



Supervised temporal link prediction in large-scale real-world networks

Gerrit Jan de Bruin¹ · Cor J. Veenman^{1,2} · H. Jaap van den Herik³ · Frank W. Takes¹

Received: 23 March 2021 / Revised: 21 June 2021 / Accepted: 29 July 2021 / Published online: 24 August 2021
© The Author(s) 2021

Abstract

Link prediction is a well-studied technique for inferring the missing edges between two nodes in some static representation of a network. In modern day social networks, the timestamps associated with each link can be used to predict *future links* between so-far unconnected nodes. In these so-called temporal networks, we speak of *temporal link prediction*. This paper presents a systematic investigation of supervised temporal link prediction on 26 temporal, structurally diverse, real-world networks ranging from thousands to a million nodes and links. We analyse the relation between global structural properties of each network and the obtained temporal link prediction performance, employing a set of well-established topological features commonly used in the link prediction literature. We report on four contributions. First, using temporal information, an improvement of prediction performance is observed. Second, our experiments show that degree disassortative networks perform better in temporal link prediction than assortative networks. Third, we present a new approach to investigate the distinction between networks modelling discrete events and networks modelling persistent relations. Unlike earlier work, our approach utilises information on *all* past events in a systematic way, resulting in substantially higher link prediction performance. Fourth, we report on the influence of the temporal activity of the node or the edge on the link prediction performance, and show that the performance differs depending on the considered network type. In the studied information networks, temporal information on the node appears most important. The findings in this paper demonstrate how link prediction can effectively be improved in temporal networks, explicitly taking into account the type of connectivity modelled by the temporal edge. More generally, the findings contribute to a better understanding of the mechanisms behind the evolution of networks.

Keywords Temporal link prediction · Supervised learning · Temporal networks · Network evolution · Multigraphs

1 Introduction and problem statement

Link prediction is a frequently employed method within the broader field of social network analysis (Barabási 2016). Many important real-world applications exist in a variety of domains. Two examples are the prediction of (1) missing links between pages of Wikipedia and (2) which users are likely to be friends on an online social network (Kumar et al. 2020). Link prediction is often defined as the task to predict missing links based on the currently observable

links in a network (Linyuan and Zhou 2011). Many real-world networks have temporal information on the times when the edges were created (Divakaran and Mohan 2020). Such *temporal networks* are also called dynamic or evolving networks. They open up the possibility of doing *temporal link prediction*. This means that they are able to infer future edges between two nodes as opposed to predicting only missing links (Liben-Nowell and Kleinberg 2007). For instance, in friendship networks, temporal link prediction might (1) facilitate friend recommendations, and (2) may actually predict which people will form new friendships in the future.

Existing work on temporal link prediction is typically performed on one or a handful of specific networks, making it difficult to assess the generalisability of the approaches used (Marjan et al. 2018). This paper presents the first large-scale empirical study of temporal link prediction on 26 different large-scale and structurally diverse temporal networks originating from various domains. In doing so, we provide a

✉ Gerrit Jan de Bruin
g.j.de.bruin@liacs.leidenuniv.nl

¹ Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands

² Data Science Department, TNO, The Hague, The Netherlands

³ Leiden Centre of Data Science, Leiden University, Leiden, The Netherlands

systematic investigation of *how* temporal information is best used in a temporal link prediction model.

This can be illustrated briefly by the social networks used in this study. They have a higher density than the other networks. This might improve performance in temporal link prediction, because the pairs of nodes in the networks used in our study are likely to have more common neighbours, providing more information to the supervised link prediction model. Thus, it is important to have an understanding of the relation between structural characteristics of the network and performance of topological features.

A common approach in temporal link prediction is to employ a supervised machine learning model that utilises multiple features to classify which links are missing or, in case of temporal link prediction, will appear in the future (de Bruin et al. 2021). Features are typically computed for every pair of nodes that is not (yet) connected, based on the topology of the network (Kumar et al. 2020). These topological features essentially calculate a similarity score for a node pair, where a higher similarity signals a higher likelihood that this pair of nodes should be connected. Commonly used topological features, both used in supervised and unsupervised learning, include Common Neighbours, Adamic-Adar, Jaccard Coefficient and Preferential Attachment (Sect. 4.1.1). These features clearly relate to the structural position of the nodes in the network. Previous work has suggested a straightforward approach to take the temporal evolution into account in these topological features (Tylenda et al. 2009; Bütün et al. 2018). We describe the process of obtaining the set of temporal topological features in Sect. 4.1.2. The benefits of using this set of features are that they are well-established and interpretable. Moreover, recent work has shown that in a supervised classifier these topological features perform as well as other types of features that are less interpretable and more complex (Ghasemian et al. 2020). A further comparison with other types of features is provided in Sect. 2.

In our work, we extend the set of state-of-the-art temporal topological features by considering that two types of temporal networks can be distinguished: networks with *persistent relationships* and networks with *discrete events* (O'Madadhain et al. 2005). The aforementioned example of friendship networks contains edges marking persistent relationships, that occur at most once for related persons. In case of communication networks, an edge usually marks a discrete event at an associated timestamp, representing a message sent from one person to another. In contrast to networks with persistent relationships, multiple edges can occur between two persons in discrete event networks. Previous studies have ignored that each link is not of the same type. In our approach, we address this gap in the literature by means of what we coin *past event aggregation*. This allows us to take both types of temporal links into account, where all

information of two-faceted past interactions (i.e. persistent and discrete) are incorporated into the temporal topological features.

Last but not least, the temporal topological features implicitly assume so-called edge-centred temporal behaviour, suggesting that phenomena at the level of links determine the evolution of the network. Here, we may challenge the usual assumption that the temporal aspect is merely caused by the activity of nodes, being the decision-making entities in the network, and operating somewhat independently of the structure of the remainder of the network (Hiraoka et al. 2020). To investigate whether this assumption holds, we present a comparison between (1) temporal topological features and (2) features consisting of static topological features along with features capturing temporal node activity. By testing this distinction on the 26 different temporal networks, we can better understand whether the temporal aspect is best captured by considering edge-centred or node-centred temporal information.

To sum up, the four contributions of this work are as follows. First, to the best of our knowledge, we are the first to present a large-scale empirical study of temporal link prediction on a variety of networks. In total, we assess the performance of a temporal link prediction model on 26 structurally diverse networks, varying in size from a few hundred to over a million nodes and edges. Second, we analyse possible relations between structural network properties and the observed performance in temporal link prediction. We find that networks with degree disassortativity, signalling frequent connections between nodes with different degrees, show better performance in temporal link prediction. Third, we propose to account for all past interactions in discrete event networks. Fourth, in an attempt to understand the relation between node-centred and edge-centred temporal behaviour, we find that the information networks used in this study stand out, as they appear to have more node-centred temporal behaviour.

This work is structured as follows. In Sect. 2, we further elaborate on related work. Section 3 provides the notation used in this work, leading up to the definition of temporal link prediction. After that, we continue with the approach in Sect. 4. This will be followed by a description of the temporal networks in Sect. 5. In Sect. 6 the four results of the experiments are presented and discussed. In Sect. 7, the conclusion is presented, together with suggestions for future work.

2 Related work

Although there is much literature available on link prediction, we found that attention for temporal networks and temporal link prediction is relatively limited. Some reviews have been published. They are pointing out the various

approaches that exists towards temporal link prediction (Dhote et al. 2013; Divakaran and Mohan 2020). Consequently, we will start with an exploration of four types of approaches presented therein.

First, probabilistic models require (1) additional node or edge attributes to obtain sufficient performance (which hinders a generic approach to all networks) or (2) techniques that do not scale to larger networks (Kumar et al. 2020) (rendering them unusable for the larger networks used in the study).

Second, approaches such as matrix factorisation, spectral clustering (Romero et al. 2020), and deep learning approaches, like DeepWalk (Perozzi et al. 2014) and Node2Vec (Grover and Leskovec 2016), all try to find a lower dimensional representation of the temporal network and use the obtained representation as a basis for link prediction. These approaches all learn a representation of the network without requiring explicit engineering of features. However, the downside is that the obtained features are hard to interpret, thereby making it difficult to explain why a certain link is predicted to appear. In applied scenarios, under some jurisdictions, this explanation can be required by law when an employed machine learning model affects people, which is often the case in for example the health and law enforcement domain (Holzinger et al. 2017). As an example, in previous work we examined driving patterns of trucks in a so-called truck co-driving network, where trucks are connected when they frequently drive together (de Bruin et al. 2020). When an inspection agency would use gathered network information to predict which trucks should be inspected for possible misconduct, truck drivers may legally have the right to know why they were selected. Since we aim to provide approaches towards temporal link prediction that are applicable to any scientific domain, we disregard approaches that learn a lower dimensional representation.

Third, in the time series forecasting approach, the temporal network is divided into multiple snapshots (Potgieter et al. 2007; Da Silva Soares and Prudencio 2012; Öczan and Ögüdücü 2015; Güneş et al. 2016; Öczan A, Ögüdücü 2017). For each of these snapshots, static topological features are learned. By using time series forecasting, the topological features of a future snapshot are learned, enabling link prediction. This approach does scale well to larger networks and is interpretable. However, it is unclear into how many snapshots the temporal network should be divided and whether the number of snapshots should remain constant across all networks used. Again, hindering a truly generic approach.

Finally, we focus on temporal topological features in this work (Tylenda et al. 2009; Bütün et al. 2016). Recent work has suggested that the use of topological features in supervised learning can outperform more complex features learned from a lower dimensional representation of the temporal network (Ghasemian et al. 2020). Section 4 provides

further details on this concept. These topological features are provided to a supervised link prediction classifier. Many different classification algorithms are known to work well in link prediction. Commonly used classifiers include logistic regression (Potgieter et al. 2007; O'Madadhain et al. 2005), support vector machines (Al Hasan et al. 2006; Öczan A, Ögüdücü 2017), k-nearest neighbours (Al Hasan et al. 2006; Bütün et al. 2018, 2016), and random forests (Öczan A, Ögüdücü 2017; Bütün et al. 2016, 2018; Ghasemian et al. 2020; de Bruin et al. 2021, 2020). We report performances using the logistic regression classifier. This classifier provides the following benefits, (1) it allows intuitive explanation on how each instance is classified (Bishop 2013), (2) the classifier is relatively simple and hence interpretable (Molnar 2020), (3) the classifier scales well to larger networks, and (4) good results are achieved without any parameter optimisation (O'Madadhain et al. 2005).

To sum up, in contrast to earlier works on temporal link prediction, which has been applied on only a handful networks (Bliss et al. 2014; Öczan and Ögüdücü 2015; Potgieter et al. 2007; Bütün et al. 2016, 2018; Da Silva Soares and Prudencio 2012; Güneş et al. 2016; Öczan A, Ögüdücü 2017; Tylenda et al. 2009; O'Madadhain et al. 2005; Soares and Prudêncio 2013; Muniz et al. 2018; Romero et al. 2020), we apply link prediction on a structurally diverse set of 26 large-scale, real-world networks. We aim to do so using a generic, scalable and interpretable approach.

3 Preliminaries

This section starts by describing the notation used throughout this paper in Sect. 3.1. In Sect. 3.2, we explain the various network properties and measures used in this work. Finally, in Sect. 3.3 we formally describe the temporal link prediction problem.

3.1 Notation

An undirected, temporal network $H_{[t', t'']}(V, E_H)$ consists of a set of nodes V and edges (or, equivalently, links) $E_H = \{(u, v, t) \mid u, v \in V \wedge t' \leq t \leq t''\}$ that occur between timestamps t' and t'' . Networks with discrete events, where multiple events can occur between two nodes, can be seen as a multigraph, where multi-edges exist: links between the same two nodes, but with different timestamps (Gross et al. 2013). In this work, removal of edges is not considered, since this information is not available for most temporal networks.

A static representation of the underlying network is needed for the comparison between static and temporal features (see Sect. 4). This static, simple graph $G_{[t', t'']}(V, E_G)$ with edges $(u, v) \in E_G$ is obtained from the

temporal network $H_{[t', t'']}$ by considering all edges that occur between t' and t'' , collapsing the multi-edges into a single edge. The number of nodes (also called the size) of the graph is $n = |V|$ and the number of edges is $m = |E_G|$. For convenience in later definitions, $\Gamma(u)$ is the set of all neighbours of node $u \in V$. The size of this set, i.e. $|\Gamma(u)|$, is the *degree* of node u .

3.2 Real-world network properties and their measures

Several properties exist that characterise the global structure of a network (Barabási 2016). These properties guide us in the exploration of how structure relates to the performance of a temporal link prediction algorithm. Below we discuss four of the main properties. Each of the properties is defined on the static underlying graph $G_{[t', t'']}$.

- **Density** The density of a network is calculated by dividing the number of edges by the total number of possible edges, i.e. $2m/n(n-1)$. For networks of the same size, higher density means that the average degree of nodes is higher, which has implications for the overall structural information available to the link prediction classifier.
- **Diameter** The diameter, sometimes called the maximum distance, is the largest distance observed between any pair of nodes. This property, together with density, captures how well-connected a network is.
- **Average clustering coefficient** The average clustering coefficient is the overall probability that two neighbours of a randomly selected node are linked to each other. The average clustering coefficient is given by

$$C = \frac{1}{n} \sum_{u \in G} \frac{2L_u}{|\Gamma(u)|(|\Gamma(u)| - 1)},$$

where L_u represents the number of edges between the neighbours of node u . In real-world networks, and in particular social networks, often highly clustered networks are observed.

- **Degree assortativity** It is often observed that nodes do not connect to random other nodes, but instead connect to nodes that are similar in some way. For instance, in social networks degree assortativity is observed, meaning that nodes often connect to other nodes with a similar degree. We can measure the degree assortativity of a network, by calculating the Pearson correlation coefficient, ρ , between the degree of nodes at both ends of all edges (Newman 2002). In case low degree nodes more frequently connect with high degree nodes, the obtained value is negative.

3.3 The goal of a supervised link prediction model

The goal of a supervised link prediction model is to predict for unconnected pairs of nodes in the temporal network $H_{[t_{q=0}, t_{q=s}]}$ whether they will connect in an evolved interval $[t_{q=s}, t_{q=1}]$ where q marks the q -th percentile of observed timestamps in the network and $0 < s < 1$. Hence, timestamps $t_{q=0}$ and $t_{q=1}$ mark the time associated to the first and last edge in the network, respectively. Moreover, timestamp $t_{q=s}$ marks the time used to split the network into two intervals. The examples provided to the supervised link prediction model are pairs of nodes that are not connected in $[t_{q=0}, t_{q=s}]$. For each example (u, v) in the dataset, a feature vector $x_{(u,v)}$ and binary label $y_{(u,v)}$ is provided to the supervised link prediction model. The label for each pair of nodes (u, v) is $y_{(u,v)} = 1$ when it will connect in $[t_{q=s}, t_{q=1}]$ and $y_{(u,v)} = 0$ otherwise. Because parameter s determines the number of considered nodes, it affects the class imbalance encountered in the supervised link prediction; values close to 1, results in a larger number of node pairs to consider, while limiting the number of positives.

The features used in the supervised link prediction model are only allowed to use information of network $H_{[t_{q=0}, t_{q=s}]}$, preventing any leakage from nodes that will connect in the evolved time interval $[t_{q=s}, t_{q=1}]$. Note that the temporal information contained in the network is used for two purposes; (1) it allows to split the network into two temporal intervals and (2) it is used in feature engineering to model temporal evolution.

4 Approach

This section explains the features used towards a supervised temporal link prediction. We start with an explanation of the different sets of features used in this study in Sect. 4.1. In particular, in step 2 of Sect. 4.1.2, we present a novel and intuitive approach to incorporate information on past interactions in the case of discrete event networks. In Sect. 4.2, we discuss the supervised link prediction model.

4.1 Features

In this subsection, we explain three types of features used. First, in Sect. 4.1.1 static topological features are provided. Second, temporal topological features are given in Sect. 4.1.2. The node activity features are specified in Sect. 4.1.3.

4.1.1 Static topological features

We use four common static topological features, which together form the feature vector for each candidate pair of nodes (u, v) . These features are computed on the static graph

G underlying the temporal network considered, as defined in Sect. 3.1. Below we define each of them.

Common Neighbours (CN) The CN feature is equal to the number of common neighbours of two nodes.

$$CN_{\text{static}}(u, v) = |\Gamma(u) \cap \Gamma(v)| \quad (1)$$

Adamic-Adar (AA) The AA feature considers all common neighbours, favouring nodes with low degrees (Adamic and Adar 2003).

$$AA_{\text{static}}(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log |\Gamma(z)|} \quad (2)$$

Jaccard Coefficient (JC) The JC feature is similar to the CN feature, but normalises for the number of unique neighbours of the two nodes.

$$JC_{\text{static}}(u, v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|} \quad (3)$$

Preferential Attachment (PA) The PA feature takes into account the observation that nodes with a high degree are more likely to make new links than nodes with a lower degree.

$$PA_{\text{static}}(u, v) = |\Gamma(u)| \cdot |\Gamma(v)| \quad (4)$$

4.1.2 Temporal topological features

Straightforward temporal extensions to topological features have been proposed in the literature (Tylenda et al. 2009; Bütün et al. 2018). Our method extends these approaches to also capture past interactions in case of aforementioned discrete event networks. The construction of these features then requires three steps, namely:

- A. Temporal weighting.
- B. The proposed approach of past event aggregation.
- C. Computation of weighted topological features.

The resulting feature vector for a given pair of nodes, after applying the three steps, consists of all possible combinations of 3 different temporal weighting functions (linear, exponential, square root), 8 different past event aggregations (see below under B) and 4 different weighted topological features (CN, AA, JC, PA). Thus, for discrete event networks the feature vector is of length $3 \cdot 8 \cdot 4 = 96$ and for networks with persistent relationships it is of length $3 \cdot 4 = 12$.

A: Temporal weighting The topological features need weighted edges (step C), while the networks used in this study have edges with an associated timestamp. In the temporal weighting step, we obtain these weights in a

procedure described by Tylenda et al. (2009). The definitions of the temporal weighting functions are provided in Eqs. (5)–(7). In these definitions, a numeric timestamp t is converted to a weight w . Note that t_{\min} and t_{\max} denote the earliest and latest observed timestamp over all edges of the considered network.

$$w_{\text{linear}} = l + (1 - l) \cdot \frac{t - t_{\min}}{t_{\max} - t_{\min}} \quad (5)$$

$$w_{\text{exponential}} = l + (1 - l) \cdot \frac{\exp\left(3 \frac{t - t_{\min}}{t_{\max} - t_{\min}}\right) - 1}{e^3 - 1} \quad (6)$$

$$w_{\text{square root}} = l + (1 - l) \cdot \sqrt{\frac{t - t_{\min}}{t_{\max} - t_{\min}}} \quad (7)$$

In Fig. 1 the behaviour of the different weighting functions is shown, when applied to the DBLP network (Ley 2002). It is further described in Sect. 5. The exponential weighting function (Eq. 6) assigns a higher weight to more recent edges than the linear (Eq. 5) and square root (Eq. 7) functions. In contrast, the square root function assigns higher weights to older edges in comparison to the linear and exponential functions. When weights of older edges become close to zero, these edges are discarded by the weighted topological features. To prevent that edges far in the past are discarded completely, we bound the output of each weighting function between a positive value l and 1.0 (l stands for lower bound).

B: Past event aggregation In case of networks with discrete events, each multi-edge has an associated weight after the previous temporal weighting step. To allow the weighted topological features to be computed, we

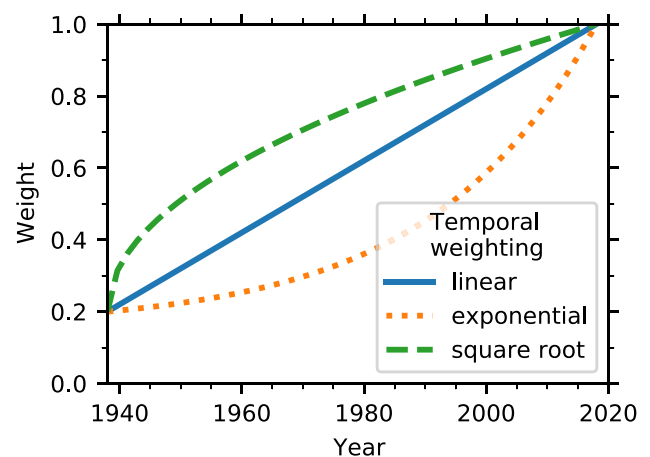


Fig. 1 The mapping of the three different weighting functions for the entire DBLP network

need to obtain a single weight for each node pair, capturing their past activity. Here we propose to aggregate all past events using eight different aggregation functions. All eight functions use as input a set containing all the weights of past events. The following functions are used: (1) the zeroth, (2) first, (3) second, (4) third, (5) fourth quantile and the (6) sum, (7) mean, and (8) variance of all past weights. By means of these summary statistics, we aim to capture the fact that depending on which network is considered, it may matter whether interaction took place very often, far away in the past, or very recent. These aggregation functions aim to capture different temporal behaviours. The quantile functions bin the set of weights, which is a common step in feature engineering. Taking the sum, mean, variance of the set of weights, allow the model to capture also more complex trends. An example of these complex trends, is the so-called bursty behaviour, which is often observed in real-world data (Barabási 2005).

C: *Weighted topological features* In Eqs. (8)–(11), the weighted topological features are presented, which are taken from Bütün et al. (2018). In these equations, $wtf(u, v)$ denotes the weight obtained for a given pair of nodes (u, v) after edges have been temporal weighted and, in case of networks with discrete events, events have been aggregated.

$$AA_{\text{temporal}}(u, v) = \sum_{z \in \Gamma(v) \cap \Gamma(u)} \frac{wtf(u, z) + wtf(v, z)}{\log(1 + \sum_{x \in \Gamma(z)} wtf(z, x))} \quad (8)$$

$$CN_{\text{temporal}}(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} wtf(u, z) + wtf(v, z) \quad (9)$$

$$JC_{\text{temporal}}(u, v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{wtf(u, z) + wtf(v, z)}{\sum_{x \in \Gamma(u)} wtf(u, x) + \sum_{y \in \Gamma(v)} wtf(v, y)} \quad (10)$$

$$PA_{\text{temporal}}(u, v) = \sum_{a \in \Gamma(x)} wtf(u, x) \cdot \sum_{b \in \Gamma(y)} wtf(v, y) \quad (11)$$

4.1.3 Node activity features

The goal of the node activity features is to capture node-centred temporal activity. To this end, we create the node activity features in the following three steps: (1) temporal weighting, (2) aggregation of node activity, and (3) combining node activity. These steps are explained below. The feature vector for a given pair of nodes consists of all combinations of three different temporal weighting functions, seven different aggregation functions applied to the node activity, and four different combinations of the node activity. This results in a feature vector of length $3 \cdot 7 \cdot 4 = 84$.

- (1) *Temporal weighting* The temporal weighing procedure is the same as used in feature engineering of the temporal weighted topological features (see Sect. 4.1.2).
- (2) *Aggregation of node activity* For each node, the set of weights from all edges adjacent to the node under investigation is collected. To obtain a fixed feature vector for each node, the set of weights is aggregated using the following functions: (1) the zeroth, (2) first, (3) second, (4) third, (5) fourth quantile and the (6) sum and (7) mean of the node activity vector (here the variance of all node weights is suppressed). Similar to the engineering of the temporal topological features, these aggregations are used to capture different kinds of activity that a node may exhibit. In particular, nodes are known to show bursty activity patterns in some networks (Hiraoka et al. 2020).
- (3) *Combining node activity* To take the activity, obtained in the previous two steps, of both nodes under consideration into account, we use four different combination functions. These four functions are the (1) sum, (2) absolute difference, (3) minimum, and (4) maximum. By doing this, we obtain the node activity feature vector.

4.2 Supervised link prediction

The features discussed in Sect. 4.1 serve as input for a supervised machine learning model that predicts whether or not a pair of currently disconnected nodes will connect in the future (see Sect. 3.3).

Here we use the logistic regression classifier. It was chosen because of its simplicity, overall good performance on this type of task and its explainability (see Sect. 2). We did not consider optimisation of any set of parameters, because that is considered outside the scope of the current paper.

In theory, a number quadratic in the number of nodes (i.e. the node pairs) could be selected as input for the classifier, with positive instances being node pairs that connect in the future, resulting in a significant class imbalance. To counter this problem and at the same time limit the computation time needed to train the model, we reduce the number of node pairs given as input to the classifier by the following two steps.

First, a well-known step is to select only pairs of nodes that are at a specific distance of each other (Lichtenwalter and Chawla 2012). Given the large sizes of networks used in this study, we limit the selection to only include pairs of nodes that are at distance two. Second, we sample with replacement from the remaining pairs of nodes a total of 10,000 that will connect (positives) and 10,000 that will not connect in the future (negatives). By doing this, we obtain a balanced set of examples which does not require any further post-processing, and can be used directly by

the classifier. The training set for the logistic regression classifier is obtained by means of stratified sampling, taking 75% of all examples. The remaining instances are used as a test set. Because we do not optimise any parameters of the logistic regression classifier, no validation set is used.

Analogously to previous work Divakaran and Mohan (2020) we measure the performance of the classifier on the test set by means of the Area Under the Receiver Operating Characteristic Curve (AUC). The AUC only considers the ranking of each score obtained for each pair of nodes provided to the logistic regression classifier. The AUC does not consider the absolute value of the score. This makes the measured performance robust to cases where the applied threshold on the scores is chosen poorly. An AUC of 0.5 signals random behaviour, i.e. no classifier performance at all. A perfect performance is obtained when the AUC is equal to 1, which is highly unlikely in practical settings.

5 Data

In this work, we use a structurally diverse and large collection of in total 26 temporal networks. The networks can be categorised into the three different domains, namely social, information, and technological networks. The distinction of networks in these three domains are taken from other network repositories. In Table 1 some common structural properties of these datasets are presented (see Sect. 3.2 for definitions). It is apparent from Fig. 2, which shows the relation between the number of nodes and edges for each of the 26 datasets, that the selected networks span a broad range in terms of size. Also, for each network it is indicated whether the edges marks persistent relationships or discrete events. In the latter case, the network has a multigraph structure, which requires preprocessing as discussed in Sect. 4.1.2. We observe seventeen networks showing degree disassortative behaviour, meaning that high degree nodes tend to connect to low degree nodes more frequently. The other nine networks show the opposite behaviour. We do not observe any significant relation between the domain of a network and its degree assortativity, or any other global property of the network.

A total of 21 networks were obtained from the Konect repository (Kunegis 2013), four networks from SNAP (Leskovec and Krevl 2014) and one from AMiner (Zhuang et al. 2013). The last column in Table 1 provides a reference to the work where each network is first introduced. Any directed network is converted to an undirected network by ignoring the directionality. In originally signed networks, we use only positive edges.

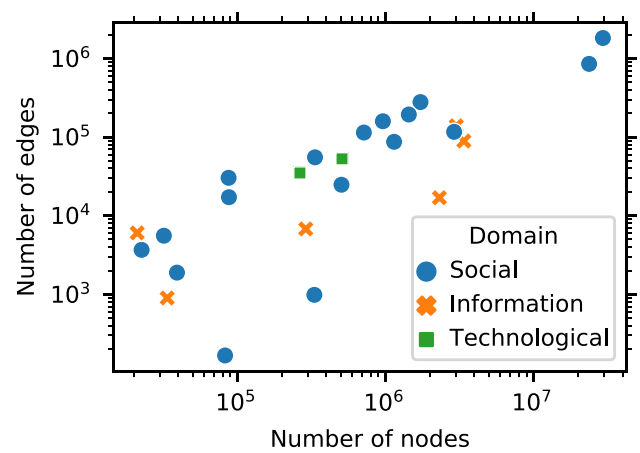


Fig. 2 The number of nodes and edges of the networks used in this paper. The horizontal and vertical axes have logarithmic scaling

6 Experiments

In Sect. 6.1 we start with the experimental setup. Then, the structure of this section follows the four contributions of this work. In Sect. 6.2 the performance of temporal link prediction on 26 networks is assessed. Section 6.3 continues with the analysis of the relation between structural network properties and the performance in temporal link prediction. In Sect. 6.4, we show the results of our methodological contribution to temporal link prediction in networks with discrete events. We finish in Sect. 6.5 with a comparison between node-centred and edge-centred temporal behaviour.

6.1 Experimental setup

In Sect. 3.3, the procedure to obtain examples and labels that serve as input for the classifier have been explained. In this procedure, we need to determine the value s for each network. Commonly, around two thirds of the edges are used for extraction of features (Lichtenwalter et al. 2010; Bütün et al. 2018, 2016; Al Hasan et al. 2006) and hence we choose $s = \frac{2}{3}$.

In the feature engineering of the temporal topological and node activity features, the first step is to temporally weight each edge. In Sect. 4.1.2, step 1, parameter l is introduced to prevent the discarding of old edges in the temporal weighting procedure. Based on earlier work (Tylenda et al. 2009), we set $l = 0.2$, giving a minimal weight to links far away in the past, while still sufficiently discounting these older links.

In this work, we use four sets of features. These feature sets, indicated by capital Roman numerals, are as follows.

- I Static topological (as defined in Sect. 4.1.1).
- II-A Temporal topological (as defined in Sect. 4.1.2).

Table 1 Networks used in this work

| Label | Domain | Edge type | Nodes | Edges | Density | D.a. | A.c.c. | Diam. | |
|-------|--------|------------|-----------|------------|-------------------|-------|--------|-------|-------------------------------|
| Rado | Social | Persistent | 167 | 82,927 | $2 \cdot 10^{-1}$ | 0.15 | 0.59 | 5 | Michalski et al. (2011) |
| UC | Inf. | Persistent | 899 | 33,720 | $2 \cdot 10^{-2}$ | 0.10 | 0.07 | 6 | Opsahl (2013) |
| EU | Social | Persistent | 986 | 332,334 | $3 \cdot 10^{-2}$ | 0.05 | 0.41 | 7 | Yin et al. (2017) |
| Dem | Social | Persistent | 1891 | 39,264 | $2 \cdot 10^{-3}$ | -0.15 | 0.21 | 8 | Wikileaks (2016) |
| bitA | Social | Event | 3683 | 22,650 | $2 \cdot 10^{-3}$ | -0.15 | 0.17 | 10 | Kumar et al. (2017) |
| bitOT | Social | Event | 5573 | 32,029 | $1 \cdot 10^{-3}$ | -0.15 | 0.16 | 14 | Kumar et al. (2017) |
| chess | Inf. | Event | 6050 | 21,163 | $1 \cdot 10^{-3}$ | 0.36 | 0.05 | 13 | Kunegis (2013) |
| HepTh | Inf. | Persistent | 6798 | 290,597 | $9 \cdot 10^{-3}$ | 0.08 | 0.77 | 11 | Leskovec et al. (2007) |
| HepPh | Inf. | Persistent | 16,959 | 2,322,259 | $8 \cdot 10^{-3}$ | 0.17 | 0.61 | 8 | Leskovec et al. (2007) |
| Condm | Social | Persistent | 17,218 | 88,090 | $4 \cdot 10^{-4}$ | 0.29 | 0.64 | 19 | Lichtenwalter et al. (2010) |
| SX-MO | Social | Persistent | 24,818 | 506,550 | $6 \cdot 10^{-4}$ | -0.05 | 0.31 | 9 | Paranjape et al. (2017) |
| D-rep | Social | Event | 30,398 | 87,627 | $2 \cdot 10^{-4}$ | 0.02 | 0.01 | 12 | De Choudhury et al. (2009) |
| Rbody | Tech. | Persistent | 35,010 | 265,491 | $2 \cdot 10^{-4}$ | 0.03 | 0.18 | 11 | Kumar et al. (2018) |
| Rtit | Tech. | Persistent | 53,018 | 510,787 | $1 \cdot 10^{-4}$ | -0.01 | 0.18 | 17 | Kumar et al. (2018) |
| FB-w | Social | Event | 55,387 | 335,708 | $2 \cdot 10^{-4}$ | -0.02 | 0.12 | 16 | Viswanath et al. (2009) |
| FB-l | Social | Event | 55,387 | 335,708 | $2 \cdot 10^{-4}$ | 0.22 | 0.12 | 16 | Viswanath et al. (2009) |
| Enron | Social | Persistent | 87,273 | 1,148,072 | $8 \cdot 10^{-5}$ | 0.22 | 0.12 | 14 | Klimt and Yang (2004) |
| loans | Inf. | Event | 89,269 | 3,394,979 | $8 \cdot 10^{-4}$ | -0.04 | 0.00 | 8 | Redmond and Cunningham (2013) |
| trust | Social | Event | 114,467 | 717,667 | $9 \cdot 10^{-5}$ | -0.07 | 0.13 | 14 | Richardson et al. (2003) |
| Wiki | Social | Persistent | 116,836 | 2,917,785 | $3 \cdot 10^{-4}$ | -0.06 | 0.36 | 10 | Brandes et al. (2009) |
| D-v | Inf. | Event | 139,409 | 3,018,197 | $3 \cdot 10^{-4}$ | -0.21 | 0.14 | 4 | Hogg and Lerman (2012) |
| SX-AU | Social | Persistent | 159,316 | 964,437 | $4 \cdot 10^{-5}$ | -0.10 | 0.11 | 13 | Paranjape et al. (2017) |
| SX-SU | Social | Persistent | 194,085 | 1,443,339 | $4 \cdot 10^{-5}$ | -0.08 | 0.12 | 12 | Paranjape et al. (2017) |
| D-f | Social | Event | 279,374 | 1,729,983 | $4 \cdot 10^{-5}$ | -0.05 | 0.09 | 18 | Hogg and Lerman (2012) |
| AMin | Social | Persistent | 855,165 | 23,787,273 | $9 \cdot 10^{-6}$ | 0.16 | 0.61 | 22 | Zhuang et al. (2013) |
| DBLP | Social | Persistent | 1,824,701 | 29,487,744 | $5 \cdot 10^{-6}$ | 0.15 | 0.63 | 23 | Ley (2002) |

The following abbreviations are used in the columns; *D.a.* degree assortativity, *A.c.c.* average clustering coefficient, *Diam.* diameter. In the column 'domain', Technological is abbreviated to Tech. and Information to Inf

II-B Temporal topological without past event aggregation (as defined in Sect. 4.1.2, skipping step 2 and using only the last occurring event).

III Static topological + node activity (Sect. 4.1.2 + Sect. 4.1.3).

It is common practice to standardise features by subtracting the mean and scaling the variance to unit. The logistic regression classifier provided in the Python scikit-learn package (Pedregosa et al. 2011) is used. Although the goal of this paper is not to extensively compare machine learning classifiers, in Appendix A2, results on the performance in terms of AUC obtained using two other commonly used classifiers, being random forests (Pedregosa et al. 2011) and XGBoost (Chen and Guestrin 2016) are presented. For almost all datasets, similar relative performance is observed. The code used in this research, is available at <http://github.com/gerritjandebuyn/snarn2021-code>. It uses the Python language and the packages NetworkX (Hagberg et al. 2008)

for network analysis, scikit-learn (Pedregosa et al. 2011) for the machine learning pipeline, and the Scipy ecosystem (Virtanen et al. 2020) for some of the feature engineering and statistical tests. The C++ library teexGraph (Takes and Kusters 2011) was used to determine the diameter of each network. The package versions, as well as all dependencies, can be found in the aforementioned repository.

6.2 Improvement of prediction performance with temporal information

We examine whether temporal information improves the overall prediction performance. Baseline performance is obtained by ignoring temporal information, using only static topological features (feature set I). In contrast, temporal topological features (feature set II-A) are used to obtain the performance of link prediction utilising temporal information.

The results of this comparison are presented in Table 2 and in Fig. 3. They clearly indicate that using temporal

information improves the prediction performance of new links, i.e. performance reported in column ‘II-A’ is always higher than that in ‘I’. So, every single network shows better performance when temporal topological features are used. The average improvement in performance is 0.07 ± 0.04 (\pm standard deviation). For some networks, performance improves considerably more when temporal information is used in prediction. For example, the loans network has a mediocre baseline performance of 0.79, but a high performance of 0.95 is observed when temporal information is employed. This improvement in performance can be related to the structure of the network. Hence, in the next section the relation between the structural properties of networks and the performance in temporal link prediction is explored.

6.3 Structural network properties and link prediction performance

In this section, we examine which structural properties are associated to high link prediction performance. In Fig. 4, the Pearson correlations between the performance in (temporal) link prediction and various structural network properties (see

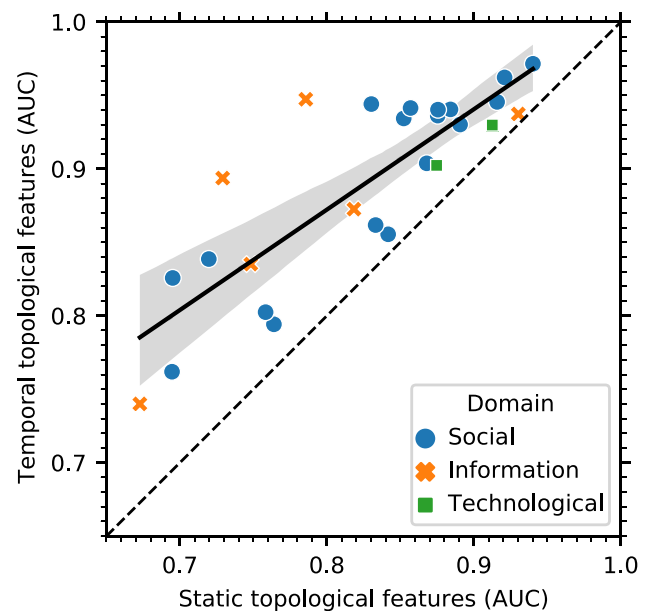


Fig. 3 Performance of the link prediction classifier for the 26 different networks

Table 2 Performance obtained in (temporal) link prediction, using the following sets of features; static topological (I), temporal topological with (II-A) and without (II-B) past event aggregation, and static topological + node activity (III)

| Label | Domain | Edge type | Nodes (<i>n</i>) | AUC | | | |
|-------|---------------|-----------|--------------------|-------|-------|-------|-------|
| | | | | I | II-A | II-B | III |
| Rado | Social | Multi | 167 | 0.864 | 0.921 | 0.852 | 0.902 |
| UC | Information | Multi | 899 | 0.731 | 0.893 | 0.744 | 0.873 |
| EU | Social | Multi | 986 | 0.839 | 0.873 | 0.811 | 0.849 |
| Dem | Social | Multi | 1891 | 0.920 | 0.944 | 0.919 | 0.938 |
| bitA | Social | Simple | 3683 | 0.868 | 0.945 | 0.945 | 0.940 |
| bitOT | Social | Simple | 5573 | 0.821 | 0.947 | 0.947 | 0.939 |
| chess | Information | Simple | 6050 | 0.665 | 0.735 | 0.735 | 0.736 |
| HepTh | Information | Multi | 6798 | 0.757 | 0.835 | 0.776 | 0.819 |
| HepPh | Information | Multi | 16,959 | 0.828 | 0.879 | 0.834 | 0.868 |
| Condm | Social | Multi | 17,218 | 0.688 | 0.760 | 0.706 | 0.728 |
| SX-MO | Social | Multi | 24,818 | 0.859 | 0.944 | 0.909 | 0.933 |
| D-rep | Social | Simple | 30,398 | 0.837 | 0.866 | 0.866 | 0.865 |
| Rbody | Technological | Multi | 35,010 | 0.880 | 0.905 | 0.854 | 0.890 |
| Rtit | Technological | Multi | 53,018 | 0.903 | 0.931 | 0.906 | 0.925 |
| FB-w | Social | Simple | 55,387 | 0.762 | 0.809 | 0.809 | 0.788 |
| FB-l | Social | Simple | 55,387 | 0.762 | 0.803 | 0.803 | 0.775 |
| Enron | Social | Multi | 87,273 | 0.847 | 0.912 | 0.873 | 0.909 |
| loans | Information | Simple | 89,269 | 0.786 | 0.947 | 0.947 | 0.946 |
| trust | Social | Simple | 114,467 | 0.889 | 0.936 | 0.936 | 0.937 |
| Wiki | Social | Multi | 116,836 | 0.864 | 0.936 | 0.896 | 0.939 |
| D-v | Information | Simple | 139,409 | 0.933 | 0.941 | 0.941 | 0.939 |
| SX-AU | Social | Multi | 159,316 | 0.937 | 0.970 | 0.959 | 0.970 |
| SX-SU | Social | Multi | 194,085 | 0.921 | 0.965 | 0.946 | 0.961 |
| D-f | Social | Simple | 279,374 | 0.891 | 0.926 | 0.926 | 0.924 |
| AMin | Social | Multi | 855,165 | 0.725 | 0.849 | 0.804 | 0.816 |
| DBLP | Social | Multi | 1,824,701 | 0.704 | 0.826 | 0.743 | 0.786 |

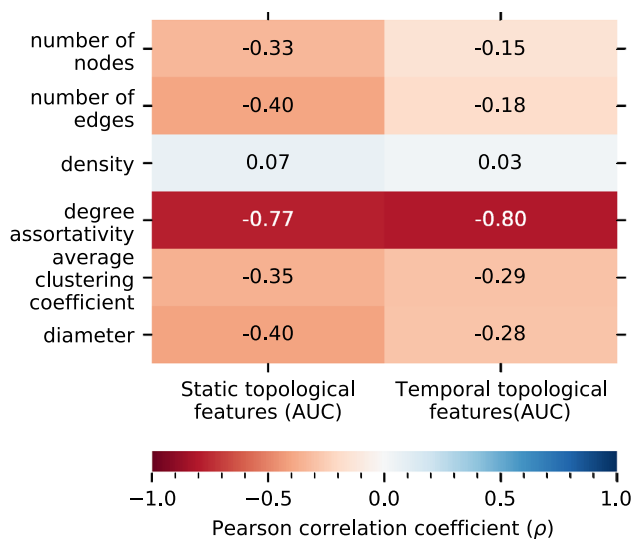


Fig. 4 Correlations between network properties and the performance in a supervised classifier learned only with static topological features (feature set I) and with temporal topological features (feature set II-A)

Sect. 3.2) are presented. While most properties show at best modest correlation with the link prediction performance, we observe a significant negative correlation between the degree assortativity of a network and the prediction performance of new links using static topological features ($p = 3 \cdot 10^{-6}$) and temporal topological features ($p = 5 \cdot 10^{-7}$). This means that strong disassortative behaviour in networks, where nodes of low degree are more likely to connect with nodes of high degree, show better performance in link prediction. The relation between degree assortativity and the link prediction performance is shown in more detail in Fig. 5. The observed negative correlation might be explained as follows. In real-world networks, low degree nodes typically largely outnumber the high degree nodes. However, nodes with a degree that by far exceeds the average degree, so-called hubs, are also relatively often observed in real-world networks (Barabási 2016). In degree disassortative networks, the numerous low degree nodes by definition connect more frequently with hubs than with other low degree nodes. For these low degree nodes, the preferential attachment feature will provide higher scores for candidate nodes having a high degree. Therefore, the supervised model can use this information in a straightforward manner to obtain a better performance.

To confirm the relation between the degree assortativity and temporal link prediction performance of a network, we conducted additional experiments. By performing assortative and disassortative degree-preserving rewiring, we further substantiate the claim that disassortative networks indeed show higher link prediction performance. Detailed results can be found in Appendix A1.

In Fig. 5 we observe that the temporal topological features show an even stronger correlation ($\rho = -0.82$) than the static

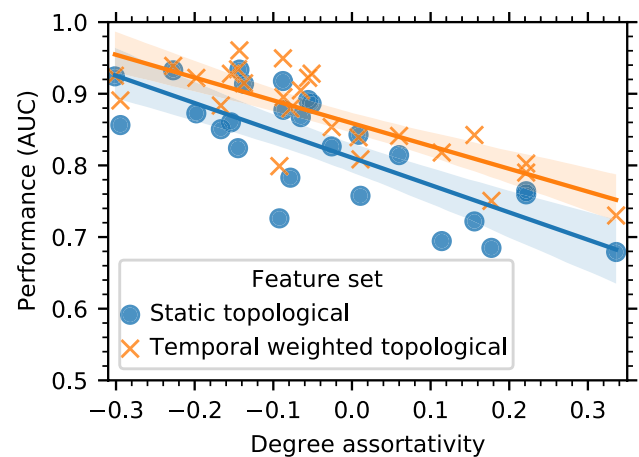


Fig. 5 Degree assortativity and performance in a supervised classifier for each network, learned only with static topological features (feature set I) and temporal topological features (feature set II-A). The lines indicate the relation between the degree assortativity of the network and the performance of the classifier

topological features ($\rho = -0.78$). A possible explanation is that the temporal features are able to determine with higher accuracy which nodes will grow to active hubs, linking to many low degree nodes, whereas this information would be lost in a static network representation. This observation provides additional evidence that the temporal topological features are likely capturing relevant temporal behaviour.

6.4 Enhancement of performance with past event aggregation

To assess how networks with discrete events should be dealt with in temporal link prediction, we use two different sets of features. The first set of features (II-A) is constructed with past event aggregation, which allows to make fully use of information contained in all discrete events. The second set of features (II-B) considers only the last occurring edge between two nodes, thereby ignoring any past events. For networks with persistent edges, the two sets of features yield the same results, because the networks do not contain past events. The performance obtained with these two different sets of features is reported in Table 2. In Fig. 6, we show the difference between the two performances of the networks with discrete events in more detail. From this figure, we learn that these networks all show better performance when past events are aggregated using the various aggregation functions. This result is interesting more broadly for link prediction research, as the derived feature modification steps can be inserted into any topological network feature aiming to capture the similarity of nodes in an attempt to predict their future connectivity.

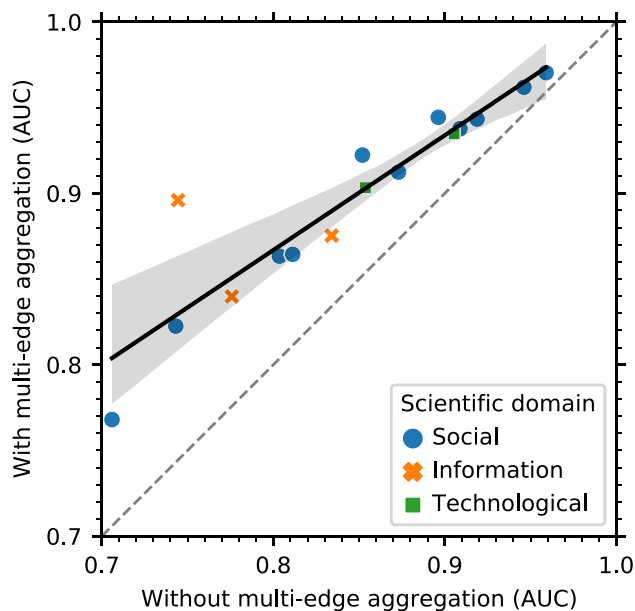


Fig. 6 Performance obtained in temporal link prediction, using temporal topological features without (feature set II-B, x-axis) and with (feature set II-A, y-axis) past event aggregation

Interestingly, when looking in more detail at the performance improvement by past event aggregation for each discrete event network, we observe large differences. On the one hand, we observe networks with only minor improvement when past events are aggregated. For example, the Condense matter (scientific) collaboration network (Condm.) shows only a minor improvement of 0.706–0.760 AUC. A possible explanation is that temporal information of discrete events has only limited use, since it takes time to come to a successful collaboration. On the other hand, the UC Irvine message network (UC), shows a major improvement in AUC from 0.744 to 0.893. This might be caused by the more variable nature of messages, which takes only a short time to establish. In that case, the feature set with past event aggregation might provide higher scores to pairs of nodes that are both actively messaging.

6.5 Comparison of node- and edge-centred temporal link prediction

In the experiments performed so far, we used temporal topological features to assess temporal link prediction performance. These features assume edge-centred temporal behaviour. In the experiments below, we compare the performance of the edge-centred features (feature set II-A) with features that assume node-centred temporal behaviour (feature set III). The results of both feature sets are presented in Table 2 and in more detail in Fig. 7. We observe a strong correlation ($\rho = 0.92$, $p = 0.009$) between the obtained performances

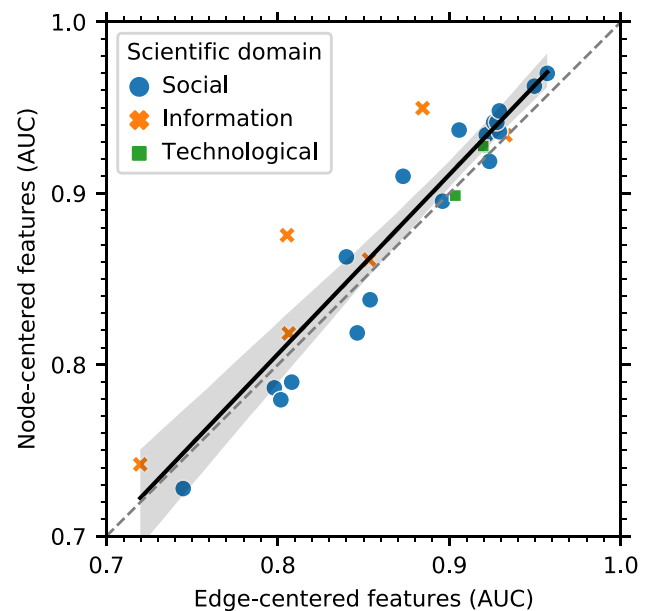


Fig. 7 Performance obtained in temporal link prediction, using node-centred features (feature set III, x-axis) and edge-centred features (feature set II, y-axis)

using the two sets of features on all 26 networks. This finding suggests that the temporal aspect of most networks can be modelled by using either node-centred or edge-centred temporal features.

However, for the four information networks the performance of the node-centred features seems to be higher than the edge-centred features. This finding hints that in information networks temporal behaviour may be node-centred. Given the low number of information networks available in this study, further research should be conducted to a larger set of information networks to verify this finding.

A note of caution is due here since we analyse the temporal link performance only on pairs of nodes at a distance of two; different findings may be observed whether the findings still hold when more global features of node similarity are used. Notwithstanding this limitation, the study shows that both node- and edge-centred features in supervised temporal link predictions are able to achieve a high performance.

7 Conclusion and outlook

In this paper, the aim was to perform a large-scale empirical study of temporal link prediction, using a wide variety of structurally diverse networks. Moreover, we aimed to demonstrate the benefit of past event aggregation, allowing to take the rich interaction history of nodes into account in predicting their future linking activity. This study resulted in four findings.

Table 3 Assortativity of all networks after rewiring, for disassortative rewiring (up to -100%) and assortative rewiring (up to 100%) for each of the network datasets in Table 1

| Label | -100% | -80% | -60% | -40% | -20% | 0% | 20% | 40% | 60% | 80% | 100% |
|-------|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Rado | 0.01 | 0.01 | 0.07 | 0.09 | -0.00 | 0.15 | 0.14 | 0.18 | 0.16 | 0.09 | 0.19 |
| UC | -0.05 | -0.03 | -0.02 | 0.01 | 0.06 | 0.10 | 0.14 | 0.17 | 0.18 | 0.21 | 0.23 |
| EU | 0.23 | 0.16 | 0.36 | 0.34 | 0.15 | 0.05 | 0.12 | 0.11 | 0.10 | -0.18 | -0.11 |
| Dem | -0.21 | -0.21 | -0.16 | -0.14 | -0.14 | -0.15 | -0.06 | -0.00 | 0.06 | 0.09 | 0.13 |
| bitA | -0.25 | -0.24 | -0.22 | -0.19 | -0.17 | -0.15 | -0.10 | -0.04 | 0.01 | 0.10 | 0.22 |
| bitOT | -0.23 | -0.22 | -0.20 | -0.17 | -0.16 | -0.15 | -0.11 | -0.07 | -0.02 | 0.04 | 0.14 |
| chess | -0.17 | -0.14 | -0.05 | 0.04 | 0.18 | 0.36 | 0.52 | 0.62 | 0.69 | 0.74 | 0.78 |
| HepTh | -0.18 | -0.13 | -0.08 | -0.03 | 0.03 | 0.08 | 0.18 | 0.31 | 0.46 | 0.57 | 0.61 |
| HepPh | -0.11 | -0.07 | -0.02 | 0.04 | 0.10 | 0.17 | 0.26 | 0.35 | 0.43 | 0.48 | 0.52 |
| Condm | -0.04 | 0.00 | 0.05 | 0.11 | 0.20 | 0.29 | 0.42 | 0.53 | 0.59 | 0.62 | 0.63 |
| SX-MO | -0.24 | -0.21 | -0.17 | -0.13 | -0.09 | -0.05 | 0.02 | 0.09 | 0.16 | 0.22 | 0.29 |
| D-rep | -0.19 | -0.16 | -0.12 | -0.08 | -0.04 | 0.02 | 0.13 | 0.29 | 0.46 | 0.56 | 0.64 |
| Rbody | -0.11 | -0.09 | -0.06 | -0.03 | 0.00 | 0.03 | 0.07 | 0.10 | 0.12 | 0.13 | 0.15 |
| Rtit | -0.11 | -0.09 | -0.07 | -0.05 | -0.04 | -0.02 | 0.04 | 0.09 | 0.14 | 0.14 | 0.13 |
| FB-w | -0.12 | -0.09 | -0.06 | -0.00 | 0.08 | 0.22 | 0.43 | 0.61 | 0.71 | 0.77 | 0.81 |
| FB-l | -0.12 | -0.09 | -0.06 | -0.00 | 0.08 | 0.22 | 0.43 | 0.61 | 0.71 | 0.77 | 0.81 |
| Enron | -0.14 | -0.11 | -0.09 | -0.07 | -0.05 | -0.04 | 0.01 | 0.03 | 0.06 | 0.08 | 0.09 |
| loans | -0.20 | -0.17 | -0.14 | -0.12 | -0.09 | -0.07 | -0.02 | 0.06 | 0.22 | 0.47 | 0.61 |
| trust | -0.26 | -0.23 | -0.19 | -0.14 | -0.09 | -0.01 | 0.13 | 0.33 | 0.52 | 0.64 | 0.70 |
| Wiki | -0.08 | -0.08 | -0.07 | -0.06 | -0.06 | -0.06 | -0.04 | -0.03 | -0.02 | -0.01 | 0.00 |
| D-v | -0.27 | -0.26 | -0.24 | -0.23 | -0.21 | -0.21 | -0.20 | -0.16 | -0.06 | 0.13 | 0.31 |
| SX-AU | -0.25 | -0.22 | -0.20 | -0.17 | -0.13 | -0.10 | -0.06 | -0.01 | 0.03 | 0.08 | 0.13 |
| SX-SU | -0.16 | -0.15 | -0.13 | -0.11 | -0.10 | -0.08 | -0.05 | -0.03 | -0.00 | 0.03 | 0.07 |
| D-f | -0.13 | -0.12 | -0.10 | -0.09 | -0.07 | -0.05 | 0.02 | 0.18 | 0.47 | 0.64 | 0.71 |
| AMin | 0.01 | 0.03 | 0.05 | 0.07 | 0.11 | 0.16 | 0.21 | 0.24 | 0.28 | 0.30 | 0.33 |
| DBLP | 0.01 | 0.03 | 0.05 | 0.07 | 0.11 | 0.15 | 0.21 | 0.26 | 0.30 | 0.33 | 0.36 |

First, performance in supervised temporal link prediction is consistently higher when temporal information is taken into account. Second, the performance in temporal link prediction appears related to the global structure of the network. Most notably, degree disassortative networks perform better than degree assortative networks. Third, the newly proposed method of past event aggregation, is able to better model link formation in networks with discrete events. It substantially increases the performance of temporal link prediction. The derived feature modification steps can be inserted into any topological feature, potentially improving the performance of any supervised (temporal) link prediction endeavour. Fourth, we showed that in four information networks, features capturing node activity, together with static topological features, outperform features that consider edge-centred temporal information, suggesting that the temporal mechanisms in these networks reside with the nodes.

A natural next step of this work is to analyse even bigger temporal networks, or networks originating from different domains. It appears that publicly available networks from other domains, such as biological, economic and transportation networks, typically do not contain temporal information (Ghasemian et al. 2020). However, it would be interesting

to investigate whether findings presented in this paper also hold for these types of networks. In addition, it is evident that there is an advantage to taking temporal information into account when performing supervised link prediction on temporal networks. It could be interesting to see whether such temporal information also benefits prediction performance in other machine learning tasks on networks, such as node classification (Hamilton et al. 2017).

Appendix

Appendix A1: Relation degree assortativity and temporal link prediction performance

To further assess the relation between degree assortativity and temporal link prediction performance, as derived from the empirical results in Sect. 6.3, we conducted additional experiments. By means of simulation, we modified a number of network datasets from Table 1 using assortative and disassortative degree-preserving rewiring, following an approach similar to the one proposed in Van Mieghem et al. (2010). In particular, we aim to retain the local clustering properties

Table 4 Performance (in AUC) for the rewired networks as reported on in Table 3

| Label | −100% | −80% | −60% | −40% | −20% | 20% | 40% | 60% | 80% | 100% |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Rado | −0.074 | −0.107 | −0.106 | −0.096 | −0.103 | −0.131 | −0.126 | −0.136 | −0.138 | 0.024 |
| UC | −0.311 | −0.266 | −0.270 | −0.356 | −0.297 | −0.312 | −0.388 | −0.373 | −0.303 | −0.083 |
| EU | −0.061 | −0.119 | −0.088 | −0.084 | −0.074 | −0.070 | −0.106 | −0.067 | −0.107 | −0.109 |
| Dem | −0.152 | −0.162 | −0.134 | −0.171 | −0.105 | −0.130 | −0.124 | −0.123 | −0.169 | −0.021 |
| bitA | −0.259 | −0.243 | −0.267 | −0.280 | −0.245 | −0.309 | −0.373 | −0.413 | −0.390 | −0.052 |
| bitOT | −0.252 | −0.263 | −0.264 | −0.308 | −0.325 | −0.376 | −0.395 | −0.353 | −0.371 | −0.014 |
| chess | −0.317 | −0.349 | −0.368 | −0.377 | −0.410 | −0.406 | −0.403 | −0.281 | −0.382 | 0.036 |
| HepTh | −0.142 | −0.189 | −0.202 | −0.234 | −0.276 | −0.248 | −0.249 | −0.220 | −0.177 | −0.020 |
| HepPh | −0.162 | −0.193 | −0.208 | −0.213 | −0.226 | −0.234 | −0.201 | −0.177 | −0.137 | −0.034 |
| Condm | −0.243 | −0.252 | −0.269 | −0.294 | −0.344 | −0.273 | −0.263 | −0.252 | −0.243 | −0.095 |
| SX-MO | −0.161 | −0.167 | −0.179 | −0.187 | −0.178 | −0.205 | −0.212 | −0.194 | −0.200 | −0.015 |
| D-rep | −0.416 | −0.445 | −0.506 | −0.586 | −0.332 | −0.233 | −0.202 | −0.187 | −0.167 | −0.006 |
| Rbody | −0.162 | −0.169 | −0.187 | −0.178 | −0.182 | −0.219 | −0.220 | −0.243 | −0.248 | 0.015 |
| Rtit | −0.136 | −0.124 | −0.132 | −0.144 | −0.132 | −0.156 | −0.191 | −0.198 | −0.188 | 0.031 |
| FB-w | −0.240 | −0.250 | −0.239 | −0.239 | −0.242 | −0.251 | −0.273 | −0.291 | −0.326 | 0.084 |
| FB-l | −0.246 | −0.253 | −0.257 | −0.244 | −0.232 | −0.236 | −0.266 | −0.291 | −0.325 | 0.096 |
| Enron | −0.165 | −0.171 | −0.177 | −0.191 | −0.188 | −0.211 | −0.214 | −0.228 | −0.200 | 0.004 |
| loans | −0.347 | −0.413 | −0.459 | −0.333 | −0.300 | −0.230 | −0.215 | −0.229 | −0.265 | −0.024 |
| trust | −0.198 | −0.215 | −0.216 | −0.253 | −0.246 | −0.300 | −0.301 | −0.264 | −0.205 | 0.012 |
| Wiki | −0.003 | −0.211 | −0.218 | −0.243 | −0.296 | −0.446 | −0.407 | −0.378 | −0.336 | −0.029 |
| D-v | 0.097 | −0.011 | −0.019 | −0.044 | −0.073 | −0.077 | −0.047 | −0.047 | −0.043 | 0.017 |
| SX-AU | −0.276 | −0.281 | −0.279 | −0.280 | −0.287 | −0.408 | −0.440 | −0.445 | −0.468 | −0.005 |
| SX-SU | −0.244 | −0.265 | −0.272 | −0.309 | −0.302 | −0.389 | −0.397 | −0.408 | −0.392 | −0.002 |
| D-f | −0.170 | −0.202 | −0.227 | −0.263 | −0.292 | −0.325 | −0.295 | −0.278 | −0.213 | 0.012 |
| AMin | −0.278 | −0.292 | −0.292 | −0.310 | −0.320 | −0.385 | −0.337 | −0.375 | −0.372 | −0.095 |
| DBLP | −0.335 | −0.331 | −0.330 | −0.358 | −0.361 | −0.431 | −0.357 | −0.443 | −0.427 | −0.046 |
| mean | −0.202 | −0.229 | −0.237 | −0.253 | −0.245 | −0.269 | −0.269 | −0.265 | −0.261 | −0.012 |

by not selecting two edges at random, but rather selection two edges that are close to each other, ensuring that not too many triangles and therewith clustering is destructed, as this is a determining feature in link prediction.

The procedure, which we repeat for a certain number of times (explained below), consists of the following five steps. First, an edge (u, v) is randomly selected. Second, we randomly select a node x from the neighbourhood of u . Third, we sample a node y that is connected to x , but not to u or v . At this time, pairs of nodes (u, v) and (x, y) are connected while the link (v, y) is absent. The fourth step is to determine from the pairs of nodes (u, v) , (v, y) and (x, y) which node pair has a maximum difference in degree. Step five is the actual rewiring of edges. There can be three outcomes from step 4, (a) node pair (v, y) has the maximum difference in degree and there is no gain in assortativity by rewiring any edges, (b) node pair (u, v) has the maximum difference in degree and by moving all edges (recall, there can be multiple links between two nodes) between (u, v) to (v, y) the assortativity is increased, and (c) node pair (x, y) has the maximum difference in degree and by moving all edges between (x, y) to (v, y) the assortativity is increased. In

case we want to perform disassortative degree-preserving rewiring, we consider in step four and five the node pair with the lowest difference in degree. The five steps are repeated, with increments of $0.2m$ from the original network up to m of the edges that gets a chance to rewire.

The degree assortativity values of the rewired networks can be found in Table 3. We observe that for degree disassortative rewiring, a larger performance is attained than for assortative rewiring, strengthening our result from Sect. 6.3. This finding is further explored in Table 4, in which we list the percentual increase in performance for both disassortativity and assortativity rewired datasets. In all cases, we observe higher performance for disassortativity rewired networks.

Appendix A2: Choice of classifier

As described in Sect. 2, many classifiers are known to work well in link prediction. We used the logistic regression classifier in this work, for reasons of interpretability and explainability, as further discussed in Sect. 2. In Table 5,

Table 5 Performance obtained with the II-A feature set (see Sect. 6.1)

| Label | Logistic Regression | Random Forest | XGBoost |
|-------|---------------------|---------------|---------|
| Rado | 0.921 | 0.951 | 0.955 |
| UC | 0.893 | 0.942 | 0.946 |
| EU | 0.873 | 0.953 | 0.942 |
| Dem | 0.944 | 0.984 | 0.981 |
| bitA | 0.945 | 0.974 | 0.974 |
| bitOT | 0.947 | 0.973 | 0.967 |
| chess | 0.735 | 0.833 | 0.830 |
| HepTh | 0.835 | 0.867 | 0.856 |
| HepPh | 0.879 | 0.816 | 0.798 |
| Condm | 0.760 | 0.875 | 0.870 |
| SX-MO | 0.944 | 0.959 | 0.959 |
| D-rep | 0.866 | 0.973 | 0.976 |
| Rbody | 0.905 | 0.944 | 0.938 |
| trust | 0.936 | 0.971 | 0.969 |
| Rtit | 0.931 | 0.948 | 0.946 |
| FB-w | 0.809 | 0.769 | 0.772 |
| FB-l | 0.803 | 0.769 | 0.772 |
| Enron | 0.912 | 0.973 | 0.970 |
| loans | 0.947 | 0.941 | 0.943 |
| WikiC | 0.936 | 0.979 | 0.981 |
| D-v | 0.941 | 0.908 | 0.910 |
| SX-AU | 0.970 | 0.990 | 0.990 |
| SX-SU | 0.965 | 0.981 | 0.982 |
| D-f | 0.926 | 0.977 | 0.977 |
| AMin | 0.849 | 0.872 | 0.865 |
| DBLP | 0.826 | 0.919 | 0.923 |
| mean | 0.892 | 0.925 | 0.923 |

we provide, for each of the datasets as introduced in Table 1, the performance in terms of AUC obtained using two other commonly used classifiers, being random forests (Pedregosa et al. 2011) and XGBoost (Chen and Guestrin 2016), with default parameters. For almost all datasets, similar relative performance is observed.

Acknowledgements Gerrit Jan de Bruin was supported by funding of the Dutch Ministry of Infrastructure and Water Management. We thank Jasper van Vliet for providing comments.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Adamic LA, Adar E (2003) Friends and neighbors on the Web. *Soc Netw* 25(3):211–230. [https://doi.org/10.1016/S0378-8733\(03\)00009-1](https://doi.org/10.1016/S0378-8733(03)00009-1)
- Al Hasan M, Chaoji V, Salem S, Zaki M, Hasan MA, Chaoji V, Salem S, Zaki M, York N (2006) Link prediction using supervised learning. In: *SDM06: workshop on link analysis, counter-terrorism and security*, vol 30, pp 798–805
- Barabási AL (2005) The origin of bursts and heavy tails in human dynamics. *Nature* 435(7039):207–211. <https://doi.org/10.1038/nature03459>
- Barabási AL (2016) *Network science*. Cambridge University Press, Cambridge
- Bishop CM (2013) *Pattern recognition and machine learning*. Springer, New York. <https://doi.org/10.1117/1.2819119>
- Bliss CA, Frank MR, Danforth CM, Dodds PS (2014) An evolutionary algorithm approach to link prediction in dynamic social networks. *J Comput Sci* 5(5):750–764. <https://doi.org/10.1016/j.jocs.2014.01.003>
- Brandes U, Kenis P, Lerner J, Van Raaij D (2009) Network analysis of collaboration structure in Wikipedia. In: *Proceedings of the 18th international world wide web conference. Association for Computing Machinery, New York*, pp 731–740. <https://doi.org/10.1145/1526709.1526808>
- Büttün E, Kaya M, Alhaji R (2016) A new topological metric for link prediction in directed, weighted and temporal networks. In: *Proceedings of the 2016 IEEE/ACM international conference on advances in social networks analysis and mining. IEEE, Los Alamitos*, pp 954–959. <https://doi.org/10.1109/ASONAM.2016.7752355>
- Büttün E, Kaya M, Alhaji R (2018) Extension of neighbor-based link prediction methods for directed, weighted and temporal social networks. *Inf Sci* 463–464:152–165. <https://doi.org/10.1016/j.ins.2018.06.051>
- Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp 785–794
- Da Silva Soares PR, Prudencio RBC (2012) Time series based link prediction. In: *Proceedings of the international joint conference on neural networks. IEEE, Brisbane*, pp 1–7. <https://doi.org/10.1109/IJCNN.2012.6252471>
- de Bruin GJ, Veenman CJ, van den Herik HJ, Takes FW (2020) Understanding dynamics of truck co-driving networks. *Stud Comput Intell* 882 SCI:140–151. https://doi.org/10.1007/978-3-030-36683-4_12
- de Bruin GJ, Veenman CJ, van den Herik HJ, Takes FW (2021) Experimental evaluation of train and test split strategies in link prediction. In: Benito RM, Cherifi C, Cherifi H, Moro E, Rocha LM, Sales-Pardo M (eds) *Complex networks & their applications IX*. Springer, Cham, pp 79–91. https://doi.org/10.1007/978-3-030-65351-4_7
- De Choudhury M, Sundaram H, John A, Seligmann DD (2009) Social synchrony: predicting mimicry of user actions in online social media. In: *2009 International conference on computational science and engineering*, vol 4. IEEE, Vancouver, pp 151–158. <https://doi.org/10.1109/CSE.2009.439>
- Dhote Y, Mishra N, Sharma S (2013) Survey and analysis of temporal link prediction in online social networks. In: *Proceedings of the 2013 international conference on advances in computing, communications and informatics. IEEE, Mysore*, pp 1178–1183. <https://doi.org/10.1109/ICACCI.2013.6637344>
- Divakaran A, Mohan A (2020) Temporal link prediction: a survey. *N Gener Comput* 38(1):213–258. <https://doi.org/10.1007/s00354-019-00065-z>

- Ghasemian A, Hosseinmardi H, Galstyan A, Airolidi EM, Clauset A (2020) Stacking models for nearly optimal link prediction in complex networks. *Proc Natl Acad Sci* 117(38):23393–23400. <https://doi.org/10.1073/pnas.1914950117>
- Gross JL, Yellen J, Zhang P (2013) Handbook of graph theory, 2nd edn. Chapman Hall/CRC, London
- Grover A, Leskovec J (2016) Node2vec: scalable feature learning for networks. Association for Computing Machinery, New York, pp 855–864. <https://doi.org/10.1145/2939672.2939754>
- Güneş İ, Gündüz-Öğüdücü Ş, Çataltepe Z (2016) Link prediction using time series of neighborhood-based node similarity scores. *Data Min Knowl Disc* 30(1):147–180. <https://doi.org/10.1007/s10618-015-0407-0>
- Hagberg A, Swart P, Chult S, D. (2008) Exploring network structure, dynamics, and function using NetworkX. Tech. rep., Los Alamos National Lab. Los Alamos, NM, USA
- Hamilton WL, Ying R, Leskovec J (2017) Representation learning on graphs: methods and applications. [arXiv:1709.05584](https://arxiv.org/abs/1709.05584)
- Hiraoka T, Masuda N, Li A, Jo HH (2020) Modeling temporal networks with bursty activity patterns of nodes and links. *Phys Rev Res* 2(2):023073. <https://doi.org/10.1103/PhysRevResearch.2.023073>
- Hogg T, Lerman K (2012) Social dynamics of Digg. *EPJ Data Sci* 1(1):5. <https://doi.org/10.1140/epjds5>
- Holzinger A, Biemann C, Pattichis CS, Kell DB (2017) What do we need to build explainable AI systems for the medical domain? [arXiv:1712.09923](https://arxiv.org/abs/1712.09923)
- Klimt B, Yang Y (2004) The enron corpus: a new dataset for email classification research. In: Boulicaut JF, Esposito F, Giannotti F, Pedreschi D (eds) Machine learning: ECML. Springer, Berlin, pp 217–226. https://doi.org/10.1007/978-3-540-30115-8_22
- Kumar S, Spezzano F, Subrahmanian VS, Faloutsos C (2017) Edge weight prediction in weighted signed networks. In: Proceedings—IEEE international conference on data mining. IEEE, Barcelona, pp 221–230. <https://doi.org/10.1109/ICDM.2016.175>
- Kumar S, Hamilton WL, Leskovec J, Jurafsky D (2018) Community interaction and conflict on the web. In: Proceedings of the 2018 world wide web conference. International World Wide Web Conferences Steering Committee, Geneva, Switzerland, pp 933–943. <https://doi.org/10.1145/3178876.3186141>
- Kumar A, Singh SS, Singh K, Biswas B (2020) Link prediction techniques, applications, and performance: a survey. *Physica A* 553:124289. <https://doi.org/10.1016/j.physa.2020.124289>
- Kunegis J (2013) KONECT: the Koblenz network collection. In: Proceedings of the 22nd international conference on world wide web. Association for Computing Machinery, New York, pp 1343–1350. <https://doi.org/10.1145/2487788.2488173>
- Leskovec J, Krevl A (2014) SNAP datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>
- Leskovec J, Kleinberg J, Faloutsos C (2007) Graph evolution: densification and shrinking diameters. *ACM Trans Knowl Discov Data* 1(1):2–43. <https://doi.org/10.1145/1217299.1217301>
- Ley M (2002) The DBLP computer science bibliography: evolution, research issues, perspectives. In: Laender AHF, Oliveira A (eds) String processing and information retrieval, string processing and information retrieval, vol 2476. Springer, Berlin, pp 1–10. https://doi.org/10.1007/3-540-45735-6_1
- Liben-Nowell D, Kleinberg J (2007) The link-prediction problem for social networks. *J Am Soc Inform Sci Technol* 58(7):1019–1031. <https://doi.org/10.1002/asi.20591>
- Lichtenwalter R, Chawla NV (2012) Link prediction: fair and effective evaluation. Proceedings of the 2012 IEEE/ACM international conference on advances in social networks analysis and mining, pp 376–383. <https://doi.org/10.1109/ASONAM.2012.68>
- Lichtenwalter RN, Lussier JT, Chawla NV (2010) New perspectives and methods in link prediction. In: Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining. Association for Computing Machinery, New York, pp 243–252. <https://doi.org/10.1145/1835804.1835837>
- Linyuan LL, Zhou T (2011) Link prediction in complex networks: a survey. *Physica A* 390(6):1150–1170. <https://doi.org/10.1016/j.physa.2010.11.027>
- Marjan M, Zaki N, Mohamed EA (2018) Link prediction in dynamic social networks: a literature review. In: 5th International congress on information science and technology. IEEE, Marrakech, pp 200–207. <https://doi.org/10.1109/CIST.2018.8596511>
- Michalski R, Palus S, Kazienko P (2011) Matching organizational structure and social network extracted from email communication. In: Abramowicz W (ed) Business information systems, vol 87. Springer, Berlin. https://doi.org/10.1007/978-3-642-21863-7_17
- Molnar C (2020) Interpretable machine learning. Lulu.com
- Muniz CP, Goldschmidt R, Choren R (2018) Combining contextual, temporal and topological information for unsupervised link prediction in social networks. *Knowl Based Syst* 156:129–137. <https://doi.org/10.1016/j.knsys.2018.05.027>
- Newman MEJ (2002) Assortative mixing in networks. *Phys Rev Lett* 89(20):208701. <https://doi.org/10.1103/PhysRevLett.89.208701>
- Öczan A, Öğüdücü ŞG (2015) Multivariate temporal Link Prediction in evolving social networks. In: 2015 IEEE/ACIS 14th international conference on computer and information science. IEEE, Las Vegas, pp 185–190. <https://doi.org/10.1109/ICIS.2015.7166591>
- Öczan A, Öğüdücü ŞG (2017) Supervised temporal link prediction using time series of similarity measures. In: 2017 Ninth international conference on ubiquitous and future networks. IEEE, Milan, pp 519–521. <https://doi.org/10.1109/ICUFN.2017.7993838>
- O'Madadhain J, Hutchins J, Smyth P (2005) Prediction and ranking algorithms for event-based network data. *ACM SIGKDD Explorations Newsletter* 7(2):23–30. <https://doi.org/10.1145/1117454.1117458>
- Opsahl T (2013) Triadic closure in two-mode networks: redefining the global and local clustering coefficients. *Soc Netw* 35(2):159–167. <https://doi.org/10.1016/j.socnet.2011.07.001>
- Paranjape A, Benson AR, Leskovec J (2017) Motifs in temporal networks. In: Proceedings of the 10th ACM international conference on web search and data mining, pp 601–610. <https://doi.org/10.1145/3018661.3018731>
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12:2825–2830
- Perozzi B, Al-Rfou R, Skiena S (2014) DeepWalk: Online learning of social representations. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining. Association for Computing Machinery, New York, pp 701–710. <https://doi.org/10.1145/2623330.2623732>
- Potgieter A, April KA, Cooke RJE, Osumakinde IO (2007) Temporality in link prediction: understanding social complexity
- Redmond U, Cunningham P (2013) A temporal network analysis reveals the unprofitability of arbitrage in the Prosper Marketplace. *Expert Syst Appl* 40(9):3715–3721. <https://doi.org/10.1016/j.eswa.2012.12.077>
- Richardson M, Agrawal R, Pedro D (2003) Trust management for the semantic web. In: Fensel D, Sycara K, Mylopoulos J (eds) The semantic web—ISWC. Springer, Berlin, pp 351–368. <https://doi.org/10.1109/ICCEE.2009.241>

- Romero M, Finke J, Rocha C, Tobón L (2020) Spectral evolution with approximated eigenvalue trajectories for link prediction. *Soc Netw Anal Min* 10(1):60. <https://doi.org/10.1007/s13278-020-00674-3>
- Soares PR, Prudêncio RB (2013) Proximity measures for link prediction based on temporal events. *Expert Syst Appl* 40(16):6652–6660. <https://doi.org/10.1016/j.eswa.2013.06.016>
- Takes FW, Kusters WA (2011) Determining the diameter of small world networks. In: *Proceedings of the 20th ACM international conference on Information and knowledge management*. Association for Computing Machinery, New York, pp 1191–1196. <https://doi.org/10.1145/2063576.2063748>
- Tylenda T, Angelova R, Bedathur S (2009) Towards time-aware link prediction in evolving social networks. In: *Proceedings of the 3rd workshop on social network mining and analysis*, vol 9. Association for Computing Machinery, New York, pp 1–10. <https://doi.org/10.1145/1731011.1731020>
- Van Mieghem P, Wang H, Ge X, Tang S, Kuipers FA (2010) Influence of assortativity and degree-preserving rewiring on the spectra of networks. *Eur Phys J B* 76(4):643–652. <https://doi.org/10.1140/epjb/e2010-00219-x>
- Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat I, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P (2020) SciPy 1.0 Contributors: SciPy 1.0: fundamental algorithms for scientific computing in python. *Nat Methods* 17:261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- Viswanath B, Mislove A, Cha M, Gummadi KP (2009) On the evolution of user interaction in Facebook. In: *Proceedings of the 2nd ACM workshop on Online social networks*. Association for Computing Machinery, New York, pp 37–42. <https://doi.org/10.1145/1592665.1592675>
- Wikileaks (2016) US Democratic National Committee leak. <https://www.wikileaks.org/dnc-emails/>
- Yin H, Benson AR, Leskovec J, Gleich DF (2017) Local higher-order graph clustering. In: *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*, pp 555–564. <https://doi.org/10.1145/3097983.3098069>
- Zhuang H, Sun Y, Tang J, Zhang J, Sun X (2013) Influence maximization in dynamic social networks. In: *13th International conference on data mining*. IEEE, Dallas, pp 1313–1318. <https://doi.org/10.1109/ICDM.2013.145>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.