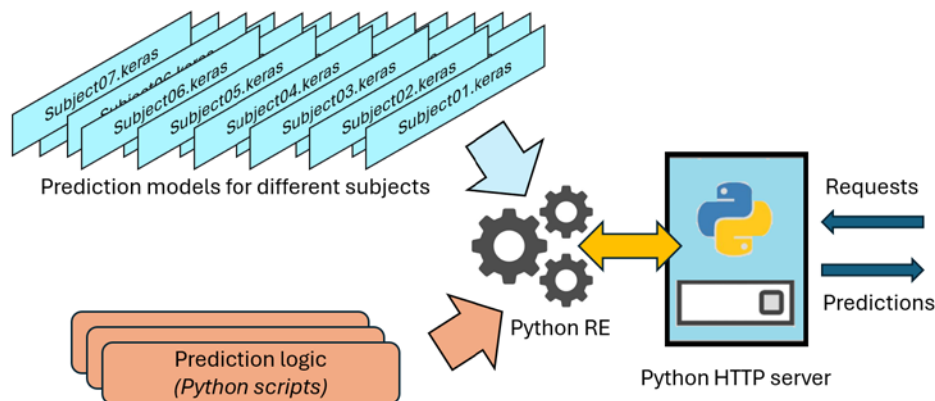


Figure 8 – Overall system architecture

Conclusion

The presented research proposes a solution based on artificial neural networks for predicting students' performance. Originally, the research dealt with solving the problem of classification. The nature of the data directed the way the problem was solved. The presented research covers the theoretical aspects, methods, design, implementation and achieved results in students' performance prediction based on heterogeneous data collected from different sources.

The proposed method for development insists on simultaneous consideration of data and prediction models. This method uses a UML activity diagram to describe complex relationships among the activities of data preparation and the activities involved in the design of prediction models. After the sequential activities of data preprocessing and initial setup of the prediction model, there is a loop of activities for fine-tuning of both the data and the model. In other words, the method proposes an iterative sub-process that will lead to the maximum fitting of the data with the prediction model. After its completion, the prediction model is ready for use.

There are three basic limitations of the research. One of them is the use of ANN as a prediction model due to its portability as easy integration based on Web services in the existing educational environment. There are other efficient models found appropriate for the research (e.g. logistic regression, support vector machine and random forest). These models are excluded from consideration as their serialization includes only basic predefined parameters and consequently, they need to be fitted each time after loading. The second limitation of the research is the dataset that consists of many features (originally 36) and a small number of records (159) due to only one subject (OOD) dataset used. As the raw data included numerous independent features, PCA with ANOVA F-value was performed for selection of principal components. Finally, eight features were selected for the training of the model. The third limitation is the dataset based on only one subject. The data should be prepared separately for each subject due to teachers' evidence that differ in structure and criteria.

Small number of samples made the training process difficult. Missing data completion was performed without teacher intervention and the *k-nearest neighbor imputer* was used instead. Overfitting problem represented the main difficulty (due to a small dataset). It was compensated by two approaches – adding a Dropout layer in the model and L2 regularization of the loss function. The Dropout layer forces the

model to learn more from the consistent features, while L2 regularization enables learning from new samples. Although stratification was performed, the confusion matrix pointed out another problem – an imbalanced number of samples among categories. SMOTE (*Synthetic Minority Over-sampling Technique*) is used for this purpose. SMOTE generates synthetic samples and adds them to the original dataset, changing its structure to be better balanced. The training process was controlled by monitoring the loss function over epochs. Finally, the obtained model accuracy is ~0.65, which represents a satisfactory result due to the previously mentioned problems with dataset.

For experimentation, the research proposed a Web framework based on the *Python* runtime environment and its modules. This framework integrates prediction models and the logic necessary for loading the models, on-demand prediction of students' performance and *request – response* interaction with Web clients.

The proposed solution gives many possibilities for making improvements. One of them should be to extend the research to more subjects – at least to subjects of one study program. Often, data preparation takes more time than other activities. Elimination of unnecessary features, derivation of features to reduce data dimensionality, and dealing with incomplete records are the most important activities during the data preparation stage. On the other hand, prediction models should be updated after data changes.

The proposed application framework offers lightweight components that provide reusability and changeability of prediction models. Moreover, some of the components enable easy integration of the prediction models into different application environments (e.g. Web services), giving engineers the opportunity to improve the systems they are responsible for.

References

- Ahmad, T., Iqbal, J., Ashraf, A., Truscan, D., Porres, I., 2019. Model-based testing using UML activity diagrams: A systematic mapping study. *Computer Science Review*, 33, pp. 98-112. Available at: <https://doi.org/10.1016/j.cosrev.2019.07.001>.
- Al-Theeb, N. et al. 2025. Masarat-Integrated Study Management and Supporting System for University Students. In: Berkouk, D., Chatterjee, U., Bouzir, T.A.K., Dhaou, I.B. (eds) *Proceedings of the 1st International Conference on Creativity, Technology, and Sustainability*. CCTS 2024. *Proceedings in Technology Transfer*, Springer, Singapore., pp. 229-238. Available at: https://doi.org/10.1007/978-981-97-8588-9_22.

Apicella, A., Donnarumma, F., Isgrò, F., Prevete, P., 2021. A survey on modern trainable activation functions. *Neural Networks*, Volume 138, pp. 14-32, Available at: <https://doi.org/10.1016/j.neunet.2021.01.026>.

Bakhouyi, A., Dehbi A. & Broumi, S. 2022. Performance Evaluation of JSON2RDF TransLRS Semantic Solution in SMART EDUCATION by Using CMI-5 Specification. *2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, Riyadh, Saudi Arabia, 2022, pp. 20-25, Available at: <https://doi.org/10.1109/SMARTTECH54121.2022.00020>.

Benkhalfallah, F., & Laouar, M. R. (2023). Predicting Student Exam Scores: Exploring the Most Effective Regression Technique. *2023 International Conference on Networking and Advanced Systems (ICNAS)*, Algiers, Algeria, 2023, pp. 1–9. Available at: <https://doi.org/10.1109/icnas59892.2023.10330495>

Böck, F., Landes, D. and Sedelmaier, Y. 2024. Learner Models: A Systematic Literature Research in Norms and Standards. In *Proceedings of the 16th International Conference on Computer Supported Education (CSEDU 2024)*-Vol.2, pp. 187-196. Available at: <https://doi.org/10.5220/0012556100003693>.

Dwivedi, S. K. & Rawat, B. 2015. A review paper on data preprocessing: A critical phase in web usage mining process. *International Conference on Green Computing and Internet of Things (ICGCloT)*, Greater Noida, India, pp. 506-510. Available at: <https://doi.org/10.1109/ICGCloT.2015.7380517>.

Elreedy, D., Atiya, A.F. & Kamalov, F. 2024. A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. *Mach Learn* 113, pp. 4903–4923. DOI:<https://doi.org/10.1007/s10994-022-06296-4>

Essa, Saadia & Human-Hendricks, Nadia. 2023. Personalised Adaptive Learning Technologies Based on Machine Learning Techniques to Identify Learning Styles: A Systematic Literature Review, *IEEE Access*, 11, pp. 48392-48409. Available at: <https://doi.org/10.1109/ACCESS.2023.3276439>.

Johns, C. 2022. 'Research-based training for undergraduate mathematics tutors'. *International Journal of Mathematical Education in Science and Technology*, 55(10), pp. 2410–2435. Available at: <https://doi.org/10.1080/0020739X.2022.2153759>.

Kaiser, H.F. 1960. The Application of Electronic Computers to Factor Analysis. *Educational and Psychological Measurement*, 20, pp. 141 - 151. Available at: <https://journals.sagepub.com/doi/10.1177/001316446002000116>

Kosareva, L., Demidov, L., Ikonnikova, I., & Shalamova, O. 2023. Ispring platform for learning Russian as a foreign language. *Interactive Learning Environments*, 31(5), pp. 2872-2883. Available at: <https://www.tandfonline.com/doi/full/10.1080/10494820.2021.1913423>.

Labba, Chahrazed & Roussanaly, Azim & Boyer, Anne. 2020. An Operational Framework for Evaluating the Performance of Learning Record Stores. *EC-TEL 2020. Lecture Notes in Computer Science*, vol 12315. Springer pp 45–59. Available at: https://doi.org/10.1007/978-3-030-57717-9_4.

Manso-Vázquez, M., Caeiro-Rodríguez, M. & Llamas-Nistal, M. 2018. An xAPI Application Profile to Monitor Self-Regulated Learning Strategies. *Journal*

IEEE Access, vol. 6, pp. 42467-42481. Available at: <https://doi.org/10.1109/ACCESS.2018.2860519>.

Mao, A. Mohri, M. and Zhong, Y., 2023. Cross-entropy loss functions: theoretical analysis and applications. In *Proceedings of the 40th International Conference on Machine Learning (ICML'23)*, Vol. 202. JMLR.org, Article 992, 23803–23828. Available at: <https://doi.org/10.48550/arXiv.2304.07288>

Mitrovic, A., Ohlsson, S. 2016. Implementing CBM: SQL-Tutor After Fifteen Years. *International Journal of Artificial Intelligence in Education*, 26, pp. 150–159. Available at: <https://doi.org/10.1007/s40593-015-0049-9>.

Ocheja, Patrick & Flanagan, Brendan & Ueda, Hiroshi & Ogata, Hiroaki. 2018. Managing lifelong learning records through blockchain. *Research and Practice in Technology Enhanced Learning*, 14(4), pp.1-19. Available at: <https://doi.org/10.1186/s41039-019-0097-0>.

Rastrollo-Guerrero JL, Gómez-Pulido JA, Durán-Domínguez A. Analyzing and Predicting Students' Performance by Means of Machine Learning: A Review. *Applied Sciences*. 2020; 10(3):1042. Available at: <https://doi.org/10.3390/app10031042>.

Ren, J., Wang, H., 2023. Chapter 3 - Calculus and optimization, Editor(s): Jingli Ren, Haiyan Wang, *Mathematical Methods in Data Science*, Elsevier, pp. 51-89. Available at: <https://doi.org/10.1016/B978-0-44-318679-0.00009-0>.

Robert Frost, H. 2022. Eigenvectors from Eigenvalues Sparse Principal Component Analysis. *Journal of Computational and Graphical Statistics*, 31(2), 486–501. Available at: <https://doi.org/10.1080/10618600.2021.1987254>

Schröer, C., Kruse, F., Marx-Gómez, J., 2021. A Systematic Literature Review on applying CRISP-DM Process Model. *Procedia Computer Science*, Volume 181, 2021, pp. 526-534, Available at: <https://doi.org/10.1016/j.procs.2021.01.199>.

Tsai, Y. S., Perrotta, C., & Gašević, D. 2019. Empowering learners with personalised learning approaches? Agency, equity and transparency in the context of learning analytics. *Assessment & Evaluation in Higher Education*, 45(4), pp. 554–567. Available at: <https://doi.org/10.1080/02602938.2019.1676396>.

Virtanen, M.A., Kääriäinen, M., Liikanen, E. et al. 2017. Use of Ubiquitous 360 learning environment enhances students' knowledge in clinical histotechnology: A Quasi-Experimental study. *Medical Science Educator*, 27(4), pp. 589-596. Available at: <https://link.springer.com/article/10.1007/s40670-017-0429-x>.

Wang, H., Tlili, A., Huang, R. et al. 2023. Examining the applications of intelligent tutoring systems in real educational contexts: A systematic literature review from the social experiment perspective. *Educ Inf Technol* 28, pp. 9113–9148. Available at: <https://doi.org/10.1007/s10639-022-11555-x>.

Woolf, B.P. (2010). Student Modeling. In: Nkambou, R., Bourdeau, J., Mizoguchi, R. (eds) *Advances in Intelligent Tutoring Systems. Studies in Computational Intelligence*, 308, pp. 267–279. Available at: https://doi.org/10.1007/978-3-642-14363-2_13.

Yongmin S., Kim, E., Kim, K., Kim, J.S., Kim, S., Kim, B.N., 2025. Predicting Efficacy of Virtual Reality-Based Stabilization for Individuals with Posttraumatic Stress Symptoms: A Machine Learning Approach. *Psychiatry Investig.*; 22(10):1178-1190. Available at: <https://doi.org/10.30773/pi.2025.0168>.

Predviđanje uspeha studenata na osnovu heterogenih i nejasnih podataka

Goran P. Šimić, **autor za prepisku**, Ana Miletić, Jelena Mitić
Akademija tehničko-umetničkih strukovnih studija Beograd, Odsek Visoka škola elektrotehnike i računarstva, Beograd, Republika Srbija

OBLAST: informacione tehnologije
KATEGORIJA (TIP) ČLANKA: originalni naučni rad

Sažetak:

Uvod/cilj: Predstavljeno istraživanje opisuje primenu veštačkih neuronskih mreža za predviđanje performansi studenata na osnovu heterogenih i nejasnih podataka.

Metode: Podaci studenata prikupljeni tokom tri školske godine na predmetu Objektno orijentisano projektovanje (OOD) na Elektrotehničkom i računarskom fakultetu (Akademija primenjenih studija tehnike i umetnosti u Beogradu), predstavljaju osnovu istraživanja. Veličina uzorka je sto učenika. Istraživanje obuhvata trideset šest različitih karakteristika iz tri različita izvora prikupljenih podataka. Prikazani podaci i rezultati su potpuno depersonalizovani. Istraživanje opisuje metode pripreme podataka kao i modelovanje, obuku, testiranje i evaluaciju veštačkih neuronskih mreža primenjenih za predviđanje performansi studenata.

Rezultati: Eksperimentisanje podacima i modelom predviđanja kao i tačnost modela predstavljaju glavne rezultate istraživanja. Pored toga, istraživanje uključuje diskusije o prethodnoj obradi podataka i tehnikama dubokog učenja koje se primenjuju na modelima predviđanja.

Zaključak: Metode razvoja, modeli predviđanja i rezultati predviđanja kao ishodi, njihov kritički pregled i mogućnosti poboljšanja predstavljaju glavne zaključke istraživanja.

Ključne reči: veštačke neuronske mreže, predviđanje uspeha studenata, arhitektura veb servisa, analiza podataka.

Paper received on: 10 April 2025.

Manuscript corrections submitted on: 23 February 2026.

Paper accepted for publishing on: 24 February 2026.

© 2026 The Authors. Published by Vojnotehnički glasnik / Military Technical Courier (www.vtg.mod.gov.rs). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/rs/>)

