ISCTE IUL
REPOSITÓRIO
INSTITUTO UNIVERSITÁRIO DE LISBOA

Repositório ISCTE-IUL

Deposited in *Repositório ISCTE-IUL*:
2026-04-01

Deposited version:
Accepted Version

Peer-review status of attached file:
Peer-reviewed

Citation for published item:
Yamaguchi, C., Stefenon, S., de Paz Santana, J. F. & Leithardt, V. (2025). Are graph neural networks better than standard classifiers?. In Daniel H. de la Iglesia, Juan F. de Paz Santana, Alfonso J. López Rivero (Ed.), Proceedings of the 5th Int. Conference on Disruptive Technologies, Tech Ethics and Artificial Intelligence (DiTTEt 2025). (pp. 36-47). Salamanca: Springer.

Further information on publisher's website:
10.1007/978-3-031-99474-6_4

Publisher's copyright statement:
This is the peer reviewed version of the following article: Yamaguchi, C., Stefenon, S., de Paz Santana, J. F. & Leithardt, V. (2025). Are graph neural networks better than standard classifiers?. In Daniel H. de la Iglesia, Juan F. de Paz Santana, Alfonso J. López Rivero (Ed.), Proceedings of the 5th Int. Conference on Disruptive Technologies, Tech Ethics and Artificial Intelligence (DiTTEt 2025). (pp. 36-47). Salamanca: Springer., which has been published in final form at https://dx.doi.org/10.1007/978-3-031-99474-6_4. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

Use policy

Creative Commons CC BY 4.0
The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Serviços de Informação e Documentação, Instituto Universitário de Lisboa (ISCTE-IUL)
Av. das Forças Armadas, Edifício II, 1649-026 Lisboa Portugal
Phone: +(351) 217 903 024 | e-mail: administrador.repositorio@iscte-iul.pt
https://repositorio.iscte-iul.pt

Are graph neural networks better than standard classifiers?

Cristina Keiko Yamaguchi[1], Stefano Frizzo Stefenon[1], Juan Francisco De Paz Santana[2], and Valderi Reis Quietinho Leithardt[3]

[1] Postgraduate Program in Productive Systems in Association with UNIPLAC, UNC, UNESC, and UNIVILLE, SC, Brazil
{cristinayamaguchi,stefano}@uniplaclages.edu.br
[2] Expert Systems and Applications Lab, Faculty of Science, University of Salamanca, 37008 Salamanca, Spain
fcofds@usal.es
[3] Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, Lisboa, Portugal
valderi.leithardt@iscte-iul.pt

Abstract. Graph neural networks (GNNs) are becoming very popular these days due to their ability to perform classification and prediction depending on node connections. Since features of samples belonging to the same class can be related, graph-based models may perform classification better than other classifiers. The big challenge for this evaluation is to know if there is a sufficiently adequate relationship between the connections of the nodes to justify the use of these models, since connections to unrelated classes can reduce the capacity of these models. This paper proposes a thorough comparative evaluation between graph models and other well-established classifiers to assess the extent to which GNNs may be superior. This will be done by evaluating the relationships between the probability of connections between nodes and changing database features. The evaluation is performed using synthetic data, which is a task that can be evaluated in future work. The results show that when there are connections of classes different from the node under evaluation, the GNNs lose their advantages over other classifiers.

Keywords: Graph neural networks · graph attention networks · graph convolutional networks · classification.

1 Introduction

Nowadays, the use of logical reasoning to create more robust models has been widely applied [1]. Through models that use more information related to the features of the problem, they tend to have better classification capabilities. Graph neural networks (GNNs) play an important role in these new challenges because, besides using the sample's features, they consider its connections with other samples, called nodes in graphs [2].

As long as there are no connections between nodes, the problem becomes a standard classification problem where the classification is only based on the

sample features. When the problem becomes standard, the GNNs tend to lose their classification advantages over well-established classifier algorithms. Within this hypothesis, few works evaluate the performance of GNNs concerning the connections between their nodes [3].

For this reason, this paper proposes to evaluate the influence of the probability of connections between nodes in relation to its performance, aiming to verify how much the GNN structure excels for classifications when there are a larger number of connections between distinct classes. From this work, it will be possible to determine when GNNs are worth applying to these problems. Observing that in some cases connections can represent a reduction in classification performance if they do not make logical sense [4].

In this paper, two of the main GNN structures are compared with other well-established classifiers. The GNNs used in this comparison are the graph convolutional network (GCN) proposed by Kipf and Welling [5], and the graph attention network (GAT) proposed by Veličković *et al.* [6]. The GCN is a semi-supervised network that is similar to the convolutional neural networks that perform convolutions on images, in view that the filter parameters are generally shared across all locations on the graph. The attention on GAT represents a weighted average of multiple elements, with the weights computed dynamically based on an input query and element keys. The contributions of this paper to graph neural networks are outlined below:

- The first contribution is related to the definition of a threshold value for connections to other classes in graph-based problems. Connections to classes other than the class of the evaluated node have a negative influence on the performance of graph-based models and can make these models inefficient.
- The second contribution is related to evaluating the influence of varying the size and features of a database for applying GNNs. In many classifier models, the variation in the dataset makes the classifier inadequate due to the difficulty in network training when the problem is considerably complex.
- The third contribution is related to proposing a synthetic data generator for graphs that can be used for benchmarking, having variability in the degree of complexity depending on the user's needs. Many models are validated on well-known data sets, sometimes hindering other applications.

The remainder of this work is organized as follows: In Section 2, the method used for generating the database is described as well as the applied models. Section 3 evaluates the influence of changing model parameters and using different database configurations. Finally, Section 4 presents a conclusion.

2 Methodology

The GAT and GCN models are used in this paper for comparison against well-established classifier models, which are not based on graphs. In this paper, the classifiers logistic regression (LR) [7], linear discriminant analysis (LDA) [8], k-nearest neighbors (KNN) [9], decision tree classifier (DTC) [10], gaussian naive

bayes (NB) [11], and support-vector machine (SVM) [12] are used. These models are trained based on synthetically generated features. In this way, the compared classifiers consider the features of the nodes and not their connections. This is a major difference between the generally used classifier and the graph-based models that also consider the node connections. In this paper, the synthetic data generation and the evaluated models were written using Python language.

2.1 Graphical Attention Networks

GAT proposed by Veličković *et al.* [6] are among the most popular GNN architectures, being state-of-the-art models for learning graph representation. The graphical attention layer creates a message for each node using a linear layer matrix. For the attention, the GAT uses the node's message as a query, and the messages to the average as keys and values.

GAT employs masked self-service layers to address the weaknesses of previous methods based on graph convolutions. By stacking layers where nodes can attend over the characteristics of their neighbors, it is possible to specify different weights of different nodes in a neighborhood without requiring an expensive matrix operation or relying on prior knowledge of the structure [13]. The nodular formulation of the GAT is given by:

$$\mathbf{x}'_i = \alpha_{i,i}\mathbf{\Theta}\mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j}\mathbf{\Theta}\mathbf{x}_j, \tag{1}$$

where the attention coefficients $\alpha_{i,j}$ are calculated as:

$$\alpha_{i,j} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top[\mathbf{\Theta}\mathbf{x}_i \,\|\, \mathbf{\Theta}\mathbf{x}_j]\right)\right)}{\sum_{k \in \mathcal{N}(i)\cup\{i\}} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top[\mathbf{\Theta}\mathbf{x}_i \,\|\, \mathbf{\Theta}\mathbf{x}_k]\right)\right)}. \tag{2}$$

The normalized attention coefficients are used to compute a linear combination of the features corresponding to them, to serve as the final output for each node [14]. When the graph has multidimensional edge features $\mathbf{e}_{i,j}$, the attention coefficients $\alpha_{i,j}$ are given by:

$$\alpha_{i,j} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top[\mathbf{\Theta}\mathbf{x}_i \,\|\, \mathbf{\Theta}\mathbf{x}_j \,\|\, \mathbf{\Theta}_e\mathbf{e}_{i,j}]\right)\right)}{\sum_{k \in \mathcal{N}(i)\cup\{i\}} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top[\mathbf{\Theta}\mathbf{x}_i \,\|\, \mathbf{\Theta}\mathbf{x}_k \,\|\, \mathbf{\Theta}_e\mathbf{e}_{i,k}]\right)\right)}. \tag{3}$$

2.2 Graph Convolutional Network

GCNs are a class of neural networks designed to operate directly on graph-structured data, effectively capturing the relationships and features of nodes within a graph. They extend the concept of convolution from traditional convolutional neural networks, enabling the processing of data represented as graphs [15]. The operation of a GCN layer is defined as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{4}$$

where $H^{(l)}$ is the feature matrix at layer l, with $H^{(0)} = X$ being the input feature matrix. $\tilde{A} = A + I$ is the adjacency matrix with added self-loops. $\tilde{D}$ is the degree matrix of $\tilde{A}$, with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $W^{(l)}$ is the weight matrix for layer l. σ is the activation function [16].

2.3 Synthetic Data

Through the synthetic generation of the data, it is possible to create a problem with a higher level of difficulty, thus making it suitable for benchmarking [17]. It is possible to make the problem even more difficult by reducing the relationship between the features by increasing the standard deviation of the clusters. The synthetic generation of the data was carried out in two stages. In the first stage, the sample features were generated using the function `make classification` from `sklearn` library.

Another function that is also available from this library is the `make blobs`, where it is also possible to define the standard deviation of the clusters. In this step, it is defined how many samples, classes, and features are used. The number of features and the number of samples define the size of the feature matrix used to perform the benchmarking of the classifiers.

For an evaluation of the database configuration parameters, a standard dataset is initially fixed with the same number of samples, features, and classes. This way, it is possible to compare the influence of changing the graph configuration parameters one by one. In this evaluation to standardize the comparisons, 1,000 samples distributed in 3 classes with 8 features per sample were used. A representative image of the data distribution is presented in Figure 1, considering the relationship between 2 features of the 8 used in this evaluation.

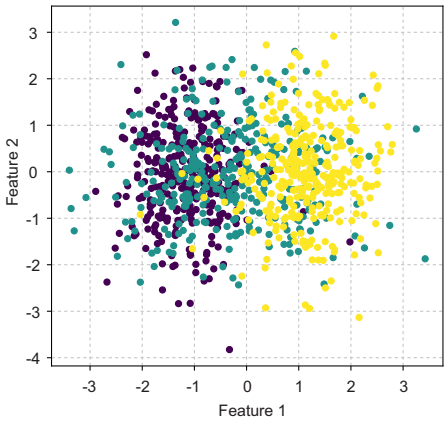


Fig. 1. Features generated by synthetic data for 3 classes.

In the second step, the graph is created, in which connections between samples are defined; after that, the samples are called nodes. To create the graph, the function `stochastic block model` from `networkx` library was used. In this evaluation, the community sizes are given by the distribution of the classes from the result of the `make classification` function, thus, the communities have the same size as the sample classes.

To generate the connections between the nodes, a probability matrix is created, which determines the probability of a sample being connected to another. This variation will be evaluated in this paper to verify the influence of connections on the performance of graph-based models. In the probability matrix, the position of the elements of the main diagonal defines the probability that a node is connected to other nodes of the same class, where η is the number of classes used, previously defined for the generation of features.

For future evaluations, the algorithm to create synthetic data is available at: https://github.com/SFStefenon/SyntheticData4Classification.

3 Analysis of Results

This section presents the simulation results for comparing GNN models with other well-established classifiers. For a data selection that does not depend on its distribution, the selection of nodes for training and testing is done randomly. After the data selection, the database is created. All models use the same database for performance comparison.

For a complete evaluation of the classification performance of the graph-based models, in the first evaluation, an identity matrix is used to generate the probabilities between the graph connections. This is an ideal case, in which all nodes of the same class are connected without connections between other classes. Since in the GAT model, variation in the number of multi-head attentions (MHA) may lead to a change in the results, Table 1 presents the analysis of the variation of this parameter to define a standard configuration for comparison.

Table 1. Evaluation of variation in the number of multi-head attentions.

MHA	Accuracy by number of MHA output					
input	1	3	5	8	10	15
1	0.4252	0.6611	0.7276	0.9767	0.9668	0.9668
3	0.8671	0.9635	0.9834	0.9834	0.9900	0.9801
5	0.9535	0.9767	0.9934	0.9635	0.9934	0.9900
8	0.9934	0.9967	0.9834	0.9701	0.9867	0.9999
10	0.9601	0.9967	0.9967	0.9934	0.9834	0.9767
15	0.9668	0.9934	0.9867	0.9834	0.9900	0.9934

Using fewer MHA results in lower computational cost. From the use of 8 MHA, all results were satisfactory, even with a smaller number in the output layer of the network. For this reason, 8 MHA were set as the default for the GAT

model comparison. Since the same dataset was used for this evaluation, similar results can be considered equivalent due to the random characteristic of model initialization.

The GCN and GAT approaches are compared to well-known classifiers to evaluate the influence of changing database features on model performance. The results of this analysis are presented in Table 2. Besides model evaluation, the ratio between the percentage of data used for training and testing is evaluated.

Table 2. Performance of classifiers in an ideal graph case.

Classifier	Accuracy given by train/test ratio (%)				
	50/50	60/40	70/30	80/20	90/10
LR	85.60	84.83	85.12	85.00	86.56
LDA	84.20	83.33	83.83	83.00	83.89
KNN	89.00	87.83	88.41	88.25	88.67
DTC	85.40	82.83	87.26	85.38	86.67
NB	84.60	84.33	83.26	84.25	84.33
SVM	90.40	88.83	90.56	90.50	90.89
GCN	99.00	99.75	96.68	96.53	98.03
GAT	98.01	98.50	99.02	99.01	99.99

In this evaluation, the graph-based models were superior to the other classifiers. Even with the variation in the percentage of data for network training, the GNNs maintained results higher in accuracy than the other models. Comparing the other classifiers, SVM had its best performance using 90% of the data for network training, with an accuracy of 90.89%, being considerably lower than the GNN models, which had at least 98.03% accuracy using this ratio.

For comparison purposes, 70% of the data was standardized for training and 30% for testing. In this ratio, the accuracy values of the other models were closer to the GNNs, that is, the most unfavorable case for the GNNs.

The GAT presents stability in the convergence process. After 100 epochs, the model maintains the same accuracy, with no significant improvement in the loss. For this reason, all comparative evaluations are done using 100 epochs. Next, an evaluation of the influence of connections of the nodes with other classes is performed. For this evaluation, equivalent probabilities of connection between nodes that do not belong to the same class were considered, maintaining the previously determined probability of connection between nodes of the same class.

The results of the influence of connections other than the node class are presented in Table 3 with respect to the classifiers being compared. This inclusion can be understood as noise since this information has no meaning for the graph and it impairs its classification.

The inclusion of connections that do not belong to the same class as the node being evaluated has a disadvantageous result for the GNNs. When the probability of connections with other classes is equal to or greater than 10%, the GNNs have inferior accuracy results compared to the SVM. In other words,

Table 3. Performance with a higher probability of connections to other classes.

Classifier	Accuracy by probability of connections with other classes				
	1%	3%	5%	10%	20%
LR	85.12	85.84	85.26	84.11	84.12
LDA	83.84	83.26	84.83	82.68	83.27
KNN	86.84	87.55	87.40	85.97	87.12
DTC	83.27	84.11	86.70	83.12	84.84
NB	84.84	82.97	85.83	83.54	83.41
SVM	89.84	89.84	90.55	89.98	90.56
GCN	99.67	96.01	93.69	79.07	84.72
GAT	97.67	97.34	94.68	88.04	86.05

when samples have connections with classes that are distinct from their own class, the GNNs show to be not promising. In a practical example, this would mean performing a search on a subject that has no relation to the interests of the user's profile and making this suggestion to another user who is connected to this one.

To evaluate the influence of fewer node connections in the same class, connections with classes other than their own class were not considered. This is an ideal case where the connections only occur with their class, so it is possible to specifically compare the influence of the variation in the number of these connections. This evaluation is done by increasing or reducing the probability of connections from their class without the influence of others. The results of this evaluation are presented in Table 4.

Table 4. Evaluation of different probabilities of connection to the same class.

Classifier	Accuracy by probability of connections to the same class								
	99%	97%	95%	90%	80%	70%	50%	30%	10%
LR	84.54	86.55	85.41	86.98	85.12	85.40	86.41	84.98	85.26
LDA	83.97	84.55	82.98	83.55	83.26	82.25	84.41	83.12	82.83
KNN	86.41	87.41	87.84	88.13	87.55	87.54	88.55	88.28	87.70
DTC	83.98	84.98	84.70	86.41	83.26	83.83	85.98	85.12	86.12
NB	83.40	84.26	83.98	84.69	82.69	84.83	83.13	82.84	83.26
SVM	90.13	90.55	90.42	90.28	90.27	89.55	89.56	89.56	90.41
GCN	97.67	98.67	98.67	99.34	98.34	96.35	98.01	93.69	83.72
GAT	98.01	96.01	97.67	98.67	97.67	97.34	94.35	91.69	87.71

The GNNs remained superior in classification accuracy until the time when the probability between connections of the same class is 10%, when the probability of connections is low, the graph loses its properties, becoming a classical classification problem in which the classifier evaluates only the features of the node and not its connections.

3.1 Variation in the Dataset Size

The variation in the number of features, the inclusion of more classes, and the reduction in the number of samples can make the classification problem even more difficult. To evaluate the influence of changing these parameters on the compared models, this section will present the results of these variations in an ideal graph case. In Table 5, the results of the variation in the number of classes are shown.

Table 5. Performance of classifiers by changing the number of classes.

Classifier	Accuracy by number of classes						
	2	3	4	5	6	7	8
LR	94.14	85.98	64.53	52.24	48.79	46.35	30.92
LDA	93.00	84.13	65.10	51.95	49.06	45.49	31.20
KNN	93.57	87.84	72.96	68.25	65.38	63.09	53.23
DTC	92.14	86.55	69.38	63.38	54.37	45.49	41.35
NB	93.14	84.27	60.24	58.39	51.21	48.07	31.05
SVM	94.00	90.41	74.68	71.11	67.38	65.10	56.65
GCN	99.99	97.01	95.67	95.33	96.71	90.33	84.87
GAT	99.67	99.00	92.67	94.67	93.09	93.67	82.57

The use of two classes from the synthetic generation of data makes the problem more easily resolvable, with superior results using all the compared classifiers. As more classes are included in the problem, it becomes more difficult to solve it, and the classifiers have a significantly lower accuracy.

Although the GNNs also show lower accuracy in relation to the increase in the number of classes, for this problem, using up to 7 classes, the results remain over 90%. This shows that these models are adequately classifying the nodes in relation to their connections. In Figure 2, the problem of classification using 2 features is presented for several numbers of classes. It's presented 2, 4, 5, 6, 7, and 8 classes as 3 classes were previously presented in Figure 1.

In many models, the use of a smaller dataset can result in lower accuracy due to difficulty in network training. Table 6 shows the results of the use of different numbers of samples for the classification problem using synthetic data.

Although there is variation among the results, all models remain with similar accuracy results when the number of samples is changed. This shows that if the distribution of features follows the same trend, the classifier will have similar results. This statement can be made considering there is a minimum amount of data to train the model adequately. The variations in the results are justified by the randomness in the generation of the database. In Table 7, the results of the variation in the number of features in the dataset are presented.

The variation in the number of features had an even smaller influence on the performance of the model than the other variations. It is noticeable that the big difference in performance results occurs when the connections between nodes are

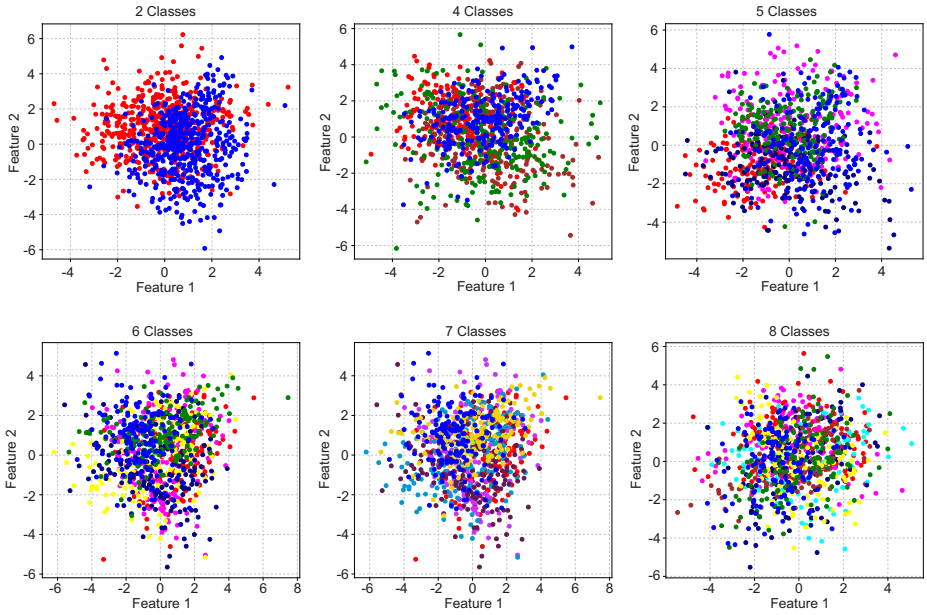


Fig. 2. Features generated by synthetic data using different number of classes.

Table 6. Performance of classifiers by changing the number of samples of the dataset.

Classifier	Accuracy by number of samples					
	100	300	500	1000	1500	2000
LR	80.95	80.33	81.66	87.27	80.93	79.77
LDA	79.52	79.40	81.67	84.69	80.36	80.06
KNN	69.52	76.95	82.80	86.97	83.41	80.35
DTC	74.05	70.81	75.06	85.26	84.84	78.34
NB	72.38	77.95	82.82	85.55	82.55	77.27
SVM	82.38	82.26	84.55	90.84	87.32	83.99
GCN	93.55	93.33	99.99	98.34	98.67	96.18
GAT	96.77	96.67	98.68	99.00	98.67	99.83

changed. The use of a smaller dataset, which is sufficient for training, does not represent a problem for these models for the classification performed.

3.2 Limitations

The computational effort required for computing graph-based models can exceed 10 times the time needed to train standard models. In this paper, this limitation does not represent an issue because the classifier can be used offline, in which case the use of graph models is very promising when the connections are appropriate to improve the model's performance.

Table 7. Performance of classifiers by changing the number of features of the samples.

Classifier	Accuracy by number of features				
	5	10	15	20	30
LR	86.69	85.98	83.84	85.12	83.97
LDA	84.55	84.26	81.26	83.97	83.26
KNN	92.13	85.40	81.83	82.41	78.25
DTC	87.12	84.97	81.40	82.97	82.41
NB	85.54	84.83	83.26	84.54	85.40
SVM	91.70	90.13	86.55	87.12	87.55
GCN	99.34	99.99	99.67	99.99	99.67
GAT	99.00	97.67	99.99	99.67	98.67

4 Conclusion

The results presented in this paper show that for graph-based models to be superior for classification, the connections between nodes must occur mostly among nodes of the same class. When there are connections between nodes of different classes, GNNs lose their advantages over other classifiers. When the nodes are connected to the same class, the performance of GNNs is better than other classification models.

The change in the size of the dataset to the number of samples and alteration in the number of features did not have a major influence on the results, mainly because these alterations maintain the features of the degree of difficulty in classification. The inclusion of more classes made it more difficult to classify the problem, since GNNs perform classification taking into account node connections as well as features, the GNNs' results were superior.

Future work can be conducted using hyperparameter tuning methods to assess the maximum improvement capacity of each model as a function of using a refined model. Based on multicriteria optimization [18], classification models can be tuned using algorithms such as genetic algorithm [19], Coyote optimization algorithm, [20], or tree-structured Parzen estimator applied in Optuna [21].

Acknowledgements

A realização desta investigação foi parcialmente financiada por fundos nacionais através da FCT - Fundação para a Ciência e Tecnologia, I.P. no âmbito dos projetos UIDB/04466/2020 e UIDP/04466/2020. Esta pesquisa foi parcialmente financiada pela Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina (FAPESC).

Bibliography

[1] Ma, J., Liu, B., Li, K., Li, C., Zhang, F., Luo, X., Qiao, Y.: A review of graph neural networks and pretrained language models for knowledge graph reasoning. Neurocomputing p. 128490 (2024). https://doi.org/10.1016/j.neucom.2024.128490

[2] Xu, J., Wang, F., Ai, J.: Defect prediction with semantics and context features of codes based on graph representation learning. IEEE Transactions on Reliability **70**(2), 613–625 (2021). https://doi.org/10.1109/TR.2020.3040191

[3] Starke, L., Hoppe, A.F., Sartori, A., Stefenon, S.F., Santana, J.F.D.P., Leithardt, V.R.Q.: Interference recommendation for the pump sizing process in progressive cavity pumps using graph neural networks. Scientific Reports **13**(1), 16884 (2023). https://doi.org/10.1038/s41598-023-43972-4

[4] Westarb, G., Stefenon, S.F., Hoppe, A.F., Sartori, A., Klaar, A.C.R., Leithardt, V.R.Q.: Complex graph neural networks for medication interaction verification. Journal of Intelligent & Fuzzy Systems **44**(6), 10383–10395 (2023). https://doi.org/10.3233/JIFS-223656

[5] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv:1609.02907v4 **4**, 1–14 (2017). https://doi.org/10.48550/arXiv.1609.02907

[6] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv:1710.10903v3 **3**, 1–12 (2018). https://doi.org/10.48550/arXiv.1710.10903

[7] Shipe, M.E., Deppen, S.A., Farjah, F., Grogan, E.L.: Developing prediction models for clinical use using logistic regression: an overview. Journal of Thoracic Disease **11**(4), S574–S584 (2019). https://doi.org/10.21037/jtd.2019.01.25

[8] Wen, J., Fang, X., Cui, J., Fei, L., Yan, K., Chen, Y., Xu, Y.: Robust sparse linear discriminant analysis. IEEE Transactions on Circuits and Systems for Video Technology **29**(2), 390–403 (2019). https://doi.org/10.1109/TCSVT.2018.2799214

[9] Tuleski, B.L., Yamaguchi, C.K., Stefenon, S.F., dos Santos Coelho, L., Mariani, V.C.: Audio-based engine fault diagnosis with wavelet, markov blanket, rocket, and optimized machine learning classifiers. Sensors (Basel, Switzerland) **24**(22), 7316 (2024). https://doi.org/10.3390/s24227316

[10] Wang, F., Wang, Q., Nie, F., Li, Z., Yu, W., Ren, F.: A linear multivariate binary decision tree classifier based on k-means splitting. Pattern Recognition **107**, 107521 (2020). https://doi.org/10.1016/j.patcog.2020.107521

[11] Ali, L., Khan, S.U., Golilarz, N.A., Yakubu, I., Qasim, I., Noor, A., Nour, R.: A feature-driven decision support system for heart failure prediction based on statistical model and gaussian naive bayes. Computational and

Mathematical Methods in Medicine **2019** (2019). https://doi.org/10.1155/2019/6314328

[12] Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., Lopez, A.: A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing **408**, 189–215 (2020). https://doi.org/10.1016/j.neucom.2019.10.118

[13] Vesselinova, N., Steinert, R., Perez-Ramirez, D.F., Boman, M.: Learning combinatorial optimization on graphs: A survey with applications to networking. IEEE Access **8**, 120388–120416 (2020). https://doi.org/10.1109/ACCESS.2020.3004964

[14] Ye, Y., Ji, S.: Sparse graph attention networks. IEEE Transactions on Knowledge and Data Engineering **35**(1), 905–916 (2021). https://doi.org/10.1109/TKDE.2021.3072345

[15] Wei, Y., Wu, D.: State of health and remaining useful life prediction of lithium-ion batteries with conditional graph convolutional network. Expert Systems with Applications **238**, 122041 (2024). https://doi.org/10.1016/j.eswa.2023.122041

[16] Bhatti, U.A., Tang, H., Wu, G., Marjan, S., Hussain, A.: Deep learning with graph convolutional networks: An overview and latest applications in computational intelligence. International Journal of Intelligent Systems **2023**(1), 8342104 (2023). https://doi.org/10.1155/2023/8342104

[17] Wolf, M., Tritscher, J., Landes, D., Hotho, A., Schlör, D.: Benchmarking of synthetic network data: Reviewing challenges and approaches. Computers & Security p. 103993 (2024). https://doi.org/10.1016/j.cose.2024.103993

[18] Ribeiro, M.H.D.M., da Silva, R.G., Moreno, S.R., Canton, C., Larcher, J.H.K., Stefenon, S.F., Mariani, V.C., dos Santos Coelho, L.: Variational mode decomposition and bagging extreme learning machine with multi-objective optimization for wind power forecasting. Applied Intelligence **54**, 3119–3134 (2024). https://doi.org/10.1007/s10489-024-05331-2

[19] Stefenon, S.F., Seman, L.O., Klaar, A.C.R., Ovejero, R.G., Leithardt, V.R.Q.: Hypertuned-YOLO for interpretable distribution power grid fault location based on EigenCAM. Ain Shams Engineering Journal **15**(6), 102722 (2024). https://doi.org/10.1016/j.asej.2024.102722

[20] Larcher, J.H.K., Stefenon, S.F., dos Santos Coelho, L., Mariani, V.C.: Enhanced multi-step streamflow series forecasting using hybrid signal decomposition and optimized reservoir computing models. Expert Systems with Applications **255**, 124856 (2024). https://doi.org/10.1016/j.eswa.2024.124856

[21] da Silva, E.C., Finardi, E.C., Stefenon, S.F.: Enhancing hydroelectric inflow prediction in the Brazilian power system: A comparative analysis of machine learning models and hyperparameter optimization for decision support. Electric Power Systems Research **230**, 110275 (2024). https://doi.org/10.1016/j.epsr.2024.110275