Imbalance Node Classification with Graph Neural Networks (GNN): A Study on a Twitter Dataset

Alda Kika, Arbër Ceni, Denada Çollaku, Emiranda Loka, Ledia Bozo, Klesti Hoxha
University of Tirana, Faculty of Natural Sciences, Department of Informatics
Bulevardi Zogu I, Tiranë 1001, Albania

***Abstract*—Social networks produce a large volume of information, a part of which is fake. Social media platforms do a good job in moderating content and banning fake news spreaders, but a proactive solution is more desirable especially during global threats like COVID-19 pandemic and war. A proactive solution would be to ban users who spread fake news before they become important spreaders. In this paper we propose to model user's interactions in a social media platform as a graph and then evaluate state of the art (SOTA) graph neural networks (GNN) that can classify users' (nodes) profiles as being suspended or not. As with other real world data, we are faced with the imbalanced data problem and we evaluate different algorithms that try to fix this issue. Data used for this study were collected from X (Twitter) by using Twitter API 1.1 from November 2021 to July 2022 with the focus to collect information spread through tweets about vaccines. The aim of this paper is to evaluate if current models can deal with real world imbalanced data.**

***Keywords*—GNN; imbalanced data; Twitter; social networks; GCN; GraphSage; GAT; GraphSMOTE; ReNode**

I. Introduction

Social media has changed the way that the news is created and spread worldwide. According to Statista [1] in 2022 over 4.59 billion people were using social media, a number which is estimated to increase to almost six billion in 2027. In March 2020 World Health Organization declared Coronavirus disease (COVID-19) pandemic. Very soon a lot of people started to use social media to spread information about this new disease resulting in a massive infodemic. Infodemic[1] refers to false information related to a disease. This phenomenon is amplified through social networks spreading farther and faster like a virus [2]. World Health Organization (WHO) Director-General, Tedros Adhanom Ghebreyesus, in [3] emphasized the importance of fighting not only the pandemic but also the infodemic that was spreading. During difficult time for all the countries worldwide it is very important to spread the correct information rather than false and fake news that undermines the global response and jeopardizes measures to control the pandemic [4].

There are various definitions of fake news. Allcott and Gentzkow [5] defined fake news as "news articles that are intentionally and verifiably false and could mislead readers". Other studies have defined it as "a news article or message published and propagated through media, carrying false information regardless of the means and motives behind it" [6], [7], [8]. Much fake news about COVID-19 on social media has been circulating but the spread of fake news and disinformation about vaccines had a significant influence on vaccine acceptance, with many people opting not to get vaccinated posing a threat to individual and collective health. In order to prevent the spreading of fake news it is important to develop tools that will predict this.

Tweets associated with COVID-19 have been collected for a period of nine months from November 2021 until July 2022 for the purpose of applying social network analysis techniques on these posts. In [9], the authors focused their analysis on three main terms: pfizer, moderna and AstraZeneca. For these terms and for the nine month period, 27 graphs were built and an analysis of these graphs was performed. It was pointed out that most of the time the most influential users (based on beetweenness centrality) in the network were engaged in a form of information disorder and their accounts were suspended at the time of the study. Unfortunately the real reason behind the account suspension is not declared by X (Twitter), but one can safely presume that the main reason an account can be suspended is if a user does not comply with the TOS and engaging in mis/dis information is not in compliance with the platforms' TOS [2], [3]. We value that it is of great interest to be able to predict a user's status based on his position in the network as well as his attributes (date of joining the platform, number of posts, number of followers etc.). To this purpose, we applied different graph neural networks (GCN [10], GraphSage [11] and GAT [12]) on our dataset for the downstream task of classifying nodes. Given the fact that the dataset is imbalanced (there are more active users than inactive users), models which claim to fix this problem [13], [14], [15], [16], [17] were also studied.

The paper is organized as follows: Section II reviews recent studies on deep learning models on node classification in graph neural networks about fake news and infodemic prediction. The data that is used in this study, as well as the methodology that is applied for node classification and evaluation, is presented in Section III. In Section IV, the results are discussed. The conclusion and a concise summary conclude this paper.

II. Related Work

Graph neural networks are the most reliable and effective way to automate the process of fake news detection [18]. Feng et al. in [19] propose a bot detection framework that encode multi-modal user information such as user account description, the content of the post and numerical and categorical features that they can assign to an account available from the

[1] https://www.who.int/health-topics/infodemic

[2] https://help.twitter.com/en/resources/addressing-misleading-info

[3] https://help.twitter.com/en/rules-and-policies/crisis-misinformation

Twitter API. Users are treated as nodes of the heterogeneous graph. Relational Graph Convolutional Network is applied to the graph to tackle the challenges of bot disguise and bot communities.

In another work [20] for bot detection the authors propose a transductive model that combines symmetrically BERT and GCN. They constructed a heterogenous graph composed of unique words and documents in a collection as nodes. Word occurrences in a document or word co-occurrence are weighted using TD-IFD and PMI and stored as information on the edges of the graph. The experiment shows that a better performance can also be achieved by the proposed framework on a wide range of social robot detection datasets.

Li and Goldwasser [21] constructed a social information graph by embedding the items that are shared by Twitter users, the users who can be political users and other users who spread content. The edges represent the follower relations between political users and other users that follow them. They used two-layer Graph Convolutional Networks to capture the documents' social context and to detect news items as fake or not by node classification.

In [22] the dataset of CheckThat-2022 Task 3 dataset [23] is used as the primary dataset to analyze a corpus with unknown topics through multiclass classification, encompassing true, false, partially false, and other categories. They have explored three BERT-based models—SBERT, RoBERTa, and mBERT. The problem of imbalanced dataset is tackled by enhancing results via ChatGPT-generated artificial data for class balance and improving the results for the true, partially false, and other classes witnessed improvements of 5%, 9%, and 3%, respectively, while the results for false news experienced a decline of 9%.

Data augmentation has been applied by adopting synthetic minority oversampling in Zhao et al. [16]. They proposed a novel framework, GraphSMOTE, based on SMOTE [24] approach which addresses the imbalance problem by generating new samples, performing interpolation between samples in minority classes and their nearest neighbors. In [17] the authors have followed the approach of the imbalance of topological structure on the graph for handling the imbalance problem in graph-structured data. They proposed a ReNode framework for solving the problem of edge graph structure by re-weighting the influence of labeled nodes adaptively based on their relative positions to class boundaries.

III. Methodology and Data

In the following section we describe the dataset creation process as well as a description of different GNN models.

A. The Dataset

The data used in this paper is described in [9]. The data consist of 27 graphs. The size of each graph is displayed in Table I. To be able to run any graph neural network in any of these graphs, first we need to convert these graphs into a dataset that can be fed to the GNN model. To this purpose, we decided to use PyTorch[4] and PyG[5] libraries.

TABLE I. Size of the Created Graphs

	Pfizer		**Moderna**		**AstraZeneca**	
Month	**Nodes**	**Edges**	**Nodes**	**Edges**	**Nodes**	**Edges**
November 2021	367294	731875	358701	602031	216976	390200
December 2021	278266	543490	267581	430523	167942	288595
January 2022	198258	451691	247518	413592	110906	192169
February 2022	229197	673992	298231	600610	117994	213708
March 2022	244193	717114	323143	634862	93629	176314
April 2022	203168	608647	275389	569098	66175	201440
May 2022	210783	626985	273315	629752	79209	222855
June 2022	172249	526720	238628	494884	56726	130264
July 2022	40060	75511	56759	79718	23721	39747

We used the procedure[6] described in the PyG documentation to create our dataset. The advantage of converting the data into a dataset is that the data can then be fed into different neural networks without having to adapt the data each time. Another advantage is that we can make use of the many functions available in PyG to correctly split the dataset into development, test and validation sets. Creating a dataset from the available graphs, involves choosing node and edge features from the possible attributes. All node and edge features need to be of numeric type which limits our choice. In Tables III and IV node and edge features are shown. A transformation was required for some of the features as not all of them are numeric. Boolean features were transformed to 0 or 1. Date/time features were transformed to UNIX timestamp and category features such as "Relationship" were transformed to a number from 1 to 5 (Tweet = 1, Retweet = 2, Mentions = 3, Replies to = 4 and MentionsInRetweet = 5). The active status of a user was obtained by automatically making an HTTP request to the X (Twitter) web site since the API is no longer available. The active status for every user (node) in all the 27 constructed graphs was obtained in August 2023. As we expect in every real world dataset, we can notice by looking at Table II that the data is imbalanced for this dataset (i.e. there are fewer inactive users than active users). This will affect the prediction results of the standard models making them biased towards the majority class. Special models which account for the imbalance of the data need to be tested in this case.

B. GNN Models

Graph Neural Networks (GNNs) are a type of deep learning model designed for handling data represented in graph structures. In recent years, the field of graph neural network research has witnessed significant advancements [25], with a notable expansion in the range of GNN designs [25], [26]. Among these various designs in this paper we use Graph Convolutional Network (GCN) [10], GraphSAGE [11], and Graph Attention Networks (GAT) [12]. GCNs [10] are adapted to handle irregular and non-Euclidean data. For each node in the graph, GCN aggregates the features of all the neighbors of the node and the node itself. Different functions can be used for feature aggregation. The aggregated values are passed to the neural network which returns the feature vector resulting from the model. The GNC model can use several GCN layers on top of each other where the output of one layer will

[4] https://pytorch.org/

[5] https://pyg.org/

[6] https://pytorch-geometric.readthedocs.io/en/latest/tutorial/create_dataset.html

TABLE II. DATASET STATISTICS

	Pfizer			**Moderna**			**AstraZeneca**		
Month	**Inactive**	**Active**	**IR**	**Inactive**	**Active**	**IR**	**Inactive**	**Active**	**IR**
November 2021	20216	347078	0.058	16968	341733	0.049	10015	206961	0.048
December 2021	15515	262751	0.059	13278	254303	0.052	7843	160099	0.048
January 2022	11623	186635	0.062	12174	235344	0.051	5415	105491	0.051
February 2022	13402	215795	0.062	17089	281142	0.060	5611	112383	0.049
March 2022	14087	230106	0.061	17384	305759	0.056	4313	89316	0.048
April 2022	11678	191490	0.060	14217	261172	0.054	3638	62537	0.058
May 2022	12492	198291	0.062	14813	258502	0.057	4117	75092	0.054
June 2022	10128	162121	0.062	12207	226421	0.053	2853	53873	0.052
July 2022	2181	37879	0.057	2056	54703	0.037	1139	22582	0.050

TABLE III. NODE FEATURES

Feature Name
ID
Active
Degree
In-Degree
Out-Degree
Betweenness Centrality
Closeness Centrality
Eigenvector Centrality
PageRank
Clustering Coefficient
Reciprocated Vertex Pair Ratio
User ID
Followed
Followers
Tweets
Favorites
Joined Twitter Date (UTC)
Listed Count
Verified
Tweeted Search Term?
Vertex Group

TABLE IV. EDGE FEATURES

Feature Name
Relationship
Relationship Date (UTC)
Imported ID
In-Reply-To Tweet ID
Favorited
Favorite Count
In-Reply-To User ID
Is Quote Status
Retweet ID
Unified Twitter ID
Vertex 1 Group
Vertex 2 Group

be the input for the next layer. GraphSage [11] generates node embeddings by sampling and aggregating information from their neighbors. This model is suitable for large graphs because it is not necessary to process the entire graph at once, thus avoiding the limitations derived from the processing power and memory capacity available to us. The input of the GraphSage model is the graph which consists of nodes and edges. Each node in the graph has some features. To find node embeddings, the model selects for each node a fixed-size subset of its neighbors. The features of the selected neighbors are aggregated by means of an activation strategy thus creating an aggregated representation for each node. This aggregation reveals information about the local neighborhood of the node. These aggregated representations are passed as input to an activation function. To discover more complex relationships, updated representations can be passed through several layers of the GraphSage model. The last layer of the network is determined by the task to be performed. This model can be used to classify a node, predict a link. GAT [12] discovers dependencies and relationships in graph-structured data. Unlike GCN where all neighbors of a node have the same importance, GAT uses an “attention” mechanism to weight the importance of each neighbor during the aggregation process. In this way the network works only with the information of the nodes that are important. Weights are learned during the training process based on the importance each neighbor has on the specific task. Despite the progress in learning from graphs with GNN, existing work mainly focuses on balanced datasets [13]. In real world applications, we are faced with imbalanced data and GNNs fail to accurately predict the samples that belong to the minority class. Imbalanced class problems can be solved by modifying the GNN model to bias toward minority class, or by resampling that consists in altering the dataset by adjusting the number of instances for each class to achieve a balanced distribution [16], [14]. The most used approach is resampling because it can be integrated with any classifier [16], [14]. Resampling can be achieved either by undersampling or oversampling. Undersampling methods remove instances that belong to the majority class, but this can result in a loss of valuable information [16], [14]. Oversampling methods increase the number of minority classes which can lead to overfitting [16]. The results of applying these approaches to graphs are suboptimal because they consider each sample as independent and don’t take into account the relation that exists in the graph data [16]. In this paper we first apply the three standard SOTA models (GCN, GraphSage and GAT) to classify nodes as being active or inactive. Given the fact that our data is imbalanced we also use GraphSMOTE [16] and ReNode [17] to deal with imbalance data. GraphSMOTE uses a feature extractor to learn node representations. Node representation should reflect the similarities and dissimilarities between samples considering node attributes, node labels, and

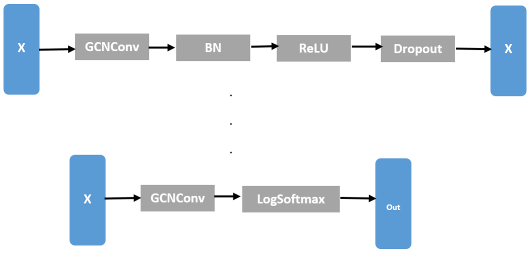

Fig. 1. The GCN model.

local graph structures [16]. After obtaining node representation for each node, GraphSMOTE uses the SMOTE algorithm to generate synthetic minority nodes in the latent space [16]. In order to include the new nodes in the graph, GraphSMOTE simulates connections between synthetic nodes by performing a training on existing nodes and edges. The GNN classifier can then perform node classification based on the augmented graph created by GraphSMOTE. Another framework that we have used to handle the topology imbalance issue of our network is ReNode. ReNode framework re-weight the influence of labeled nodes according to their position. Their structural position are located by using the conflict detection-based Topology Relative Location (Totoro) metric to leverage the interaction among them. The training weights of nodes with small conflict are increased and vice versa [17].

IV. Experimental Results

In this section, we present the results of our experiments in our imbalanced dataset. It will try to answer the following research questions:

- RQ1: Do standard GNN models yield acceptable results in case of imbalanced data?
- RQ2: Do special GNN models designed for imbalanced data help in our case?

A. Experimental Setup

The models for GCN, GraphSage and GAT are implemented in Pytorch and PyG by following a message passing paradigm. All the training is done in python 3.8 in a machine with 1 Titan RTX with 24GB, AMD Ryzen Threadripper PRO 5995WX 64-Cores CPU with 451 GB of RAM.

1) GCN implementation: The GCN model consists of three layers as shown in Fig. 1. Each layer consists of a GCNConv module, a batch normalization module, a ReLu activation function and a dropout. The last layer of the model contains only a GCNConv module and a LogSoftmax which yields the output. The input dimensionality is the number of features we have selected (21). Each intermediate (hidden) layer has 256 dimensions and the output layer has a dimensionality equal to the number of classes in the dataset (2). The learning rate is set to 0.01, dropout to 0.5 and the training was set to run for 500 epochs.

2) GraphSage implementation: The GraphSage implementation follows the architecture described in [11]. It consists of two message passing layers with 21 input dimensions, 32 hidden dimensions and 2 output dimensions. The dataset was divided into batches with a batch size of 32. Other parameters include: Adam optimizer, learning rate is set to 0.01, dropout probability equal to 0.5 and the training was set to run for 500 epochs.

3) GAT implementation: Our GAT implementation follows the architecture presented in [12]. It consists of 2 message passing layers with 21 input dimensions (equal to the number of node features), 32 hidden dimensions and 2 output dimensions. The number of attentions heads was set equal to 2. Similar to GraphSage, the dataset was divided into batches with batch size of 32, Adam optimizer was used, learning rate was set to 0.01, dropout equal to 0.5 and number of epochs equal to 500.

4) GraphSMOTE implementation: We have used the GraphSMOTE implementation from [16] available from the official GitHub page of the authors[7]. This model implementation expects the graph to be represented with an adjacency matrix while in a PyG dataset (ours as well) the graph is represented as a COO edge list. In order to run GraphSMOTE in our dataset we had to first convert the edge list to an adjacency matrix which was performed using Pytorch's built-in functions. The following parameter values have been used: GraphSage as embedding model, batch size was set to 40, learning rate 0.001, dropout probability equal to 0.1, epochs equal to 2000.

5) ReNode implementation: The implementation of the ReNode [17] algorithm used in this paper was adopted from the author's Github page[8]. Similar to the GraphSMOTE case also here we needed to adapt our dataset to the ReNode implementation especially in the case of the inductive settings. The main change was related to converting the edge list into a CSR adjacency matrix. Some of the parameter values used for this algorithm are: learning rate equal to 0.005, hidden layer dimensionality is set to 32, number of layers equal to 2, personalized page rank teleportation is set to 0.15.

6) Evaluation metrics: For every prediction problem, the correct evaluation of the results is very important. In the specific case of imbalanced data, choosing the appropriate evaluation metric is of critical importance. Our dataset has imbalanced data as can be seen from Imbalance Ratio, IR value from able II. Two classes Active versus Inactive users have IR values in the range from 0.037-0.062. In order to be considered balanced the dataset should have an IR equal to 1. For the above mentioned models we have used Macro F1 score and Macro recall to evaluate their performance in our dataset. **Recall** is one of the most used evaluation metrics. It gives accurate measurements regarding the detection of samples from the minority class. We can distinguish $recall^{+}$, also known as *sensitivity* and $recall^{-}$ also known as *specificity*. These two metrics can be calculated using the following formulas

$$recall^{+} = \frac{TP}{TP + FN}$$

$$recall^{-} = \frac{TN}{TN + FP}$$

where TP is the number of true positives and FN is the number of false negatives. Macro recall is the arithmetic mean of

[7] https://github.com/TianxiangZhao/GraphSmote

[8] https://github.com/victorchen96/ReNode

recalls for different classes without considering the importance of different classes. **F1 score** is another very important metric used to evaluate the performance of machine learning models. It is defined as

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

where TP = True Positive, FP = False Positive and FN = False Negative. Macro F1 score is the arithmetic mean of F1 score of each class.

B. *Results of Standard GNN Models (RQ1)*

Given the extreme imbalance ratio that our data has, it is to be expected that the standard GNN models will have problems to correctly classify nodes. Our implementations of GCN, GraphSage and GAT are really good at learning really fast how to cheat so they can achieve a high accuracy. After only five iterations the model reports a high accuracy of around 95%. Looking closely at the results one can notice that the model only predicts “TRUE”. By always predicting “TRUE” (the user is active) the model is certain to achieve around 95% accuracy because 95% of the users in the data is active. The reported Recall and F1 score for all these models is the same for our dataset. Sensitivity = 1 and Specificity = 0. These two metrics can be combined into Macro recall which in this case is equal to 0.5. The F1 score is equal to 0.97. After we were presented with these results, we noticed the need to test other models which are designed to handle imbalanced data.

C. *Results of Specific GNN Models (RQ2)*

The first specific GNN model that we tested on our data is GraphSMOTE [16]. Given the fact that GraphSMOTE is based on SMOTE [24] which in turn is one of the most used techniques in case of imbalanced non-graph data, this method is expected to perform better than other methods. After running the algorithm on our dataset, we noticed that this algorithm did classify some users as being inactive, in contrast to the standard GNN models. However, not all the predictions were correct. Calculating Sensitivity, Specificity and F1 for this method yields the following results: Sensitivity = 0.95, Specificity = 0.04, Macro recall = 0.49 and F1 score = 0.96. We notice a decrease in both these measures which gives the impression that this algorithm performs worse than the standard GNN models. One can argue that an algorithm which detects from both classes, even though not always correct, is better than an algorithm which is biased towards the majority class. Other evaluation metrics may capture this fact better than F1 score. The second algorithm that claims to handle imbalanced data and that we tested was ReNode [17]. This model classified more samples to be part of the minority class than what was expected. We ran both the inductive and transductive settings and also played with TINL and QINL settings, however the results were poor. This algorithm scored a Sensitivity of 0.37, Specificity of 0.67 and F1 score = 0.54 (see Table V).

TABLE V. Evaluation Metrics for the Implemnted Models

	Sensitivity	**Specificity**	**Macro F1**
GCN	1	0	0.97
GraphSage	1	0	0.97
GAT	1	0	0.97
GraphSMOTE	0.95	0.04	0.96
ReNode	0.37	0.67	0.54

V. Conclusions

In this paper we tackled the problem of fake news detection and spreading by providing a proactive solution: to detect and ban fake news spreaders before they become important spreaders. We created a dataset from November 2021 to July 2022 that contains information about the users features and relationships that spread news through tweets about vaccines. The problem of classification in imbalanced data is raised since our real data dataset was deeply imbalanced for all the periods that were taken into consideration. Three graph neural network models were trained in our datasets for node classification: Graph Convolutional Network, GraphSAGE, and Graph Attention Networks. They achieve high accuracy after few iterations because they learn really fast by predicting always True having Sensitivity = 1, Specificity = 0 and F1 = 0.97. In order to reach our goal to find and prevent the potential spreaders, increasing true negative which will yield increase in specificity becomes very important. We have tested two other frameworks to overcome the problem of imbalanced data: GraphSMOTE and ReNode. Both of these frameworks claim to give better results in imbalanced data. We found that GraphSMOTE does a better job in increasing the Specificity, but this comes at the cost of increasing the false positives rate. For this model we report Sensitivity = 0.95, Specificity = 0.04 and F1 score = 0.96. We found ReNode to not be as good as the authors claims in our case. For our dataset this technique scores Sensitivity = 0.37, Specificity = 0.67 and F1 score = 0.54.

Acknowledgment

This publication, is made possible with the financial support of National Agency for Scientific Research and Innovation(AKKSHI). Its content is the responsibility of the author, the opinion expressed in it is not necessarily the opinion of AKKSHI.

References

[1] S. J. Dixon. (2023, Aug.) Number of social media users worldwide from 2017 to 2027. [Online]. Available: https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/

[2] J. Zarocostas, “How to fight an infodemic,” *The Lancet*, vol. 395, no. 10225, p. 676, Feb. 2020.

[3] R. Datta, A. Yadav, A. Singh, K. Datta, and A. Bansal, “The infodemics of covid-19 amongst healthcare professionals in india,” *Medical Journal Armed Forces India*, vol. 76, no. 3, pp. 276–283, Jul. 2020.

[4] W. H. Organization. (2020, Sep.) Managing the covid-19 infodemic: Promoting healthy behaviours and mitigating the harm from misinformation and disinformation. [Online]. Available: https://www.who.int/news/item/23-09-2020-managing-the-covid-19-infodemic-promoting-healthy-behaviours-and-mitigating-the-harm-from-misinformation-and-disinformation

[5] H. Allcott and M. Gentzkow, “Social media and fake news in the 2016 election,” *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–236, May 2017.

[6] N. Kshetri and J. Voas, “The economics of “fake news”,” *IT Professional*, vol. 19, no. 6, pp. 8–12, Nov. 2017.

[7] A. Kucharski, “Study epidemiology of fake news,” *Nature*, vol. 540, no. 7634, pp. 525–525, Dec. 2016.

[8] J. Golbeck, M. Mauriello, B. Auxier, K. H. Bhanushali, C. Bonk, M. A. Bouzaghrane, C. Buntain, R. Chanduka, P. Cheakalos, J. B. Everett, W. Falak, C. Gieringer, J. Graney, K. M. Hoffman, L. Huth, Z. Ma, M. Jha, M. Khan, V. Kori, E. Lewis, G. Mirano, W. T. Mohn IV, S. Mussenden, T. M. Nelson, S. Mcwillie, A. Pant, P. Shetye, R. Shrestha, A. Steinheimer, A. Subramanian, and G. Visnansky, "Fake news vs satire: A dataset and analysis," in *Proceedings of the 10th ACM Conference on Web Science*, ser. WebSci '18. ACM, May 2018.

[9] A. Ceni, A. Kika, and D. X. Çollaku, "A social network analysis of COVID-19 vaccines tweets," in *Proceedings of the 5th International Conference on Recent Trends and Applications in Computer Science and Information Technology, Tirana, Albania, April 26-27, 2023*, ser. CEUR Workshop Proceedings, E. Xhina and K. Hoxha, Eds., vol. 3402. CEUR-WS.org, 2023, pp. 1–12. [Online]. Available: https://ceur-ws.org/Vol-3402/paper01.pdf

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[11] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 1024–1034. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html

[12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[13] Z. Huang, Y. Tang, and Y. Chen, "A graph neural network-based node classification model on class-imbalanced graph data," *Knowledge-Based Systems*, vol. 244, p. 108538, May 2022.

[14] Y. Liu, Z. Zhang, Y. Liu, and Y. Zhu, "Gatsmote: Improving imbalanced node classification on graphs via attention and homophily," *Mathematics*, vol. 10, no. 11, p. 1799, May 2022.

[15] Z. Zhu, H. Xing, and Y. Xu, "Balanced neighbor exploration for semi-supervised node classification on imbalanced graph data," *Information Sciences*, vol. 631, pp. 31–44, Jun. 2023.

[16] T. Zhao, X. Zhang, and S. Wang, "Graphsmote: Imbalanced node classification on graphs with graph neural networks," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, ser. WSDM '21. ACM, Mar. 2021.

[17] D. Chen, Y. Lin, G. Zhao, X. Ren, P. Li, J. Zhou, and X. Sun, "Topology-imbalance learning for semi-supervised node classification," Oct. 2021.

[18] I. A. Pilkevych, D. L. Fedorchuk, M. P. Romanchuk, and O. M. Naumchak, "Approach to the fake news detection using the graph neural networks," *Journal of Edge Computing*, vol. 2, no. 1, pp. 24–36, May 2023.

[19] S. Feng, H. Wan, N. Wang, and M. Luo, "Botrgcn: Twitter bot detection with relational graph convolutional networks," in *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ser. ASONAM '21. ACM, Nov. 2021.

[20] Q. Guo, H. Xie, Y. Li, W. Ma, and C. Zhang, "Social bots detection via fusing bert and graph convolutional networks," *Symmetry*, vol. 14, no. 1, p. 30, Dec. 2021.

[21] C. Li and D. Goldwasser, "Encoding social information with graph convolutional networks forpolitical perspective detection in news media," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019.

[22] E. Shushkevich, M. Alexandrov, and J. Cardiff, "Improving multi-class classification of fake news using bert-based models and chatgpt-augmented data," *Inventions*, vol. 8, no. 5, p. 112, Sep. 2023.

[23] J. Köhler, G. K. Shahi, J. M. Struß, M. Wiegand, M. Siegel, T. Mandl, and M. Schütz, "Overview of the CLEF-2022 checkthat! lab: Task 3 on fake news detection," in *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022*, ser. CEUR Workshop Proceedings, G. Faggioli, N. Ferro, A. Hanbury, and M. Potthast, Eds., vol. 3180. CEUR-WS.org, 2022, pp. 404–421. [Online]. Available: https://ceur-ws.org/Vol-3180/paper-30.pdf

[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002.

[25] A. M. Mahmoud, A. S. Desuky, H. F. Eid, and H. A. Ali, "Node classification with graph neural network based centrality measures and feature selection," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 2, p. 2114, Apr. 2023.

[26] H. T. Phan, N. T. Nguyen, and D. Hwang, "Fake news detection: A survey of graph neural network methods," *Applied Soft Computing*, vol. 139, p. 110235, May 2023.