Harnessing the Power of Ego Network Layers for Link Prediction in Online Social Networks

Mustafa Toprak©, Chiara Boldrini©, Andrea Passarella©, and Marco Conti

Abstract—Being able to recommend links between users in online social networks is important for users to connect with like-minded individuals as well as for the platforms themselves and third parties leveraging social media information to grow their business. Predictions are typically based on unsupervised or supervised learning, often leveraging simple yet effective graph topological information, such as the number of common neighbors. However, we argue that richer information about personal social structure of individuals might lead to better predictions. In this article, we propose to leverage well-established social cognitive theories to improve link prediction performance. According to these theories, individuals arrange their social relationships along, on average, five concentric circles of decreasing intimacy. We postulate that relationships in different circles have different importance in predicting new links. To validate this claim, we focus on popular feature extraction prediction algorithms (both unsupervised and supervised) and we extend them to include social circles’ awareness. We validate the prediction performance of these circle-aware algorithms against several benchmarks (including their baseline versions as well as node-embedding- and graph neural network (GNN)-based link prediction), leveraging two Twitter datasets comprising a community of video gamers and generic users. We show that social awareness generally provides significant improvements in prediction performance, beating also state-of-the-art solutions such as node2vec and learning from Subgraphs, Embeddings and Attributes for Link prediction (SEAL), and without increasing the computational complexity. Finally, we show that social awareness can be used in place of using a classifier (which may be costly or impractical) for targeting a specific category of users.

Index Terms—Dunbar’s model, link prediction, social circles, Twitter.

I. INTRODUCTION

In early 2021, the number of social media users worldwide has reached 4.2 billion, up 13% from the previous year [1]. Social media are now large ecosystems, where we gather to interact with people we already know or to join communities of people that share our interests. Clearly, being able to pair together people with similar interests is crucial for social-driven and content-driven platforms such as Facebook, Twitter, Instagram, and Youtube. This spurred intense research on link prediction, whereby algorithms suggest potential new links with users that are similar to each other. Solving this problem is equivalent to finding a needle in a haystack: online social networks (OSNs) feature millions of users, the number of potential links between them grows with the square of the number of users, but the actually existing links are quite sparse.

The link prediction algorithms can be broadly divided into feature extraction methods and feature learning methods [2]. The former are based on predefined similarity metrics/heuristics, typically computed using topological information about the network, be it local information (such as the number of common neighbors (CNs) or global one (such as the length of paths connecting two nodes). Despite their apparent simplicity, local similarity-based prediction algorithms perform generally well, even when compared with more complex approaches [3] and are also computationally efficient. The latter are new link prediction methods (leveraging graph neural networks (GNNs) [4]–[8] and network embeddings [9], [10]) that automatically learn latent features (i.e., not previously hand-engineered) that reflect the relevant structure of the graph, and, for this reason, are denoted as feature learning methods. These new methods have been shown to perform extremely well on the link prediction task [5], [9].

One of the well-known mechanisms that drive link formation is the homophily principle, which states that individuals tend to bond with those who are more similar to them [11]. Common-neighbor-based algorithms leverage this principle for predicting new links. However, findings from anthropology show that not all bonds are equal: some relationships are just acquaintances and do not imply a significant cognitive engagement. With respect to the meaningful relationships, each individual organizes them into concentric social circles where the intimacy progressively decreases from the innermost to the outermost circle [12], [13]. This model of how humans organize their relationships stems from the social brain hypothesis, which links our social life to cognitive constraints related to the size of our neocortex [14]. This hierarchical social structure is known to impact significantly on who we trust [15], on the way information spreads in OSNs [16], on the diversity of information that can be acquired by users [17], and on academic performance [18]. This layered social structure is typically represented using an ego network.
graph [12], [19]–[21], in which the individual, referred to as ego, is at the center of the graph, and the edges connect her to the peers (called alters) with which she interacts. Fig. 1 illustrates the classical ego network structure. Note that the second layer includes all the alters in the first layer, the third layer includes all the alters in the first and second layers, and so on. The ego-alter tie strength is typically computed as a function of the frequency of interactions between the ego and the alter. The ego network structure is characterized by a striking regularity: the typical sizes of the layers are 1.5, 5, 15, 50, and 150, with an approximately constant ratio of 3 between the size of consecutive layers. Online relationships have been shown to exhibit similar regularities [22], [23], thus proving that friendship links (which can be in thousands for OSNs) not associated with online interactions do not change the innate social structure of humans.

A. Motivation

Since social links are not all the same, simply considering the set of CNs (or analogous topological metrics), disregarding the importance of each social link, might not give a precise estimate of the homophily between two individuals. In fact, a common best friend is intuitively much more important than a common acquaintance when predicting the formation of new links. Thus, in this article, we set out to improve feature extraction link prediction algorithms with information about the different importance of social relationships. To this end, we exploit well-established models in the anthropology literature that describe the relative importance of our social bonds by grouping them into different social circles. Note that the social circles we refer to are very loosely related to the concept of communities in social networks. In fact, social circles are (local) groupings entirely from the perspective of the ego, they include only nodes with which the ego is directly connected, and the links between the alters (which are crucial for regular communities) do not play a role at all here. Note also that our goal is not to propose a new prediction algorithm, but to add social awareness to popular prediction solutions. To the best of our knowledge, the only other work in the literature that investigates ego networks in relation to link prediction is [24]. However, in the context of that work, an ego network is a different structure from the one studied here (only the links between the alters are considered), social cognitive models are not leveraged, and the effect of using information on the different social circles for link prediction is not investigated.

While the general problem of link prediction entails recommending any of the useful links that a user may want to create with other users, in certain applications we may want to recommend links between users that share a specific common interest together. Predicting links in such contextualized communities might be much more relevant (and valuable) than predicting links between generic users. This could be the case, for example, of a third-party service that wants to connect users interested in a specific topic. In this article, we showcase one application of this concept by focusing on a set of gamers on Twitter, and we investigate how to make new link recommendations leveraging key properties of the way users allocate their cognitive capacity to social relationships. As discussed in Section III, we believe that focusing on such a target set of users, characterized by a specific type of online engagement, is particularly significant for link recommendation in OSNs. Then, we apply the same method to generic Twitter users, to assess the efficacy of the proposed solution also beyond the contextualized communities’ settings.

B. Key Findings

The key findings presented in the article are the following.

1) In the majority of cases, regardless of the prediction approach (unsupervised or supervised), the specific heuristic or learning algorithm, and the metric (precision, area under the curve (AUC), F1 score) considered, leveraging social circles information outperforms the corresponding baseline in which circles are ignored.

2) The contribution of the different social circles to the achieved precision varies with the recommendation algorithm considered. In unsupervised similarity-based approaches, innermost circles help policies with little or no penalization of high degree nodes, while outermost circles boost the performance of algorithms relying on high-degree nodes’ penalization. The precision of supervised strategies is generally considerably increased when only the innermost circle, corresponding to the small set (composed of three to five people typically) of the most intimate relationships, is considered.

3) Supervised link prediction that uses circle-aware explicit features is able to beat feature learning link prediction algorithms such as node2vec and learning from Subgraphs, Embeddings and Attributes for Link prediction (SEAL), which exploit node embedding and GNNs, respectively. This happens when the most intimate circle is leveraged, which comprises only the two or three strongest relationships of the ego. Thus, using knowledge on a few common strong ties is more effective than complex black box approaches.

4) Leveraging social circles’ information provides the same performance as using additional classifiers on nodes, which might be impractical or costly to use.

The rest of the article is organized as follows. We review the related work in Section II. In Section III, we present the two datasets that we leverage in our analysis. The ego-aware unsupervised and supervised link prediction approaches based on feature extraction methods that we study and the feature learning methods that we use as benchmarks are presented in
Section IV. Their prediction performance is then evaluated in Section V. Finally, in Section VI we conclude the article. An analysis of the computational complexity of the proposed approach is provided in Appendix F of the Supplemental Material (SM) [25].

II. RELATED WORK

A. Dunbar’s Model and Ego Networks

According to the social brain hypothesis from anthropology [14], the social life of primates is constrained by the size of their neocortex. Specifically, for humans, the typical group size is estimated around an average of 150 members, a limit that goes under the name of Dunbar’s number. This limit is related to the cognitive capacity that humans are able to allocate to nurturing their social relationships. The 150 friends, in fact, do not include acquaintances, but only people with whom a coherent quality relationship is entertained. In a modern society, this entails, at the very least, exchanging birthday or Christmas cards every year. Unsurprisingly, humans are not fair in how they distribute their cognitive attentions among the 150 important persons they have around them. To the contrary, it is possible to group our social relationships in circles of increasing intimacy, as illustrated in Fig. 1. Typically, each of us has at least four circles of intimacy [12], [13]: the innermost one (support clique) includes our close family, the sympathy group comprises all those people whose death tomorrow would leave us deeply affected, the affinity group includes colleagues, extended family, and people you hang out with, and finally the active network is made up of all the people you meaningfully interact with at least once a year. The circles are conventionally concentric, with the inner layers contained in the outer ones. Many people also feature an additional inner layer (contained in the support clique) comprising on average 1.5 alters for which they have a very high emotional investment [23].

The social brain theory was developed for the offline world, in which relationships were nurtured via face-to-face interactions, letters, or landline phone calls. Many had postulated that this theory would not carry over to the online world, where OSN allow us to engage conveniently with a huge number of people scattered across the globe. However, Dunbar’s model stood the test of the cyberworld as well: Dunbar’s number hold for email communications [26], mobile phone calls [27], and Twitter [22]. More importantly, the same layered structure of social relationships has emerged on both Facebook and Twitter [23]. These findings are extremely important. In fact, since ego network structures are known to impact significantly on the way information spreads in OSN, and on the diversity of information that can be acquired by users Aral and Van Alstyne [17], embedding these models of human cognition into services for OSN may drastically improve the quality of service provided to the users.

B. Link Prediction

The literature on link prediction algorithms is large. Here we only summarize the main approaches and we refer the interested readers to [2], [3], [28] for detailed surveys on the topic. Following the taxonomy proposed by Mutlu et al. [2], we discuss separately feature extraction and feature learning methods.

1) Feature Extraction Methods: The similarity-based methods make up the largest class of link prediction algorithms proposed in the literature. The rationale of this approach is that nodes are more likely to form links with nodes that are similar to them. This idea is grounded in the widespread and well-documented social phenomenon of homophily [11]. Algorithms in this class differ in how they define the similarity between nodes. We can broadly distinguish them based on whether they use local information, global information, or a hybrid combination of the two (this approach is one of the least popular, and hence, we will not treat it further). For a given node pair, local similarity-based solutions rely on node neighborhood-related structural information, such as the number of CNs [29], an inverse function of the degree of CNs [30], [31], or a preferential attachment index [29]. Local similarity-based approaches are very efficient, even on large networks, due to their easy parallelization. Their main theoretical limitation is that they are able to predict only new links between neighbors-of-neighbors. Their counterpart, global similarity-based approaches, relies on metrics computed considering the whole network topology. These metrics focus typically on the possible paths inside the network, such as the shortest paths [32], paths with different lengths by means of the Katz Index [29], random walks [33], or community membership [34].

The algorithmic methods map the link prediction problem into well-known algorithmic approaches. Within this category, the classifier-based methods treat the link prediction problem as a binary classification problem [5], [35]–[37]. Each node pair can be characterized with a variety of attributes, including the similarity-based heuristics discussed above. This labeled set can be fed to virtually any classifier (such as decision trees, support vector machine (SVM), k-nearest neighbors, random forest (RF), neural networks). The advantage of this approach is that it can be extended and adapted to any new attribute that one wants to test, and that it generally significantly outperforms purely unsupervised similarity-based methods [35], [37], [38].

The preprocessing methods are considered meta-approaches, as they are intended to be used in conjunction with other algorithms, for which they provide preprocessing/prefiltering intended to remove some noise in the network. As an example, the clustering method discussed in [32] suggests the removal of the weakest links (those between nodes with few or zero CNs). The solution that we propose in this article falls into this category. The meta-approach that we investigate is based on considerations related to how people distribute their social capacity across their relationships, rather than on pure graph-related properties.

Orthogonally to the above classification, we can also distinguish between approaches that consider the weights of the links or not. Among the former, we mention [31], [39]. As discussed in [40], though, the performance of weighted indices is often worse than their unweighted counterpart. For this reason, in this work we do not consider the weight of links besides what is needed to compute the ego network structure.
3) Feature Learning Methods: The feature learning methods map the graph into a low-dimensional feature space. The difference between the feature extraction and feature learning methods is that in the latter, the features are learned by the system and not hand-engineered. The mapping, or graph representation, can be learned and optimized via both supervised and unsupervised methods [2]. The random walk methods use graph exploration methods such as breadth-first search (BFS), depth-first search (DFS), and random walks to capture features and node properties such as centrality, being a hub, or community membership. DeepWalk [42] and node2vec [9] are the most popular methods of this category. For the link prediction task, the node feature vectors (node embeddings) are transformed into edge feature vectors (edge embeddings) using operators such as the Hadamard product or cosine similarity. Extracted edge embeddings are fed to learning algorithms (such as SVM, regression methods, deep neural networks) to train a model that is used to predict future links. These methods are considered semisupervised, since the information on the existence/nonexistence of a link is already present in the studied graph.

The GNN models leverage natural networks to map the graph structure to a low-dimensional vector space. Graph differentiable pooling, graph autoencoders, and GNNs have been used for the link prediction algorithms based on neural networks. In this study, we compare against the popular GNN-based link prediction algorithm SEAL [5], which relies on a convolutional neural network (CNN) architecture (deep graph CNN (DGCNN), to be specific). Note that the network embedding methods and GNNs are highly related to each other [43], as they both map the network to a lower dimensional representation space. However, GNNs allow for greater flexibility, by leveraging node features in the encoder, by sharing parameters between nodes in the encoder, and by fixing the transductivity problem of shallow embeddings [44]. From the link prediction perspective, GNNs typically provide end-to-end frameworks (i.e., the learned presentation is optimized for the link prediction problem), while graph embedding methods capture topology-level properties that are then used to train a separate classifier for link prediction.

III. DATA PREPROCESS

As discussed in Section I, in this article we investigate social circle-aware link prediction considering both a community of users with shared interests (gaming, in our case) and generic users. Before presenting the algorithms for predictions in Section IV, here we first introduce the datasets that we use as case study. Without loss of generality, some definitions required in the prediction algorithms will be discussed with reference to these datasets. However, note that each such definition can be used as it is for any other dataset describing different communities, and the algorithm works unchanged in all such cases.

In the gaming-related dataset, we consider as shared topic of interest “indie games.” An independent video game, or indie game, is a video game that is often created without the financial support of a publisher, although some games funded by a publisher are still considered “indie.” Indie games often focus on innovation and rely on digital distribution. Growing a community of interested users is thus a critical success factor for them. The goal of our link prediction task is to recommend users interested in indie games to other users interested in indie games. Our gaming-related dataset consists of 8932 users (labeled as gamers) engaging in game-related conversations. We have collected (June 2018) their timeline (most recent 3200 tweets) using the Twitter Search Application Programming Interface (API). Details on the collection process are provided in Appendix A of the SM [25]. In the generic user dataset, we focus on a group of users whose interests are not associated with a specific domain. To this aim, we use a dataset presented in [45], containing the timelines (most recent 3200 tweets) of 1930802 Twitter users obtained by means of a snowball sampling starting from U.S. President Barack Obama in November 2012. Refer to [45] for further details on the collection process.

Note that we use two Twitter datasets because Twitter is especially amenable to computation of ego networks: social interactions are typically public and can be easily downloaded. For this reason, the majority of the literature on online ego networks (Section II-A) leverages Twitter data. Facebook interactions would be similarly suitable but they are not public, hence could not be accessed. The social network graphs available to the research community do not typically include information on the frequency of interactions, and thus ego networks cannot be extracted.

A. Extracting the Ego Networks

We filter the 8932 gamers and 1930802 generic users and extract their ego network (including their social circles) using the same methodology as in [46], which we briefly recollect here. We simply need a weighted social graph, whose edge weights correspond to the contact frequency between the corresponding nodes. To extract reliable information, we filter users according to the same policy used in [46]. We only consider users whose Twitter activity is regular (i.e., they post, on average, at least one tweet every three days for at least 50% of the total number of months of their activity) and stationary (i.e., they are not in their initial stage of engagement with the Twitter platform, which typically features a transient spike of activity). The strength of ego–alter relationships is inferred from the frequency of direct tweets (mentions, reply, or retweets) between the ego and the alters. The optimal number of social circles per users is obtained using the mean shift clustering algorithms to group such frequencies. The code for computing the ego networks can be found at https://egonetworks.readthedocs.io/en/latest/.

Approximately one third of the gamers passed all the filters described above, and this left us with 3061 gamers with reliable ego network information. On the other hand, approximately 7% of the generic users passed all the filters, amounting to 148 105 generic users with reliable ego network information. We further filter them as suggested in [29] by considering only nodes that are connected to the giant component. Also, to have a generic user dataset with a similar size as that of
the gaming-related dataset and to reduce its computational intensity, we sampled 3000 egos from the giant component of the generic users network by snowball sampling [47].

We denote the set of nodes for which we have ego networks as $V_e$. In the gaming-related dataset all egos are gamers. Alternately, in the generic users’ dataset, all egos are generic users. An exploratory analysis of the ego networks in our datasets, which is substantially in agreement with the results from the related literature [23], [46], can be found in Appendix C of the SM [25]. Note that when we focus on an ego $i$, its alters can now be associated with the social circle of $i$ to which they belong. Note also that we do not typically have the ego network of the alters. In the gaming-related dataset, the alters of ego $i$ can be generic users, other gamers, or even games. This implies that some nodes are domain-specific (in our case, gaming-related), while others are generic. Domain-specific nodes are homogeneous, since they share a common interest. Summarizing, we have three classes of nodes in the gaming-related dataset: gamer users with ego network ($V_e$), regular (unlabelled) users without ego network ($V_o$), and domain-specific users that may or may not have an ego network ($V_d$), including games and gamers for which ego network information is not available or not reliable because they did not satisfy the filters discussed previously. On the other hand, in the generic user dataset, all alters of ego $i$ are also generic users. Generic nodes make prediction more challenging with respect to the case of homogeneous nodes (as, e.g., in [29]), because they tend to differ more from each other. In the generic user dataset, we can identify only two classes of nodes: generic users with ego network ($V_e$) and generic users without ego networks ($V_o$) that are basically alters.

**B. Dataset Summary**

The final datasets are summarized in Tables I and II. The imbalance ratio, i.e., the ratio between the negatives (potential links that do not exist in practice) and the positives (links actually existing) in the class of both gamers and generic users, is around $10^{4}:1$, which is quite high. This is a well-known problem in link recommendation for OSNs (remember from Section I that links are sparse). This implies that special care should be taken in the evaluation, as explained in detail in Section V-B.

### TABLE I

| Gaming-Related Dataset |
|-------------------------|
| # of gamer nodes $|V_e|$ | 2,995 |
| # of domain-specific nodes $|V_d|$ | 70,859 |
| # of all nodes $|V|$ = $|V_e|$  $∪$ $|V_d|$ | 470,485 |
| # of gamer edges $|E_e|$ | 2,614 |
| # of domain-specific edges $|E_d|$ | 154,581 |
| # of edges $|E|$ | 1,004,011 |

### TABLE II

| Generic Users’ Dataset |
|-------------------------|
| # of ego nodes $|V_e|$ | 3000 |
| # of all nodes $|V|$ = $|V_e|$ $∪$ $|V_o|$ | 278,510 |
| # of ego edges $|E_e|$ | 7,158 |
| # of edges $|E|$ | 567,739 |

IV. LINK PREDICTION BASED ON SOCIAL CIRCLES

We denote with $G = (V, E)$ the graph modeling the relationships in our dataset. Since our goal is to evaluate the contribution of social circles to link prediction, we will focus on the subset of nodes in $V$ for which social circles can be computed. Thus, predictions will be made for the set of possible edges between nodes in $V_e$.

A. Similarity-Based Unsupervised Learning With Circle Awareness

To showcase the effect of social circles for unsupervised link prediction, we focus on the simple, yet effective class of similarity-based approaches. Like for all the similarity-based link prediction algorithms, the goal of the proposed algorithm is to associate each nonexisting link between two users $i, j$ in $V_e$ with a score proportional to the likelihood that the link will actually be formed in the future. Hence, the goal of the link predictor is to associate with each edge in $V_e$ × $V_e$ − $E_e$ a probability/confidence that the link will indeed appear in the future. Qualitatively, this is equivalent to suggesting users to other users for possible interactions, friendships, and so on.

To study the effect of social circles on link prediction, we propose to slice the social graph based on the membership to a specific circle (recall that our concept of social circles is different from that of communities). Let us focus on a user $i$ in $V_e$. We denote with $C_1(i), C_2(i), \ldots,$ the sets comprising the neighbors of $i$ that belong to $i$’s first, second, $\ldots$, social circle, respectively. We have discussed how to obtain these circles in Section III-A. Thus, the graph $G$ sliced according, e.g., to circle $C_3$, is the graph including only links between egos and their alters up to layer 3 (remember that $C_3$ includes $C_1$ and $C_2$). More formally, it can be defined as $G_{C_3} = (V, E_{C_3})$, where $E_{C_i} = \{e_{ij} \in E : i \in V_e, j \in C_i\}$. Note that the slicing introduces asymmetry in the graph (e.g., a relationship $i, j$ can be in $C_4$ for ego $i$ and in $C_3$ for ego $j$). However, it is exactly this edge filtering that helps link prediction, as we will show in Section V.

Social circles’ slicing can be extended to also include domain-based slicing for domain-specific datasets. For example, from $G_{C_3}$, we can retain only domain-specific nodes and edges (i.e., nodes in $V_e$ $∪$ $V_d$ and edges in $E_e$ $∪$ $E_d$). We generalize the notation related to slicing preprocessing by denoting with $\omega$ the specific slicing considered, and with $G_{\omega}$ the resulting graph. In Section V, we will discuss the performance of link prediction both with social-based slicing and with social-based plus domain-based slicing. We anticipate here that when the social circles’ information is not used for prediction, leveraging the category information provides a significant advantage. Conversely, social circles’ awareness makes information on the category less relevant (and this is extremely important, since the extraction of categories typically requires an additional, domain-specific, classifier, or manual labeling).
To perform unsupervised link predictions, for each user–user pair \( i, j \) for which a link does not exist in \( E_{\text{ev}} \), we follow Definition 1 below. Note that each slicing will yield different predicted links. Hence, we will treat each slicing choice as a separate link prediction approach.

**Definition 1 (Unsupervised Circles-Aware Link Prediction):** For a fixed \( K \), the social circles’ aware link prediction algorithm suggests to a user \( i \), \( K \) users \( j \) (with \( j \in V_e \cap (i, j) \notin E_{\text{ev}} \)) associated with the top-K \( \omega_{i}(i, j) \) values, where \( \omega \) denotes the selected slicing and \( \sim_{\omega}(i, j) \) is the similarity computed on \( G_{\omega} \).

The definition of a new similarity function is out of the scope of the article. What we want to do here is to evaluate popular similarity functions available in the literature when they are enriched with knowledge about the ego network circles. To isolate the effect of social circles’ awareness, we need approaches that are simple (so it is easy to gauge the role of social circles) yet effective. For these reasons, our choice fell on the strategies described below, often used in the related literature. In summarizing them, we denote with \( \Gamma_{\omega}(i) \) the neighborhood of node \( i \) in \( G_{\omega} \).

1. **CNs** [29]: the similarity is given by the number of CNs between users \( i \) and \( j \) in \( G_{\omega} \).
   \[
   \sim_{\omega}(i, j) = |\Gamma_{\omega}(i) \cap \Gamma_{\omega}(j)|. \tag{1}
   \]

2. **Jaccard’s Coefficient (JC)** [29]: the similarity is computed as the Jaccard similarity of the set of CNs of users \( i \) and \( j \) in \( G_{\omega} \).
   \[
   \sim_{\omega}(i, j) = \frac{|\Gamma_{\omega}(i) \cap \Gamma_{\omega}(j)|}{|\Gamma_{\omega}(i) \cup \Gamma_{\omega}(j)|}. \tag{2}
   \]

3. **Adamic-Adar (AA)** [30]: the AA similarity reduces the importance of CNs having high degree
   \[
   \sim_{\omega}(i, j) = \frac{1}{\log(|\Gamma_{\omega}(i) \cap \Gamma_{\omega}(j)|)}. \tag{3}
   \]

4. **Resource Allocation (RA)** [31]: the RA score is similar to the AA one, but it penalizes even more the CNs with high degree
   \[
   \sim_{\omega}(i, j) = \sum_{z \in \Gamma_{\omega}(i) \cap \Gamma_{\omega}(j)} \frac{1}{|\Gamma_{\omega}(z)|}. \tag{4}
   \]

Among these four link prediction algorithms, RA is the one that consistently performs better in the related literature [39], [40], [48]. In Sections V-D and V-E, we will investigate whether this is still the case when the policies can leverage the knowledge of the social circles.

### B. Supervised Learning With Social Circle Awareness

Each topological metric discussed in Section IV-A captures a single possible mechanism yielding to the formation of new links in the network. Lichtenwalter et al. [49] discusses how this unsupervised approach can take advantage of adding supervised learning on top of it. They show that simple ranking of heuristics is outperformed by supervised classifiers, because the latter has the capability of identifying multiple differentiating boundaries in the similarity score domain, even when just a single heuristic is used as feature.

Thus, in this section, we cast our social-aware link prediction problem into a supervised learning problem. This entails computing a vector of features for each user pair \( i, j \). Each of the metrics in Section IV-A becomes a feature that describes the user pair. Each pair is also labeled to mark whether the link exists or not. To test the different performance of different supervised approaches, in this work we consider the following learning algorithms in parentheses we report the link prediction papers in which they have been previously used): logistic regression [50], RF [49], decision trees, naïve Bayes (NB), and SVM [35], the latter with both linear and polynomial kernels. We use their R implementations (glm, randomForest, rpart, klaR, kernlab, respectively) together with the caret package for training and test. Parameter optimization is applied on the training set with 10-fold cross validation.

#### C. Prediction Based on Feature Learning Methods

In the literature, there are newly suggested link prediction methods based on learning latent features of the graphs, where these feature vectors are low-dimensional vector representations produced by approaches such as graph representation learning [51] and GNNs [43]. For a detailed description and classification of the methods, the reader may refer to [2], [43], [51], [52] and Section II-B.1. In this work, we select two feature learning algorithms to be used as benchmarks for comparison with our social-aware approach. Specifically, we have selected one of the most popular link prediction methods, node2vec [9], and one of the most popular GNN-based link prediction methods, SEAL [5]. For details on these algorithms and their settings, refer to Appendix B of the SM [25]. Note that differently from Sections IV-A–IV-B, the selected feature learning approaches are not modified to include social awareness. In fact, the key idea of feature learning is to autonomously learn the important graph features. Thus, our objective, in this case, is to assess whether the automatically learned features are better than the social-aware hand-engineered ones in predicting new links.

#### V. Evaluation

In this section, we carry out the performance evaluation of the link prediction approach proposed in Section IV using the Twitter datasets described in Section III.

#### A. Training and Test Data

For obtaining longitudinal data, we downloaded (December 2019) the Twitter timeline of the gamers in \( V_e \) one year and a half after the initial download and the timeline of the generic users in \( V_e \) eight years after the initial download. We then identified the links between gamers in the gaming-related dataset and between generic users in the generic users’ dataset that have appeared in the meanwhile. These new 843 links between gamers and new 1216 links between generic users constitute the set \( E_{\text{new}} \) of links to be
predicted (i.e., the test set), while $E_{\text{old}}$ contains the links existing in the first temporal snapshot (corresponding to the $E_e$ sets in Tables I and II, so $E_{\text{old}} = E_e$ for each dataset). The edges in $E_{\text{old}}$ are used as positives for computing the heuristics in unsupervised link prediction and as positive training set for supervised link prediction.

While the above considerations are sufficient for unsupervised link prediction, with supervised learning we also need to assign the negatives (i.e., missing links) to the train and test sets. To this aim, we split them 90%–10% between train and test. Clearly, the set of edges selected for test set may offer only a partial view of the performance (i.e., they may be easier or harder than average to predict). For this reason, we perform a $k$-fold cross-validation [28], with $k = 10$, each time selecting a new 10% of the negative links for the test set. The performance metrics are aggregated via microaveraging [53].

Using 90% of negatives for training entails training supervised schemes on millions of negatives and only thousands of positives (Tables I and II). While the impact of class imbalance is typically limited for similarity-based unsupervised approaches like the ones discussed in Section IV-A, it can create serious problems for the scalability and reliability of supervised learning. To mitigate the problem, the common approach in the related literature [5], [35], [54] is to undersample the negative class. Despite its widespread use, this technique is not without drawbacks [55]. To provide reliable measurements, we will show both the results obtained with undersampling (Section V-E) and the results obtained with complete negative class (Appendix E of the SM [25]).

### B. Evaluation Metrics

Similar to the related literature [28], [29], [39], unsupervised link prediction algorithms are evaluated using a top-$K$ analysis, i.e., we compare their performance in predicting $K$ new links. This allows for a fair comparison among the different approaches, as it avoids fixing a similarity threshold for approaches in which the same threshold may have a different meaning. To span a reasonable $K$ range around the number of positives in the test set (which, as we discussed in Section V-A, are 843 and 1216 for the gaming and generic users’ datasets, respectively), we consider $K \in \{100, 843, 1000\}$ for the gaming-related dataset and $K \in \{100, 1216, 1500\}$ for the generic users’ dataset. For a given slice $\omega$ and for a fixed $K$, each unsupervised link predictor described in Section IV-A outputs a list $L^K_{\omega}$ of $K$ pairs in $V_e \times V_e - E_{\text{old}}$, which are the newly predicted links (each with its associated confidence in prediction, which, in our case, is the similarity value). Then we can compute classic metrics such as

1. $TP = |L^K_{\omega} \cap E_{\text{new}}|$
2. $FP = |L^K_{\omega} - E_{\text{new}}|$
3. $FN = |E_{\text{new}} - L^K_{\omega}|$
4. $TN = |V_e \times V_e - E_{\text{old}}| - |L^K_{\omega} \cup E_{\text{new}}|.$

Supervised approaches automatically label all the elements in the test set (without the need to set $K$ or a similarity threshold), hence the number of positives predicted is part of the workings of the supervised approach. In this case, thus, we do not pick the top $K$ potential edges, but we directly rely on the labeling of the trained predictor for the test edges. It is then straightforward to compute the above metrics (e.g., true positives (TPs) are those edges in $E_{\text{new}}$ that are marked as new by the supervised prediction algorithm).

The related literature on evaluating systems with class imbalance suggests using metrics such as precision ($=TP/TP + FP$), recall (aka TPR), and $F_1$ score. As already argued by Wang et al. [54], in the context of link prediction, precision is more important than the other metrics (such as recall), because if the precision is high, one can live with some false negatives (FNs). Therefore, the focus of this evaluation will be mostly on precision. For the completeness of results, we also provide, when relevant, the AUC of the precision-recall curve. The AUC provides an aggregate measure of performance across all possible $K$ classification threshold, and hence it complements the results obtained for fixed $K$s. The results for the F1 score can be found in Appendix E of the SM [25].

Precision and recall are evaluated based on metrics (TP, false positive (FP), true negative (TN), FN) which corresponds to specific realizations of a target phenomenon. Thus, the question arises of how confident we can be about the results obtained on a particular collection that is the result of random sampling (as in the case of cross-validation or subsampling of negatives for supervised prediction). To quantify this confidence, we use credible intervals [56]. For further details, please refer to Appendix D of the SM [25].

### C. Experimental Setup

We are interested in comparing the performance obtained using the baseline approach (no social circles information, all edges are considered) and the other strategies in which only edges belonging to the ego network circles are considered. The majority of egos in our datasets feature approximately five circles in their ego networks (Appendix C of the SM [25], Fig. A.1). For this reason, in our circle-based slicing we will consider circles from C1 to C5, and then we group together into the ACTIVE circle all the circles beyond C5 (for those nodes that have more than five circles). Note that since social layers are concentric and the $i$th also includes the $(i - 1)$th up to the first one, for egos with less than five circles the predictions in, e.g., C5, are simply based on the last nonempty social layer (which includes all the innermost ones). Recall that the alters in the ACTIVE circle are all those with a significant relationship with the ego, i.e., they interact with the ego at a frequency of at least one contact per year [13]. We denote with ALL the situation in which all relationships are considered, which implies considering both the nodes in the ACTIVE circle (significant bonds) and those outside (acquaintances). Note that for each strategy, the ALL case corresponds to the baseline from the related literature: all relationships are treated as equal, without factoring in the role of social circles. As anticipated in Section IV, we will consider two scenarios. In the first one, denoted as ALL EDGES, we consider the full graph $G$, slicing it based on the social circles both in the
Fig. 2. Precision in unsupervised settings—ALL EDGES. (a) Gaming-related dataset. (b) Generic users' dataset.

The precision for varying $K$ (number of new links recommended) is shown in Fig. 2(a). The maximum precision across all policies decreases when we increase $K$: the more the links recommended, the more the mistakes (as it is generally the case in the related literature). RA clearly outperforms the others in the gaming-related dataset, and this is consistent with previous findings in the related literature [40]. RA seems to benefit significantly from circle awareness. Specifically, ignoring the outer circle gives a clear advantage to RA, which at least doubles its precision for all $K$ values. Thus, the mechanism whereby high-degree penalization and social awareness work in hand in hand is very effective from a link prediction standpoint. It is also interesting to note that this advantage seems lost when only the innermost circle is considered. However, leveraging only the innermost circle $C_1$ means using very little information (only a few links per ego, as can be seen in Fig. A.2, Appendix C of the SM [25]). This can be an advantage in case of resource limitations (in terms of computational time, computing the similarity is faster on smaller neighborhoods, as we discuss in Appendix F of the SM [25]). In such cases, we can have a good prediction (still better than that at ALL) using only $C_1$. Thus, the predictive power of using only the most intimate relationships unexpectedly outperforms baselines relying on all relationships. The drastically different performance of AA with respect to RA in Fig. 2(a) tells us that the way we penalize high-degree nodes impacts on the effect that social circle information has on link prediction effectiveness. However, it is interesting to note that while in general RA is better than AA, this is not the case when resource constraints are considered. When only the most intimate links are kept in the ego networks, AA performs at its best and its precision is not only equivalent to that of RA but also comparable to the precision of RA in the baseline (ALL). In this case, then, strong degree penalization seems to be less important than strong intimacy. The CNs’ policy performs quite similar to AA, even if, in CN, all the neighbors contribute the same to the similarity score, regardless of their degree. This is evidence that here AA is working in a regime in which all nodes weight approximately the same (i.e., its logarithmic degree penalization is not enough). Given the similarity with AA, the same considerations we made for AA hold for CN. Finally, we highlight the poor precision of the Jaccard-based strategy, for all $K$s and for all circles in the gaming-related dataset. Jaccard similarity gives more weight to neighborhoods that are very similar to each other. This means that not only the overlapping part is considered but also the number of nodes that are not in common (union of neighbors). As testified by the plots, this restriction does not give an advantage for link prediction in the gaming-related dataset. With respect to social circles’ awareness, Jaccard seems to suffer from using only the intimate relationships, while using the ACTIVE layer yields the same precision as the baseline ALL while saving computational resources.

We report in Table III the area under the prediction–recall curve. Recall that the AUC is not dependent on $K$, since it is obtained by exploring the whole range of similarities for making recommendations. When considering the relationship between precision and recall through AUC, we observe that the best overall results (values underlined in the table) are achieved by RA that leverages $C_4$ and $C_5$. Looking at the layers that, for each policy, provide the best AUC, we note that the ALL case (which is the baseline for similarity-based policies from...
the related literature) is never the best performing. From the circle awareness standpoint, there seem to exist two classes of policies, those that benefit most from using the outermost circles and those that benefit from the innermost ones.

2) The AllEdges Case With the Generic Users' Dataset: Similar to the gaming-related dataset, we observe that the maximum precision across all policies decreases when we increase $K$ [Fig. 2(b)]. In this case, though, the baseline results (i.e., those without social awareness) are different: while for the gamers' network, RA was already the best predictor in the ALL case, here we observe a generalized advantage of the JC approach for all $K$ values. Given this different starting point, let us study how the prediction policies react to circle awareness. RA preserves the characteristics we observed for the gamers and still very much benefits from social awareness: all the circle-based scenarios have better or comparable precision than the baseline ALL. Note that as mentioned in the gaming-related dataset results, even if the prediction precision is similar—as it is the case in Fig. 2(b) for C1 and ALL with large $K$—using the most intimate links in C1 gives us the advantage of smaller computation time against traditional methods (see Appendix F of the SM [25] for a thorough discussion). Turning our attention from RA to AA, Fig. 2(b) shows that AA performs significantly better compared with what we observed for the gaming-related dataset: here, AA substantially mimics the performance of RA, both in terms of achieved precision and in terms of social circles in which it performs best (specifically, around C4). Interestingly, AA was mimicking instead the performance of CN in the gaming-related dataset, with peak performance in C1 and a generally poor precision both in the other social circles and in the baseline ALL. The fact that this difference is due to the different structural properties of the generic users' graph with respect to the gamers one follows directly from the definitions of RA, AA, and CN in Section IV-A. Indeed, all the three policies are based on the principle that the more the CNs, the better the link prediction. Then, in CN all CNs weight the same, while in AA and RA the CNs with high degree weight less (much less in RA than in AA, since in RA the penalization is linear while in AA it is logarithmic) than those with small degree. The fact that the performance of AA is approaching that of RA implies that the degree of the CNs is not very high (hence, the logarithmic penalization of AA and the linear one of RA are in a regime in which they yield similar scores). Taking into account the degree of

| AUC (×10) |
|---|
| C1 | C2 | C3 | C4 | C5 | ACTIVE | ALL |
| RA | 0.0179 | 0.0316 | 0.0469 | 0.0536 | 0.0516 | 0.0364 | 0.0191 |
| AA | 0.0214 | 0.0074 | 0.0103 | 0.0116 | 0.0127 | 0.0137 | 0.0112 |
| CN | 0.0194 | 0.0054 | 0.0076 | 0.0092 | 0.0100 | 0.0114 | 0.0092 |
| JC | 0.0033 | 0.0044 | 0.0064 | 0.0083 | 0.0092 | 0.0100 | 0.0096 |

the CNs still provides an advantage, though, as highlighted by the worse performance of CN with respect to AA and RA. As a final remark, note that while JC is performing extremely well with the generic users' dataset in the baseline ALL, its performance rapidly deteriorates as we incorporate circle information. This tendency was already present in the gamers’ datasets [Fig. 2(a)] but was somewhat masked by the generally poor precision achieved by JC regardless of the circle/baseline considered. It is interesting to speculate on why circle awareness does not help JC in general. To this aim, let us consider the definition of the Jaccard similarity in Section IV-A. JC captures the fraction of CNs out of all neighbors for a pair of users $i, j$. Thus, its very nature requires “useless” neighbors (i.e., those that are not in common) to be present, to differentiate between high and low neighborhood overlap. However, slicing based on social circles effectively remove “useless” nodes, hence impairing the discriminating capabilities of the JC score.

In Table IV, we report the area under the prediction–recall curve for the generic users’ dataset. Again, recall that the AUC captures the tradeoff between precision and recall and that it is not dependent on the specific $K$ value considered, as it spans the whole $K$ range. The best overall precision–recall performance is achieved by RA when it leverages social information in C5. The second best AUC is provided by JC in the baseline case (no circle-awareness). AA and CN provide the highest AUC when leveraging C5 information. Overall, from the results in Table IV, we can thus conclude that even though the advantage of circle awareness is less evident in the generic users’ dataset, it still outperforms noncircle-related approaches.

3) The DomainSpecificEdges Case: Recall that in this case, for any social-based slicing, we also remove all nodes (and associated edges) that are not domain-specific (this is only relevant to the gaming-related dataset). All the nodes left after this additional, category-based slicing are either gamers or games, i.e., they belong to the specific community for which we are making link recommendations. Making a link prediction is now much easier, because the network has been pruned by nodes potentially irrelevant or misleading. Indeed, as shown in Fig. 3, the precision of AA and CN significantly improves with respect to the AllEdges scenario in Fig. 2(a). This improvement is also present when no social circle information is used [ALL circle in Fig. 2(a)]. This implies that when social

| AUC (×10) |
|---|
| C1 | C2 | C3 | C4 | C5 | ACTIVE | ALL |
| RA | 0.0302 | 0.0575 | 0.1028 | 0.1423 | 0.1514 | 0.1394 | 0.0931 |
| AA | 0.0273 | 0.0498 | 0.0793 | 0.1135 | 0.1207 | 0.1003 | 0.0627 |
| CN | 0.0110 | 0.0191 | 0.0393 | 0.0467 | 0.0471 | 0.0417 | 0.0308 |
| JC | 0.0092 | 0.0173 | 0.0357 | 0.0782 | 0.1073 | 0.1249 | 0.1439 |
circle awareness is not used for prediction, leveraging the category information provides a significant advantage. Alternately, social circles’ awareness seems to make information on the category unnecessary. Since the extraction of user categories typically requires a domain-specific classifier or manual labeling, being able to skip this phase without affecting the link prediction quality would be very important. The precision of the Jaccard-based approach remains very low also when considering domain-specific edges only. We conclude this part on the DOMAIN SPECIFIC EDGES scenario by analyzing the AUC of the precision–recall curve, reported in Table V. Table V shows that, also in this case, the best performance is achieved by RA in the outermost circles (specifically, ACTIVE and C5). We observe again the duality of behaviors with respect to the gain from social circles’ awareness: RA and AA achieve their best AUC in ACTIVE, CN and JC in C1. The baseline ALL is never optimal (neither from an absolute standpoint nor for specific policies). This further confirms the advantage of using social circle awareness in general.

E. Evaluation of Supervised Link Prediction

In this section, we assess the advantages brought about by including a supervised classifier in the link prediction approach. As discussed in Section IV-B, our classifier uses as features the similarity-based metrics introduced in Section IV-A. The supervised approaches that we test are logistic regression, decision trees, NB, RF, and SVM. Recall (from Section V-A) that with supervised learning we investigate the performance both on the original dataset and on the undersampled one. Also, with supervised learning we do not carry out a top-\(K\) analysis but we allow the algorithms to freely classify all the edges in the test set.

Figs. 4(a) and (b) and 5 show the results for the ALL EDGES and the DOMAIN EDGES case, respectively. Recall that, as explained in Section V-A, these results are obtained with the negatives undersampled to make the positive and
negative classes balanced. In Appendix E of the SM [25] we also investigate the link prediction performance when no undersampling is carried out (the findings are substantially confirmed). Fig. 4(a) shows the precision of the supervised link prediction approaches in the ALL EDGES scenario of gaming-related dataset. We can identify two important results. First, all approaches that rely on C1 information provide better precision than with any other layer. In particular, using all edges (case ALL) is never better than using C1 alone. Second, all circle-based approaches that leverage information on the outermost social layers perform at least as well as the baseline (ALL). When considering the generic users’ dataset [Fig. 4(b)], we observe that the peak performance in C1 is confirmed for RF and NB. However, for the other learning strategies, the precision drastically drops in C1. For generic users, we can conclude that all the social circles but the first one provide approximately the same precision as the baseline ALL. Finally, note that for both the datasets, the precision achieved with supervised strategies is much higher than that obtained with unsupervised ones (consistently with the results in the related literature [54]).

In the DOMAIN SPECIFIC EDGES scenario for the gaming-related dataset (Fig. 5), again NB and RF achieve the highest precision when leveraging only C1 relationships. Differently from the previous case, the precision of decision trees in C1 is lower than with other layers. The remaining learning strategies still gain from using C1 information, but the margin is smaller in this case. Differently from the unsupervised case, supervised learning seems able to overcome the difficulties of predicting new links in the ALL EDGES scenario, and hence the relative advantage of filtering out links not domain-specific is partially lost.

In summary, supervised learning consistently yields better predictions than their unsupervised counterparts. This may be due to more flexible identification of boundaries between positives and negatives (with respect to the simple threshold-based approach of similarity-based heuristics) or to a smart combination of the similarity-based heuristics. Regardless, social circles again prove very effective in boosting the precision of link prediction strategies, especially in the gaming-related dataset. In particular, supervised strategies seem able to effectively exploit the innermost layer C1 in the domain-dependent dataset and the active network layer in domain-independent dataset much better than the unsupervised cases.

### F. Comparison Against Link Prediction Methods Based on Feature Learning

Until now, we have studied the social-aware link prediction method by embedding the intimacy levels of Dunbar’s ego network model into the existing unsupervised and supervised approaches. The ego network model is a way of capturing the local information of relationships of the individuals. Since the aim of the study is to understand the contribution of the social circles to the performance of the link prediction methods, we selected prediction methods where the feature calculation is not a black-box, and the link prediction method is not an end-to-end method. All the methods discussed so far were based on explicit graph features. In this section, we compare the prediction performance of the methods based on latent features—node2vec, leveraging shallow node embeddings, and SEAL, which is GNN-based (see discussion in Section IV-C)—to feature extraction methods leveraging social circle information.

Table VI shows the precision results of node2vec (for the best \( p, q \) pair), SEAL (whose best precision is achieved with \( h = 1 \) without embeddings), and the best supervised social-aware algorithms (whose best results always occur with C1 on the undersampled\(^2\) generic users’ and gamers’ datasets. Recall that we are using node2vec and SEAL approaches as baseline methods where all edges are included without any slicing (this corresponds to the case ALL in Section V-E). Note that this comparison is fair, since we are studying machine-learning-based algorithms trained on explicit features against a machine-learning-based algorithm trained on latent features. Contrarily, we do not consider the unsupervised approaches from Section V-D as they do not leverage machine learning techniques. Table VI shows that the best circle-aware link prediction methods always outperform latent feature–based algorithms. Looking at Figs. 4(a) and (b) and 5 for a circle-by-circle comparison, we observe that either or both feature learning strategies always provide the best precision in the ALL case. As expected, feature learning strategies are also generally very competitive in the other cases, outperforming circle-aware approaches for several slicing. However, they are never able to surpass the precision achieved by the best circle-aware strategy.

The implications of the above results are far-reaching: using knowledge on a few common strong ties, circle-aware link prediction consistently beats black-box approaches. This advantage is not even paid in terms of computational complexity: as we show in Appendix F of the SM [25], all the strategies in Table VI are linear in the number of edges.

|               | Generic Users | Gamers       |
|---------------|---------------|--------------|
|               | ALL EDGES     | DOMAIN EDGES |
| node2vec \( (p = 1, q = 4) \) | 0.955 | 0.894 |
| node2vec \( (p = 0.25, q = 0.5) \) | 0.926 | 0.920 |
| SEAL \( (p = 0.25, q = 4) \) | 0.920 | 0.931 |
| circle-aware link prediction | 0.974 | 0.945 |
| \( \text{C1}; \ RF \ & \ NB \) | 0.985 | 0.985 |
| \( \text{C1}; \ NB \) | 0.985 | 0.985 |
| \( \text{C1}; \ RF \) | 0.985 | 0.985 |

\(^2\)For consistency with the original node2vec and SEAL papers, where the negatives are undersampled in the evaluation.

### VI. Conclusion

In this article, we have studied the performance of circle-aware feature extraction link prediction algorithms.
Specifically, relying on very well-established models from anthropology, we have considered the social circles in individual ego networks, using the circle as a proxy of intimacy. We have selected four benchmark heuristics and we have modified them to include awareness of the social circles. Our results show that social-circle-based link prediction is generally extremely effective. Specifically, in the majority of cases, regardless of the prediction approach (unsupervised or supervised), the specific heuristic or learning algorithm, and the metric (precision, AUC, F1 score) considered, leveraging social circles’ information outperforms the corresponding baseline in which circles are ignored. In addition, using only information about the innermost social circles guarantees the same performance achieved when using the whole network. Using social circles information also seems to provide the same performance as using additional classifiers on nodes, which might be impractical or costly to set up. Finally, and most importantly, circle-aware supervised link prediction outperformed recent state-of-the-art feature learning-link prediction approaches, including a GNN-based solution. Interestingly, the best performing circle is C1, which comprises only the two or three strongest relationships of the ego: using knowledge on a few common strong ties, circle-aware link prediction consistently beats black-box approaches.

As future work, we plan to investigate in detail the reasons why different social circles play a different role for link predictions, linking them to macroscopic or microscopic topological properties of the social graph. Another promising future direction is to study whether dynamic ego networks (i.e., time-varying ego network models) may prove useful for link prediction in dynamic social graphs.

REFERENCES

[1] The Global State of Digital in 2021 Report, We are social and Hootsuite, Global State of Digital, 2021.
[2] E. C. Mutlu, T. Oghaz, A. Rajabi, and I. Garibay, “Review on learning and extracting graph features for link prediction,” Mach. Learn. Knowl. Extraction, vol. 2, no. 4, pp. 672–704, Dec. 2020.
[3] V. Martínez, F. Berzal, and J.-C. Cubero, “A survey of link prediction in complex networks,” ACM Comput. Surveys, vol. 49, no. 4, pp. 1–33, Dec. 2017.
[4] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2016, arXiv:1609.02907.
[5] M. Zhang and Y. Chen, “Link prediction based on graph neural networks,” in Proc. NeurIPS, 2018, pp. 5165–5175.
[6] L. Cai, J. Li, J. Wang, and S. Ji, “Line graph neural networks for link prediction,” 2020, arXiv:2010.10046.
[7] J. Chen et al., “E-LSTM-D: A deep learning framework for dynamic network link prediction,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 51, no. 6, pp. 3699–3712, Jun. 2021.
[8] J. Zhang, J. Chen, and Q. Xuan, “Link prediction based on hyper-substructure network,” in Graph Data Mining, Singapore: Springer, 2021, pp. 29–48.
[9] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Discovery Data Mining, 2016, pp. 855–864.
[10] S. De Winter, T. Decuyper, S. Mitrovic, B. Baesens, and J. De Weerdt, “Combining temporal aspects of dynamic networks with Node2 Vec for a more efficient dynamic link prediction,” in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM), Aug. 2018, pp. 1234–1241.
[11] M. McPherson, L. Smith-Lovin, and J. M. Cook, “Birds of a feather: Homophily in social networks,” Ann. Rev. Sociol., vol. 27, no. 1, pp. 415–444, 2001.
[12] R. A. Hill and R. I. M. Dunbar, “Social network size in humans,” Hum. Nature, vol. 14, no. 1, pp. 53–72, Mar. 2003.
[13] W.-X. Zhou, D. Sornette, R. A. Hill, and R. I. M. Dunbar, “Discrete hierarchical organization of social group sizes,” Proc. Roy. Soc. B, Biol. Sci., vol. 272, no. 1561, pp. 439–444, 2005.
[14] R. I. M. Dunbar, “The social brain hypothesis,” Evol. Anthropol.: Issues, News, Rev., vol. 6, no. 5, pp. 178–190, 1998.
[15] A. G. Sutcliffe, D. Wang, and R. I. M. Dunbar, “Modelling the role of trust in social relationships,” ACM Trans. Internet Technol., vol. 15, no. 4, pp. 1–24, Dec. 2015.
[16] V. Arnaboldi, M. Conti, M. La Gala, A. Passarella, and F. Pezzeni, “Information diffusion in OSNs: The impact of nodes’ sociality,” in Proc. 29th Annu. ACM Symp. Appl. Comput., 2014, pp. 616–621.
[17] S. Aral and M. Van Alstyne, “The diversity-bandwidth trade-off,” Amer. J. Sociol., vol. 117, no. 1, pp. 90–171, Jul. 2011.
[18] P. Paraskevopoulos, C. Baldrini, A. Passarella, and M. Conti, “The academic wanderer: Structure of collaboration network and relation with research performance,” Appl. Netw. Sci., vol. 6, no. 1, p. 31, Dec. 2021.
[19] M. Everett and S. P. Borgatti, “Ego network betweenness,” Social Netw., vol. 27, no. 1, pp. 31–38, 2005.
[20] N. Lin, K. S. Cook, and R. S. Burt, Social Capital: Theory and Research. Piscataway, NJ, USA: Transaction Publishers, 2001.
[21] C. McCarty, “Structure in personal networks,” J. Social Struct., vol. 3, no. 1, p. 20, 2002.
[22] B. Gonçalves, N. Perra, and A. Vespignani, “Modeling users’ activity on Twitter networks: Validation of Dunbar’s number,” PLoS ONE, vol. 6, no. 8, Aug. 2011, Art. no. e22656.
[23] R. I. Dunbar, V. Arnaboldi, M. Conti, and A. Passarella, “The structure of online social networks mirrors those in the offline world,” Social Netw., vol. 43, pp. 39–47, Oct. 2015.
[24] L. M. Aiello and N. Barbieri, “Evo-lution of ego-networks in social media networks,” in Proc. 10th ACM Int. Conf. Web Search Data Mining, Feb. 2017, pp. 111–120.
[25] M. Toprak, C. Boldrini, A. Passarella, and M. Conti, “Harnessing the power of ego network layers for link prediction in online social networks,” 2021, arXiv:2109.09190.
[26] J. O. Haerter, B. Jamtveit, and J. Mathiesen, “Communication dynamics in finite capacity social networks,” Phys. Rev. Lett., vol. 109, no. 16, Oct. 2012, Art. no. 168701.
[27] G. Miriello et al., “Time as a limited resource: Communication strategy in mobile phone networks,” Social Netw., vol. 35, no. 1, pp. 89–95, Jan. 2013.
[28] L. Lü and T. Zhou, “Link prediction in complex networks: A survey,” Phys. A, Statist. Mech. Appl., vol. 390, no. 6, pp. 1150–1170, 2011.
[29] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” J. Amer. Soc. Inf. Sci. Technol., vol. 58, no. 7, pp. 1019–1031, 2007.
[30] L. A. Adamic and E. Adar, “Friends and neighbors on the web,” Soc. Netw., vol. 25, no. 3, pp. 211–230, 2003.
[31] T. Zhou, L. Lü, and Y.-C. Zhang, “Predicting missing links via local information,” Eur. Phys. J. B, vol. 71, pp. 625–630, Oct. 2009.
[32] D. Liben-Nowell, “An algorithmic approach to social networks,” Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2005.
[33] W. Liu and L. Lü, “Link prediction based on local random walk,” EPL (Europhys. Lett.), vol. 89, no. 5, p. 58007, 2010.
[34] S. Soundarajan and J. Hopcroft, “Using community information to improve the precision of link prediction methods,” in Proc. 21st Int. Conf. Companion World Wide Web (WWW Companion), 2012, pp. 607–608.
[35] M. A. Hasan, V. Chaoji, S. Salem, M. Zaki, and N. York, “Link prediction using supervised learning,” in Proc. SDM, 2006, pp. 1–10.
[36] L. Chen and Y. Chen, “Weisfeiler-Lehman neural machine for link prediction,” in Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Discovery Data Mining, Aug. 2017, pp. 575–583.
[37] W. Cuikerski, B. Hamner, and B. Yang, “Graph-based features for supervised link prediction,” in Proc. Int. Joint Conf. Neural Netw., Jul. 2011, pp. 1237–1244.
[38] Y. Wang and J. Li, “Credible intervals for precision and recall based on a R-k fold cross-validated bet distribution,” Neural Comput., vol. 28, no. 8, pp. 1694–1722, Aug. 2016.
[39] B. Zhu and Y. Xia, “Link prediction in weighted networks: A weighted mutual information model,” PLoS ONE, vol. 11, no. 2, Feb. 2016, Art. no. e0148265.
[40] L. Lü and T. Zhou, “Link prediction in weighted networks: The role of network structure,” EPL (Europhys. Lett.), vol. 89, no. 1, p. 18001, Jan. 2010.
[41] H. Kou et al., “Building trust/distrust relationships on signed social service network through privacy-aware link prediction process,” Appl. Soft Comput., vol. 100, Mar. 2021, Art. no. 106942.
[42] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in Proc. ACM SIGKDD, 2014, pp. 701–710.

[43] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 1, pp. 4–24, Jan. 2021.

[44] W. L. Hamilton, “Synthesis lectures on artificial intelligence and machine learning,” in Graph Representation Learning. San Rafael, CA, USA: Morgan & Claypool, 2020.

[45] V. Arnaboldi, M. Conti, A. Passarella, and F. Pezzoni, “Ego networks in Twitter: An experimental analysis,” in Proc. IEEE INFOCOM, Apr. 2013, pp. 3459–3464.

[46] C. Boldrini, M. Toprak, M. Conti, and A. Passarella, “Twitter and the press: An ego-centred analysis,” in Proc. Companion Web Conf., 2018, pp. 1471–1478.

[47] L. A. Goodman, “Snowball sampling,” Ann. Math. Statist., vol. 32, no. 1, pp. 148–170, Mar. 1961.

[48] P. Zhang, X. Wang, F. Wang, A. Zeng, and J. Xiao, “Measuring the robustness of link prediction algorithms under noisy environment,” Sci. Rep., vol. 6, no. 1, May 2016.

[49] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, “New perspectives and methods in link prediction,” in Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2010, pp. 243–252.

[50] J. O’Madadhain, J. Hutchins, and P. Smyth, “Prediction and ranking algorithms for event-based network data,” ACM SIGKDD Explor. Newslett., vol. 7, no. 2, pp. 23–30, 2005.

[51] H. Cai, V. W. Zheng, and K. C.-C. Chang, “A comprehensive survey of graph embedding: Problems, techniques, and applications,” IEEE Trans. Knowl. Data Eng., vol. 32, no. 1, pp. 1616–1637, Sep. 2020.

[52] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, “Link prediction techniques, applications, and performance: A survey,” Phys. A. Stat. Mech. Appl., vol. 553, Sep. 2020, Art. no. 124289.

[53] G. Forman and M. Scholz, “Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement,” ACM SIGKDD Explor. Newslett., vol. 12, no. 1, pp. 49–57, 2010.

[54] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, “Human mobility, social ties, and link prediction,” in Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2011, pp. 1100–1108.

[55] Y. Yang, R. N. Lichtenwalter, and N. V. Chawla, “Evaluating link prediction methods,” Knowl. Inf. Syst., vol. 45, no. 3, pp. 751–782, Dec. 2015.

[56] C. Goutte and E. Gaussier, “A probabilistic interpretation of precision, recall and F-score, with implication for evaluation,” in Proc. ECIR. Berlin, Germany: Springer, 2005, pp. 345–359.

Chiara Boldrini is a Researcher with IIT-CNR, Pisa, Italy. She has published more than 60 articles on these topics, which have received more than 2800 citations. Her research interests are in social computing and computational social science, human-centered artificial intelligence (AI), smart transportation and urban data science, and mobile and ubiquitous systems.

M. Boldrini is the IIT co-PI for the projects CHIST-ERA SAI, H2020 SoBigData++, and H2020 HumanE-AI-Net. She has been involved in several national and European projects, among which are HAGGLE, SOCIALNETS, and ESPRIT. She is in the Editorial Board of Pervasive and Mobile Computing (Elsevier) and Computer Communications (Elsevier). She is the TPC Co-Chair of IEEE SMARTCOMP 2022 and served as the TPC Vice-Chair for IEEE PerCom 2021. She has been in the organizing committee of several IEEE and ACM conferences and workshops, among which are IEEE PerCom, and IEEE SECON ACM MobiHoc.

Andrea Passarella received the Ph.D. degree from the University of Pisa, Pisa, Italy, in 2005.

He is a Research Director with the Institute for Informatics and Telematics (IIT), National Research Council of Italy (CNR), Pisa. Prior to joining IIT, he was with the Computer Laboratory, University of Cambridge, Cambridge, U.K. He has published more than 170 articles on online and mobile social networks, decentralized artificial intelligence (AI), next-generation Internet, opportunistic, ad hoc, and sensor networks. He is coauthor of the book Online Social Networks: Human Cognitive Constraints in Facebook and Twitter Personal Graphs (Elsevier, 2015).

Dr. Passarella was a recipient of the Best Paper Award at IFIP Networking 2011 and IEEE WoWMoM 2013. He currently serves as the General Chair for IEEE PerCom 2022. He is the founding Associate Editor-in-Chief of Elsevier Online Social Networks. He was a guest co-editor of several special sections in ACM and Elsevier journals. He is the PI of the EU CHIST-ERA SAI (Social Explainable AI) project.

Marco Conti is the CNR Research Director and, currently, he is the Director of IIT-CNR, Pisa, Italy. He has published more than 400 scientific articles related to design, modeling, and experimentation of computer and communication networks, pervasive systems, and online social networks.

Mr. Conti is the founding EiC of Online Social Networks and Media, EiC for Special Issues of Pervasive and Mobile Computing, and, from 2009 to 2018, EiC of Computer Communications. He has received several awards, including the Best Paper Award at the IFIP Networking 2011, the IEEE ISCC 2012, and the IEEE WoWMoM 2013. He was included in the “2017 Highly Cited Researchers” list compiled by Web of Science for the most cited articles in computer science. He served as the General/Program Chair of several major conferences, including IFIP Networking 2002, IEEE WoWMoM 2005 and 2006, IEEE PerCom 2006 and 2010, ACM MobiHoc 2006, IEEE MASS 2007, and IEEE SmartComp 2021.