KYBERNETIKA — VOLUME *59* (2023), NUMBER 6, PAGES 807–826

INTERPRETABLE RANDOM FOREST MODEL FOR IDENTIFICATION OF EDGE 3-UNCOLORABLE CUBIC GRAPHS

Adam Dudáš and Bianka Modrovičová

Random forest is an ensemble method of machine learning that reaches a high level of accuracy in decision-making but is difficult to understand from the point of view of interpreting local or global decisions. In the article, we use this method as a means to analyze the edge 3-colorability of cubic graphs and to find the properties of the graphs that affect it most strongly. The main contributions of the presented research are four original datasets suitable for machine learning methods, a random forest model that achieves 97.35% accuracy in distinguishing edge 3-colorable and edge 3-uncolorable cubic graphs, and the identification of crucial features of graph samples from the point of view of its edge colorability using Shapley values.

Keywords: random forest, proper edge coloring, interpretable machine learning, snark

Classification: 68T20, 68T10

1. INTRODUCTION

Proper edge 3-coloring of a cubic graph is an NP-complete problem, which consists of assigning colors to the edges of the cubic graph in such a way that none of the vertices of the graph is incident to two edges colored with the use of the same color. From the point of view of edge coloring of a cubic graph, we can distinguish two groups of graphs – standard edge 3-colorable cubic graphs and snarks, or edge 3-uncolorable cubic graphs. The research presented in this article is focused on searching for properties and design of methods, which could be used to identify the number of colors needed to edge color a graph more effectively. Since the random forest method used in this article reaches a high level of accuracy in classification but is hard to interpret, we aim to use techniques of explainable artificial intelligence in order to understand how the model classifies the graphs into considered classes.

Graph coloring can be used in order to model several interesting practical problems. When compiling code from a high-level source language to machine-interpretable instructions, a method for correct and efficient assigning of variables used in the program to registers of the system can be carried out through the solution of graph coloring problem [3]. Scheduling problems such as scheduling of planes to flights during specified time

DOI: 10.14736/kyb-2023-6-0807

intervals without overlap, scheduling of a set of tasks to a set of processors while each task has to be executed on number of processors simultaneously or frequency assignment of radio stations without interferences are all typical instances of problem modeled by graph coloring [9].

The presented research contains a novel approach to the solution of the edge 3-coloring problem with the use of machine learning models. The contribution of the article can be summarized as follows:

- Presentation of graph property datasets usable in any machine learning model. This type of dataset is for the moment unique to our approach.
- Building of model based on interpretable random forest method which uses created datasets in order to classify input graph sets into one of two considered classes – properly edge 3-colorable graphs or improperly edge 3-colorable graphs.
- Evaluation of the model on the basis of its classification accuracy, precision, and interpretability of the decisions made. The interpretability of the model is important from the point of view of identification of properties that are significant in the context of graph edge coloring. If these properties can be computed in lower time complexity than edge coloring itself, we can obtain the edge colorability of the graph in a lower time.

In addition to the presentation of the problem of proper edge 3-coloring of cubic graphs and a brief overview of related works, which are both described in other subsections of this section, the presented work contains three chapters. Section 2 is focused on the description of the created graph property datasets, the tools used for data collection and computation of graph properties, and the specification of the interpretable model that uses the random forest approach in order to classify graphs into two categories based on their edge 3-colorability. The following section contains the evaluation of the model for graph classification, while we focus on the evaluation from the point of view of the prediction quality of the model and the interpretation of decisions made by the model. In the Conclusion section, we summarize the findings presented in the article, describe the strengths and weaknesses of the work, and offer several other directions for the development of research in the given area.

1.1. Edge coloring of cubic graphs

Graph G is described as pair of sets V (vertices) and E (edges) while [2]

$$G = (V, E), E \subseteq V^2.$$

We refer to the graph G as cubic if each of the vertices of G is incident to exactly three edges – every vertex is of degree equal to 3.

The edge coloring of a graph is a problem consisting of assigning colors to the edges of the graph. This coloring is called proper, in the case there is one instance of each color incident to every vertex, which makes such coloring an instance of an NP-complete problem. Figure 1 presents an example of edge coloring of the smallest cubic graph (K_4) which is properly edge 3-colorable (the left side of the figure) in comparison with the

graph (right side of the figure). As can be seen in Figure 1, there is coloring confPetersen graph, which is the smallest known instance of edge 3-uncolorable cubic lict solved with the use of the fourth edge color (twodased line).

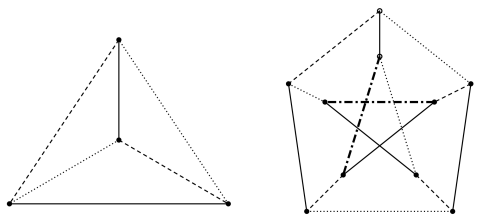

Fig. 1. Example of smallest cubic graph colored properly (L), and smallest know edge 3-uncolorable cubic graph (R).

Vizing's theorem [17] for graph G is formulated as

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$$

where $\chi'(G)$ is the so-called chromatic index of the graph G (minimal number of colors needed in order to properly edge color the graph G) and $\Delta(G)$ is the maximum degree of the graph G. From the point of view of this theorem, we consider two possibilities for edge coloring of cubic graphs – either with the use of three colors or edge coloring using four colors.

Therefore, there are two considered classes of cubic graphs from the point of view of edge 3-coloring:

- Standard cubic graphs, where $\Delta(G) = 3$ and $\chi'(G) = 3$.
- Snarks or edge 3-uncolorable cubic graphs, where $\Delta(G) = 3$ but $\chi'(G) = 4$.

1.2. Previous approaches to graph coloring

Over the years of research focused on the edge colorability of cubic graphs, several algorithms and optimizations of already existing algorithms have been proposed. The most basic approach to determining the edge colorability of a cubic graph is the naive backtracking algorithm, which works on the principle of the gradual coloring of the edges of the graph with a predetermined sequence of three colors. When the algorithm finds a contradiction in the coloring of the graph, it returns to the previous edge, which is recolored and the algorithm continues with the next coloring of the graph. If neither of the possible colorings of the problematic edge is proper, the algorithm returns to the edge preceding both recolored edges. The algorithm continues this procedure until the entire graph is properly edge colored or until the algorithm goes through all the possibilities of coloring the graph. The time complexity of edge backtracking is $O(2^{|E(G)|})$, where $|E(G)|$ is the number of edges in graph G.

Kowalik [8] proposed an optimization of older existing algorithms. The time complexity of the edge coloring using Kowalik's EdgeColor algorithm is $O(2^{0.427|V(G)|})$, where

$|V(G)|$ denotes the number of vertices of the colored graph G. The algorithm itself consists of $EdgeColor$, $FittingMatching$, and $SetSwitches$ procedures while using the approach of generating inclusionmaximum matchings of the graph.

Our previous work [6] examines the possibilities of increasing the efficiency of the computation of proper edge $k-$coloring of cubic graphs with the use of machine learning methods. The main focus of the research is the use of a machine learning model of decision trees for the problem of identification of properly edge 3-uncolorable graphs (snarks) while reaching an average accuracy of 93 %. In the presented work, we build on these results with the objective of increasing the accuracy and precision of the binary classification of cubic graphs.

	30 vertex		32 vertex		34 vertex		36 vertex	
	Min	Max	Min	Max	Min	Max	Min	Max
Clique number	2	3	2	3	2	2	2	2
Diameter	4	10	5	11	4	11	5	8
Edge connectivity	1	3	1	3	2	4	3	3
Matching number	14	15	15	16	17	17	18	18
Planar	0	0	0	0	0	1	0	0
Radius	3	8	4	7	4	8	4	6
Vertex connectivity	2	3	1	3	2	3	3	3
Largest L-eigenvalue	5,36	6	5,48	6	5,42	5,97	5,41	5,84
Second largest Eigenvalue	2,18	2.96	2,24	2,96	2,20	2,95	2,27	2,91
Smallest Eigenvalue	-3	-2	-3	-2,48	-2,86	-2,42	-2,84	-2,41
Laplacian spectrum	0,043	0,82	0,04	0,76	0,05	0,79	0,19	0,61
Chromatic number	3	3	3	4	3	3	3	3
Girth	3	7	3	6	4	7	5	7
Group size	1	3	1	4	1	384	1	2
Domination number	8	10	8	10	9	10	10	11
Independence number	10	15	13	16	14	16	15	17
Chromatic index	3	4	3	4	3	4	3	4

Tab. 1. Description of the range of measured graph property values.

2. MODEL FOR GRAPH CLASSIFICATION

Random forest is an ensemble method of machine learning, whose decision-making process consists of partial decisions of the decision trees forming this forest. When creating a random forest, a defined number of decision tree classifiers is created on subsets of the input data, which take care to preserve the principles of balance and purity of each tree. Every tree in the forest classifies the input data into one of the considered categories, while the final decision of the random forest is an aggregation of these decisions using a voting mechanism [5].

Since the random forest method exchanges the interpretability of a single decision tree for a high level of accuracy in decision-making, the local or global decisions of the model need to be interpreted through other devices [10].

This section of the article focuses on:

- description of graph property datasets for the random forest model and their collection using selected graph portals and software tools,
- description of the model that uses interpretable random forest methods to classify the created graph data into considered categories.

2.1. Graph datasets created for the model

Data suitable for the needs of building machine learning models, which are focused on working with graph properties, are not available. However, without such datasets, it is not possible for the model to learn to distinguish between the types of cubic graphs we are considering. Therefore, the first part of this work is focused on a collection of datasets that are created for this specialized purpose.

As the main source of data for interesting graphs (snarks and some of the standard cubic graphs), we used the *House of Graphs* portal [4]. However, the portal does not list the properties of the graphs in a consistent format, and, above all, it does not contain a sufficient number of standard cubic graphs that met our criteria. Therefore, we used the software tools *graphFilter* [7] and *SageMath* [14] to compute some graph properties and generate number of cubic graphs.

The presented datasets are structurally defined as follows:

- We have created four datasets containing measurements of properties of cubic graphs. These datasets differ from each other only based on the size of the graphs – we distinguish dataset for 30 vertex cubic graphs, dataset for 32 vertex cubic graphs, dataset for 34 vertex cubic graphs, and dataset for 36 vertex cubic graphs.
- Each of the created datasets contains the measurement of 27 graph properties. We can divide these properties into two basic groups:
  - Consistent properties usable in the identification of the basic properties of the graph, which are not usable in the actions associated with the proposed random forest model directly. There are 10 of these properties, such as the number of vertices of the graph, the number of edges of the graph, the density of the graph, the number of triangles in the graph, the average degree of the graph, and so on.

– Properties whose values range in certain intervals (see Table 1). These properties represent features for the decision-making process of random forest. There are 17 of these properties, our datasets contain measurements of [6]:
  * The clique number of a graph is the size of the largest complete graph that can be constructed out of the input graph.
  * Diameter of a graph denotes the length (number of edges) of the longest path in this graph.
  * Edge connectivity of a graph is the minimum number of edges, which can be deleted in such a way, that the graph is disconnected into more than one component.
  * Matching number of a graph is number of edges that do not share a set of common vertices.
  * Any input graph is planar in the case, we can visualize the graph on a plane without any edge, vertex or other graph component crossing each other. This property of a graph is called planarity.
  * Radius of a graph is the minimum graph eccentricity of any graph vertex in a graph. Such eccentricity of graph vertex is measured as the maximal number of edges between the vertex and any other vertex of the graph.
  * Vertex connectivity of a graph denotes the smallest number of vertices, the dropping of which causes the input graph to be disconnected into several components (discrete subgraphs).
  * Since a graph consisting of n vertices is commonly represented as a matrix $A = n \times n$ called adjacency matrix, we are able to measure eigenvalues of the adjacency matrix of a graph as one of the graph's properties. In this study, the largest, second largest and smallest eigenvalues of the adjacency matrix are considered.
  * Laplacian spectrum or algebraic connectivity of a graph is the second smallest eigenvalue of Laplacian matrix L for the input graph computed as $L = D - A$, where A is the adjacency matrix of a graph and D is a diagonal matrix containing degree of the vertex i on each position $D_{i,i}$.
  * The chromatic number of a graph is the minimal number of colors needed for the proper coloring of its vertices.
  * Girth of a graph is the number of edges contained in the shortest cycle (in the case of the existence of cycles) of the graph.
  * The group size of a graph is the size of the automorphism class for the given graph.
  * Domination number of a graph with the value of n is the smallest set of vertices, where every vertex not in the set, is adjacent to at least n vertices of the set.
  * The independence number of a graph is number of an independent set of vertices in the graph. Vertices are independent in the case they do not share common edges.
  * Chromatic index of a graph denotes the minimal number of colors needed in order to color the edges of the graph properly. This property is the

key to defining whether the cubic graph is a snark (chromatic index of 4) or standard cubic graph (chromatic index of 3) and serves as a class labelling for the approach used in this work.

- Each dataset consists of 500 cubic graphs while maintaining an even ratio between standard cubic graphs and snarks – 250 graphs for each class.

Since the work is focused on building a machine learning model of random forest, in Tables 2 – 5 we present the values of the prediction potential of the created datasets measured by Pearson and Spearman rank correlation coefficients.

Pearson correlation coefficient is focused on the linear prediction of values and the relationship between attributes A and B. In this case, the chromatic index of a graph is considered one of the attributes and the relationship with other graph properties is measured as follows [11]:

$$r(A,B) = \frac{\sum_{i=1}^{n}(A_i - \mu(A))(B_i - \mu(B))}{\sqrt{\sum_{i=1}^{n}(A_i - \mu(A))^2}\sqrt{\sum_{i=1}^{n}(B_i - \mu(B))^2}} \tag{1}$$

where $\mu(A)$ is the mean of attribute A, similarly $\mu(B)$ is the mean value of attribute B, and n is the number of records in dataset.

The correlation analysis methods for datasets containing non-linear relationships are necessitated by so-called rank methods, from which, the most representative is the Spearman rank correlation coefficient. This method is based on creating a ranking of individual attribute values for its functionality and therefore, we measure the monotonicity of the values within the attribute. Spearman rank correlation coefficient is computed as [1]:

$$\rho = 1 - \frac{6\sum(rank(A_i) - rank(B_i))^2}{n(n^2-1)} \tag{2}$$

where A_i and B_i are considered attributes of the dataset and n is the number of measurements of these attributes in the dataset.

In the tables, we list only the five highest values of correlation coefficients for each of the datasets.

In the presented tables, we can observe that the datasets containing 30 and 32 vertex graphs do not have any significant linear or non-linear prediction potential. In datasets of graphs with 34 and 36 vertices, we can observe stronger relationships between several graph properties and the chromatic number of graph.

This feature can be typical for the entire population of cubic graphs, but it can also be present only in the created sample. In further research, it is necessary to verify the connection between the prediction potential of graph property data and the chromatic index of the graph.

Pearson correlation	χ'
Clique number	0.48117262
Laplacian spectrum	-0.45993645
Second Largest Eigenvalue	0.45808661
Girth	-0.37773186
Edge connectivity	-0.30008138
Spearman rank correlation	χ'
Clique number	0.48117262
Laplacian spectrum	-0.34818507
Second Largest Eigenvalue	0.34753374
Diameter	0.3214077
Girth	-0.30625721

Tab. 2. Correlation analysis for 30 vertex graph property dataset.

Pearson correlation	χ'
Smallest eigenvalue	0.6471203
Largest L eigenvalue	-0.6450505
Girth	0.4261262
Edge connectivity	-0.3610132
Vertex connectivity	-0.3610132
Spearman rank correlation	χ'
Smallest eigenvalue	0.64680637
Largest L eigenvalue	-0.64561929
Girth	0.4621975
Edge connectivity	-0.38287723
Vertex connectivity	-0.38287723

Tab. 3. Correlation analysis for 32 vertex graph property dataset.

2.2. Model specification

After creating graph property datasets suitable for the needs of machine learning models, it is necessary to build a system that will be able to find patterns and properties in considered groups of cubic graphs. In this work, we focus on the use of an interpretable random forest model in order to classify the graph data.

Therefore, the proposed system consists of three main parts as shown in Figure 2:

- The input for the proposed model is the set of datasets described in Section 2.1, which are randomly divided into two data subsets – training data, on which the random forest model is learned, and testing data, which are used to evaluate the quality of the predictions of the created model. The ratio of the distribution of input data to the training and testing subset is 80 % to 20 %.

Pearson correlation	χ'
Girth	-0.95468871
Second Largest Eigenvalue	0.95213809
Laplacian spectrum	-0.95070542
Diameter	0.81449956
Vertex connectivity	-0.78006313
Spearman rank correlation	χ'
Girth	-0.93733609
Group size	0.89698202
Diameter	0.87716856
Second Largest Eigenvalue	0.85811785
Laplacian spectrum	-0.85280021

Tab. 4. Correlation analysis for 34 vertex graph property dataset.

Pearson correlation	χ'
Girth	-0.97644237
Laplacian spectrum	-0.91334022
Second Largest Eigenvalue	0.829689
Diameter	0.76745437
Smallest eigenvalue	0.7447873
Spearman rank correlation	χ'
Girth	-0.97644237
Laplacian spectrum	-0.85498749
Second Largest Eigenvalue	0.81567458
Diameter	0.79008435
Smallest eigenvalue	0.77449781

Tab. 5. Correlation analysis for 36 vertex graph property dataset.

- Both subsets of the input data are subsequently used in the classification module of the system. This module is based on a random forest classifier consisting of 50 decision trees – the number of decision trees was detected heuristically, based on the accuracy of classification of the graphs into the considered classes. The output of this module is the classification of the input graph into one of two classes – properly edge 3-colorable graphs (standard cubic graphs) or properly edge 3-uncolorable cubic graphs (snarks).
- Since the random forest model achieves a high accuracy of classification but the interpretability of its decisions is very complex, the proposed model includes an interpretation module. This interpretation of the decision-making process is performed based on the analysis of the contribution of individual graph properties to the final classification result. For these needs, we use the evaluation of individual features using Shapley values.

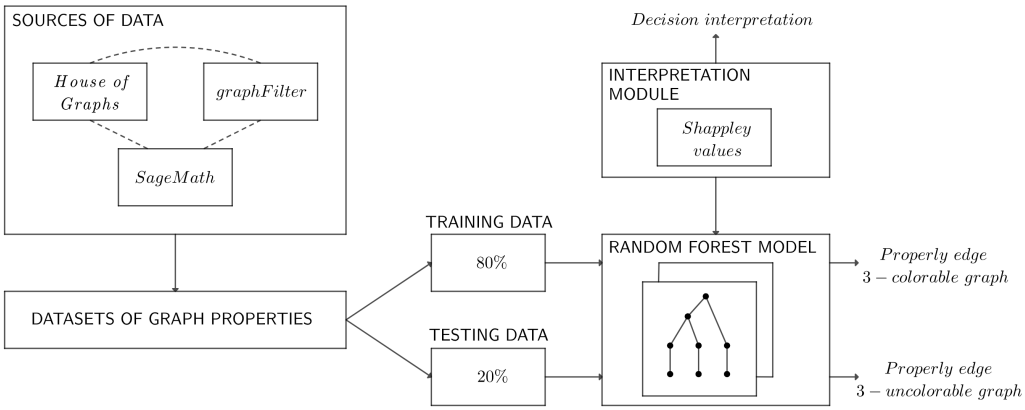

Fig. 2. Schema of interpretable random forest model for binary classification of cubic graphs based on edge colorability

3. EVALUATION OF A MODEL FOR GRAPH CLASSIFICATION

In this part of the article, we focus on evaluating the quality of the model from two perspectives:

- Evaluation of prediction quality of the created model – to measure the quality of random forest model predictions, we use two standard metrics – model accuracy and model precision.
- Interpretation of decisions made by the created model – to interpret the inner workings of the model, we consider Shapley values with the use of the Shapley Additive Explanations method.

3.1. Evaluation of prediction quality

In order to evaluate the decision-making quality of the created random forest model, we measured the classification accuracy and precision for each of the used datasets. By accuracy we denote the decision-making quality of the model as a whole – we measure the closeness of the predicted value to the real value of the feature [12]. In our case, this means that under accuracy we will understand the number of correctly identified edge 3-colorable cubic graphs and edge 3-uncolorable graphs in proportion to all decisions of the model. Precision refers to the ability of the created classification model to identify only the relevant entities [12]. In our case, we are considering the correctness of identifying the graph as edge 3-colorable and the correctness of identifying the graph as edge 3-uncolorable.

For us to be able to compute the accuracy and precision of the random forest model, we need to compile confusion matrices for all graph datasets. In Table 6, we present all four confusion matrices for the created model – the value False indicates a standard

cubic graph (properly edge 3-colorable cubic graph) and the value True indicates snark (properly edge 3-uncolorable cubic graph).

	30 vertex		**32 vertex**		**34 vertex**		**36 vertex**	
Predicted value	False	True	False	True	False	True	False	True
False	68	2	64	7	80	0	79	1
True	1	79	4	76	1	70	0	70

Tab. 6. Confusion matrices for the created random forest model containing 50 trees.

Accuracy is computed on the basis of the confusion matrix for each dataset as follows:

$$accuracy_S = \frac{t_n + t_p}{t_n + t_p + f_p + f_n}$$

where S is the number of vertices of graphs in the dataset, t_n is the number of true negative samples, t_p is the number of true positive samples, f_p is the number of falsely positive samples and f_n is the number of false negative samples.

The created random forest containing 50 decision trees reached average accuracy of binary classification equal to 97.35 %. The accuracy measured on individual datasets was:

$$accuracy_{30} = 98\,\%$$
$$accuracy_{32} = 92.72\,\%$$
$$accuracy_{34} = 99.34\,\%$$
$$accuracy_{36} = 99.34\,\%$$

Compared to previous research in the given area [6], this is an improvement of the average accuracy of classification by 4.35 %. Similar to the accuracy, the precision of the model is computed from the values of confusion matrices as follows:

$$precision_S(C) = \frac{t_p}{t_p + t_n}$$

where S is the number of vertices of graphs in the used dataset, C is considered class (in our case standard cubic graph or snark), t_p is the number of true positive samples and t_n is the number of true negative samples.

Random forest created in the presented research reached the following precision of classification:

$$precision_{30}(standard) = 97.14\,\%,\ precision_{30}(snark) = 98.75\,\%$$
$$precision_{32}(standard) = 90.15\,\%,\ precision_{32}(snark) = 95\,\%$$
$$precision_{34}(standard) = 100\,\%,\ precision_{34}(snark) = 98.59\,\%$$
$$precision_{36}(standard) = 98.75\,\%,\ precision_{36}(snark) = 100\,\%$$

With the average precision of the model equal to 97.3 %.

3.2. Interpretation of decision making

To interpret the decisions of the created model, we use Shapley values through the Shapley Additive Explanations technique. This method measures how individual graph

properties or sets of graph properties contribute to the overall quality of the created random forest model.

Even though the Shapley values are a technique of local interpretation of the model, by correct aggregation, we get a usable global interpretation of the model's decisions. The Shapley value ϕ for the i-th graph property (feature) is computed as [5, 10]:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \backslash i)]$$

where x is the specific graph considered by the model, f is the created random forest model, z' is the subset of features whose contribution to the decision is measured, x' is the simplified form of the graph x (sum of its' features) and M is the number of graph properties active in the used model.

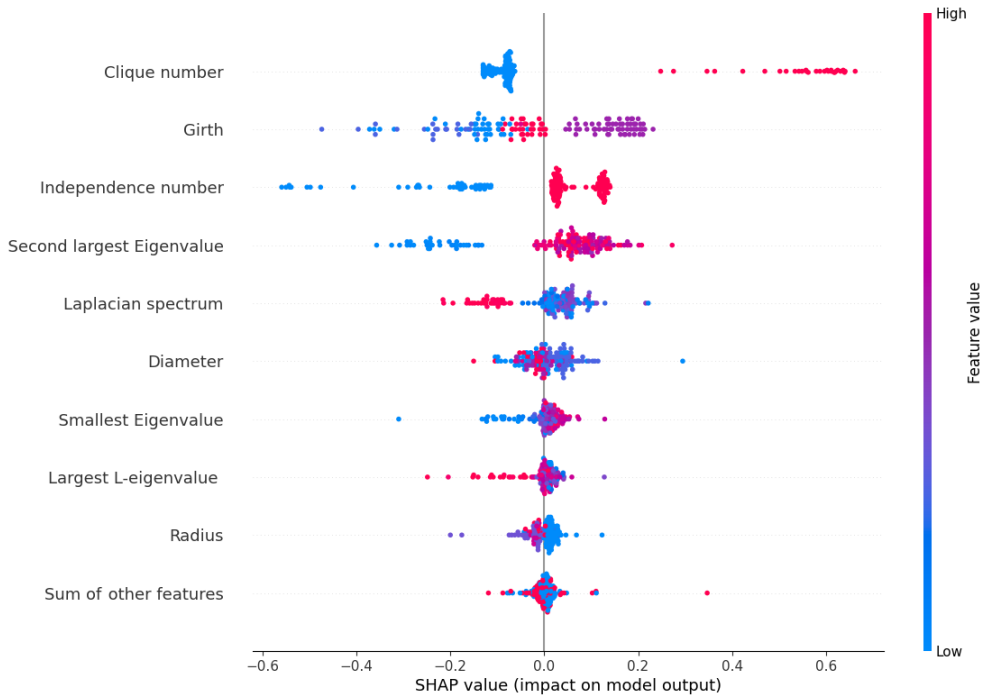

Fig. 3. Visualization of Shapley values for the set of 30 vertex cubic graphs.

With the use of Shapley values, we can measure the average contribution of a graph property value to the prediction in various combinations of features. For the set of 30 vertex graphs, the five properties most important from this point of view are (Figure 3):

- Clique number – Shapley values for this property form two groups. Low clique number values bring a weak negative contribution to the decision-making model (Shapley values between -0.05 and -0.2), while high values of this property contribute positively towards the decisions, even though their Shapley values are scattered.
- Girth of a graph – the situation with Shapley values is slightly more complex for the girth of a graph. From Figure 3 we can observe that the extremes of the value interval of this property contribute negatively towards the quality of the decision, but values near the average contribute positively.
- Independence number – with the values of this graph property, we see similar behavior as with clique number. The difference is that in the case of independence number, low values of the property are scattered more and the higher values are grouped in two clusters near the 0 Shapley value.
- Second largest eigenvalue of the adjacency matrix of a graph – the values of this property are – again – divided into two groups. Large and average values of the second largest eigenvalue of the adjacency matrix of a graph reach slightly positive Shapley values, while low values of the property represent low Shapley values.
- Laplacian spectrum – the situation with the Laplacian spectrum is reversed compared to the previous property – high values of the Laplacian spectrum are reflected in low Shapley values, and other values of the property are slightly positive.

Figure 4 presents a visualization of Shapley values for the set of 32 vertex graphs. The most crucial decision-making properties of the set are:

- Smallest eigenvalue of the adjacency matrix of a graph – when making decisions in this data sample we see the growing ambiguity of Shapley values. For the smallest eigenvalue of the adjacency matrix of a graph, we can observe that low values of the property bring negative Shapley values, high values of the property mean positive Shapley values, while the average values of the property are scattered throughout the interval.
- Girth of a graph – in the created model, the girth of a graph appears in the following way: high values of the property are scattered in the interval of positive Shapley values, and low values of girth reach negative or near-to-zero Shapley values.
- Largest L-eigenvalue of the adjacency matrix of a graph – high values of this property bring negative Shapley values, low values of the property reach positive Shapley values, and average values of Largest L-eigenvalue of the adjacency matrix of a graph are mostly slightly positive with outliers in negative.

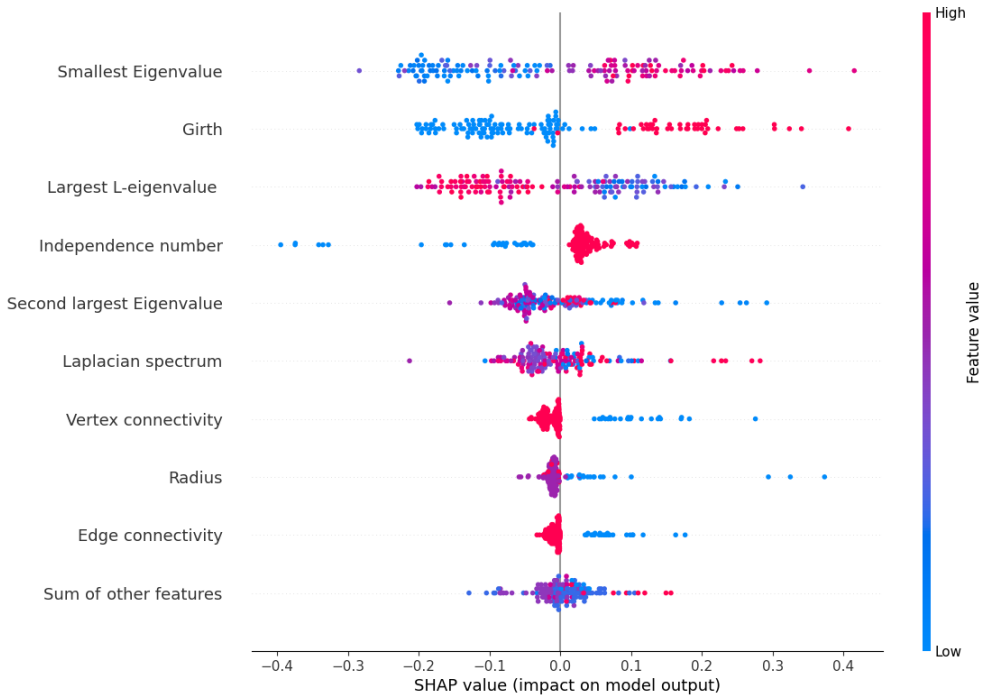

Fig. 4. Visualization of Shapley values for the set of 32 vertex cubic graphs.

- Independence number – with Shapley values of independence number we can see a clear separation of two groups of values. High property values are grouped on the positive side, and low independence number values are scattered on the negative side.
- Second largest eigenvalue of the adjacency matrix of a graph – Shapley values of this property are highly ambiguous. Average values of the property bring mostly negative Shaley values for the model, the rest of the values is scattered in the interval.

In Figure 5 we can see a slight diminution of ambiguity of Shapley values for a set of 34 vertex graphs compared to the previous set of graphs. The 5 most important properties of graphs are similar to the previous two cases:

- Girth of a graph – we can observe that the high values of the girth of a graph contribute negatively towards the quality of the decision, while low values and values near the average contribute positively.
- Laplacian spectrum – there is a clear separation of Shapley values for the Laplacian

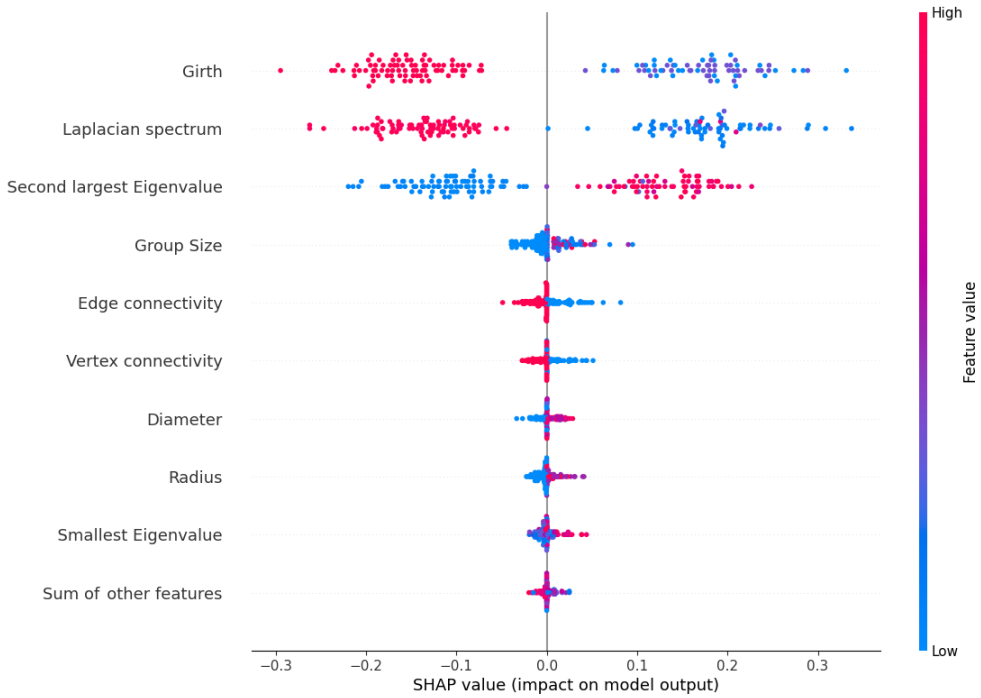

Fig. 5. Visualization of Shapley values for the set of 34 vertex cubic graphs.

spectrum. High values of the Laplacian spectrum are reflected in low Shapley values and vice versa. There are some outliers present on the positive side of the interval.

- Second largest eigenvalue of the adjacency matrix of a graph – similar to the Laplacian spectrum, for the second largest eigenvalue of the adjacency matrix of a graph we can observe a clear separation of Shapley values. High values of the property bring positive Shapley values, low values of the property bring negative Shapley values and both are scattered throughout the interval on their respective side. Similar to the previous cases, we can see a small number of outliers.
- Group size – in the created model, the Shapley values of group size form close to 0. Most of the low feature values are in the interval from 0 to -0.05, other low group size numbers are mixed with high values of the property and reach positive Shapley values.
- Edge connectivity of a graph – this graph property contributes low decision-making values to the model. From Figure 5 we see that the edge connectivity of a graph reaches lower Shapley values for high connectivity and higher Shapley values for the low value of this property.

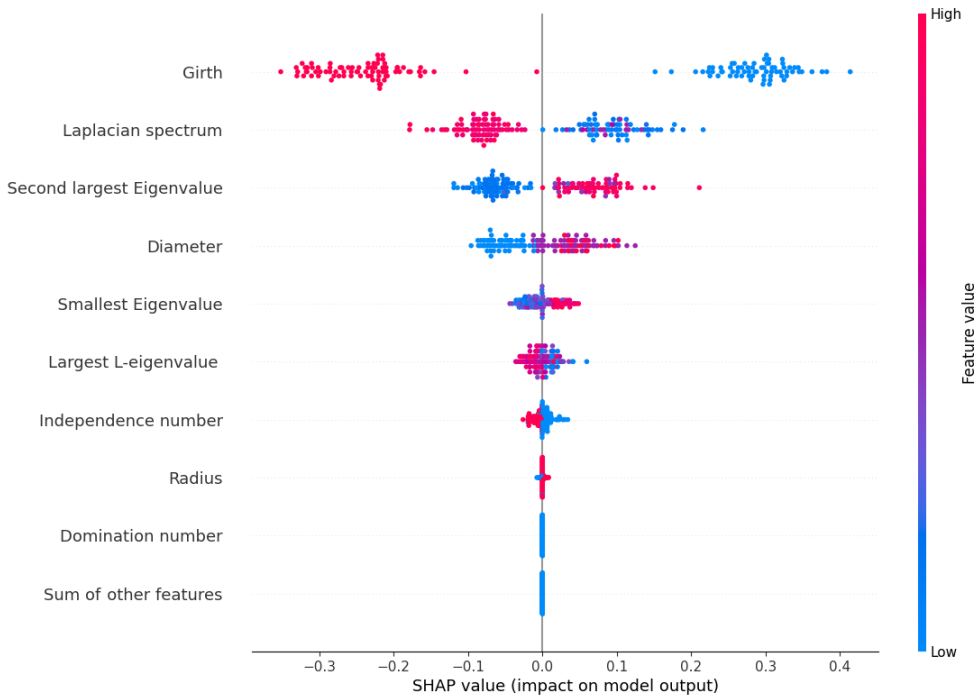

Fig. 6. Visualization of Shapley values for the set of 36 vertex cubic graphs.

The last considered dataset consists of cubic graphs with 36 vertices. Figure 6 presents the Shapley values for graph properties used in the decision-making process, the most important of which are:

- Girth of a graph – the first of the properties essential for decision-making of the created model is the girth of a graph. The Shapley values of this property form two separate groups, with high girth values reaching negative Shapley values and low property values being reflected in higher Shapley values. We can see a few outliers scattered among the two main groups of values.

- Laplacian spectrum – with this property, we can observe behavior similar to the girth of a graph, but with a much less pronounced separation of Shapley values. Low values of the Laplacian spectrum are also mixed with average values in the positive Shapley values.

- Second largest eigenvalue of the adjacency matrix of a graph – the Shapley values for this property are reversed compared to the previous two cases. High and average values of the property contribute positive Shapley values, while low values of the second largest eigenvalue of the adjacency matrix of a graph reach lower Shaley values.

- Diameter of a graph – the Shapley values of the diameter of a graph behave very similarly to the previous property with the exception of slightly negative Shapley values for some of the average diameter values.

- Smallest eigenvalue of the adjacency matrix of a graph – the Shapley values of this property are grouped close to 0, with low values of property achieving lower Shapley values and high values of the smallest eigenvalue of the adjacency matrix of a graph yielding higher Shapley values. Average values are scattered in this interval.

4. CONCLUSION

The presented research points to the possibility of applying prediction analysis techniques to the precise estimation of graph property values. Since most of the algorithms that are used in the measurement of graph properties do not use data mining methods, the potential of prediction analysis described in this article can be used in the optimization of the measurement of graph property values.

In the work, we show the application of interpretable random forests, which use several graph properties in order to classify cubic graphs into edge 3-colorable or edge 3-uncolorable sets. The approach achieves an average accuracy of classification equal to 97.35 %. Table 7 contains the time complexity of computation of various properties which reached the highest Shapley values in the described decision-making process and their comparison to the time complexity of standard edge coloring algorithms presented in [8, 13, 16]. As can be seen in this table, all used properties can be computed in a lower time complexity compared to an edge coloring of a graph.

Graph property	**Time complexity**
Eigenvalues of graph	$O(V)$
Diameter of graph	$O(V\sqrt{E})$
Girth, Radius, Matching number of graph	$O(VE)$
Edge connectivity of graph	$O(E + k^2 V ln(\frac{V}{k})$ for k edges
Edge coloring – naive backtracking	$O(2^{\lvert E(G)\rvert})$
Edge coloring – Beigel & Eppstein	$O(2^{\frac{\lvert V(G)\rvert}{2}})$
Edge coloring – Kowalik	$O(2^{0.427\lvert V(G)\rvert})$

Tab. 7. Time complexity of graph property computations.

In a dataset containing 30 vertex graphs, the graph property of clique number reached the highest Shapley values. The computation of this property represents – just like edge 3-coloring of the graph – an NP-complete problem. If we do not take this feature into account during classification, we can build a random forest classifier that achieves the results described in Table 8.

The accuracy of the model for 30 vertex cubic graphs after the exclusion of clique number from the dataset is 89.34 %, the precision of the standard cubic graph classification is 88.57 %, and the precision of the snark classification is 90 % with an overall

	False	True
False	62	8
True	8	72

Tab. 8. Confusion matrix for random forest model built on 30 vertex cubic graph dataset without clique number property.

precision of the model equal to 89.29 %. The ambiguity of Shapley values, which was present in previous measurements, is also reflected in this sample. All graph properties that reached high Shapley values are scattered in the considered interval. The distribution of Shapley values of this model is presented in Figure 7.

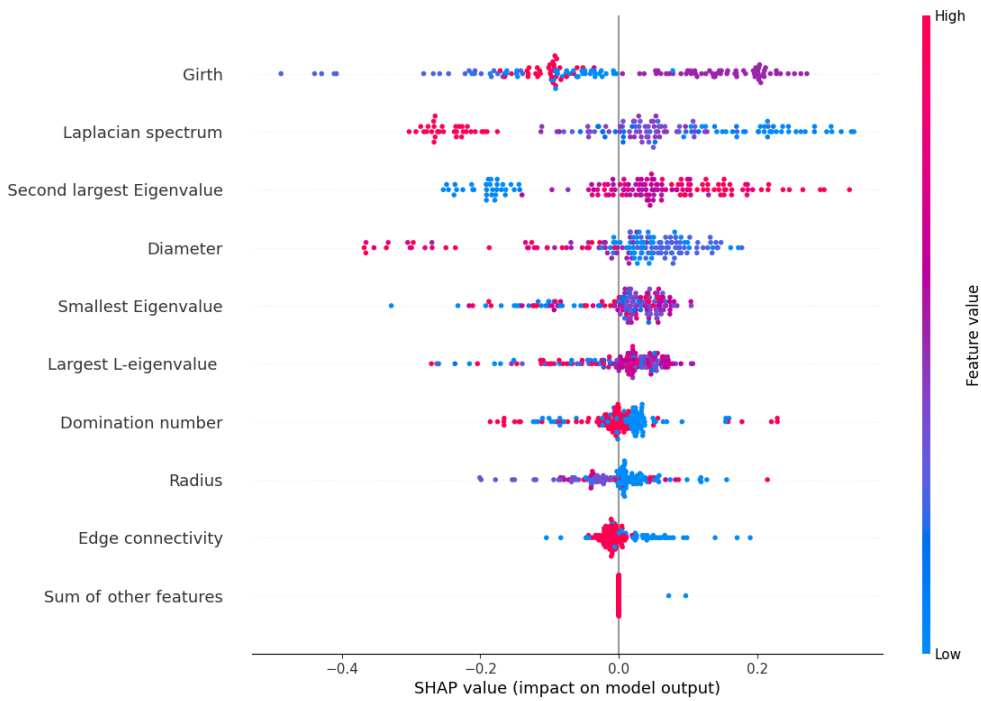

Fig. 7. Visualization of Shapley values for the set of 30 vertex cubic graphs without graph properties with high time complexity.

Graph properties, which were most influential for the decision-making process were the girth of a graph, the Laplacian spectrum of a graph and the eigenvalues of the adjacency matrix of a graph. The authors of [15, 18] specify these properties as having a relationship with the chromatic index for certain groups of graphs – most of which specify bounds on the chromatic index in relationship with one of the influential properties. The novelty of the findings of the presented study lies in the combination of these bounds

with the use of machine learning models in order to find deeper relationships between the cubic graph properties. As mentioned before, in further research, it is necessary to verify the connection between the prediction potential of graph property values and the chromatic index of the graph, and the way this potential behaves with the growing size of graphs.

Other future work in the research area is inspired by the fact that achieved results are measured on a relatively small sample of data. Therefore it is necessary to create similar graph datasets in the size of millions of graphs in the future.

The last of the objectives of future work is the design and implementation of an interpretable neural network for binary classification of larger sets of cubic graphs with the aim of increasing the accuracy and precision of the classification.

ACKNOWLEDGEMENTS

Computing was performed in the High Performance Computing Center of the Matej Bel University in Banska Bystrica using the HPC infrastructure acquired in project ITMS 26230120002 and 26210120002 (Slovak infrastructure for high-performance computing) supported by the Research & Development Operational Programme funded by the ERDF.

The support of Advtech_AirPollution project (Applying some advanced technologies in teaching and research, in relation to air pollution, 2021-1-RO01-KA220-HED-000030286) funded by European Union within the framework of Erasmus+ Program is gratefully acknowledged.

CODE AND DATA AVAILABILITY

The code and data used in the experiments presented in this study are freely available on the following link:

`https://github.com/AdamDudasUMB/cubicGraphData`

In the case of any questions don't hesitate to contact the authors of the work via e-mail adam.dudas@umb.sk.

(Received August 29, 2023)

REFERENCES

[1] H. Bon-Gang: Performance and Improvements of Green Construction Projects. Elsevier 2018.

[2] Y. Caro, M. Petrusevski, and R. Skrekovski: Remarks on proper conflict-free colorings of graphs. Discrete Math. *346* (2023), 2, 113221. DOI:10.1016/j.disc.2022.113221

[3] G. J. Chaitin: Register allocation & spilling via graph colouring. In: Proc. 1982 SIGPLAN Symposium on Compiler Construction 1982, pp. 98–105.

[4] K. Coolsaet, S. D'hondt, and J. Goedgebeur: House of Graphs 2.0: A database of interesting graphs and more. Discrete Appl. Math. *325* (2023), 97–107. DOI:10.1016/j.dam.2022.10.013

[5] L. L. Custode and G. Iacca: Evolutionary learning of interpretable decision trees. IEEE Access *11* (2023), 6169–6184. DOI:10.1109/ACCESS.2023.3236260

[6] A. Dudáš and B. Modrovičová: Decision trees in proper edge k-coloring of cubic graphs. In: Proc. 33rd Conference FRUCT Association 2023, pp. 21–29.

[7] GraphFilter: Software to help Graph researches providing filtering and visualization to a given list of graphs. Graphfilter 2021.

[8] L. Kowalik: Improved edge-coloring with three colors. Theoret. Computer Sci. *410* (2009), 38–40, 3733–3742. DOI:10.1016/j.tcs.2009.05.005

[9] D. Marx: Graph colouring problems and their applications in scheduling. Periodica Polytechn., Electr. Engrg. *48* (2004), 1–2, 11–16.

[10] C. Molnar: Interpretable Machine Learning. Published independently, 2019.

[11] D. Nettleton: Commercial Data Mining. Elsevier 2014.

[12] J. Rabčan, P. Rusnák, and S. Subbotin: Classification by Fuzzy decision trees inducted based on cumulative mutual information. In: Proc. 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET), 2018, pp. 208–212.

[13] L. Roditty, V. V. Williams: Fast approximation algorithms for the diameter and radius of sparse graphs. In: STOC '13, Proc. forty-fifth annual ACM symposium on Theory of Computing 2013, pp. 515–524. DOI:10.1145/2488608.2488673

[14] SageMath: Free open-source mathematics software system licensed under the GPL. Sagemath 2005.

[15] O. Suil, R. P. Jeong, P. Jongyook, and Z. Wenqian: Sharp spectral bounds for the edge-connectivity of regular graphs. Europ. J. Combinatorics *110* (2023). DOI:10.1016/j.ejc.2023.103713

[16] T. Tantau: Complexity of the undirected radius and diameter problems for succinctly represented graphs. Technical Report SIIM-TR-A-08-03, Universität zu Lübeck, Lübeck, Germany, (2008).

[17] V. G. Vizing: On an estimate of the chromatic class of a p-graph. Diskret. Analiz. *3* (1964), 25–30. DOI:10.1515/crll.1964.216.25

[18] H. Zhen-Mu, L. Hong-Jian, and X. Zheng-Jiang: Connectivity and eigenvalues of graphs with given girth or clique number. Linear Algebra Appl. *607* (2020), 319–340. DOI:10.1016/j.laa.2020.08.015

Adam Dudáš, Department of Computer Science, Faculty of Natural Sciences, Matej Bel University, Tajovského 40, 974 01 Banská Bystrica. Slovak Republic.
e-mail: adam.dudas@umb.sk

Bianka Modrovičová, Department of Computer Science, Faculty of Natural Sciences, Matej Bel University, Tajovského 40, 974 01 Banská Bystrica. Slovak Republic.
e-mail: bianka.modrovicova@student.umb.sk