Universiteit
Leiden
The Netherlands

Network analysis methods for smart inspection in the transport domain

Bruin, G.J. de

Citation
Bruin, G. J. de. (2023, November 16). *Network analysis methods for smart inspection in the transport domain*. *SIKS Dissertation Series*. Retrieved from https://hdl.handle.net/1887/3656981

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/3656981

Note: To cite this publication please use the final published version (if applicable).

2

Supervised link prediction in large-scale temporal networks

Missing link prediction is a well-studied technique for inferring the missing edges between two nodes in a static representation of a network. In temporal networks, such as modern-day social networks, the temporal information associated with each link can be used to predict *future links* between thus far unconnected nodes, thereby enabling *temporal* link prediction (Definition 10). In the continuation of the thesis, this is referred to as link prediction. In this chapter, we address Research question 1, which reads as follows.

> **Research question 1:** *What is the relation between network structure and model performance in link prediction?*

The chapter presents a systematic investigation of link prediction, making use of 26 temporal, structurally diverse, real-world networks ranging from thousands to millions of nodes and links. We analyze for each network the relationship between (1) the typology and (2) the obtained link prediction performance. Meanwhile, we employ well-established topological features.

The current chapter corresponds to the following publication:
G. J. de Bruin, C. J. Veenman, H. J. van den Herik, and F. W. Takes. „Supervised temporal link prediction in large-scale real-world networks." *Social Network Analysis and Mining* 11, 80 (2021). DOI: 10.1007/s13278-021-00787-3

2.1 Link prediction

Link prediction is a frequently employed method within the broader field of social network analysis [10]. Many critical real-world applications exist in a variety of domains. Two examples are the prediction of (1) missing links between pages of Wikipedia and (2) users that are friends on an online social network [101]. As mentioned in Section 1.4, link prediction is often defined as predicting missing links based on the currently observable links in a network [114]. Many real-world networks have temporal information on when the edges were created [50]. Such *temporal networks* are also called dynamic or evolving networks (see also Definition 9). They open up the possibility of doing link prediction (contrasting with the aforementioned missing link prediction). The availability of temporal information means that we can infer future edges between two nodes as opposed to only predicting missing links [110]. For instance, in friendship networks, link prediction may (1) facilitate friend recommendations and (2) predict who will form new friendships in the future.

Existing work on link prediction is typically performed on one or a handful of specific networks, making it challenging to examine the generalizability of the approaches used [115]. This chapter provides the first large-scale empirical study of link prediction on 26 different large-scale and structurally diverse temporal networks originating from various domains. In doing so, we provide a systematic investigation of *how* temporal information is best used in link prediction.

We illustrate how the performance of social networks will likely be higher. Because they have a higher density than other networks, nodes have more common neighbors. Thus a given instance may provide more information to the link prediction model. By using this example, we demonstrate that it is essential to understand the relationship between the network's structural characteristics and the performance of link prediction features.

A common approach in link prediction is to learn a classifier that utilizes multiple features to classify which links are missing or, in case of temporal link prediction, will appear in the future. Features are typically computed for every pair of nodes that is not (yet) connected, based on the topology of the network [101]. 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 in machine learning include Common Neighbors (CN), Adamic-Adar (AA), Jaccard Coefficient (JC), and Preferential Attachment (PA) (Subsection 2.4.1A). These features clearly relate to the structural position of the nodes in the network. Previous work has suggested a straightforward approach to taking the temporal evolution into account in topological features [35, 189]. We describe the process of obtaining the set of temporal topological features in Subsection 2.4.1B. The benefit of using such a set of features is that they are well-established and interpretable. Moreover, recent work has

shown that in a supervised classifier, the topological features perform as well as other features that are less interpretable and more complex [63]. A further comparison with other features is provided in Section 2.2.

As we have seen, previous studies ignore that two types of temporal networks can be distinguished (see also Section 1.3): networks with *persistent relationships* and networks with *discrete events* [133]. The example of friendship networks, as mentioned earlier, contains edges marking persistent relationships that occur at most once for related persons. In the case of discrete event networks, an edge marks a discrete event (e.g., a communication) 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. So far, previous studies have ignored that each link is not of the same type. In our approach, we address this literature gap by 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.

Finally, the temporal topological features implicitly assume so-called edge-centered temporal behavior. This suggests 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. At this point, we remark that the nodes are operating somewhat independently of the structure of the remainder of the network [78]. To investigate whether the assumption on the temporal aspect holds, we compare (1) temporal topological features with (2a) features consisting of static topological features and/or (2b) features capturing temporal node activity. By testing this distinction on the 26 different temporal networks, we are able to better understand whether the temporal aspect is best captured by considering edge-centered or node-centered temporal information.

Below we sum up the four contributions of this chapter.

1. To the best of our knowledge, we are one of the first to present *a large-scale empirical study of link prediction on various networks*. In total, we examine the performance of a link prediction model on 26 structurally diverse networks, varying in size from a few hundred to over a million nodes and edges.
2. We analyze possible relations between structural network properties and the observed performance in link prediction. We find that *networks with degree disassortativity* (see Subsection 1.2.1), signaling frequent connections between nodes with different degrees, *show better performance in link prediction*.
3. We show that the performance of link prediction can significantly be improved by *taking multiple past interactions* between two nodes *into account*.
4. To understand the relation between *node-centered* and *edge-centered* temporal behavior, the information networks used in this study stand out, as they appear to have more node-centered temporal behavior.

The remainder of this chapter is structured as follows. In Section 2.2, we further elaborate on related work. Section 2.3 provides the preliminaries of this chapter, leading up to a formal definition of link prediction. We continue with the research methodology in Section 2.4. It will be followed by describing the temporal networks in Section 2.5. In Section 2.6, we report on the four experiments and their results. In Section 2.7, the conclusion is presented, together with an outlook.

2.2 Related work on link prediction

Although much literature is available on link prediction, we found that attention to *temporal* networks and how to apply link prediction to them is relatively limited. Some reviews have been published. They are pointing out the various approaches toward link prediction [49, 50]. 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 satisfactory performance (which hinders a generic approach to all networks) or (2) techniques that do not scale to larger networks [101] (rendering them unusable for the larger networks used in the study).

Second, approaches such as matrix factorization, spectral clustering [168], and deep learning approaches, such as DeepWalk [158] and Node2Vec [70], all try to find a lower-dimensional representation of the temporal network and use the obtained representation as a basis for link prediction. Apart from hindering the generic approach desired in this work, the need for interpretability of lower-dimensional representations usually is a significant problem in domains where the model needs to be interpreted by law, such as in medicine or businesses dealing with personal information [84]. For example, in Chapter 4, we will examine the driving patterns of trucks in a so-called truck co-driving network, where trucks are connected when they frequently drive together. When an inspectorate uses 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 toward link prediction that apply 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 [71, 134, 135, 161, 177]. For each of these snapshots, static topological features are learned. The topological features of a future network snapshot are learned using forecasting, thereby 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.

Fourth, we focus on temporal topological features [34, 189]. Recent work has suggested that using topological features in supervised learning may outperform more complex features learned from a lower-dimensional representation of the temporal network [63]. Section 2.4 provides further details on this concept. The topological features are provided to a supervised link prediction classifier. Many different machine learning algorithms are known to work well in link prediction. Commonly used classifier algorithms include logistic regression [133, 161], support vector machines [3, 135], k-nearest neighbors [3, 34, 35], and random forests [32–35, 63, 135]. We report performances using the logistic regression classifier. This classifier provides the four following benefits, (1) it allows an intuitive explanation of how each instance is classified [17], (2) the classifier is relatively simple and hence interpretable [123], (3) the classifier scales well to larger networks, and (4) good results are achieved without parameter optimization [133].
To sum up, in contrast to earlier works on link prediction, which has been applied on only a handful of networks [18, 34, 35, 71, 124, 133–135, 161, 168, 177, 178, 189], 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.

2.3 Preliminaries

This section describes the notation used in this chapter in Subsection 2.3.1. In Subsection 2.3.2, we explain the various network properties and measures used in this chapter. Finally, in Subsection 2.3.3, we formally describe the link prediction task.

2.3.1 Notation

In this chapter, we use the following notation for the link prediction task (see Definition 10).

An undirected, temporal network $G_{[t_a,t_b]}(V, E)$ consists of a set of nodes V and edges (or, equivalently, links) $E = \{(u, v, t_i) \mid u, v \in V \wedge t_a \leq t_i \leq t_b\}$ that occur between timestamps t_a and t_b.

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 [69]. In this work, the removal of edges is not considered since this information is unavailable for most temporal networks.

A static representation of the underlying network is needed to compare static and temporal features (Section 2.4). The static, simple graph is obtained from the temporal network by collapsing multi-edges into a single edge. The graph's number of nodes (also called the size) is $n = |V|$, and the number of edges is $m = |E|$. For convenience in later definitions, $N(u)$ is the set of all neighbors of node $u \in V$. The size of the set, i.e., $|N(u)|$,

is the number of neighbors of node u, which is in a simple graph equal to the degree of node u. In case of a multigraph, $|E(u)|$ is the degree of node u.

2.3.2 Real-world network properties and their measures

Several properties exist that characterize the macro-scale of a network [10]. These properties guide us in exploring how the structure relates to the link prediction performance. In this work, we use at least the following five properties: (1) the number of nodes and edges (not explained below), (2) average clustering coefficient, (3) degree assortativity, (4) density, and (5) diameter. Each property is derived from the underlying static graph.

- **Average clustering coefficient:** The average clustering coefficient (see also Subsection 1.2.2) is given by $C = n^{-1} \sum_{u \in V} 2L_u / \big(|N(u)| \cdot \big(|N(u)| - 1\big)\big)$, when $|N(u)| - 1 > 1$. L_u represents the number of edges between the neighbors of node u. Highly clustered networks are often observed in the real world and particularly in social networks.
- **Degree assortativity:** It is often observed that nodes do not connect to random other nodes but instead connect to similar ones (see also Subsection 1.2.1). For instance, degree assortativity is observed in social networks, 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 [126] (see also Definition 8). In case low-degree nodes more frequently connect with high-degree nodes, the obtained value is negative.
- **Density:** The density of a network indicates what fraction of the pairs of nodes are connected. 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. For a network with m edges and n nodes, the *density* is equal to $2m/n(n-1)$.
- **Diameter:** The diameter is the largest distance observed between any pair of nodes (see also Section 1.2). The distance is measured in terms of the number of nodes in the path between the pair of nodes. This property, together with density, captures how well-connected a network is.

2.3.3 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 $G_{[t_{q=0}, t_{q=s}]}$ 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 with 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 result 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 the information of network $G_{[t_{q=0}, t_{q=s}]}$, 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.

2.4 Chapter research methodology

This section explains the research methodology used in this chapter. It can be seen as an addition to the general research methodology described in Section 1.7. We emphasize described features used in supervised link prediction. We start by explaining the different sets of features in Subsection 2.4.1. We then present a novel research method, i.e., an intuitive approach to incorporate information on past interactions in the case of discrete event networks. Additionally, in Subsection 2.4.2, we discuss the supervised link prediction model.

2.4.1 Features

We explain three types of features in this section. First, the static topological features are provided in Subsection 2.4.1A. Second, the temporal topological features are given in Subsection 2.4.1B. Finally, the node activity features are specified in Subsection 2.4.1C.

2.4.1A Static topological features

We use four common static topological features, forming the feature vector for each candidate pair of nodes (u, v). These features are computed on the static graph underlying the temporal network, as defined in Subsection 2.3.1. Below we define each of them.

- **Adamic-Adar (AA):** The AA feature considers all common neighbors, favoring nodes with low degrees [1].

$$AA_{\text{static}}(u, v) = \sum_{z \in N(u) \cap N(v)} 1/\log \left|N(z)\right| \text{ with } \left|N(z)\right| > 1 \tag{2.1}$$

- **Common Neighbors (CN):** The CN feature equals the number of common neighbors of two nodes.

$$CN_{\text{static}}(u, v) = |N(u) \cap N(v)| \tag{2.2}$$

- **Jaccard Coefficient (JC):** The JC feature is similar to the CN feature but normalizes for the number of unique neighbors of the two nodes.

$$JC_{\text{static}}(u,v) = |N(u) \cap N(v)| / |N(u) \cup N(v)| \text{ with } |N(u) \cup N(v)| > 0 \qquad 2.3$$

- **Preferential Attachment (PA):** The PA feature considers that nodes with a high degree are more likely to make new links than nodes with a lower degree (see also Section 1.3).

$$PA_{\text{static}}(u,v) = |N(u)| \cdot |N(v)| \qquad 2.4$$

2.4.1B Temporal topological features

The temporal topological features are extended versions of the static topological features presented above in Subsection 2.4.1A. The construction of these features then requires three steps, namely:
Step I. Temporal weighting
Step II. Past event aggregation
Step III. Weighted topological features
The resulting feature vector for a given pair of nodes, after applying the three steps, consists of all possible combinations of three different temporal weighting functions (exponential, linear, square root), eight different past event aggregations (see below under step II), and four different weighted topological features (AA, CN, 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 $3 \cdot 4 = 12$.

Step I: Temporal weighting
The topological features need weighted edges (see Step III), while the networks used in this study have edges with an associated timestamp. In the temporal weighting step, we obtain these weights with the help of a methodology described by Tylenda *et al.* [189]. The temporal weighting functions are provided in Equations 2.5 to 2.7. In these functions, a numeric timestamp t_i 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.

In Figure 2.1, the behavior of the different weighting functions is shown when applied to the DBLP network [109]. It is further described in Section 2.5. The exponential weighting function (Equation 2.6) assigns a higher weight to more recent edges than the linear (Equation 2.5) and square root (Equation 2.7) functions. In contrast, the square root function assigns higher weights to older edges than the linear and exponential functions. When the weights of older edges become close to zero, these edges are discarded by the weighted topological features. To prevent the edges from 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), with $0 \geq \ell < 1$.

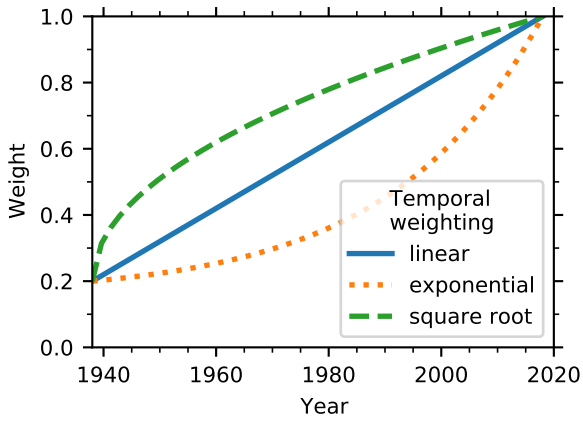

Figure 2.1: Mapping of three weighting functions for the DBLP network.

$$w_{\text{linear}} = \ell + (1-\ell)\frac{t_i - t_{\min}}{t_{\max} - t_{\min}} \tag{2.5}$$

$$w_{\text{exponential}} = \ell + (1-\ell) \cdot \frac{\exp\left(3 \cdot (t_i - t_{\min})/(t_{\max} - t_{\min})\right) - 1}{\mathrm{e}^3 - 1} \tag{2.6}$$

$$w_{\text{square root}} = \ell + (1-\ell) \cdot \sqrt{(t_i - t_{\min})/(t_{\max} - t_{\min})} \tag{2.7}$$

Step II: Past event aggregation

In the 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 need to obtain a single weight for each node pair, capturing the past activity of these nodes. For this purpose, we propose to obtain the weight by using eight different aggregation functions. All eight functions use as input a set containing all 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. Utilizing these as summary statistics, we capture the different types of linkage in networks that occur in the real world. For example, it may matter whether interaction occurred often, far away in the past, or recently. The aggregation functions aim to capture different temporal behaviors. Quantile functions bin the set of weights, a common feature-engineering step. Taking the mean, sum, and variance of the set of weights allows the model to capture more complex trends. An example of these complex trends is the so-called bursty behavior, which is often observed in real-world data [9].

Step III: Weighted topological features
In Equations 2.8 to 2.11, the Weighted Temporal Features (WTF) are presented, which are taken from Bütün *et al.* [35]. In these equations, $WTF(u, v)$ denotes the weight obtained for a given pair of nodes (u, v) after edges have been temporally weighted and, in case of networks with discrete events, events have been aggregated.

$$AA_{\text{temporal}}(u, v) = \sum_{z \in N(v) \cap N(y)} \frac{WTF(u, z) + WTF(v, z)}{\log\left(1 + \sum\limits_{x \in N(z)} WTF(z, x)\right)} \tag{2.8}$$

$$CN_{\text{temporal}}(u, v) = \sum_{z \in N(u) \cap N(v)} WTF(u, z) + WTF(v, z) \tag{2.9}$$

$$JC_{\text{temporal}}(u, v) = \sum_{z \in N(u) \cap N(v)} \frac{WTF(u, z) + WTF(v, z)}{\sum\limits_{x \in N(u)} WTF(u, x) + \sum\limits_{y \in N(v)} WTF(v, y)} \tag{2.10}$$

$$PA_{\text{temporal}}(u, v) = \sum_{u \in N(x)} WTF(u, x) \cdot \sum_{v \in N(y)} WTF(v, y) \tag{2.11}$$

2.4.1C Node activity features

The goal of the node activity features is to capture node-centered 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 aggregation functions applied to the node activity, and four combinations of the node activity. It results in a feature vector of length $3 \cdot 7 \cdot 4 = 84$.

Step 1. Temporal weighting. The temporal weighing method is the same as used in feature engineering of the temporal weighted topological features (see Subsection 2.4.1B and Figure 2.1).

Step 2. Aggregation of node activity. The weights from all edges adjacent to the node under investigation are collected for each node. We obtain a fixed feature vector for each node by aggregating using the following seven functions: (1) the zeroth, (2) first, (3) second, (4) third, (5) fourth quantile, and (6) sum and (7) mean of the node activity vector (here the variance of all node weights is suppressed because some nodes have only one edge, rendering the variance undefined). Similar to the engineering of the temporal topological features, these aggregations capture different kinds of activity that a node may exhibit. Nodes show bursty activity patterns in some networks [78].

Step 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 (1) absolute difference, (2) minimum, (3) maximum, and (4) sum. By doing this, we obtain the *node activity feature vector*.

2.4.2 Supervised link prediction

The features discussed in Subsection 2.4.1 serve as input for a supervised machine learning model that predicts whether a pair of currently disconnected nodes will connect in the future (see Subsection 2.3.3). 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 Section 2.2). We did not consider optimizing parameters because it is outside the scope of the current work.

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. This would result in a significant class imbalance. To address the imbalance 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.

Step 1. Pairs of nodes are only selected if they are exactly the same distance apart [111]. In our study, which involves large networks, the selection is limited to include only pairs of nodes with a distance of two.

Step 2. Pairs of nodes are sampled by replacement such that 10,000 will connect (positive instances) and 10,000 will not connect (negative instances). By following this sampling procedure, we obtain a balanced set of examples that do not require further post-processing and can be used directly by the classifier. In practice, the train set for the logistic regression classifier is obtained using stratified sampling, taking 75% of all examples. The remaining instances are used as a test set. Because we do not optimize any parameters of the logistic regression classifier, no validation set is used.

Analogously to previous work [50], we measure the classifier's performance on the test set utilizing the Area Under the ROC Curve (AUC). The AUC only considers the ranking of each score obtained for each pair of nodes provided to the logistic regression classifier. It makes the measured performance robust to cases where the applied threshold on the scores is chosen poorly. An AUC of 0.5 signals random behavior, i.e., no classifier performance. A perfect performance is obtained when the AUC equals 1, which is highly unlikely in practical settings.

2.5 Data and the statistics used

Our experiments are performed on a structurally diverse and large collection of 26 temporal networks. The networks can be categorized into three domains: social, information, and technological. The distinction of networks in these three domains is taken from other network repositories [104, 107]. In Table 2.1, some common structural properties of these datasets are presented (see Subsection 2.3.2 for properties and measures). It is apparent from Figure 2.2, showing the relation between the number of nodes and edges for each of the 26 datasets, that the selected networks span a broad range in size.

Table 2.1: Summary statistics of the 26 temporal networks (sorted by number of nodes). (The following abbreviations and symbols are used in the heading of the columns; D.a.: Degree assortativity, A.c.c: Average clustering coefficient, ø: Diameter. In the column "Domain", Technological is abbreviated to Tech. and Information to Inf. The column label provides an abbreviated name of the specific dataset. The full names are in the references. For * and **, see Subsection 2.6.4.)

	Label	Domain	Edge type	Nodes (n)	Edges (m)	Density	D.a.	A.c.c.	ø	Ref.
	emails	Social	persistent	167	82,927	$2 \cdot 10^{-1}$	0.15	0.59	5	[118]
**	UC	Inf.	persistent	899	33,720	$2 \cdot 10^{-2}$	0.10	0.07	6	[139]
	EU	Social	persistent	986	332,334	$3 \cdot 10^{-2}$	0.05	0.41	7	[211]
	Dem	Social	persistent	1,891	39,264	$2 \cdot 10^{-3}$	−0.15	0.21	8	[200]
	bitA	Social	event	3,683	22,650	$2 \cdot 10^{-3}$	−0.15	0.17	10	[103]
	bitOT	Social	event	5,573	32,029	$1 \cdot 10^{-3}$	−0.15	0.16	14	[103]
	chess	Inf.	event	6,050	21,163	$1 \cdot 10^{-3}$	0.36	0.05	13	[104]
	HepTh	Inf.	persistent	6,798	290,597	$9 \cdot 10^{-3}$	0.08	0.77	11	[106]
	HepPh	Inf.	persistent	16,959	2,322,259	$8 \cdot 10^{-3}$	0.17	0.61	8	[106]
*	Condm	Social	persistent	17,218	88,090	$4 \cdot 10^{-4}$	0.29	0.64	19	[112]
	SX-MO	Social	persistent	24,818	506,550	$6 \cdot 10^{-4}$	−0.05	0.31	9	[143]
	D-rep	Social	event	30,398	87,627	$2 \cdot 10^{-4}$	0.02	0.01	12	[46]
	Rbody	Tech.	persistent	35,010	265,491	$2 \cdot 10^{-4}$	0.03	0.18	11	[102]
	Rtit	Tech.	persistent	53,018	510,787	$1 \cdot 10^{-4}$	−0.01	0.18	17	[102]
	FB-w	Social	event	55,387	335,708	$2 \cdot 10^{-4}$	−0.02	0.12	16	[194]
	FB-l	Social	event	55,387	335,708	$2 \cdot 10^{-4}$	0.22	0.12	16	[194]
	Enron	Social	persistent	87,273	1,148,072	$8 \cdot 10^{-5}$	0.22	0.12	14	[97]
	loans	Inf.	event	89,269	3,394,979	$8 \cdot 10^{-4}$	−0.04	0.00	8	[163]
	trust	Social	event	114,467	717,667	$9 \cdot 10^{-5}$	−0.07	0.13	14	[165]
	Wiki	Social	persistent	116,836	2,917,785	$3 \cdot 10^{-4}$	−0.06	0.36	10	[25]
	D-v	Inf.	event	139,409	3,018,197	$3 \cdot 10^{-4}$	−0.21	0.14	4	[79]
	SX-AU	Social	persistent	159,316	964,437	$4 \cdot 10^{-5}$	−0.10	0.11	13	[143]
	SX-SU	Social	persistent	194,085	1,443,339	$4 \cdot 10^{-5}$	−0.08	0.12	12	[143]
	D-f	Social	event	279,374	1,729,983	$4 \cdot 10^{-5}$	−0.05	0.09	18	[79]
	AMin	Social	persistent	855,165	23,787,273	$9 \cdot 10^{-6}$	0.16	0.61	22	[215]
	DBLP	Social	persistent	1,824,701	29,487,744	$5 \cdot 10^{-6}$	0.15	0.63	23	[109]

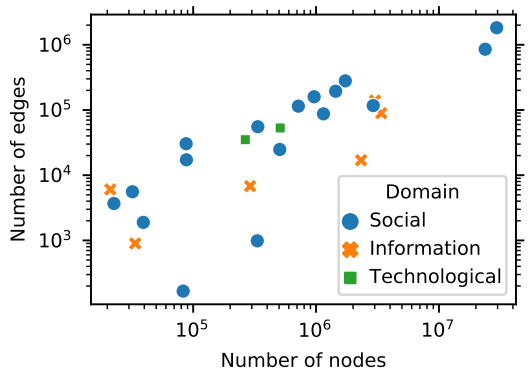

Figure 2.2: Number of nodes and edges of the 26 temporal networks. (The horizontal and vertical axes have logarithmic scaling.)

Also, for each network, it is indicated whether the edges mark persistent relationships or discrete events. In the latter case, the network has a multigraph structure, which requires preprocessing as discussed in Subsection 2.4.1B. We observe seventeen networks showing degree disassortative behavior, meaning high-degree nodes tend to connect to low-degree nodes more frequently. The other nine networks show the opposite behavior. 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 KOblenz NEtwork CollecTion (KONECT) repository [104]. Four networks (EU, Rbody, Rtit, and trust, see Table 2.1) were obtained from the Stanford Network Analysis Project (SNAP) repository [107]. The Arnetminer (Aminer) network was obtained directly from http://www.cn.aminer.org/data. The last column in Table 2.1 references the work where each network is introduced for the first time. Any directed network is converted into an undirected network by ignoring the directionality. In originally signed networks, we use only positive edges.

2.6 Experiments

In Subsection 2.6.1, we start with the experimental setup. Then, the structure follows the four experiments described in four separate subsections. In the first experiment (Subsection 2.6.2), we examine *the performance of link prediction* on 26 networks. The second experiment (Subsection 2.6.3) continues with analyzing *the relation between structural network properties and the performance in link prediction*. In the third experiment (Subsection 2.6.4), we show the results of past event aggregation to *the link prediction in networks with discrete events*. We finish with the fourth experiment (Subsection 2.6.5) with a *comparison between node-centered and edge-centered temporal behavior*.

2.6.1 Experimental setup

The research methodology to obtain examples and labels that serve as input for the classifier has been briefly explained in Subsection 2.4.2. We need to determine the value s for each network. Around two-thirds of the edges are commonly used for extraction of features [3, 34, 35, 112], and hence we choose $s = 2/3$.

The first step in the creation of temporal topological and node activity features is to assign temporal weight to each edge. In Subsection 2.4.1B, Step I, parameter ℓ is introduced to prevent discarding old edges in the temporal weighting method. Based on earlier work [189], we set $\ell = 0.2$, giving minimal weight to links far away in the past while still sufficiently discounting these older links.

In the four experiments, we use four sets of features. These feature sets, which are indicated by capital Roman numerals, are as follows. They are defined in Subsection 2.4.1.

I Static topological (as defined in Subsection 2.4.1A)

II-A Temporal topological (as defined in Subsection 2.4.1B)

II-B Temporal topological without past event aggregation (like Subsection 2.4.1B but skipping Step II and using only the last occurring event)

III Static topological + node activity (Subsection 2.4.1B + Subsection 2.4.1C)

Standardizing features by subtracting the mean and scaling the variance to unit is standard practice. The logistic regression classifier provided in the Python scikit-learn package [149] is used. Although the goal of this work is not to extensively compare machine learning classifiers, in Subsection 2.6.2 results on the performance in terms of AUC obtained using two other commonly used classifiers, random forests [149] and eXtreme Gradient Boosting (XGBoost) [41] are presented. For almost all datasets, similar relative performance is observed.

The code used in this research is publicly available [28]. It uses the Python language and the packages NetworkX [72] for network analysis, scikit-learn [149] for the machine learning pipeline, and the Scipy ecosystem [193] for some of the feature engineering and statistical tests. The C++ library teexGraph [185] was used to determine the diameter of each network. The package versions and all dependencies can be found in the repository.

2.6.2 Experiment 1: Improvement of prediction performance with temporal information

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

The results of this comparison are presented in Figure 2.3 and Table 2.2. (Feature sets II-B and III are used in later experiments.) They indicate that using temporal information improves the prediction performance of new links, i.e., performance reported in column

Table 2.2: Link prediction performance of the 26 temporal networks. (The following sets of features are used: I: Static topological features; II-A: Temporal topological features with past event aggregation; II-B: Temporal topological features without past event aggregation; and III: Static topological + node activity features.)

	Label	Domain	Edge type	Nodes (n)	AUC			
					I	II-A	II-B	III
	emails	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	1,891	0.920	0.944	0.919	0.938
	bitA	Social	simple	3,683	0.868	0.945	0.945	0.940
	bitOT	Social	simple	5,573	0.821	0.947	0.947	0.939
	chess	Information	simple	6,050	0.665	0.735	0.735	0.736
	HepTh	Information	multi	6,798	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

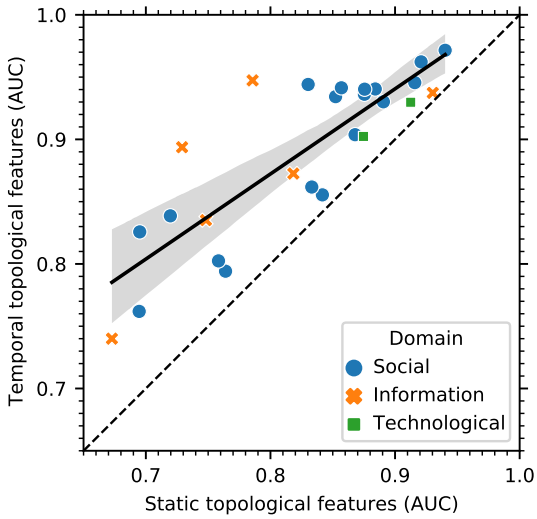

Figure 2.3: Link prediction performance of the 26 temporal networks. (The black line and gray band indicates the best linear regression fit and its 95% confidence interval, respectively.)

"II-A" is always higher than that in "I". So, every network performs better when temporal topological features are used. The average improvement in performance is 0.07 ± 0.04 (mean $\pm$ standard deviation).

For some networks, performance improves considerably when temporal information is used in the 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. From the results of this experiment we may conclude that the performance improvement can be related to the network's structure. Next, the relation between the structural properties of networks and the performance in link prediction is explored.

Choice of classifier

The logistic regression classifier was used for interpretability (see Definition 6), as further discussed in Section 2.2. In Table 2.3, we provide, for each of the datasets as introduced in Table 2.1, the performance in terms of AUC obtained using two other commonly used classifier algorithms, being the random forest [149] and XGBoost [41] algorithms, with default parameters. For almost all datasets, similar relative performance is observed. We continue the other experiments (Subsections 2.6.3 to 2.6.5) using the logistic regression classifier.

Table 2.3: Link prediction performance with past event aggregation (Feature set II-A, see experimental setup in Section 2.6).

	Label	Logistic Regression	Random Forest	XGBoost
	emails	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

2.6.3 Experiment 2: Structural network properties and link prediction performance

In the second experiment, we examine and discuss *which structural properties are associated with high link prediction performance*. We do so by exploration of the Pearson correlation coefficient, ρ, with the link prediction performance obtained.

In Figure 2.4, the Pearson correlations between the performance in link prediction and the structural network properties discussed in Section 2.5 are presented. Most properties show a modest correlation with the link prediction performance. However, a significant negative correlation is found 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 ($p = 5 \cdot 10^{-7}$). It means that strong disassortative behavior 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 Figure 2.5.

We observe a negative correlation between degree assortativity and link prediction performance. It can be explained as follows. In real-world networks, low-degree nodes typically vastly outnumber the high-degree nodes. However, nodes far exceeding the average degree, so-called hubs, are also relatively often observed in real-world networks [10]. In degree disassortative networks, the numerous low-degree nodes connect more frequently with hubs than 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 to perform better.

From Figure 2.5, we also learn that the temporal topological features have an even stronger correlation ($\rho = -0.82$) than the static topological features ($\rho = -0.78$). A possible explanation is that the temporal features can determine which nodes will grow to active hubs, linking to many low-degree nodes. This information would be lost in a static network representation. From the results of the experiment as shown in Figure 2.5, we may conclude that the temporal topological features likely capture relevant temporal behavior.

Degree-preserving rewiring

By performing assortative and disassortative degree-preserving rewiring, we further substantiate that disassortative networks indeed show higher link prediction performance. Utilizing simulation, we modified many network datasets from Table 2.1 using assortative and disassortative degree-preserving rewiring, following an approach similar to the one proposed in [191]. In particular, we aim to retain the local clustering properties by not selecting two edges at random, but rather selecting two edges that are close to each other, ensuring that there are not too many triangles and, in addition to that, clustering is destructed, as this is a determining feature in link prediction.

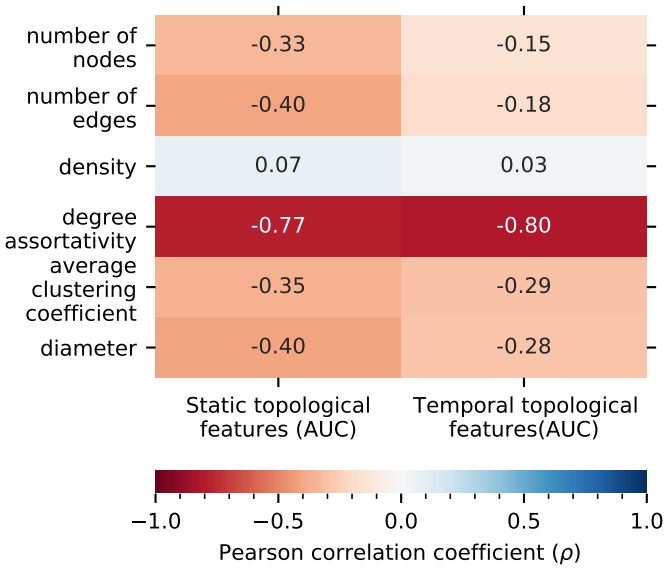

Figure 2.4: Correlations between network properties and performance (in a classifier learned only with static [Feature set I] and with temporal topological features [Feature set II-A]).

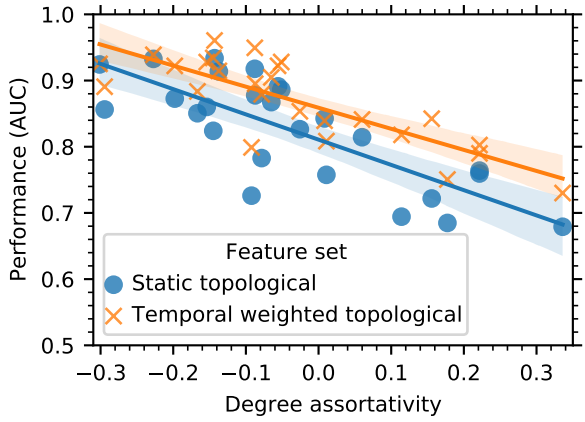

Figure 2.5: Degree assortativity and link prediction performance (in a classifier learned only with static topological features [Feature set I] and temporal weighted topological features [Feature set II-A]). (The lines indicate the relation between the network's degree assortativity and the classifier's performance, the band indicates the 95% confidence interval between the two.)

The research methodology, which we repeat several times (explained below), consists of five steps.

1. An edge (u, v) is randomly selected.
2. We randomly select a node x from the neighborhood of u.
3. We sample a node y 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.
4. We determine from the pairs of nodes (u, v), (v, y), and (x, y) which node pair has a maximum difference in degree.
5. We rewire the edges such that this pair with a maximum difference in degree becomes connected, see below.
   (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.
   (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 Steps 4 and 5 the node pair with the smallest difference in degree. The five steps are repeated $0.2 \cdot m$ times, with increments of $0.2 \cdot m$, until m.

The resulting degree assortativity values of the rewired networks can be found in Table 2.4. We observe that degree disassortative rewiring (compared to assortative rewiring) is associated with a more significant change in the degree assortativity.

We list the percentual increase in performance for both disassortativity and assortativity rewired datasets in Table 2.5. In both cases, we observe higher performance for disassortativity rewired networks, which confirms that dissassortative networks show higher link prediction performance.

Table 2.4: Degree assortativity of networks after rewiring (from degree disassortative rewiring [up to -100%] to degree assortative rewiring [up to 100%]).

	Label	-100%	-80%	-60%	-40%	-20%	0%	20%	40%	60%	80%	100%
	emails	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

Table 2.5: Link prediction performance after rewiring.

	Label	-100%	-80%	-60%	-40%	-20%	20%	40%	60%	80%	100%
	emails	-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

2.6.4 Experiment 3: Enhancement of performance with past event aggregation

In this subsection we address Experiment 3. Two sets of features are used to examine *how networks with different types of temporal information can be exploited in link prediction to improve link prediction performance*. Feature set (II-A) is constructed with past event aggregation, allowing the use of the information contained in all discrete events. Feature set (II-B) considers only the last occurring edge between two nodes, ignoring past events.

The performance obtained with these two sets of features is reported in Table 2.2. The two sets of features yield the same results for networks with persistent edges because the networks do not contain past events. In Figure 2.6, we show the difference between the two performances of the networks with discrete events in more detail.

From the results of the experiment, we may conclude that networks with discrete events perform better when aggregating past events. The result is broadly interesting 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. Interestingly, we observe significant differences in the performance improvement of past event aggregation for each discrete event network.

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, indicated with * in Tables 2.1 to 2.3) shows only a minor improvement of 0.706 to 0.760 AUC. A possible explanation is that temporal information of discrete events has limited use since it takes time to develop a successful collaboration.

On the other hand, the UC Irvine message network (indicated with ** in Tables 2.1 to 2.3) shows an improvement in AUC from 0.744 to 0.893. The improvement might be caused by the more variable nature of messages, which take only a short time to establish. In that case, the feature set with past event aggregation might provide higher scores to pairs of actively messaging nodes.

2.6.5 Experiment 4: Comparison of node- and edge-centered link prediction

Earlier, in Subsection 2.6.2 we examined whether temporal topological features improve link prediction performance. These features assume edge-centered temporal behavior. Now, in Experiment 4, we compare *the performance of edge-centered features with features that assume node-centered temporal behavior*. The link prediction of edge-centered features is done with Feature set II-A, and node-centered features are contained in Feature set III.

The results of both feature sets are presented in Table 2.2 and more detailed in Figure 2.7. We observe a strong correlation ($\rho = 0.92$, $p = 0.009$) between obtained performances using both feature sets. It suggests that most networks' temporal aspects can be modeled using either node-centered or edge-centered temporal features.

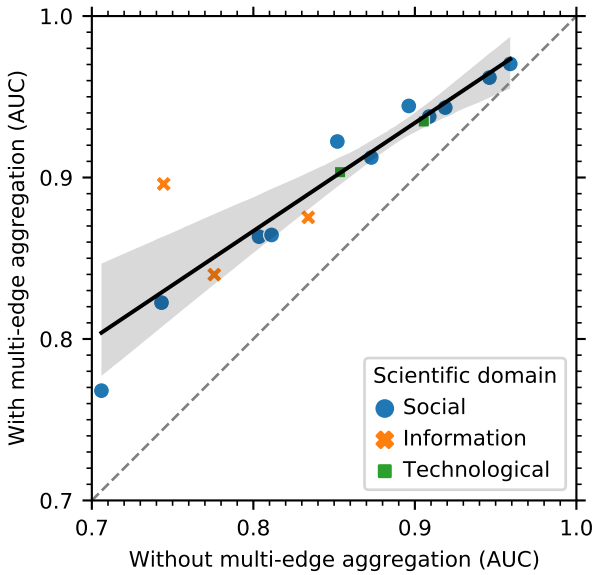

Figure 2.6: Link prediction performance with (Feature set II-A, y-axis) and without (Feature set II-B, x-axis) past event aggregation.

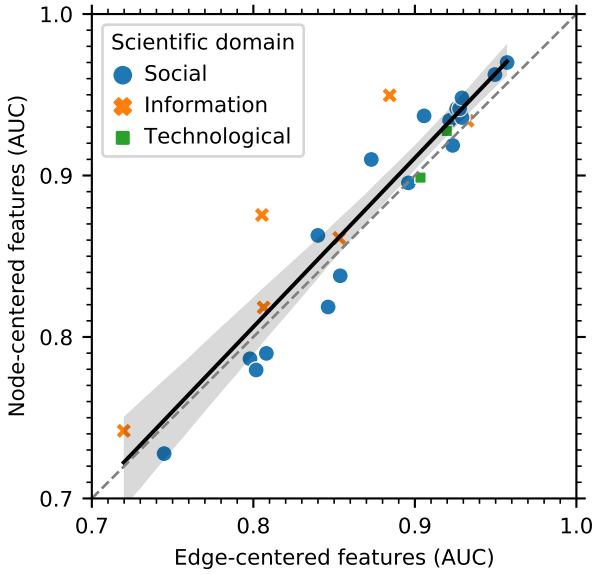

Figure 2.7: Link prediction performance of node-centered features (Feature set III, x-axis) and edge-centered features (Feature set II, y-axis). (The dotted line indicates equal performance. The solid black line indicates the best fit using linear regression. All networks are shown.)

A new indicator for further research is that for the four information networks, the performance of the node-centered features is higher than edge-centered features. This finding hints that in information networks, temporal behavior may be node-centered. Given this study's low number of information networks, further research should be conducted on a more extensive set of information networks to verify this finding.

We analyze the link performance only on pairs of nodes at a distance of two; different findings may be observed using more global features of node similarity are used. Notwithstanding this limitation, based on the current results, we may conclude that both node- and edge-centered features in supervised link predictions can achieve high performance.

2.7 Chapter conclusions and outlook

This chapter addressed Research question 1: "Wat is the relation between network structure and model performance in link prediction?" We performed a large-scale empirical study of link prediction using various structurally diverse networks. Moreover, we aimed to demonstrate the benefit of past event aggregation, allowing us to take the rich interaction history of nodes into account in predicting their future linking activity. This study resulted in four findings, that substantiate the relation between network structure and model performance in link prediction.

- Supervised link prediction performance is consistently higher when temporal information is considered (Subsection 2.6.2).
- The performance in link prediction appears related to the global structure of the network (Subsection 2.6.3). Most notably, degree disassortative networks perform better than degree assortative networks.
- We proposed an approach to deal with event-based links by aggregating information from multiple past interactions (Subsection 2.6.4). It increases the performance of link prediction. The derived feature modification steps can be inserted into any topological feature, potentially improving the performance of any supervised link prediction.
- We showed that in four information networks, features capturing node activity and static topological features outperform features that consider edge-centered temporal information, suggesting that the temporal mechanisms in these networks may reside with the nodes (Subsection 2.6.5).

Chapter outlook

The next step of this work may be to analyze 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 [63]. However, it would be interesting to investigate whether the findings presented in this chapter also hold for these types of networks.