information

MDPI

Article

Link Prediction in Time Varying Social Networks

Vincenza Carchiolo *,† , Christian Cavallo †, Marco Grassia †, Michele Malgeri † and Giuseppe Mangioni †

DIEEI-Dipartimento di Ingegneria Elettrica, Elettronica Informatica, Universitá degli Studi di Catania, I95125 Catania, Italy; chrychry96@hotmail.it (C.C.); marco.grassia@unict.it (M.G.); michele.malgeri@unict.it (M.M.); giuseppe.mangioni@unict.it (G.M.)

* Correspondence: vincenza.carchiolo@unict.it

† These authors contributed equally to this work.

Abstract: Predicting new links in complex networks can have a large societal impact. In fact, many complex systems can be modeled through networks, and the meaning of the links depend on the system itself. For instance, in social networks, where the nodes are users, links represent relationships (such as acquaintance, friendship, etc.), whereas in information spreading networks, nodes are users and content and links represent interactions, diffusion, etc. However, while many approaches involve machine learning-based algorithms, just the most recent ones account for the topology of the network, e.g., geometric deep learning techniques to learn on graphs, and most of them do not account for the temporal dynamics in the network but train on snapshots of the system at a given time. In this paper, we aim to explore Temporal Graph Networks (TGN), a Graph Representation Learning-based approach that natively supports dynamic graphs and assigns to each event (link) a timestamp. In particular, we investigate how the TGN behaves when trained under different temporal granularity or with various event aggregation techniques when learning the inductive and transductive link prediction problem on real social networks such as Twitter, Wikipedia, Yelp, and Reddit. We find that initial setup affects the temporal granularity of the data, but the impact depends on the specific social network. For instance, we note that the train batch size has a strong impact on Twitter, Wikipedia, and Yelp, while it does not matter on Reddit.

Keywords: deep learning; geometric deep learning; information spreading; link prediction; social networks

check for updates

Citation: Carchiolo, V.; Cavallo, C.; Grassia, M.; Malgeri, M.; Mangioni, G. Link Prediction in Time Varying Social Networks. *Information* **2022**, *13*, 123. https://doi.org/10.3390/info13030123

Academic Editor: Gennady Agre

Received: 5 January 2022
Accepted: 26 February 2022
Published: 1 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

In the last decades, the quantity of information about the environment that surrounds us is growing exponentially and has assumed more and more importance in our life, mainly due to the social networks that are the main actors in their production and dissemination. This extremely large availability of data has created new and exciting challenges in data and information analysis, for example, to analyze the dissemination process and the reliability of information as well as their impact on society.

The link prediction problem consists of studying the possible future connections among nodes in a complex network, aiming at explaining the evolutionary dynamics of network [1]. A common approach is to evaluate the similarity between the nodes belonging to a network at time t in order to know the topology of the future network at time $t + 1$. The key concept is the similarity measure, which can depend on both the semantic [2] and the topology of the network [3]. Link prediction has been applied to a variety of domains including highly dynamic systems, such as social networks [4], for instance, to predict people we may know but we are not still connected to [5] or products we could be interested in, in an electronic commerce system [6]. The simultaneous development of machine learning techniques, such as deep learning, allows researchers to analyze more and more features of social networks adopting several, different approaches, for instance, McAuley and Leskovec proposed to discover users' *social circles* (e.g., lists on

Facebook and Twitter) using machine learning [7]. However, since in many cases the existing topological correlation plays an important role in defining the social relations, Bronstein et al., in [8], report that, while deep learning has been successfully applied on a wide variety of tasks [9], from speech recognition [10,11] and machine translation [12] to image analysis and computer vision [13–15], there is now a growing interest in trying to learn on non-Euclidean data. They list several open issues, such as *generalization, support of directed graph, data synthesis, computation, and time-varying domains*, the latter being the focus of this paper. Dynamic complex non-Euclidean graphs have been utilized to infer social relations from trajectory data [16], to perform semi-supervised community detection [17], to detect social spammers [18], or to protein–protein interaction networks [19] and knowledge graphs [20]; however, few approaches have been developed so far for learning on time varying graphs, as most only support static graphs [3,8,21]. Moreover, the authors of [22] highlighted that, in some cases, the lack of temporal analysis does not lead to an optimal solution since many behaviors are closely related to the temporal evolution of the system. In general, it should be noted that dynamic graph learning is a hard task due to the time-varying nature of the graph structure itself, as nodes and links are constantly evolving (i.e., new nodes or links may appear, and others may disappear). Consequently, the computation of node embeddings requires to preserve the structural information of the graph but efficiently capture *temporal causality*. To model the flow of time, we can use a discrete-time representation using a sequence of (static) snapshots, i.e., we represent the graph at time t, $t + 1$, ..., $t + n$ and study their correlations; however, a continuous representation is more general. Usually, a continuous-time representation is just a list of events with a timestamp. Let us note that latter representation is able to capture the event when it happens and avoid the problem of defining the correct dimension of unit. Recently, new geometric deep learning techniques to capture the time evolution of networks have been developed. Two of the most promising approaches are Temporal Graph Networks (TGN) [23] by Rossi et al. and the Temporal Graph Attention (TGAT) [22] layer by Xu et al.

The main issue in using non-Euclidean data in machine learning, such as graphs, is to describe and encode their structure so that it can be easily exploited by the models. Geometric deep learning is an emerging umbrella term for techniques that generalize deep learning to non-Euclidean domains, such as certain graphs and manifolds [24,25], and has been successfully used in several applications such as graph-wise classification, signal processing [8], and vertex-wise classification and regression [26]. The most common approach to learn on such data is Graph Neural Networks (GNN) that typically work with a *message passing* mechanism, aggregating information from the neighborhood of a node and creating representations (called *embeddings*) that are later used for node/graph classification or link prediction.

Temporal Graph Attention (TGAT) layers [22] are an extension of Graph Attention Networks [27] meant to handle continuous time-dynamics graphs. Just as other Graph Neural Network (GNN) layers, the TGAT aggregate the local neighborhood representations (or features) to produce the node's embedding, with the main difference being that they also account for temporal information (i.e., the timestamps) by also learning a time embedding. Like GAT, TGAT are inductive, i.e., they can generalize to new nodes and graphs, and use the self-attention mechanism, proposed by Vaswani et al. [28], to capture the temporal interactions with both node and topological features. Moreover, they also support the multi-head setting and, unlike the original GAT, edge features.

While geometric deep learning and its application on social networks are rather new topics, extensive literature already exists. For instance, in [29], the authors present a survey of the uses of geometric deep learning in Facebook post analysis in order to study the way to preserve the social network integrity, while in [30], the authors present a geometric deep learning-based approach to automatic detection of fake news based on analysis of existing content instead of using Natural Languages Processing techniques that often are not able to manage the context. Another example is to study geometric properties of the propagation scheme, whereby the authors of [31–34] show that fake and real news spread differently on social media, and Suno et al., in [35], focus on realistic propagation mechanism and highlight the different propagation evolution of fake news and real news tracking large real databases. In fact, online users tend to acquire information adhering to their worldviews [36], ignoring dissenting information [37] that increases the probability of the spreading of misinformation [38], which may cause fake news and inaccurate information to spread faster and wider than fact-based news [39]. In [40], the authors study a Twitter's retweet network using geometric deep learning. Leng at al. [41] present an example of using geometric deep learning for recommendation tasks that uses a Multi-Graph Graph Attention Network (MG-GAT) on two large-scale datasets collected from some standard datasets in Recommender systems such as Yelp. MG-GAT learn the latent representations of users and businesses by aggregating different sets of information from each user's neighbors (business) on a chart of close importance. However, the previous analysis does not take into account the flow of time that is one of the most important characteristics to be studied in a social network.

In this paper, we analyze the performance of the Temporal Graph Network (TGN) model [42] applied to some datasets extracted from widely used social networks.

We compared the impact of initial batch size on the selected datasets, arguing about its correlation with the peculiarities of each social network, if any. The model has been trained and tested on Twitter, Wikipedia, Yelp, and Reddit, and our experiments highlight that temporality must be considered to obtain a significant increase in knowledge. The contribution of this work is threefold. Specifically, we argue about the creation of appropriate social network datasets for temporal analysis, then we employ the TGN-based model, and finally we present the results of the evaluation procedure that uses these datasets to assess the geometric deep learning algorithm mentioned.

Section 2 sketches the Temporal Graph Networks (TGN). Section 3 presents the datasets used and their extraction, while Section 4 presents the simulation setup and the results. Section 5 outlines some conclusions.

2. Temporal Graph Networks

The Temporal Graph Networks [23] (TGN) is a generic framework for deep learning on dynamic graphs represented as sequences of timed events, which, according to the experimental results reported by the authors, outperforms the state-of-the-art approaches while being more efficient.

The TGN framework is modular, and each module can be modified depending on the application scenario. More in detail, the main module is the node memory that stores a representation of the current state of each node, for instance, the node's history and long-term dependencies. Each event is first transformed into a message for each node involved using a "message function", and then each message is used to update the memory entry of that node. It should be noted that, for computational efficiency reasons, during the training, events are divided in batches, and multiple messages involving a node are aggregated using some function.

More specifically, TGNs aggregate all the messages $\mathbf{m}_i(t)$ that involve the same node i in a given batch using an aggregation mechanism, defined as in the following equation:

$$\bar{\mathbf{m}}_i(t) = \text{agg}(\mathbf{m}_i(t_1), ..., \mathbf{m}_i(t_b)) \tag{1}$$

where $t_1, ..., t_b$ are time events, with $t_b \leq t$.

Moreover, while authors use an identity message function and non-learnable message aggregation ones, replacing these functions with learnable ones is left as future work. We will explore such an option in Section 4.1, where we will present a set of different aggregation functions evaluating their performance.

Regarding the node embeddings, a GNN model employing slightly modified TGAT layers is used to compute them at each timestamp. Specifically, the main difference is that the input node features also include the node's state.

It is also worth mentioning that TGN supports either node-wise events (i.e., they involve a single node) or interactions between two nodes and that the creation and deletion of nodes and links are also supported.

3. Dataset

We use four datasets in our experiments, specifically, we use the same Reddit and Wikipedia [43] networks used in the TGN paper [23], and we also build two additional datasets from Yelp and from Twitter. In the following, we provide further details about the four networks and about their edge features.

The Reddit and Wikipedia networks are bipartite: the nodes of the first mode correspond to users, while the second mode to sub-reddits and to pages, respectively. In the Reddit network, each link represents a new post, whereas in the Wikipedia network links represent page edits. In both networks, links have a timestamp and 172 features extracted with Linguistic Inquiry and Word Count (LIWC) [44].

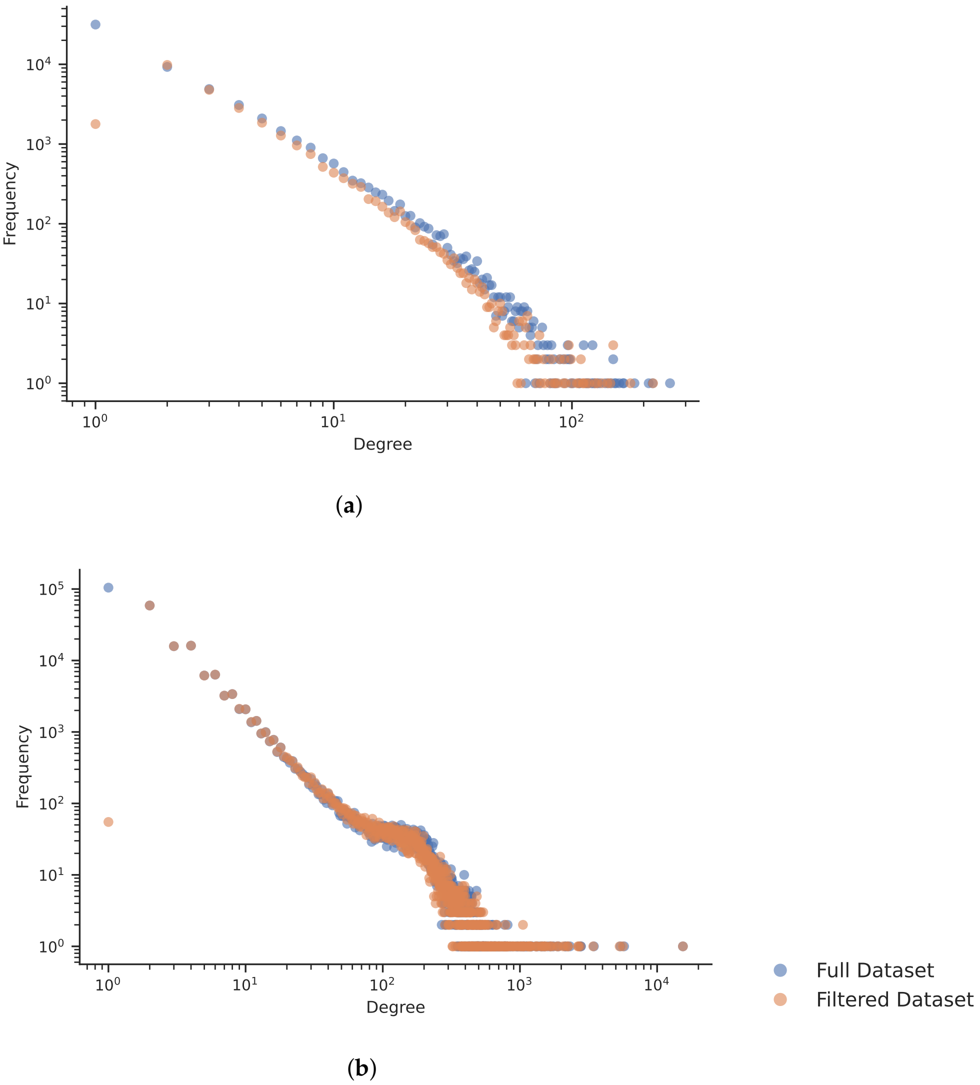
The Yelp network is a bipartite network where the modes represent users and businesses, and links represent reviews. The network was built from the raw data published by Yelp in December 2020. In particular, we only keep the reviews written since 2020 of the businesses in the most active country, in order to build a dense network. Note that we filter out nodes with degree equal or lower than one. The filtered and the full degree distributions of the network resemble a power-law and are shown in Figure 1a. Each link has a timestamp, and we extract 20 features using SEANCE [45] and also include the number of words in the review.

The Twitter network was built by collecting tweets of users that interacted with the first 100 tweets that Elon Musk made since 15 May 2020; he was chosen because of the large number of interactions with his tweets. Specifically, for each tweet, we selected 100 retweets by verified users only, and for each retweet, we collect a random number (between 100 and 1000) of their tweets in order to obtain a limited number of events per user. We stop the collection process after reaching one million tweets. Regarding the network, it was built from the collected tweets that were made between 1 and 15 May 2015. In the resulting monoplex network, nodes are users and links are tweets, each with its own timestamp. We extract 768 features for each link using BERT (Base, Multilingual Cased) [46], which was chosen to keep the edge features comparable with the ones used in the original TGN paper. Again, we filter out nodes with degree equal or lower than one. The degree distribution of the network resembles a power-law and is shown in Figure 1b.

The statistics of the networks are reported in Table 1.

Table 1. Dataset.

				Edge Features		
Network	**Type**	$\lvert N\rvert$	$\lvert E\rvert$	**#**	**Type**	**Timespan**
Reddit	Bipartite (users and posts)	11.000	672.447	172	LIWC	30 days
Twitter	Monoplex (users)	38.458	715.030	768	BERT	15 days
Wikipedia	Bipartite (users and pages)	9.227	157.474	172	LIWC	30 days
Yelp	Bipartite (users and reviews)	29.608	61.522	21	SEANCE	1 year



(**a**)

(**b**)

Figure 1. Degree distribution of the networks built for this study, before and after filtering. (**a**) Yelp. (**b**) Twitter.

4. Simulations and Results

In this section, we present the results of various experiments aiming to predict links among nodes (new interactions) on the base of the previous (i.e., occurred temporally before) events. More specifically, we explore the role of the aggregation function and training batch size in shaping the TGN's performance.

The final goal of these experiments is to stress the role of the hyperparameters in the TGN with the aim of helping other researchers to correctly choose TGN's parameter during the training phase.

We run the experiments on top of the official implementation of TGNs [23].

4.1. Experiments Settings

As mentioned in Section 2, TGNs can support different aggregation functions. Here, we define two new aggregation mechanisms and compare them against the two proposed in the TGN paper on the four networks discussed in Section 3. In particular, we compare the last message aggregator (that only keeps the most recent message) and the mean aggregator (that performs the average of the messages)—both proposed in the TGN paper [23] and defined as in Equations (2) and (3)—plus two weighted mean aggregators proposed by us as in Equation (4), which respectively assign an importance score to messages that depend on their relative timestamp (with respect to the first message in the batch) and learn the message importance.

$$\text{agg}_{\text{last}}(\mathbf{m}(t_1), ..., \mathbf{m}(t_b)) = \mathbf{m}(t_b) \tag{2}$$

where $t_b = \max\{t_j\}$ is the greatest timestamp in the batch.

$$\text{agg}_{\text{mean}}(\mathbf{m}(t_1), ..., \mathbf{m}(t_b)) = \frac{1}{B}\sum_{j}^{B} \mathbf{m}(t_j) \tag{3}$$

where B is the number of messages in the batch.

More in detail, we define a non-trainable importance function that assigns a weight to each message in the batch as the difference between its own timestamp and the one of the oldest message in the batch (Equation (5)), and a trainable importance function, a multi-layer perceptron (MLP) that produces a scalar weight for each message in the batch independently (Equation (6)). Both the proposed aggregators then scale the weight of each message in the batch using the softmax function, so that each weight is in $[0, 1]$, and perform a weighted sum.

$$\text{agg}_{\text{weighted mean}}(\mathbf{m}(t_1), ..., \mathbf{m}(t_b)) = \sum_{j}^{B} w_j \cdot \mathbf{m}(t_j) \tag{4}$$

where B is the number of messages in the batch, and

$$w_j = [\text{softmax}(f(\mathbf{m}(t_1), ..., \mathbf{m}(t_b)))]_j$$

$$f_{\text{relative timestamp}}(\mathbf{m}_i(t_1), ..., \mathbf{m}_i(t_b)) = (t_1 - t_{min}, ..., t_b - t_{min}) \tag{5}$$

and $t_{min} = min_b\{t_j\}$

$$f_{\text{MLP}}(\mathbf{m}(t_1), ..., \mathbf{m}(t_b)) = (\text{MLP}(\mathbf{m}(t_1)), ..., \text{MLP}(\mathbf{m}(t_b))) \tag{6}$$

where MLP is a Multi-Layer Perceptron with a scalar output for each message.

4.2. Results

In this subsection, we report the link prediction results by using different aggregation functions, as introduced in Section 4.1, and different batch sizes.

4.2.1. Experiments on Aggregation Functions

Here, we present experimental results of TGN's in performing link prediction by using different aggregation functions, as introduced in Section 4.1. Our results suggest that the choice of the aggregator does not always affect the performance significantly, as reported in Tables 2 and 3 for the inductive and transductive link prediction task. In fact, the four aggregators achieve similar performance in all but in the Reddit network, where the trainable aggregator outperforms the others and where the last message aggregator scores 10% lower average precision than the other aggregators. In particular, such behavior can be observed in both the inductive and in the transductive settings. This also suggests that, in the Reddit dataset, all the events play a significant role and poor performance with the last message aggregator is mainly due to the loss of some important messages. On the other hand, while the mean aggregator provides the best results in the Twitter, Yelp, and Wikipedia datasets in the inductive settings, it never outperforms the others in the transductive one, as the last message, the weighted mean, and the trainable aggregators outscore the other aggregators, respectively, in the Twitter, Yelp, and Wikipedia networks.

Table 2. Inductive average precision results using different aggregators.

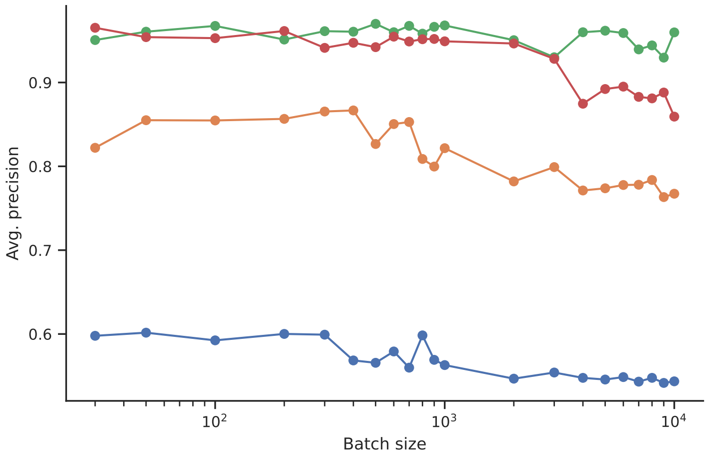
Network	Last m.	Mean	Weighted Mean	MLP
Reddit	86.76	96.62	95.56	**97.45**
Twitter	87.16	**87.32**	85.69	86.14
Yelp	63.43	**63.87**	63.51	62.83
Wikipedia	96.05	**96.21**	95.88	96.02

Table 3. Transductive average precision results using different aggregators.

Network	Last m.	Mean	Weighted Mean	MLP
Reddit	85.34	96.78	95.14	**97.58**
Twitter	**95.62**	95.19	94.70	94.52
Yelp	75.34	75.47	**75.82**	74.35
Wikipedia	96.80	96.78	96.89	**96.98**

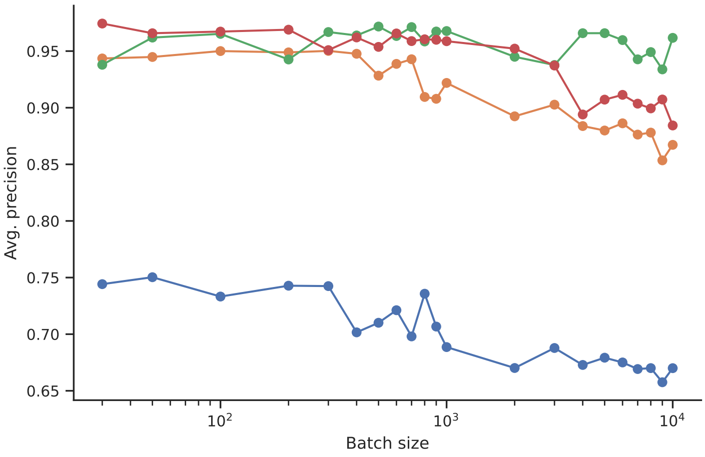
4.2.2. Experiments on Batch Size

The batch size, as mentioned by TGN authors [23], affects efficiency significantly. In fact, in many datasets—where many consecutive events involve the same nodes—not only a larger batch size translates into a smaller number of model's weights updates but also into a larger number of aggregated events. In other words, the batch size may affect the temporal granularity of the data. During the choice of the training hyperparameters, one may wonder how much the final performance is hit by the different batch sizes and running an extensive grid search that covers all the parameters (or a wide range of values) is not always feasible due to computational constraints. Here, we try to answer this question by showing the ROC AUC values in the inductive (Figure 2a) and in the transductive (Figure 2b) link prediction tasks as a function of the batch size. The number of aggregated messages, using the mean aggregator used in the TGN paper, is also reported in Figure 2c. In particular, the performance on the Twitter, Wikipedia, and Yelp networks is heavily affected by the increased batch size, as the ROC AUC drops by 0.1 when the batch size is 10^4 with regard to the peak value, while this is not the case with the Reddit one, even if the number of aggregated messages increases similarly. This suggests that, in the Reddit dataset, the temporal order of the events does not play an important role as in the other datasets. Note that, due to computational constraints, it was not possible to test with batch sizes lower than 30.

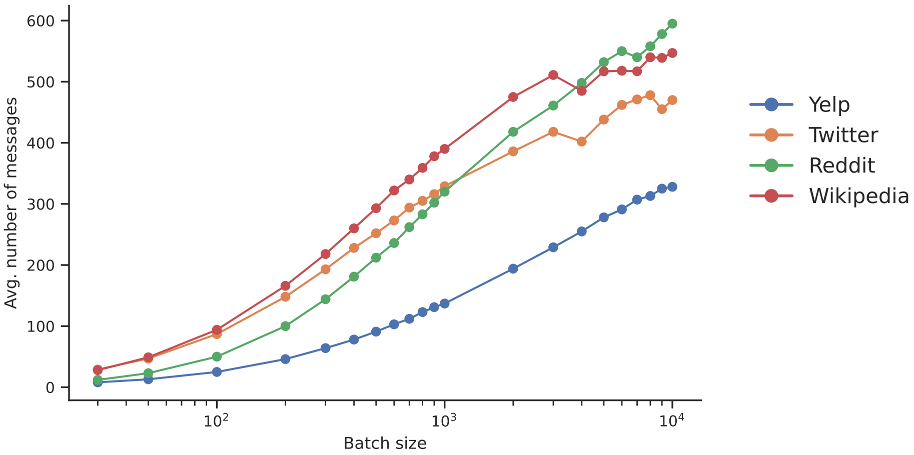


(**a**)

Figure 2. *Cont.*



(**b**)



(**c**)

Figure 2. Batch size experiments. Average precision results in two different settings and average number of aggregations as a function of the batch size. (**a**) **Inductive setting.** (**b**) **Transductive setting.** (**c**) **Average number of aggregations.**

5. Conclusions and Future Work

The paper discusses the application of a recent deep learning approach to the problem of link prediction in complex networks and the role of some of its hyperparameters. In more detail, we use geometric deep learning, specifically Temporal Graph Networks [23], since the time evolution is one of the most important features of social networks that are intrinsically time variant, often very quickly.

We performed several experiments aiming at predicting the evolving topology, i.e., creating and/or deleting links, based on previous events, on some dataset related to very popular social networks, such as Reddit, Wikipedia, Yelp, and Twitter. We selected four different aggregators to group messages involving the same node into batches and compared the results.

The main goals of this study were to assess that geometric deep learning algorithms are able to manage time dependencies and to analyze the dependencies of the performance on the hyperparameters used. Results show that batch size may affect the temporal granularity of the data even if running an extensive grid search that covers all the parameters is not always feasible due to computational constraints. We noted that batch size strongly impacts on Twitter, Wikipedia, and Yelp, while it does not matter on Reddit.

Future studies should both extend the analysis to more social networks and improve the geometric deep learning technique itself. Moreover, an explanation of the reason why TGN works will help to a better selection of the way to spatially and temporally represents the social networks.

Author Contributions: Data curation, C.C.; Methodology, V.C., M.G., M.M. and G.M.; Software, C.C. and M.G.; Supervision, V.C., M.G., M.M. and G.M.; Validation, V.C. and M.M.; Writing—original draft, V.C., M.G. and G.M.; Writing—review & editing, M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially supported by the project of University of Catania PIACERI, *PIAno di inCEntivi per la Ricerca di Ateneo*.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Wikipedia and Reddit datasets are the same discussed in paper [23], further details available in https://paperswithcode.com/dataset/reddit (accessed on 3 January 2022). Yelp was derived from data available in https://www.yelp.com/dataset (accessed on 3 January 20220 February 2022). Twitter was derived using a specific tool https://github.com/twintproject/twint (accessed on 3 January 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Linyuan, L.L.; Zhou, T. Link prediction in complex networks: A survey. *Phys. A Stat. Mech. Its Appl.* **2011**, *390*, 1150–1170.
2. Biondi, G.; Franzoni, V. Discovering Correlation Indices for Link Prediction Using Differential Evolution. *Mathematics* **2020**, *8*, 2097. [CrossRef]
3. Martínez, V.; Berzal, F.; Cubero, J.C. A Survey of Link Prediction in Complex Networks. *ACM Comput. Surv.* **2016**, *49*, 1–33. [CrossRef]
4. Perozzi, B.; Al-Rfou, R.; Skiena, S. DeepWalk: Online Learning of Social Representations. In *KDD '14, Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014*; Association for Computing Machinery: New York, NY, USA, 2014; pp. 701–710. [CrossRef]
5. Liben-Nowell, D.; Kleinberg, J. The Link Prediction Problem for Social Networks. In *CIKM '03, Proceedings of the Twelfth International Conference on Information and Knowledge Management, New Orleans, LA, USA, 3–8 November 2003*; Association for Computing Machinery: New York, NY, USA, 2003; pp. 556–559. [CrossRef]
6. Chen, H.; Li, X.; Huang, Z. Link prediction approach to collaborative filtering. In Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '05), Denver, CO, USA, 7–11 June 2005; pp. 141–142. [CrossRef]
7. Leskovec, J.; Mcauley, J. Learning to Discover Social Circles in Ego Networks. In *Advances in Neural Information Processing Systems*; Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Nice, France, 2012; Volume 25.
8. Bronstein, M.M.; Bruna, J.; LeCun, Y.; Szlam, A.; Vandergheynst, P. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Process. Mag.* **2017**, *34*, 18–42. [CrossRef]
9. Deng, L.; Yu, D. Deep Learning: Methods and Applications. Deep Learning: Methods and Applications; Foundations and Trends in Signal Processing. Now Publishers. 2014. Available online: https://www.nowpublishers.com/ (accessed on 20 February 2022).
10. Mikolov, T.; Deoras, A.; Povey, D.; Burget, L.; Černocký, J. Strategies for training large scale neural network language models. In Proceedings of the 2011 IEEE Workshop on Automatic Speech Recognition Understanding, Waikoloa, HI, USA, 11–15 December 2011; pp. 196–201. [CrossRef]
11. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.R.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
12. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Nice, France, 2014; Volume 27.
13. LeCun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks and applications in vision. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 30 May–2 June 2010; pp. 253–256. [CrossRef]
14. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
15. Conneau, A.; Schwenk, H.; Barrault, L.; Lecun, Y. Very Deep Convolutional Networks for Text Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Valencia, Spain, 3–7 April 2017; Association for Computational Linguistics: Valencia, Spain, 2017; pp. 1107–1116.

16. Wu, Y.; Lian, D.; Jin, S.; Chen, E. Graph Convolutional Networks on User Mobility Heterogeneous Graphs for Social Relationship Inference. In *IJCAI'19, Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019*; AAAI Press: Menlo Park, CA, USA, 2019; pp. 3898–3904.
17. Jin, D.; Liu, Z.; Li, W.; He, D.; Zhang, W. Graph Convolutional Networks Meet Markov Random Fields: Semi-Supervised Community Detection in Attribute Networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 152–159. [CrossRef]
18. Wu, Y.; Lian, D.; Xu, Y.; Wu, L.; Chen, E. Graph Convolutional Networks with Markov Random Field Reasoning for Social Spammer Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1054–1061. [CrossRef]
19. Fout, A.; Byrd, J.; Shariat, B.; Ben-Hur, A. Protein Interface Prediction Using Graph Convolutional Networks. In *NIPS'17, Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 6533–6542.
20. Xu, C.; Nayyeri, M.; Chen, Y.Y.; Lehmann, J. Knowledge Graph Embeddings in Geometric Algebras. *arXiv* **2021**, arXiv:2010.00989.
21. Hamilton, W.L. Graph Representation Learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2020**, *14*, 1–159. [CrossRef]
22. Xu, D.; Ruan, C.; Korpeoglu, E.; Kumar, S.; Achan, K. Inductive representation learning on temporal graphs. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020.
23. Rossi, E.; Chamberlain, B.; Frasca, F.; Eynard, D.; Monti, F.; Bronstein, M. Temporal Graph Networks for Deep Learning on Dynamic Graphs. *arXiv* **2020**, arXiv:2006.10637.
24. Monti, F.; Otness, K.; Bronstein, M.M. Motifnet: A Motif-Based Graph Convolutional Network for Directed Graphs. In Proceedings of the 2018 IEEE Data Science Workshop (DSW), Lausanne, Switzerland, 4–6 June 2018; pp. 225–228. [CrossRef]
25. Zhang, Z.; Cui, P.; Zhu, W. Deep Learning on Graphs: A Survey. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 249–270. [CrossRef]
26. Grassia, M.; De Domenico, M.; Mangioni, G. Machine learning dismantling and early-warning signals of disintegration in complex systems. *Nat. Commun.* **2021**, *12*, 5190. [CrossRef] [PubMed]
27. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv*, **2018**, arXiv:1710.10903.
28. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Nice, France, 2017; Volume 30.
29. Halevy, A.Y.; Canton-Ferrer, C.; Ma, H.; Ozertem, U.; Pantel, P.; Saeidi, M.; Silvestri, F.; Stoyanov, V. Preserving Integrity in Online Social Networks. *arXiv* **2020**, arXiv:2009.10311.
30. Monti, F.; Frasca, F.; Eynard, D.; Mannion, D.; Bronstein, M.M. Fake News Detection on Social Media using Geometric Deep Learning. *arXiv* **2019**, arXiv:1902.06673,
31. Zhou, X.; Zafarani, R. A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities. *ACM Comput. Surv.* **2020**, *53*, 1–40. [CrossRef]
32. Carchiolo, V.; Longheu, A.; Malgeri, M.; Mangioni, G.; Previti, M. A Trust-Based News Spreading Model. In *Complex Networks IX*; Cornelius, S., Coronges, K., Gonçalves, B., Sinatra, R., Vespignani, A., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 303–310.
33. Carchiolo, V.; Longheu, A.; Malgeri, M.; Mangioni, G.; Previti, M. Mutual influence of users credibility and news spreading in online social networks. *Future Internet* **2021**, *13*, 107. [CrossRef]
34. Previti, M.; Rodriguez-Fernandez, V.; Camacho, D.; Carchiolo, V.; Malgeri, M. Fake news detection using time series and user features classification. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin, Germany, 2020; Volume 12104 LNCS, pp. 339–353.
35. Sano, Y.; Zhao, Z.; Zhao, J.; Levy, O.; Takayasu, H.; Takayasu, M.; Li, D.; Wu, J.; Havlin, S. Fake news propagates differently from real news even at early stages of spreading. *EPJ Data Sci.* **2020**, *9*, 7.
36. Bessi, A.; Coletto, M.; Davidescu, G.A.; Scala, A.; Caldarelli, G.; Quattrociocchi, W. Science vs. Conspiracy: Collective Narratives in the Age of Misinformation. *PLoS ONE* **2015**, *10*, e0118093. [CrossRef]
37. Baronchelli, A. The emergence of consensus: A primer. *R. Soc. Open Sci.* **2018**, *5*, 172189. [CrossRef]
38. Vicario, M.D.; Quattrociocchi, W.; Scala, A.; Zollo, F. Polarization and Fake News: Early Warning of Potential Misinformation Targets. *ACM Trans. Web* **2019**, *13*, 1–22. [CrossRef]
39. Vosoughi, S.; Roy, D.; Aral, S. The spread of true and false news online. *Science* **2018**, *359*, 1146–1151. [CrossRef] [PubMed]
40. Villalba-Diez, J.; Molina, M.; Schmidt, D. Geometric Deep Lean Learning: Evaluation Using a Twitter Social Network. *Appl. Sci.* **2021**, *11*, 6777. [CrossRef]
41. Leng, Y.; Ruiz, R.; Dong, X.; Pentland, A.S. Interpretable Recommender System with Heterogeneous Information: A Geometric Deep Learning Perspective. *SSRN Electron. J.* **2020**. [CrossRef]
42. Rossi, E.; Frasca, F.; Chamberlain, B.; Eynard, D.; Bronstein, M.M.; Monti, F. SIGN: Scalable Inception Graph Neural Networks. *arXiv* **2020**, arXiv:2004.11198,
43. Kumar, S.; Zhang, X.; Leskovec, J. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In *KDD '19, Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1269–1278. [CrossRef]

44. Pennebaker, J.W.; Francis, M.E.; Booth, R.J. Linguistic inquiry and word count: LIWC 2001. *Mahway Lawrence Erlbaum Assoc.* **2001**, *71*, 2001.
45. Crossley, S.A.; Kyle, K.; McNamara, D.S. Sentiment Analysis and Social Cognition Engine (SEANCE): An automatic tool for sentiment, social cognition, and social-order analysis. *Behav. Res. Methods* **2017**, *49*, 803–821. [CrossRef]
46. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.