A Graph Neural Network For Fuzzy Twitter Graphs

Georgios Drakopoulos[1], Eleanna Kafeza[2], Phivos Mylonas[1] and Spyros Sioutas[3]

[1]*Department of Informatics, Ionian University, Tsirigoti Sq. 7, Kerkyra 49100, Hellas*
[1]*College of Technological Innovation, Dubai Academic City, E-L1-108, UAE*
[3]*Computer Engineering and Informatics Department, University of Patras, Patras 26504, Hellas*

Abstract
Social graphs abound with information which can be harnessed for numerous behavioral purposes including online political campaigns, digital marketing operations such as brand loyalty assessment and opinion mining, and determining public sentiment regarding an event. In such scenarios the efficiency of the deployed methods depends critically on three factors, namely the account behavioral model, the social graph topology, and the nature of the information collected. A prime example is Twitter which is especially known for the lively activity and the intense conversations. Here an extensible computational methodology is proposed based on a graph neural network operating on an edge fuzzy graph constructed by a combination of structural, functional, and emotional Twitter attributes. These graphs constitute a strong algorithmic cornerstone for engineering cases where a properly formulated potential or uncertainty functional is linked to each edge. Starting from the ground truth in each individual vertex, the graph neural network progressively computes in an unsupervised manner a global graph state which can in turn be subject to further processing. The results, obtained using as a benchmark a recent similar graph neural network architecture along with two Twitter graphs, are promising.

Keywords
Fuzzy graphs, graph mining, graph neural networks, behavioral analytics, emotional polarity, Twitter

1. Introduction

Graph mining is an integral part of the interconnected era since it lays the groundwork for numerous applications across a wide array of financial and technological fields including among others social network analysis, database query optimization, graph signal processing (GSP), supply chain and logistics networks, and brain circuit analysis. In this context modeling a graph in terms of vertices, connectivity patterns, and associated features is tantamount to data model selection. Edge fuzzy graphs extend classical graphs as probabilities drawn from a single distribution which may well have unknown parameters to be estimated. Said distribution is closely linked to the semantics and functional nature of the underlying graph. For instance, in a transportation network edge existence probabilities can show how likely a specific road is to be blocked from snow in winter months, whereas in a computer network they may model the chance of a virus being propagated along a given link.

In order to compute an estimation of the global graph state which allows not only a higher level overview but also subsequent processing, in this work will be used a graph neural network (GNN) architecture. GNNs constitute a class of unsupervised neural networks where each vertex, representing a processing node, starts with a local ground truth information vector and iteratively a global status is derived based on the fundamental fact that graphs contain inherently higher order information in a distributed manner. The resulting graph global state can be subsequently further processed in order to derive global properties such as community discovery.

The primary research objective of this conference paper is the development of a GNN architecture designed for edge fuzzy Twitter graphs constructed from incorporating structural, functional, and behavioral features. The proposed methodology can be inherently extended to other possible attribute types, making it thus appropriate for mining graphs originating from social media or evolving computational ecosystems for that matter. This work differentiates itself from previous ones in two aspects, namely the fusion of various heterogeneous attributes and the induced edge fuzzy topology.

The remaining of this work is structured as follows. In section 2 the recent scientific literature regarding GNNs, graph mining, and computational behavioral science is briefly reviewed. The proposed methodology along with the relevant intuition are given in 3. The results obtained from the experiments are the focus of section 4. Future research directions are given in 5. Technical acronyms are defined the first time they are encountered in the text. Finally, the notation of this conference paper is summarized in table 1.

CIKM'21: 30th ACM International Conference on Information and Knowledge Management, November 01–05, 2021, Virtual Event, QLD, Australia
c16drak@ionio.gr (G. Drakopoulos); eleana.kafeza@zu.ac.ae (E. Kafeza); fmylonas@ionian.gr (P. Mylonas); sioutas@ceid.upatras.gr (S. Sioutas)
0000-0002-0975-1877 (G. Drakopoulos); 0000-0001-9565-2375 (E. Kafeza); 0000-0002-6916-3129 (P. Mylonas)

Table 1
Notation of this conference paper.

Symbol	Meaning	First in
$\triangleq$	Definition or equality by definition	Eq. (1)
$\{s_1, \ldots, s_n\}$	Set with elements $s_1, \ldots, s_n$	Eq. (2)
$(t_1, \ldots, t_n)$	Tuple with elements $t_1, \ldots, t_n$	Eq. (1)
$\|S\|$	Set cardinality functional	Eq. (3)
$\text{prob}\{\Omega\}$	Probability of event Ω occurring	Eq. (4)

2. Previous Work

GNNs operate on irregular domains expressing relationships. Heterogeneous GNN architectures are examined in [1] and representative GNNs designed to complete versatile tasks in [2]. Edge labeling is proposed in [3] in the context of few-short learning for GNNs. The technique of aggregated neural path in conjunction with machine learning (ML) tasks is described in [4]. The emotional coherency of Twitter graphs with GNNs is explored in [5], whereas in [6] are given guidelines for social recommendation based on GNNs.

Graph mining is a mainstay of current ML [7]. In a graph signal processing (GSP) context adjacency matrices are considered as two-dimensional signals and signal processing techniques are then employed to extract patterns of interest [8]. An overview of the connections to deep learning are given in [9]. In [10] a tensor stack network (TSN) is trained to estimate the topological correlation of graph pairs compressed with the two-dimensional discrete cosine transform (DCT2), while the same architecture evaluates graph resiliency in [11]. Flow-based GSP is examined in [12]. The basic operations of GSP such as shifting and sampling are defined in [13]. A graph version of the LMS adaptive filtering algorithm is presented in [14]. A versatile and space efficient data structure for persistent graphs is described in [15].

Behavioral attributes have recently emerged as an integral part of many recent computational systems [16]. The connection between behavioral systems and data driven analysis is explored in [17]. Digital trust is a paramount factor for recruiting candidates from LinkedIn [18]. Clustering fMRI images with tensor distances for emotion recognition [19], while gamification strategies are explored in [20]. An overview of behavioral systems is given in [21].

3. Proposed Architecture

In this section the proposed GNN architecture as well as the notions underlying it are described.

3.1. Fundamental concepts

In order to describe the proposed architecture a few basic concepts must be first revised or defined. First the class of edge fuzzy graphs is introduced in definition 1.

Definition 1 (Edge fuzzy graph). *An edge fuzzy graph is a combinatorial object represented by the ordered triplet shown in equation (1).*

$$G \triangleq (V, E, h) \tag{1}$$

The elements in (1) have the following meaning:

- *The vertex set V. In the context of this work each vertex corresponds to a single Twitter account through a bijection.*
- *The set of fuzzy edges E where $E \subseteq V \times V$. The connectivity patterns therein reflect the underlying graph dynamics.*
- *The functional $h : E \to [0, 1]$ maps each edge to a probability drawn from a single distribution. These result from graph semantics and functionality.*

In the general case the digital account behavior for any online social network is given in definition 2.

Definition 2 (Account behavior). *The online behavior of an account consists of the total peer interaction over all possible ways offered by the given social medium.*

The above definition can be readily extended in the case two or more accounts are connected over multiple social media, expanding thus the interaction potential. However, this is outside the scope of this work.

In this work the online behavior of Twitter accounts has three distinct components, namely the *follow* relationships, retweet patterns, and emotional polarity with respect to a reference hashtag set. The intuition behind their selection is as follows:

- The *follow* relationships capture the structural aspect of the Twitter graph since they constitute the core of its edges.
- The retweet patterns are an integral part of the functionality taking place bridging accounts in a different way.

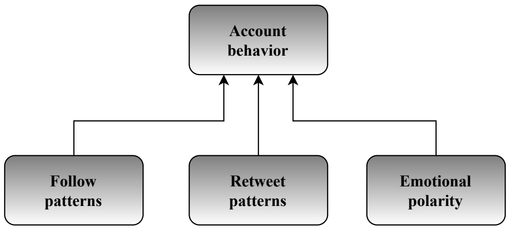

Figure 1: Behavioral model.

- The emotional coherency is a factor evaluating the similarity of sentiments towards selected topics expressed as hashtags.

The above are also shown in figure 1.

In order to model the behavioral aspects of the Twitter accounts, a set of the most common hashtags from each graph is selected. The inspiration for the selection of such a set is the concept of node cover. Definition 3 clarifies it.

Definition 3 (Reference hashtag set). *Let H_0 be the set of hashtags in a Twitter graph T. A hashtag $h \in H_0$ is also belongs to the reference hashtag set if and only if the accounts who have used h constitute a vertex cover for T.*

Let H be the set of hashtags satisfying definition 3.

$$H \triangleq \{h_1, \ldots, h_p\} \tag{2}$$

The ratio of the cardinality of H to that of H_0 can be taken as a measure of the important information existing in the underlying Twitter graph as in (3):

$$\rho \triangleq \frac{|H|}{|H_0|} \tag{3}$$

Using the notion of the vertex set to find popular hashtags has the following advantages:

- Selecting hashtags h does not depend on any hyperparameters or on any thresholds whatsoever.
- The widespread use of hashtag h is a clear indication of its popularity.
- Although vertex cover is an NP-hard problem, approximation algorithms for it exist.

However, it should be noted that for larger benchmark graphs or for dynamic ones alternative criteria should be sought in order to avoid the overwhelming complexity of determining a vertex cover.

3.2. Architecture

The proposed GNN architecture relies on the fundamental fact that edges are fuzzy, namely that they belong to the graph with a certain probability which in the general case depends on an attribute set. The latter is frequently strictly local or a function of a small neighborhood and rarely global since updating such a set is costly and prone to dependency bottlenecks. In the context of this conference paper the probability $p_{i,j}$ for edge $e_{i,j}$ between vertices v_i and v_j is computed as in equation (4):

$$\mathrm{prob}\{e_{i,j}\} \triangleq p_{i,j} = w_f F_{i,j} + w_r \frac{r_{i,j}}{R} + w_c c_{i,j} \tag{4}$$

In (4) three factors are taken into consideration:

- Whether there is a directed *follow* link from the i-th account to the j-th one denoted by the binary indicator $F_{i,j}$.
- The ratio of the number $r_{i,j}$ of retweets of the i-th account coming from the j-th one to the total retweets R in the graph.
- The signed correlation factor $c_{i,j}$ expressing the emotional coherency of the i-th and j-th accounts with respect to the reference hashtag set.

The sentiment $l^{[t]}$ of the i-th account during iteration t consists of a vector containing an emotional polarity score, namely the percentage of the how positive, neutral, of negative the i-th account feels towards the as shown in (5). Initially the ground truth vector of the i-th account is the respective average percentage of positive, neutral, or negatively charged words in the tweets containing at least one hashtag from the reference set.

$$l_i^{[t]} \triangleq \begin{bmatrix} n_{p,i} & n_{n,i} & n_{g,i} \end{bmatrix}^T \tag{5}$$

Given the iteration-dependent value of $l^{[t]}$, the value of the correlation factor $c_{i,j}$ should be computed during each iteration as shown in (6). It should be highlighted that $c_{i,j}$ is the only term of (4) which is signed, thereby reinforcing or weakening the strength between two accounts depending on whether they have similar sentiments towards the reference hashtag set.

$$\frac{\sum_{k=1}^{3}\left(l_i^{[t]}[k] - 1/3\right)\sum_{k=1}^{3}\left(l_j^{[t]}[k] - 1/3\right)}{\sqrt{\sum_{k=1}^{3}\left(l_i^{[t]}[k] - 1/3\right)^2}\sqrt{\sum_{k=1}^{3}\left(l_j^{[t]}[k] - 1/3\right)^2}} \tag{6}$$

The weights of the linear combination in (4) encode the sign and relative strength of each factor contribution, namely how much each factor participates to the edge probability existence and whether such participation reinforces or weakens said probability respectively. Moreover, they ensure the numerical stability of $p_{i,j}$.

Intuitively speaking, equation (4) is a linear estimator of the true edge existence probability. The weights w_f, w_r, and w_c express the relative contribution of each term and in our experiments follow the semantic strength of the respective factor. This means that w_f is higher since the *follow* denotes a high degree of coherency between the two accounts. Along a similar line of reasoning, frequent retweets between two accounts indicate a somewhat strong connection between them. Moreover, a consistent emotional coherency between two accounts may well suggest a behavioral link between them.

Additionally the weight $\delta_{i,j}$ assigned to each edge is a function of the strength of the corresponding edge.

$$\delta_{i,j} \triangleq f(p_{i,j}) \tag{7}$$

The weight function $f(\cdot)$ of (7) is the same for each edge and it is directly or at least indirectly linked to the semantics of the underlying graph. One of the most common options is that shown in (8).

$$\delta_{i,j} \triangleq \frac{1}{p_{i,j}} \tag{8}$$

However, the weight selection of (8) has the disadvantage of being almost singular close to zero, generating thus excessive weight values. A viable alternative is the *inverse linear* weight function of (9).

$$\delta_{i,j} \triangleq \frac{1}{1+p_{i,j}} \tag{9}$$

Another option for the weight function is that the *inverse square* function of equation (10). The latter typically expresses a potential function in various applications.

$$\delta_{i,j} \triangleq \frac{1}{1+p_{i,j}^2} \tag{10}$$

In figure 2 the weight functions of (9) and (10) are shown for their entire range. It can be immediately inferred they are strictly decreasing and everywhere smooth, expressing the fact that the more likely is an edge to belong to the graph, the easier to cross it.

At the core of the proposed GNN architecture is the update mechanism of (11). For the i-th vertex the $l_i^{[t]}$ is computed as in (11). Therein the index j ranges over all inbound neighbors of the i-th vertex and thus it depends on local connectivity patterns. However, since the state vector of its neighbors depends on recursively on that of its own vectors, this mechanism is essentially a higher order status computation. During an update it may be possible that certain neighbors may have already had their own state vectors updated, whereas others not. Thus, the iteration indicator $*$ will be used. This process terminates when the state vectors remain unchanged under a threshold of η_0 for three consecutive iterations.

$$l^{[t+1]} = \varphi\left(\frac{\beta_0}{2} l^{[t-1]} + \frac{\beta_0}{2}\sum_j \frac{\delta_{i,j}}{\Delta} l_j^{[*]}\right) \tag{11}$$

The hyperparameter β_0 scales input to a practical domain for the sigmoid function $\varphi(\cdot)$, $\delta_{i,k}$ is the weight of the edge, and Δ is the sum of the edge weights of the inbound neighbors. In (11) the sigmoid function is defined as in (12) which is differentiable and smooth everywhere.

$$\varphi(s;\sigma_0) \triangleq \frac{1}{1+\exp(-\sigma_0 s)} \tag{12}$$

The derivative of the sigmoid function is given in (13).

$$\begin{aligned} \frac{\partial \varphi(s;\sigma_0)}{\partial s} &= \sigma_0 \varphi(s;\sigma_0)\,\varphi(-s;\sigma_0) \\ &= \sigma_0 \varphi(s;\sigma_0)\,(1-\varphi(s;\sigma_0)) \end{aligned} \tag{13}$$

The last form of (13) comes from the fundamental property of the sigmoid function described in (14) below:

$$\varphi(s;\sigma_0) + \varphi(-s;\sigma_0) = 1 \tag{14}$$

The preceding properties ensure that $\varphi(\cdot)$ is smooth enough to prevent divergence in most cases for a broad spectrum of distributions.

4. Results

The results of the proposed GNN methodology are presented in this section along with intuition about them. They are divided to four groups, one for each possible combination of benchmark graph (1821 / US2020) and weight function (inverse linear / inverse square).

4.1. Dataset

The two benchmark graphs used in the experiments were taken from [5]. They represent two characteristic cases of social graphs, namely one with a relative quiet and coherent one (1821) and one containing heated conversations and a considerable degree of dissonance (US2020). The Twitter sampling interval was 8/2020-10/2020.

4.2. Number of iterations

In table 3 the parameters used in the experimental setup of this work are shown. This allows for the easy exploration of the parameter space. Observe that the actual values of these parameters are in accordance of the strength of the respective factor.

Table 4 contains the normalized number of iterations as a function of the hyperparameter β_0 of (11) for the two benchmark graphs of table 2 and for the two possible weight functions shown in equations (9) and (10). Normalization takes place per graph and per weight function in order to show the comparative effect of β_0 in each case. In order to demonstrate the effect of the emotional attribute $c_{i,j}$ of (4) on the convergence rate the same

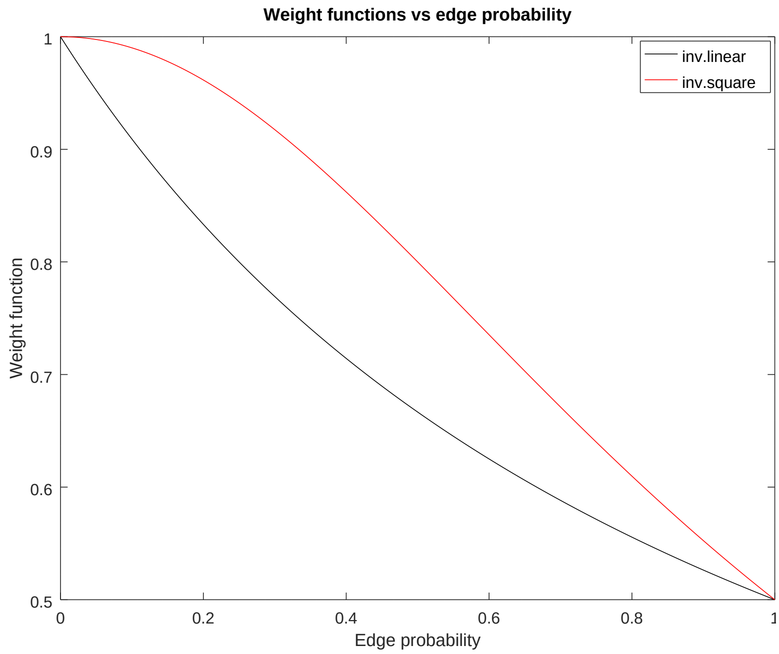

Figure 2: Weight functions.

Table 2
Dataset synopsis (from [5]).

Property	1821 graph	US2020 graph
Number of vertices	132.317	147.881
Number of edges	2.225.177	2.447.224
Density / Log-density	16.8170 / 1.2393	16.5486 / 1.2357
Completeness / Log-completeness	$2.54e^{-4}$ / 0.6196	$2.38e^{-4}$ / 0.6173
Number of triangles	446.513	489.773
Number of squares	215.387	218.633
Number of cliques of size four	102.044	125.806
Graph diameter	10	11
Percentage of vertices reachable at diameter-1	95.33%	98.17%
Percentage of vertices reachable at diameter-2	93.26%	96.44%
Percentage of vertices reachable at diameter-3	89.11%	91.22%
Percentage of vertices reachable at diameter-4	84.73%	87.47%
Number of favorites	36.994.815	42.114.509
Number of tweets	17.465.844	22.773.674

Table 3
Parameters of the experiments.

Parameter	Meaning	Value
w_f	Edge *follow* weight	0.5
w_r	Edge retweet weight	0.25
w_c	Edge hashtag emotional coherence	0.25
η_0	State vector equality threshold	0.05

Table 4
Normalized number of iterations as a function of the hyperparameter β_0.

Hyperparameter	Graph	inv.linear	inv.linear+beh	inv.square	inv.square+beh
0.5	1821	1.49	1.26	1.37	1.38
0.7		1.45	1.24	1.32	1.23
0.8		1.39	1.16	1.27	1.15
0.9		1.33	1.08	1.22	1.07
1		1.28	1	1.19	1
0.5	US2020	1.41	1.17	1.33	1.13
0.7		1.42	1.14	1.29	1.11
0.8		1.37	1.09	1.25	1.08
0.9		1.29	1.05	1.21	1.03
1		1.24	1	1.18	1

Table 5
Emotional distributions (pos/neu/neg) computed with the best value of β_0.

Graph	init	i.linear	i.linear+beh	i.square	i.square+beh
1821	0.64/0.24/0.12	0.68/0.14/0.18	0.73/0.15/0.12	0.69/0.15/0.16	0.74/0.14/0.12
US2020	0.28/0.23/0.49	0.21/0.22/0.57	0.17/0.16/0.67	0.22/0.22/0.56	0.16/0.16/0.68

GNN is run with the latter removed from the initial local ground truth vectors at the vertices.

From table 4 it follows immediately that the inclusion of the behavioral factor in (4) leads to quicker convergence of the proposed GNN architecture. This can be attributed to the following reasons:

- **Information enrichment:** From an algorithmic perspective, the behavioral factor adds an independent dimension to the profile of each vertex. Hence, the new vertex profile space can differentiate adequately between dissimilar vertices while maintaining close enough similar ones.
- **Numerical variation:** The above is enhanced by having more diversified edge weights. Besides the additional factor, the *behavioral term* is also signed. In turn this expands the range of weights, increasing thus the possible number of values.

The above factors suggest that variability in the weight space as well as in the vertex profile increase the flexibility of the update mechanism of (11). This is in accordance with the standard pattern recognition maxim stating that mapping data to a space of higher dimensionality facilitates their clustering. On the other hand, the curse of dimensionality imposes a limit on how big this new space can get. As both spaces used in this work however are low dimensional, this does not constitute a problem.

Regarding the total sentiment, in table 5 is shown the average emotional distribution before and after the GNN execution in each case using the value of hyperparameter β_0 which leads to the quickest convergence in each case. There it can be seen that the US2020 yields for both weight choices slightly different results from the initial distribution when the emotional factor is excluded but considerably different ones when they are included. Thus, it is a graph with a heavy emotional charge. On the other hand, the 1821 graph tends to yield similar results in every case, signifying thus greater coherency.

5. Conclusions And Future Work

This conference paper focuses on a graph neural network architecture for discovering community structure in large Twitter graphs. In this approach said structure is formed using a Twitter account behavioral model which results from fusing structural and functional attributes with emotional ones. The proposed model can be naturally extended to include additional features from these categories or even ones belonging to different categories as long as they can be expressed in a numerical scale where normalization does not influence semantics. In our experiments the inclusion of behavioral attributes leads consistently to quicker GNN convergence.

This work can be extended in a number of ways. First, multiple weight functions can map each edge to a weight vector and hence to a multidimensional weight space where each dimension has its own semantics. Then the fundamental parameters of candidate distributions describing this space can be derived through signal estimation techniques. Second, alternative behavioral models depending only on local properties or on local estimates of global ones should be developed as this would be most appealing for a distributed implementation. Third, models for computing or estimating the edge existence probability which reflect the underlying graph semantics should be research objectives. Finally, given the evolving nature of online social networks, an architecture for non-stationary graphs should also be developed. In this case certain transient global graph states can be used to obtain intermediate status results regarding clustering, flow, degree distribution, or any other global property. These transient states can well serve as starting points for new partitioning techniques.

Acknowledgments

This conference paper was supported by the Research Incentive Fund (RIF) Grant R18087 provided by Zayed University, UAE.

References

[1] C. Zhang, D. Song, C. Huang, A. Swami, N. V. Chawla, Heterogeneous graph neural network, in: ICDM, 2019, pp. 793–803.

[2] C. Yu, Y. Liu, C. Gao, C. Shen, N. Sang, Representative graph neural network, in: ECCV, Springer, 2020, pp. 379–396.

[3] J. Kim, T. Kim, S. Kim, C. D. Yoo, Edge-labeling graph neural network for few-shot learning, in: CVPR, 2019, pp. 11–20.

[4] X. Fu, J. Zhang, Z. Meng, I. King, Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding, in: WWW, 2020, pp. 2331–2341.

[5] G. Drakopoulos, I. Giannoukou, P. Mylonas, S. Sioutas, A graph neural network for assessing the affective coherence of Twitter graphs, in: IEEE Big Data, IEEE, 2020, pp. 3618–3627. doi:10.1109/BigData50022.2020.9378492.

[6] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, in: The WWW conference, 2019, pp. 417–426.

[7] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, P. Vandergheynst, Graph signal processing: Overview, challenges, and applications, Proceedings of the IEEE 106 (2018) 808–828.

[8] G. Mateos, S. Segarra, A. G. Marques, A. Ribeiro, Connecting the dots: Identifying network structure via graph signal processing, IEEE Signal Processing Magazine 36 (2019) 16–43.

[9] M. Cheung, J. Shi, O. Wright, L. Y. Jiang, X. Liu, J. M. Moura, Graph signal processing and deep learning: Convolution, pooling, and topology, IEEE Signal Processing Magazine 37 (2020) 139–149.

[10] G. Drakopoulos, E. Kafeza, P. Mylonas, L. Iliadis, Transform-based graph topology similarity metrics, NCAA 33 (2021) 16363–16375. doi:10.1007/s00521-021-06235-9.

[11] G. Drakopoulos, P. Mylonas, Evaluating graph resilience with tensor stack networks: A keras implementation, NCAA 32 (2020) 4161–4176. doi:10.1007/s00521-020-04790-1.

[12] M. T. Schaub, S. Segarra, Flow smoothing and denoising: Graph signal processing in the edge-space, in: GlobalSIP, IEEE, 2018, pp. 735–739.

[13] A. Gavili, X.-P. Zhang, On the shift operator, graph frequency, and optimal filtering in graph signal processing, IEEE Transactions on Signal Processing 65 (2017) 6303–6318.

[14] F. Hua, R. Nassif, C. Richard, H. Wang, A. H. Sayed, A preconditioned graph diffusion LMS for adaptive graph signal processing, in: EUSIPCO, IEEE, 2018, pp. 111–115.

[15] S. Kontopoulos, G. Drakopoulos, A space efficient scheme for graph representation, in: ICTAI, IEEE, 2014, pp. 299–303. doi:10.1109/ICTAI.2014.52.

[16] T. M. Cihon, M. A. Mattaini, Emerging cultural and behavioral systems science, Perspectives on behavior science 42 (2019) 699–711.

[17] I. Markovsky, F. Dörfler, Behavioral systems theory in data-driven analysis, signal processing, and control, Annual Reviews in Control (2021).

[18] G. Drakopoulos, E. Kafeza, P. Mylonas, H. Al Katheeri, Building trusted startup teams from LinkedIn attributes: A higher order probabilistic analysis, in: ICTAI, IEEE, 2020, pp. 867–874. doi:10.1109/ICTAI50040.2020.00136.

[19] G. Drakopoulos, I. Giannoukou, P. Mylonas, S. Sioutas, On tensor distances for self organizing maps: Clustering cognitive tasks, in: DEXA, volume 12392 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 195–210. doi:10.1007/978-3-030-59051-2_13.

[20] S. Kim, K. Song, B. Lockee, J. Burton, What is gamification in learning and education?, in: Gamification in learning and education, Springer, 2018, pp. 25–38.

[21] N. Wilkinson, M. Klaes, An introduction to behavioral economics, Macmillan International Higher Education, 2017.