Machine Learning (2024) 113:2207–2226
https://doi.org/10.1007/s10994-023-06475-x

Temporal graph learning for dynamic link prediction with text in online social networks

Manuel Dileo[1] · **Matteo Zignani**[1] · **Sabrina Gaito**[1]

Received: 11 March 2023 / Revised: 11 October 2023 / Accepted: 17 October 2023 /
Published online: 29 November 2023

Abstract

Link prediction in Online Social Networks—OSNs—has been the focus of numerous studies in the machine learning community. A successful machine learning-based solution for this task needs to (i) leverage global and local properties of the graph structure surrounding links; (ii) leverage the content produced by OSN users; and (iii) allow their representations to change over time, as thousands of new links between users and new content like textual posts, comments, images and videos are created/uploaded every month. Current works have successfully leveraged the structural information but only a few have also taken into account the textual content and/or the dynamicity of network structure and node attributes. In this paper, we propose a methodology based on temporal graph neural networks to handle the challenges described above. To understand the impact of textual content on this task, we provide a novel pipeline to include textual information alongside the structural one with the usage of BERT language models, dense preprocessing layers, and an effective post-processing decoder. We conducted the evaluation on a novel dataset gathered from an emerging blockchain-based online social network, using a live-update setting that takes into account the evolving nature of data and models. The dataset serves as a useful testing ground for link prediction evaluation because it provides high-resolution temporal information on link creation and textual content, characteristics hard to find in current benchmark datasets. Our results show that temporal graph learning is a promising solution for dynamic link prediction with text. Indeed, combining textual features and dynamic Graph Neural Networks—GNNs—leads to the best performances over time. On average, the textual content can enhance the performance of a dynamic GNN by 3.1% and, as the collection of documents increases in size over time, help even models that do not consider the structural information of the network.

Keywords Graph neural networks · Dynamic graphs · Network analysis · Online social networks

Editors: Dino Ienco, Roberto Interdonato, Pascal Poncelet.

Extended author information available on the last page of the article

1 Introduction

Real-world networks such as traffic, citation, or social networks often evolve over time. To extract, learn, and make predictions from these dynamic networks the field of Temporal Graph Learning has gained increasing attention from the ML community. In particular, the problem of dynamic link prediction is meaningful for solving numerous issues in many application domains such as fraud detection or recommender systems. Alongside the structural information, i.e. the graph, many dynamic networks can also provide attributes describing the links or the nodes of the network, and even these attributes may change over time.

The combination of multi-modal interactions, networked data, and dynamicity reaches its highest expression in online social networks—OSNs—where thousands of new links between users are created every month, thanks to different mechanisms that characterize the behaviors of people within society, such as the balance theory or the homophily principle (Khanam et al., 2022). Moreover, users typically also produce content like textual content—posts and comments to posts—describing their activities, hobbies, or opinions which may change over time (Monti et al., 2013). Hence, when it comes down to dynamic link prediction in OSNs using textual information, a successful machine learning-based solution needs (i) to leverage global and local properties of the graph structure surrounding links; (ii) to include the textual content produced by OSN accounts; and (iii) to allow their representations to change over time dynamically. Finally, both the training and evaluation settings of the problem must take into account the evolving nature of data and models.

However, nowadays solutions and methods lack at least one of the described requirements. As for the second requirement—textual content—only a few works consider additional contextual information such as textual content alongside the traditional structural one. Therefore, our knowledge of how text affects link creation is currently restricted. One of the main reasons is that it is hard to obtain appropriate data for the task: current research lacks high-resolution temporal annotated data on network growth and/or on textual information. This a crucial problem to cope with as the information acquired from the text may enhance prediction and provide insight into the mechanisms guiding the link creation process. In fact, the text is fundamental in OSNs since it is one of the strongest drivers for user involvement, and content advertisements are the primary source of income for these platforms. As for the third requirement—handling changes over time—Graph Neural Networks (GNNs) are becoming a very promising deep learning model for graph-structured data and they are currently successfully applied to many static real-world attributed graphs; however, despite various GNNs proposed for dynamic graphs (Gupta & Bedathur, 2022), a vast majority of these approaches have limitations in model design, evaluation settings, and training strategies (You et al., 2022).

Hence, in this work, we performed future link prediction with textual information on a temporal attributed network using a dynamic GNN model. We adopted the ROLAND (You et al., 2022) graph learning framework for dynamic graphs which helps to repurpose any static GNN to a dynamic setting and introduces new training and evaluation procedures suited for learning from dynamic networks. Starting from this kind of temporal and heterogeneous data, in this work, we define a methodology to include text information to perform future link prediction on multiple following graph snapshots. Nodes—users—in the graph are characterized by a set of textual features that capture the semantics of their content and the topics they talk about. Then, we investigate the impact of these textual features on the link prediction task in a dynamic setting, understanding (i) how using textual features can enhance the performance on future link prediction tasks, (ii) which is a good strategy to update the textual representation; and highlight (iii) strengths and weaknesses of different GNN-based models.

We choose Steemit, a blockchain-based online social network, as a case study, since it makes it possible to retrieve high-resolution temporal data on social relationships and textual content, leading to the creation of an attributed temporal network. Specifically, we focused on and gathered temporal data about "follow" relationships between users and text content produced by users—posts and comments. By the nature of blockchains, data are publicly available, validated, and affordable by interfacing with the blockchain API. Moreover, each piece of information is also timestamped, since each blockchain block has a validation timestamp, and each block reports multifaceted interactions and content (social, economic, financial, and textual). In light of this, these data sources are fully equipped to handle the challenges and concerns related to modern techno-social networks and to provide a comprehensive and in-depth examination of users and network characteristics.

The outcome of the prediction task, resulting from the application of the above methodology, shows that the combination of textual features and dynamic GNN leads to the best performances over time. Moreover, through an exhaustive comparison of different ROLAND-based models, we also provide insights on the importance of textual content snapshot by snapshot, as the collection of documents increases in size. We find a good compromise between the importance of past and current node embeddings in the dynamic representation of textual content by testing different embedding update-modules. Finally, we discuss potential extensions for this work.

We can summarize our main contributions as follows: (i) we propose a methodology to leverage topological and user-generated textual content from online social network data by learning both structural information and document embeddings to predict future "follow" links, introducing a novel model architecture based on the ROLAND framework, with the addition of an effective decoder for link prediction and dense pre-processing layers to fine-tune the text embeddings obtained from a pre-trained language model; (ii) we train and evaluate temporal graph neural networks over the "follow" prediction task on a novel dataset gathered from an emerging blockchain online social network, which offers high-resolution temporal information, using a recently introduced live update protocol; and (iii) we investigate the role of textual content in learning from multiple following graph snapshots by analyzing different node embedding update strategies and baselines to highlight how textual content influences their performances.

The paper is organized as follows. Section 2 provides a brief introduction to the nature of blockchain-based online social networks and a review of works related to dynamic link prediction and link prediction with text. In Sect. 3 we describe the construction of the multiple following snapshot graphs, the graph learning framework for dynamic graphs, the model for predicting links, and how textual features are extracted. In Sect. 4 we provide a description of the dataset, while Sects. 5 and 6 report the main findings of link prediction and a discussion about strengths and weaknesses of the proposed approach.

2 Background

Dealing with the task of dynamic or future link prediction in online social networks within the framework of temporal graph learning involves methods from different research fields, especially when we also consider user-generated content and text. In the following, we describe works related to the standard setting for dynamic link prediction focusing on solutions based on graph neural networks for dynamic graphs. Then, we review the main methods of temporal graph learning, with a special focus on the integration of textual information. We also discuss works integrating topological and content-based approaches for social network analysis. Finally, we summarize the main features of the platform our social data comes from, along with works dealing with blockchain-based online social networks.

Dynamic link prediction. The problem of dynamic link prediction is meaningful for solving numerous issues in many application domains such as fraud-detection (Bruss et al., 2019), recommender systems (You et al., 2019), or online social networks (Barracchia et al., 2022). Its main objective is to estimate network evolution by inferring the likelihood that pairs of nodes have to either form links or not in the future. It is also known as the future link prediction task. Kumar et al. (2020) reviewed several approaches to link prediction from classical to recent network embedding and deep learning techniques. Graph Neural Networks (GNNs) are a family of deep learning models that represent cutting-edge technology to learn from dynamic networks. Although various GNNs have been proposed for dynamic graphs (Pareja et al., 2020; Zhao et al., 2020; Seo et al., 2018; Yu et al., 2018; Li et al., 2018), these approaches have limitations in model design, evaluation settings, and training strategies. In fact, most of the works do not incorporate state-of-the-art designs from static GNNs, and training and evaluation procedures, heavily influenced by static graph learning, are performed using a fixed train-test split strategy. To overcome these limitations, You et al. (2022) propose a graph learning framework that allows the repurposing of any static GNN to dynamic graphs and performs training and evaluation procedures in a live update setting. We adopt their framework in this work to benefit from its model design and training and evaluation strategies. In addition to the ROLAND model design, our model leverages an effective decoder for link predictison and two dense pre-processing layers for fine-tuning the document embedding representations obtained from a pre-trained language model.

Link prediction with text. Only a few studies have evaluated the role of textual node-related data in enhancing performances in link prediction tasks. Among these works, Parimi and Caragea (2011) rely on users' textual attributes to model user profile data, using Latent Dirichlet Allocation—LDA—to model topics; but here, link prediction is only based on the resulting topic distributions, and not on the network structure. Other works, such as Wang et al. (2018), have improved prediction performance by fusing a network generated from users' posts with the original "follow" network, but they do not consider content-based features. Xu et al. (2021) used unstructured text content from heterogeneous datasets (e.g. papers from DBLP) to obtain topic-aware node embedding representations with GNNs. Only recently, in the field of recommender systems, LMs and GNNs were combined to obtain knowledge-aware recommendations (Spillo et al., 2023). Overall, using text to make predictions seems to improve performance; however, all these methods have only been tested on static networks. Additionally, there is a lack of knowledge regarding the best strategies to employ text-based features. In this work, we cope with the latter problems using sentence encoders based on LMs, which represent the state-of-the-art on different NLP tasks (Reimers & Gurevych, 2019), in combination with temporal graph neural networks (Longa et al., 2023), to analyze an evolving online social network in a recently proposed temporal training and evaluation setting.

Content-based social network analysis. Approaches integrating topological and content have also been successfully used in a few works dealing with different online social network analysis applications. For instance, in Garimella et al. (2021), authors analyze browsing histories of users leveraging both the link structure of online news networks and the users' explicit content choices by contributing to understanding polarization in online news consumption. In Villa et al. (2021), they propose an approach based on the application of a community detection strategy to distinct topology—and content-aware representations of the COVID-19 conversation graph on Twitter to detect echo chambers. In Kumar et al. (2018), they exploit the Reddit hyperlink network and Word2vec-based user and subreddit embeddings for analyzing community interactions and conflicts on the platform.

Blockchain-based OSNs. By Blockchain-based Online Social Network (BOSN), we refer to a web application that (i) enables online interactions between accounts by providing a set of "social action", such as following, commenting, and voting; and (ii) whose core functions are supported by an underlying blockchain that ensures the persistence and validity of the operations and stores each piece of information with a high-resolution timestamp. As every action is stored on a blockchain, these platforms provide a detailed data source of network activity, covering not only the social side but also the economic sphere; for example, cryptocurrency exchanges between users. In recent years, many different research fields have benefited from these large collections of temporal and heterogeneous data, which capture different aspects of the interactions among people and between people and platforms. Most of the research studies on BOSNs have been focused on Steemit since is one of the most widespread BOSN platforms and is considered a pioneer in the BOSNs ecosystem. The most relevant advancements and issues are illustrated in a few recent works (Guidi, 2021; Ba et al., 2022a, 2022b).

In Dileo et al. (2022) we applied state-of-the-art graph neural networks to evaluate the impact of textual content on link prediction in Blockchain-OSNs. We modeled Steemit as a temporal attributed graph and we showed that (a) GNNs outperform well-established methods such as logistic regression or ensemble methods; and (b) Prediction performance of GNNs can be enhanced using textual features as node attributes. Although the work has highlighted the impact of textual content on link prediction and on the network evolution as well, (i) it considers only a few text-based statistics and shallow features to obtain textual representations, without taking into account pre-trained deep learning language models that offer high-dimensional and semantic-based text embeddings; and (ii) it performs future link prediction on one future snapshot only without proposing a methodology to deal with multiple following snapshots. Here we extend this previous work by acting in both aforementioned directions.

3 Methodology

In online social networks, users post content for other users. If user A is interested in the content user B writes, A can start following user B to receive updates on her/his posts. Due to this mechanism, A can also see the post written by user C reshared by B and start to follow C, if interested. Alongside this information, we also have user-generated content, i.e. posts and comments on posts, that may impact the formation of "follow" links. Here, we aim to answer the following research questions: how do we handle dynamic textual and structural information to perform link prediction in online social networks? How do textual features impact future link prediction tasks on multiple following snapshots?

To answer these questions we propose a methodology based on the ROLAND graph learning framework for dynamic graphs (You et al., 2022). In the next subsections, we will describe how we model the problem, the textual features used for link prediction, the

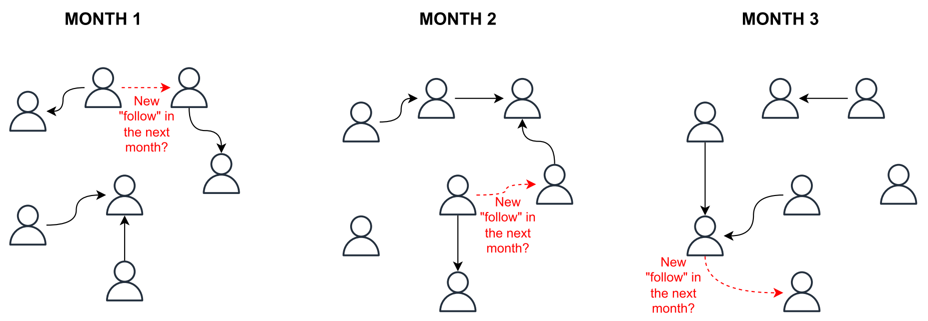
MONTH 1
MONTH 2
MONTH 3
New "follow" in the next month?
New "follow" in the next month?
New "follow" in the next month?

Fig. 1 Example of future link prediction in online social networks. We use information up to time t to predict potential edges at time $t + 1$

architecture of our model, and the ROLAND framework. Data used in this work refers to a high-resolution temporal annotated OSN with textual information associated with nodes.

3.1 Future link prediction and graphs construction

"Follow" links and text information can be modeled as an attributed temporal directed graph $\mathcal{G} = (V, E, T, X)$, where V is the set of users, links $(u, v, t) \in E$ denote a directed "follow" link from user u to user v at time t (the time in which user u starts to follow user v), T is the set of timestamps, and X is a $|V| \times f$ matrix of node attributes, with f the dimension of attribute vectors (Liu et al., 2023). Given a time interval $[t_0, t_1]$, the snapshot graph $\mathcal{G}_{[t_0,t_1]}$ represents the directed graph, where for each link $e = (u, v, t) \in E$, we have that $t \in [t_0, t_1]$. For simplicity, since all the edges in a certain time interval are treated as they share the same timestamp, we use the notation $\mathcal{G}_t$ to denote a snapshot graph, where t is a time interval.

Given a snapshot graph $\mathcal{G}_t$, the purpose of future link prediction is to predict which edges will appear at a successive graph snapshot $\mathcal{G}_{t+1}$. The problem can be treated as a binary classification task, where we assign label 1 if the link is predicted to form in the following time interval, 0 otherwise (Liben-Nowell & Kleinberg, 2003). Figure 1 shows an example of our setting for future link prediction on an online social network. We use information up to time t to predict potential edges at time $t + 1$. We adopt a transductive setting to predict new "follow" links between the nodes observed in the initial snapshot along all the future snapshots.

The main idea is to realize a sequence-based framework so that training and evaluation of the link prediction algorithms can be assessed on successively built datasets. To this aim, we rely on the experimental setting for temporal link prediction presented in Liu et al. (2016). Given a sequence of time intervals $[t_0, ..., t_n]$:

- $\mathcal{G}_{t_0}$ is used to retrieve the list of edges, their relative nodes, and the textual features related to the collection of documents posted in time interval t_0.
- $\mathcal{G}_{t_i}$ is obtained as an induced sub-graph constrained around the nodes of G_{t_0}. This limitation makes it possible to effectively understand how a graph and its connections evolve. Then, only the edges closed in t_i and not in t_{i-1} are considered to form the positive set. Simultaneously, starting from the same seed of nodes, a set of randomly extracted non-existing edges is considered to form the negative set. The final dataset results in the

combination of the positive and the negative sets; and for each item, a binary label y is added to indicate if that item is an existing edge or not. If $i < n$, where n is the number of snapshots, textual features related to the collection of documents posted in time interval t_i are computed.

Selecting a subset of edges at random from the original complete set is standard practice to obtain a negative set (Pareja et al., 2020). Despite this strategy may lead to over-optimistic results, there are evaluation measures for which subsampling negatives from the test set has no negative effects (Yang et al., 2014).

3.2 Textual representation

We collect for each user the collection of posts and comments s/he wrote in a certain time interval to obtain textual information as node features. For instance, we were able to retrieve all the content written by a user during September 2016 and process them according to the methodology described below to obtain a vector-based representation of the textual information s/he wrote and use it as attributes of the node representing the user in the "follow" graph.

We use a pre-trained BERT-based model to get Euclidean representations for the documents, as the BERT language model pre-training represents the state-of-the-art performance on various NLP tasks like sentence classification or sentence-pair regression tasks (Devlin et al., 2019). In particular, we choose Sentence-BERT (SBERT) (Reimers & Gurevych, 2019) as the text embedding model because it is able to derive semantically meaningful sentence embeddings. This means that semantically similar sentences will be close in vector space so it allows us to easily follow the homophily principle between users w.r.t. the textual content (Khanam et al., 2022).

An example of the architecture of an SBERT model is shown in Fig. 2. SBERT adds a pooling operation to the output of BERT to derive a fixed-sized sentence embedding. In our work, we use the average pooling operation. Then, in order to fine-tune BERT, it creates siamese and triplet networks (Schroff et al., 2015) to update the weights such that the produced sentence embeddings are semantically meaningful and can be compared by cosine similarity.

We use the `all-MiniLM-L6-v2` SBERT model to obtain a vectorial representation for each post and comment written by a user. We choose this model because (i) is trained on all available training data (more than 1 billion training pairs), (ii) is designed as a general purpose model, and (iii) is five times faster than the best SBERT model but still offers good quality.[1]

To obtain the initial features for each node, we average the document embeddings on each user's collection. If a user has not written posts and comments during a certain time interval, its initial textual features will be a zero vector. Note that, as described in Sect. 4, no user has missing node features because, in the first time interval, they all have written at least one post or comment.

Formally, we summarize the procedure to obtain text embeddings as node features as follows. For each time interval t, we call $D_{(u,t)}$ the collection of documents (posts and comments) posted by user u during time interval t. To obtain the initial node features $X_{(u,t)}$ of u

[1] https://www.sbert.net/docs/pretrained_models.html

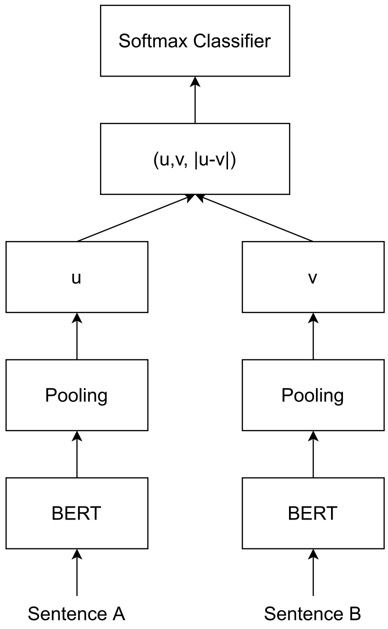

Fig. 2 Example of an SBERT architecture with classification objective function, e.g. for fine-tuning on a certain dataset. The two BERT networks have tied weights (siamese network structure)

at time t, we average its document embeddings, i.e. $X_{(u,t)} = \frac{1}{|D_{(u,t)}|} \sum_{d \in D_{(u,t)}} \text{SBERT(d)}$ using the element-wise sum.

3.3 The ROLAND framework

ROLAND You et al. (2022) is a graph representation learning system for building, training, and evaluating dynamic Graph Neural Networks. This subsection will summarize the main ideas behind this framework.

Graph Neural Networks Graph Neural Networks (GNNs) are deep learning models that work naturally with graph-structured data Wu et al. (2021). GNNs learn to map individual nodes to fixed-size real-valued vectors called embeddings. The learned embeddings summarize the structural information of the network taking into consideration also the attributes of the nodes. Then, those vectorial representations can be used to solve different useful problems on graphs (e.g. link prediction). They generate a node v's representation by aggregating neighbors' features h_u, where $u \in N(v)$ (u's neighborhood), and combining them with its features h_v. Formally, we can use the matrix $H^{(L)} = \{h_v^{(l)}\}_{u \in V}$ to denote the embedding for all the nodes after applying an L-layer GNN. The l-th layer of a GNN, $H^{(l)} = \text{GNN}^{(l)}(H^{(l-1)})$, can be written as:

$$\begin{aligned} m_{u\to v}^{(l)} &= \text{MSG}^{(l)}\left(h_u^{(l-1)}, h_v^{(l-1)}\right) \\ h_v^{(l)} &= \text{AGG}^{(l)}\left(\left\{m_{u\to v}^{(l)} | u \in \mathcal{N}(v)\right\}, h_v^{(l-1)}\right) \end{aligned} \tag{1}$$

where $h_v^{(l)}$ is the node embedding for $v \in V$ after passing through l layers, $h_v^{(0)}$ is the initial representation for the node (i.e. its node features), $m_v^{(l)}$ is the message embedding, and $\mathcal{N}(v)$ is the direct neighbors of v. Different GNNs can have various definitions of message-passing (MSG) and aggregations (AGG) functions.

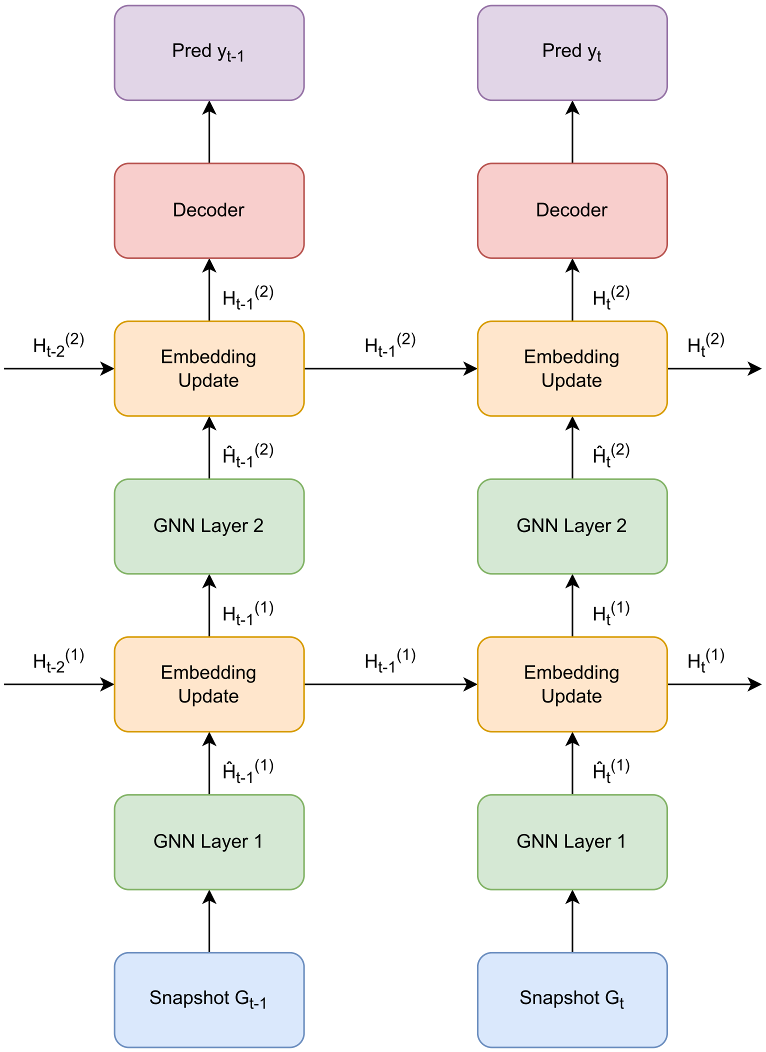

Fig. 3 ROLAND model design principle. ROLAND extends any static GNN to dynamic graphs by inserting embedding update modules that update hierarchical node states H_t over time

ROLAND model design Fig. 3 shows how to generalize a GNN to a dynamic setting following the ROLAND model design principle. The two main ROLAND innovations are viewing the node embeddings at different GNN layers as hierarchical node states, and then recurrently updating them over time through a customizable embedding module. We summarize the forward computation of a ROLAND-based model in Algorithm 1. The UPDATE function at line (4) can be any function that takes the previous and current node state at a certain layer and produces a new current node state mixing the two pieces of information. The DECODER function at line (6) takes any candidate pair of the current snapshot and emits in output a score s: the more likely the link between the two nodes

```
    Input: Dynamic graph snapshot $\mathcal{G}_t$, hierarchical node state $H_{t-1}$
    Output: Prediction $y_t$, updated node state $H_t$
1: $H_t^{(0)} \leftarrow X_t$                                   ▷ Initialaze node embeddings from $\mathcal{G}_t$
2: for $l = 1, ..., L$ do
3:     $\hat{H}^{(l)} = \text{GNN}^{(l)}(H_t^{(l-1)})$                          ▷ Equation 1
4:     $H_t^{(l)} = \text{UPDATE}^{(l)}(H_{t-1}^{(l)}, \hat{H}_t^{(l)})$          ▷ Embedding update module
5: end for
6: $y_t = \text{DECODER}(h_u^{(L)}, h_v^{(L)}), \forall(u, v, ts) \in E, ts \in t$
```

Algorithm 1 ROLAND GNN forward computation

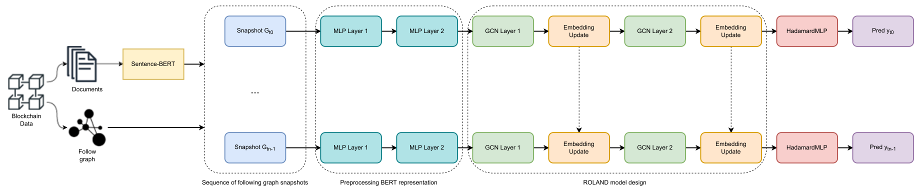

Fig. 4 Pipeline of our methodology to perform dynamic link prediction with text on a blockchain online social network

exists, the higher the score is. For simplicity, the algorithm only reports the notation to compute the scores on the positive set.

Training and evaluation procedure We can summarize the training and evaluation procedure of a ROLAND-based model in the following steps:

1. Split the dataset of the current snapshot into train and validation set.
2. Train the model on the train set optimizing the binary cross-entropy loss until the prediction performance on the validation set does not get worse (early-stopping condition).
3. Compute the prediction performance on the next snapshot.
4. Repeat steps 1–3 from the first snapshot until the second last.
5. Compute the prediction performance of the model over time by averaging the performance over the single snapshots.

Note that the model is not re-initialized after each snapshot, so the training procedure, after the first snapshot, is a fine-tuning step on the trained model. This kind of training and evaluation procedure is called *live-update* setting.

3.4 Model for dynamic link prediction with text

Figure 4 shows the pipeline of our methodology. Starting from the blockchain, we collect both "follow" links and documents written by users in a certain time interval. We process text with Sentence-BERT to produce Euclidean textual representation for the collection of documents posted by each user, as described in Sect. 3.2; then, we construct the sequence of snapshot graphs as described in Sect. 3.1, to finally feed our model using the live-update setting described in Sect. 3.3.

The architecture of our model includes: (i) two MLP layers to preprocess the node features, i.e. the high-dimensional BERT representations, (ii) a dynamic two-layer GCN (Kipf & Welling, 2017) based on the ROLAND model design; and (iii) an HadamardMLP (Wang

Table 1 Number of input and output channels for each deep learning layer in our model

Layer	in_channels	out_channels
MLP1	384	256
MLP2	256	128
GCN1	128	64
GCN2	64	32
HadamardMLP	32	2

et al., 2022) as decoder, typically more effective than other decoders for link prediction. Table 1 resumes the dimensions of each deep learning layer. For the embedding update modules, we consider three simple yet effective methods:

1. *Learnable weighted average (lwa)* Node embeddings $\hat{H}_t^{(l)}$ are updated using the formula

$$H_t^{(l)} = \tau * H_{t-1}^{(l)} + (1 - \tau) * \hat{H}_t^{(l)}$$

   where τ is a learnable parameter.
2. *ConcatMLP* Node embeddings are updated by a 1-layer MLP

$$H_t^{(l)} = \text{MLP}\left(\text{CONCAT}\left(H_{t-1}^{(l)}, \hat{H}_t^{(l)}\right)\right)$$

3. *GRUCell* Node embeddings are updated by a GRU cell (Chung et al., 2014)

$$H_t^{(l)} = \text{GRU}\left(H_{t-1}^{(l)}, \hat{H}_t^{(l)}\right)$$

   where $H_{t-1}^{(l)}$ is the hidden state and $\hat{H}_t^{(l)}$ is the input for the GRU Cell.

The implementation of our model is written using PyTorch Geometric (Fey & Lenssen, 2019) and is available on a GitHub repository.[2]

4 Dataset

In this section, we will briefly describe the features of the social platform Steemit more related to our work and introduce the dataset we collected to evaluate our methodology.

Users on Steemit have access to a wide range of operations that keep track of user activity with a *three-second* temporal precision and are retrievable through specific API. Data from June 3, 2016, through January 02, 2017, have been gathered through the API. The starting date corresponds to when the "follow" operation has been made available on Steemit.

We considered two types of information: (a) the "follow" relationships, available in the `custom_json` transactions; and (b) posts, comments, and tags, available in the `comment` transactions. Overall, we processed 1,273,657 "follow" operations and 2,122,163 documents. To produce the sequence of attributed graphs, the dataset was processed in accordance with the technique described in Sect. 3.

[2] https://github.com/manuel-dileo/dynamic-gnn

Table 2 Summary of dataset statistics

Snapshot	New edges	Num. of documents	Avg. docs per author	Authors' rate (%)
Jun–Aug	–	1,083,390	41.35	100
September	52,842	321,119	36.63	42
October	29,966	249,264	37.9	31.5
November	19,855	226.717	41.28	26.3
December	18,378	241,677	43.27	26.8

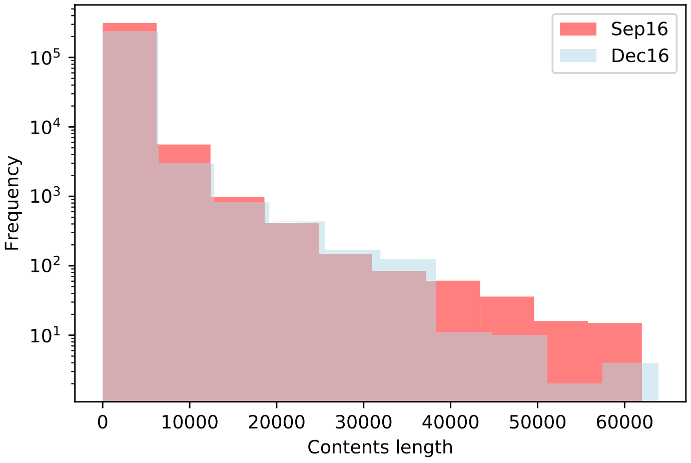

Fig. 5 Distributions of contents length produced in September (red histogram) and December (light blue histogram) 2016. The users' effort to write posts and comments is quite similar (Color figure online)

As the network has just started to grow, we used the first three months of data as the initial train snapshot to have enough nodes on which to predict new links and a sufficient number of edges to learn the networked structure. The first snapshot has 20,849 nodes and 138,604 edges. Then, we used a monthly snapshot frequency.

Table 2 provides summary statistics of the dataset snapshots. For each snapshot, we reported the number of new edges, documents (posts and comments), the average number of documents written by authors, i.e. users that have written at least one comment or post, and the authors' rate, i.e. the number of authors over the number of nodes. We observe that the number of new edges slightly decreases over time but it remains in the tens of thousands, while the number of authors drastically plums. However, the average number of documents written by authors remains more or less the same and the initial authors' rate is equal to 100%, therefore no node has started with missing features.

Figure 5 shows the distributions of contents length produced in September and December 2016, i.e. the first and last test snapshots. Despite the number of documents and authors decreasing over time, the two distributions seem quite similar, so users' effort to write posts and comments on Steemit does not decrease and the new textual content cannot be ignored.

Lastly, over 85% of the contents are written in English,[3] so the production of content is almost monolingual.

5 Experiments

In this section, we describe the experimental evaluation of temporal graph learning models for dynamic "follow" link prediction in a novel blockchain-based online social network, i.e. Steemit. Specifically, we detail the evaluation setting as well as other alternative approaches and architectures. We delve into the analysis of the experimental evaluation results, emphasizing the optimal combination of modules within the overarching architecture. The evaluation has highlighted that the integration of content-based features within a temporal graph learning framework can enhance performance when dealing with the link prediction task. Finally, code, data, and supplementary information about the experiments can be found in our GitHub repository.[4]

5.1 Experimental Setup

Task. We evaluate our model based on the ROLAND framework over the future link prediction task. At each time t, the model utilizes information accumulated up to time t to predict edges in snapshot $t + 1$. We use the area under the precision-recall curve (AUPRC) to evaluate candidate models, as suggested in Yang et al. (2014) and adopted in prior works as well (Rossi et al., 2020). We perform random negative sampling[5] to obtain the same number of positive and negative examples in the dataset. We consider the *live-update* train-test split method (You et al., 2022), which evaluates model performance over all the available snapshots. We randomly choose 25% of edges in each snapshot to construct the validation set and determine the early-stopping condition. We use Adam (Kingma & Ba, 2015) as the optimizer, conducting hyperparameter tuning via grid search over learning rate and weight decay, testing orders of magnitudes from 10^{-1} up to 10^{-5}. We run all the experiments with 3 different random seeds, reporting the average result and standard deviation for each method, as in You et al. (2022). *Baselines.* We evaluate our model using *lwa*, *ConcatMLP* and *GruCell* embedding-update modules. We compare our method with five baselines:

1. *GCN* A GCN model (Kipf & Welling, 2017) with document embeddings as node features. GCN is a deep learning model for attributed static graphs so it does not take into account the dynamicity of the network.
2. *MLPText* A MLP model on textual features. This model does not take into account the "follow" links between users and makes predictions based only on textual content.
3. *ROLANDStruct* A ROLAND-based model without textual features (i.e. a single constant value equal for all the nodes as a feature) that makes predictions based only on the dynamic structure of the network.

[3] Analysis done using the algorithm in Google's language-detection library https://code.google.com/archive/p/language-detection/

[4] https://github.com/manuel-dileo/dynamic-gnn

[5] We sampled negative edges in the dataset due to memory constraint.

Table 3 AUPRC of ROLAND-based models for dynamic link prediction with textual features over different snapshots (months from September 2016 to December 2016) and over time using the learnable weighted average (lwa), ConcatMLP (MLP), GRUCell (GRU) as embedding updated modules

Month/model	LWA	MLP	GRU
September16	0.897 ± 0.008	0.915 ± 0.002	0.915 ± 0.002
October16	0.899 ± 0.007	0.906 ± 0.005	0.900 ± 0.001
November16	0.854 ± 0.009	0.868 ± 0.006	0.844 ± 0.003
December16	0.820 ± 0.032	0.862 ± 0.009	0.834 ± 0.009
Over time	0.867 ± 0.014	0.887 ± 0.003	0.873 ± 0.003

The best performances are achieved using the ConcatMLP model

Table 4 AUPRC of no dynamic baselines models for link prediction over different snapshots (months from September 2016 to December 2016) and over time

Month/model	Our model	GCN	MLPText	Dummy
September16	0.915 ± 0.002	0.894 ± 0.015	0.642 ± 0.119	0.498 ± 0.002
October16	0.906 ± 0.005	0.434 ± 0.115	0.454 ± 0.075	0.500 ± 0.001
November16	0.868 ± 0.006	0.817 ± 0.042	0.566 ± 0.144	0.504 ± 0.002
December16	0.862 ± 0.009	0.815 ± 0.039	0.636 ± 0.141	0.505 ± 0.002
Over time	0.887 ± 0.003	0.741 ± 0.014	0.574 ± 0.066	0.502 ± 0.001

Our model outperforms the others thanks to the fine-tuning and embedding update steps

4. *ROLANDMLP* MLP with hierarchical document embeddings update modules (i.e. a ROLAND-based model without network structure). As *MLPText*, it does not consider the structure of the “follow” graph but, on the contrary, it allows the textual representation to be updated over time.
5. *DummyClassifier* A dummy fuzzy classifier that emits in output a value $v \in [0, 1]$ at random for each example.

Note that the ROLAND-based models are updated through fine-tuning over the snapshots, while *GCN* and *MLPText* are re-initialized on a new snapshot. We made this choice because we want to compare our model with solutions outside the temporal graph learning framework and typically used on static graphs.

5.2 Results

All the results will be presented in terms of AUPRC scores for future link prediction tasks over different snapshots (months from September ’16 up to December ’16) and over time, i.e. by averaging the AUPRC scores over the months. Table 3 shows the results of our model using *learnable weighted average*, *ConcatMLP*, or *GRUCell* as the embedding update module. The best performances are achieved using the *ConcatMLP* model because its final representation likely represents a good compromise between past and current node embeddings. The GRU Cell, instead, gives too much importance to the past representation, as the *lwa* update module achieves its best performance with small values of τ. However, *lwa* possibly suffers from being too simple as an aggregation method.

Table 5 AUPRC of ROLAND-based baselines models for link prediction over different snapshots (months from September 2016 to December 2016) and over time

Month/model	Our model	ROLANDStruct	ROLANDMLP
September16	0.915 ± 0.002	0.908 ± 0.016	0.611 ± 0.060
October16	0.906 ± 0.005	0.903 ± 0.008	0.671 ± 0.094
November16	0.868 ± 0.006	0.853 ± 0.006	0.787 ± 0.088
December16	0.862 ± 0.009	0.821 ± 0.020	0.802 ± 0.048
Over time	0.887 ± 0.003	0.871 ± 0.012	0.718 ± 0.065

The combination of textual features and dynamic GNN leads to the best performances

Table 4 shows the results of *GCN*, *MLPText*, and *DummyClassifier*, which are the baselines designed outside the ROLAND framework (i.e. without node embedding updates and fine-tuning over time). Our model outperforms these three baselines. These results show the effectiveness of the model design principles behind the ROLAND framework on the future link prediction task.

Finally, Table 5 shows the results of ROLAND-based models for dynamic link prediction using textual features and graph convolutions (our model), no node features (*ROLANDStruct*), and no graph convolutions (*ROLANDMLP*). As described in Sect. 5.1, we recall that the *ROLANDStruct* model learns the structure of the graphs only, while *ROLANDMLP* is fed with the collection of documents only. The combination of textual features and dynamic GNN, implemented by our solution, leads to the best performances. The *ROLANDStruct* model works well on the first two snapshots but its performance gets worse as the collection of documents increases in size, while an opposite trend has characterized the performances of *ROLANDMLP*. The two different behaviors show the importance of textual content on future link prediction on multiple following snapshots: after the first months of Steemit's life, using textual information or not leads to a change in performance up to 10% even for neural networks designed for dynamic graphs.

5.3 Scalability and generality of the solution

The proposed solution is general in nature and can be applied to make future link predictions on every kind of social platform that offers high-resolution temporal networked and user-generated textual data. To this aim, we run our framework on another BOSN dataset. Specifically, we focus on the financial layer of Steemit, gathering the economic transactions between the users and their textual content over one year and a half period. The period starts on June 3, 2016, and ends on December 31, 2017. The first six months are used as the initial training snapshot while the remaining year is divided into two-week snapshots. Overall, we process 274, 872 transaction operations and 241, 677 comment operations, obtaining a graph with 14, 814 nodes and 106, 614 edges over 26 snapshots. We report the results of our model snapshot-by-snapshot in Fig. 6. Over time, the average performance of our model is an AUPRC score of 0.893, which is in line with the results presented on the "follow" link prediction tasks.

These experiments show also the scalability of our solution with more than four graph snapshots, as in the case of "follow" links. We run our experiments on NVIDIA Corporation GP107GL [Quadro P400]. The training procedure of one configuration of our model on the financial layer takes about 5 min.

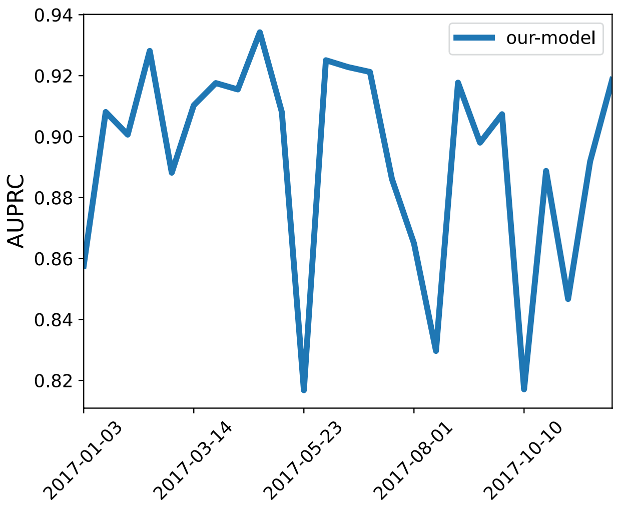

Fig. 6 Performances of our temporal graph neural network model for the transaction prediction task, an instance of the future link prediction, over the two-week snapshots of Steemit data, in terms of AUPRC

6 Discussion

In this work, we used temporal graph learning to investigate the role of textual information on link formation on a dynamic network, an important task as text could improve prediction and give insight into the link formation process. To this end, we performed future link prediction with text on a temporal attributed network. We relied on Steemit, a blockchain-based online social network, that allows the retrieval of validated high-resolution temporal information. We provided a methodology based on the ROLAND graph learning framework for dynamic graphs including how to obtain user textual features using a pre-trained BERT-based language model. We evaluated our model on the first 6 months of Steemit, splitting them into monthly following graph snapshots.

We showed that the combination of textual features and dynamic GNN leads to the best performances over time. This means that, in general, the best results are achieved using both structural and textual information, allowing their representation to be updated along the different snapshots. The performance gain of using textual content compared to a ROLAND-based model without node features increases over time and, on average, text features can enhance the performance by 3.1%.

We also investigated three simple yet effective methods to perform the update of node embeddings. The best embedding update module, among the ones considered, is the *ConcatMLP* model because its final representation likely represents a good compromise between past and current node embeddings. The GRU Cell, instead, gives too much importance to the past representation, whereas the learnable weighted average update achieves better performance with small values of τ.

Lastly, we exhibited the role of textual content on multiple following snapshots. A dynamic model that does not use textual information works well on the first two snapshots but its performance gets worse as the collection of documents increases in size. Most importantly, we can observe the opposite trend for a dynamic model that uses text data only.

In general, temporal graph learning is a promising solution for dynamic link prediction with text. However, it is a framework that is still relatively under-explored and heavily influenced by static graph learning. For example, the evaluation of the link prediction task on dynamic graphs involves random negative edge sampling which can lead to quite optimistic results. Further work is needed to construct "hard" negative sets, especially on Online Social Networks where edges that appeared in previous snapshots remain closed in the current time interval.

Future works will cope with the weaknesses of the temporal graph learning framework. In addition, we plan to propose a methodology that works with heterogeneous and dynamic graphs in inductive settings. Finally, we want to analyze how the decisions made by dynamic graph neural networks can be explained and if the model can also give us a better understanding of the evolution of the network being studied.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by MD, MZ, and SG. The first draft of the manuscript was written by MD and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by Università degli Studi di Milano within the CRUI-CARE Agreement. No funding received.

Availability of data and materials All information about the data are available here: https://github.com/manuel-dileo/dynamic-gnn. To guarantee the reproducibility of the results and the pseudo-anonymization, we release the input of the architecture in a tensor-based representation.

Code Availability Code of our ROLAND-based architecture is available here: https://github.com/manuel-dileo/dynamic-gnn

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethics approval The data collected are publicly available and accessible on the Steem blockchain. We followed well-established ethical procedures for social data and obtained a waiver from the ethics committee at the University of Milan. All data is pseudo-anonymized before usage, it is stored within a secure silo, while the text is not stored in a readable format but in a vector representation.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Ba, C. T., Michienzi, A., Guidi, B., Zignani, M., Ricci, L., & Gaito, S. (2022a). Fork-based user migration in blockchain online social media. In *14th ACM web science conference 2022*, (pp. 174–184).

Ba, C. T., Zignani, M., & Gaito, S. (2022b). The role of cryptocurrency in the dynamics of blockchain-based social networks: The case of steemit. *PloS one, 17*(6), e0267612.

Barracchia, E., Pio, G., Bifet, A., Gomes, H. M., Pfahringer, B., & Ceci, M. (2022). Lp-robin: Link prediction in dynamic networks exploiting incremental node embedding. Information Sciences *606*. https://doi.org/10.1016/j.ins.2022.05.079

Bruss, C. B., Khazane, A., Rider, J., Serpe, R. T., Gogoglou, A., & Hines, K. E. (2019). Deeptrax: Embedding graphs of financial transactions. In *2019 18th IEEE international conference on machine learning and applications* (ICMLA) (pp. 126–133).

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 workshop on deep learning, 2014*.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, T. Solorio (eds.) *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2019, Minneapolis, MN, USA*, June 2–7, 2019, (Long and Short Papers), (Vol. 1, pp. 4171–4186). Association for Computational Linguistics . https://doi.org/10.18653/v1/n19-1423.

Dileo, M., Ba, C. T., Zignani, M., & Gaito, S. (2022). Link prediction with text in online social networks: The role of textual content on high-resolution temporal data. In P. Pascal & D. Ienco (Eds.), *Discovery science* (pp. 212–226). Cham: Springer Nature Switzerland.

Fey, M., & Lenssen, J. E. (2019). *Fast graph representation learning with pytorch geometric*. arxiv:1903.02428

Garimella, K., Smith, T., Weiss, R., & West, R. (2021). Political polarization in online news consumption. *Proceedings of the International AAAI Conference on Web and Social Media, 15*(1), 152–162. https://doi.org/10.1609/icwsm.v15i1.18049

Guidi, B. (2021). An overview of blockchain online social media from the technical point of view. *Applied Sciences, 11*(21), 9880.

Gupta, S., & Bedathur, S. (2022). *A survey on temporal graph representation learning and generative modeling*. arxiv:2208.12126

Khanam, K. Z., Srivastava, G., & Mago, V. (2022). The homophily principle in social network analysis: A survey. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-021-11857-1

Kingma, D.P., & Ba, J. (2015). Adam: A method for stochastic optimization. In: Y. Bengio, Y. LeCun (eds.) *ICLR (Poster)*. http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#KingmaB14

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International conference on learning representations* (ICLR).

Kumar, A., Singh, S. S., Singh, K., & Biswas, B. (2020). Link prediction techniques, applications, and performance: A survey. *Physica A-statistical Mechanics and Its Applications, 553*, 124289.

Kumar, S., Hamilton, W. L., Leskovec, J., Jurafsky, D. (2018). Community interaction and conflict on the web. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, (pp. 933–943). International World Wide Web Conferences Steering Committee

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SJiHXGWAZ

Liben-Nowell, D., & Kleinberg, J. (2003). The link prediction problem for social networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management, CIKM '03*, (pp. 556–559). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/956863.956972.

Liu, P., Guarrasi, V., & Saryuce, A. (2023). Temporal network motifs: Models, limitations, evaluation. *IEEE Transactions on Knowledge & Data Engineering, 35*(01), 945–957. https://doi.org/10.1109/TKDE.2021.3077495

Liu, Q., Tang, S., Zhang, X., Zhao, X., Zhao, B. Y., & Zheng, H. (2016). Network growth and link prediction through an empirical lens. In *Proceedings of the 2016 Internet Measurement Conference*.

Longa, A., Lachi, V., Santin, G., Bianchini, M., Lepri, B., Lio, P., Scarselli, F., & Passerini, A. (2023).Graph neural networks for temporal graphs: State of the art, open challenges, and opportunities. arxiv:2302.01018

Monti, C., Rozza, A., Zappella, G., Zignani, M., Arvidsson, A., & Colleoni, E. (2013). Modelling political disaffection from twitter data. In *Proceedings of the second international workshop on issues of sentiment discovery and opinion mining, WISDOM '13*. Association for Computing Machinery, New York, NY, USA . https://doi.org/10.1145/2502069.2502072.

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., & Leiserson, C. (2020). Evolvegcn: Evolving graph convolutional networks for dynamic graphs. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(04), 5363–5370. https://doi.org/10.1609/aaai.v34i04.5984

Parimi, R., & Caragea, D. (2011). Predicting friendship links in social networks using a topic modeling approach. In *PAKDD*.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (EMNLP-IJCNLP), (pp. 3982–3992). Association for Computational Linguistics, Hong Kong, China . https://doi.org/10.18653/v1/D19-1410. https://aclanthology.org/D19-1410

Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., & Bronstein, M. (2020). Temporal graph networks for deep learning on dynamic graphs. In *ICML 2020 workshop on graph representation learning*.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE conference on computer vision and pattern recognition* (CVPR). IEEE. https://doi.org/10.1109/cvpr.2015.7298682. https://doi.org/10.1109%2Fcvpr.2015.7298682

Seo, Y., Defferrard, M., Vandergheynst, P., & Bresson, X. (2018). Structured sequence modeling with graph convolutional recurrent networks. In *Neural information processing: 25th international conference, ICONIP 2018, Siem Reap, Cambodia*, December 13–16, 2018, Proceedings, Part I 25, (pp. 362–373). Springer

Spillo, G., Musto, C., Polignano, M., Lops, P., de Gemmis, M., & Semeraro, G. (2023). Combining graph neural networks and sentence encoders for knowledge-aware recommendations. In *Proceedings of the 31st ACM conference on user modeling, adaptation and personalization, UMAP '23*, (pp. 1–12). Association for Computing Machinery, New York, NY, USA . https://doi.org/10.1145/3565472.3592965.

Villa, G., Pasi, G., & Viviani, M. (2021). Echo chamber detection and analysis: A topology- and content-based approach in the COVID-19 scenario. *Social Network Analysis and Mining, 11*(1), 78.

Wang, Y., Hooi, B., Liu, Y., Zhao, T., Guo, Z., & Shah, N. (2022). *Flashlight: Scalable link prediction with effective decoders*. arxiv:2209.10100

Wang, Z., Liang, J., & Li, R. (2018). Exploiting user-to-user topic inclusion degree for link prediction in social-information networks. *Expert Systems with Applications, 108*, 143–158.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems, 32*(1), 4–24. https://doi.org/10.1109/tnnls.2020.2978386

Xu, S., Yang, C., Shi, C., Fang, Y., Guo, Y., Yang, T., Zhang, L., & Hu, M. (2021). Topic-aware heterogeneous graph neural network for link prediction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management, CIKM '21*, (pp. 2261–2270). Association for Computing Machinery, New York, NY, USA . https://doi.org/10.1145/3459637.3482485.

Yang, Y., Lichtenwalter, R. N., & Chawla, N. V. (2014). Evaluating link prediction methods. *Knowledge and Information Systems, 45*(3), 751–782. https://doi.org/10.1007/s10115-014-0789-0

You, J., Du, T., & Leskovec, J. (2019). Roland: Graph learning framework for dynamic graphs. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, KDD '22*, (pp. 2358–2366). Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3534678.3539300. https://doi.org/10.1145/3534678.3539300

You, J., Wang, Y., Pal, A., & Eksombatchai, P., Rosenberg, C., & Leskovec, J. (2019). Hierarchical temporal convolutional networks for dynamic recommender systems. In L. Liu, R.W. White, A. Mantrach, F. Silvestri, J. J. McAuley, R. Baeza-Yates, L. Zia (eds.) *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA*, May 13–17, 2019, (pp. 2236–2246). ACM. https://doi.org/10.1145/3308558.3313747.

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th international joint conference on artificial intelligence, IJCAI'18*, (pp. 3634–3640). AAAI Press

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., & Li, H. (2020). T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems, 21*(9), 3848–3858. https://doi.org/10.1109/tits.2019.2935152

Authors and Affiliations

Manuel Dileo[1] · **Matteo Zignani**[1] · **Sabrina Gaito**[1]

✉ Manuel Dileo
manuel.dileo@unimi.it

Matteo Zignani
matteo.zignani@unimi.it

Sabrina Gaito
sabrina.gaito@unimi.it

[1] Department of Computer Science, University of Milan, Milan, Italy