Technology Growth Ranking Using Temporal Graph Representation Learning

Daniel Cummings
Ashrita Brahmaroutu
Marcel Nassar*
Nesreen K. Ahmed*
daniel.j.cummings@intel.com
Intel Corporation
USA

ABSTRACT

A key component of technology sector business strategy is understanding the mechanisms by which technologies are adopted and the rate of their growth over time. Furthermore, predicting how technologies grow in relation to each other informs business decision-making in terms of product definition, research and development, and marketing strategies. An important avenue for exploring technology trends is by looking at activity in the software community. Social networks for developers can provide useful technology trend insights and have an inherent temporal graph structure. We demonstrate an approach to technology growth ranking that adapts spatiotemporal graph neural networks to work with structured temporal relational graph data.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; *Social networks*; • **Mathematics of computing → Graph algorithms**; *Time series analysis.*

KEYWORDS

graph neural networks, social networks, spatiotemporal graphs, business intelligence

ACM Reference Format:
Daniel Cummings, Ashrita Brahmaroutu, Marcel Nassar, and Nesreen K. Ahmed. 2022. Technology Growth Ranking Using Temporal Graph Representation Learning. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3487553.3524255

1 INTRODUCTION

Technology trend analysis is a key component of business intelligence across all industry sectors. Many technologies follow the technology adoption life-cycle behavior which is modeled as a bell curve with the categories in order of innovators, early adopters, early majority, late majority, and laggards [19]. In addition to analyzing current and past trends, the capability to forecast at what pace different technologies are adopted or diffuse through various technology domains enables companies to strategically plan how they define future products, allocate resources, predict future market behavior, and invest in research and development (R&D)[9]. Across publicly traded firms listed in the NYSE, Amex, and NASDAQ, it was shown that R&D expenditures were about 400 billion USD with a median of 20% of total expenditures [4]. Interestingly, while companies are spending more on R&D than ever, the analysis showed a decrease in the R&D return-on-investment (ROI) in recent years. These insights highlight the importance of understanding how technologies evolve in time to enhance strategic business-level decision-making and R&D ROI.

We focus on the task of ranking future technology growth in the software domain. Software technologies can be associated with specific terms (e.g. Python, Java, C##), and the rate at which terms are mentioned or discussed in social networks can indicate adoption and popularity. Social communities contain useful relational data through their neighborhood structure: related topics are connected by an edge with a label that represents the underlying relationship. There are a variety of ways to define the graph structure for these sources such as from a user-to-user, user-to-item interaction perspective, or as a citation network [7]. In this work, we define a graph as a set of technology terms (nodes) connected by edges that are weighted by term co-occurrence counts (e.g. how often terms are mentioned together). The temporal feature of the graph nodes is the term count over time. We use the Stack Overflow[1] (SO) question and answer platform as the dataset source because it is easily accessible, is a reliable indicator of developer community activity, and represents a broad range of software motifs that are pervasive across various technology sectors.

2 METHODOLOGY

For this work, we demonstrate an approach for forecasting technology growth by adapting spatiotemporal graph neural networks (GNNs) to a structured temporal graph as illustrated in Figure 1. Specifically, we are interested in how technologies rank in growth with respect to each other across a 1-year forecast horizon. A very popular application of GNNs in the spatiotemporal domain centers around the task of traffic flow prediction [8]. However, the spatial

*Both authors contributed equally to this research.

[1]https://stackoverflow.com

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524255

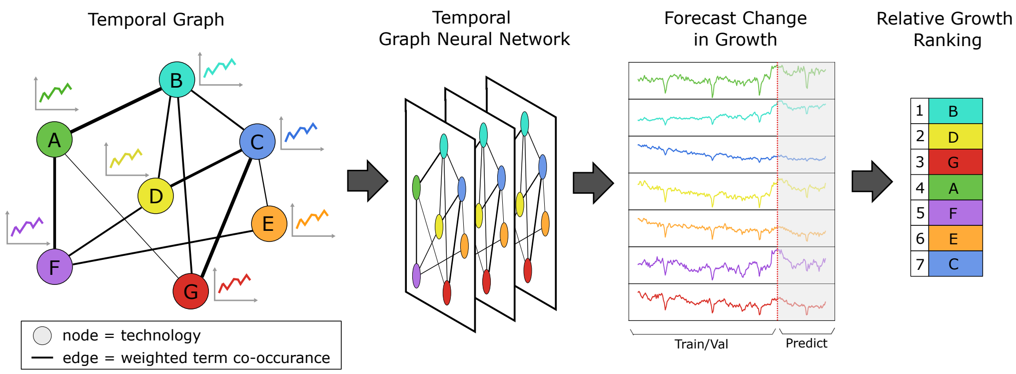

Figure 1: Technology trend modeling flow using temporal GNNs.

proximity component can easily be replaced with any measure of relationship where we choose to use the technology term co-occurrence as the weighted edge mechanism. The graph features for each node are the Stack Overflow post tag counts (e.g. time series observations) where each time step represents the aggregated values for the week. Since we are training across nodes with varying observation scales, we apply z-score normalization as $z = (x - \mu)/\sigma$ where x the sample, μ the mean of the training samples, and σ the standard deviation of the training samples. For edge weighting, we note that each time a user posts a question, a list of relevant technology terms are tagged for that question. The tag list provides a basis for creating weighted edges between each term. To create the graph edge connections and edge weightings, we normalize the co-occurrence against other connected terms and average the shared co-occurrence edges.

2.1 Models

The technology term growth ranking is calculated from the time series forecast produced by the temporal GNN. Time series forecasting involves fitting a model to predict future values up to a forecast horizon based on past values (lags) in a series. In our approach, we represent a set of technology domain terms as a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$, where $\mathcal{V}$ is a set of nodes (e.g., terms), $\mathcal{E}$ is a set of edges, and $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is a weighted adjacency matrix representing the node-to-node affinity (e.g., normalized term co-occurrence with values from 0 to 1). We denote the technology term occurrence observed on $\mathcal{G}$ as a graph signal $\boldsymbol{X} \in \mathbb{R}^{N \times P}$, where P is the number of node attribute features (e.g. length of historical time series). Given that $\boldsymbol{X}_t$ represents the graph signal observed at time t, the technology term forecasting problem aims to learn a function f that maps historical graph signals to future T graph signals:

$$[\boldsymbol{X}_{(t+1)}, ..., \boldsymbol{X}_{(t+T)}] = f(\mathcal{G}; [\boldsymbol{X}_{(t-n)}, ..., \boldsymbol{X}_{(t-1)}, \boldsymbol{X}_{(t)}]) \quad (1)$$

A spatiotemporal GNN model combines temporal deep learning techniques such as recurrent neural networks (RNNs) with graph representation learning. The graph representation learning component can be generalized by the message passing formalism: node and edge attributes are parameterized functions that generate compressed representations which propagate between nodes based on a message-passing and aggregation strategy to form new representations at each layer. Popular examples of this include Graph Convolutional Network (GCN) [10] and ChebyConv [5]. The temporal learning component can be added by incorporating recurrent deep learning approaches such as Long Short-Term Memory (LSTM) [6] and Gated Recurrent Unit (GRU) [3] networks. Since the graph has a temporal sequence associated with each node, message passing at each time point allows the fused temporal and graph learning layers to train jointly. This generalized spatiotemporal framework allows for different combinations of temporal and GNN layers to be applied. For additional background on the spatiotemporal algorithms and GNN theory, we refer to the prior mentioned model references and work by [21].

To perform our forecasting and ranking task, we choose to evaluate our structured graphs using the following spatiotemporal GNN models as shown in Table 1: Graph Convolutional Recurrent Network (GConvGRU, GConvLSTM) [16], Diffusion Convolutional Recurrent Neural Network (DCRNN) [11], and Temporal Graph Convolutional Network (T-GCN) [20]. While not comprehensive of all possibilities, these models represent well-known spatiotemporal modeling approaches and can be leveraged from the *PyTorch Geometric Temporal*[2] library [14]. For parameter tuning we search over the learning rate (0.01-0.001), graph filter support K (1-4 graph hops) [16], and node feature input lags. Additionally, we use the Adam optimizer and find that L1 loss gives a better model performance than L2.

Table 1: Spatiotemporal GNN Model Comparison.

Model	Temporal Layer	GNN Layer
GConvGRU [16]	GRU	Chebyshev
GConvLSTM [16]	LSTM	Chebyshev
DCRNN [11]	GRU	DiffConv
T-GCN [20]	GRU	GCN

For non-graphical model comparisons, we evaluate several well-known models used for business intelligence applications. The first is Prophet, an additive regression model that accounts for weekly and yearly seasonal components [17]. The next is DeepAR, a popular model that uses an autoregressive recurrent neural network approach and is often used as a benchmark [15]. Additionally, we

[2]https://pytorch-geometric-temporal.readthedocs.io/

run N-BEATS [13] which uses a deep stack of fully-connected layers and has emerged as a state-of-the-art benchmark given its performance on the M4 competition dataset [12]. Finally, we use a baseline that always predicts the last known value before the forecast.

To measure the performance of the models in consideration, we looked at two aspects of the problem statement, the time series forecasting and growth ranking metrics. To evaluate the time series forecasting performance between the models, we use the symmetric mean absolute percentage error (sMAPE) given its prevalence in benchmarks (e.g. M3, M4) and interpretability as a relative error metric [1]. SMAPE is defined as

$$\text{sMAPE} = \frac{100}{T} \sum_{t=1}^{T} \frac{|F_t - Y_t|}{(|Y_t| + |F_t|)/2} \tag{2}$$

where T is the number of samples within forecast horizon, F_t is the prediction, Y_t is the target.

Most time series evaluation approaches focus on the predicted versus ground truth performance for each individual series. However, we highlight that from a business intelligence perspective, it can be more important to understand how the trends evolve with respect to each other. If a model gives a good error score on the forecast but fails in the growth ranking of each series with respect to each other, then some key insights are lost. In other words, unique combinations of over-forecasting and under-forecasting between the multiple time series under consideration can lead to a good aggregated sMAPE score but a poor ranking performance. Given that we're interested in the trends of all nodes in the graph relative to each other, we create a ground truth *ranking* of the technology terms at each time step in the forecast by ranking the difference of the last known value of each term's time series y_t and the value at $\{y_{t+1}, ..., y_{t+T}\}$ in order of highest growth to lowest. We similarly create a predicted ranking and end up with two ordinal lists for each time step. The Kendall rank correlation coefficient (τ_B) is used to measure the degree of similarity between the two rankings.

2.2 Datasets

We demonstrate the use of the structured approach on four datasets extracted from Stack Overflow as shown in Table 2. *SO-Top50* represents the top Stack Overflow tags, *SO-Prg* contains the terms for the top ninety programming language terms, and *SO-Bigdata* and *SO-ML* contain popular terms from big data processing platforms and machine learning libraries respectively.

Table 2: Stack Overflow dataset descriptions.

Topic Graph	$\mathcal{V}$	$\mathcal{E}$	Example Terms
SO-Top50	50	1225	c#, c++, java, python, r
SO-Prg	90	4005	reactjs, javascript, sql
SO-Bigdata	40	780	apache-spark, scala, sql
SO-ML	30	435	pandas, numpy, pytorch

For the temporal GNN model training and validation, we use five years of historic data (June 2015 to June 2020) and reserve the past year (June 2020 to June 2021) for the forecast evaluation. Each time step represents the term count observations by week. A portion of the newer technology terms did not have any observations earlier than 2015 hence the 5-year window training limit. We train on 260-weeks (5 years) of observations and predict on a 52-week (1 year) multi-step forecast horizon. While certain high-level business R&D strategies can be formulated on a long-term basis (e.g. 2-5 years), specific research allocation decisions change year-to-year due to the rapidly evolving technology landscape. A year forecast (with weekly granularity) can lend important insight during the yearly strategic planning cycle.

3 RESULTS

The time series forecasting component of the results for our datasets is shown in Table 3. The averaged sMAPE is evaluated over the 52-week multi-step forecast horizon and the results are the mean of 10 trials for each model. The *last known* baseline represents the last known observation count for each technology term prior to forecasting. We find that the temporal GNN models perform well across the datasets but note that sMAPE does not necessarily indicate a good ranking as shown later.

Table 3: Time series forecasting results (sMAPE) for a 52-week horizon.

Model	SO-Top50	SO-Prg	SO-Bigdata	SO-ML
Last Known	74.7	107.0	130.7	103.2
Prophet	19.0	32.3	51.0	33.7
DeepAR	24.9±0.8	23.9±0.9	40.2±1.3	22.8±0.7
N-BEATS	15.1±0.4	21.3±0.5	37.7±0.9	23.2±0.5
GConvGRU	13.8±0.1	24.8±0.1	35.9±0.2	21.8±0.1
GConvLSTM	18.9±0.1	20.1±0.2	35.8±0.2	24.8±0.2
DCRNN	16.0±0.1	24.1±0.1	37.3±0.2	20.6±0.1
TGCN	20.1±0.2	31.6±0.2	39.8±0.3	26.7±0.2

The technology growth ranking performance, as given by the rank correlation coefficient, is given in Table 4. For each model, we average the coefficients over the forecast horizon of 52-weeks. The GConvGRU and GConvLSTM models perform consistently well in rank correlation. To offer a more detailed view, Figure 2 shows the rank correlation using the *SO-Bigdata* graph illustrating that the GConvGRU model generally outperforms the others for each week in the forecast period.

Table 4: Kendall (τ_B) rank correlation results averaged over a 52-week forecast period by dataset.

Model	SO-Top50 (τ_B)	SO-Prg (τ_B)	SO-Bigdata (τ_B)	SO-ML (τ_B)
Last Known	0.052	-0.041	-0.154	-0.025
Prophet	0.011	0.010	0.061	-0.0220
DeepAR	0.057	0.427	0.244	0.457
N-Beats	0.354	0.202	0.143	0.360
GConvGRU	**0.484**	0.463	0.486	0.459
GConvLSTM	0.345	**0.491**	**0.498**	**0.544**
DCRNN	0.184	0.321	0.412	0.458
T-GCN	0.437	0.480	0.356	0.515

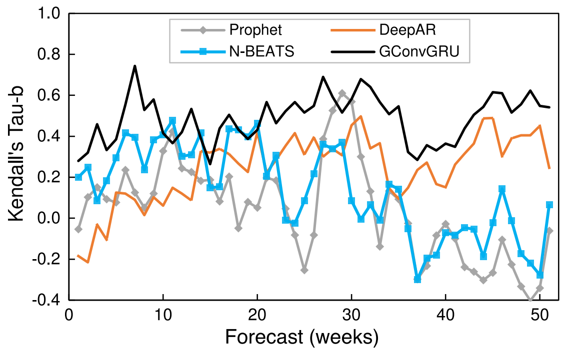

Figure 2: Rank correlation performance on the SO-Bigdata graph 52-week forecast task.

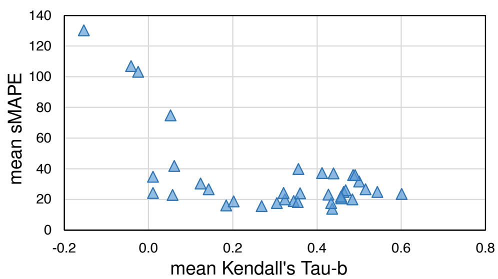

Figure 3: Comparison between mean sMAPE and Kendall's τ_b for the evaluated datasets and models that shows sMAPE is not always a reliable indicator for forecast ranking.

When comparing the sMAPE scores to the rank correlation coefficient results, Figure 3 highlights that the sMAPE metric is not necessarily a good indicator of forecast ranking. Across the datasets and models, most results land in an sMAPE window between 20-40 do not show a clear correlation between sMAPE and ranking. We attribute this primarily to the aforementioned over-forecasting and under-forecasting characteristics between each of the predicted time series in the datasets.

4 CONCLUSION

We have shown a structured approach for forecasting and ranking technology growth by adapting spatiotemporal graph neural networks to work with temporal social networks. Moreover, we have highlighted the importance of accounting for the relative growth ranking during model evaluation in the context of technology trend analysis. Future work will address dataset sparsity issues for lesser-known technologies and include a broader range of social network sources to enable heterogeneous graphs. Furthermore, the forecasting model comparison will be expanded to include a wider variety of algorithms including novel approaches that use spectral mechanisms such as StemGNN [2] and those with self-attention mechanisms. Finally, continuing to bridge the gap between graph deep learning and efforts in advanced econometric fields [18] could lead to new innovative approaches that enhance forecasting accuracy and insights in the business intelligence domain.

REFERENCES

[1] A. Botchkarev. 2018. Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology. *ArXiv* abs/1809.03006 (2018).

[2] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Conguri Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. 2021. Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting. arXiv:2103.07719 [cs.LG]

[3] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:1406.1078 [cs.CL]

[4] Asher Curtis, Sarah McVay, and Sara Toynbee. 2020. The changing implications of research and development expenditures for future profitability. *Review of Accounting Studies* 25, 2 (2020), 405–437. https://EconPapers.repec.org/RePEc:spr:reaccs:v:25:y:2020:i:2:d:10.1007_s11142-019-09528-6

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2017. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. arXiv:1606.09375 [cs.LG]

[6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[7] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2021. Open Graph Benchmark: Datasets for Machine Learning on Graphs. arXiv:2005.00687 [cs.LG]

[8] Weiwei Jiang and Jiayun Luo. 2021. Graph Neural Network for Traffic Forecasting: A Survey. arXiv:2101.11174 [cs.LG]

[9] Hyunji Kim, Sun-Young Park, and Won-IL Joh. 2019. A Study on Technology Development Performance and Technology Commercialization Performance According to the Technology Development Capability of SMEs Focusing on a Comparative Analysis of Technology Business Groups. *Journal of Open Innovation: Technology, Market, and Complexity* 5, 3 (2019). https://doi.org/10.3390/joitmc5030065

[10] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs.LG]

[11] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. arXiv:1707.01926 [cs.LG]

[12] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2020. The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting* 36, 1 (2020), 54–74. https://doi.org/10.1016/j.ijforecast.2019.04.014 M4 Competition.

[13] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2020. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *ArXiv* abs/1905.10437 (2020).

[14] Benedek Rozemberczki, Paul Scherer, Yixuan He, G. Panagopoulos, Maria Sinziana Astefanoaei, Olivér Kiss, Ferenc Béres, Nicolas Collignon, and Rik Sarkar. 2021. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021).

[15] David Salinas, Valentin Flunkert, and Jan Gasthaus. 2019. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. arXiv:1704.04110 [cs.AI]

[16] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2016. Structured Sequence Modeling with Graph Convolutional Recurrent Networks. arXiv:1612.07659 [stat.ML]

[17] Sean J. Taylor and Benjamin Letham. 2018. Forecasting at Scale. *The American Statistician* 72, 1 (2018), 37–45. https://doi.org/10.1080/00031305.2017.1380080

[18] Yoshimasa Uematsu and Shinya Tanaka. 2019. High-dimensional macroeconomic forecasting and variable selection via penalized regression. *The Econometrics Journal* 22, 1 (02 2019), 34–56. https://doi.org/10.1111/ectj.12117 arXiv:https://academic.oup.com/ectj/article-pdf/22/1/34/30795022/ectj12117.pdf

[19] Thomas W. Valente and Everett M. Rogers. 1995. The Origins and Development of the Diffusion of Innovations Paradigm as an Example of Scientific Growth. *Science Communication* 16, 3 (1995), 242–273. https://doi.org/10.1177/1075547095016003002 PMID: 12319357.

[20] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2020. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems* 21, 9 (Sep 2020), 3848–3858. https://doi.org/10.1109/tits.2019.2935152

[21] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001