https://doi.org/10.58245/ipsi.tir.2502.09

Kolmogorov-Arnold Networks in Identification of Chromatic Index of Cubic Graphs

Modrovičová, Bianka; and Dudáš, Adam

Abstract: ***The chromatic index of a graph denotes the number of colors needed for such a coloring of this graph, that no two adjacent edges are colored with the use of the same color. The value of this graph property is commonly used in several real-world problems, such as the determination of the number of time slots for traffic lights placed in the intersection system, or the allocation of registers to variables in the compilation of code. Since the problem of chromatic index identification is NP-complete, the conventional computing methods are of high time complexity. This motivated the recent use of machine and deep learning models for the approximate determination of the value of the graph property. In this study, the Kolmogorov-Arnold Network is designed, implemented, and experimentally evaluated in the context of the selected task. This model has been utilized for its high decision-making quality and strong interpretability, which are both explored in the presented work through conventional classification metrics such as accuracy and precision and through means of visualization and symbolic approach to the interpretation of the decision-making process.***

Index Terms: ***Kolmogorov-Arnold Networks, Chromatic Index, Cubic Graphs, Classification, Data Analysis***

Manuscript received February 15, 2025

This research was funded by a grant from the Grant Agency of the Ministry of Education, Science, Research, and Sport of the Slovak Republic (KEGA No. 014UMB-4/2023).

Bianka Modrovičová is with the Department of Computer Science, Matej Bel University in Banská Bystrica, Slovakia (e-mail: `bianka.modrovicova@student.umb.sk`). Adam Dudáš (corresponding author) is with the Department of Computer Science, Matej Bel University in Banská Bystrica, Slovakia (e-mail: `adam.dudas@umb.sk`).

1. Introduction

Graphical structures and problems defined on them are widely used in several computer science and interdisciplinary applications, while all of them are based on the representation of the state space of the problem via a graph G such that [16]:

$$G = (V, E), E \subseteq V^2 \tag{1}$$

where V denotes a set of vertices interconnected using a set of edges E.

The specific graph problem arising in various areas, from transportation to register allocation, is the problem of proper edge coloring of a graph. This type of edge coloring can be described as a task of assignment of colors to the edges of a graph in such a way, that no two edges, incident to the same vertex, are colored with the use of the same color [27]. The number of colors needed for proper edge coloring of a graph G is determined by the so-called chromatic index of a graph, denoted by $\chi'(G)$ for which Vizing's theorem specifies [27]:

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1 \tag{2}$$

where $\Delta(G)$ is the highest degree of a graph G, meaning the highest number of edges incident to one of the vertices of the graph.

This work focuses strictly on cubic graphs, where $\Delta(G) = 3$. Therefore the value of $3 \leq \chi'(G) \leq 4$ is relevant, where cubic graphs with $\chi'(G) = 3$ are called standard 3-edge colorable cubic graphs and those with $\chi'(G) = 4$ can be called either 3-edge uncolorable graphs or snarks. Even with this constraint, the problem of proper edge coloring of a graph remains NP-complete.

Conventional methods for the determination of the chromatic index value of a graph are computationally expensive, which is natural due to the NP-completeness of the problem [20]. This computational complexity led to the use of machine and deep learning methods for the solution of the task, similar to the problems from other areas of interest, eg. transportation [14, 22] or mycology [5] where the time and computational complexity are of the essence.

The main objective of the work is the utilization of Kolmogorov-Arnold Networks as a classifier for the identification of the chromatic index of a cubic graph. Even though other classification approaches, such as Support Vector Machines, Decision Trees and Random Forests, or Multilayer Perceptron Networks were used in the studied problem, the proposed Kolmogorov-Arnold Network-based model presents a more complex, yet more interpretable solution to the approximation of the chromatic index values [18, 19].

Hence, the novelty of the work can be summarized into the following three points:

- Proposal of Kolmogorov-Arnold Network-based approach to unambiguous identification of the chromatic index of the cubic graph through the classification process.
- Identification of cubic graph properties relevant to the decision-making of the proposed network model.
- Interpretation of the decision-making of the created network model through visualization and symbolic representation approaches.

The body of this study is structured as follows – Section 2 presents a brief overview of conventional methods used in edge coloring of cubic graphs, utilization of machine and deep learning models in graph property value estimation, and summarization of results of our previous work focused on the use of prediction models in the context of identification of cubic graph's chromatic index value. In Section 3 the original cubic graph property dataset is introduced and described. This dataset is then used for the training and testing procedures of Kolmogorov-Arnold Network described in Section 4. Lastly, the main results of the study and potential future work areas are summarized in Section 5.

2. Related Works

Since the work presented in this study is focused on the use of machine learning models for the estimation of the value of graph properties, in this section, we offer a brief description of focal points and results in the area of modern graph theory applications, incorporation of the machine and deep learning approaches to the determination of graph property values, and overview of previous results in the area of chromatic index identification with the use of predictive analysis methods.

The concepts from the area of graph theory are commonly used as a component of several real-world problems – from geospatial sciences, to blockchain technology, and medical sciences.

Specifically, the work described in [13] highlights the role of graph theory in various aspects of geospatial sciences, such as transportation, urban planning, environmental management, and disaster studies. Research presented in this work emphasizes the potential of graph theory for innovative spatial analytics, including digital twin technology, and links these applications to internationally set policies.

On the other hand, in [7], authors utilize graph theory concepts to model and analyze blockchain networks, focusing on decentralization, security, privacy, and scalability. The study uses a graphical representation of blockchains and their properties to provide insights into the structure of the systems, their properties and disadvantageous points.

In [15] the main focus is put on the exploration of the use of graph theory as a valuable tool in liver research, facilitating advancements in the detection, treatment, and understanding of liver diseases by analyzing complex interactions across various organizational levels.

The machine and deep learning methods are applied in a number of modern approaches to graph theory problems.

Among others, the study [1] addresses the register allocation problem, a critical phase in compiler optimization commonly solved using graph coloring, where the number of needed registers corresponds to the number of colors needed for proper vertex coloring of the graph. This study focuses on the lack of suitable datasets for machine learning approaches by developing a code interference graph generator and creating the first dataset for training machine learning models in this domain. With the use of this dataset, authors build machine learning models to explore novel register allocation strategies and refine conventional techniques.

In [11] authors design and implement a graph convolutional network model as the surrogate for the greedy algorithm of computation of the zero-forcing sets in graphs – a coloring problem with applications in network science and control systems. The main benefit of the presented work lies in the addition of natural scalability, when compared to the traditional heuristic-based method, thus achieving significant speed improvements over the baseline heuristic approaches.

The use of machine and deep learning models in chromatic index identification has been explored to some depth in our previous works, mainly [2, 3, 4]. Hence, as a last part of the works related to the presented article, we present a brief overview of the focuses and results reached in these studies.

The first of these works [2] focuses on the design and gathering of datasets of graph properties suitable for machine learning tasks and the building of decision tree models for effective identification of chromatic index value. The most significant contribution of this work lies in the creation of a small graph property dataset for machine learning purposes, which was experimentally verified via the construction of decision tree models.

In [4] the main objective concerns the application of an interpretable random forest ensemble method to analyze the edge 3-colorability of cubic graphs and identify critical graph properties influencing this edge colorability, and therefore chromatic index value. A random forest model trained on the data presented in the previously mentioned work achieves the accuracy of classification of 97.35% while the utilization of the Shapley value method highlights the importance of individual graph properties in the process of supervised learning.

Finally, the research [3] explores the effectiveness of various machine learning-based classification models in the problem relevant to this study and presents an in-depth comparative analysis of their qualitative and explainative properties.

Even though the results of these works are relatively satisfactory, there are two aspects of the approach, which need to be improved upon:

- The size of the training and testing dataset used in the previous studies was limiting for proper generalization of the edge-coloring properties.
- The quality and interpretability of the models offered only constrained decision-making guidelines to replace conventional models for the identification of chromatic index value.

Both of these problems create the focus of the presented Kolmogorov-Arnold Network-based apporach to the task of chromatic index value determination.

3. Cubic Graph Property Dataset for Machine Learning Models

As mentioned briefly in Section 2, as a part of our previous work presented in [2, 3, 4] the cubic graph property dataset of limited size was created. Specifically, this dataset uses 27 structural and algebraic graph properties to describe 2,000 cubic graphs distributed equally between standard 3-edge colorable cubic graphs with $\chi' = 3$ and 3-edge uncolorable cubic graphs having $\chi' = 4$. The dataset has been collected using the House of Graphs portal [9] in combination with the Graph Filter [6] and Sage Math [26] tools.

Other than the limited size of the dataset, some of the measured graph properties were redundant, which resulted in the model either not using these properties at all, or using them and reducing its decision-making quality (eg. the time of the training process was significantly increased). Therefore, this dataset served the feature selection purposes for future sets of data.

These disadvantages of the datasets and tools used in the previous work motivated the design and implementation of the original graph property data generator based on the *Python* language. With the use of this generator, a set of graph property datasets consisting of one main dataset of a million graphs and its four subdatasets of various sizes and distributions was created [24]. For the training and testing of the Kolmogorov-Arnold Network, the main of these datasets which consists of 13 attributes and 1,000,000 graphs in equal distribution between the standard cubic graphs and snarks is used. The attributes of this dataset include the following graph properties:

- The number of vertices of a graph.
- The smallest eigenvalue of the adjacency matrix of a graph.
- The matching number of a graph.
- The second largest eigenvalue of the adjacency matrix of a graph.
- The radius of a graph.
- The diameter of a graph.
- The girth of a graph.
- The group size of a graph.
- The vertex connectivity of a graph.
- The density of a graph.
- The largest eigenvalue of the adjacency matrix of a graph.
- The independence number of a graph.
- The chromatic index of a graph.

From the point of view of supervised learning, the first 12 attributes represent input data for the designed model and, naturally, the chromatic index is considered the output data value from the approximation task. The distribution of values of these attributes in the dataset is described in Figure 1.

4. Application of Kolmogorov-Arnold Networks in the Identification of Chromatic Index of Cubic Graphs

Since this study focuses on the application of machine learning principles in the identification of the chromatic index of cubic graphs based on their simply computable properties, the main process of the presented work consists of the following:

- Splitting the labelled graph property data described above into training and testing samples in the ratio of $80 : 20$.
- Training procedure, in which the function for the classification of a graph described by its properties into one of two considered classes is approximated. For the selected problem, we consider the class of standard cubic graphs and the class of snarks.
- Testing procedure for evaluation of the quality of the decision-making done by the model created in the training procedure.
- Interpretation of the results using visualization methods and symbolic representation of the created model.

As mentioned above, for this work an alternative to Multilayer Perceptron Networks – Kolmogorov-Arnold Networks – are considered as the main classifier. The main reason for the utilization of this type of model is its strong interpretability features, namely its ability to identify relevant features, reveal modular structures, and discover symbolic formulas [19]. This interpretability is caused by the fact that the Kolmogorov-Arnold Networks are based on the generalized Kolmogorov-Arnold representation theorem [17], which consists of univariate functions with learnable coefficients of local B-spline basis functions [18].

The proposed approach is implemented in *Python* language with the use of *pandas*, *numpy*, *torch*, *sklearn*, *matplotlib*, and *kan* packages on the Intel i7 10700F processor with 32 GB of RAM.

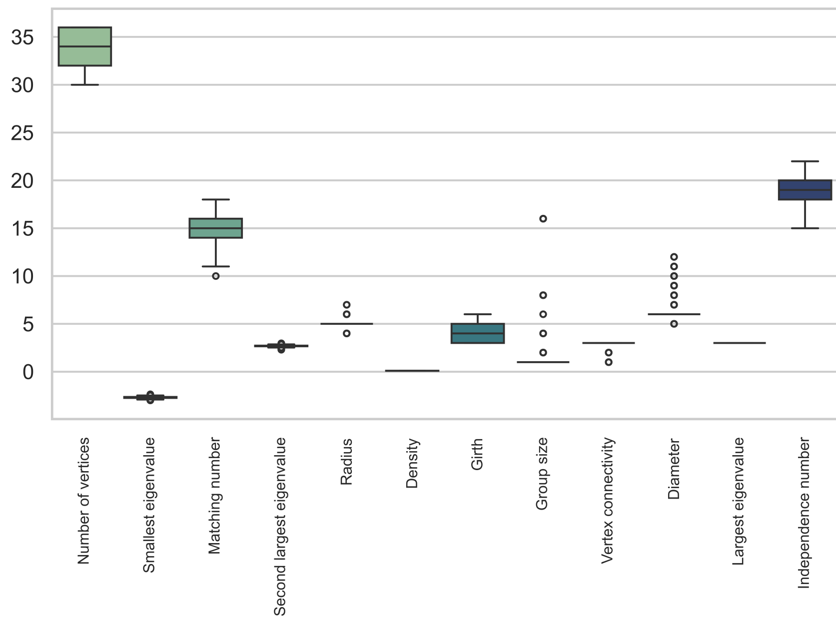

Figure 1: Visualization of the key values in the attributes of the used graph property dataset

4.1 Evaluation of Quality of the Proposed Model in Chromatic Index Identification

A well-known problem in the design and implementation of any neural network model is the definition of its structure and setting of the so-called hyperparameters of the model. Other than the number of layers of the network and the number of neurons in these layers, various activation functions often produce vastly different results. However, Kolmogorov-Arnold Networks are not dependent on user-defined activation functions, and instead, learn their own activation functions from the training data sample, which simplifies the hyperparameter tuning process slightly [19].

Therefore, in this work, the following structural properties and hyperparameter values are tuned [25]:

- Structure of the network – consists of the number of layers and the number of neurons in individual layers of the proposed neural network. This structure for a network of l layers is denoted by $[n_1, n_2, ..., n_l]$, where n_i is the number of neurons in the i-th layer of the network. Since the dataset used in this work contains 12 input attributes and two output classes, the number of neurons for the input and output layers is set accordingly. The tuning process itself, therefore, concerns the structuring of hidden layers of the network.
- The number of grid points – determines the level of the input domain discretization.
- Piecewise polynomial order – the degree of the polynomials used in each piece of the partition.
- Optimizer – the optimization algorithm used in the training process.
- The number of steps – the total number of iterations in the process of training of the network.
- Learning rate – the size of the step used by the selected optimizer when updating model parameters in the function approximation process.

For chromatic index identification several network structure and hyperparameter configurations were examined. Table 1 presents five

Table 1: Description of individual Kolmogorov-Arnold Network models and their hyperparameters

Model	**Structure of layers**	**Number of grid points**	**Piecewise polynomial order**	**Optimizer**	**Number of steps**	**Learning rate**
model$_1$	[12, 6, 2]	3	5	Adam	100	0.05
model$_2$	[12, 8 ,4, 2]	5	5	Adam	100	0.05
model$_3$	[12, 6, 3, 2]	5	3	Adam	200	0.05
model$_4$	[12, 6, 3, 2]	5	3	Adam	100	0.01
model$_5$	[12, 6, 3, 2]	5	3	LBFGS	100	0.01

of the considered model configurations, which reached the highest quality results.

Each of the models presented in Table 1 was applied to the conventional training and testing processes and for each of the models the following quality metrics were measured [8]:

- Accuracy of the model (A) – the proportion of correct predictions among total predictions. For two classes (C_1 and C_2) the accuracy of a model is computed as:

$$A = \frac{n_c(C_1) + n_c(C_2)}{n_c(C_1) + n_c(C_2) + n_i(C_1) + n_i(C_2)} \tag{3}$$

  where $n_c(C_1)$ denotes the number of correctly estimated instances of the class C_1, $n_c(C_2)$ is the number of correctly estimated instances of the class C_2, $n_i(C_1)$ is the number of incorrectly estimated instances of the class C_1, and $n_i(C_1)$ denotes the number of incorrectly estimated instances of the class C_2. Such a notation is used in all of the measured decision-making quality metrics.

- Precision of the model (P) – the proportion of correct predictions in individual classes to the amount of falsely positive classifications into the other class. The precision for a class C_1 is computed as:

$$P_{C_1} = \frac{n_c(C_1)}{n_c(C_1) + n_i(C_1)} \tag{4}$$

- Recall of the model (R) – the proportion of correct predictions in individual classes to the amount of falsely negative classifications in another class. The recall for a class C_1 is computed as:

$$R_{C_1} = \frac{n_c(C_1)}{n_c(C_1) + n_i(C_2)} \tag{5}$$

- F1 score of the model ($F1$) – computed as the harmonic mean of precision and recall metrics:

$$F1_{C_1} = 2 \times \frac{Precision_{C_1} \times Recall_{C_1}}{Precision_{C_1} + Recall_{C_1}} \tag{6}$$

- Training time of the model (T) – total time needed in the training process of the model, computed as:

$$T = \sum_{i=1}^{\#s} t_i \tag{7}$$

  where $\#s$ denotes the number of steps used in the training of the model, and t_i is the time needed for the computation of the i-th step of this process.

A summarization of the measured quality metrics for all of the abovementioned Kolmogorov-Arnold Networks is presented in Table 2 and Figure 2. Best results for individual metrics are marked with the use of bold lettering.

Based on the results of the qualitative analysis of the considered network configurations, the decision-making conducted by the $model_5$ can be perceived as the most fitting for this study. The model consists of two hidden layers of neurons using pairs of additive and multiplicative neurons – the first

Table 2: Quality metrics of considered Kolmogorov-Arnold Network models

Model	**A [%]**	**P**$_{standard}$ **[%]**	**P**$_{snark}$ **[%]**	**R**$_{standard}$ **[%]**	**R**$_{snark}$ **[%]**	**F1**$_{standard}$ **[%]**	**F1**$_{snark}$ **[%]**	**Training time [min]**
model$_1$	81.71	80.26	83.31	84.11	79.31	82.14	81.26	60
model$_2$	97.51	99.26	95.94	95.79	99.29	97.5	97.58	105
model$_3$	97.77	99.14	96.47	96.37	99.16	97.74	97.8	90
model$_4$	97.81	99.12	96.57	96.48	99.15	97.78	97.84	60
model$_5$	**98.15**	**99.5**	**96.87**	**96.78**	**99.51**	**98.12**	**98.17**	60

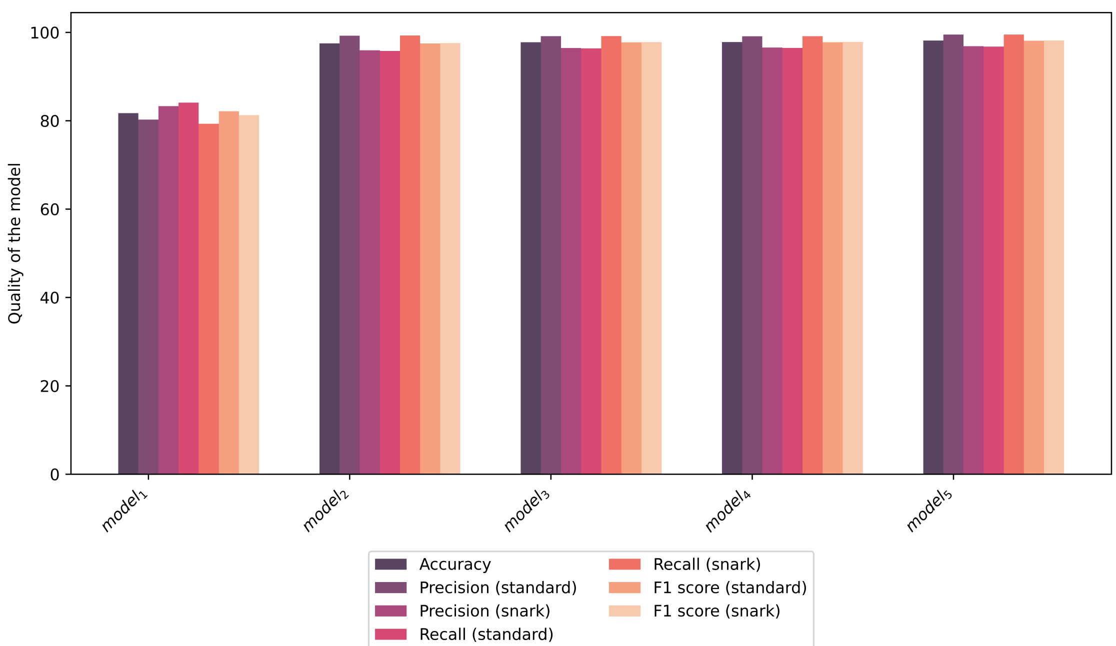

Figure 2: Quality metrics of considered Kolmogorov-Arnold Network models

of the hidden layers is composed of six pairs of such neurons, while the second contains three pairs of neurons. The Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (LBFGS) [12] is utilized for the optimization in the process of learning, while having 100 steps in training, each with a learning rate of 0.01.

Regarding the quality of the model, the accuracy, precision, recall, and F1 score of the $model_5$ is comparable to, or marginally higher, than other complex machine learning approaches used for the chromatic index value identification in cubic graphs, like Multilayer Perceptron Networks or Random Forest [3]. However, the strong advantage of the Kolmogorov-Arnold Network-based approach to the problem is its inherent interpretability and explainability.

4.2 Interpretability of the Proposed Model in Chromatic Index Identification

Even though large and complex machine learning models reach a high quality of decision-making, the problem of interpreting these decisions is often of great significance to human experts and analysts. The lack of such an interpretation of the decision-making leads to lacking description of the problem's state space, and therefore, new knowledge is hard to obtain from these models. The area of interpretability of machine learning approaches consists of a large number of methods and algorithms, which can be used to explain various aspects of automated decision-making processes based on artificial intelligence [23].

In the context of Kolmogorov-Arnold Networks, the natural interpretability is one of the model's strong advantages, implemented in the form of [19]:

- Visualization of the network and strength of the contribution of individual neurons to the decision-making process leading to natural model-based feature selection.
- Possibility of symbolic representation of the network's decision-making via generated, learned decision-making equations.

Figure 3 presents a visualization of the $model_5$ network with the input layer being presented on the bottom of the visualization and the output layer on the top. Each of the neurons of the input layer represents of the input attributes, specifically from the left – the number of vertices, smallest eigenvalue, matching number, second largest eigenvalue, radius, diameter, girth, group size, vertex connectivity, density, largest eigenvalue, and independence number. The output layer contains two neurons, each for one of the considered classes of cubic graphs – standard cubic graphs on the left and snarks on the right.

This visualization of the proposed network contains highly usable information sought after in the area of interpretability – the strength of influence of individual input graph properties or their subsets on the decisions made by the model as a whole. Specifically, in Figure 3 the stronger the influence of the property on the model, the higher the alpha of the connection between two neurons. Therefore, the strongest influence of graph properties on the model is denoted by the darkest neuron-to-neuron connections. This leads to the possibility of categorization of input graph properties based on their influence on the proposed network:

- Strong influence – the number of vertices, second largest eigenvalue, and girth of the graph.
- Significant influence – smallest eigenvalue, matching number, diameter, density, and independence number of the graph.
- Marginal influence – radius, group size, and vertex connectivity.
- No influence – largest eigenvalue of the graph.

The strength of the contribution of individual graph properties is significant for the potential building of less complex models in the future, where only the properties of strong or significant influence can be used for decision-making purposes.

Besides the visualization usable in graph property selection, the proposed model computes a symbolic representation of the decision-making process done by the network in the

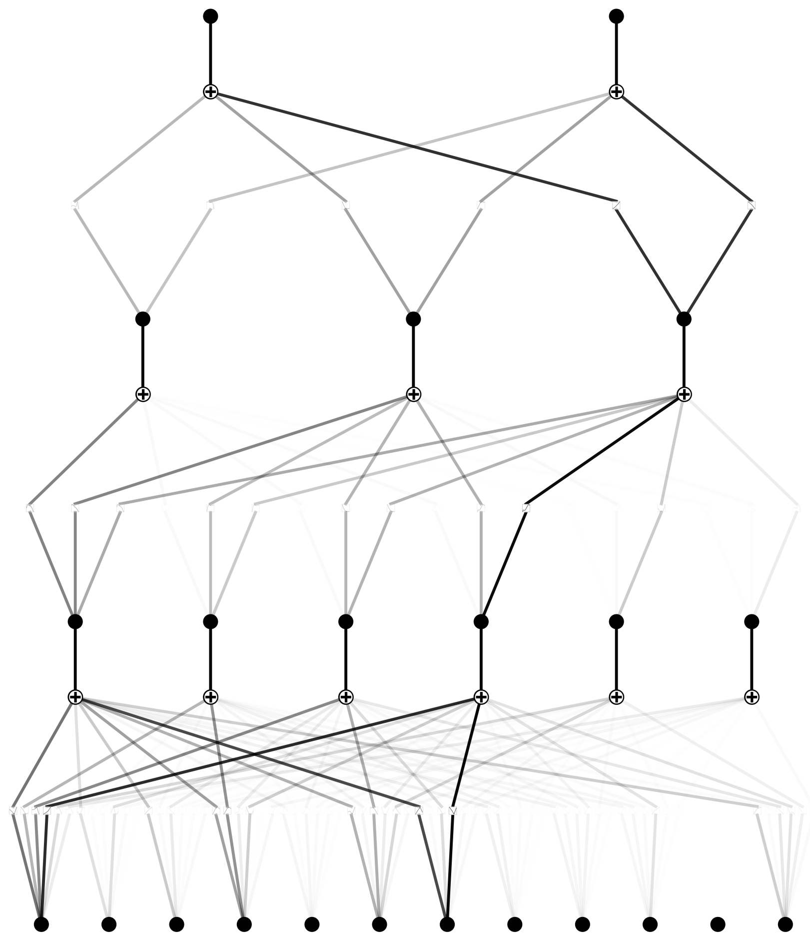

Figure 3: Visualization of the proposed Kolmogorov-Arnolds Network for chromatic index identification (for the better readability of learned activation functions of the model, see Appendix A)

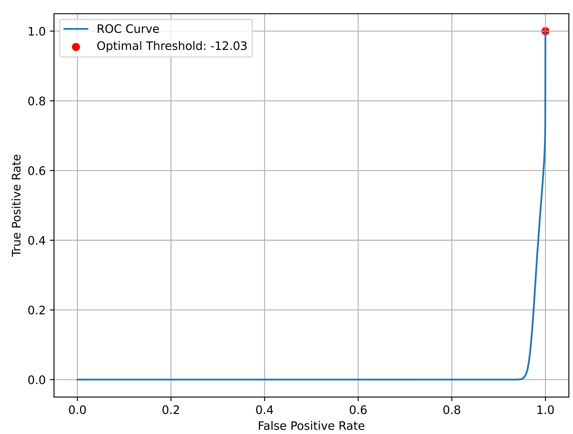

Figure 4: Identification of threshold for the symbolic representation of the proposed Kolmogorov-Arnold Network

form of a formula. For a graph G described by its properties $[x_1, x_2, ..., x_{12}]$, this formula computes a value of decision metric d as noted in Appendix B. Based on this value, the chromatic index χ' of a G can be estimated as follows:

$$\chi'(G) = \begin{cases} 4, & \text{if } d(G) \geq -12.0292, \\ 3, & \text{otherwise.} \end{cases} \quad (8)$$

The value of the threshold for this decision metric d was determined via exploration of the data from the dataset used for the training of the model (see Figure 4). Even though the symbolic formula of the model is complex and contains a large number of elements, its computational complexity is very low, and therefore, it is easily usable to compute the chromatic index of a cubic graph based on the values of its properties.

5. Conclusion

Since the identification of the chromatic index of a cubic graph is relevant as a subtask of several larger complex problems, such as transportation planning, machine scheduling, or register allocation, the development of fast and inexpensive solvers is necessary [10, 21]. This work focuses on the design and implementation of the Kolmogorov-Arnold Network-based method for the identification of the value of this graph property with a particular emphasis on the system's quality of decision-making and interpretability.

The main results of this work include the proposal of the aforementioned type of neural network model reaching the accuracy of 98.15% on the used dataset of 1,000,000 cubic graphs, identification of cubic graph properties relevant to the decision-making of the proposed network model, and interpretation of the decision-making of the created network model through visualization and symbolic representation approaches. This symbolic representation offers a solution to the problem of identification of the chromatic index of a cubic graph with the need for minimal computational costs – memory- and time-wise. Specifically, the time of computation of the symbolically represented network created in this work takes approximately 10% of the time necessary for other, more conventional, approaches to the problem (such as chromatic index identification heuristics).

Regarding future work, there are three main areas, which need to be explored for a potential improvement of the results reached in this study:

- Since this work focuses on the identification of the chromatic index of strictly cubic graphs, in the future this task needs to be generalized to the graphs of any size and degree of vertices.
- Kolmogorov-Arnold Networks are notoriously well adapted for their pruning and building of small models, which can be trained fast and still maintain their decision-making quality and interpretability. Such pruning would be fitting for the model in order to reduce the computational complexity of the produced symbolic representation of the network even further.
- Lastly, the employment of the proposed system in the context of real-world problems is needed in the future. Mainly the scheduling of traffic light patterns in intersection systems of urban areas necessitates the use of a similar system.

Data and Code Availability

The data used in the experiments presented in this work are freely available under Create Commons Attribution 4.0 International License in the following repository:
`https://gitfront.io/r/bmodrovicova/nujnQqs7C9bp/SnarkGenerator/`

The code for the proposed Kolmogorov-Arnold Network is available at:
`https://github.com/bibisvk/KAN`

Appendix A: Learned Activation Functions of the Proposed Kolmogorov-Arnold Network

Since activation functions of the Kolmogorov-Arnold Network are not set by a user, but instead, are learned from data, Figure 5 presents activation functions learned by the network with the highest decision-making quality described in Section 4.1. This figure contains only those activation functions, graph properties, and neurons of the network, which contributed to the decision-making process of identification of cubic graph chromatic index.

The proposed network consists of two hidden layers of neurons – the first of six pairs and the second of three. Therefore, in Figure 5, these neurons are denoted simply by the numbers 1-6 for the neurons of the first hidden layer from left to right and by 7-9 for the neurons of the second hidden layer.

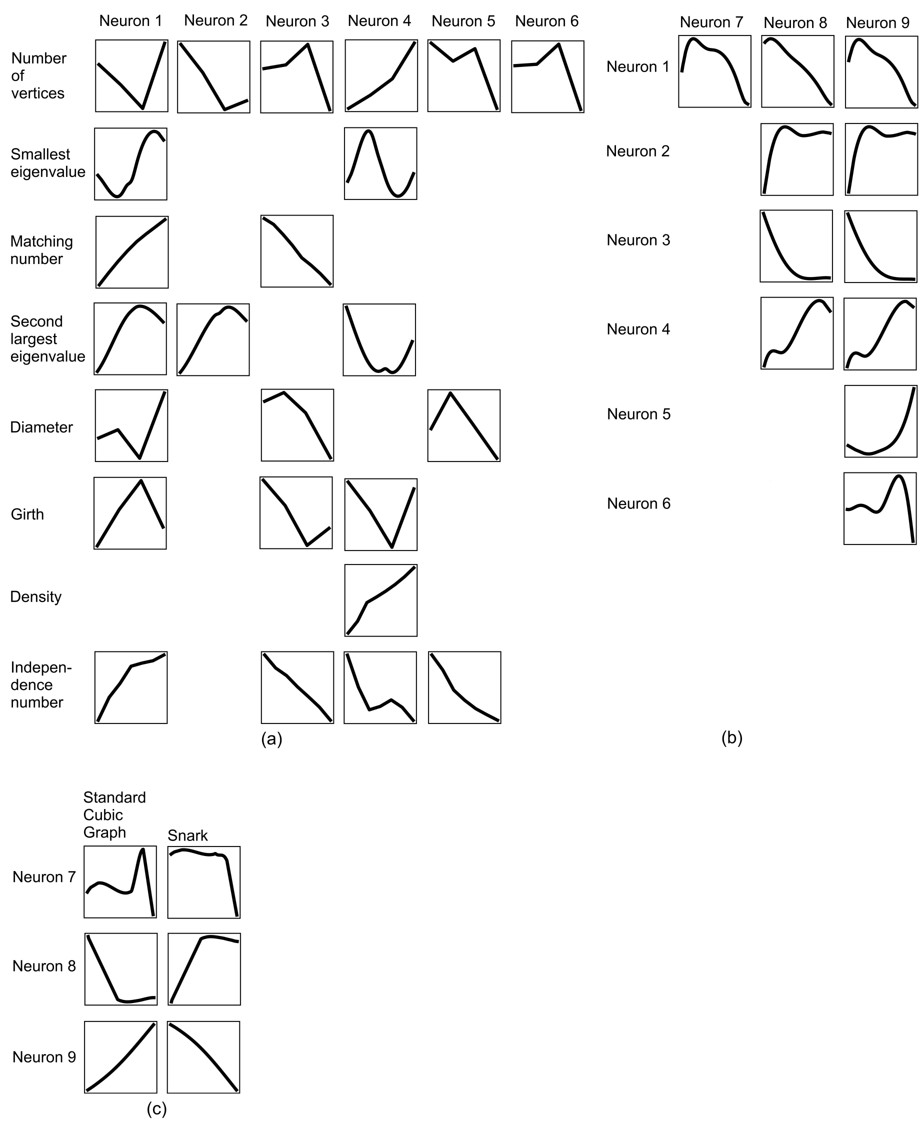

Figure 5: Activation functions between (a) the input layer and the first hidden layer, (b) the first and the second hidden layer, (c) the second hidden layer and output layer of the proposed network

Appendix B: Symbolic Representation of the Proposed Kolmogorov-Arnold Network

The proposed Kolmogorov-Arnold Network can be written symbolically in the form of the formula presented on the following pages. It is important to note, that the formula utilizes all 12 input graph properties (coded into $x_1, x_2, ..., x_{12}$ in the order specified above) and is based on simple arithmetic operations on functions of low computational complexity, such as sinus or exponential function.

$$d[x_1, x_2, ..., x_{12}] = 0.0107095266773318 * x_1 + 0.205407675711068 * x_{1}0 - 0.242535025206612 * x_{1}2 - 3.85891316097653 * x_2 + 0.0229479383564535 * x_3 - 10.4837526362644 * x_4 - 47.678909902052 * x_6 - 0.227666069893135 * x_8 + 1256.13758131579 * (0.0949923110198147 - x_6) * *2 - 5.22791701938276 * sqrt(1 - 0.0277508375039543 * x_1) + 0.938824018333019 * (1 - 0.251976060154688 * x_7) * *2 + 0.0220615816233658 * (1 - 0.761163582712494 * x_9) * *2 - 2.18324837685857 * (1 - 0.415002474199746 * x_9) * *2 - 6.69533749999619e - 8 * exp(3.87999963760376 * x_{1}1) - 8.20252752349714e - 7 * exp(4.0479998588562 * x_{1}1) + 0.722986264912735 * exp(0.1872798204422 * x_9) - 0.000343120202141907 * exp(0.869759798049927 * x_9) + 810998.515851917 * exp(-0.217508074538778 * x_1 - 0.0777717749532465 * x_{1}0 - 0.171460502731447 * x_{1}2 - 1.0177368606997 * x_2 - 0.0518738074314326 * x_4 - 71.9600432616145 * x_6 - 0.0412249592387055 * exp(0.869759798049927 * x_9) + 0.137130454056456 * sin(6.98776006698608 * x_3 - 6.02040004730225) - 0.214402511897815 * sin(9.59679985046387 * x_5 - 1.18351984024048) + 0.179022479707269 * sin(5.20864009857178 * x_7 - 8.23416042327881) + 3.5706993480165 * sin(3.10967993736267 * x_8 - 6.21591949462891)) + 0.0270585469243417 * exp(-0.0758680214659809 * x_{1}0 + 0.0166777460135076 * x_{1}2 + 0.616987674080374 * x_2 + 0.00306169428553416 * x_3 + 0.448459029647388 * x_4 + 0.00851199068506341 * x_8 - 795.312133623488 * (0.0949923110198147 - x_6) * *2 + 3.31000831512147 * sqrt(1 - 0.0277508375039543 * x_1) + 1.38230393760736 * (1 - 0.415002474199746 * x_9) * *2 + 0.0953550300214872 * sin(2.44479990005493 * x_5 + 6.20807981491089) - 1.08828319852829 * sin(4.23672008514404 * x_7 + 8.25799942016602)) + 0.112397667819086 * exp(0.0846979957553083 * x_{1}2 + 0.991672524366427 * x_2 + 16.4511034308287 * x_6 + 0.0794403354256281 * x_8 + 3.04175824992699e - 7 * exp(4.0479998588562 * x_{1}1) - 1.24447809117906 * sin(8.00008010864258 * x_1 - 7.80599975585938) + 0.109579705408338 * sin(7.22832012176514 * x_{1}0 - 7.79504013061523) - 2.18611613683477 * sin(4.73056030273438 * x_4 - 2.00808000564575) - 0.090498896259128 * sin(8.01175975799561 * x_5 + 6.74535989761353) - 0.696572994290733 * sin(1.77279961109161 * x_7 + 8.70903968811035) - 0.086369729858581 * sin(9.03264045715332 * x_9 - 3.13176012039185) - 6.21403559611692 * exp(-0.134959861636162 * x_3)) + 0.585080807839796 * sin(3.79584002494812 * x_1 - 2.264240026474) + 3.35590962744342 * sin(8.00008010864258 * x_1 - 7.80599975585938) - 0.295497036837222 * sin(7.22832012176514 * x_{1}0 - 7.79504013061523) - 0.0104218321340591 * sin(7.82839965820313 * x_{1}0 - 9.58400058746338) + 0.00114135295666917 * sin(6.98776006698608 * x_3 - 6.02040004730225) + 5.89516861912969 * sin(4.73056030273438 * x_4 - 2.00808000564575) - 0.150606323874081 * sin(2.44479990005493 * x_5 + 6.20807981491089) - 0.245980648028537 * sin(7.16328001022339 * x_5 - 8.83207988739014) + 0.244042960162737 * sin(8.01175975799561 * x_5 + 6.74535989761353) - 0.0144349445001026 * sin(8.03687953948975 * x_5 + 5.62815999984741) - 0.00178449741565884 * sin(9.59679985046387 * x_5 - 1.18351984024048) + 1.87840672674487 * sin(1.77279961109161 * x_7 + 8.70903968811035) + 1.71886403713931 * sin(4.23672008514404 * x_7 + 8.25799942016602) + 0.00149002523130288 * sin(5.20864009857178 * x_7 - 8.23416042327881) - 0.0290301723586934 * sin(8.40824031829834 * x_7 + 6.96704006195068) + 0.0297193521765595 * sin(3.10967993736267 * x_8 - 6.21591949462891) - 7.42675299615256 * sin(3.12784004211426 * x_8 + 9.43072032928467) - 0.415405447047944 * sin(3.1560001373291 * x_8 + 9.41520023345947) + 0.232908084125038 * sin(9.03264045715332 * x_9 - 3.13176012039185) - 78.8442748124012 * sin(0.00694879640191669 * x_{1}0 + 0.0131942323054101 * x_{1}2 + 0.0131930057259315 * x_3 - 0.172337413153575 * x_4 + 0.039533069963014 * (1 - 0.664667507905802 * x_9) * *2 - 0.897729475278538 * sin(7.89584016799927 * x_1 - 4.44848012924194) - 0.020820681266116 * sin(6.32735967636108 * x_2 + 9.20287990570068) + 0.0143079907340962 * sin(4.83655977249146 * x_5 + 0.801280200481415) + 36.8906728260918 * sin(7.04695987701416 * x_6 - 2.20424008369446) - 0.0334293809610244 * sin(1.44383955001831 * x_7 - 9.16384029388428) + 0.194241510160737 * sin(9.43496036529541 * x_8 - 6.31999969482422) + 32.1313612589991) + 0.0641252600281076 * sin(0.0403308117073846 * x_{1}0 + 0.0765793193460391 * x_{1}2 +$$

$$0.0765722002792057*x_3 - 1.00024628122899*x_4 + 0.229449923220285*(1 - 0.664667507905802*x_9)**2 - 5.21042153741058*sin(7.89584016799927*x_1 - 4.44848012924194) - 0.120843226250171*sin(6.32735967636108*x_2 + 9.20287990570068) + 0.083043572847906*sin(4.83655977249146*x_5 + 0.801280200481415) + 214.113451229834*sin(7.04695987701416*x_6 - 2.20424008369446) - 0.194024114544732*sin(1.44383955001831*x_7 - 9.16384029388428) + 1.12737765203336*sin(9.43496036529541*x_8 - 6.31999969482422) + 228.428914196474) - 59.9744453430176*sin(-0.00094571205746334*x_1 + 0.00154504134328363*x_10 - 0.00402434411662292*x_12 - 0.036542677034809*x_2 - 0.000854101907711542*x_3 - 0.144137536093254*x_4 - 0.677105061808072*x_6 - 0.0030345435733246*x_8 + 0.0168221380114972*(1 - 0.251976060154688*x_7)**2 - 0.00166646301593138*(1 - 0.761163582712494*x_9)**2 + 5.05744896871084e - 9*exp(3.87999963760376*x_11) - 1.16192207641979e - 8*exp(4.0479998588562*x_11) + 0.0129546906462557*exp(0.1872798204422*x_9) + 114663886.316496*exp(-0.318181224950382*x_1 - 0.113768275838319*x_10 - 0.250820632316221*x_12 - 1.48879420546227*x_2 - 0.0758834890445911*x_4 - 105.266596474702*x_6 - 0.0603058440791922*exp(0.869759798049927*x_9) + 0.20060099351349*sin(6.98776006698608*x_3 - 6.02040004730225) - 0.313638259235859*sin(9.59679985046387*x_5 - 1.18351984024048) + 0.261882654277088*sin(5.20864009857178*x_7 - 8.23416042327881) + 5.22338995869723*sin(3.10967993736267*x_8 - 6.21591949462891)) + 0.0104836581778504*sin(3.79584002494812*x_1 - 2.264240026474) + 0.0475378530754842*sin(8.00008010864258*x_1 - 7.80599975585938) - 0.00418583820211817*sin(7.22832012176514*x_10 - 7.79504013061523) + 0.000787232670175394*sin(7.82839965820313*x_10 - 9.58400058746338) + 0.0835075108637195*sin(4.73056030273438*x_4 - 2.00808000564575) - 0.00440755703783642*sin(7.16328001022339*x_5 - 8.83207988739014) + 0.0034569698449122*sin(8.01175975799561*x_5 + 6.74535989761353) + 0.00109037065234551*sin(8.03687953948975*x_5 + 5.62815999984741) + 0.0266084111031396*sin(1.77279961109161*x_7 + 8.70903968811035) + 0.00219284860930544*sin(8.40824031829834*x_7 + 6.96704006195068) - 0.133074848362329*sin(3.12784004211426*x_8 + 9.43072032928467) + 0.0313784308822473*sin(3.1560001373291*x_8 + 9.41520023345947) + 0.0032992397032048*sin(9.03264045715332*x_9 - 3.13176012039185) - 0.336378643873537*sin(-0.078974928336364*x_10 + 0.0173607241994499*x_12 + 0.642254225210899*x_2 + 0.0031870751617839*x_3 + 0.466824085350357*x_4 + 0.00886056920113713*x_8 - 827.881333192793*(0.0949923110198147 - x_6)**2 + 3.44555801546376*sqrt(1 - 0.0277508375039543*x_1) + 1.43891131338604*(1 - 0.415002474199746*x_9)**2 + 0.0992599585035379*sin(2.44479990005493*x_5 + 6.20807981491089) - 1.13284999335299*sin(4.23672008514404*x_7 + 8.25799942016602) + 0.45002265715685) + 1.5674624356659*sin(0.00694879640191669*x_10 + 0.0131942323054101*x_12 + 0.0131930057259315*x_3 - 0.172337413153575*x_4 + 0.039533069963014*(1 - 0.664667507905802*x_9)**2 - 0.897729475278538*sin(7.89584016799927*x_1 - 4.44848012924194) - 0.020820681266116*sin(6.32735967636108*x_2 + 9.20287990570068) + 0.0143079907340962*sin(4.83655977249146*x_5 + 0.801280200481415) + 36.8906728260918*sin(7.04695987701416*x_6 - 2.20424008369446) - 0.0334293809610244*sin(1.44383955001831*x_7 - 9.16384029388428) + 0.194241510160737*sin(9.43496036529541*x_8 - 6.31999969482422) + 47.8624017367518) + 0.0742629432676223*Abs(6.26327991485596*x_6 - 0.604319930076599) + 10.8645145102936 + 0.237370117857327*exp(-0.134959861636162*x_3)) - 0.983134920383671*Abs(6.26327991485596*x_6 - 0.604319930076599) + 150.618786570519 + 16.757018086616*exp(-0.134959861636162*x_3)$$

References

[1] P.Z. de Silva et al. Interference graph dataset for machine learning-based register allocation. *IEEE Access*, 2024.

[2] A. Dudáš and B. Modrovičová. Decision trees in proper edge k-coloring of cubic graphs. In *Proceedings of Conference of Open Innovation Association, FRUCT*. 2023.

[3] A. Dudáš and B. Modrovičová. Determining chromatic index of cubic graph with the use of explainable classifiers: A comparative study. *Journal of Applied Mathematics, Statistics and Informatics*, 2024.

[4] A. Dudáš and B. Modrovičová. Interpretable random forest model for identification of edge 3-uncolorable cubic graphs. *Kybernetika*, 2024.

[5] A. Michalíková et al. Can wood-decaying urban macrofungi be identified by using fuzzy interference system? an example in central european ganoderma species. *Scientific Reports*, 2021.

[6] A.A. Jones et al. Graph filter. *Zenodo*, 2023.

[7] G. Jayabalasamy et al. Application of graph theory for blockchain technologies. *Mathematics*, 2024.

[8] J. Ondriga et al. Generation of bicycle frame image designs using dcgan network. 2023.

[9] K. Coolsaet et al. House of graphs 2.0: A database of interesting graphs and more. *Discrete Applied Mathematics*, 2023.

[10] M. Kvet et al. Concept of temporal data retrieval: Undefined value management. *Concurrency Computation - Practice and Experience*, 2020.

[11] O.U. Ahmad et al. A graph machine learning framework to compute zero forcing sets in graphs. *IEEE Transactions on Network Science and Engineering*, 2024.

[12] P. Shaik et al. Analysis of multi-user-based uav system with outdated csi. *IEEE Opern Journal of the Communications Society*, 2024.

[13] S. Ghosh et al. Graph theory applications for advanced geospatial modelling and decision-making. *Applied Geomatics*, 2024.

[14] V. Beneš et al. Situation model of the transport, transport emissions and meteorological conditions. *Neural Network World*, 2024.

[15] X.M. Hu et al. Application of graph theory in liver research: A review. *Portal Hypertension & Cirrhosis*, 2024.

[16] Y. Caro et al. Remarks on proper conflict-free colorings of graphs. *Discrete mathematics*, 2023.

[17] Y. Peng et al. Predictive modeling of flexible ehd pumps using kolmogorov–arnold networks. *Biomimetic Intelligence and Robotics*, 2024.

[18] Z. Liu et al. Kan 2.0: Kolmogorov-arnold networks meet science. *arXiv*, 2024.

[19] Z. Liu et al. Kan: Kolmogorov-arnold networks. *arXiv*, 2024.

[20] L. Kowalik. Improved edge-coloring with three colors. *Theoretical Computer Science*, 2009.

[21] M. Kvet and K. Matiaško. Concept of dynamic index management in temporal approach using intelligent transport systems. *Advances in Intelligent Systems and Computing*, 2017.

[22] M. Kvet and K. Matiaško. Temporal transaction integrity constraints management. *Cluster Computing*, 2017.

[23] S. Masís. Interpretable machine learning with python. 2023.

[24] B. Modrovičová and A. Dudáš. Cubic graph property dataset for machine learning models. In *Proceedings of 17th IEEE International Scientific Conference on Informatics*. 2024.

[25] T.S. Rodrigues and P.R. Pinheiro. Hyperparameter optimization in generative adversarial networks (gans) using gaussian ahp. *IEEE Access*, 2025.

[26] SageMath. Mathematical software. *Available at: sagemath.org*, 2005.

[27] V.G. Vizing. On an estimate of the chromatic class of a p-graph. *Diskret. Analiz.*, 1964.

Bianka Modrovičová is a student at the Department of Computer Science, Faculty of Natural Sciences of Matej Bel University in Banská Bystrica, Slovakia and works in IBM Slovakia. She is the author and co-author of 8 research works and her research activities focus on the use of machine and deep learning models in graphical problems, specifically connected to proper edge coloring of graphs.

Adam Dudáš is an assistant professor at the Department of Computer Science, Faculty of Natural Sciences of Matej Bel University in Banská Bystrica. He is author and co-author of more than 35 research works and his research activities are related to descriptive, explorative and predictive data analysis with emphasis on visualization in the context of statistical analysis of data.