Hybrid Graph Embedding Techniques in Estimated Time of Arrival Task

Vadim Porvatov^{1,2*}, Natalia Semenova^{1*}, and Andrey Chertok^{1,3}

¹ Sberbank, Moscow 117997, Russia,

² National University of Science and Technology "MISIS", Moscow 119991, Russia, ³ Artificial Intelligence Research Institute (AIRI), eighonet@gmail.com semenova.bnl@gmail.com achertok@sberbank.ru

Abstract. Recently, deep learning has achieved promising results in the calculation of Estimated Time of Arrival (ETA), which is considered as predicting the travel time from the start point to a certain place along a given path. ETA plays an essential role in intelligent taxi services or automotive navigation systems. A common practice is to use embedding vectors to represent the elements of a road network, such as road segments and crossroads. Road elements have their own attributes like length, presence of crosswalks, lanes number, etc. However, many links in the road network are traversed by too few floating cars even in large ridehailing platforms and affected by the wide range of temporal events. As the primary goal of the research, we explore the generalization ability of different spatial embedding strategies and propose a two-stage approach to deal with such problems.

Keywords: Graph Embedding, Machine Learning, ETA, Geospatial Linked Data.

1 Introduction

The modern state of traffic induces a remarkable number of forecasting challenges in a variety of related areas. According to the industrial needs, a relevant computation of the estimated time of vehicle arrival can be considered as one of the most actual problems in the logistics domain. In particular, intelligent traffic management systems [15] require significant accuracy in case of arrival time estimation. Besides such an application, computation of ETA also appears as a common issue in the commercial areas which are strongly dependent on optimal routing. The explicit examples of such services are taxi [19], railway [17], vessels [14] and aircraft transportation [3].

Accurate prediction of ETA for cars is a complex task requiring the relevant processing of heterogeneous data. It is frequently represented as time series and graph structure with feature vectors associated with its nodes and/or

^{*} equal contribution



Fig. 1. Demonstration of temporal traffic dynamics: cumulative frequencies of car activity and distribution of trips duration for Abakan and Omsk in the two hours interval.

edges. In comparison with other vehicles, computation of ETA for cars is considerably influenced by the road network topology, nonlinear traffic dynamics, unexpected temporal events, and unstable weather conditions, Figure 1. The stochastic nature of the introduced problem requires an implementation of a powerful domain-specific regression model with a high generalization ability.

Machine learning proved its outstanding efficiency in a wide range of regression tasks. However, not every model can be efficiently applied to the ETA forecasting due to the mentioned constraints of available data. Previously performed attempts of a simple model implementation (e. g., linear regressions and gradient boosting) were reported as inefficient [11, 27], while the more sophisticated approaches allowed to achieve more optimistic results [21]. Thus, in order to obtain a better performance, we assume the necessity of applying graph neural networks [29] as a part of the presented pipeline.

According to the extensive growth of graph machine learning in recent years, many promising architectures [10, 16] emerged and soon were applied in a wide range of graph-related studies [6, 18]. These models quickly became useful in terms of feature extraction in downstream tasks. Applied to the underlying graph structure of a city road network, such algorithms have the potential to dramatically increase the expressiveness of regression models and therefore should be explored.

In the present paper, we propose and compare different architectures of the hybrid graph neural network for ETA prediction. Our main contributions are the following:

 We introduce and publish the first to our best knowledge dataset¹ with intermediate trip points. This dataset is relevant for consistent ETA prediction task and future usage as a benchmark. We provide common information

 $^{^{1}}$ to receive an access to data you need to send a request to semenova.bnl@gmail.com

about trips and city road network as well as road structural properties, marking, and weather conditions (other features are described in Section 3 in detail). Additionally, the route data includes auxiliary information which can be used both for evaluation of the ETA and independent prediction of real traveled distance as a separate problem.

- Absence of methodological review of subgraph embeddings in the domain of interest encourages us to overwhelm such a limitation. Instead of focusing on more general approaches which include both spatial and recurrent temporal aspects, we prefer to precisely explore the domain of spatial embeddings as an underdeveloped one at the present moment.
- We conduct a comprehensive evaluation of our method on two real-world datasets which correspond to tangibly different cities. Obtained results of computational experiments motivate us to further develop our research in accordance with achieved significant performance improvements.

2 Related work

As it has been mentioned above, the ETA-related tasks are a fundamental part of logistic services. In overwhelming number of cases, they demand two properties from the predictive algorithms: computational efficiency and relevant accuracy. The first part of this challenge was unequivocally solved by simple learning models like gradient tree boosting, multi-layer perceptron, and linear regression. However, the quality of these models cannot be reported as sufficient even beyond the commercial logistics.

Along with the simple learning models, deterministic algorithms were also developed in huge amount [2,26]. In the majority of cases they cannot be compared with learning models in terms of quality. However, some of them were inspiring enough to influence the future development of their concepts in a more sophisticated way.

Limitations of mentioned approaches were partially overwhelmed in DeepTTE [25] and MURAT [13]. The first approach includes a recurrent neural network (RNN) which subsequently predicts the travel time along the trip. As many other recent methods, this algorithm is dependent on intermediate GPS coordinates. At the same time, the second method is closely related to the proposed architecture in the sense of graph embedding usage. In spite of the deep development of the temporal forecasting part, no more than one spatial embedding method was observed in any of this papers.

The most recent studies introduce new solutions with the potential to significantly increase the quality of ETA prediction. WDR [27] is a wide-deep architecture that outperformed a lot of previously established approaches. Its further improvement and computational experiments led the same authors to the design of RNML-ETA architecture [21] which allows to achieve even better results. Simultaneously, another intriguing paper [4] emerged as a prospective modification of ST-GCN methods family [7, 20, 28]. All of these methods use datasets with intermediate points in contrary to the overwhelming majority of early papers. Following this positive trend, we continue studies in the same direction.



Fig. 2. Edges usage frequencies projected as a heatmap on the road networks of Abakan (a) and Omsk (b). The patterns of edges demand are clearly distinguishable as the topology of networks remains significantly different.

3 Data

In the present work, we use two datasets related to the city networks of Abakan and Omsk. The cities have significantly different scales. Hence, their infrastructure pattern cannot be compared directly. Such a diversity allows us to check the generalization ability of the proposed architectures in a more explicit way. General properties of the dataset are established in Table 1 when the frequencies of road network segments usage are represented in Figure 2(a, b).

Each dataset consists of both road networks and the routes associated with their edges. City networks contain an abundant number of meaningful features that can be translated to the predictive model in different ways. The route sample includes information about the start and destination point and a set of visited nodes during the ride.

The trip data was collected in the period from December 1, 2020 up to December 31, 2020 by subsidiary companies of Sberbank. A comprehensive description

Property	Abakan	Omsk
Nodes	65524	231688
Edges	340012	1149492
Total trips number	119986	120000
Trips coverage	0.535	0.392
Edges usage median	12	8

Table 1. Description of the datasets in terms of common networks characteristics

of the proposed data is given in Table 2 for the city network and in Table 3 for car routes.

Feature	Values	Description
Road class	fake road, intra-quarter driveway, dirt road, other city street, main city street, highway, intercity road, federal highway, cycle path, walkway	General road segments categories
Length	\mathbf{Z}_+	Length of a road segment in meters
Width	\mathbf{Z}_+	Width of a road segment in meters
Def speed	$\{3, 15, 20, 60, 90\}$	Speed limit on a road section in km/h
Lanes	$\{0, 1, 2, 3, 4, 5\}$	Number of lanes in a road segment
Barrier	$\{0, 1\}$	Defines the presence of road barriers
Payment flag	$\{0, 1\}$	Defines a road segment as toll
Turn restrictions	$\{0, 1\}$	Defines an ability to turn on a road section
Pedo offset	$\{0, 1\}$	Defines the presence of crosswalk offsets
Bad road	$\{0, 1\}$	Defines the condition of a road segment
Style	undefined, archway, crosswalk, stairway, bridge, overground way, invisible, normal, park path, park footpath, subway, pedestrian bridge, underground way, tunnel, living zone, ford	Additional road segments categories

 Table 2. Edge features of city network

According to the complexity of input data, it cannot be directly translated to a predictive model as an input. In order to correctly solve the desired task, it is recommended to filter the established dataset and perform feature engineering. Trips that have a rebuild count more than 1 should be optionally separated from the main volume of routes as well as anomaly short and long routes. Values of start (finish) point parts and dist to a(b) can be also added or subtracted from the total estimated length of the route in order to obtain a better spatial resolution of subgraph embeddings.

Feature	Values	Description		
Nodes	$\{\hat{V}\subset V\}$	Subset of nodes		
Dist to a	\mathbf{Z}_+	Length of a segment between actual start point and its projection on the first edge Length of a segment between actual end point and its projection on the last edge		
Dist to b	\mathbf{Z}_+			
Start point part	\mathbf{Z}_+	Part of the first edge where the trip starts in meters		
Finish point part	\mathbf{Z}_+	Part of the last edge where the trip ends in meters		
Start UTC	\mathbf{Z}_+	Start time of the trip in UTC format		
Real time of arrival	$\mathbf{Z}_{+}^{'}$	Trip duration in seconds		
Real dist [*]	\mathbf{Z}_+	Actual traveled distance in meters		
Rebuild count [*]	\mathbf{Z}_+	Number of route rebuilds that corresponds to the destination change		

Table 3. Features of trip dataset

4 Methods

The task can be mathematically formulated as a regression problem that extended by a special procedure of an automatic feature engineering. In order to handle this challenge, we generate vector representations of the road segments via GNNs, aggregate them to the trips embeddings and then apply a regression model which predicts ETA.

Given a graph G = (V, A, X) of the city road network, where $V = \{v_1, v_2, ..., v_n\}$ denotes the set of graph vertexes (road segments), $A: n \times n \to \{0, 1\}$ denotes the adjacency matrix (each edge encodes connectivity of the road segments), and $X: n \times m \to \mathbf{R}$ is a matrix of node features.

The goal is to compute such a representation of each node $v_i \in V$ that can be effectively aggregated in accordance with structural properties of the route $s_j := \{v_{j_1}, ..., v_{j_t}\}, s_j \in S$. There are two main aggregation strategies that potentially allow to construct a meaningful route subgraph embedding. The first one based on basic summation of all representations of the nodes that are included to the exact route

$$z_{s_j} = \sum_{i=1}^{\# s_j} Z(v_{j_i}),\tag{1}$$

where $Z(\cdot)$ is the node embedding function.

Another approach related to initial graph extension by virtual nodes. This procedure induces a new graph $\hat{G}(V', A', X')$, where $V' = \{v_1, ..., v_n, v_{n+1}, ..., v_{n+\#S}\}$, $A':(n + \#S) \times (n + \#S) \rightarrow \{0, 1\}, \forall v_i, v_n \in V$ adjacency matrix defined as $A'(v_i, v_j) = A(v_i, v_j)$. For the other edges, we propose the bijective function

 $f: V' \setminus V \to S$ that defines $\forall v'_k \in V' \setminus V$ and $\forall v_l \in f(v'_k)$ values in remaining part of the extended adjacency matrix as $A'(v_l, v'_k) = 1$. In agreement with this method,

$$z_{s_j} = Z(f^{-1}(s_j)). (2)$$

For both strategies it is crucial to find the appropriate node embedding function $Z(\cdot)$ which has a significant impact on the relevance of the final route subgraph representations. We propose graph convolutional networks [10], GAT [23], and GraphSAGE [8] as the main candidates for nodes representation learning. The ideas behind these methods are quite similar as they all encode nodes to vectors of a fixed size via a repeated aggregation over a local neighborhood. However, while the GCN is based on mean aggregation, GraphSAGE pretends to be a more flexible and representative instrument due to its different aggregators and embedding concatenation stage. On the other hand, GAT adopts the mechanism of attention [22] firstly proposed in Natural Language Processing (NLP) to the needs of graph machine learning. To explicitly reveal the relevance of the mentioned approaches, in the following we briefly introduce the main aspects of each method.

Graph Convolutional Network (GCN). For a given graph G(V, A, X) this method defines an effective approach to network information aggregation. Single graph convolution layer is its atomic unit that can be represented as

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right),$$
(3)

where l + 1 is the current convolution layer number, σ is an arbitrary nonlinear function (e. g., ReLU), $H^{(0)} = X$, $\tilde{A} = A + I_N$, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and W^l is the matrix of learning parameters.

GraphSAGE. This algorithm mostly inherits the notation of convolutions from the GCN architecture, but instead of using full graph it directly computes convolution for each node v in the iterative manner

$$h_v^{l+1} = \sigma \left(W^l \cdot \text{CONCAT} \left(h_v^l, h_{N(v)}^{l+1} \right) \right), \tag{4}$$

where $h_{N(v)}^{l+1}$ can be extracted by a few different aggregate functions for the set of neighbour nodes N(v).

Graph Attention Network. The last considered method is based on the attention mechanism which also avoids transductive GCN constraints and apply the iterative aggregation procedure

$$h_i^{l+1} = \text{CONCAT}_{k=1}^K \sigma\left(\sum_{j \in N(i)} \alpha_{ij}^k W^k h_j^l\right).$$
(5)

The attention coefficient is computed as follows:

$$\alpha_{ij} = \frac{\exp\left(\sigma\left(a^T \cdot \text{CONCAT}(Wh_i^l, Wh_j^l)\right)\right)}{\sum_{k \in N(i)} \exp\left(\sigma\left(a^T \cdot \text{CONCAT}(Wh_i^l, Wh_k^l)\right)\right)},\tag{6}$$

8 V. Porvatov et al.

where a^T is a transposed vector of attention trainable parameters.

In order to boost the expressiveness of these methods and convert supervised setups to unsupervised, we propose to embed them as a part of the Deep Graph InfoMax pipeline [24]. This approach is based on minimizing of a two-component loss function

$$L = \frac{1}{N+M} \sum_{i=1}^{N} E_G \left[\log D(d_i, T) \right] + \sum_{j=1}^{M} E_C \left[\log \left(1 - D(\tilde{d}_j, T) \right) \right]$$
(7)

which aims to learn how to distinguish initial nodes representations d and corrupted ones \tilde{d} , Figure 3.



Fig. 3. Deep Graph Infomax corrupts feature vectors of the input graph G by function S (in the used realisation it shuffles features), constructs regular and corrupted node embeddings by applying $Z(\cdot)$, and finally estimates their similarity to the ground-truth vector T by the discriminator function D.

Once embeddings of routes z_{s_j} are computed, each vector can be extended by additional information about the weather conditions and corresponding temporal categorical features. After these manipulations with route vectors z_{s_j} they can be finally fed to the regression model.

5 Results

In order to perform the training and evaluation of proposed architectures, we need to split the datasets into three samples. We trained our model on the first 100 000 trips, while the test and validation steps were performed on equal parts of the remaining datasets.

Following the evaluation standards, we use a common set of metrics for the ETA prediction task: Mean Average Error (Eq. 8), Mean Average Percentage Error (Eq. 9), and Rooted Mean Square Error (Eq. 10).

MAE =
$$\frac{1}{N} \sum_{i=1}^{N} |y_i - y'_i|$$
, (8)

MAPE =
$$\frac{100}{N} \sum_{i=1}^{N} \left| \frac{y_i - y'_i}{y_i} \right|,$$
 (9)

RMSE =
$$\sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - y'_i)^2}$$
. (10)

5.1 Implementation details

Computational experiments were provided with the use of StellarGraph [5] library. All models were trained on 2 GPU Tesla V100, the total training time of the pipeline for the best models is 9 hours. During the embedding construction process, we used three types of each observed architecture with the number of layers from 1 to 3 and the fixed output of size 128. Neural networks weights were trained by Adam optimizer [9] due to its good convergence and stability. We use the static learning rate parameters $L_1 = 0.001$ for node embedding generation and $L_2 = 0.0001$ for regression.

5.2 Experiments

We performed series of computational experiments varying the strategy of subgraph embedding generation and the method of node representation extraction. As the final regression model, we leverage a multi-layer perceptron (MLP). For the purpose of Deep Graph InfoMax tests extension, we also compute the values of the metrics for regular unsupervised GraphSAGE and regression baseline to illustrate the general capabilities of different approaches. The final values of metrics for each configuration are shown in Table 4.

	Abakan			Omsk		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Baseline(MLP only)	111.05	316.39	27.129	145.819	296.86	25.019
GraphSAGE + VN	111.23	316.82	27.213	146.003	297.028	25.108
GraphSAGE + Sum	96.575	310.114	22.881	129.831	279.773	22.416
DGI(GCN) + Sum	97.927	310.628	23.506	141.017	289.32	24.335
DGI(GAT) + Sum	101.808	313.01	25.737	133.262	283.22	23.175
DGI(GS) + Sum	95.819	309.627	22.622	130.296	280.058	22.593

 Table 4. Evaluation results on test sample

9

10 V. Porvatov et al.

As it seen from the table, the best performance was achieved by the Graph-SAGE setup with Deep Graph InfoMax in the case of Abakan. Meanwhile, common GraphSAGE also demonstrates promising embeddings quality (especially for Omsk) which is slightly different from its DGI modification. The error distributions of the best models for each dataset are shown in Figure 4.



Fig. 4. Error distribution for the regression models trained on Abakan (a) and Omsk (b) datasets.

Unfortunately, the test series of virtual nodes route embeddings turned down our pursuit to report any significant results. We conclude that the expressiveness of this method is limited in the area of interest, despite previous positive attempts of implementation in other tasks [12]. However, such a result was partially foreordained by the studies which also explored subgraph embeddings [1].

6 Conclusion and Outlook

In this work, we implemented and explored a pipeline that includes state-ofthe-art algorithms of graph machine learning that emerged in recent years. We trained and tested our model on two consistent datasets which correspond to cities with different road topology types. Our results allow us to conclude that GraphSAGE-based models capture spatial patterns of city networks more substantially.

Our own perspectives include future development and modification of more specific methods based on obtained results. As the primary goal of this research was to find the most efficient methods of subgraph embedding construction in the context of ETA problem, we intend to use this knowledge to construct a more complex spatial approach in the upcoming papers. In the spotlight of our research, we also have an idea to design an powerful generalizing approach to various kinds of road networks with the potential of applying it to a bunch of cities.

Acknowledgements

The work was supported by the Joint Stock Company "Sberbank of Russia".

References

- Alsentzer, E., Finlayson, S.G., Li, M.M., Zitnik, M.: Subgraph neural networks. CoRR abs/2006.10538 (2020), https://arxiv.org/abs/2006.10538
- Asghari, M., Emrich, T., Demiryurek, U., Shahabi, C.: Probabilistic estimation of link travel times in dynamic road networks. In: Proceedings of the 23rd SIGSPA-TIAL International Conference on Advances in Geographic Information Systems. SIGSPATIAL '15, Association for Computing Machinery, New York, NY, USA (2015), https://doi.org/10.1145/2820783.2820836
- Ayhan, S., Costas, P., Samet, H.: Predicting estimated time of arrival for commercial flights. pp. 33–42 (07 2018)
- Dai, R., Xu, S., Gu, Q., Ji, C., Liu, K.: Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data. CoRR abs/2006.12715 (2020), https://arxiv.org/abs/2006.12715
- 5. Data61, C.: Stellargraph machine learning library. https://github.com/stellargraph/stellargraph (2018)
- Fout, A., Byrd, J., Shariat, B., Ben-Hur, A.: Protein interface prediction using graph convolutional networks. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6533–6542. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
- Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. Proceedings of the AAAI Conference on Artificial Intelligence 33, 922–929 (07 2019)
- Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs (2018)
- Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014), cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015
- 10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2017)
- 11. Kviesis, A., Zacepins, A., Komasilovs, V., Munizaga, M.: Bus arrival time prediction with limited data set using regression models. pp. 643–647 (01 2018)
- 12. Li, J., Cai, D., He, X.: Learning graph-level representation for drug discovery (2017)
- Li, Y., Fu, K., Wang, Z., Shahabi, C., Ye, J., Liu, Y.: Multi-task representation learning for travel time estimation. In: International Conference on Knowledge Discovery and Data Mining (KDD '18) (2018)
- Park, K., Sim, S., Bae, H.: Vessel estimated time of arrival prediction system based on a path-finding algorithm. Maritime Transport Research 2, 100012 (2021), https://www.sciencedirect.com/science/article/pii/S2666822X21000046

- 12 V. Porvatov et al.
- Paruchuri, V., Chellappan, S., Lenin, R.B.: Arrival time based traffic signal optimization for intelligent transportation systems. In: 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA). pp. 703-709 (2013)
- Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (Aug 2014), http://dx.doi.org/10.1145/2623330.2623732
- 17. Prokhorchenko, A., Panchenko, A., Parkhomenko, L., Nesterenko, G., Muzykin, M., Prokhorchenko, G., Kolisnyk, A.: Forecasting the estimated time of arrival for a cargo dispatch delivered by a freight train along a railway section. Eastern-European Journal of Enterprise Technologies 3, 30–38 (06 2019)
- Sanchez-Gonzalez, A., Heess, N., Springenberg, J.T., Merel, J., Riedmiller, M.A., Hadsell, R., Battaglia, P.W.: Graph networks as learnable physics engines for inference and control. CoRR abs/1806.01242 (2018), http://arxiv.org/abs/1806.01242
- Shi, C., Chen, B.Y., Li, Q.: Estimation of travel time distributions in urban road networks using low-frequency floating car data. ISPRS International Journal of Geo-Information 6, 253 (08 2017)
- Song, C., Lin, Y., Guo, S., Wan, H.: Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. Proceedings of the AAAI Conference on Artificial Intelligence 34, 914–921 (04 2020)
- 21. Sun, Y., Fu, K., Wang, Z., Zhang, C., Ye, J.: Road network metric learning for estimated time of arrival (2020)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. CoRR abs/1706.03762 (2017), http://arxiv.org/abs/1706.03762
- 23. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks (2018)
- 24. Veličković, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax (2018)
- Wang, D., Zhang, J., Cao, W., Li, J., Zheng, Y.: When will you arrive? estimating travel time based on deep neural networks. In: AAAI (2018)
- 26. Wang, Y., Zheng, Y., Xue, Y.: Travel time estimation of a path using sparse trajectories. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 25–34. KDD '14, Association for Computing Machinery, New York, NY, USA (2014), https://doi.org/10.1145/2623330.2623656
- 27. Wang, Z., Fu, K., Ye, J.: Learning to estimate the travel time. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 858–866. KDD '18, Association for Computing Machinery, New York, NY, USA (2018), https://doi.org/10.1145/3219819.3219900
- Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. CoRR abs/1801.07455 (2018), http://arxiv.org/abs/1801.07455
- Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. AI Open 1, 57–81 (2020)