Social Network Analysis and Mining (2021) 11:65
https://doi.org/10.1007/s13278-021-00775-7

ORIGINAL ARTICLE

Improving team performance prediction in MMOGs with temporal communication networks

Raji Ghawi[1] · Siegfried Müller[1] · Jürgen Pfeffer[1]

Received: 14 February 2021 / Revised: 5 June 2021 / Accepted: 13 July 2021 / Published online: 22 July 2021

Abstract
Virtual teams are becoming increasingly important. Since they are digital in nature, their "trace data" enable a broad set of new research opportunities. Online Games are especially useful for studying social behavior patterns of collaborative teams. In our study, we used longitudinal data from the massively multiplayer online game Travian collected over a 12-month period that included 4753 teams with 18,056 individuals and their communication networks. For predicting team performance, we selected several social network analysis-based attributes frequently used in team and leadership research. We find that using these features, the accuracy of predicting the team performance, in terms of R^2, is about 60%; whereas the accuracy of classifying the top-performing teams exceeds 95%. Moreover, we examine the ability to predict the team performance based on historic data of the network features, i.e., before several weeks. We find that the best accuracy can be achieved using the features in the present and the past, as well as the past performance. For a delay of one week, the accuracy of this model is about $R^2 = 97\%$.

Keywords Performance prediction · Virtual teams · Social network analysis · Communication network · Machine learning · Massively multiplayer online game

1 Introduction

Traditionally, teams have worked together at the same location, whereas today, virtual teams have become a reality in most organizations (Lipnack and Stamps 1999). A study conducted by the Society for Human Resource Management stated that approximately 66% of multinational organizations utilize virtual teams (Gilson et al. 2015). These new ways of collaboration produce a vast amount of "digital exhaust," as Leonardi & Contractor call the electronic traces created by modern communication technologies (e.g., e-mail, messenger or VOIP) (Leonardi and Contractor 2018).

When studying teams and their functioning, "communication has always been viewed as a key element" (Krackhardt and Hanson 2003). Especially when team members have never met in person, their communication "is often the only visible artifact of the group's existence" (Ahuja et al. 2003). "Relational theories have depicted leadership as socially constructed through communication exchanges" (Cullen-Lester et al. 2017). "Scholars adopting the social network approach further argue that by focusing on informal social contexts, i.e., social networks, researchers can examine 'how work really gets done in organization's (Cross and Parker 2004)" (Jokisaari 2016). Cross and Parker claim that "one has to examine how people are connected to each other and to focus on the wider social environment rather than formal dyadic relations between a leader and her followers" (Jokisaari 2016).

Beside traditional work environments, a very promising field of research for studying teams is the analysis of Online Games. Within these virtual worlds, thousands of players are organized in virtual teams (Chan and Vorderer 2006). The Harvard Business Review stated: "Online game leaders operate in a context that may well foreshadow the business environment of the future" (Reeves et al. 2008). *Massively Multiplayer Online Games (MMOGs)* in particular allow and also require cooperation and competition on a large scale.

✉ Raji Ghawi
raji.ghawi@tum.de

Siegfried Müller
siegfried.mueller@tum.de

Jürgen Pfeffer
juergen.pfeffer@tum.de

[1] Bavarian School of Public Policy, Technical University of Munich, Munich, Germany

Unlike in traditional experiments, the participants solve engaging problems and challenges. It is not necessary to incentivize participants since they are already highly motivated through the social interaction and the game design (Assmann et al. 2010). Castronova states that, even if "this place isn't 'real' by any means [...], it does feel real enough to the users that they can fairly easily immerse themselves in it for hours on end, month after month, year after year, in a sort of parallel existence" (Castronova 2008).

In our previous work, Müller et al. (2020), we conducted our research within the environment of an online simulation game, called *Travian*.[1] The game is organized in rounds lasting approximately one year. Therefore, repeated interaction between users plays an important role in the game's social ecosystem. Within the early phase, players team up with others to form alliances containing up to 60 members. These alliances are necessary to protect each other and to achieve the goal, which is to be the first to complete a monument at the end of the game. The game can only be won in cooperation and coordination. Therefore, intra-alliance communication plays an important role in succeeding. Communication takes place in an in-game messaging system (IGM) that is part of the game server. In addition to the IGM, where players can send self-written text messages to other team members, there is an internal forum that can be used for team discussions and information sharing. In our current work, we limit ourselves to the exchange of free text messages between individuals on an intra-team basis.

In Müller et al. (2020), the motivation for our research was to show that communication networks (or the information they contain) can be applied as predictors for team performance. Therefore, we developed two distinct models. We applied a *baseline model* to enable our prediction task to cover the main effects originating from the game design. This baseline model includes the age, the time since formation of the team, and the group size (N), which plays an important role as the game favors alliances bigger in size. Secondly, we built a *network model* that extends the baseline model by including 13 network attributes, commonly used in team and leadership literature. A major finding of Müller et al. (2020) was that the *network model* outperforms the *baseline model* in terms of the accuracy of predicting the team performance.

In this paper, we build upon Müller et al. (2020) and extend it by adding a dynamic perspective on the networks, incorporating a temporal dimension in order to improve the performance prediction. By bringing in historical data, we examine the ability to predict the team current performance based on the past performance and/or the features of communication networks (present and/or past).

Mainly, with a proper data preparation, we construct datasets that combine both the performance and the network features at two different time points (used as present and past). Then, with such datasets, we examine different predictive models based on different combinations of a) past performance, b) past network features, and c) current network features, in order to predict the current performance.

The contributions of this paper are as follows:

1. We demonstrate how social network patterns in communication networks can be applied to predict team performance.
2. We provide an overview of the ability of different machine learning approaches to deliver accurate prediction outcomes.
3. We examine the ability of various predictive models to predict the current performance based on different combinations of the past performance, and the past and current network features.
4. Finally, we deliver insight into a set of general aspects to consider when tackling the world of MMOG datasets.

The rest of the paper is organized as follows. Section 2 shows related work; while Sect. 3 describes the mechanics of the game, the dataset and our preprocessing steps; and Sect. 4 shows how we calculate network features. In Sects. 5 and 6, we describe how we analyze and predict performance; while Sect. 7 addresses the classifications of top-performing teams. Section 8 is dedicated to the performance prediction with historical data. Finally, Sect. 9 concludes the paper and describes limitations and future work.

2 Related work

Online games are not limited to their potential as a laboratory where leadership and its outcomes can be studied.

"Anthropologists see new cultures, entrepreneurs see new markets, lawyers see new precedence, and social and political experts see new pressures and looming crises" Castronova (2008).

Given the "scientific research potential of virtual worlds" as discussed in the 2007 article in Science (Bainbridge 2007), the fields of application are wide–especially in the area of team research, where massively multiplayer online games (MMOGs) and massively multiplayer role-playing games (MMORPGs) have been widely used (Pearce 2011; Chan and Vorderer 2006; Sourmelis et al. 2017). Assmann et al. assessed the "opportunities to overcome some limitations of traditional research environments" (Assmann et al. 2010). They point out that they "offer a unique opportunity to study virtual organizational structures" (Assmann et al. 2010). In communication research, for example, Gloor et al.

[1] https://www.travian.com/international.

Fig. 1 Travian - Screenshots from the game indicate different zoom levels (map, fields, and village) and final monument

(2008) have been working on how online communication behavior can be optimized and how it is influencing individual and team performance. Williams et al. (2011) conducted an interdisciplinary "study of behavior within a game and also game activities that parallel those in 'real life' ", whereas Korsgaard et al. (2010) worked in the area of "emergence and persistence of trust and cooperation, as well as [in the area of] the impact of different communication media for coordination and information management in virtual organizations". Other work investigated the effect of shared leadership within groups and its relationship with group trust development (Drescher et al. 2014). Further, MMOGs have been applied as research frameworks for military training and education (Bonk and Dennen 2005; De Freitas and Griffiths 2007). Even combat activities within these games have been studied (Huang et al. 2009; Suznjevic et al. 2011).

Regarding the prediction of team performance, team processes and leadership behavior have been studied frequently. Pobiedina et al. (2013), for example, have used role distribution, experience, the number of friends, and national diversity in Dota2 to study their influence on team performance. Prediction models on team performance have been applied and tested by Shim and Srivastava (2010); Shim et al. (2011) using data from EverQuest and Halo3. Tawa et al. (xxx) analyzed interpersonal interactions in experimental setups in *Second Life* to study the effects of resource competition on racial group interactions.

Working with data from Travian, Wigand et al. (2012) proposed using centrality measures from the game's message network as performance indicators and for predictive modeling.

3 Data

3.1 The world of Travian

Travian is a commercial *Massively Multiplayer Online Game (MMOG)* operated in 53 countries around the world. Up to 20,000 users play at any one time in game worlds adapted to the local market. The game world used for this study (travian.de) is a version that has been localized for German-speaking countries. The players start with one village where they grow resources, level up their infrastructure, and build armies to protect their kingdom. Troops can also be used to raid resources from other players, instead of producing those resources themselves, or to fight wars to conquer new territories (Fig. 1).

In addition to founding and developing new villages, the most important aspect of the game is to be part of an alliance. The environment of the game is highly competitive and only a high degree of cooperation allows a team to survive and achieve its goals. The alliance leaders are highly dependent on the contribution of every single member. Therefore, there is a great amount of social pressure to take things seriously and to invest a significant amount of time. Players who do not show a certain amount of commitment and/or performance (e.g., growth rate) are not invited to join alliances or are even dismissed. Alliance leaders face a trade-off when it comes to achieving a high-ranking position. The easiest way to increase alliances ranking is to invite additional members to the alliance. But doing this comes at a price. Leading and coordinating bigger groups/organizations are challenging and evidence from the game shows that often a smaller team of highly experienced players is more effective in reaching their goals. Therefore, some-not all-top-ranked alliances opt to remain small in number rather than expand to include the maximum 60 members allowed. Looking at the data, it should also be mentioned that it takes a critical size of about 35–40 members to even be able to rank among the top-ranked alliances.

To enable communication between players, the game provides an in-game messaging system and an (internal) forum that can only be accessed by a specific alliance. For our study, we used messages sent via the IGM, which means that our data collection has been non-obtrusive and not reactive. All players were informed by the game operator about their (anonymous) participation in a scientific research project to which they agreed by accepting the general terms and conditions. Completing additional surveys[2] has been voluntary and had no impact on regular participation in the game.

[2] Not part of this study.

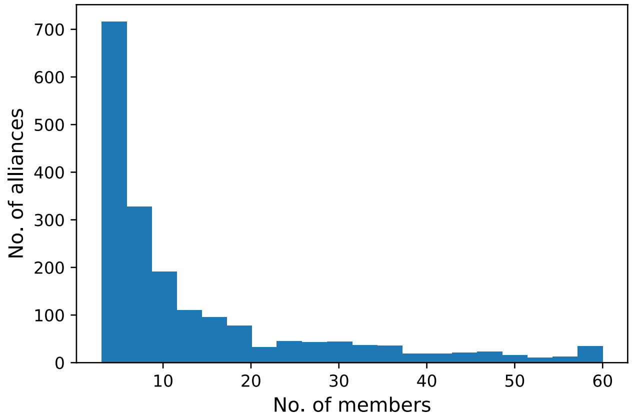


Fig. 2 Distribution of alliance size

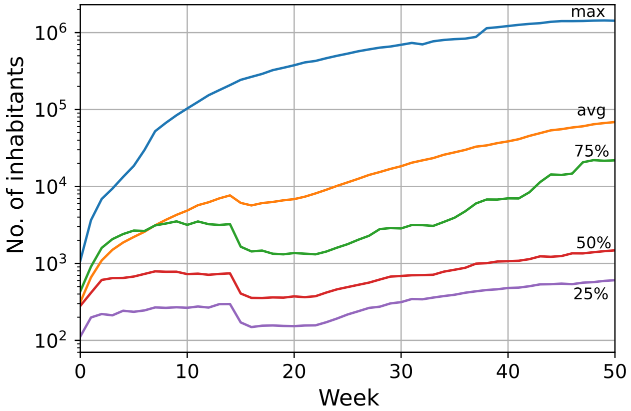


Fig. 3 Number of inhabitants per alliance over time

Alliances (as teams are called) can be established by players whose villages reach a certain threshold/number of inhabitants. An invitation is required for joining a team. The game tracks when this invitation has been sent and when it has been accepted. The same applies when members are leaving the team or have been dismissed. Teams can therefore be regarded as having clearly defined boundaries.

3.2 The dataset

In 2009/10, the operator Travian Games GmbH granted access to its game databases, which enabled an extensive data collection for scientific research. The operator of the game provided a daily download of a *cleaned* version of the game database (MySQL). The majority of the players were from the German-speaking countries: Germany, Austria, and Switzerland. Participants were 77% male, averaging 30.3 years old. 62% had a permanent employment. To comply with privacy protection, the operator removed all personal information and communication content before sharing the data with the researchers.

Alliance size ranged from 2 to 60 members, which is the maximum number of members that the game design allows. On average, the size of a group on any day is 14.5 individuals. As shown in Fig. 2, the distribution of group sizes is skewed with a long tail to the right, meaning that: many groups have small sizes, while few groups have large sizes, up to 60 members[3].

A total of 4758 alliances have been formed during this particular game. The data collection period was 51 weeks (356 days). Using this raw data, we extracted the following two datasets:

3.2.1 Performance dataset

The game *Travian* uses specific rankings, also referred to as *alliance rankings*, to indicate alliance performance. Rankings are based on the sum of inhabitants each alliance member has. The number of inhabitants a player has under him increases each time the player's infrastructure is upgraded. The alliance with the most inhabitants is rated as number one, the alliance with the second-most inhabitants as number two, and so on. Rankings within the game are calculated in real time. Since our raw data only contained one data point (MySQL snapshot) per day, we reverse engineered the ranking algorithm and adapted it via aggregation to a weekly measure. The decision for weekly aggregation was based on preliminary analysis of the data to avoid artifacts from the daily snapshots.

As the game proceeds, player's villages develop, and the overall number of inhabitants increases constantly. Figure 3 shows how the number of inhabitants evolves over time. In order to use the number of inhabitants as a performance measure, we needed to normalize it in a way that makes it comparable across weeks since start of the game world. Thus, we used min–max normalization on a weekly basis. Let $H(a, w)$ denote the number of inhabitants of alliance a at week w, then: $H_{min}(w) = \min_a\{H(a, w)\}$ and $H_{max}(w) = \max_a\{H(a, w)\}$ are, respectively, the minimum and maximum number of inhabitants per alliance at week w. The performance $P(a, w)$ of alliance a at week w is then stated as:

$$P(a, w) = \frac{H(a, w) - H_{min}(w)}{H_{max}(w) - H_{min}(w)}$$

[3] Over time, players join and leave alliances, and hence, the overall size of a group (over a long period of time) can easily exceed 60 members.

Table 1 Statistics of datasets

Dataset	Performance	Communication
No. of records	53,766	526,002
No. of alliances	4753	2074
No. of weeks	51	52
No. of alliance-week pairs	53,766	16,532

3.2.2 Communication dataset

This dataset indicates intra-alliance communications among players as expressed on a weekly basis. Each entry associates the IDs of two players: the sender and the receiver of a message, with the alliance ID (of which the sender and receiver are members) and the week ID (during which the message was sent).

Table 1 provides some statistics about both datasets, including the number of records, number of alliances, number of weeks, and number of alliance-week pairs.

3.3 Pre-processing

From Table 1, we observe that the two datasets have different numbers of alliances, weeks, and alliance-week pairs; hence, there are some incompatible data entries. This is due to the availability of raw data, with only one data point per day available for performance data, as opposed to the availability of real-time communication data. For instance, there are some data entries that appear in the performance dataset but not in the communication dataset, and vice versa. Moreover, in some alliance-week pairs, the number of alliance members in the performance dataset is different from the number in the communication dataset. To fix these issues, we performed the following pre-processing steps.

- Since we did not possess performance data within the first week of existence of some alliances, we opted to exclude this first week of *all* alliances.
- Since some alliances have missing communication information at the end of their lifespan, we opted to exclude the last week(s) of those alliances.
- To fix the discrepancy in the number of alliance members between the two datasets, we opted to use the maximum of these two numbers as the number of alliance members, for all alliance-week pairs. This step allowed us to minimize data loss by merging all available information from the two data sets.

Overall, as a result of pre-processing steps, we got rid of incompatible data. To this end, the communication dataset consists of (the remaining) 14,954 alliance-week pairs (corresponding to 50 weeks, and 1,852 alliances). The number of remaining entries is reduced to 510,285 (97%).

Figure 4 gives an overview of the distribution of alliances over time. Figure 4-a shows a histogram of the alliance age (in weeks), where we observe a skewed relationship between the age and the number of alliances having that age (survived that number of weeks). Most alliances have a relatively short lifespan, whereas few alliances survived for an extended period of time. Figure 4-b shows how the number of alliances changes over the entire period of the game.

4 Communication networks

Based on the communication dataset, we constructed the communication network as a directed graph for each alliance-week pair. Since we are interested in network structure, not in communication frequency, we opted to use the unweighted version of the graph. In this type of network, the nodes are the alliance members, and an edge links a

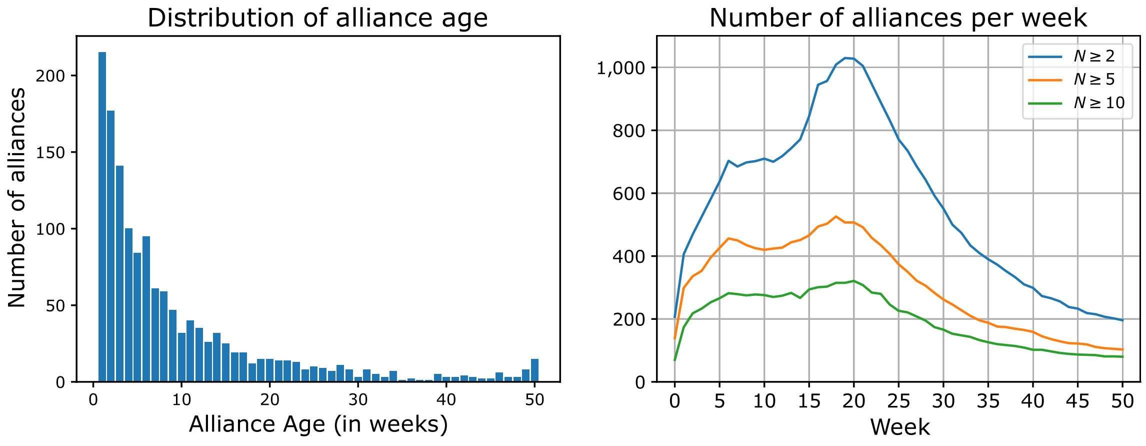
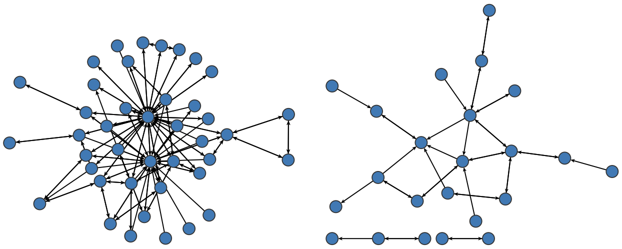


Fig. 4 Alliance distribution over time (N: alliance members)



(a) Alliance 203, week 12 **(b)** Alliance 2, week 45

Fig. 5 Two examples of communication networks

node u to another v, whenever the member represented by u sends one or more messages to another player represented by v; i.e., whenever there is an entry in the communication dataset that associates u to v with the corresponding alliance and week. Figure 5 shows two examples of communication networks.

Overall, we have communication networks for 14,954 alliance-week pairs (corresponding to 1,852 alliances, and 50 weeks). In addition to the number of nodes, N, and the number of edges, E, we calculated several network metrics for each network, including: density, average in-degree, transitivity, reciprocity, centralization, and k-core:

- **Density**: the ratio of the number of actual edges to the number of possible edges:

$$\text{density} = \frac{2E}{N(N-1)}$$

- **Average in-degree** (avg_din).
- **Transitivity**: the fraction of present triangles to all possible triangles (triads).
- **Reciprocity**: the ratio of the number of edges pointing in both directions to the total number of edges.

4.1 Centralization

In network analysis, centrality is a node-level index of the structural importance of nodes. Many metrics have been developed in the literature to measure the centrality of nodes, including degree centrality, closeness centrality, and betweenness centrality (Freeman 1978; Wasserman and Faust 1994). Let $c_1, \cdots, c_n$ be node-level centrality measures, where c_i is the centrality of node i by some metric. It is often useful to standardize the c_i 's by their maximum possible value: $\widetilde{c_i} = c_i/c_{max}$

While *centrality* is a node-level index, *centralization* is a group-level index that refers to how centralized the network is, i.e., to what extent the network is dominated by the most central node. Let $c^* = max\{c_1, \cdots, c_n\}$. Let $S = \sum_i [c^* - c_i]$. Then $S = 0$ if all nodes are equally central; S is large if one node is more central than the other nodes. Thus, network centralization is stated as:

$$C = \frac{\sum_i [c^* - c_i]}{\max \sum_i [c^* - c_i]}$$

where the "max" in the denominator is over all possible networks. With this formula, we get $0 \leq C \leq 1$. In particular, $C = 0$ when all nodes have the same centrality (e.g., cycle), whereas $C = 1$ if one actor has maximal centrality and all others have minimal (e.g., star).

As such, degree centralization is given by:

$$C^d = \frac{\sum_i [c^{d*} - c_i^d]}{2(N-1)(N-2)}$$

closeness centralization:

$$C^c = \frac{2N-3}{3(N-1)(N-2)} \sum_i [\widetilde{c}^{c*} - \widetilde{c}_i^c]$$

betweenness centralization:

$$C^b = \frac{\sum_i [\widetilde{c}^{b*} - \widetilde{c}_i^b]}{N-1}$$

For our alliance communications networks, we actually calculated five centralization metrics: three versions of degree centralization using in-degree, out-degree, and degree, as well as closeness and betweenness centralization.

4.2 k-Core

A k-core is a maximum subgraph that contains nodes of degree k or more. In our networks, for each node, we find the *core number*: $kcore(u), u \in G$, from which we then compute three network features:

- k-core k_{max}: $k_{max} = \max_{u \in G}\{kcore(u)\}$
- k-core size: number of nodes in the k-core, i.e., nodes whose core number is k_{max}.
- k-core relative size: fraction of nodes in the k-core to all nodes in the network.

To this end, we obtain a new dataset that summarizes the communication network of each of the $\langle$alliance, week$\rangle$ pairs (14,954 pairs). Where each pair is associated with 14 attributes.

As a new attribute, we introduce the *age* of the alliance, as we assume a strong interdependence with the maturity of the group. The age of an alliance at a given week (stated in weeks) is the number of weeks elapsed since the alliance creation. Formally, the age of an alliance a at week w is stated as:

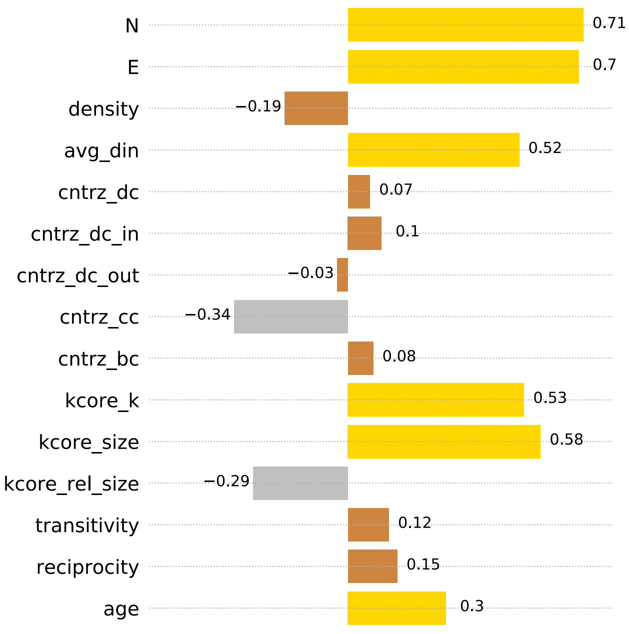


Fig. 6 Correlation of network attributes with the performance

$$\text{age}(a, w) = w - \text{firstweek}(a) + 1$$

where firstweek(a) denotes the first week within the life-span of the alliance a.

We also excluded the cases with the number of alliance members $N \leq 3$. The result of this step was the removal of 1,542 alliance-week pairs. Hence, the dataset then consisted of the remaining 13,412 alliance-week pairs.

The last step is to join the `network attributes` dataset with the `performance` dataset, such that for each alliance-week pair, we have the network features along with the performance of the alliance during that week.

5 Analysis

Now, since we have completed our dataset, we start looking at the features we have at hand. First, we look at the correlation of these features with the target feature, the performance of an alliance in a week. This can be seen in Fig. 6 (including alliance age). We observe that the features that are the most correlated with the performance are the number of nodes N (alliance members), and the number of edges E. Some other features also have a relatively high positive correlation with the performance, including the avg. in-degree, k-core k, k-core size, and age. There are also other features that show a relatively high negative correlation with the performance, including the closeness centralization (cntrz_cc), and k-core relative size (kcore_rel_size). The remaining features have weak, positive, or negative correlation, such as density, centralization, transitivity, and reciprocity.

In order to have insight into how the performance is related to each feature, Fig. 7 shows scatter plots of each feature with respect to the performance.

6 Performance prediction

In this section, we address the prediction of the alliance performance based on the network attributes. For this purpose, we used our final dataset, which comprises 13,412 records (alliance-week pairs) corresponding to 1,431 alliance over 50 weeks. Besides the alliance ID, the week, and the performance, it consists of 15 features including the alliance age and the attributes of the communication network. Since many of these features have their values on different scales, we opt to perform a min–max scaling of all features such that their values are in the range [0,1].

Then, we split the dataset into 80% training, and 20% test subsets. As a prediction algorithm, we used the classic linear regression approach (also known as Ordinary Least Squares (OLS)), as implemented in the linear regression module of python's scikit-learn library.[4]

For the evaluation of the prediction accuracy, we use the coefficient of determination (R^2), which is stated as:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

where y_i and $\hat{y}_i$ are the actual and predicted values of the target variable (the alliance performance in our case). The coefficient of determination is the proportion of the variance in the dependent variable that is predictable from the independent variables. Thus, it is a statistical measure of how well the regression predictions approximate the real data points. An R^2 of 1 indicates that the regression predictions perfectly fit the data.[5]

In this paper, we consider two models for the prediction task:

- **Baseline model**: The purpose of this model is to cover the main effects originating from the game design, i.e., the features that are not related to the communication network, namely the number of alliance members N, and the alliance *age*. Actually, having more alliance members

[4] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html.

[5] In Müller et al. (2020), we used the adjusted R^2 (see Theil 1961), however, we stick with R^2 in this paper since the number of features is small comparing to the number of instances; hence, there is no need for the adjusted R^2.

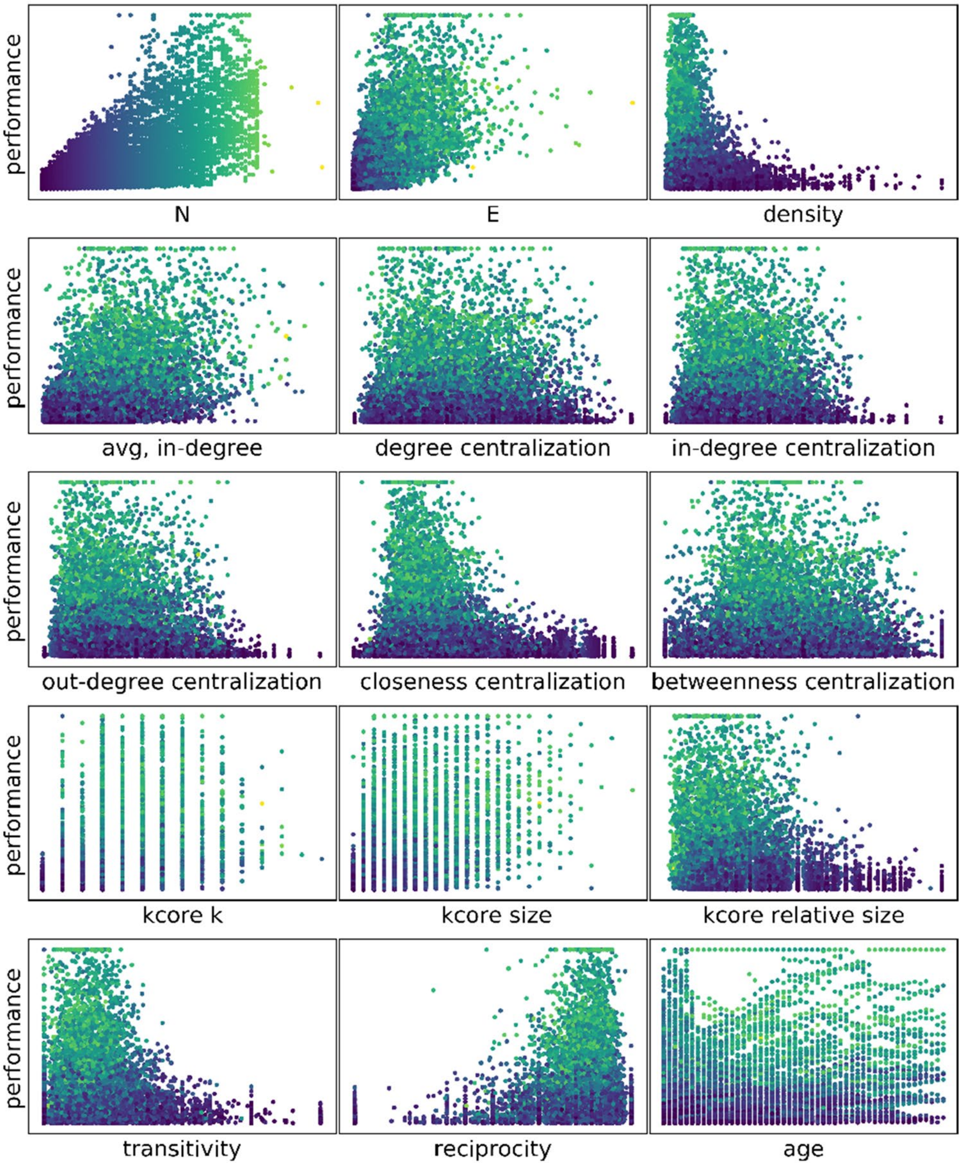


Fig. 7 Scatter plots of network features with performance. The data points are colored based on *N*, the number of nodes/ members (lighter points indicate more members)

automatically leads to a higher-ranking position. Holding a certain limit in alliance members is required, but not sufficient for reaching a high-ranking position. Therefore, we included *group size* (*N*) to capture this effect. Secondly, groups need time to form and to arrive at the *performing stage* (Tuckman and Jensen 1977). Therefore, we included time since foundation of the alliance (*age*).

- **Network model**: As a second step, we extended the baseline model by adding 13 network features derived from the intra-alliance communication networks. To capture (collective) leadership structures, we included *density*, *centralization* (D'Innocenzo et al. 2016; Nicolaides et al. 2014), and *k-core* (Seidman 1983; Contractor et al. 2012). We used *average in-degree* to track prestige (Moreno 1946). Finally, we included *transitivity* and *reciprocity* to cover the most important structural tendencies (Wasserman and Faust 1994).

Namely, the baseline model comprises two features: the number of nodes *N*, and the *age* of the alliance. The network model comprises 15 features: N, E, density, avg_din, cntrz_dc, cntrz_dc_in, cntrz_dc_out, cntrz_cc, cntrz_bc, kcore_k, kcore_size, kcore_rel_size, transitivity, reciprocity, and age.

We used the 80% training set to train the linear regression model, and the 20% test set is then used to test the trained model.

The prediction results, in terms of R^2, for the two models (baseline, and network) are shown in Fig. 8. We can see that while the accuracy of the baseline model is 0.52, the network model achieves an accuracy of 0.595. This corresponds to

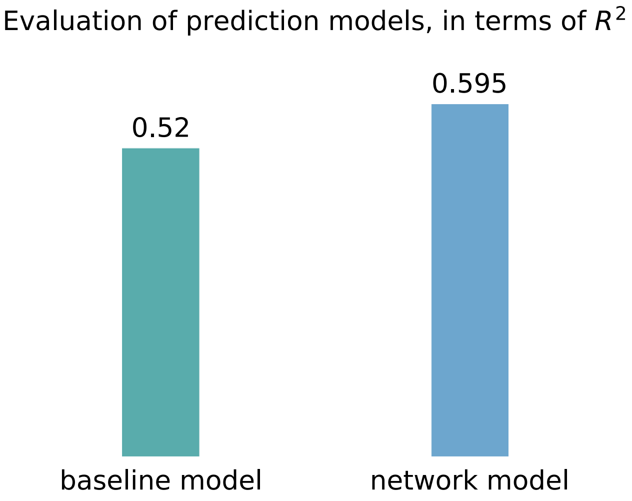


Fig. 8 Prediction results

14% increase with respect to the accuracy of the baseline model.

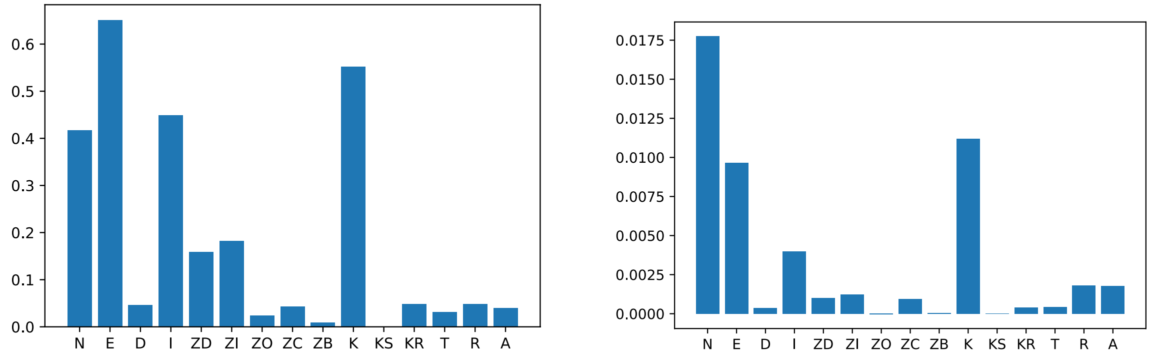
In fact, in our previous work, Müller et al. (2020), we had a third model which involved the logarithms of several features; thus, it outperformed the network model. We also considered other two outcomes besides the performance, namely the log performance, and the square root of the performance. The accuracy of predicting those outcomes was slightly higher than predicting the 'pure' performance. However, in this paper, we skip the log model and these two outcomes for several reasons, including: 1) to focus on the original (raw) values of the features and the performance, 2) to be consistent with the rest of this paper, and 3) for sake of brevity.

Now in order to have insight into the data and the prediction model, let us have a look on the importance of features in predicting the performance. Feature importance scores are assigned to input features based on how useful they are at predicting a target variable. Many techniques can be used to assign feature importance scores. In this work, we opt to use two techniques:

- Coefficients as Feature Importance: Linear machine learning algorithms, such as linear regression, fit a model where the prediction is the weighted sum of the input values. These algorithms find a set of coefficients to use in the weighted sum in order to make a prediction. These coefficients can be used directly as a crude type of feature importance score. In our case, this is possible since the features are already scaled (using min–max transformation). We can also ignore the sign values because negative sign states an inversely proportional correlation.
- Exclusion-based Feature Importance: Basically, we repeatedly: 1) exclude each feature, 2) run the linear regression algorithm using the rest of features, and 3) assess the drop in accuracy in comparison to the original model (using all features). The higher the difference is, the more important the excluded feature is.

In our case, we used both of these techniques to assess the importance of all the 15 features, in comparison with the network model (which uses all of them as predictors). The results are shown in Fig. 9.

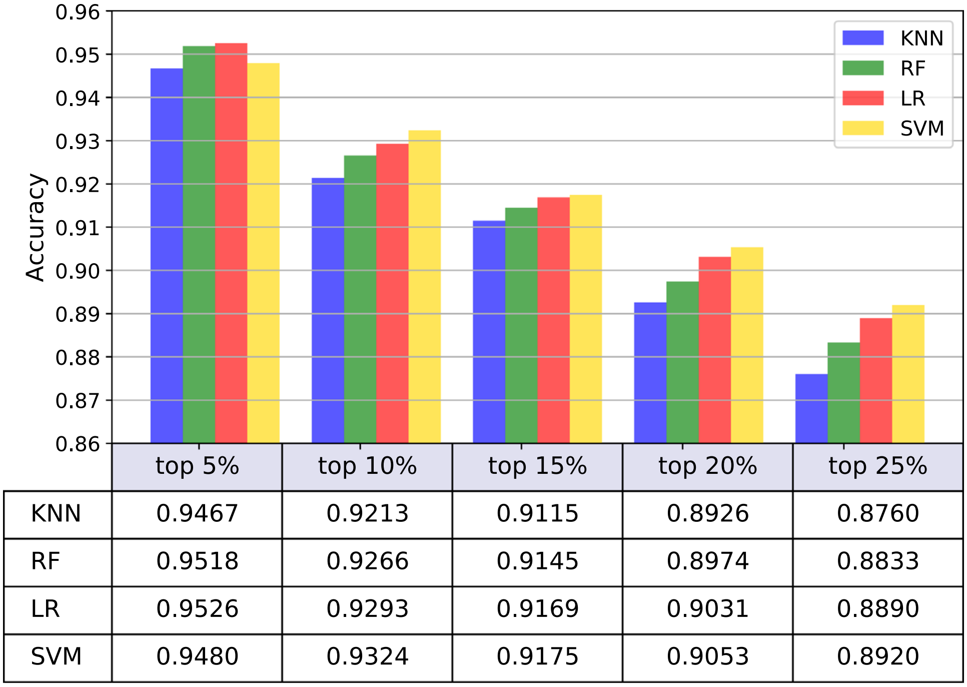
We can see that both techniques provide similar results of the importance of features. For instance, we can observe that the most important features are: the number of nodes N, the number of edges E, the average in-degree, and k_{max} of the k-core. In particular, if we use only these four features as predictors, the prediction accuracy, in terms of R^2, will be 0.581 which corresponds to 98% of the accuracy of the network model. On the other hands, we can see that the least important features are: k-core size, betweenness centralization, and out-degree centralization. In particular,



(a) based on Coefficients

(b) based on Exclusion

Fig. 9 Feature importance



	top 5%	top 10%	top 15%	top 20%	top 25%
KNN	0.9467	0.9213	0.9115	0.8926	0.8760
RF	0.9518	0.9266	0.9145	0.8974	0.8833
LR	0.9526	0.9293	0.9169	0.9031	0.8890
SVM	0.9480	0.9324	0.9175	0.9053	0.8920

Fig. 10 Classification results

if we exclude these three features from the network model, the prediction accuracy will not change.[6]

7 Classification of top-performing alliances

In this section, we address the problem of classifying the top alliances based on their performance. First, we need to specify what the top-performing alliances are. To do so, we choose a threshold α (e.g., 10%), and then for each week, find the $\tau = (1 - \alpha)\%$ quantile of the performance during that week (e.g., 90% quantile). We constructed a binary variable (target) that indicates whether an alliance is among top-performing alliances. Each alliance having a performance greater than or equal to τ during that week is considered as top-performing alliance, i.e., $target = 1$; otherwise, $target = 0$. Thus, the classification turns out into a binary classification task.

To address this binary classification task, we used four different classification approaches:

- **kNN**: k-nearest neighbors ($k = 45$).
- **RF**: Random forest classifier (nr. estimators=100).
- **LR**: Logistic regression.
- **SVM**: Support vector machine (*linear* kernel).

The features used for the classification are all 15 features in our final dataset (including N, E, and age). Thus, this corresponds to the **network model** as mentioned earlier in the prediction section. (No logarithm features are used.) Moreover, all the features are transformed using min–max scaling, such that each feature falls within the [0,1] range.

The evaluation of the classification is tackled using the *accuracy* metric, which is the fraction of correctly classified instances to all instances (in the test set). In all the classification experiments, we used cross-validation over fivefold, where the reported accuracy is the average over the fivefold classifications. The results of the classification, in terms of accuracy, for the five different thresholds (from top 5% to top 25%), and for the four classification approaches (kNN, RF, LR, and SVM), are shown in Fig. 10.

First, we observe that for any classification approach, the classification accuracy decreases as we increase the threshold of the top alliances. For instance, when we classify the top 5% alliances, the accuracy is about 95%, whereas when we classify the top 20% alliances, the accuracy is about 90%. Second, when we compare the different classification approaches, we observe that in general, the best classifier is SVM, followed by logistic regression, followed by random forest, where kNN is the least accurate classifier.

[6] Recall that the k-core k_{max} is the maximum of the core number and indicates the strength of the connections within the k-core, while the k-core size is the number of nodes in the k-core, i.e., nodes whose core number is k_{max}. Here, k_{max} seems to be an important feature, while the k-core size is not!

8 Performance prediction with temporal data

In this section, we add a dynamic perspective to our study, by incorporating a temporal dimension and considering the past events. We examine the ability to improve performance prediction by bringing in historic data. That is, we aim at predicting team performance either using the past performance, or the features of the communication network (past and present), or both.

8.1 Prediction models

Since in our case, the time is indexed by weeks, let $\mathcal{N}^{[w]}$ denote the network features[7] at a time point w, i.e., current week, and let $\mathcal{N}^{[w-\Delta]}$ denote the network features at a past time point $w - \Delta$ which occurred Δ weeks before w, where Δ is a non-negative integer representing a timespan, in weeks, between the past and the present. In our study, we opt to use $\Delta \in \{1, 2, \cdots, 8\}$. Moreover, let $p^{[w]}$ and $p^{[w-\Delta]}$ denote the team performance at the present week w and past week $w - \Delta$, respectively.

With these notations, what we have done in previous sections is actually predicting $p^{[w]}$, as a target variable, using $\mathcal{N}^{[w]}$ as features (or equivalently, predicting $p^{[w-\Delta]}$ using $\mathcal{N}^{[w-\Delta]}$), i.e., predicting the team performance at a given week using the features of communication network at that same week. Henceforth, we will refer to this model as the **basic network model**, as it does not involve any time delay (Δ is irrelevant).

In the following, we will symbolically express a predictive model as:

$$\mathcal{Y} \sim f(\mathcal{X})$$

which means that a dependent variable $\mathcal{Y}$ (on left-hand side) is expressed as a linear function in terms of a set of independent variables $\mathcal{X}$ (on the right-hand side).

With this notation, we can symbolically express the *basic network model* as:

$$p^{[w]} \sim f(\mathcal{N}^{[w]})$$

which means that we seek to express the performance $p^{[w]}$ (dependent variable) as a linear function in terms of the network features $\mathcal{N}^{[w]}$ (independent variables). Besides this basic network model, we examine several other models:

A. *Past Performance model* (`PP`): This model predicts the current performance $p^{[w]}$ using only the past performance $p^{[w-\Delta]}$:

$$p^{[w]} \sim f(p^{[w-\Delta]})$$

B. *Past Network model* (`PN`): This model predicts the current performance $p^{[w]}$ using the past network features $\mathcal{N}^{[w-\Delta]}$:

$$p^{[w]} \sim f(\mathcal{N}^{[w-\Delta]})$$

C. *Past Performance, and Past Network model* (`PP-PN`): This model predicts the current performance $p^{[w]}$ using the past performance $p^{[w-\Delta]}$ and the past network features $\mathcal{N}^{[w-\Delta]}$:

$$p^{[w]} \sim f(p^{[w-\Delta]}, \mathcal{N}^{[w-\Delta]})$$

D. *Past Performance, and Current Network model* (`PP-CN`): This model predicts the current performance $p^{[w]}$ using the past performance $p^{[w-\Delta]}$ and the current network features $\mathcal{N}^{[w]}$:

$$p^{[w]} \sim f(p^{[w-\Delta]}, \mathcal{N}^{[w]})$$

E. *Past- and Current Network model* (`PN-CN`): This model uses the network features, at present $\mathcal{N}^{[w]}$ and in the past $\mathcal{N}^{[w-\Delta]}$ to predict the current performance:

$$p^{[w]} \sim f(\mathcal{N}^{[w-\Delta]}, \mathcal{N}^{[w]})$$

F. *Past Performance, and Past- and Current- network model* (`PP-PN-CN`): This model uses the past performance $p^{[w-\Delta]}$ as well as the network features, at present $\mathcal{N}^{[w]}$ and in the past $\mathcal{N}^{[w-\Delta]}$ to predict the future performance:

$$p^{[w]} \sim f(p^{[w-\Delta]}, \mathcal{N}^{[w-\Delta]}, \mathcal{N}^{[w]})$$

Figure 11 shows a graphical representation of these models. We can see that in all these models, the target (dependent) variable is always the current performance $p^{[w]}$ (highlighted in blue), while the independent variables (highlighted in green) vary according to the model and can be either the past performance $p^{[w-\Delta]}$, the past network features $\mathcal{N}^{[w-\Delta]}$, the current network features $\mathcal{N}^{[w]}$, or any combination of them. In each model in Fig. 11, we used arrows to link independent variables (predictors) to the target variable.

8.2 Data preparation

We have shown in Sect. 4 that our dataset combines the features of the communication network of each alliance with the performance of that team, on a weekly basis. That is, for each alliance-week pair, we have the network features along with the performance of the alliance during that week.

Now, in order to examine the predictive models of historic data, we need to prepare the dataset such that it contains, besides the network features and the performance

[7] This includes all the 15 features mentioned in Fig. 6.

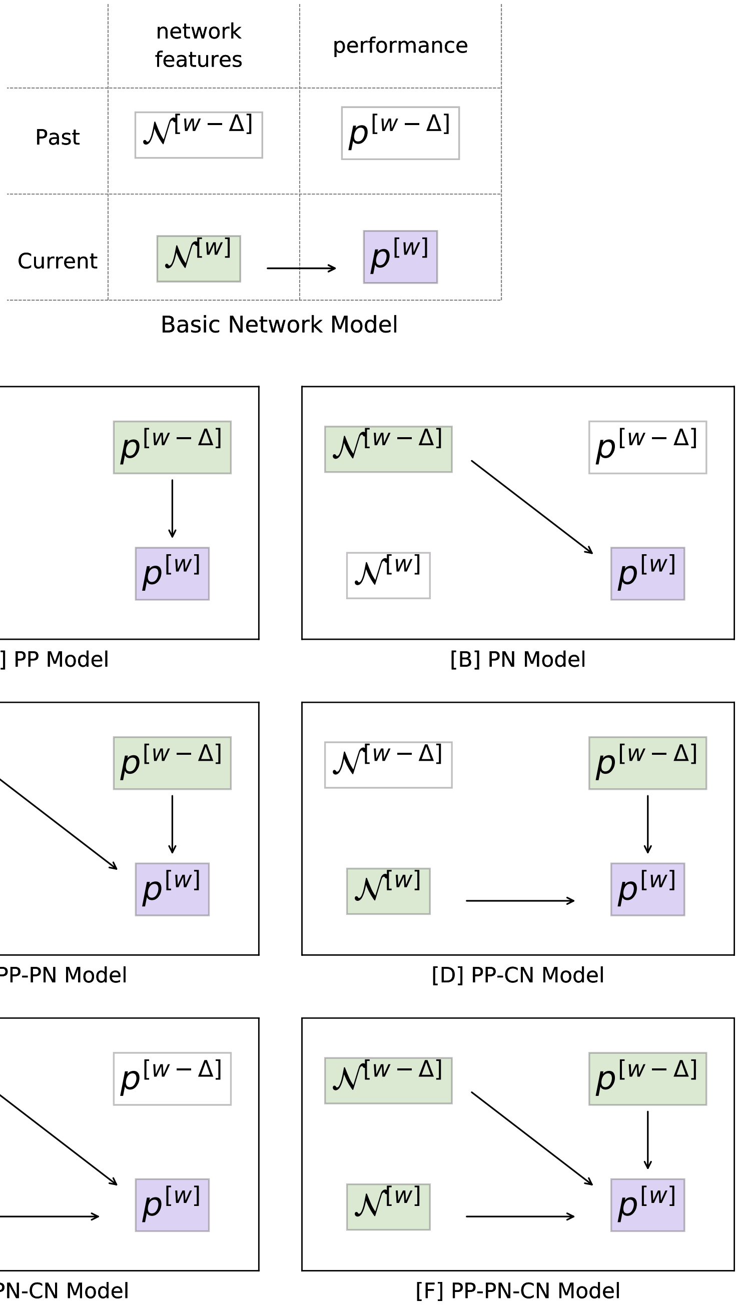


Fig. 11 Graphical representation of the different predictive models. The arrows point from independent variables (predictors) to the target variable. PP: past performance, PN: past network features, CN: current network features

at a present week w, the network features and the performance at a past week $w - \Delta$. Let us denote the dataset as $D(a, w, \mathcal{N}, p)$, where a is the alliance ID, w is the week, $\mathcal{N}$ is the set of features of communication network (such as N, E, age), and p is the performance of alliance a at week w.

For $\Delta = 1$, we create a new dataset D_1 by performing the following steps (demonstrated in Table 2):

- We make a copy of the dataset D, and we name it D', it will represent the past.
- We rename the columns of D' as follows: `w` becomes `w1`, `p` becomes `p1`, and each `x` in $\mathcal{N}$ becomes `x1`.
- We add to D' a new column `w_1`, whose values equal the values of the column `w1` plus 1:

$$D'[\texttt{w\_1}] := D'[\texttt{w1}] + 1$$

- Then, we join the two datasets, D and D', using $D[\texttt{a}] = D'[\texttt{a}]$ and $D[\texttt{w}] = D'[\texttt{w\_1}]$.

Table 2 Demonstrative example of preparing a dataset with current and past attributes

D

a	w	$\mathcal{N}$	p
a_0	3	$\cdots$	$\cdots$
a_0	4	$\cdots$	$\cdots$
a_0	5	$\cdots$	$\cdots$

D'

a	w1	$\mathcal{N}$1	p1	w_1
a_0	3	$\cdots$	$\cdots$	4
a_0	4	$\cdots$	$\cdots$	5
a_0	5	$\cdots$	$\cdots$	6

$D_1 := D \bowtie D'$

using `D[a]=D'[a]` and `D[w]=D'[w_1]`

a	w	$\mathcal{N}$	p	w1	$\mathcal{N}$1	p1	w_1
a_0	4	$\cdots$	$\cdots$	3	$\cdots$	$\cdots$	4
a_0	5	$\cdots$	$\cdots$	4	$\cdots$	$\cdots$	5
		$\mathcal{N}^{[w]}$	$p^{[w]}$		$\mathcal{N}^{[w-\Delta]}$	$p^{[w-\Delta]}$	

Notice that D and $D\prime$ contain the same data, but $D\prime$ has an additional column w_1, which indicates the next week (i.e., each entry in $D\prime$ with week w, the value of w_1 is $w+1$). Therefore, when we join the two datasets, such that the next week of $D\prime$ equals the current week of D, the content of $D\prime$ (w1, $\mathcal{N}$1, p1) will be the past with respect to the present content of D (w, $\mathcal{N}$, p).

This process is demonstrated by an example in Table 2, where we can see that D' (top right) has the same content as D (top left), but with renamed columns and an additional column w_1. The result of the join is a combination of the present attributes: w, $\mathcal{N}$, and p, and the past attributes: w1, $\mathcal{N}$1, and p1.[8] In this example, we see that the second row in D is joined with the first row in D'. Similarly, the third row in D is joined with the second row in D'.

We can also see that the size of the joined dataset is smaller than the original one, since there are some rows that are not included in the join. For instance, for any alliance, the row, which corresponds to the very first week of that alliance, will not be in the joined dataset, because it has no past.

The data preparation process is also repeated for $\Delta = 2, \cdots 8$. Thus, we obtain eight datasets D_Δ for $\Delta = 1, 2, \cdots, 8$, each of which combines the present attributes (week w, network features $\mathcal{N}$, and performance p) with the past ones (after Δ weeks).

Before proceeding to the predictive analysis, it is interesting to examine how the past network features are related to the current performance. Figure 12 shows, for each of the 15 features, the correlation of the past of that feature, over the different values of the timespan Δ, with the current performance (correlation of $x^{[w-\Delta]}$ with $p^{[w]}$ for $x \in \mathcal{N}$) ; in comparison to that correlation in the present (correlation of $x^{[w]}$ with $p^{[w]}$), as per the original dataset (see Fig. 6).

We can see that for many of the features, the correlation of their past with performance is lower than the correlation of their present with performance and decreases as Δ increases. Examples of such features include: N, E, avg in-degree, k-core k, and k-core size. This behavior indicates a degradation of the correlation with performance.

On the other hands, some features, such as density, do not change significantly. Other features, such as team age, exhibit an increase in correlation with the performance. This means that the past age has more impact on the performance than the present age, in other words, the age of the team has more impact on its performance in the future rather than in the present.

It is also interesting to examine the auto-correlation of the features; that is, the correlation of each feature in the present with itself in the past. Figure 13 shows the auto-correlation of each feature over the different timespans $\Delta = 1, \cdots, 8$.

We can see that the features have a positive, but slowly decreasing auto-correlation, as the timespan increases. However, the strength of the auto-correlation varies significantly according to the feature. For instance, the strongest feature, w.r.t auto-correlation, is the performance followed by N and E, whereas the weakest features are betweenness centralization and reciprocity.

Notice here that the strength or the weakness of the auto-correlation of a feature provides an indication of the robustness or volatility of that feature. For instance, having a very strong auto-correlation, the performance, N and E are robust features that do not change much over several weeks. In contrast, betweenness centralization and reciprocity, having weak auto-correlation, are volatile features that significantly change their values over time.

[8] After the joining, the column w_1 is not needed anymore and can be deleted.

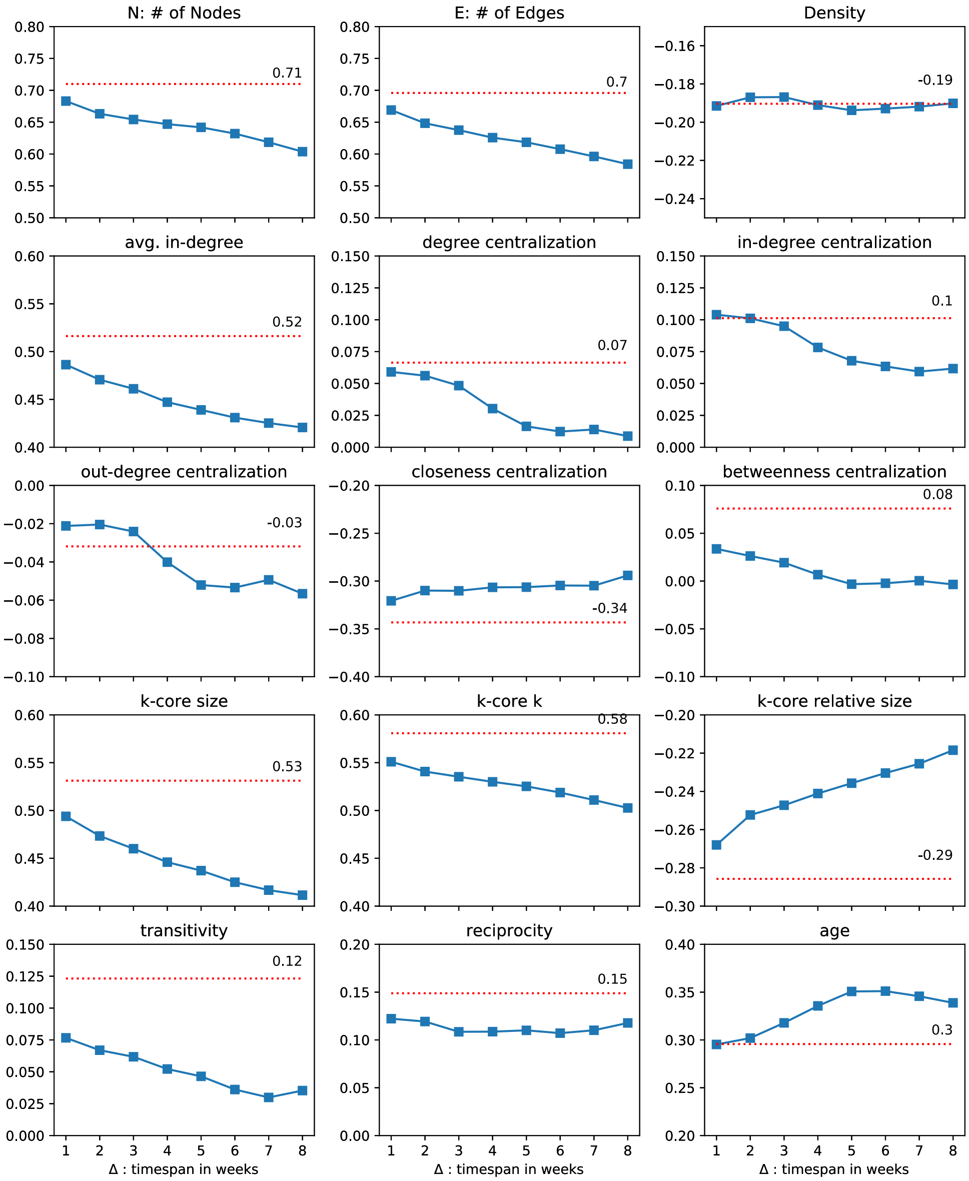


Fig. 12 Correlation of the past network features with the current performance for $\Delta = 1, \cdots, 8$

8.3 Prediction results

Using the eight datasets D_Δ for $\Delta = 1, 2, \cdots, 8$, we examine the different prediction models mentioned in Sect. 8.1.

For each dataset, we split it into 80% training and 20% test subsets. As a prediction algorithm, we use the classic linear regression approach (ordinary least squares, OLS). For the evaluation of the prediction accuracy, we use the coefficient of determination (R^2). We use the 80% training set to train the linear regression model, and the 20% test set is then used to evaluate the trained model in terms of R^2 as an evaluation measure.

The results are shown in Table 3 which depicts the accuracy, in terms of R^2, of the six prediction models over the eight datasets D_Δ for $\Delta = 1, 2, \cdots, 8$.

We can see that the dataset size decreases over time, i.e., as the timespan Δ increases. In particular, when the delay is eight weeks ($\Delta = 8$), the dataset size is less than the half of the size of the original dataset.

First, let us compare the models that use the network features only. Recall that the basic network model predicts the team performance in a week based on the network features of that same week. We have shown in Sect. 6 that this model has an accuracy of $R^2 = 0.595$. Two other models use the network features only, namely: `PN` model (past- network model), and `PN−CN` model (past and current network model).

Figure 14 shows the accuracy of those models, in terms of R^2, in comparison to the accuracy of the basic network model. We can observe that the accuracy of `PN` model is

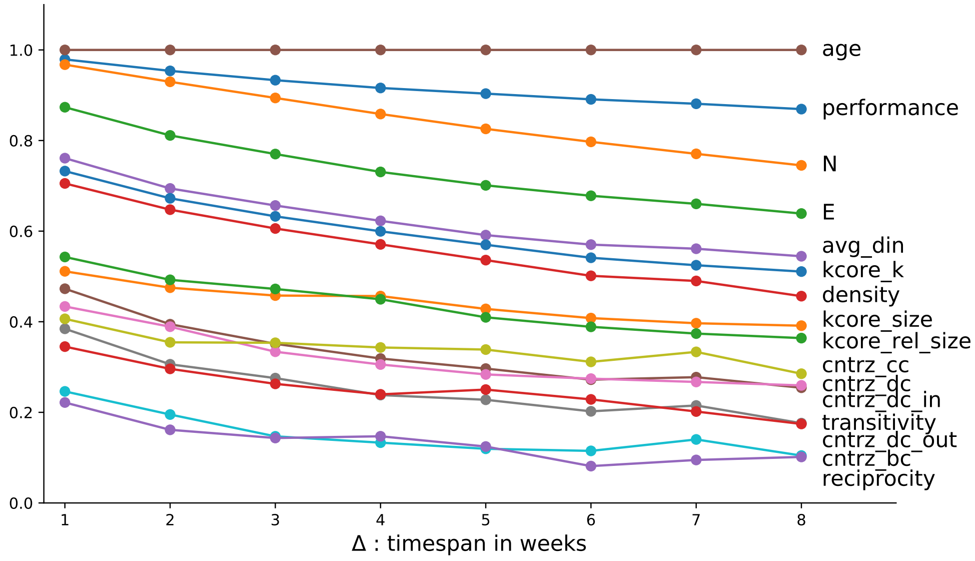


Fig. 13 Auto-correlation of each feature: correlation between the past and the present of each feature, for $\Delta = 1, \cdots, 8$

Table 3 Accuracy of prediction models, in terms of R^2, for $\Delta = 1, 2, \cdots, 8$

Δ	size	PP	PN	PP-PN	PP-FN	PN-FN	PP-PN-FN
1	11,415	0.961	0.545	0.961	0.964	0.594	0.972
2	10,315	0.910	0.528	0.911	0.922	0.608	0.937
3	9,453	0.872	0.492	0.874	0.893	0.608	0.913
4	8,644	0.833	0.504	0.840	0.870	0.625	0.893
5	7,907	0.804	0.475	0.816	0.854	0.637	0.878
6	7,254	0.784	0.486	0.793	0.847	0.648	0.868
7	6,689	0.758	0.452	0.767	0.833	0.622	0.856
8	6,155	0.785	0.449	0.792	0.842	0.644	0.859

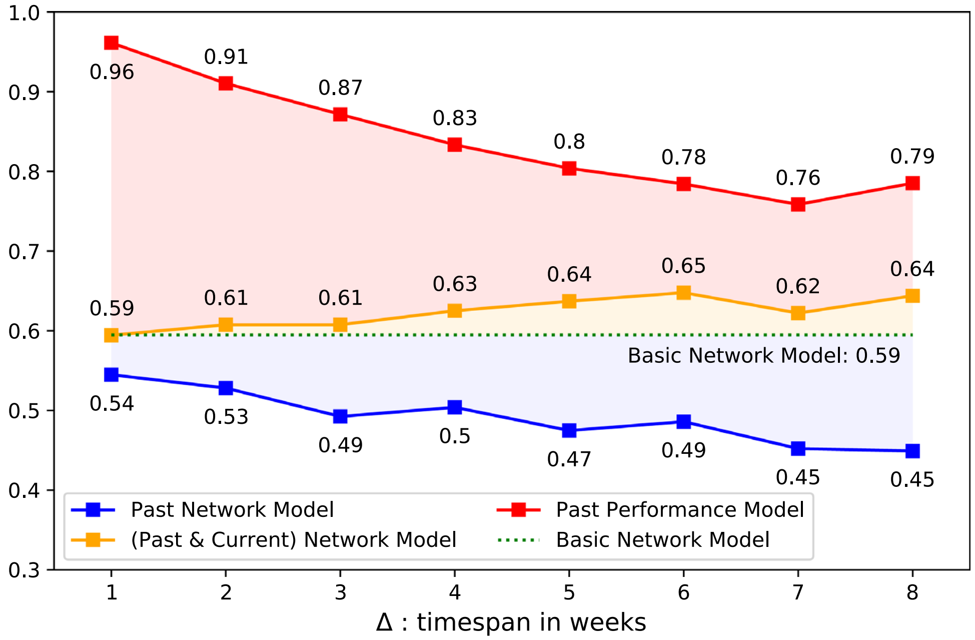


Fig. 14 Accuracy of the (Past) network model, the (Past and Current) network model, and the past performance model

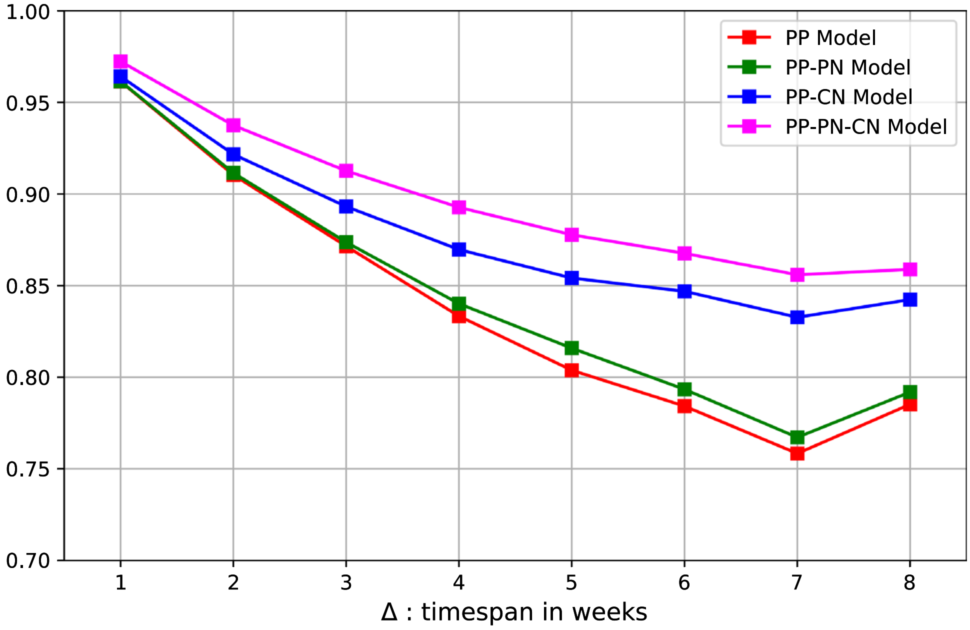


Fig. 15 Accuracy of the different models that involve the past performance

lower than the accuracy of the basic network model, and it decreases as the timespan Δ increases. In contrast, we can observe that the accuracy of the PN−CN model is higher than the accuracy of the basic network model, and it increases as the timespan Δ increases. The best accuracy is achieved when the delay is six weeks ($\Delta = 6$), with $R^2 = 0.65$. This means that in order to predict the performance of a team using the features of its communication network, it is better to know those features during the same week, than knowing them during any previous week; the longer that period is, the worse the accuracy gets. However, it is even better to know these features at both: the same week and a previous week; the longer that period is, the better the accuracy gets, up to a certain limit, namely a period of six weeks.

Now let us examine the predictive models that involve the (past) performance as a predictor. The first one of those models is PP model that uses the past performance only, to predict the current performance. As shown in Fig. 14, the accuracy of this model is generally very high. For instance, this accuracy is $R^2 = 0.96$ when the delay is one week, which means that we can 96% accurately predict the performance of a team at a current week by knowing only its performance last week. We can also see that the accuracy of this model decreases as the delay Δ increases; the longer the delay is, the lower the accuracy gets.

Clearly, the performance model, PP, is more accurate than the models that use network features only (basic network model, PN, and PN−CN). However, as the timespan Δ increases, the accuracy of PP model decreases, whereas the accuracy of PN−CN model increases, which suggests that with a long enough timespan, the PN−CN model would beat the PP model in accuracy[9].

Besides PP model, there are three models that are based on the past performance as a predictor: PP−PN, PP−CN and PP−PN−CN. All these models use the past performance and involve additional network features, namely PP−PN model involves past network features, and PP−CN model involves current network features, while PP−PN−CN model involves both past and current network features. Figure 15 shows how the accuracy of those models changes as Δ changes.

We can observe that these three models have higher accuracy than the PP model (which purely uses the performance), which means that the network features have an added value over the past performance in predicting the current performance.

At any given value of Δ, the PP−PN model has a slightly higher accuracy than the PP model, which means that adding the past network features to the performance improves its prediction accuracy.

Moreover, the PP−CN model has even a pretty higher accuracy than the PP−PN model. This means that to predict the team performance in a current week, given that we know its performance in a past week, it is better to also know the network features at this current week than knowing them at that past week.

Furthermore, for any value of Δ, the PP−PN−CN model has a higher accuracy than the PP−CN model. In fact, it is the most accurate model among all the examined models.

[9] This indeed happens for $\Delta \geq 14$ as we found in a subsidiary experiment!

This means that the best-known way to predict the current team performance is to know its past performance, as well as its network features in the present and the past.

9 Conclusion

The goal of this study was to find out whether it is possible to predict alliance performance using SNA-features from *communication networks*. Moreover, we wanted to test the ability of *classification tasks* to identify the best-performing alliances. Furthermore, we tested the ability to predict the current team performance using the past and current network features as well as past performance. In all cases, we have been able to show that it is possible to do this. Future research will help us to deepen our understanding of the underlying dynamics and enable us to apply our findings in a less specific context.

One major challenge we faced was the fact that we conducted our research within the environment of an online simulation game. We were able to track the interaction of 18,000 individuals, but we also had to learn that it is not an easy task to study these communication effects in isolation.

Despite the fact that the applied machine learning algorithms delivered excellent results in the classification tasks, we identified two effects that made it difficult to interpret the results outside the specific context: (1) the game's definition of performance and (2) the effect of group size on certain network attributes.

Definition of team performance: We opted to define success in the same way the game does. By using a slight modification to the official alliance ranking, we were able to ensure that our definition of performance matched the player's incentives provided by the game design. With this clear advantage, we faced a challenging hurdle: the **ranking** is highly influenced by **group size (N)**. As described above, alliance leaders face a trade-off. One option is to add as many members as possible to the alliance. Having more members automatically leads to more inhabitants, which leads to a higher position in ranking. On the other hands, it is more difficult to coordinate a bigger group as opposed to a small team of highly experienced players. Evidence from the game shows that both strategies have been applied successfully for top-performing teams. Nevertheless, there is a clear restriction. Figure 7 shows that alliances need to exceed a certain number of members (about $N > 35$) to be able to reach a top position in ranking. However, it is not sufficient to have many members in order to become a highly ranked team. We were able to show that the additional information extracted from the communication networks ia able to make the difference. Applying these measures makes it possible to successfully forecast team performance.

Effect of group size on network attributes: One critical effect is that **N** is included in the formulas used to calculate certain **network attributes**. This leads to an unwanted dependency between these network attributes and N. Hence, the correlations of density, avg. in-degree, centralization, and k-core show two different effects (the intended network effect *and* the indirect effect of N). Neither effect can be separated from the other.

Given these limitations, our future work will focus on eliminating these restrictions, which will make it possible to generalize our findings, i.e., to better explain team dynamics in real-world work teams.

One approach could be to (1) develop alternative measures for team performance that are either not or are less correlated with group size N. Further, it will be helpful to split the dataset to be able to (2) take group size into account (e.g., separate small and big teams). We also propose (3) refining the theoretic foundation. In this study, we have already implemented insights from team and leadership research. Additional theoretical models such as outside connectivity, core-periphery structure, or the role of strong and weak ties can be expected to improve prediction results.

As we have shown in our paper, the opportunities for conducting research into online games are manifold. Researching online gaming is a very promising field, especially in view of the vast amount of data, it can offer. We also demonstrated how important it is to oversee the effects coming from the special environment of these virtual worlds. The opportunities in this field are promising and will be even more so once these very special frameworks and their limitations are better understood and mastered.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Ahuja M, Galletta D, Carley K (2003) Individual centrality and performance in virtual r&d groups: an empirical study. Manag Sci 49(1):21–38

Assmann J, Drescher M, Gallenkamp J, Picot A, Welpe I, Wigand R (2010) MMOGs as emerging opportunities for research on virtual organizations and teams. In: 16th Americas Conference on Information Systems 2010, vol 1, pp 232–240

Bainbridge W (2007) The scientific research potential of virtual worlds. Science 317(5837):472–476

Bonk C, Dennen V (2005) Massive multiplayer online gaming: a research framework for military training and education. Indiana University, Bloomington, Tech. rep

Castronova E (2008) Synthetic worlds: the business and culture of online games. University of Chicago press

Chan E, Vorderer P (2006) Massively multiplayer online games. In: Vorderer P, Bryant J (eds) Playing video games: motives, responses, and consequences. Lawrence Erlbaum Associates Publishers, pp 77–88

Contractor N, DeChurch L, Carson J, Carter D, Keegan B (2012) The topology of collective leadership. Leadersh Quart 23(6):994–1011

Cross R, Parker A (2004) The hidden power of social networks: Understanding how work really gets done in organizations. Harvard Business Press, Boston

Cullen-Lester K, Maupin C, Carter D (2017) Incorporating social networks into leadership development: a conceptual model and evaluation of research and practice. Leadersh Quart 28(1):130–152

De Freitas S, Griffiths M (2007) Online gaming as an educational tool in learning and training. Bt J Educ Technol 38(3):535–537

D'Innocenzo L, Mathieu J, Kukenberger M (2016) A meta-analysis of different forms of shared leadership-team performance relations. J Manag 42(7):1964–1991

Drescher M, Korsgaard A, Welpe I, Picot A, Wigand R (2014) The dynamics of shared leadership: building trust and enhancing performance. J Appl Psychol 99(5):771

Freeman L (1978) Centrality in social networks conceptual clarification. Soc Netw 1(3):215–239

Gilson L, Maynard T, Jones Young N, Vartiainen M, Hakonen M (2015) Virtual teams research: 10 years, 10 themes, and 10 opportunities. J Manag 41(5):1313–1337

Gloor P, Paasivaara M, Schoder D, Willems P (2008) Finding collaborative innovation networks through correlating performance with social network structure. Int J Prod Res 46(5):1357–1371

Huang Y, Zhu M, Wang J, Pathak N, Shen C, Keegan B, Williams D, Contractor N (2009) The formation of task-oriented groups: exploring combat activities in online games. In: 2009 international conference on computational science and engineering, vol 4, pp 122–127. IEEE

Jokisaari, M (2016) A Social network approach to examining leadership. In: Handbook of methods in leadership research, chap. 11. Edward Elgar Publishing, pp 1–56

Korsgaard A, Picot A, Wigand R, Welpe I, Assmann J (2010) Cooperation, coordination, and trust in virtual teams: insights from virtual games. In: Online worlds: convergence of the real and the virtual. Springer, pp 253–264

Krackhardt D, Hanson J (2003) Informal networks: the company behind the chart. In: Cross R, Parker A, Sasson L (eds) Networks in the knowledge economy. Oxford University Press, New York, pp 235–247

Leonardi P, Contractor N (2018) Better people analytics. Harvard Business Review

Lipnack J, Stamps J (1999) Virtual teams: the new way to work. Strategy Leaders 27(1):14–19

Moreno J (1946) The sociometric view of the community. J Educ Sociol 19(9):540–545

Müller S, Ghawi R, Pfeffer J (2020) Using communication networks to predict team performance in massively multiplayer online games. In: IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM). The Hague, Netherlands (virtual)

Nicolaides V, LaPort K, Chen T, Tomassetti A, Weis E, Zaccaro S, Cortina J (2014) The shared leadership of teams: a meta-analysis of proximal, distal, and moderating relationships. Leadersh Quart 25(5):923–942

Pearce C (2011) Communities of play: emergent cultures in multiplayer games and virtual worlds. MIT Press, Cambridge

Pobiedina N, Neidhardt J, Moreno C, del Carmen M, Werthner H (2013) Ranking factors of team success. In: International World Wide Web Conference. ACM, Rio de Janeiro, Brazil, pp 1185–1194

Reeves B, Malone T, O'Driscoll T (2008) Leadership's online labs. Harvard Bus Rev 86(5):58–66

Seidman S (1983) Network structure and minimum degree. Soc Netw 5(3):269–287

Shim KJ, Damania S, Delong C, Srivastava J (2011) Player and team performance in EverQuest II and Halo 3. IEEE Potentials 30(5):21–26

Shim KJ, Srivastava J (2010) Team performance prediction in massively multiplayer online role-playing games (mmorpgs). In: 2010 IEEE second international conference on social computing. IEEE, pp 128–136

Sourmelis T, Ioannou A, Zaphiris P (2017) Massively multiplayer online role playing games (mmorpgs) and the 21st century skills: a comprehensive research review from 2010 to 2016. Comput Hum Behav 67:41–48

Suznjevic, M., Stupar, I., Matijasevic, M.: Mmorpg player behavior model based on player action categories. In: 2011 10th annual workshop on network and systems support for games. IEEE, pp 1–6 (2011)

Tawa J, Negrón R, Pfeffer J (2019) The measurement of interpersonal interactions with continuous spatiotemporal data: application to a study of the effects of resource competition on racial group interactions. Behav Res Methods 52:881–900

Theil H (1961) Economic Forecasts and Policy. Contributions to economic analysis. North-Holland Pub, Co

Tuckman B, Jensen MA (1977) Stages of small-group development revisited. Group Org Stud 2(4):419–427

Wasserman S, Faust K (1994) Social network analysis: methods and applications. Cambridge University Press, Cambridge

Wigand R, Agarwal N, Osesina I, Hering W, Korsgaard A, Picot A, Drescher M (2012) Social network indices as performance predictors in a virtual organization. In: 2012 Fourth international conference on computational aspects of social networks (CASoN). IEEE, Sao Carlos, Brazil, pp 144–149

Williams D, Contractor N, Poole MS, Srivastava J, Cai D (2011) The virtual worlds exploratorium: using large-scale data and computational techniques for communication research. Commun Methods Meas 5(2):163–180