Edge Information Assisted Decoder for Business Process Anomaly Detection

Teoman Berkay Ayaz*, Alper Özcan†, Akhan Akbulut*‡
**R&D Center of Next4biz* Sahrayicedit Mah, Pakpen Plaza,
No:40/4, 34734 Istanbul, Turkey
teoman.ayaz@next4biz.com, akhan.akbulut@next4biz.com
†*Department of Computer Engineering Akdeniz University* Antalya, Turkey
alperozcan@akdeniz.edu.tr
‡*Department of Computer Engineering Istanbul Kültür University* 34536 Istanbul, Turkey
a.akbulut@iku.edu.tr

***Abstract*—Anomaly detection as a subject focuses on the identification of data point which significantly deviate from what is the norm or the standard of the dataset. This gives anomaly detection a wide range of applications where the detection of irregularities is often times of crucial importance such as Business Process Management (BPM). In this study we present a novel type of decoder referred to as "Edge Information Assisted Decoder" (EIAD), working on graph data to incorporate edge indexes and attributes into the decoding to achieve improved anomaly detection. We tested a total of 8 encoder-decoder combinations to comparatively evaluate them and prove the effectiveness of the proposed method. The proposed method and the best encoder-decoder combination, the graph attention network (GAT) encoder and the edge-conditioned convolution (ECC) decoder yielded an increase of 0.31 in F1-score from 0.32 to 0.63 when compared to the baseline multi-layer perceptron (MLP) decoder model, both with the optimal optimizer. The empirical results show that the proposed approach has a potential to improve graph based anomaly detection.**

***Index Terms*—Anomaly Detection, Graph Neural Networks, Auto Encoders, Edge Attributes, Business Process Management**

I. Introduction

Artificial intelligence (AI) is increasingly being integrated into various domains, enhancing our quality of life by automating tasks [1], improving decision-making [2], [3], and optimizing systems [4]. One of the critical areas where AI plays a significant role is anomaly detection also known as outlier detection [5] focuses on identifying data points, which drastically deviate from the norm within a dataset [6]. Since deviations can occur in any given domain, anomaly detection is regarded as an important domain of research with a wide range of applications [7], spanning anywhere from healthcare to security, from fraud detection to Business Process Management (BPM). BPM solutions in the digitalized modern markets are used to maintain and govern a businesses operations. Given the nature of these operations, BPM solutions generate mountains of data in the form of business process event logs. The events and incidents which occur in the real world where the operations are carried out, are reflected within the process event logs. The reflections of said incidents makes it so, that event logs are prone to containing anomalous occurrences taking place in the real world. This makes it possible to do anomaly detection on business process data which can be used in a variety of process management related applications [8].

From the perspective of a business, BPM anomaly detection can be a rather attractive usage of artificial learning based approaches due to several reasons: BPM anomaly detection as previously mentioned can help detect these anomalies, helping businesses increase their operational efficiency, which in turn can help them increase their profitability. Anomaly detection can also be leveraged when doing risk assessments, to mitigate any risks regarding the operational noncompliance or financial risks. Anomaly detection for the goal of proactive prevention can also be useful to avoid unnecessary costs when acting out the operations. Anomaly detection can also help to make sure that businesses act in full compliance, as significant deviations from the norm or the standard management of operations has the potential to indicate a breach of legality. In its essence, BPM anomaly detection can be of significant assistance to any given company trying to maintain and increase their operational efficiency, making it an effective and highly useful tool in the arsenal of any Business Intelligence (BI) specialist.

Despite how lucrative of a subject it is from a business perspective, the research surrounding BPM anomaly detection had its fair share of challenges. The most notable of these challenges is the scarcity of publicly available process event log datasets [9]. The scarcity of publicly available datasets makes it difficult for researchers to test their approaches across different sectors with alternating dependencies and process flows, forcing researchers into leveraging synthetically generated process data. Another commonly faced challenge is the lack of labeled datasets [10]. Given that the human cost of assigning BI specialists to navigate through the process logs to detect genuine instances of anomalies is so high and often times not possible. Due to the expense of manual labeling of genuine anomalies, researchers either end up having to resort to the usage of injected anomalies, where the the evaluation assumes that the pre-injection dataset does not contain any anomalies and the only anomalies existing within the dataset are the injected ones, hence yielding lower F1-scores than what should be due to the increasing number of false positives, or the usage of data analysis to detect deviations to evaluate the

979-8-3315-3149-2/24/$31.00

model based on.

In this study, we deployed a series of Auto-Encoder (AE) models with varying types of decoders to present a novel type of Graph Neural Network (GNN) based AE previously used in our related works, with the novelty being the decoder which will be referred to as "Edge-Information Assisted Decoder" (EIAD). It leverages the information found on edges to enhance anomaly detection performance. The study was done on the dataset the predecessor studies were done on to yield a more rigorous comparison between the available architectural performances achieved. For evaluation, we used the injections which are commonly used by other researchers to do model performance testing [11]. Throughout our study, we've:

- Conducted exhaustive testing to properly evaluate different architectures.
- Successfully deployed and demonstrated a novel type of decoder architecture for enhanced anomaly detection.

The remainder of this proceeding is split into 4 main sections: the Related Works section will go over the predecessor studies we've done to this alongside providing the reader with brief insights of relevant studies conducted by other researchers, the Methodology section will elaborate on the approach deployed within this study, the Experimental Results section will go over the results of the leveraged methodology, lastly in the conclusion section will go over the findings of our study.

II. Related Works

Anomaly detection as a subject has been researched extensively for its applications in various domains involving data. Within the business process anomaly detection domain, anomaly detection is usually classified into three types of anomalies: trace level, event level and attribute level [8]. Trace level anomalies represent business process instances or records which happen to have at least one anoamlous event or attribute. An anomalous event refers to an event that at least has one anomalous attribute and an attribute level anomaly represents an anomalous attribute. For detection on different levels, many researchers before us leveraged AE based approaches such as Nolle et al. [12] in 2016. The researchers leveraged denoising AEs to build a model that can learn what business process event logs are supposed to be like despite the existing noise found within natural data. In 2021, Huo et al. [13] leveraged a graph data structure, representing business process event logs alongside a Graph Auto-Encoder (GAE) model to do business process anomaly detection. Using this method, researchers got a reasonably high trace level F1-score proving the effectiveness of their proposed approach. In a different study by Guan et al. [14], the researchers developed a weakly supervised AE referred to as "WAKE". By leveraging a pre-trained AE, WAKE generates feature representations from the input vectors, creating 3 components: latent feature representation, the reconstruction error and maximum reconstruction error vector. The process then continues by feeding the features into a multi-layer perceptron (MLP) which works as an anomaly score generator. By leveraging the proposed method, authors achieved to get higher F1-score than state of the art models at the time on both real world, and synthetic data.

Processing business process data in the form of graph data structure is also something that is used to do anomaly detection. Graph data structure can help capture the complex nature of highly interconnected business processes, helping to do better anomaly detection [15]. In our previous researches regarding the model which now we are doing a comparison of, we worked on a way to automate supervision for Auto-Encoder training. Based on Z-scores generated form the reconstruction error we selected and left out models with a score higher than a threshold, achieving 0.07 F1-score improvement at the most significant threshold. Following that study, another reasearch was started about the incorporation of natural language based embeddings into the graph structure, in the form of additional node and edge features [16]. Leveraging the same model, we've achieved an increase in F1-score of 0.097 which indicates that the incorporation of natural language embeddings can significantly increase anomaly detection performance when working with AEs.

III. Methodology

In this research, we experimented on different GAE architectures with the main differences between them being the decoders. Leveraging graph convolutions and additional information found on edges for the decoding task, the EIAD model deviates away from the standard definition of an AE. The study was conducted on the same dataset, which its predecessor was done on [15], with injection labels at 0.10 contamination, albeit with some differences in the graph construction.

A. Graph Construction

Constructed graphs consist of two main components: transitions which are represented by the edges and states which are represented by the nodes. By creating a mapping of the states found within the case, initial step is to create unique nodes, representing unique states within the process trace. The same mapping is latter used in the generation of pre-aggregated edges where edges can be duplicates of one another. Once their generation is done, the edge features are summed for each unique type of edge, finalizing the structure. The resulting graphs with respect to the specific dataset, are graphs with $n \times 12$ node features and $m \times 32$ edge features.

B. Edge Information Assisted Decoder (EIAD)

Multiple tests were conducted using various convolutional operators which were previously used in our experiments, testing their yielded performances. Yet the main novelty proposed is what will be referred to as EIAD, which as shown in Figure 1, composes of graph convolutional layers and leverages additional features as input from the edges. The additional information comes in the form of edge indexes and edge attributes from the original input graph, passed to the decoder alongside the latent features. This causes the EIAD model to deviate away from the standard definition of an AE where: a typical AE composes of an encoder, latent feature

representations and a decoder, where the encoder learns to generate meaningful latent feature representations of the input features, later feeding them into the decoder which learns how to reconstruct said feature representations [17]. In contrast, our proposed approach feeds more than the latent feature representation into the decoder. This effectively turns the task of the model into prediction problem, where the input feature set X composes of latent feature representations of node features and edge information, and original node feature set is the target variable y.

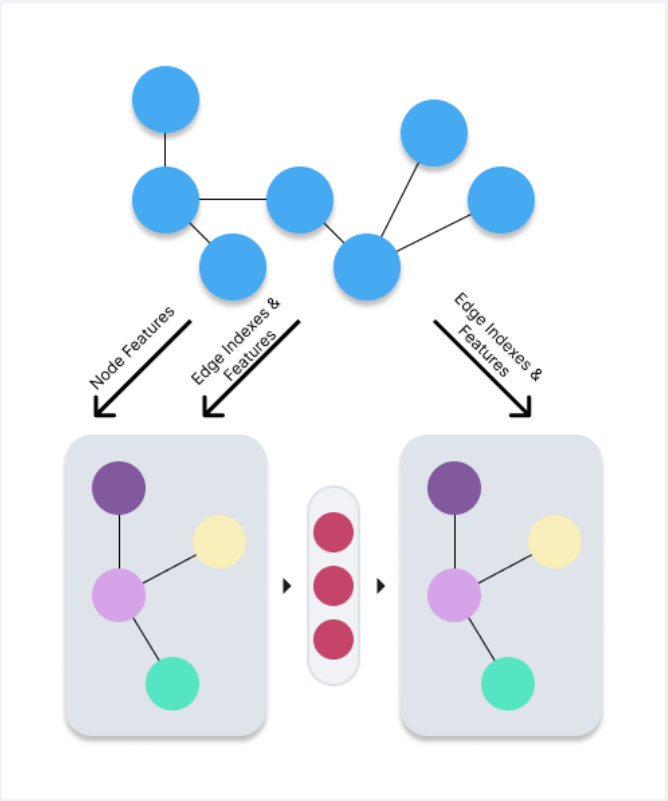

Fig. 1. GAE Model with Edge Information Assisted Decoder

C. Evaluated Encoders and Decoders

As mentioned, throughout the experiment a pool of various encoder and decoder architectures were tested to evaluate their performance. With regards to the encoders deployed we leveraged two different types of neural networks that can compute on edge attributes. Due its success within the previous two studies which it was leveraged in [15], [16], the initial encoder leverages the Edge-Conditioned Convolutional (ECC) layers (A.K.A. Neural Network Convolutions, NNConv), where the layers use a neural network to compute the edge attributes as one of its parameters. The other encoder architecture leveraged uses another edge computation capable type of network referred to as "Graph Attention Network" (GAT), which leverages masked self-attentional layers [18]. On the side of the decoders, an MLP architecture was used as a baseline decoder architecture, as MLPs are the most commonly used neural network architecture and are proven to be effective solutions across various domains where deep learning based solutions are applicable [19]. The remaining decoders are built with the same operators as the ECC and GAT encoders with the exception of Graph Convolutional Network (GCN) Encoder, which leverages spectral graph convolutions to convolute over graph structured data [20]. The main reason behind the utilization of the GCN Encoder is that the model does not compute on edge features, but solely on edge indexes and edge weights when available. This helps create a baseline for when working solely on node features and edge indexes.

D. Model Training and Evaluation

A total of 8 combinations between 2 different types of encoders and 4 distinct types of decoders were trained using 3 different optimizers: Adam, Adamax and Adagrad. Each model was trained for a 100 epochs and checkpointing was applied to get each model in its best performing state with regards to the training loss. The evaluation was done using the Mean Squared Error (MSE) of each reconstructed node. Similar to what was done in the predecessor study [15], 3 significant thresholds of 99th, 95th and 90th percentile points of reconstruction errors were chosen for F1-score calculation. Any node with a reconstruction error above any of these thresholds were labeled as anomalous for the respective threshold, resulting in three sets of predictions. The significance of the thresholding approach for F1-score calculation, lies within the real-world applicability of the proposed solution. A system which employs reconstruction based anomaly detection is likely to return an anomaly score or a probability of sorts, in contrast to a prediction. A potential user would likely expect an instance with a higher score to have a higher likelihood of being anomalous. Given this, each percentile point represents an anomaly score and above, such as the 90th percentile representing an anomaly score of 90 and above. The usage of F1-score for the evaluation is also significant because it is a measure of class balance as well as a measure on how well the positive class is predicted, the positive class in our case being the anomalies. Hence, the thresholded F1-score calculation is a way to calculate the effectiveness of the proposed approach at different scores in a real world deployment scenario. Lastly, as the graph construction does not allow for event or attribute level detection, each trace that contained at least one or more anomalous nodes were predicted as anomalous and a set of three F1-scores for each model was acquired.

IV. Experimental Results

When we take a look at Table I, it can be seen that GAT based encoder model yields much superior performance with ECC Encoder only matching it at the minimum for 95th percentile. Across all other points of evaluation, GAT Encoder outperforms its ECC equivalent. Specifically, at the 99th minimum we can see that the GAT Encoder more than doubles the performance of its counterpart, displaying its superiority in higher thresholds where one would expect to find more true positives. At the maximums, we can see that the

ECC Encoder models at a maximum can get a 0.47 F1-score at the 95th percentile threshold, whereas at the same threshold GAT Encoder yields a 0.63 maximum F1-score, an almost 40% increase displaying superiority over its counter part. The results show that the trend of GAT Encoder outperforming ECC Encoder is consistent at every aggregation and at all threshold points, with the exception of 95th minimum. This indicates that the GAT Encoder significantly outperforms its counterpart when capturing essential features regarding the nodes. Additionally, the consistency accross aggregations suggest that the model is rather stable in comparison to its counterpart. The results show that the GAT based encoder is far more suitable in comparison to its counterpart, especially where more precisionn is necessary as demonstrated by its performance at the 99th percentile making GAT Encoder far more reliable. The most notable indicator to this might be the mentioned more than double performance at the 99th minimum.

TABLE I
ENCODER AGGREGATED TRACE LEVEL F1-SCORES

Encoder	Aggregation	99th	95th	90th
ECCEncoder	Max	0.33	0.47	0.44
	Mean	0.22	0.30	0.27
	Min	0.09	0.19	0.17
GATEncoder	Max	0.35	0.63	0.52
	Mean	0.28	0.37	0.28
	Min	0.19	0.19	0.18

When we inspect Table II, we can see that all the decoders perform about the same for the 99th max with the exception of ECC Decoder that performs 0.02 higher. We start to see the real differences at the mean and minimum points, where the ECC Decoder starts to show its superiority over its equivalents. At mean score points the only decoder that can match it is the GAT Decoder both getting an F1-score of 0.29 whereas the baseline MLP Decoder only got 0.20. The difference is much more represented at the minimum point where the ECC Decoder got 0.25 in contrast to the 0.16 by GAT and GCN Decoders and the 0.09 by the MLP. When we move towards the 95th percentile threshold the difference in performance is further displayed by the ECC Decoder yielded a maximum of 0.63 in contrast to the 0.35 yielded by GCN Decoder, 0.33 yielded by the GAT and the 0.32 yielded by the MLP. The mean points at the 95th also show that the ECC Decoder is the superior choice, yielding almost double the performance of the GCN, GAT and the MLP.

One other thing that should be considered is the importance of the optimizer choice. The results show us that the optimizer which performs the best is the Adamax with an F1-score 0.63 at the 95th percentile. Despite Adamax yielding the top performance, we can see that other points have different stories such as Adam toppling Adamax with an F1-score of 0.35 at the 99th percentile or matching it with 0.51. Adagrad on the other hand yielded disappointing results yielding lower F1-scores across the board. With respect to the previously noted observations, when regarding the worst possible case,

TABLE II
DECODER AGGREGATED TRACE LEVEL F1-SCORES

Encoder	Aggregation	99th	95th	90th
MLPDecoder	Max	0.31	0.32	0.25
	Mean	0.20	0.23	0.20
	Min	0.09	0.19	0.17
GCNDecoder	Max	0.33	0.35	0.30
	Mean	0.23	0.30	0.25
	Min	0.16	0.24	0.21
GATDecoder	Max	0.33	0.33	0.26
	Mean	0.29	0.31	0.23
	Min	0.16	0.24	0.18
ECCDecoder	Max	0.35	0.63	0.52
	Mean	0.29	0.51	0.43
	Min	0.25	0.38	0.31

Adamax seems to be the safest option, since in every threshold it yielded the highest minimum F1-score.

TABLE III
OPTIMIZER AGGREGATED TRACE LEVEL F1-SCORES

Encoder	Aggregation	99th	95th	90th
Adagrad	Max	0.31	0.57	0.41
	Mean	0.24	0.34	0.25
	Min	0.08	0.21	0.18
Adam	Max	0.35	0.61	0.51
	Mean	0.24	0.32	0.28
	Min	0.09	0.18	0.16
Adamax	Max	0.33	0.63	0.51
	Mean	0.26	0.34	0.28
	Min	0.15	0.21	0.19

Lastly, regarding the 90th percentile F1-scores, in Figure 2 we can see that there is a highly noticable increase in F1-score, most notably for the ECC decoder, which the mean F1-score almost doubles. After the 95th percentile however, there is a significant drop in F1-scores when we move towards the 90th percentile we can notice a drop in F1-scores across all of our decoder architectures. This can be attributed to the increasing number of false positives lowering our F1-score.

V. CONCLUSION

In this research, we proposed the Edge Information Assisted Decoder, as a novel way of enhancing graph based anomaly detection using AEs. By utilizing the information found on the edges we successfully managed to apply the proposed methodology in the BPM anomaly detection domain, and proved its effectiveness. The EIAD demonstrated superior performance in comparison to the baseline MLP method, especially the decoder with the ECC operators. The superior performance of the ECC Encoder can be attributed to its superior edge feature computation capabilities, whereas the fact that it performed best when paired with the GAT Encoder shows to us that when it comes to the generation of the node feature representations the GAT Encoder works far better when considering the maximum F1-scores of 0.63 and 0.47. Lastly, the optimizer choice did yield a non-negligible difference where the Adamax performed 0.02 better than Adam and 0.06 better than Adagrad in F1-score at the 95th point. This research contributes to the graph based anomaly detection approaches

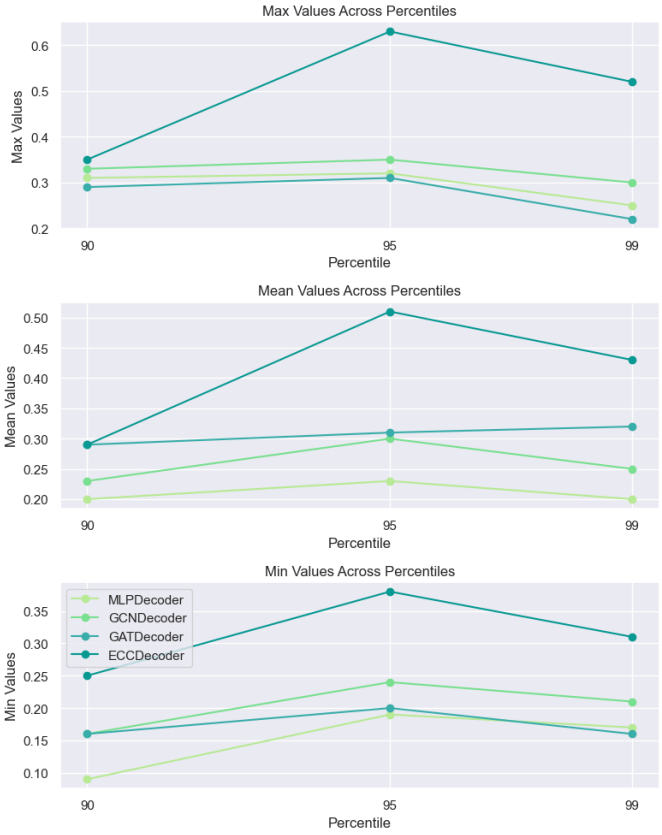

Fig. 2. Decoder Percentile F1-scores Plotted

and the ongoing efforts to refine it. Future work may focus on leveraging a more complex architecture or verifying the approach on multiple datasets.

Acknowledgements

This study is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant no TEYDEB-3231136.

References

[1] C. Catal and A. Akbulut, "Automatic energy expenditure measurement for health science," *Computer methods and programs in biomedicine*, vol. 157, pp. 31–37, 2018.

[2] J. Hanna, F. Patlar, A. Akbulut, E. Mendi, and C. Bayrak, "Hmm based classification of sports videos using color feature," in *2012 6th IEEE International Conference Intelligent Systems*. IEEE, 2012, pp. 388–390.

[3] G. Dogan and F. P. Akbulut, "Multi-modal fusion learning through biosignal, audio, and visual content for detection of mental stress," *Neural Computing and Applications*, vol. 35, no. 34, pp. 24 435–24 454, 2023.

[4] M. Cakir and A. Akbulut, "A bayesian deep neural network approach to seven-point thermal sensation perception," *IEEE Access*, vol. 10, pp. 5193–5206, 2022.

[5] C. Iscan, O. Kumas, F. P. Akbulut, and A. Akbulut, "Wallet-based transaction fraud prevention through lightgbm with the focus on minimizing false alarms," *IEEE Access*, 2023.

[6] D. Samariya and A. Thakkar, "A comprehensive survey of anomaly detection algorithms," *Annals of Data Science*, vol. 10, no. 3, pp. 829–850, 2023.

[7] T. H. A. Musa and A. Bouras, "Anomaly detection: A survey," in *Proceedings of Sixth International Congress on Information and Communication Technology*, X.-S. Yang, S. Sherratt, N. Dey, and A. Joshi, Eds. Singapore: Springer Singapore, 2022, pp. 391–401.

[8] J. Ko and M. Comuzzi, "A systematic review of anomaly detection for business process event logs," *Business & Information Systems Engineering*, vol. 65, no. 4, pp. 441–462, 2023.

[9] J. N. Adams, J. Peeperkorn, T. Brockhoff, I. Terrier, H. Göhner, M. S. Uysal, J. De Weerdt, W. M. van der Aalst *et al.*, "Discovering high-quality process models despite data scarcity," *arXiv preprint arXiv:2310.11332*, 2023.

[10] R. Sarno, F. Sinaga, and K. R. Sungkono, "Anomaly detection in business processes using process mining and fuzzy association rule learning," *Journal of Big Data*, vol. 7, no. 1, p. 5, 2020.

[11] W. Guan, "Survey and benchmark of anomaly detection in business processes," https://github.com/guanwei49/BPAD, 2024, commit ID: fcf61a2.

[12] T. Nolle, A. Seeliger, and M. Mühlhäuser, "Unsupervised anomaly detection in noisy business process event logs using denoising autoencoders," in *Discovery Science*, T. Calders, M. Ceci, and D. Malerba, Eds. Cham: Springer International Publishing, 2016, pp. 442–456.

[13] S. Huo, H. Völzer, P. Reddy, P. Agarwal, V. Isahagian, and V. Muthusamy, "Graph autoencoders for business process anomaly detection," in *Business Process Management: 19th International Conference, BPM 2021, Rome, Italy, September 06–10, 2021, Proceedings 19*. Springer, 2021, pp. 417–433.

[14] W. Guan, J. Cao, H. Zhao, Y. Gu, and S. Qian, "Wake: A weakly supervised business process anomaly detection framework via a pre-trained autoencoder," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[15] T. B. Ayaz, E. Gülce, A. Özcan, and A. Akbulut, "Semi-supervised detection of contaminated business process instances using graph autoencoders and dynamic edge convolutions for bpm anomaly detection," in *Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2024.

[16] T. B. Ayaz, E. Gülce, S. Hsu, A. Özcan, and A. Akbulut, "Business process management anomaly detection through semantic embedding-integrated graph neural networks," in *9th International Conference on Computer Science and Engineering (UBMK 2024)*. IEEE, 2024.

[17] U. Michelucci, "An introduction to autoencoders," *arXiv preprint arXiv:2201.03898*, 2022.

[18] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[19] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Transactions on Circuits and Systems*, vol. 8, no. 7, pp. 579–588, 2009.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.