Classification of Complex Networks Using Structural Analysis of Random Graph Models

1st Ali Baran TAŞDEMİR
Computer Science
Hacettepe University.
Ankara, TURKEY
alibaran@tasdemir.us

2nd Barkın ATASAY
Computer Science
Hacettepe University
Ankara, TURKEY
barkinatasay96@gmail.com

***Abstract*—Nowadays, we use networks to represent data in lots of areas like social relationships, brain activities, molecular bonds, and hyperlinks. Our study mainly focuses on network analysis which is a very popular topic in these days. All graphs have distinguishable features that specify their characteristics. By analyzing these features and combining them with machine learning techniques, we achieved a classifier that enables us to find the most suitable algorithm for achieving the most similar synthetic graph to real-world graph. We approached to this problem in 3 steps. Firstly, we extracted the features of the graph we work on then we obtained real-world networks from http://networkrepository.com/. Secondly, we produced random graphs with chosen random graph models to use as data. At the last step, we trained a machine learning model to classify random graph algorithms and we use that in order to predict the model which is used to generate real network. The result gives us the best random graph algorithm to create a synthetic graph which is the most similar to complex real world graph.**

***Index Terms*—Graph Theory, Machine Learning,Synthetic Random Graphs, Real World Networks ,Network Analysis,Classification.**

I. Introduction

Nowadays, complex networks are used in so many areas to represent complex or big data. Using network representation provides a more analytical perspective to raw data and makes easier to perform data analysis. That is why complex networks are very popular these days.

There are a lots of researches about networks. Researchers try to make network algorithms faster and more efficient to help the industry for the usage of networks.

In this paper, we have a different perspective on networks. Our main focus is, producing the most realistic synthetic network gathered by random graph producing algorithms. We use graphs in different domains from http://networkrepository.com/. The common features of graphs are all unweighted and undirected.

We combined the machine learning and graph theory to achieve our goal. The main steps of our approach are producing random graphs, mining graph's features and finally, applying some machine learning techniques with the feature data.

At the beginning of our research, we started working on real-world networks with various domains. With the help of softwares like Gephi [1] and Python library Networkx [2], we gathered some statistical information. We observed statistical information like density, max-min-average degree, diameter, average path length, triangle count, average eigenvector centrality and max k-core count can be used as a feature vector for identifying graphs.

At the next step, we focused on producing random graphs with well-known algorithms. For the sake of stability, we used real graphs node count and the average degree to use random graphs. For example, for the Chung-Lu model, we used that graph's degree of sequence. Graph producing was done with 100 graphs for each algorithm. So in total, we produced 400 graphs to use as a training data.

In the final step, we trained different machine learning algorithms with produced random graphs and test the model with a real-world graph. At the training phase, our model learns (almost perfectly) to classify random graph algorithms for any given graph and we predict the real-world graph's category with the trained model. That prediction gives the information of the most similar random graph generation algorithm for the tested real-world graph.

II. Previous Work

There have been several interesting and informative studies about Network Analysis and Graph Theory and we inspired by some of them. However, we especially take two studies as a role model for our study. These are [3] and [4].

In the first study we focused on, the problem was predicting the domain of arbitrary networks by using a small set of graph features. They investigated if they can classify synthetic graphs using some classification models with simple structural features. As a result, they saw that these graphs have an vital importance for classification. These actions taught us that complex networks from different domains have distinct structural properties that allows us to predict the domain of a new unseen network with high accuracy and synthetic graphs are trivial to predict the graph model is which used for generation. In detail, in order to achieve the best classification they select the most effective features: density, average degree, assortativity and maximum k-core so we followed the method but we added some more features too. One challenge that they faced was insufficient amount of data to make classification so they generated random graphs. Naturally, we did the same in our study.

When we take a look at the second study [4], they developed an approach for model selection for character networks by using machine learning techniques, motifs and eigenvalues. The configuration model, preferential attachment model, Erdös-Renyi model and Chung-Lu model were the considered ones. According to the results they had, the Chung-Lu model seems to be the most realistic model. Our study and this one is similar but they have some significant differences.We extended the research with different networks from various domains. We work with simple statistical features and 5-profile motifs with the help of a recent algorithm ”ESCAPE” [5]. Types and size of graphs and number of features are some of them. In addition, we used real world graphs from different domains and we increased the diversity of machine learning techniques.

Our contribution:

III. Methods and Experimental Design

In order to achieve our goal, we train a classifier by using the random graphs we generated and distinguishable features of them.In other words, our model learns the characteristic features which enables it to distinguish between the random graph models. In the last step, we behave a real world graph as a random graph and we use our classifier to predict which random graph algorithm which generates the most similar graph to chosen real world graph.

IV. Methodology

The main goal of the experiments is to get a random model that has the maximum similarity to the features of the real graph. We will achieve that goal by obtaining the statistical information about the networks. In detail, we will use density, maximum and average degree, maximum k-core number, average clustering coefficient, network diameter, average path length, number of triangles, average eigenvector centrality and 5-profile information of the network. Along with the standard measures of the graph we are using 5-profile information.

As we mentioned in previous sections, we use unweighted, undirected graphs for this experiment. Also, we use connected graphs to get the information of diameter, average path length, and 5-profile.

n-profile is a subgraph counting method. Subgraph is a graph that its vertex set is a subset of the larger graph and its edges set is a subset of the larger graph. For n-profile, we count the n vertex subgraphs in a graph with different edge combinations (motifs). In this work, we use only connected motifs with 5 vertexes.

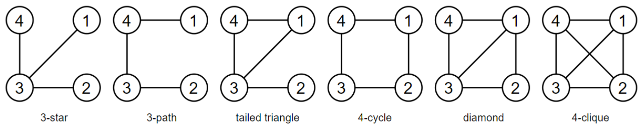

Fig. 1. 4-profile patterns

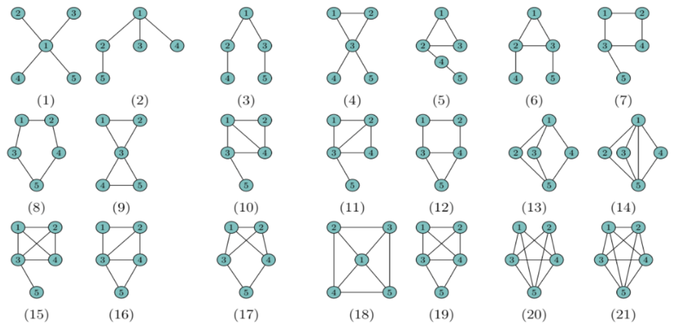

Fig. 2. 5-profile patterns [5]

We use the NetworkX package to calculate features excluding 5-profile. For 5-profile we use the” Escape Algorithm.”

A. Structural Features

*1) **Edge (E)**:* Set of lines which interconnects the nodes.

*2) **Vertex (V)**:* A graph is a set of points which called vertex.

3) Density: For undirected graphs, the graph density is defined as

$$D = \frac{2\,|E|}{|V|\,(|V|-1)}$$

4) Max.Degree: The degree of the vertex with the greatest number of edges incident to it.

5) Avg.Degree: The average degree of a graph G is another measure of how many edges are in set E compared to number of vertices in set V.

6) k-core of a graph: The k-core is the largest subgraph such that every vertex has degree at least k.

7) Avg. Clustering Coefficient: In graph theory, a clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together.

8) Diameter: It is the max eccentricity of any vertex in the graph.

9) Eccentricity: It is the maximum graph distance between and any other vertex of the graph.

10) Average Path Length: Average path length is a concept in network topology that is defined as the average number of steps along the shortest paths for all possible pairs of network nodes. It is a measure of the efficiency of information or mass transport on a network. –Wikipedia

*11) Triangles in a Graph****:* Let A[][] be adjacency matrix representation of graph. If we calculate A3, then the number of triangle in Undirected Graph is equal to trace(A3) / 6. Where trace(A) is the sum of the elements on the main diagonal of matrix A.

12) Average Eigenvector Centrality: Eigenvector centrality is a measure of the influence a node has on a network. If a node is pointed to by many nodes (which also have high Eigenvector centrality) then that node will have high eigenvector centrality.– Wikipedia

TABLE I
THE FEATURES FOR SELECTED NETWORKS

Graph	# Nodes	Edges	Density	Max Degree	Avg. Degree	Max k-core	Avg. Clustering Coefficient	Diameter	Avg. Path Length	# Triangles	Avg. Eigenvector Centrality
celegans-dir	453	2025	0.0198	237	8.94	10	0.64646	7	2.6638	3284	0.0311
macaque-rhesus_brain_2	91	582	0.1421	87	12.79	11	0.86005	3	1.8681	1902	0.0848
mouse_brain_1	213	16089	0.7126	205	151.07	111	0.75826	2	1.2874	622414	0.0677
ENZYMES_g300	49	93	0.0791	7	3.80	3	0.13780	16	5.8061	17	0.0854
ENZYMES_g540	49	92	0.0782	6	3.76	3	0.27551	17	6.1607	23	0.0880
soc-dolphins	62	159	0.0841	12	5.13	4	0.25896	8	3.3570	95	0.0908
firm-hi-tech	33	91	0.1723	16	5.52	5	0.45333	5	2.3598	77	0.1404

B. Network Data

We worked on networks in domains of brain, social, biology, and cheminformatics. In total, we have 7 different networks in 4 categories. All graphs are undirected, unweighted and connected. The graphs are obtained from http://networkrepository.com/.

Cheminformatics networks are about chemical information such as reactions, proteins, DNA, enzymes, etc... We use ENZYMES-G300 and ENZYMES-G540 networks that represent two enzymes named "Precerebellin" and "Phosphatidylethanolamine-N-Methyltransferase". For detailed information, you can read [6] [7].

Brain networks are the topology of nervous systems. We use macaque-rhesus-brain-2 and mouse-brain-1 graphs. These graphs are gathered by research named BigBrain [8] and edges represent fiber tracks in the mouse and macaque rhesus's brain. The all networks are a portion of the corresponding animals' brain.

Biological networks are networks that represent biological systems. We use only one graph named celegans-dir [9]. This graph represents a metabolic reaction in a worm name celegans (known as nematode).

Social networks are networks that represent the social connections and interactions between social livings. We use dolphins and firm-hi-tech networks. Dolphins network is a result of research [10] that researches dolphin packs' social relations with each other.

These networks selected for the experiment. The features of these networks are shown in Table 1 and we use these features and 5-profile information for machine learning.

The website http://networkrepository.com/ is a repository [11] for more than 30 domains of graphs with different sizes. We worked on this repository to obtain network data.

C. Synthetic Graph Models

We generated synthetic graphs by using four different graph models. These are: Erdős- Rényi, Chung-Lu , Preferential Attachment and Configuration models. Details about these models are explained below.

We examine four graph models with n nodes and other specific parameters :

*1) **Preferential Attachment(PA) graph model:*** This model is found by Barabasi and Albert in 1999. In this model, at each time a new node is created and connected to nodes which were existing before. At each step a node is created with the probability p which is equal to
$(\frac{2\,|E|}{n})$.

*2) **Erdős-Rényi (ER) graph model:*** This model has two parameters. n represents the nodes and p is the probability of the edges.We use $p = |E| \,/ \binom{n}{2}$ to match the average degree of the original network.

*3) **Chung-Lu (CL) model:*** The CL model generalizes the binomial random graph model to non-uniform edge probabilities. Graphs in this model are parameterized by an expected degree distribution (the character network's true degree distribution) rather than a scalar average degree. Each edge is connected with probability proportional to the product of the expected degrees w_i of its endpoints:

$$p_{ij} = \frac{1}{C} w_i w_j$$

*4) **The Configuration (CFG) graph model:*** In this model, we select a graph uniformly from the set of graphs which exactly match the target degree distribution. In practice, the degree distribution may vary slightly from the target since we disregard self loops and multi-edges created during this process.

D. Predictive Models

We used 5 different machine learning algorithms: support vector machine algorithm (linear classifier), random forest, AdaBoost , decision trees (ensemble methods) and K-Nearest Neighbors algorithm.

1) ***Support Vector Machines (SVM).*** SVM is a supervised machine learning algorithm for both classification and regression tasks. But, the algorithm is mostly used in classification tasks. In the algorithm, each data item represented as a point in n-dimensional space(where the n is the number of the features) and features are the coordinates of the point. The main idea of the algorithm is finding the best hyperplane that separates the points. This algorithm can be formulated quadratic with l1 or l2 regularization. Since we use more than two classes, we train a "one versus the rest" classifier for each random graph model.
2) ***Logistic Regression*** Logistic regression is a linear classifier that using logistic function $f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$. The result of the logistic regression algorithm is class probabilities.
3) ***Random Forest.*** This algorithm is an ensemble learning method for both classification and regression problems. It separates the data and constructs multiple decision trees. The result is a combination of these decision trees.

4) ***Boosted Decision Trees.*** Boosting is a technique to build a collection of weak learners and on top of them another tree that aims to correct other tree's mistakes. As random forests, the final result is the combination of these trees. We use the AdaBoost algorithm which is one of the most popular algorithms amongst them.

The classification model is trained with 400 networks from 4 different categories (IV-C) which is represented by a feature vector with the length D.

In the training process, the hyper-parameters of the classification model selected with stratified 5-fold cross-validation. Cross-validation is a re-sampling procedure to evaluate the machine learning algorithms on a subset of the main data source. Cross-validation mainly used on machine learning to test the model with an unseen dataset. We use 5-fold cross-validation for our model. That means we split the dataset into 5 pieces. At each iteration, we define one piece as a test dataset and the others as a training dataset. And stratified means that each set contains approximately the same percentage of samples of each target class as the complete set (see Fig. 3).

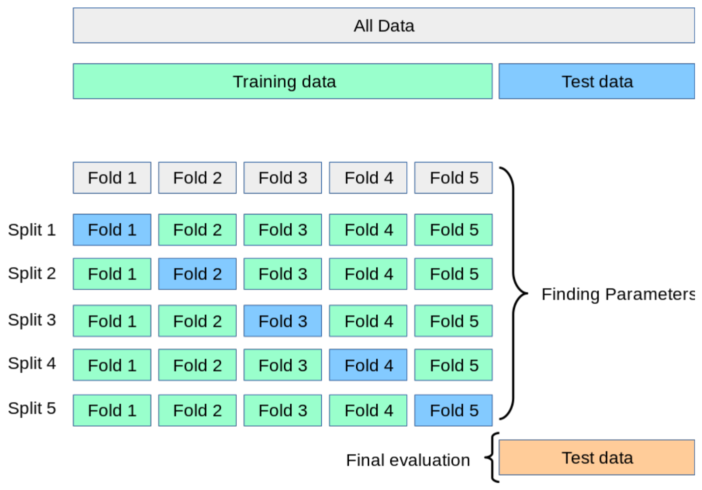

Fig. 3. 5-fold Cross Validation folding.

We choose the features which we use in feature vector with a process named ”Feature Selection”. We defined 3 different layers of features. On feature selection, we used 4 different techniques which are Chi-Square, Logistic Regression, recursive feature elimination (RFE) and Random Forest Classifier.

A chi-square test is a statistical method to calculate the independence of two events. We use this method to calculate our features independences from each other. For feature selection, we want to pick features that are highly dependent on the target(class).

Logistic Regression and Random Forest Classifier methods are predictive models. But for feature selection, we use these methods as tests. After training these models, each model determines a coefficient for every feature. So these features inform us about features in terms of 'importance'. The features get bigger coefficients if its more effect on the result.

RFE is a recursive feature elimination method to pick the most important features. This algorithm trains with the data and for every step drop the weakest feature and initiates the next step. This process continues until the algorithm reaches a given number of features left(hyper-parameter). The algorithm defines the weakest feature (like the Logistic Regression and Random Forest) with coefficients. The feature with the smallest coefficient value is defined as the weakest feature.

We combined 4 different techniques as a result we selected 3 different feature vectors with those results.

- **Vector-1:** This vector uses: Density, maximum degree, avg. degree, max. k-core, avg. clustering coefficient, avg. path length, total triangles, avg. eigenvector centrality and 5-profile motifs 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 15, 16, 17, 18, 19, 20 and 21 (Motif numbers in Figure 2).
- **Vector-2:** This vector uses: Maximum degree, max. k-core, avg. clustering coefficient, avg. path length, total triangles, avg. eigenvector centrality and 5-profile motifs 1, 2, 3, 6, 7, 11, 12, 16, 17, 18, 19, 20 and 21 (Motif numbers in Figure 2).
- **Vector-3:**This vector uses: Maximum degree, avg. clustering coefficient, avg. path length, avg. eigenvector centrality and 5-profile motifs 2, 6, 17, 20 and 21 (Motif numbers in Figure 2).

For 3 different vectors, there are some 5-profile motifs dropped. The main idea of the 3 layered droppings making a decision by comparing the results. Avoiding overfitting and the underfitting problem is a big concern here.

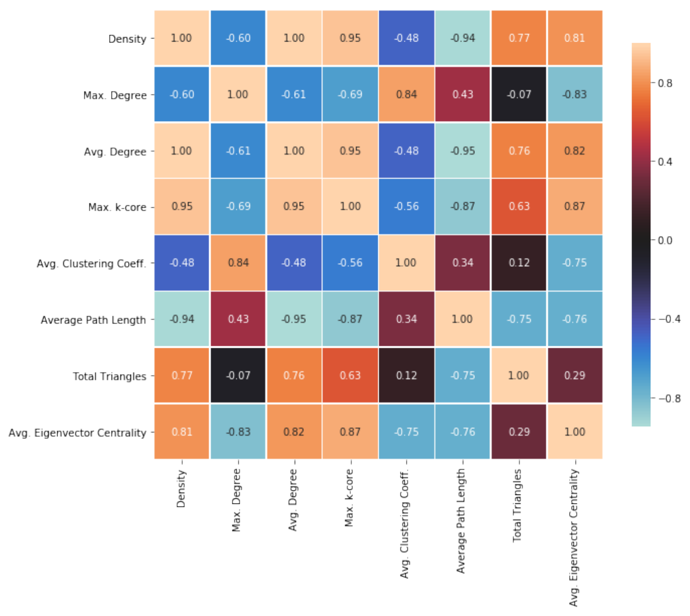

Fig. 4. ENZYMES G300 dataset correlation matrix with Vector-3 Features

V. Results

All experiments were performed with a system with 2.50 GHz Quad-core Intel i5 7300HQ, 8 GB RAM and Python

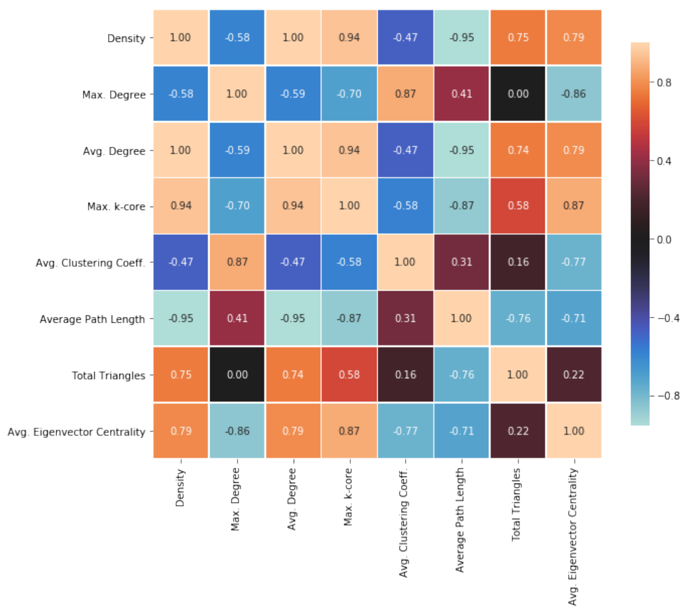

Fig. 5. ENZYMES G540 dataset correlation matrix with Vector-3 Features

3.7.5. The real-world networks are taken from networkrepository.com from a range of domains including social, brain, biological and cheminformatics. Python scikit-learn library [12] was used for the machine learning functions used in the experiments.

In the training phase, one of the most important questions answered thanks to our paper is ”*Can we distinguish the synthetic graph models from each other?*”. Before answering the question we should make a recall to Section IV. As we explained there, we create graph samples for each real-world network by using synthetic graph models we mentioned in Section IV-C. With those samples, we train 6 different machine learning models and we expect models to distinguish those synthetic graphs from each other. The figure shows the training accuracy for each network and algorithm. Models managed to learn to distinguish the synthetic graph models with very high accuracy for all the algorithms. All the algorithms can classify a generated graph with almost over 95% accuracy. This is an important result because it shows us these synthetic graph models produce networks with different characteristic features and we can accurately distinguish them with only a few features.

The results in Table-II shows us the F1 scores of the training with 4 different combinations of the feature vectors(mentioned in IV-D. We can see that the F1 scores of all combinations is higher than 0.95. So this answers the question ”Can we distinguish the random graph algorithms?”. But another critical result is that we can achieve high accuracy with less features with Vector 2 and 3. We prove that with less features we can get almost the same accuracy levels. So, some features is more important for this classification task.

With Table-II, we also see that the 5-profile motifs improve the training of the machine learning algorithm. By using 5-profile motifs, we achieve almost 1 point more score in training results. So we can decide to use this information for predicting the real-world network's prediction. The result gathered from this experiment is, *5-profile motifs improve the training accuracy for machine learning*. In other words, it represents valuable information about a synthetics network's origin algorithm. Considering these results, we can see the importance of feature selection for the classification task. As we see in the correlation matrix, our final features in Vector 3 has correlations equal to near 0. This shows us the features has neither positive or negative correlations with each other so there are no features which is related with others.

TABLE II
TRAINING SCORES WITH DIFFERENT FEATURE SETS.

	F1 Scores				
	Vector-1	**Vector-2**	**Vector-3**	**All Features**	**Without 5-profile**
CL	0.9686	0.9372	0.9217	0.9930	0.9802
CNFG	0.9849	0.9678	0.9634	0.9833	0.9695
GNP	0.9720	0.9358	0.9239	0.9783	0.9563
PA	0.9986	0.9988	0.9939	0.9976	0.9986
Average	0.9810	0.9599	0.9507	0.9881	0.9762

After training the machine learning model, we perform experiments on real-world networks. The main results we want to achieve are the predictions on real-world networks made by machine learning model trained with synthetic networks. To get those results, we determined a scoring system for each machine learning algorithm. For example, the scoring for SVM we calculate the distance of the seperating hyperplanes. In general, more positive score means more confidence in the prediction belongs to that class. The answer to the question *”Why we don't choose the 'best' machine learning algorithm and make predictions on that algorithm?”* is that we are not performing a classical machine learning classification task. In the first step, we performed trainings for classification of random graph models and we trained our model with randomly created networks with those models. However after that, we wanted to predict very different real-world networks with that trained model. The reason is we want to see the decision of the algorithm on the real-world network with the experience (training) on the random graph algorithms. Because of that, there is no true prediction on that testing phase. So we don't have the 'best' machine learning algorithm. We want to see some patterns for the decisions of those predictions.

In Table-III we have results of predictions for ENZYMES g300 and g540 networks with trained the machine learning model with Vector-3 features. Both of these networks are cheminformatics networks (Section IV-B) and we expect the same pattern of predictions before the experiment. As we see the results, we get the predictions for both networks as ”Preferential Attachment Algorithm (PA)” (For detailed information check Section IV-C). The result of these networks can be explained with Preferential Attachment in the algorithm. The preferential attachment process we can explain single molecules comes together with that approach and constructs more complex structures. (NEEDS MORE RESEARCH)

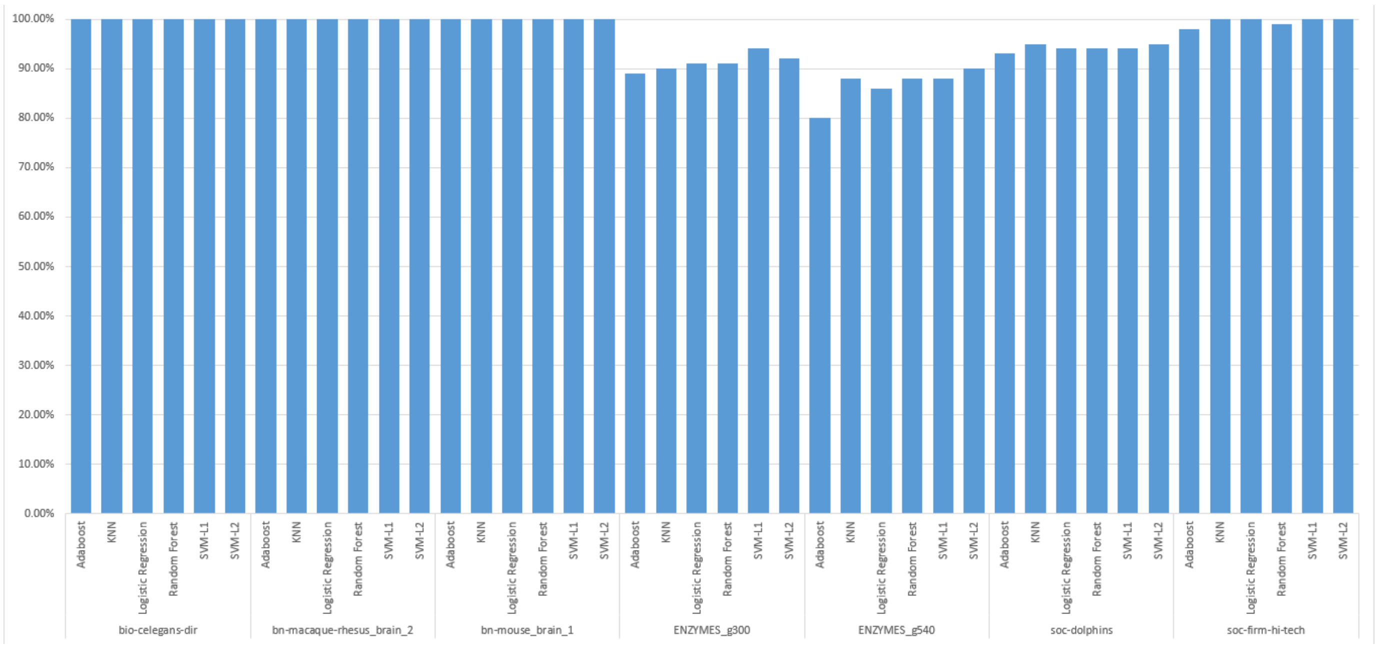

Fig. 6. Training accuracy

TABLE III
CHEMINFORMATICS NETWORKS RESULTS WITH VECTOR-3.

Graph	Classifier	CL	CNFG	GNP	PA
ENZYMES G-300	SVM L1	0.50	-48.38	-34.35	**8.95**
	SVM L2	1.98	-32.19	-34.69	**13.68**
	Logistic Regression	4.10	-21.32	-107.61	**24.18**
	Adaboost	17.94	-19.46	-21.18	**22.71**
	Random Forest	0.15	0.15	0.18	**0.52**
	K-Nearest Neigbors	0.0	0.0	0.0	**1.0**
ENZYMES G-540	SVM L1	-13.26	-13.97	-33.80	**7.16**
	SVM L2	-15.73	-17.43	-34.29	**17.53**
	Logistic Regression	-1.66	-22.58	-107.30	**30.52**
	Adaboost	3.31	-7.89	-5.37	**9.95**
	Random Forest	0.0	0.0	0.0	**1.0**
	K-Nearest Neigbors	0.0	0.0	0.0	**1.0**

In Table-IV we have results of predictions for dolphins network. This network represents a group of dolphins' social interactions with each other(Detailed in Section IV-B). The prediction result is the "Chung-Lu (CL) Model". We can explain these results with some features of the CL model. CL model takes the expected degree sequence and builds networks around this expected degrees. In social relations, we can say that these relations like friendships can be built in different combinations.

TABLE IV
SOCIAL NETWORK RESULTS WITH VECTOR-3

Graph	Classifier	CL	CNFG	GNP	PA
soc-dolphins	SVM L1	**21.45**	-9.97	-7.91	-6.64
	SVM L2	**16.75**	-5.10	-5.48	-2.10
	Logistic Regression	**41.59**	-14.27	-16.17	-5.95
	Adaboost	**8.50**	3.24	-7.35	-4.39
	Random Forest	**0.56**	0.26	0.04	0.14
	K-Nearest Neighbours	**1.0**	0.0	0.0	0.0

VI. CONCLUSION AND FUTURE WORK

In conclusion, all graphs have some structural features and we can distinguish the graphs by analysing these features. Of course, all features do not have the same significance , some of them are more valuable to consider them as a classification material.

Also, we can distinguish a generated graph for its algorithm. This leads us to result, generated graphs has some unique features that make them distinguishable. With that result, we can pick the most suitable random graph generation model for a particular network in any domain and create realistic synthetic graphs. In the future, with more real-world networks and more random graph algorithms, this method can be generalized for all networks and can produce realistic synthetics graphs for any network.

By completing all these tasks and achieving these results, we believe that we made a significant contribution to Computer Science and we hope to be a small part of the works which will be conducted in future.

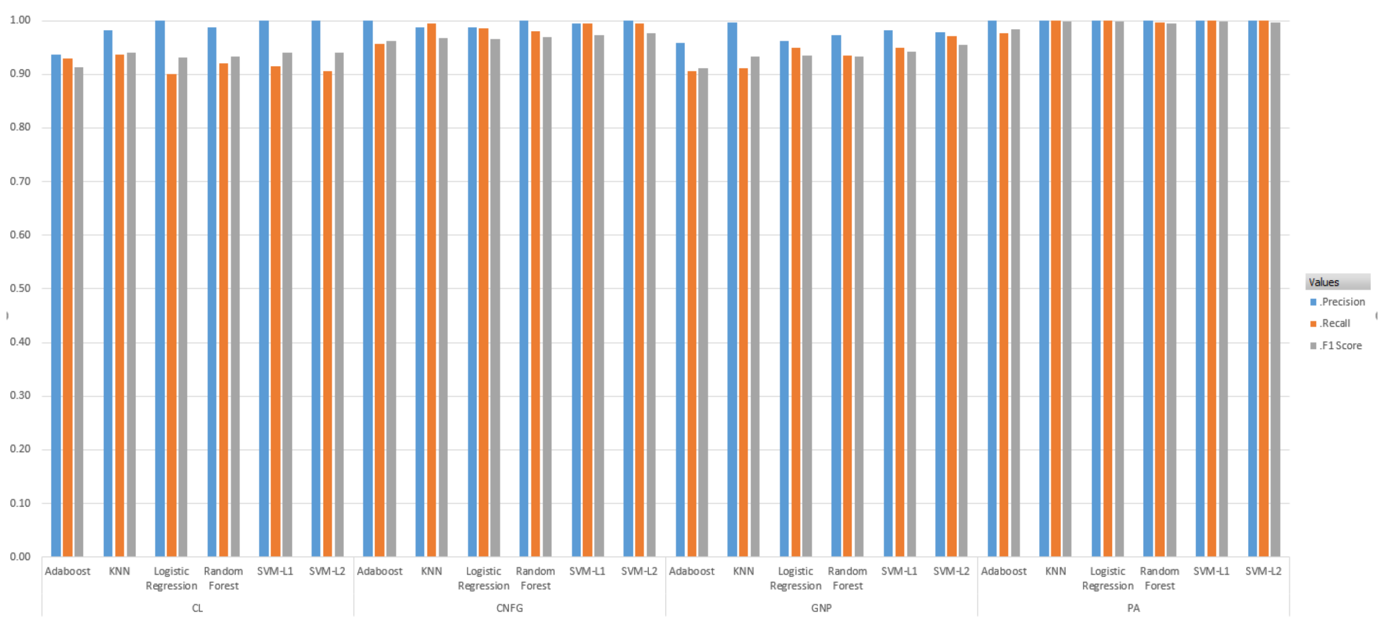

Fig. 7. Precision, recall and F1 scores for different synthetic graph models.

References

[1] "The open graph viz platform." [Online]. Available: https://gephi.org/

[2] "Networkx, python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks." [Online]. Available: https://networkx.github.io/

[3] R. A. Rossi and N. K. Ahmed, "Complex networks are structurally distinguishable by domain," *Social Network Analysis and Mining*, vol. 9, no. 1, p. 51, 2019.

[4] A. Bonato, D. R. D'Angelo, E. R. Elenberg, D. F. Gleich, and Y. Hou, "Mining and modeling character networks," in *International workshop on algorithms and models for the web-graph*. Springer, 2016, pp. 100–114.

[5] A. Pinar, C. Seshadhri, and V. Vishal, "Escape: Efficiently counting all 5-vertex subgraphs," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1431–1440.

[6] "Item no.: G300." [Online]. Available: http://www.cloud-clone.com/items/G300.html

[7] "Item no.: G540." [Online]. Available: http://www.cloud-clone.com/items/G540.html

[8] K. Amunts, C. Lepage, L. Borgeat, H. Mohlberg, T. Dickscheid, M.-É. Rousseau, S. Bludau, P.-L. Bazin, L. B. Lewis, A.-M. Oros-Peusquens, N. J. Shah, T. Lippert, K. Zilles, and A. C. Evans, "Bigbrain: An ultrahigh-resolution 3d human brain model," *Science*, vol. 340, no. 6139, pp. 1472–1475, 2013.

[9] J. Duch and A. Arenas, "Community identification using extremal optimization phys," *Rev. E*, vol. 72, p. 027104, 2005.

[10] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.

[11] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *AAAI*, 2015. [Online]. Available: http://networkrepository.com

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.