Quality Festival 2025
ISBN 978-86-6335-121-9

Milica Tufegdžić [1]
Aleksandar Mišković
Vladan Čolić
Marija Mojsilović

Research paper
DOI – 10.24874/QF.25.073

IMPROVING QUALITY MAINTENANCE THROUGH NEURAL NETWORK-BASED MACHINE FAILURE PREDICTION

Abstract: *This paper explores a neural network-based approach to machine failure prediction, leveraging deep learning techniques to analyze data. Different neural network architectures with five input features, consisting of one, two, and three hidden layers with 50, 100, and 150 neurons per layer, are tested on a synthetic dataset to identify the most effective model for detecting failure patterns. All procedures are conducted using Python, employing an Multi-layer Perceptron (MLP) classifier. Key performance metrics, including precision, recall, accuracy, and F1-score, are calculated and compared across eight architectures. Additionally, AUC-ROC, Log Loss, and training time are evaluated to determine the optimal model. The results are further analyzed based on class-wise performance for the positive (class 1) and negative (class 0) classes. The best performance is achieved with three hidden layers, each containing 50 neurons, exhibiting a high F1-score, strong recall, and balanced class-wise performance.*

Keywords: *quality maintenance, MLP classifier, neural network architecture, performance metrics, evaluation*

1. Introduction

Quality maintenance is a cornerstone of effective maintenance, particularly in today's dynamic and fast-paced production environment. Achieving production success requires consistently meeting client requirements while ensuring zero machine defects under defined conditions. Advancing smart maintenance solutions can be accomplished with machine learning (ML) algorithms that are trained to predict failures and recommend appropriate actions based on the forecasted failure (Dangut, Jennions, King, et al., 2023).

Deep learning, a subset of Artificial Neural Networks (ANN), is a branch of machine learning, and choosing the appropriate algorithm for a particular domain is challenging due to the diverse purposes of algorithms and their potential to produce different results even with similar data (Sarker, 2021).

To predict the health condition of each machine accurately, hybrid deep learning method integrating Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and attention mechanisms was proposed (Dehghan Shoorkand, Nourelfath, & Hajji, 2024). Deep Adaptation Network (DAN) architecture, which extends the deep CNN to address domain adaptation scenarios, has been proposed (Long, Cao, Wang, & Jordan, 2015). The deep learning model, built using fully connected deep neural networks with six layers, enabled the

[1] Corresponding author: Milica Tufegdžić
Email: mtufegdzic@asss.edu.rs

transfer of knowledge about failure among similar failure types (Li, Kristoffersen, & Li, 2022). Supervised machine learning models, including combined ANN architectures and an improved neuron-by-neuron training algorithm with accumulative neural networks, were used to predict mechanical component failure and estimate Remaining Useful Life (RUL) (Shaheen, Kocsis, & Németh, 2023). Four regression algorithms (Support Vector Regression (SVR), Decision Tree, MLP, and K-Nearest Neighbors (KNN)) were used for RUL prediction, while using empirical mode decomposition to preprocess the input data and improve its quality for predictive modeling (Maior & Silva, 2024). The approach presented in Hakami (2024) utilizes Generative Adversarial Networks to generate synthetic data and LSTM layers to extract temporal features, with ML algorithms, classifying component status using LSTM features and sigmoid activation.

Machine failure identification using synthetic industrial data was investigated by applying and comparing different feature selection techniques, and using three classifiers: Random Forest (RF), Support Vector Machine (SVM), and MLP (Bezerra et al., 2024). A recurrent neural network (RNN) using LSTM is employed to perform predictive maintenance of the Air Booster Compressor motor (Abbasi, Lim, & Yam, 2018). The study conducted by Espinoza-Sepulveda and Sinha (2021) used an ANN approach to develop a robust VFD-ML (variable frequency drives-machine learning) model for fault diagnosis in rotating machines, employing an MLP network structure to create a 2-step diagnosis model. Two predictive models, utilizing CNN and Recurrent Neural Networks, are introduced and assessed using data from an advanced machining process for cutting complex shapes into metal pieces (Jansen, Holenderski, Ozcelebi, Dam, & Tijsma, 2018).

Six algorithms (logistic regression, RF, SVM, LSTM, ConvLSTM, and Transformers) were employed on a binary classification task that assigns a positive label to a prediction window based on the probability of failure occurring within the interval using multivariate time series data (Pinciroli Vago, Forbicini, & Fraternali, 2024). A methodology to process and transform data from a vibration system simulating a motor, creating a dataset to train and test an MLP that predicts the future condition of equipment and forecasts potential failures has been proposed (Scalabrini Sampaio, Vallim Filho, Santos da Silva, & Augusto da Silva, 2019).

The paper will present the possibilities for applying the MLP classifier for machine failure prediction. Different configurations will be considered, with varying numbers of hidden layers and different numbers of neurons in each layer. The performance and efficiency of each configuration will be evaluated and compared based on various metrics to select the best model.

2. Methodology

The study is conducted through the following steps:

- data preprocessing,
- exploratory data analysis,
- selection of relevant features in order to determine inputs;
- determining the number of hidden layers, as well as the number of neurons in each layer;
- models training, and
- models evaluation and comparison.

For this study, the AI4I 2020 Predictive Maintenance Dataset (CSV format) is used (UCI Machine Learning Repository, 2020). This synthetic dataset mimics real-world industrial predictive maintenance scenarios, containing 10.000 instances with 14 features, including UID, Product ID, Type, Air temperature (K), Process temperature (K), (rpm), Torque (Nm), Tool wear (min). Dataset also contains five failure modes: Tool wear failure (TWF), Heat dissipation

failure (HDF), Power failure (PWF), Overstrain failure (OSF), and Random failure (RNF). The machine's overall failure, which serves as the target variable with binary values (two classes), results from any of the five independent failure modes. In other words, the machine is considered to have failed if at least one of these modes occurs. UID, Product ID, and Type are excluded from this study because they do not contribute to predicting the outcome; instead, they may add noise and reduce model performance.

The Pearson correlation coefficients for all feature pairs were calculated using Python. Air temperature and process temperature show a strong positive correlation (0.88). Although a weak positive correlation is observed between machine failure and torque (0.19), higher torque appears to be slightly associated with an increased likelihood of machine failure. Rotational speed and torque exhibit a strong negative correlation (-0.87), suggesting that as rotational speed increases, torque decreases. This is consistent with many mechanical systems. Weak or no significant correlations are noticed between the following pairs: air temperature and rotational speed (0.02); process temperature and torque (-0.01); machine failure and rotational speed (-0.04). Additionally, tool wear and rotational speed show no correlation, indicating that these features are largely independent of each other. Torque (0.19) and tool wear (0.11) show slight positive correlations with machine failure, suggesting that higher torque and tool wear may contribute to increased failure rates.

Feature importance was also considered and presented in the form of a bar chart (Figure 1). The most important feature is torque, with a score of 0.312199, followed by rotational speed (0.189519). Air temperature, tool wear, and process temperature have lower importance (0.172940, 0.166624, and 0.158718, respectively). Torque is the strongest predictor of the target variable, with rotational speed also being influential but not as much as torque. The last three features still contribute, but they are less impactful than torque and rotational speed.

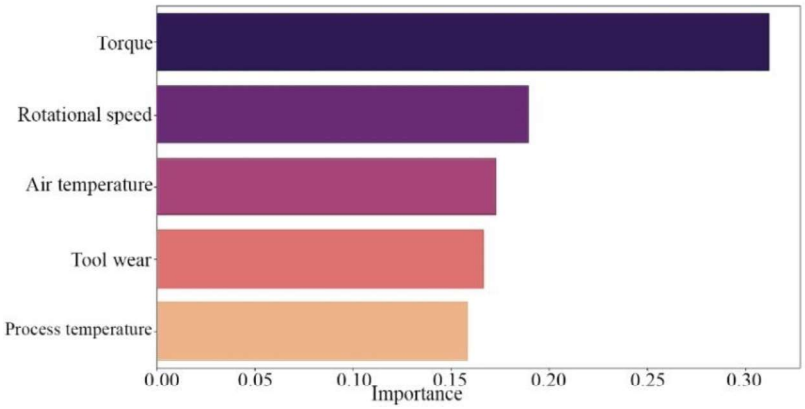

Figure 1. Feature importance analysis

After defining the features and target, the dataset is split into training and testing sets using stratified sampling to maintain class balance. SMOTE (Synthetic Minority Over-sampling Technique) is then applied to the training data to address class imbalance. The data is converted to numeric format, and missing values are filled using the column means. Finally, the features are standardized using StandardScaler to ensure uniform scaling, improving model performance.

Various MLP architectures for classification are evaluated. A list of different neural network structures is defined as follows:

single-layer architectures with (50,), (100,), and (150,); two-layer architectures with (50, 50), (100, 50), and (100, 100); and three-layer architectures with (50, 50, 50) and (100, 100, 50). The numbers in parentheses represent the number of neurons in each layer. A scikit-learn MLPClassifier is trained on the resampled training data, using the ReLU (Rectified Linear Unit) activation function for the hidden layers and the Adam optimizer, an adaptive gradient-based optimization method. The L2 regularization parameter (also known as weight decay) is set to alpha = 0.0001 to prevent overfitting. The learning rate is adjusted dynamically using the adaptive learning rate, while the maximum number of training iterations is set to 500. Reproducibility is ensured by setting a fixed random seed.

Once the model is trained, predictions are made on the test dataset, and the predicted labels are compared against the true labels to evaluate performance for each architecture. Performance metrics, including AUC-ROC, precision, recall, F1 score, accuracy, and Logarithmic Loss (Log Loss), are computed for each model, along with the training time. Additionally, ROC curves are stored for visualization.

To evaluate the model's performance for each class, a confusion matrix is computed, showing the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These values serve as key performance measures for classification problems and are used to calculate precision, recall, and the F1 score for both the positive (class 1) and negative (class 0) classes. Class-wise performance is assessed, and the results are cross-referenced with other performance metrics (precision, recall, F1 score, accuracy), AUC-ROC, log loss, and training time to determine the best overall model.

3. Results and discussion

Basic metrics, along with log loss and training time, are presented as bar charts in Figure 2. The architecture with three layers and 50 neurons per layer achieves the highest F1-score (0.647), followed by architectures with two layers (0.636, 0.629, 0.598). The architecture (100, 100, 50) has a lower F1-score (0.573) compared to the previous ones, while single-layer architectures exhibit the lowest values (0.525, 0.537, 0.536).

The highest precision is observed in the three-layer architecture with 50 neurons per layer (0.545), followed by two-layer architectures (0.519, 0.509, 0.460). The architecture (100, 100, 50) ranks lower than these, with a precision of 0.444. The lowest precision values are again recorded for single-layer architectures (0.401, 0.379, 0.388).

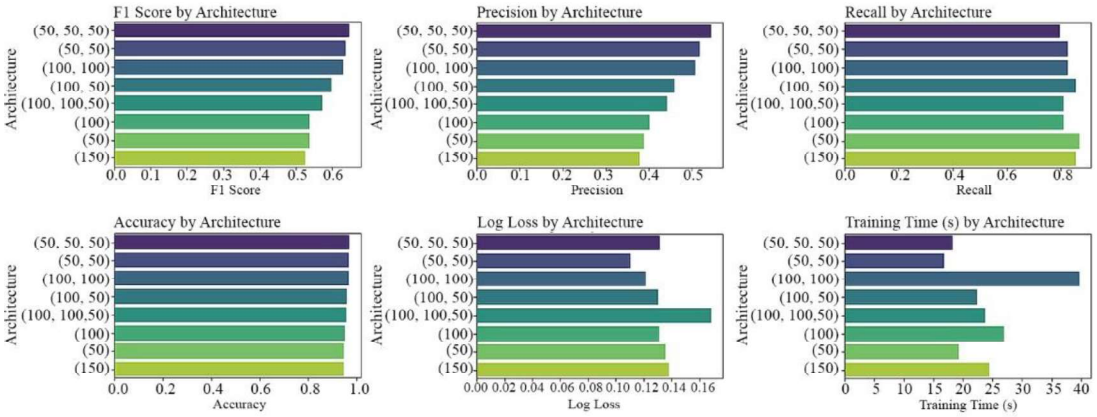

Figure 2. F1 score, precision, recall, accuracy, log loss, and training time for tested models

The recall values exhibit a different trend, with the highest recall (0.868) achieved by the single-layer architecture with 50 neurons. The second-best recall values (0.853) are observed in the architectures (100, 50) and (150,). Next are the two-layer architectures (100, 100) and (50, 50), both with a recall of 0.824. A slightly lower recall (0.809) is recorded for the architectures (100, 100, 50) and (100,). The lowest recall (0.794) is observed for the three-layer architecture (50, 50, 50).

The highest accuracy (0.705) is achieved by the three-layer architecture (50, 50, 50). Two-layer architectures exhibit slightly lower but comparable values (0.680, 0.670, 0.610), followed by (100, 100, 50) with 0.590. The lowest accuracy values are recorded for single-layer architectures (0.525, 0.475, 0.490).

Log loss, which measures the deviation of predicted probabilities from actual class labels, is lowest for two-layer architectures (0.1102, 0.1212, 0.1300), with (50, 50) performing the best. The single-layer architecture with 100 neurons ranks fourth with a log loss of 0.1313, followed by (50, 50, 50) with a slightly higher value (1.316). The next two are single-layer architectures with 50 and 150 neurons, with log losses of 0.1355 and 0.1381, respectively. The worst-performing architecture in terms of log loss is (100, 100, 50), with a value of 0.1686.

The fastest architectures, based on training times, are those with 50 neurons in two, three, and one layer, with times of 17.33s, 18.18s, and 19.52s, respectively. Training times exceeding 20 seconds are observed for the single-layer architecture with 150 neurons, followed by the two-layer architecture (100, 50), the three-layer architecture (100, 100, 50), and the single-layer architecture with 100 neurons. Their respective training times are 23.57s, 23.67s, 23.88s, and 27.32s. The architecture (100, 100) exhibits the longest training time at 40.33s.

All models exhibit very high AUC-ROC values (0.9659 – 0.9741), indicating that each MLP architecture effectively separates failure vs. non-failure classes (Figure 3).

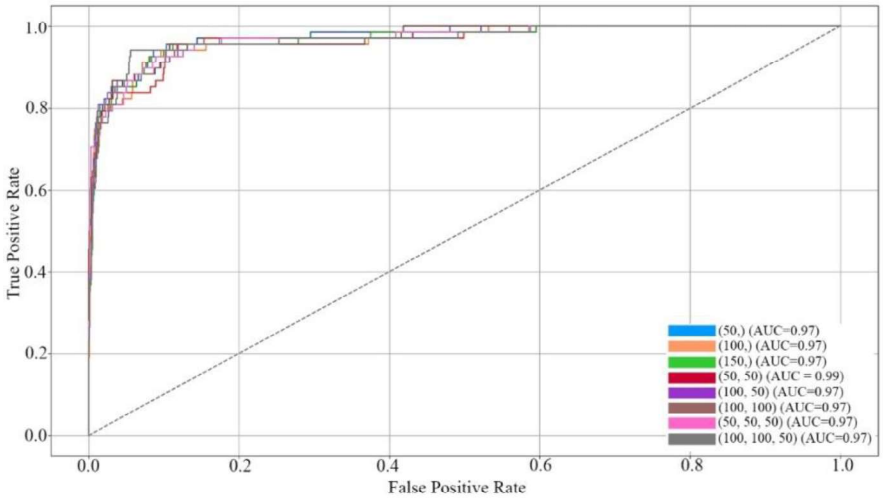

Figure 3. ROC-AUC curves for tested models

The best AUC-ROC value (0.9741) is achieved by the single-layer architecture with 50 neurons. The second-best architecture is (100, 50) with an AUC-ROC of 0.9696, followed by (100, 100) in third place with 0.9620. Slightly below is (50, 50, 50) with 0.9867, followed by (150,) with 0.9862. Next is (100, 100, 50) with an AUC-

ROC of 0.9679. The lowest-ranked architectures are (100,) and (50, 50) with values of 0.9661 and 0.9659, respectively.

Confusion matrices for all models are computed, displaying the counts of TP, TN, FP, and FN, as presented in Table 1.

These values serve as key performance metrics for classification and are used to compute precision, recall, and F1 score for both the positive (class 1) and negative (class 0) classes. The computed values are presented as a class-wise performance in Table 2.

Table 1. Metrics derived from confusion matrices

MLP Model	TP	TN	FP	FN
(50,)	59	1839	93	9
(100,)	55	1850	82	13
(150,)	58	1837	95	10
(50, 50)	56	1880	52	12
(100, 50)	58	1864	68	10
(100, 100)	56	1878	54	12
(50, 50, 50)	54	1887	45	14
(100, 100, 50)	55	1863	69	13

Table 2. Clas-wise performance

MLP Model	Class 1			Class 0		
	Precision	Recall	F1 score	Precision	Recall	F1 score
(50,)	0.388	0.867	0.537	0.995	0.952	0.973
(100,)	0.401	0.809	0.540	0.993	0.958	0.975
(150,)	0.379	0.853	0.525	0.995	0.951	0.972
(50, 50)	0.518	0.824	0.639	0.994	0.973	0.983
(100, 50)	0.460	0.853	0.595	0.995	0.965	0.980
(100, 100)	0.509	0.824	0.6311	0.994	0.972	0.983
(50, 50, 50)	0.545	0.794	0.645	0.993	0.977	0.985
(100, 100, 50)	0.443	0.809	0.562	0.993	0.964	0.978

The best precision for class 1 (positive class) is achieved by MLP (50, 50, 50), which results in fewer false positives. MLP (50,) captures the most positives, as it has the highest recall. The best F1 score is also achieved by MLP (50, 50, 50), indicating the best trade-off between precision and recall.
For class 0 (negative class), all models perform similarly, with an accuracy of approximately 99.3% to 99.5%, due to the significantly larger number of negative samples. MLP (50, 50, 50) also has the best recall and F1 score for class 1 while maintaining strong performance for class 0. MLP (50, 50) and MLP (100, 100) also emerge as well-balanced models.

All architectures have an AUC (Area Under the Curve) of approximately 0.97, indicating excellent classification performance across all models. Since AUC does not differentiate between these models and their performance differences are minimal, ranking should consider other factors such as training time, precision, recall, or F1 score for a more meaningful comparison.

Each MLP architecture effectively separates failure and non-failure classes. The highest AUC is observed for the (50,) architecture, but its performance in other metrics is mixed. Precision varies significantly, ranging from the highest (0.545) for (50, 50, 50) to the lowest (0.379) for (150,). Some models, such as (50,), sacrifice precision to achieve higher recall. All models exhibit very high recall ($\geq$ 0.79), with the highest value (0.8676) for (50,) and the lowest (0.7941) for (50, 50, 50), indicating that they successfully identify most machine failures.

The top F1 scores are 0.647 for (50, 50, 50), 0.636 for (50, 50), and 0.629 for (100, 100), highlighting (50, 50, 50) as the best-balanced model in terms of precision and recall. The strongest all-rounder is (50, 50, 50), followed by (100, 100) and (50, 50) as strong alternatives. The top performer in both F1 score and accuracy is (50, 50, 50),

which also has a competitive AUC and reasonable training time.

The (50,) architecture has the highest AUC but poor F1 and precision scores, indicating a tendency to overpredict positives, leading to false alarms. The (50, 50) model has the best calibration performance, with the lowest Log Loss (0.1102), making it the most reliable for probabilistic predictions. It also delivers strong F1 and accuracy scores while maintaining the shortest training time. Weaker architectures, such as (150,) and (50,), demonstrate high recall but poor precision and F1 scores, suggesting a bias toward overpredicting positives. The (100, 100, 50) architecture has the highest Log Loss, indicating weaker probabilistic predictions.

Based on the analysis, the (50, 50, 50) architecture stands out as the best all-around model due to its high F1 score, strong recall, and balanced class-wise performance.

4. Conclusion

Ensuring quality maintenance with the aid of neural networks is essential for achieving optimal performance, reliability, and efficiency. The selected architecture (50, 50, 50) was chosen through systematic evaluation and experimentation, demonstrating its effectiveness in balancing model complexity and performance. The results indicate that the model achieved a high accuracy rate, validating the impact of proper feature selection and preprocessing techniques.

By implementing systematic evaluation methods, adopting best practices in data preprocessing, and continuously monitoring model performance, organizations can maintain high maintenance standards. Regular updates, robust validation strategies, and adherence to ethical considerations further enhance the credibility and sustainability of neural networks. Moving forward, integrating automated quality assessment tools and fostering interdisciplinary collaboration will be key to overcoming emerging challenges and driving innovation in quality maintenance.

References:

Abbasi, T., Lim, K. H., & Yam, K. S. (2018). Predictive maintenance of oil and gas equipment using recurrent neural network. *OP Conference Series: Materials Science and Engineering, 495*, 11th Curtin University Technology, Science and Engineering (CUTSE) International Conference, 26–28 November 2018, Sarawak, Malaysia. http://dx.doi.org/10.1088/1757-899X/495/1/012067

Bezerra, F. E., Oliveira Neto, G. C. d., Cervi, G. M., Francesconi Mazetto, R., Faria, A. M. d., Vido, M., Lima, G. A., Araújo, S. A. d., Sampaio, M., & Amorim, M. (2024). Impacts of feature selection on predicting machine failures by machine learning algorithms. *Applied Sciences*, *14*(8), 3337. doi: 10.3390/app14083337

Dangut, M. D., Jennions, I. K., King, S., & et al. (2023). A rare failure detection model for aircraft predictive maintenance using a deep hybrid learning approach. *Neural Computing and Applications*, *35*, 2991–3009. doi: 10.1007/s00521-022-07167-8

Dehghan Shoorkand, A., Nourelfath, M., & Hajji, A. (2024). A hybrid deep learning approach to integrate predictive maintenance and production planning for multi-state systems. *Journal of Manufacturing Systems*, *74*, 397-410. doi: 10.1016/j.jmsy.2024.04.005

Espinoza-Sepulveda, N. F., & Sinha, J. K. (2021). Robust vibration-based faults diagnosis machine learning model for rotating machines to enhance plant reliability. *Maintenance, Reliability and Condition Monitoring, 1*(1), 2–9. doi: 10.21595/mrcm.2021.22110

Hakami A. (2024). Strategies for overcoming data scarcity, imbalance, and feature selection challenges in machine learning models for predictive maintenance. *Scientific reports*, *14*(1), 9645. doi: 10.1038/s41598-024-59958-9

Jansen, F., Holenderski, M., Ozcelebi, T., Dam, P., & Tijsma, B. (2018). Predicting machine failures from industrial time series data. In *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)* (pp. 1091–1096). IEEE. doi: 10.1109/CoDIT.2018.8394915

Li, H., Kristoffersen, E., & Li, J. (2022). Deep transfer learning for failure prediction across failure types. *Computers & Industrial Engineering*, *172*(A), 108521. doi: 10.1016/j.cie.2022.108521

Long, M., Cao, Y., Wang, J., & Jordan, M. (2015). Learning transferable features with deep adaptation networks. Proceedings of the 32nd International Conference on Machine Learning, PMLR 37, 97-105.

Maior, C. S., & Silva, T. (2024). Time-series failure prediction on small datasets using machine learning. *IEEE Latin America Transactions, 22*(5), 362–371. doi: 10.1109/TLA.2024.10500720

Pinciroli Vago, N. O., Forbicini, F., & Fraternali, P. (2024). Predicting machine failures from multivariate time series: an industrial case study. *Machines*, *12*(6), 357. doi: 10.3390/machines12060357

Sarker I. H. (2021). Machine learning: algorithms, real-world applications and research directions. *SN computer science*, *2*(3), 160. doi: 10.1007/s42979-021-00592-x

Scalabrini Sampaio, G., Vallim Filho, A. R. d. A., Santos da Silva, L., & Augusto da Silva, L. (2019). Prediction of motor failure time using an artificial neural network. *Sensors*, *19*(19), 4342. doi:10.3390/s19194342

Shaheen, B., Kocsis, Á., & Németh, I. (2023). Data-driven failure prediction and RUL estimation of mechanical components using accumulative artificial neural networks. Engineering Applications of Artificial Intelligence, *119*, 105749. doi: 10.1016/j.engappai.2022.105749

UCI Machine Learning Repository (2020). AI4I 2020 Predictive Maintenance Dataset [Dataset]. doi: 10.24432/C5HS5C

Milica Tufegdžić
Academy of professional studies Šumadija,
Kragujevac,
Serbia
mtufegdzic@asss.edu.rs
ORCID 0000-0003-3856-1498

Aleksandar Mišković
Academy of professional studies Šumadija,
Kragujevac,
Serbia
amiskovic@asss.edu.rs
ORCID 0000-0002-7390-9886

Vladan Čolić
Academy of professional studies Šumadija,
Kragujevac,
Serbia
vcolic@asss.edu.rs
ORCID 0009-0002-5125-1849

Marija Mojsilović
Academy of professional studies Šumadija,
Kragujevac,
Serbia
mmojsilovic@asss.edu.rs
ORCID 0009-0004-6818-2450