# A Holistic Approach for
# Link Prediction in Multiplex Networks

Alireza Hajibagheri[1], Gita Sukthankar[1], and Kiran Lakkaraju[2]

[1]University of Central Florida, Orlando, Florida
[2]Sandia National Labs, Albuquerque, New Mexico
alireza@eecs.ucf.edu, gitars@eecs.ucf.edu, klakkara@sandia.gov

**Abstract.** Networks extracted from social media platforms frequently include multiple types of links that dynamically change over time; these links can be used to represent dyadic interactions such as economic transactions, communications, and shared activities. Organizing this data into a dynamic multiplex network, where each layer is composed of a single edge type linking the same underlying vertices, can reveal interesting cross-layer interaction patterns. In coevolving networks, links in one layer result in an increased probability of other types of links forming between the same node pair. Hence we believe that a holistic approach in which all the layers are simultaneously considered can outperform a factored approach in which link prediction is performed separately in each layer. This paper introduces a comprehensive framework, MLP (Multiplex Link Prediction), in which link existence likelihoods for the target layer are learned from the other network layers. These likelihoods are used to reweight the output of a single layer link prediction method that uses rank aggregation to combine a set of topological metrics. Our experiments show that our reweighting procedure outperforms other methods for fusing information across network layers.

## 1 Introduction

As social media platforms offer customers more interaction options, such as *friending*, *following*, and *recommending*, analyzing the rich tapestry of interdependent user interactions becomes increasingly complicated. In this paper, we study two types of online societies: 1) players in a massively multiplayer online game (Travian) [1] 2) dialogs between Twitter users before, during, and after an exceptional event [2]. Although standard social network analysis techniques [3] offer useful insights about these communities, there is relatively little theory from the social sciences on how to integrate information from multiple types of online interactions.

Rather than organizing this data into social networks separately chronicling the history of different forms of user interaction, dynamic multiplex networks [4] offer a richer formalism for modeling the social fabric of online societies. A multiplex network is a multilayer network that shares the same set of vertices across all layers. This network can be modeled as a graph $G = < V, E >$ where $V$ is the set of vertices and $E$ is the set of edges present in the graph. The dynamic graph $G = \{G_0, G_1, ..., G_t\}$ represents the state of the network at different times. The network is then defined as:

$G_t = \langle V, E_t^1, ..., E_t^M \rangle$ with $E_t^\alpha \subseteq V \times V, \forall \alpha \in \{1, ..., M\}$, where each set $E_t^\alpha$ corresponds to the edge set of a distinct layer at time $t$. Thus a dynamic multiplex network is well suited for representing diverse user activities over a period of time.

In this paper, we address the problem of predicting future user interactions from the history of past connections. Assuming the data is represented as a graph, our goal is to predict the structure of graph $G_t$ using information from previous snapshots as well as other layers of the network. Link prediction algorithms [5–9] have been implemented for many types of online social networks, including massively multiplayer online games and location-based social networks. These systems offer great value to social networking services due to their practical applicability for friend recommendations and social network bootstrapping. Although user profiles can be mined for additional data, topological approaches 1) perform well in many networks 2) preserve user privacy since they do not rely on actor information and 3) can be combined with node content approaches to enhance prediction performance.

Despite the fact that link prediction is a well studied problem, few link prediction techniques specifically address the problem of simultaneously predicting links across multiple networks [10–13]. Basu et al. [14] note that there are many real-world cases where interdependencies between processes cause the layers of a multiplex network to coevolve, resulting in a higher number of overlapping edges between the same node pair in different network layers. In this paper, we explore the role of overlapping edges towards improving the performance of link prediction; our aim is to leverage the cross-layer link co-occurrence history to model coevolution in a multiplex network. Our contributions can be summarized as follows:

- We introduce a framework for multiplex link prediction (MLP) that integrates complementary information sources, including topological metrics, network dynamics, and overlapping edges.
- MLP uses a likelihood based method for learning cross-layer dependencies and a temporal decay function to model the network dynamics. Rank aggregation is then employed to collect information from multiple topological metrics into one scoring matrix for ranking potential links.
- Extensive experiments are conducted on datasets collected from different types of social networks, and the proposed model is shown to outperform state-of-the-art link prediction methods.

In the next section, we present related work on link prediction. The proposed framework is described in Section 4. Section 5 presents a comparison of our method vs. two other approaches for fusing information across network layers. We conclude in Section 7 with a description of possible directions for future work.

## 2   Related Work

A variety of computational approaches have been employed for predicting links in single layer networks, including supervised classifiers, statistical relational learning, matrix factorization, metric learning, and probabilistic graphical models (see surveys by [15–17] for a more comprehensive description). Regardless of the computational framework, topological network measures are commonly used as features to describe

node pairs and can be combined in a supervised or unsupervised fashion to do link prediction [6]. In this paper, we aggregate several of these metrics (listed in the next section), but our framework can be easily generalized to include other types of features.

The primary focus of this paper is leveraging cross-layer information to improve link prediction in multiplex networks, although we also introduce our own single layer link prediction technique. This process of using cross-layer information can be treated as a transfer learning problem where information is learned from a source network and applied to improve prediction performance the target network. Tang et al. [10] introduced a transfer-based factor graph (TranFG) model which incorporates social theories into a semi supervised learning framework. This model is then used to transfer supervised information from a source network to infer social ties in the target network.

Another strategy is to create more general versions of the topological measures that capture activity patterns in multilayer networks. Davis et al. [11] introduced a probabilistically weighted extension of the Adamic/Adar measure for these networks. Weights are calculated by doing a triad census to estimate the probability of different link type combinations. The extended Adamic/Adar metric is then used, along with other unsupervised link predictors, as input for a supervised classifier. Similarly, Hristova et al. [12] extend the definition of network neighborhood by considering the union of neighbors across all layers. These multilayer features are then combined in a supervised model to do link prediction. One weakness with the above mentioned models is their inability to use temporal information accrued over many snapshots, rather than relying on a single previous snapshot. In this paper, we evaluate two versions of our MLP framework, a version that only uses topological metrics calculated from one time slice vs. multiple snapshots. Rossetti et al. [13] combined multidimensional versions of Common Neighbors and Adamic/Adar with predictors that are able to utilize temporal information. However, like the standard version of these metrics, these extended versions do not necessarily generalize to networks generated from different processes.

Conversely, there are a number of approaches that ignore cross-layer network dependencies, while using the history of changes between snapshots to predict future network dynamics. We have experimented with two types of techniques: time series forecasting [5, 18] and decay models [19]. Soares and Prudêncio [18] investigated the use of time series within both supervised and unsupervised link prediction frameworks. The core concept of their approach is that it is possible to predict the future values of topological metrics with time series; these values can either be used in an unsupervised fashion or combined in a supervised way with a classifier. In previous work, we introduced a rate prediction model [5] that uses time series to predict the rate of link formation. Our proposed framework, MLP, both models the rate of link formation in each layer and uses a decay model to account for changes in the topological metrics over time. In our results, we compare the improvements achieved by temporal vs. cross-layer modeling.

However, incorporating more features is not helpful, without an effective information fusion procedure. Pujari et al. [20] employed computational social choice algorithms for aggregating multiple topological features. They evaluated the performance of two well-known rank aggregation methods, Borda and Kemeny, for single layer link prediction. In their method, weights are learned for each voter participating in the rank

aggregation, where each topological metric is treated as a voter. These weights are tuned to maximize the identification of positive examples or minimize negative examples. To extend their method to multiplex networks [21], the authors compute topological attributes for each network layer and combine them using 1) a simple aggregation of these scores across all layers or 2) an entropy-aggregation of values. These combinations are then used as a series of features in a decision tree model. In this paper, we use rank aggregation to fuse our features and compare our procedure to their aggregation methods. Another example of a supervised framework that uses rank aggregation is *RankMerging* [22]. During a learning phase, weights are assigned to each unsupervised method using a training set of node pairs. The contribution of each ranking to the merged ranking is then computed using sliding indices. At each step, the aim is to identify the ranking with the highest number of true predictions in the upcoming steps. Rank aggregation methods can be highly effective, but the more complex social choice algorithms can suffer from high computational complexity, making them less effective for large datasets. For this reason, we opted to use the Borda rank aggregation procedure in MLP.

## 3   Node Similarity Metrics

This section provides a brief description of the topological and path-based metrics for encoding node similarity that are used within our MLP framework to create ranked score lists for each node pair. These techniques are often used in isolation as unsupervised methods for link prediction. Note that $\Gamma(x)$ stands for the set of neighbors of vertex $x$ while $w(x, y)$ represents the weight assigned to the interaction between node $x$ and $y$. More details about these metrics could be found in [23].

- **Number of Common Neighbors (CN)**
  The CN measure is defined as the number of nodes with direct relationships with both evaluated nodes $x$ and $y$ [24].
- **Jaccard's Coefficient (JC)**
  The JC measure assumes higher values for pairs of nodes who share a higher proportion of common neighbors relative to their total neighbors.
- **Preferential Attachment (PA)**
  The PA measure assumes that the probability that a new link originates from node $x$ is proportional to its node degree. Consequently, nodes that already possess a high number of relationships tend to create more links  [25].
- **Adamic-Adar Coefficient (AA)**
  This metric [26] is closely related to Jaccard's coefficient in that it assigns a greater importance to common neighbors who have fewer neighbors. Hence, it measures the exclusivity of the relationship between a common neighbor and the evaluated pair of nodes.
- **Resource Allocation (RA)**
  RA was first proposed in [27] and is based on physical processes of resource allocation.

- **Page Rank (PR)**
  The PageRank algorithm [28] measures the significance of a node based on the significance of its neighbors. We use the weighted PageRank algorithm proposed in [29].
- **Inverse Path Distance (IPD)**
  The Path Distance measure for unweighted networks simply counts the number of nodes along the shortest path between $x$ and $y$ in the graph. Note that $PD(x, y) = 1$ if two nodes $x$ and $y$ share at least one common neighbor. In this article, the Inverse Path Distance is used to measure the proximity between two nodes, where:

$$IPD(x, y) = \frac{1}{PD(x, y)} \qquad (1)$$

- **Product of Clustering Coefficient (PCF)**
  The clustering coefficient of a vertex $v$ is defined as:

$$PCF(v) = \frac{3 \times \text{\# of triangles adjacent to v}}{\text{\# of possible triples adjacent to v}} \qquad (2)$$

  To compute a score for link prediction between the vertex $x$ and $y$, one can multiply the clustering coefficient score of $x$ and $y$.

Section 5 compares MLP vs. unsupervised versions of these approaches.

## 4   Proposed Method

MLP is a hybrid architecture that utilizes multiple components to address different aspects of the link prediction task. We seek to extract information from all layers of the network for the purpose of link prediction within a specific layer known as the target layer. To do so, we create a weighted version of the original target layer where interactions and connections that exist in other layers receive higher weights. After reweighting the layer, we employ the collection of node similarity metrics described in the previous section on the weighted network. To express the temporal dynamics of the network, we use a decay model on the time series of similarity metrics to predict future values. Finally, the Borda rank aggregation method is employed to combine the ranked lists of node pairs into a single list that predicts links for the next snapshot of the target network layer. Each component of the model is explained in more detail in the following sections.

### 4.1   Multiplex Likelihood Assignment and Edge Weighting

This component leverages information about cross-layer link co-occurrences. During the coevolution process, links may be engendered due to activity in other network layers. Some layers may evolve largely independently of the rest of the network, whereas links in other layers may be highly predictive of links in the target layer. In our proposed method, a weight is assigned to each layer based on its influence on the target layer. Weights are calculated using a likelihood function: where $L^i$ and $w_i$ represent the

$i$th layer and the weight calculated for it respectively. $L^{Target}$ indicates the target layer for which we want to predict future links. The *Likelihood* function computes the similarity between the target layer and the $i$th layer; to do this, we use the current ratio of overlapping edges. Next, we calculate weights for every node pair by checking the link correspondence between two layers using the likelihood of a link being present in the target layer given the existence of the link in the other layer at any other previous snapshot. This orders other layers in terms of their relative importance for a specific target layer. The process assigns higher weights to node pairs which occur in more than one layer (multiplex edges). The rate of link formation is incorporated into the model as the first term of the edge weight. Algorithm 1 shows the process of assigning likelihoods to layers and reweighting the adjacency matrix.

---

**Algorithm 1** Likelihood Assignment and Edge Weighting

---

1: Input: Edge sets $(E^1, ..., E^M)$ for $M$ layers where $E^\alpha$ is the edge set of target layer
2: Output: $E_w^\alpha$ weighted adjacency matrix for layer $\alpha$ (target layer)
   //Calculate weights for the layers
3: **for** $i \in \{1, 2, ..., M\} - \{\alpha\}$ **do**
4:     $w_i = Likelihood(\text{Link in } L^\alpha | \text{Link in } L^i)$
5: **end for**
   //Weighting target layer
6: **for** edge $e \in E^\alpha$ **do**
7:     $w_e = rate + \sum_{i=1 \& i \neq \alpha}^{M} w_i \times linkExist(e)$
8: **end for**

---

The term *rate* is defined as the average value of the source node's out-degree over previous timesteps. Function *linkExist* is used to obtain information about a link's existence in other layers during previous snapshots. It checks each layer for the presence of an edge and returns 1 if an edge is present in that layer.

### 4.2 Temporal Link Structure

Given the network history for $T$ time periods, we need to capture the temporal dependencies of the coevolution process. To do so, our framework uses a weighted exponentially decaying model [30]. Let $\{Sim_t(i,j), t = t_0 + 1, ..., t_0 + T\}$ be a time series of similarity score matrices generated by a node similarity metric on a sliding window of $T$ successive temporal slices. An aggregated weighted similarity matrix is constructed as follows:

$$Sim_{(t_0+1) \sim (t_0+T)}(i,j) = \sum_{t=t_0+1}^{t_0+T} \theta^{t_0+T-t} Sim_t(i,j) \qquad (3)$$

where the parameter $\theta \in [0, 1]$ is the smoothing weight for previous time periods. Different values of $\theta$ modify the importance assigned to the most or least recent snapshots before current time $t + 1$. This procedure generates a composite temporal score matrix
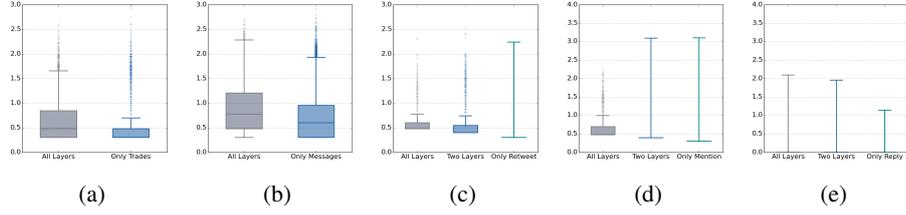
Fig. 1: Log scale box-whisker plots for user interactions in different layers of the network: (a) Travian (Trades) (b) Travian (Messages) (c) Cannes2013 (Retweets) (d) Cannes2013 (Mentions) (e) Cannes2013 (Replies)
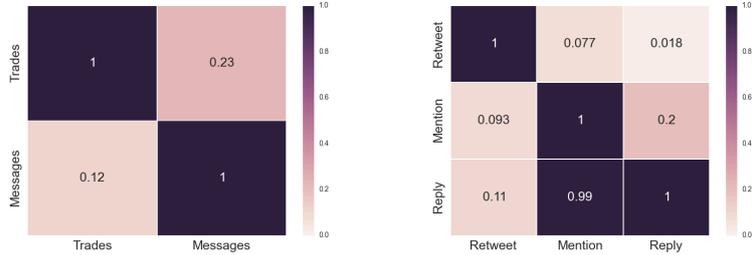


Fig. 2: Heatmap representing the edge overlap between pairs of layers for datasets (a) Travian (b) Cannes2013

for every node similarity metric. $Sim_{(t_0+1)\sim(t_0+T)}$ (shortened to $Sim$) is used by the algorithm as a summary of network activity, encapsulating the temporal evolution of the similarity matrix.

### 4.3 Rank Aggregation

Before describing the final step of our approach, let us briefly discuss existing methods for *ranked list aggregation/rank aggregation*. List merging or list aggregation refers to the process of combining a number of lists with the same or different numbers of elements in order to get one final list including all the elements. In rank aggregation, the order or rank of elements in input lists is also taken into consideration. The input lists can be categorized as *full, partial, or disjoint lists*. Full lists contain exactly the same elements but with a different ordering, partial lists may have some of the elements in common but not all, and disjoint lists have completely different elements. In this case, we are only dealing with full lists since each similarity metric produces a complete list for the same set of pairs, differing only in ordering.

Several rank aggregation methods are described in [31], including Borda's, Markov chain, and median rank methods. Borda's method is a *rank-then-combine* method originally proposed to obtain a consensus from a voting system. Since it is based on the

absolute positioning of the rank elements and not their relative rankings, it can be considered a truly positional method. For every element in the lists, a Borda score is calculated and elements are ranked according to this score in the aggregated list. For a set of complete ranked lists $L = [L_1, L_2, L_3, ...., L_k]$, the Borda score for an element $i$ and a list $L_k$ is given by:

$$B_{L_k}(i) = \{count(j) | L_k(j) < L_k(i) \& j \in L_k\} \tag{4}$$

The total Borda score for an element is given as:

$$B(i) = \sum_{t=1}^{k} B_{L_t(i)} \tag{5}$$

Borda's method is computationally cheap, which is a highly desirable property for link prediction in large networks.

Algorithm 2 shows our proposed framework which incorporates edge weighting, the temporal decay model, and rank aggregation to produce an accurate prediction of future links in a dynamic multiplex network. The Borda function produces the final output of the MLP framework. Results of the proposed algorithm are compared with other state-of-the-art techniques in the next section.

---

**Algorithm 2** Multiplex Link Prediction Framework (MLP)

---

1: Input: Weighted edge sets of the target layer for $T$ previous snapshots
2: Output: Temporal aggregated score matrix $S$ for the target layer
3: **for** each node similarity metric $u$ **do**
4:      **for** $t \in \{1, ..., T\}$ **do**
5:          Calculate score matrix $Sim_{t0+t}^u$
6:      **end for**
7:      Calculate temporal similarity matrix $Sim^u$
8: **end for**
9: Final score matrix $S = Borda(Sim^1, ..., Sim^u)$

---

## 5   Experimental Study

This paper evaluates the MLP framework on networks extracted from two real-world datasets, Travian and Cannes2013. To investigate the impact of each component of our proposed method, not only do we compare our results with two other approaches for fusing cross-layer information, but we also analyze the performance of ablated versions of our method. The complete method, MLP (Hybrid), is compared with MLP (Decay Model + Rank Aggregation) and MLP (Weighted + Rank Aggregation). All of the algorithms were implemented in Python and executed on a machine with the Intel(R) Core i7 CPU and 24GB of RAM for the purpose of fair comparison. Our implementation uses Apache Spark to speed the link prediction process.

Table 1: Dataset Summary: Number of edges, nodes, and snapshots for each network layer

| Dataset | | Travian | | Cannes2013 |
|---|---|---|---|---|
| **No. of Nodes** | | 2,809 | | 438,537 |
| **No. of Snapshots** | | 30 | | 29 |
| **Layers/No. of Edges** | Trades | 87,418 | Retweet | 496,982 |
| | Messages | 44,956 | Mention | 411,338 |
| | | | Reply | 83,534 |

## 5.1 Datasets

We use two real-world dynamic multiplex networks to demonstrate the performance of our proposed algorithm. These networks are considerably disparate in structure and were selected from different domains (a massively multiplayer online game (MMOG) and an event-based Twitter dataset). Table 1 provides the network statistics for each of the datasets:

– **Travian MMOG** [1] Travian is a browser-based, real-time strategy game in which the players compete to create the first civilization capable of constructing a Wonder of the World. The experiments in this paper were conducted on a 30 day period in the middle of the Travian game cycle. In Travian, players can execute different game actions including: sending messages, trading resources, joining alliances, and attacking enemy villages. In this research, we focus on networks created from trades and messages.

– **Twitter Interactions** [2] This dataset consists of Twitter activity before, during, and after an "exceptional" event as characterized by the volume of communications. Unlike most Twitter datasets which are built from follower-followee relationships, links in this multiplex network correspond to retweeting, mentioning, and replying to other users. The Cannes2013 dataset was created from tweets about the Cannes film festival that occurred between May 6,2013 to June 3, 2013. Each day is treated as a separate network snapshot.

## 5.2 Evaluation Metrics

For the evaluation, we measure receiver operating characteristic (ROC) curves for the different approaches. The ROC curve is a plot of the *true positive rate (tpr)* against the *false positive rate (fpr)*. These curves show achievable true positive rates (TP) with respect to all false positive rates (FP) by varying the decision threshold on probability estimations or scores. For all of our experiments, we report area under the ROC curve (AUROC), the scalar measure of the performance over all thresholds. Since link prediction is highly imbalanced, straightforward accuracy measures are well known to be misleading; for example, in a sparse network, the trivial classifier that labels all samples as missing links can have a 99.99% accuracy.

### 5.3   Analysis of Cross-layer Interaction

Figure 1 shows log scale box-whisker plots that depict the frequency of interactions between users who are connected across multiple layers. We compare the frequency of interactions in cases where the node pair is connected on all layers vs. the frequency of being connected in a single layer (Travian) or less than all layers (for Cannes which has three layers). As expected, in cases where users are connected on all layers, the number of interactions (trades, messages, retweets, mentions and replies) is higher. The heatmap of the number of overlapping edges between different network layers (Figure 2) suggests that a noticeable number of edges are shared between all layers. This clearly indicates the potential value of cross-layer information for the link prediction task on these datasets. Our proposed likelihood weighting method effectively captures the information revealed by our analysis.

### 5.4   Performance of Multiplex Link Prediction

For our experiments, we adopted a moving-window approach to evaluate the performance of our temporal multiplex link prediction algorithm. Given a specified window size $T$, for each time period $t(t > T)$, graphs of $T$ previous periods $(G_{tT}, ..., G_{t1})$ (where each graph consists of $M$ layers) are used to predict links that occur at the target layer $\alpha$ in the current period $(G_t^\alpha)$. To assess our proposed framework and study the impact of its components, we compare against the following baselines:

– **MLP (Hybrid)**: incorporates all elements discussed in the framework section. It utilizes the likelihood assignment and edge weighting procedure to extract cross-layer information. Node similarity scores are modified using the temporal decay model and combined with Borda rank aggregation.
– **MLP (Likelihood + Rank Aggregation)**: This method only uses the aggregated scores calculated from the graphs weighted with cross-layer information. It does not consider the temporal aspects of network coevolution.
– **MLP (Decay Model + Rank Aggregation)**: This method does not use the cross-layer weighting scheme and relies on temporal information alone to predict future links. The final aggregated score matrix is calculated based on forecast values at time $t$ for each node similarity metric using the decay model.
– **Likelihood**: Weights generated by the cross-layer likelihood assignment procedure are treated as scores for every node pair. We then sort the pairs based on their score and calculate the AUROC.
– **Rank Aggregation**: This method is a simple aggregated version of all unsupervised scoring methods using the Borda's rank aggregation method applied to node similarity metrics from the target layer.
– **Unsupervised Methods**: The performance of our proposed framework is compared with eight well-known unsupervised link prediction methods described in Section 3. All unsupervised methods are applied to the binary static graph from time 0 to $t - 1$ in order to predict links at time $t$. Only the structure of the target layer is used.

– **Average Aggregation**: In order to extend the rank aggregation model to include information from other layers of the network, we use the idea proposed in [21]. Node similarity metrics are aggregated across all layers. So for attribute $X$ (Common Neighbors, Adamic/Adar, etc.) over $M$ layers the following is defined:

$$X(u,v) = \frac{\sum_{\alpha=1}^{M} X(u,v)^{\alpha}}{M} \qquad (6)$$

where $X(u,v)$ is the average score for nodes $u$ and $v$ across all layers and $X(u,v)^{\alpha}$ is the score at layer $\alpha$. Borda's rank aggregation is then applied to the extended attributes to calculate the final scoring matrix.

– **Entropy Aggregation**: Entropy aggregation is another extended rank aggregation model proposed in [21] where $X(u,v)$ is defined as follows:

$$X(u,v) = -\sum_{\alpha=1}^{M} \frac{X(u,v)^{\alpha}}{X_{total}} \log(\frac{X(u,v)^{\alpha}}{X_{total}}) \qquad (7)$$

where $X_{total} = \sum_{\alpha=1}^{M} X(u,v)^{\alpha}$. The entropy based attributes are more suitable for capturing the distribution of the attribute value over all dimensions. A higher value indicates a uniform distribution of attribute values across the multiplex layers.

– **Multiplex Unsupervised Methods**: Finally, using the definition of core neighborhood proposed in [12], we extend four unsupervised methods (Common Neighbors, Preferential Attachment, Jaccard Coefficient and Adamic/Adar) to their multiplex versions.

Table 2 shows the results of different algorithms on the Travian and Cannes2013 datasets. With 30 days of data from Travian and 27 days for Cannes2013, we were able to extensively compare the performance of the proposed methods and the impact of using different elements. Bold numbers indicate the best results on each target layer considered; MLP (Hybrid) is the best performing algorithm in all cases.

## 6   Discussion

In this section, we discuss the most interesting findings:

**Does rank aggregation improve the performance of the unsupervised metrics?** As shown in Table 2, although the aggregated scores matrix produced by Borda's method achieves better results than unsupervised methods in some cases (Travian message, Cannes2013 retweet and mention networks) and comparable results on others (Travian trade and Cannes2013 reply networks), it is not able to significantly outperform all unsupervised methods in any of the networks. As discussed before, we are using the simple Borda method for the rank aggregation which does not consider the effect of each ranker on the final performance. While adding weights to the rankers or using more complex rank aggregation models such as Kemeny might achieve better results, it has been shown that those approaches have high computational complexity which makes them less suitable for large real-world networks [20, 22]. Despite the fact that the rank aggregation alone does not significantly improve the overall performance

Table 2: AUROC performances for a target layer averaged over all snapshots with a sliding time window of $T = 3$ for Travian layers and $T = 5$ for Cannes2013 layers used in the decay model. Variants of our proposed framework are shown at the top of the table, followed by standard unsupervised methods. The algorithms shown in the bottom half of the table are techniques for multiplex networks proposed by other research groups. The best performer is marked in bold.

| Algorithms / Networks | Trade | Message | Retweet | Mention | Reply |
|---|---|---|---|---|---|
| **MLP (Hybrid)** | **0.821±0.001** | **0.803±0.002** | **0.812±0.002** | **0.834±0.003** | **0.839±0.002** |
| **MLP (LH/RA)** | 0.802±0.001 | 0.790±0.0021 | 0.809±0.003 | 0.814±0.004 | 0.816±0.003 |
| **MLP (DM/RA)** | 0.722±0.002 | 0.731±0.002 | 0.727±0.002 | 0.728±0.003 | 0.733±0.002 |
| **Likelihood** | 0.770±0.033 | 0.760±0.041 | 0.752±0.022 | 0.781±0.052 | 0.757±0.042 |
| **Rank Aggregation** | 0.694±0.001 | 0.712±0.001 | 0.700±0.002 | 0.706±0.002 | 0.700±0.003 |
| **Common Neighbors** | 0.656±0.002 | 0.667±0.002 | 0.699±0.002 | 0.705±0.003 | 0.699±0.001 |
| **Jaccard Coefficient** | 0.628±0.002 | 0.680±0.003 | 0.594±0.002 | 0.733±0.002 | 0.711±0.003 |
| **Preferential Attachment** | 0.709±0.002 | 0.637±0.001 | 0.584±0.002 | 0.612±0.002 | 0.587±0.003 |
| **Adamic/Adar** | 0.635±0.003 | 0.700±0.003 | 0.700±0.002 | 0.642±0.002 | 0.516±0.003 |
| **Resource Allocation** | 0.625±0.005 | 0.690±0.003 | 0.597±0.002 | 0.622±0.002 | 0.672±0.003 |
| **Page Rank** | 0.595±0.0016 | 0.687±0.002 | 0.660±0.002 | 0.630±0.003 | 0.613±0.002 |
| **Inverse Path Distance** | 0.572±0.003 | 0.650±0.003 | 0.631±0.003 | 0.641±0.002 | 0.561±0.004 |
| **Clustering Coefficient** | 0.580±0.002 | 0.633±0.003 | 0.570±0.020 | 0.621±0.011 | 0.522±0.004 |
| **Average Aggregation** | 0.744±0.030 | 0.752±0.020 | 0.740±0.003 | 0.737±0.011 | 0.761±0.003 |
| **Entropy Aggregation** | 0.731±0.004 | 0.763±0.020 | 0.75±0.0030 | 0.758±0.031 | 0.744±0.002 |
| **Multiplex CN** | 0.729±0.0040 | 0.643±0.013 | 0.672±0.003 | 0.716±0.003 | 0.733±0.002 |
| **Multiplex JC** | 0.666±0.031 | 0.619±0.012 | 0.580±0.003 | 0.736±0.002 | 0.722±0.002 |
| **Multiplex PA** | 0.722±0.010 | 0.646±0.012 | 0.580±0.003 | 0.640±0.003 | 0.621±0.003 |
| **Multiplex AA** | 0.671±0.010 | 0.690±0.031 | 0.671±0.003 | 0.669±0.003 | 0.552±0.003 |

of the link prediction task, it enables us to effectively fuse different kinds of information (edge and node features, nodes similarity, etc.).

On the other hand, the Average and Entropy Aggregation methods, which are designed to consider attribute values from other layers, are able to outperform regular Rank Aggregation and MLP (Decay Model + Rank Aggregation). However, both methods use the static structure of all snapshots from time 0 to $t - 1$, while MLP (Decay Model + Rank Aggregation) only incorporates the past $T$ snapshots which makes it more suitable for large networks.

**Does the likelihood assignment procedure outperform the unsupervised scores?** To study the ability of our likelihood weighting method to model the link formation

process, we generate results for two methods: using likelihood explicitly as a scoring method as well as using the values to generate a weighted version of the networks. First, the *Likelihood* method is used in isolation to demonstrate the prediction power of its weights as a new scoring approach. Table 2 shows significant improvements on unsupervised scores as well as the aggregated version of them. As expected, the more overlap between the target layer and predictor layers, the more performance improvement *Likelihood* achieves. As an example, Likelihood achieves $\sim 7\%$ of improvement on Travian (Trade) compared with $\sim 5\%$ of improvement on Travian (Message). Not only is there a lower rate of overlapping edges between those layers, but also the number of interactions is higher than the two other layers. The same holds true for Cannes2013 (Retweet) compared with the mention and reply layers.

On the other hand, the method introduced in Algorithm 1 generates a weighted version of input graphs which is used to generate a weighted version of unsupervised methods to produce the final scoring matrix. This paired with the rank aggregation method generates significantly better average AUROC performance compared with other proposed methods. Also, when temporal information from previous snapshots of the network is included, MLP (Hybrid) outperforms other variants of MLP as well as well-known unsupervised methods. This indicates the power of overlapping links in improving the performance of link prediction in coevolving multiplex networks.

**Does including temporal information improve AUROC performance?** The importance of incorporating temporal information into link prediction has been discussed in our previous work [5]. However, here we are interested in analyzing the impact of this information on improving the performance of MLP. For that purpose, first, the decay model is employed in MLP (Decay Model + Rank Aggregation) to determine whether it improves the results generated by the aggregated score matrix. The final aggregated score matrix is calculated based on forecast values at time $t$ for each unsupervised method using the decay model. As expected, this version of MLP is able to achieve up to $\sim 3\%$ of AUROC improvement using only information from the last three and five snapshots of the Travian and Cannes2013 networks respectively. On the other hand, we observed the same pattern when the decay model was added to MLP (Hybrid) along with likelihood and rank aggregation. Using the scores generated by our hybrid approach outperformed all other proposed and existing methods. The results presented here have been obtained using $T = 3$ for the Travian dataset and $T = 5$ for Cannes2013. These values are based on experiments performed on both datasets. While for Travian layers, increasing the value of $T$ tends to improve the prediction performance slightly until $T = 3$; higher values of $T$ may decrease the performance. The same pattern occurs for Cannes2013 layers when $T = 5$. Similarly, the value of $\theta$ is set to $0.4$ for both datasets.

In summary, MLP (Decay Model + Rank Aggregation) is able to achieve results comparable to other baseline methods except Average and Entropy Aggregation since they benefit from the entire graph structure. Although rank aggregation by itself is not able to significantly improve the performance of unsupervised methods, paired with decay models and taking temporal aspects of the network, it can achieve better performance. On the other hand, the multiplex versions of the neighborhood based unsupervised methods are able to improve average AUROC performance, however the

results are inconsistent and they achieve lower performance in many cases. Finally, both MLP (Hybrid) and MLP (Likelihood + Rank Aggregation) achieve higher performance compared with all other methods, illustrating the importance of the cross-layer information created by the network coevolution process. A paired two-sample $t$-test is used to indicate the significance of the results produced by each method where the $p$-value is smaller than 0.0001. It is worth mentioning that, even though MLP (Hybrid) is able to outperform all other methods, its performance is not significantly better than MLP (Likelihood + Rank Aggregation) in the case of Travian (Message) and Cannes (Retweet).

## 7    Conclusion and Future Work

In this paper, we introduce a new link prediction framework, MLP (Multiplex Link Prediction), that employs a holistic approach to accurately predict links in dynamic multiplex networks using a collection of topological metrics, the temporal patterns of link formation, and overlapping edges created by network coevolution. Our analysis on real-world networks created by a variety of social processes suggests that MLP effectively models multiplex network coevolution in many domains.

The version of Borda's method used in this research assigns the same weight to all rankers. However, for different networks, each scoring method might add differing value to the final scoring matrix. In future work, it would be interesting to use weighted Borda to calculate final scores. Also, while using more network features often increases the performance of a link prediction algorithm, this might not be true for all networks. Thus it may be useful to employ a feature selection algorithm to identify the best subset of unsupervised methods to be used in MLP, based on performance improvements in early snapshots. Finally, in this research it is implicitly assumed that the existence of links in other layers increases the probability of link formation in the target layer. A promising direction for future research would be to modify our reweighting procedure to account for negative cross-layer influences, where connections in one layer lower the target link likelihood.

### Acknowledgments

### References

1. Hajibagheri, A., Lakkaraju, K., Sukthankar, G., Wigand, R.T., Agarwal, N.: Conflict and communication in massively-multiplayer online games. In: Social Computing, Behavioral-Cultural Modeling, and Prediction. Springer (2015) 65–74

2. Omodei, E., De Domenico, M., Arenas, A.: Characterizing interactions in online social networks during exceptional events. arXiv preprint arXiv:1506.09115 (2015)
3. Scott, J.: Social Network Analysis. Sage (2012)
4. Kivela, M., Arenas, A., Barthelemy, M., Gleeson, J., Moreno, Y., Porter, M.: Multilayer networks. Journal of Complex Networks **2** (2014) 203–271
5. Hajibagheri, A., Sukthankar, G., Lakkaraju, K.: Leveraging network dynamics for improved link prediction. In: Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction, Washington, D.C. (June 2016)
6. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology **58**(7) (2007) 1019–1031
7. Menon, A.K., Elkan, C.: Link prediction via matrix factorization. In: Machine Learning and Knowledge Discovery in Databases. Springer (2011) 437–452
8. Scellato, S., Noulas, A., Mascolo, C.: Exploiting place features in link prediction on location-based social networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2011) 1046–1054
9. Beigi, G., Tang, J., Liu, H.: Signed link analysis in social media networks. arXiv preprint arXiv:1603.06878 (2016)
10. Tang, J., Lou, T., Kleinberg, J.: Inferring social ties across heterogenous networks. In: Proceedings of the ACM International Conference on Web Search and Data Mining. (2012) 743–752
11. Davis, D., Lichtenwalter, R., Chawla, N.V.: Supervised methods for multi-relational link prediction. Social Network Analysis and Mining **3**(2) (2013) 127–141
12. Hristova, D., Noulas, A., Brown, C., Musolesi, M., Mascolo, C.: A multilayer approach to multiplexity and link prediction in online geo-social networks. arXiv preprint arXiv:1508.07876 (2015)
13. Rossetti, G., Berlingerio, M., Giannotti, F.: Scalable link prediction on multidimensional networks. In: 2011 IEEE 11th International Conference on Data Mining Workshops, IEEE (2011) 979–986
14. Basu, P., Dippel, M., Sundaram, R.: Multiplex networks: A generative model and algorithmic complexity. In: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). (2015) 456–463
15. Lü, L., Zhou, T.: Link prediction in complex networks: A survey. Physica A: Statistical Mechanics and its Applications **390**(6) (2011) 1150–1170
16. Al Hasan, M., Zaki, M.J.: A survey of link prediction in social networks. In: Social Network Data Analytics. Springer (2011) 243–275
17. Zhang, J., Philip, S.Y.: Link prediction across heterogeneous social networks: A survey (2014)
18. Soares, P.R.d.S., Prudêncio, R.B.C.: Time series based link prediction. In: International Joint Conference on Neural Networks, IEEE (2012) 1–7
19. Gao, S., Denoyer, L., Gallinari, P.: Temporal link prediction by integrating content and structure information. In: Proceedings of the ACM International Conference on Information and Knowledge Management. (2011) 1169–1174
20. Pujari, M., Kanawati, R.: Supervised rank aggregation approach for link prediction in complex networks. In: Proceedings of the International World Wide Web Conference. (2012) 1189–1196
21. Pujari, M., Kanawati, R.: Link prediction in multiplex networks. Networks and Heterogeneous Media **10**(1) (2015) 17–35
22. Tabourier, L., Bernardes, D.F., Libert, A.S., Lambiotte, R.: Rankmerging: A supervised learning-to-rank framework to predict links in large social network. arXiv preprint arXiv:1407.2515 (2014)

23. Wang, X., Sukthankar, G.: Link prediction in heterogeneous collaboration networks. In: Social Network Analysis-Community Detection and Evolution. Springer (2014) 165–192
24. Newman, M.E.J.: Clustering and preferential attachment in growing networks. Physical Review E (2001)
25. Barabási, A.L., et al.: Scale-free networks: a decade and beyond. Science **325**(5939) (2009) 412
26. Adamic, L.A., Adar, E.: Friends and neighbors on the web. Social Networks **25**(3) (2003) 211–230
27. Zhou, T., Lü, L., Zhang, Y.C.: Predicting missing links via local information. The European Physical Journal B **71**(4) (2009) 623–630
28. Brin, S., Page, L.: Reprint of: The anatomy of a large-scale hypertextual web search engine. Computer Networks **56**(18) (2012) 3825–3833
29. Ding, Y.: Applying weighted PageRank to author citation networks. Journal of the American Society for Information Science and Technology **62**(2) (2011) 236–245
30. Acar, E., Dunlavy, D.M., Kolda, T.G.: Link prediction on evolving data using matrix and tensor factorizations. In: Workshops at IEEE International Conference on Data Mining. (2009) 262–269
31. Sculley, D.: Rank aggregation for similar items. In: SIAM International Conference on Data Mining. (2007) 587–592