How to Predict Social Relationships — Physics–inspired Approach to Link Prediction

Akanda Wahid -Ul- Ashraf, Marcin Budka

Department of Computing and Informatics Bournemouth University Fern Barrow, Poole BH12 5BB, UK, e-mail: {aashraf, mbudka}@bournemouth.ac.uk

Katarzyna Musial

Advanced Analytics Institute, School of Software, Faculty of Engineering and IT, University of Technology Sydney, Australia, e-mail: katarzyna.musial-gabrys@uts.edu.au

---

Abstract

Link prediction in social networks has a long history in complex network research area. The formation of links in networks has been approached by scientists from different backgrounds, ranging from physics to computer science. To predict the formation of new links, we consider measures which originate from network science and use them in the place of mass and distance within the formalism of Newton's Gravitational Law. The attraction force calculated in this way is treated as a proxy for the likelihood of link formation. In particular, we use three different measures of vertex centrality as mass, and 13 dissimilarity measures including shortest path and inverse Katz score in place of distance, leading to over 50 combinations that we evaluate empirically. Combining these through gravitational law allows us to couple popularity with similarity, two important characteristics for link prediction in social networks. Performance of our predictors is evaluated using Area Under the Precision-Recall Curve (AUC) for seven different real-world network datasets. The experiments demonstrate that this approach tends to outperform the setting in which vertex similarity measures like Katz are used on their own. Our approach also gives us the opportunity to combine network's global and local properties for predicting future or missing links. Our study shows that the use of the physical law which combines node importance with measures quantifying how distant the nodes are, is a promising research direction in social link prediction.

Keywords: social network, link prediction, network dynamics, physics-inspired network predictive model, Newton Gravitational Law

---

1. Introduction

Networks are ubiquitous. Ranging from food webs, to protein, brain or social networks, they underpin many natural phenomena [1, 2, 3, 4]. In the broad landscape of network science, networks which are formed via social interactions, have been increasingly drawing research attention in recent years, due to the heterogeneity of their components and non-trivial dynamics. Data representing small-scale social networks were available and analysed in the past, for example, the famous Zachary's karate club network has been studied extensively since it was published by Zachary [5] in 1977. However, Zachary's karate club contains only 34 nodes and 78 vertices, whereas today's social networks (e.g. Facebook, scientific paper citation, Twitter), contain billions of nodes and are far more complex and dynamic [6]. Although these large-scale social networks are formed by social interactions, their topological properties and dynamics are similar to those of networks found in nature. For example, most biological networks exhibit power-law degree distribution, cellular networks have high clustering coefficient, network encoding the large-scale causal structure of spacetime in our accelerating universe exhibits power-law degree distribution and high clustering coefficient [7, 4]. Both of these characteristics are also commonly found in social networks.

The similarity between anthropogenic social networks and naturogenic networks gives the opportunity to apply many different prediction and modelling tools developed in the field of naturogenic networks, to social networks. This is due to the fact that large-scale physical and biological networks and social networks exhibit similar topological properties (e.g. degree power-law distribution, high clustering coefficient) [3, 8, 4]. However, the similarities are explored at the global level and this causes some issues with precision of adopted models and methods because the local dynamics are not considered. This raises the question if we could also adopt laws which govern a physical system to predict social network at a local level.

Tools which are primarily used in order to analyse, model, or describe physical world have been used in social network analysis on numerous occasions [9]. Some examples include Memetic algorithm for community detection in social networks, reaching of Bose gas state of complex social networks or the molecular model of social network [10, 4, 11, 12]. The field with applications of physical models to social networks has been named as social physics by Urry [13].

The main focus of this paper is the link prediction problem. The proposed model is inspired by the earliest theory of gravity, where Newton described the law of universal gravitation based on the force between two point masses. Authors have already attempted to use models from physics in the context of network structure prediction. In Budka et al. [14] and Juszczyszyn et al. [12] they adopted molecular models in the context of evolution of social network. Now, by applying Newton's gravitational law, we extend the nature-inspired link prediction framework with a new method that allows to take into account more than one characteristic of the network, and not only distance between nodes as it was done in the molecular model.

The rest of the paper is structured as follows. Section 2 presents the problem statement and related work. The proposed method is described in Section 3 and the experimental setup in Section 4. Section 5 discusses the results, while the final conclusions are given in Section 6.

2. Related Work

Given a network at time t, the link prediction problem is to identify new links that will be present in the network at time $t+1$ [15, 16]. Assuming the network has a set V of nodes and set E of edges at time t expressed as $G(V, E_t)$, and that a link between a pair of vertices v_i and v_j is denoted by $L(v_i, v_j)$, the goal of link prediction is to predict whether $L(v_i, v_j) \in E_{t+1}$, where $L(v_i, v_j) \notin E_t$. The prediction is performed by using topological and/or non-topological information about nodes' characteristics and their relationships.

2.1. Link prediction methods classifications

There are numerous works on review and classification of link prediction methods [17, 18, 19, 20, 21, 22]. One of the widely used and accepted classifications is by Liben-Nowell and Kleinberg [18], where link prediction methods were grouped as:

1. Methods based on node neighbourhoods (e.g. Common Neighbours [23], Jaccards Coefficient [24], AdamicAdar [25], Preferential Attachment [26])
2. Methods based on the ensemble of paths between a pair of nodes (e.g. Katz [27], Hitting time [18], PageRank [28])
3. Higher-level approaches (Low-rank approximation [29, 18], unseen biagrams [30, 31, 18], clustering [18])

Classifications, like the one above, give us a better understanding of the principles that are used when link prediction methods are proposed, e.g. if a method works at a local or global level of the network or use path or node based similarity, etc. However, they neglect the information about applicability of different methods, i.e. those classifications do not answer a question in what circumstances and for what networks the methods can be used. For example, for some methods (e.g. Katz) an input is a single snapshot of a network, while others (e.g. Triad Transition Matrix (TTM)) require a time series as an input (i.e. a sequence of historical snapshots of the network) [32, 33]. As a result, methods like TTM are not applicable to network datasets where only vertices and links are given without historical information [32]. Also, there are other methods which may use additional information about node attributes like age, location, etc. [34, 35]. Based on the type of information exploited by link prediction methods, we categorise link prediction methods into four groups:

1. **Unsupervised – based on topological information**, which are methods that only use structural information such as mutual friend count in social networks, path lengths, triad profiles etc. Some examples include methods like Katz, PageRank, and AdamicAdar [18]. These methods only require a snapshot of the network topology at any given time t to make predictions for time $t + 1$, and are useful when past and non-topological information is not available. These methods are applicable to any type of network dataset and do not require training.
2. **Supervised – based on topological information**, which are methods only applicable to networks where historical information regarding network's topology is available. For example, if snapshots of a network at $t-1$ and t are given, then $t-1$ is considered as historical information. Network characteristics like degree of certain nodes at time $t-1$ can also be considered as historical information. One example of such method is the Triad Transition Matrix (TTM) [32, 33]. A wide range of machine learning approaches also fall into this category if the topological information such as mutual nodes, shortest distance etc. is considered as features, and link appearance is considered as class label [34, 36, 37]. Methods in this category do not use non-topological information such as age, location etc.

3. **Unsupervised – based on non-topological and/or topological information**, which are methods that consider non-structural information like age, location, preferences etc. [38, 34, 35]. In this category topological information can also be used in combination with the non-structural attributes mentioned above.
4. **Supervised – based on non-topological and/or topological information**, which are methods applicable to the same kind of datasets as in point two above. If non-structural historical information of a network is considered (with or without topological information) any binary classifier could be used to make predictions in this setting [39].

There are multiple methods that fall into the first category [17, 18, 19, 20, 21, 22]. These methods are applicable to any kind of network where only one structural snapshot is available. Despite the fact that the methods only exploit network topology without historical information or node attributes, they make more accurate predictions for future links than a random predictor [18]. The proposed link prediction method in its current form falls into the first category. However, a supervised version or usage of non-topological information is also possible and is discussed in Section 3.

2.2. Physics-inspired approaches for link prediction in social networks

If we consider a social network, at its local level, how two people make a connection or interact could rely on two factors, 1) how popular, and 2) how similar these people are. These two concepts are known as popularity and similarity and are well established in the link prediction paradigm [40, 41]. Intuitively, for social networks, predicting the appearance of links between two people, having both the popularity and similarity factors should entail better prediction accuracy than considering only one of the factors (i.e. only popularity or only similarity). In social network analysis, we already have a wide range of measures of node popularity and similarity. Different centrality measures (e.g. degree centrality, closeness centrality or betweenness centrality) could be thought of as notions of popularity. On the other hand, scores from link prediction methods like Katz, AdamicAdar could be thought of as measurements of nodes' similarity [42, 27, 25]. However, the challenge is how to combine these two metrics in order to predict links between two particular entities in the future. This is where we make use of Newton's law of gravity. In Newton's explanation of gravity, the force between two particles is proportional to the product of their masses and inversely proportional to the squared distance between them. We argue that this law of attraction between two points of masses could also be applicable in social networks. We measure popularity or importance of a node using centrality and consider it as mass. We measure dissimilarity by the inverse of similarity (i.e. scores from link prediction methods like Katz, AdamicAdar etc.) or by the path length, and consider them as distance.

Physics-inspired approaches in networked systems have been used in the context of force-directed graph drawing, where node centralities were used as masses [43]. However, as opposed to our method, Bannister et al. [43] did not use a measurement of distance or Newton's gravitational equation for predicting future interactions. One of the first applications of gravity in social science dates back to as early as the mid-19th century, when Simini et al. [44] and Carey [45] reasoned that physical laws are also applicable in social phenomena [46]. There are also some approaches using the theory of gravity to solve link prediction problem, however, most of these works are related to modern physics i.e. quantum mechanics [13, 14, 12, 11].

In the study by Levy and Goldenberg [47], Newton's gravitational law is used in link prediction. The authors used spatial distance (i.e. not topological) and substituted friendliness for masses. In fact, inverse square law in terms of geographical distance has been used earlier than in [47]. Not specifically in link prediction but in the field of social gravity, Zipf [48] and Stewart [49] both have applied the inverse square law. In fact, they have considered the original notion of Newtonian gravitational law where the interaction between two groups of people is proportional to their cardinality, and inversely proportional to their squared geographical distance [46, 48, 49]. The problem with this approach in online social networks is that in some cases the physical distance is either not available or not indicative of the relationship strength. Therefore, in this study we take the inverse of different similarity measurements from scores of Katz, AdamicAdar, and Rooted PageRank (RPR) as distance, and use centrality as mass.

3. Proposed method

Our approach to link prediction in social networks is inspired by Newton's law of universal gravitation, which states that the force exerted between two masses is proportional to the product of those masses, and inversely proportional to the squared distance between their centres [50]:

$$F = G\frac{m_1 \cdot m_2}{r^2}, \tag{1}$$

where F is the force between masses m_1 and m_2, G is the gravitational constant, and r is the distance between m_1 and m_2. Newton derived this equation by empirical observation and inductive reasoning [51], which is an approach that we have also taken.

As discussed earlier, we use importance or popularity of a node to express mass. We argue that different centrality measures are direct measurements of how important, central or popular a node is in a given network. Dissimilarity or distance is measured via path distances (e.g. shortest path) or inverse of various similarity measures (e.g. AdamicAdar, Jaccard's Coefficient). It is also possible to define distance in terms of dissimilarity in non-topological node properties, like age, physical distance etc. A weighted sum of these factors can be incorporated into the distance, allowing to naturally exploit non-topological information. This however is not the focus of our study.

The above analogy leads to the following formula for calculating the score of two nodes forming a link in the future:

$$\boldsymbol{Score(v_i, v_j)} = \boldsymbol{Score(v_i, v_j)} \propto \frac{P(v_i) \cdot P(v_j)}{D(v_i, v_j)^2}, \tag{2}$$

where P is popularity/centrality and D is dissimilarity/distance in an undirected graph.

The formula in Equation 2 can be interpreted as a modification of the Preferential Attachment method (i.e. product of centralities), where the resultant scores are weighted by the inverse of squared distance between the two nodes in question. This arguably gives our method more expressive power by taking proximity into account, which as demonstrated in our previous work [52] not only makes sense intuitively, but also tends to produce more accurate predictions in practice.

As for the gravitational constant G, without loss of generality we have assumed $G = 1$, since in order to make a prediction, a ranked list of scores is required with their absolute values being irrelevant. Note, that if the score was to be interpreted as probability, for a given network this could be achieved by setting the value of G as:

$$G = \frac{\min \forall_{(i,j),i \neq j} D(v_i, v_j)^2}{\max \forall_i P(v_i) \cdot \max \forall_{j \neq i} P(v_j)}, \tag{3}$$

where the numerator is equal to 1, which reflects the obvious existence of a direct link between at least one pair of nodes. This essentially scales $Score(v_i, v_j)$ to be between 0 and 1. Two closest nodes (path length 1 if they are connected) with highest degrees in the entire graph will result in a score $Score(v_i, v_j) = 1$. Including the above constant value of G in Equation 2, effectively divides every score by the highest possible score for a given graph or network[1].

Different link prediction methods give different similarity scores that denote how likely two nodes are to be connected in the future. In our method we use the inverse of these scores to denote the dissimilarity/distance[2], plugging them into Equation 2.

4. Experimental Setup

In order to empirically evaluate our approach proposed in Equation 2 we use three different centrality measures along with 12 similarity measures. Definitions of both centrality and similarity measures are outlined below.

4.1. Centralities

In our experiments we use the degree, closeness and betweenness centrality, considered as a measurement of popularity in Equation 2. We draw an analogy here between these three centrality measures and mass in Equation 1:

1. **Degree Centrality (DC),** which is the degree of a vertex in a network i.e. the number of edges attached to this vertex (the number relationships a person has in a social network). This is a very simple but useful measure of centrality in social networks that indicates importance of the node within the overall structure [53].
2. **Closeness Centrality (CC),** which is calculated based on the mean geodesic path from a given vertex to all other vertices in the network [53]. High closeness centrality of a vertex means the vertex has better access to information or more direct influence on other vertices. Closeness centrality is defined as:

$$CC(v_i) = \frac{1}{\sum_{n \neq i} d(v_i, v_n)} \tag{4}$$

[1]Much like physical world, one may also estimate G from a given graph to determine the proportionality constant rather than using it to scale the score between 0 to 1.

[2]We are considering dissimilarity as distance, noting that in some cases the symmetry and triangle inequality might not hold. For an unweighted and undirected graph $Score(v_i, v_j) = Score(v_i, v_j)$ (symmetry) but other than shortest path, triangle inequality may or may not hold for every dissimilarity score.

In Equation 4, d is the geodesic distance between two vertices. If there are a total $n+1$ vertices in a graph, closeness centrality for vertex v_i is calculated using the inverse of the average length of the shortest path from/to all other vertices except itself $v_i \notin \{v_1, v_2, ..., v_n\}$. If the path does not exist between two vertices then the total number of vertices is used instead of path length [54].

3. **Betweenness Centrality (BC),** which gives score to a vertex v_i based on how many paths connecting any two vertices in the network go through that vertex v_i. If the number of those paths is high then vertex v_i will have high betweenness centrality. Vertices that are frequently on the shortest paths between any two vertices of the graph have more control over information flow [42, 55]. Removing a vertex with high betweenness centrality has negative influence on the overall information flow in a network. Betweenness centrality differs from other centrality measures as it doesn't consider how well-connected a vertex is but measures how much a vertex falls in between others. This way it is possible to have a vertex with low degree but high betweenness centrality. For example, two groups of vertices can be connected via a single path and then a vertex that connects those groups (a.k.a. bridge node or broker) will have high betweenness centrality.
If a network has set of vertices V, source vertex $s \in V$ and target vertex $t \in V$, the betweenness centrality of vertex v_i can be defined as [42, 55, 56]:

$$BC(v_i) = \sum_{s \neq v_i \neq t} \frac{\sigma_{st}(v_i)}{\sigma_{st}} \tag{5}$$

where σ_{st} is number of shortest paths between two vertices s and t and $\sigma_{st}(v_i)$ is the number of shortest paths between two vertices s and t that pass through v_i.

4.2. Similarity

We have used 12 similarity measurements to calculate node similarity and use their inverse value as a measurement of distance/dissimilarity for Equation 2. The similarity measurements we have used are described below.

1. **Common Neighbours (CN),** which is a similarity metric where the likelihood of two nodes v_i and v_j to develop a link depends on the number of mutual friends [23]. This method could be quantified via Equation 6 (Γ represents the set of neighbours of a node):

$$Score(v_i, v_j) = |\Gamma(v_i) \cap \Gamma(v_j)|, \tag{6}$$

2. **Jaccard's Coefficient (JC),** which is a version of Common Neighbours [24] normalised by the total number of neighbours of both nodes:

$$Score(v_i, v_j) = \frac{|\Gamma(v_i) \cap \Gamma(v_j)|}{|\Gamma(v_i) \cup \Gamma(v_j)|} \tag{7}$$

3. **AdamicAdar (AA),** which is a similarity metric used in information retrieval [18] similar to the Jaccards Coefficient (JC). In this method the likelihood of two

nodes being connected in the future depends on the number of Common Neighbours (e.g. mutual friends in a social network) relative to the nodes' degrees [25]:

$$Score(v_i, v_j) = \sum_{v_k \in \Gamma(v_i) \cap \Gamma(v_j)} \frac{1}{\log |\Gamma(v_k)|} \tag{8}$$

4. **Preferential Attachment (PA),** which is based on the social concept of 'rich get richer' implying that nodes with higher degree are more likely to get new links [26]:

$$Score(v_i, v_j) = |\Gamma(v_i) \cdot \Gamma(v_j)| \tag{9}$$

5. **Katz,** which considers the number of all the paths from node v_i to v_j [27]. The shorter paths have bigger weight (i.e. are more important), which is damped exponentially by path length and the β parameter (M is the adjacency matrix):

$$Score(v_i, v_j) = \beta M + \beta^2 M^2 + \beta^3 M^3 + \cdots \tag{10}$$

   β needs to be smaller than the reciprocal of the highest eigenvalue of M [57]. In our experiments we have used three different values of β. For *collegeMsg*, *contact*, *hep-th*, *hep-ph*, and *hypertext* datasets $\beta \in \{0.001, 0.0005, 0.00005\}$; for *infectiousContact* dataset $\beta \in \{0.005, 0.0005, 0.00005\}$; for *MITContact* $\beta \in \{0.1, 0.05, 0.005\}$ have been used. In Section 5 three different values of β parameter are denoted as Katz1, Katz2, and Katz3.

6. **Rooted PageRank (RPR),** which is used by search engines to rank websites. In graph analysis it works by ranking nodes, with the rank being determined by the probability of each node being reached via random walk on the graph [28]. The $Score(v_i, v_j)$ is calculated using the stationary probability distribution of B in a random walk. The random walk returns to v_i with the probability α at each step, moving to a random neighbour with probability $1 - \alpha$. We have calculated RPR for every dataset using two different α parameters and they are $\alpha \in \{0.15, 0.25\}$.

7. **Average Commute Time (ACT),** which is an average number of steps it takes to visit node v_j from node v_i and come back to v_j using random walk [19]:

$$Score(v_i, v_j) = RandWalk(v_i, v_j) + RandWalk(v_i, v_j) \tag{11}$$

   This could be obtained using pseudoinverse of the laplacian matrix (L), which is L^+, where $L = B - M$ [58, 59, 60]. Here, B is the degree matrix (a diagonal matrix which contains degree of every vertices) and M is the adjacency matrix.

$$Score(v_i, v_j) = \frac{1}{C(l^+_{ii} + l^+_{jj} - 2l^+_{ij})} \tag{12}$$

   In Equation 12, because we are considering the rank, constant C could be removed. Here l^+ are the elements in matrix L^+.

8. **Average Commute Time Normalised (ACTN),** which is the same as ACT but normalised by stationary distribution, $\pi = \frac{B}{\sum_v B}$ [61, 62].

9. **Pseudoinverse of the Laplacian matrix (PsInLap),** which is simply the pseudoinverse of the graph Laplacian L^+. PsInLap defines kernel of a graph and can be interpreted as a similarity measure [59].

10. **Local Path Index (LPI),** which is based on the number of paths of different lengths between two vertices. LPI is a generalisation of CN. While CN measures similarity based on mutual friend count, which effectively gives the number of paths with length two between two vertices, LPI also takes into account paths of length three [63, 64]. LPI is hence a more global similarity measure than CN but not as global as Katz:

$$Score(v_i, v_j) = M^2 + \epsilon M^3 \tag{13}$$

In Equation 13, ϵ is a free parameter. If we choose it to be zero then this would give us common neighbours, and if we consider all higher orders of M (the adjacency matrix) than this would essentially become Katz. In our experiments we have used two values for $\epsilon \in \{0.01, 0.02\}$.

11. **Leicht-Holme-Newman Global Index (LGI),** which is a similarity metric utilising the concept that if two nodes v_i and v_j have neighbours who are themselves similar, then they have higher similarity score [65]:

$$Score(v_i, v_j) = B^{-1}\left(I - \frac{\theta}{\lambda}M\right)^{-1}B^{-1} \tag{14}$$

In Equation 14, θ is a free parameter and λ is a the largest eigenvalue of adjacency matrix M. We have used $\theta \in \{0.5, 0.7\}$ in our setup.

12. **Matrix Forest Index (MFI),** which is a similarity score between v_i and v_j, defined as ratio of the number of spanning rooted forests, such that vertices v_i and v_j belong to the same tree which is rooted at v_i to all spanning rooted forests of the entire network [66]:

$$Score(v_i, v_j) = (I + L)^{-1} \tag{15}$$

A spanning subgraph of a graph contains the same vertices as the main graph, but not all the edges. A forest is a cycleless graph and a tree is a connected forest. A rooted tree is a tree which has only one root [66].

Reciprocal values of the similarity measures presented above (except Preferential Attachment) can be seen as inverse of different topological path measurements, hence we consider them as distance in Equation 1. Preferential Attachment (PA) is scored via the product of popularity (degree), which is the special case of numerator of proposed Equation 2 without the denominator of squared dissimilarity.

4.3. Datasets

For the experimental comparative evaluation of the proposed method we selected seven datasets from various domains and of different sizes, frequently used in the literature, all representing undirected graphs:

1. **hep-th:** Collaboration graph of authors of scientific papers from High Energy Physics – Theory (hep-th) Section, where edges between two nodes represent a common publication. This dataset is acquired from the KONECT database [67, 68, 69] and has been used in the experiment of Liben-Nowell, which is a very important research work in the area of link prediction [18].
2. **hep-ph:** Collaboration graph of authors of scientific papers from High Energy Physics – Phenomenology (hep-ph) Section, where edges between two nodes represent a common publication. This dataset is acquired from the KONECT database [70, 71].
3. **contact:** Dataset representing a network where edges are human contacts using portable wireless devices distributed among different groups of people [72, 73].
4. **hypertext:** Face-to-face contacts of ACM Hypertext 2009 conference attendees, where edges represent interactions of at least 20 seconds [74, 75].
5. **collegeMsg:** Private messages sent via an online social network at the University of California, Irvine for over 193 days [76].
6. **infectiousContact:** This dataset represents network of the face-to-face interactions of people during an exhibition INFECTIOUS: STAY AWAY in 2009 at the Science Gallery in Dublin. Each node is a person and edges between two nodes represent face-to-face contacts that lasted at least for 20 seconds. This network contains data about the interactions gathered on the day of the exhibition when highest number of contacts took place. This dataset is also acquired from KONECT database [77, 74]
7. **MITContact:** This dataset is based on human contact and it is a part of Reality Mining experiment preformed in 2004. In this network, vertices represent physical contact between a group of students from Massachusetts Institute of Technology (MIT) [78, 79]. This dataset is also acquired from KONECT. Data has been collected over a period of nine months.

As it can be seen from Table 1, the selected datasets differ greatly in size and most of them represent typical social networks with power law node degree distribution, normal distribution of shortest path and small mean shortest path length as well as high global clustering coefficient. There are of course some exceptions to this profile, e.g. *collegeMsg* has very low global clustering coefficient, making the network more similar to random rather than social network. For a fully connected graph the highest density of a network is one. However, for networks with multiple edges, density can be higher than one, as multiple links between two vertices are possible. We can observe this higher than one density for, *hypertext* and *MITContact* contact datasets. The density is higher than one for both the datasets and both of these networks have multiple edges. However, in the training portion (i.e. the part of the data which is used for making the prediction as discussed in more details in Section 4.4) of those two networks we still have many nodes where no edges exist. In Section 5 we make predictions for these missing edges or links.

dataset	no. vertices	no. edges	density	node degree dist.	avg. shortest path dist.	avg. shortest path	transitivity dist.	global clustering coeff.
collegeMsg	1899	59835	0.033	power law	normal	3.055	power law	0.057
contact	274	28244	0.755	power law	normal	2.424	power law	0.566
hep-th	6776	290484	0.013	power law	normal	3.224	normal	0.333
hep-ph	10324	955423	0.018	power law	normal	2.946	normal	0.351
hypertext	113	20818	3.290	power law	normal	1.656	power law	0.495
infectiousContact	410	17298	0.206	power law	normal	3.631	power law	0.436
MITContact	96	1086405	238.247	power law	normal	1.445	power law	0.725

Table 1: Basic statistics of the datasets selected for the experiment

4.4. Data Partition

All networks considered in this study are with timestamps that indicate when a given relationship was created. This allows us to test prediction results against actual links that appeared in the future. We have divided each of the datasets into two parts based on the timestamps available. A similar setup has been used by Liben-Nowell and Kleinberg [18] for benchmarking several link prediction methods, and in particular:

1. The **hep-th** dataset has been divided into two parts. Part one consisted of publications from years 1992-1994 and part two consisted of publications from years 1995-1997. Part one is where the link prediction is performed and part two is used as a ground truth in order to evaluate the method.
2. The **hep-ph** is also divided into two parts, part one containing publications between year 1994 and 1996, and part two with publications between year 1997 and year 1999. Similar to the previous dataset, part one is where the link prediction is performed and part two is used as ground truth.
3. Datasets **contact, hypertext, collegeMsg, infectiousContact, and MITContact** have also been divided into two parts with respect to time. However, the timespans within each part are not equal. Each part contains approximately[3] equal number of edges.

5. Results

We are using Area Under the Precision-Recall Curve (AUC) to evaluate performance of each of the predictors. In total, we have calculated AUC for combinations of 74 different predictors and seven datasets. These 74 predictors involve 1) similarity measures from Section 4.2, 2) combinations of these similarity measures with centrality measures from Section 4.1 and, 3) combinations of shortest path with the centrality measures mentioned above.

The summary of results is given in Figures 1 and 2. In Figure 1 AUC values are sorted in descending order. Each of the bars is the sum of all the AUCs over all datasets for a given approach (i.e. a given predictor from the three categories listed above) to link prediction. For example, the leftmost bar in Figure 1 represents AUCs for combination of closeness centrality and MFI using Equation 2. This predictor has the

[3]For *collegeMsg* and *MITContact* datasets total number of edges are odd

best overall performance if we sum AUCs for this method for all seven datasets. On the other hand, Figure 2 depicts individual performance for all the predictors for individual datasets. From Figures 1 and 2 we see that for some of the datasets, overall AUCs are very small. However, later in Sections 5.1.1–5.1.12 we have compared each of methods with a random predictor. The results show that overall low values of AUC for a certain dataset do not necessarily mean that particular dataset has low predictability. This is because all networks are different in size. For a larger (in terms of vertices) or less dense network, the total number of predictions made is higher. This is because, we make predictions for a total of $\frac{|V|(|V|-1)}{2} - |E|$ links. As a network gets denser, the term $|E|$ also becomes larger. As a result, the total number of predictions gets lower. Because our AUC is from Precision-Recall curve, when we make predictions for a higher number of links there is a higher chance of having more false positives. This is because of the number of new links that a network forms may not increase at the same rate as the growth of the network. The Precision is calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

From Equation 16, we could see that, if we have larger values for false positives (FP) the value for Precision gets lower.

In Figure 1 it can be seen that the first three overall best performing methods are the ones with our Newton's gravitational law inspired combination approach. On the other hand, ACTN used as a standalone method makes worst prediction among all the 74 predictors. Interestingly, when ACTN is combined with DC using Equation 2 its performance jumps to rank 32 from 74. In addition, this combination of ACTN with DC performs better than DC with shortest path. This improvement reveals that the increment in predictability is not because of DC, or ACTN's independent predictability but because of the combination that we use. More on this improvement due to the combination is discussed later in Section 5.1. We also see a similar improvement with CN, where CN combined with CC ranks as the fourth overall best method. Improvements due to the combination approach we take could also be seen in several other combinations of predictors with MFI, Katz, RPR, etc. These improvements evidence that our combination approach has a great potential in the area of link prediction.

We further analyse the results in two ways: (i) we group methods based on the similarity measure used and then we compare the results within the groups (Sections 5.1.1–5.1.12) and (ii) we discuss the results in the context of each dataset separately and try to interpret why certain methods work on some datasets and not on others (Section 5.3).

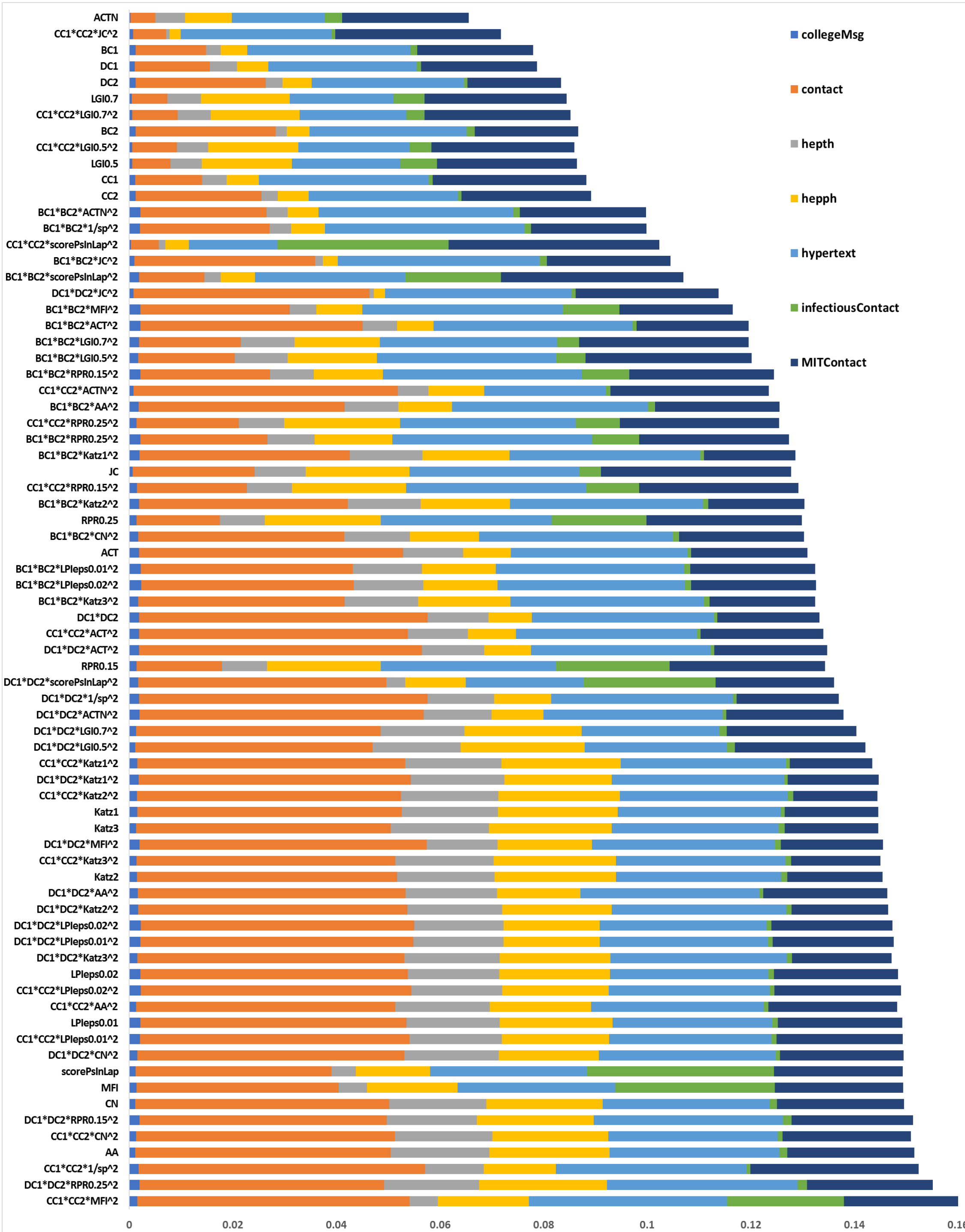

Figure 1: Combined Average (AUC)

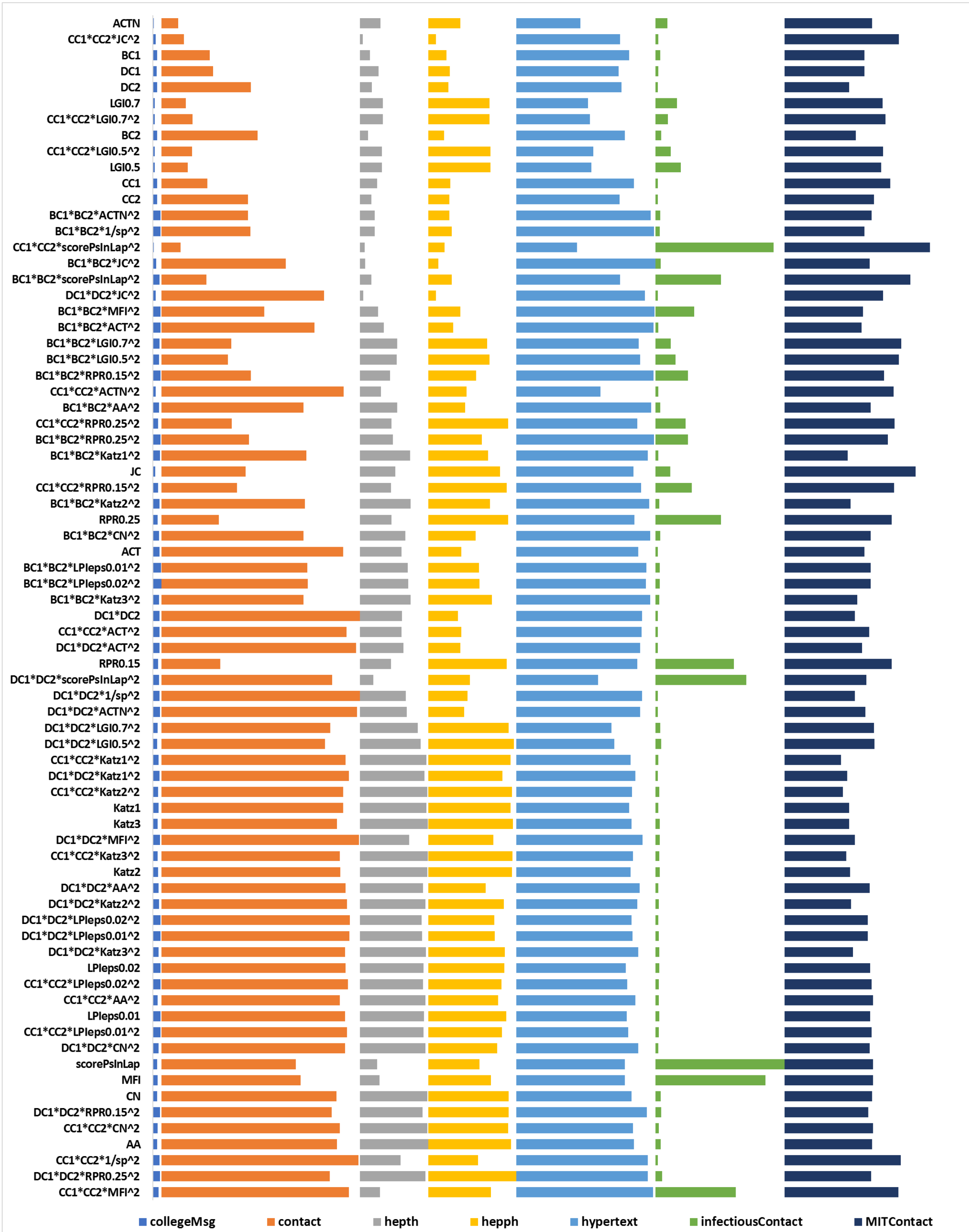

Figure 2: Individual Method's Performance (AUC)

5.1. Overall performance using AUC

For any pair of vertices v_i and v_j, we can consider all the similarity methods from Section 4.2 as a set of predictors $S = \{Katz1, Katz2, Katz3, AA, .., CN\}$. Similarly, all centrality measures from Section 4.1, could be expressed as a set $P = \{DC, BC, CC\}$ where $DC = DC_i \cdot DC_j, BC = BC_i \cdot BC_j, CC = CC_i \cdot CC_j$. As we use dissimilarity or distance by taking the inverse of each similarity measure for Equation 2, our proposed combination approach could be expressed as:

$$W = \{P \times S\}, \tag{17}$$

where each of the elements $w \in W$ is a particular predictor which gives prediction for any two vertices v_i and v_j. For any predictor $w \in W$, it is a combination of one particular similarity measure $s \in S$ and one particular centrality measure $p \in P$. For such a combined predictor w, with similarity measure s and centrality p we check if:

$$AUC(w) > \big(AUC(s) \wedge AUC(w)\big) > AUC\Big(\frac{p}{d^2}\Big) \tag{18}$$

Here in Inequality 18, d is the shortest path. If for a particular combination approach w, Inequality 18 holds, those AUC values are highlighted using dark grey boxes in Tables 2–12. The dark grey boxes indicate if a particular well-established similarity measure $s \in S$, when combined with centralities using Equation 2 performs better than the similarity measure on its own. The improvement could also be due the product of centralities in p which we have in the combination method w. In fact, product of degree centrality of v_i and v_j is a similarity measure, Preferential Attachment (PA) from Section 4.2. Similarly, it is possible to use a product of another centrality measure as a standalone predictor. Due to this we also check if AUC of a particular combination $w \in W$ is greater than the AUC of $\frac{p}{d^2}$. The denominator of d^2 results from findings of our earlier study [52], where dividing by shortest path squared mostly improves (where it does not, the difference is very small) the score as compared with the standalone product of centralities. The analysis in Table 5.1.12 confirms this improvement. As a result, if Inequality 18 holds, the inverse of similarity measure improves the predictor when used for Equation 2. It also shows that the improvement is due to the combination approach we take using Equation 2 but not due to the independent predictability of the similarity measure or product of centralities divided by squared shortest path. In Sections 5.1.1–5.1.11, when the performance of a combination method is said to be better or improved, it entails that Inequality 18 holds.

In addition to validating Inequality 18, for each of the datasets, we also identify if AUC of a predictor is smaller than the AUC of a random predictor. For each predictor, AUC is calculated using R package called PRROC [80, 81]. The AUC of a random predictor is also generated from the same package. For each dataset AUC of a random predictor is calculated from an ensemble of 1000 random predictors [80]. In Tables 2–13, values of AUC which are not higher than the AUC of a random predictor for a particular dataset, have been highlighted as light grey.

5.1.1. Combinations with Katz

	college Msg	rnk	contact	rnk	hep-th	rnk	hep-ph	rnk	hyper text	rnk	infectious Contact	rnk	MIT Contact	rnk
$Katz1$	0.01132	39	0.35702	22	0.13032	8	0.16138	8	0.22064	52	0.00532	60	0.12643	68
$Katz2$	0.01061	43	0.35138	25	0.13167	5	0.16412	5	0.22377	49	0.00815	38	0.12842	66
$Katz3$	0.00969	50	0.34395	30	0.13258	2	0.16578	3	0.22576	48	0.00826	37	0.1265	67
$DC1*DC2*Katz1^2$	0.01286	29	0.36789	9	0.12653	15	0.14505	22	0.23282	37	0.00499	66	0.12262	71
$DC1*DC2*Katz2^2$	0.01229	35	0.36401	13	0.12775	12	0.1479	21	0.23663	35	0.00673	48	0.13064	64
$DC1*DC2*Katz3^2$	0.01121	40	0.36078	18	0.12869	10	0.14986	19	0.23854	32	0.00703	46	0.13417	63
$BC1*BC2*Katz1^2$	0.01405	15	0.28444	41	0.09813	28	0.11762	40	0.25738	15	0.00534	59	0.12364	70
$BC1*BC2*Katz2^2$	0.01351	25	0.28187	42	0.09871	27	0.1207	36	0.26073	12	0.00743	44	0.12947	65
$BC1*BC2*Katz3^2$	0.01242	32	0.27896	43	0.09922	26	0.12444	31	0.2619	10	0.00766	41	0.14257	58
$CC1*CC2*Katz1^2$	0.0114	37	0.36158	16	0.13031	9	0.16125	9	0.2234	50	0.00518	62	0.11112	74
$CC1*CC2*Katz2^2$	0.01069	42	0.3567	24	0.13166	6	0.16398	6	0.22683	45	0.00753	43	0.11415	73
$CC1*CC2*Katz3^2$	0.00974	49	0.35033	26	0.13257	3	0.16557	4	0.22879	43	0.00767	40	0.12089	72

Table 2: AUC for Katz with different centralities. Highlights in dark grey represent that Inequality 18 holds, and light grey represents AUC values lower than the AUC of a random predictor

Katz similarity performs poorly for *infectiousContact* and *MITContact* datasets – we can see from Table 2, most of the AUC values are lower than random predictors. Also, we do not see any combination of Katz performing better than both the standalone Katz and the product of centralities divided by distance (Table 13), which means the combination does not satisfy Inequality 18. As a result, we do not have any empirical evidence suggesting that using inverse of Katz as distance in our proposed approach of Equation 2, could entail improved performance.

5.1.2. Combinations with AdamicAdar (AA)

	college Msg	rnk	contact	rnk	hep-th	rnk	hep-ph	rnk	hyper text	rnk	infectious Contact	rnk	MIT Contact	rnk
AA	0.00845	61	0.34479	29	0.13344	1	0.16241	7	0.22985	40	0.01069	26	0.17158	29
$DC1*DC2*AA^2$	0.01183	36	0.36166	15	0.12377	19	0.11257	42	0.24188	28	0.00541	57	0.16746	42
$BC1*BC2*AA^2$	0.01263	30	0.2785	45	0.07275	40	0.07279	53	0.26434	8	0.00978	30	0.16848	38
$CC1*CC2*AA^2$	0.00947	52	0.35018	27	0.12764	13	0.1371	26	0.23314	36	0.00658	50	0.17386	26

Table 3: AUC for AdamicAdar (AA) with different centralities. Highlights in dark grey represent that Inequality 18 holds, and light grey represents AUC values lower than the AUC of a random predictor

In Table 3 we also see similar pattern to Katz that, inverse of AdamicAdar (AA) similarity measure as a measurement of distance for Equation 2 does not entail improved[4] performance (i.e. it does not satisfy Inequality 18).

[4]Throughout this section, whenever we say a combination approach performs better or has improved performance, we imply it satisfies Inequality 18. Please see Section 5.1 for more details.

5.1.3. Combinations with Common Neighbours (CN)

	college Msg	rnk	contact	rnk	hep-th	rnk	hep-ph	rnk	hyper text	rnk	infectious Contact	rnk	MIT Contact	rnk
CN	0.00825	62	0.3433	31	0.13139	7	0.1572	12	0.22578	47	0.00984	28	0.17138	30
$DC1 * DC2 * CN^2$	0.0114	38	0.36073	19	0.12757	14	0.13528	27	0.23876	30	0.00563	55	0.1673	43
$BC1 * BC2 * CN^2$	0.01236	34	0.27863	44	0.0884	34	0.09332	49	0.26171	11	0.00898	32	0.16842	39
$CC1 * CC2 * CN^2$	0.00965	51	0.34967	28	0.13173	4	0.15674	13	0.22885	42	0.0064	51	0.17366	28

Table 4: AUC for Common Neighbours (CN) with different centralities. Highlights in dark grey represent that Inequality 18 holds, and light grey represents AUC values lower than the AUC of a random predictor

We can see in Table 4, that combining inverse of Common Neighbour (CN) with centrality (as a measurement of popularity or mass for Equation 2) improves performance of link prediction for one dataset. This is expressed by the fact that one of the values of AUC satisfies Inequality 18. There is one such case which is highlighted using dark grey box in Table 4. This improvement is seen when the combination of CN is with closeness centrality for *hep-th* dataset. However, except for combination of CN with CC in the *hep-th* dataset, there is no other evidence that any other combination of CN satisfies Inequality 18.

5.1.4. Combinations with Jaccards Coefficient (JC)

	college Msg	rnk	contact	rnk	hep-th	rnk	hep-ph	rnk	hyper text	rnk	infectious Contact	rnk	MIT Contact	rnk
JC	0.00476	68	0.16494	57	0.06883	43	0.14048	25	0.22959	41	0.02935	19	0.25703	2
$DC1 * DC2 * JC^2$	0.00615	65	0.31865	37	0.00574	73	0.01561	73	0.25224	19	0.00508	64	0.19332	18
$BC1 * BC2 * JC^2$	0.00721	64	0.24436	48	0.0103	71	0.02022	72	0.2725	1	0.01009	27	0.16706	44
$CC1 * CC2 * JC^2$	0.00541	67	0.04442	72	0.00489	74	0.0151	74	0.20351	61	0.00524	61	0.2237	7

Table 5: AUC for Jaccards Coefficient (JC) with different centralities. Highlights in dark grey represent that Inequality 18 holds, and light grey represents AUC values lower than the AUC of a random predictor

In quite a few cases, as presented in Table 5, Jaccards Coefficient (JC) combined with betweenness centrality gives improved performance (i.e. satisfies Inequality 18). These improvements are seen for *contact*, *hep-ph*, and *hypertext* datasets. In fact, for *hypertext* dataset, JC combined with betweenness centrality entails the best result (i.e. AUC value ranked one). These improvements support that, JC combined with betweenness centrality using 2 is a better link prediction method than using JC alone. Also, there is one case where for *hypertext* dataset, JC performs better when combined with degree centrality. However, closeness centrality combined with Jaccard's Coefficient (JC) does not satisfy Inequality 18.

5.1.5. *Combinations with Average Commute Time (ACT)*

	college Msg	rnk	contact	rnk	hep-th	rnk	hep-ph	rnk	hyper text	rnk	infectious Contact	rnk	MIT Contact	rnk
ACT	0.0134	26	0.35688	23	0.08106	38	0.06478	56	0.23875	31	0.00481	68	0.157	51
$DC1 * DC2 * ACT^2$	0.01371	22	0.38183	6	0.08451	35	0.06308	58	0.24294	25	0.00468	71	0.15241	56
$BC1 * BC2 * ACT^2$	0.01508	8	0.30064	38	0.04642	50	0.0492	61	0.26911	5	0.00535	58	0.1515	57
$CC1 * CC2 * ACT^2$	0.01351	24	0.3632	14	0.08108	37	0.06486	55	0.24524	23	0.00466	73	0.16568	45

Table 6: AUC for Average Commute Time (ACT) with different centralities. Highlights in dark grey represent that Inequality 18 holds, and light grey represents AUC values lower than the AUC of a random predictor

In Table 6 there are several cases when ACT combined with any of the three centrality measures gives better performance than using ACT alone or only centralities divided by the squared shortest path. However, such improvements are mainly observed for the *collegeMsg* dataset. Other than the *collegeMsg* dataset, combination of ACT with closeness centrality gives better prediction for *hep-th*. From this analysis we can see that, ACT combined with closeness centrality has more predictive power in link prediction than ACT combined with degree or betweenness centrality. This is because the first combination, ACT with closeness centrality, performs better (i.e. satisfies Inequality 18) in two (*collegeMsg* and *hep-th*) datasets and the other best performing combination, ACT with closeness centrality performs better in only one (*hep-th*) dataset. However, the number of datasets for which the combination with ACT satisfies Inequality 18 is lower than what we have seen for JC, MFI, and RPR. Combination of JC performs better i.e. satisfies Inequality 18 in two datasets whereas JC, MFI, and RPR performs better in three, four, and five datasets respectively.

5.1.6. *Combinations with Average Commute Time Normalised (ACTN)*

	college Msg	rnk	contact	rnk	hep-th	rnk	hep-ph	rnk	hyper text	rnk	infectious Contact	rnk	MIT Contact	rnk
$ACTN$	0.00216	74	0.03339	74	0.03996	56	0.0634	57	0.12492	73	0.02379	21	0.17125	31
$DC1 * DC2 * ACTN^2$	0.01398	18	0.38394	5	0.09163	32	0.07025	54	0.24261	27	0.00475	69	0.15853	50
$BC1 * BC2 * ACTN^2$	0.01516	7	0.17047	55	0.02844	64	0.04175	66	0.26346	9	0.00896	33	0.17062	34
$CC1 * CC2 * ACTN^2$	0.00581	66	0.35745	21	0.04116	55	0.07523	52	0.16509	67	0.00583	54	0.21415	11

Table 7: AUC for Average Commute Time Normalised (ACTN) with different centralities. Highlights in dark grey represent that Inequality 18 holds, and light grey represents AUC values lower than the AUC of a random predictor

Table 7 shows two cases of ACTN, where the predictability is improved when combined with degree centrality for *collegeMsg* and *hep-th* datasets. There is also one similar improvement with betweenness centrality for the *collegeMsg* dataset. However, there is no combination with closeness centrality which satisfies Inequality 18. Based on the number of datasets where combination with ACTN perform well, we could argue there is weak evidence that the two different combinations of ACTN with degree and closeness centrality may have good potential for predicting future links.

5.1.7. Combinations with Rooted PageRank (RPR)

	college Msg	rnk	contact	rnk	hep-th	rnk	hep-ph	rnk	hyper text	rnk	infectious Contact	rnk	MIT Contact	rnk
$RPR0.15$	0.01025	45	0.11534	62	0.06051	48	0.15376	17	0.23686	34	0.15386	6	0.20978	13
$RPR0.25$	0.00991	47	0.11244	63	0.06116	46	0.15653	15	0.23106	38	0.12798	8	0.21024	12
$DC1 * DC2 * RPR0.15^2$	0.01403	17	0.33405	33	0.12221*	21	0.15744	11	0.25588	16	0.01163	23	0.16394	46
$DC1 * DC2 * RPR0.25^2$	0.01404	16	0.33078	35	0.12806	11	0.17288	1	0.2575	14	0.01265	22	0.16996	35
$BC1 * BC2 * RPR0.15^2$	0.01499	11	0.17568	51	0.0588	49	0.09366	48	0.26893	6	0.06414	11	0.19521	17
$BC1 * BC2 * RPR0.25^2$	0.01504	10	0.17169	54	0.06406	44	0.10471	43	0.26978	4	0.06413	12	0.20235	15
$CC1 * CC2 * RPR0.15^2$	0.01058	44	0.1488	58	0.0607	47	0.15398	16	0.24394	24	0.07155	10	0.21519	10
$CC1 * CC2 * RPR0.25^2$	0.01018	46	0.138	59	0.06137	45	0.15673	14	0.23727	33	0.05944	13	0.21563	9

Table 8: AUC for Rooted PageRank (RPR) with different centralities. Highlights in dark grey represent that Inequality 18 holds, and light grey represents AUC values lower than the AUC of a random predictor

Inverse of Rooted PageRank (RPR) is one of the best measures for distance (according to Equation 2) from Section 4.2. Table 8 shows that for *hep-th*, *collegeMsg*, *hypertext* and, *hep-ph* datasets, when RPR is combined with degree centrality, the combination outperforms individual performance of RPR or degree centrality divided by shortest path (i.e. satisfies Inequality 18). Also, for *collegeMsg*, *hep-th* and, *Contact* datasets similar improvement is observed when RPR is combined with betweenness centrality. Only in one case (with two different values for α parameter of RPR) we see that combination of RPR with closeness centrality satisfies Inequality 18. From this analysis it is apparent that, RPR combined with degree centrality could be a better choice for link prediction than only using RPR.

5.1.8. Combinations with Pseudoinverse of the Laplacian matrix (PsInLap)

	college Msg	rnk	contact	rnk	hep-th	rnk	hep-ph	rnk	hyper text	rnk	infectious Contact	rnk	MIT Contact	rnk
$PsInLap$	0.00909	54	0.2641	47	0.03336	61	0.10005	45	0.21214	59	0.25286	1	0.17385	27
$DC1 * DC2 * PsInLap^2$	0.01239	33	0.33506	32	0.02588	65	0.08148	50	0.15964	68	0.17834	4	0.16009	49
$BC1 * BC2 * PsInLap^2$	0.01374	21	0.08809	67	0.02189	68	0.04634	62	0.20339	62	0.1288	7	0.24667	3
$CC1 * CC2 * PsInLap^2$	0.00245	73	0.03747	73	0.00873	72	0.03238	70	0.11912	74	0.23169	2	0.28475	1

Table 9: AUC for Pseudoinverse of the Laplacian matrix (PsInLap) with different centralities. Highlights in dark grey represent that Inequality 18 holds, and light grey represents AUC values lower than the AUC of a random predictor

In Table 9, there are two combinations (with betweenness centrality and closeness centrality) with Pseudoinverse of the Laplacian matrix (PsInLap) which perform better than PsInLap or product of these centralities divided by shortest path. Because these improvements are only seen for one dataset, we do not have strong evidence to support the use of the combination of PsInLap using Equation 2 for link prediction.

5.1.9. Combinations with Local Path Index (LPI)

	college Msg	rnk	contact	rnk	hep-th	rnk	hep-ph	rnk	hyper text	rnk	infectious Contact	rnk	MIT Contact	rnk
$LPIeps0.01$	0.01495	12	0.36019	20	0.12541	16	0.15286	18	0.21593	55	0.00762	42	0.16774	40
$LPIeps0.02$	0.01547	5	0.3609	17	0.12387	18	0.14961	20	0.21409	56	0.0073	45	0.16773	41
$DC1 * DC2 * LPIeps0.01^2$	0.01506	9	0.36898	8	0.12182	22	0.1301	28	0.22758	44	0.00627	52	0.16357	47
$DC1 * DC2 * LPIeps0.02^2$	0.01557	3	0.36979	7	0.12083	23	0.12967	29	0.22596	46	0.00603	53	0.16355	48
$BC1 * BC2 * LPIeps0.01^2$	0.0161	2	0.28604	40	0.09365	31	0.09973	46	0.25459	17	0.00839	36	0.16875	37
$BC1 * BC2 * LPIeps0.02^2$	0.01663	1	0.28689	39	0.09398	30	0.10043	44	0.25346	18	0.00796	39	0.16882	36
$CC1 * CC2 * LPIeps0.01^2$	0.01491	14	0.36414	12	0.12466	17	0.14473	23	0.21932	53	0.007	47	0.17081	33
$CC1 * CC2 * LPIeps0.02^2$	0.01556	4	0.36552	11	0.1234	20	0.14378	24	0.21743	54	0.00665	49	0.17082	32

Table 10: AUC for Local Path Index (LPI) with different centralities. Highlights in dark grey represent that Inequality 18 holds, and light grey represents AUC values lower than the AUC of a random predictor

From Table 10 we could see that Local Path Index (LPI) performs better when combined with betweenness centrality than on its own. This improvement can be observed for *collegeMsg* and *MITContact* datasets. In addition, for *collegeMsg* dataset, LPI improves when it is combined with degree centrality and closeness centrality. These improvements are not due to the product of centralities or LPI itself but due to the applied combination. This is because these combinations satisfy Inequality 18. However, there is more prevalent evidence that, LPI combined with betweenness centrality is a better predictor of future links than LPI combined with degree centrality.

5.1.10. Combinations with Leicht-Holme-Newman Global Index (LGI)

	college Msg	rnk	contact	rnk	hep-th	rnk	hep-ph	rnk	hyper text	rnk	infectious Contact	rnk	MIT Contact	rnk
$LGI0.5$	0.00418	70	0.05162	70	0.04235	54	0.12159	35	0.14685	70	0.04951	14	0.18933	21
$LGI0.7$	0.00385	72	0.04821	71	0.04477	52	0.12031	39	0.14028	72	0.04178	15	0.1923	20
$DC1 * DC2 * LGI0.5^2$	0.00851	59	0.32063	36	0.11859	24	0.16784	2	0.19186	65	0.01133	25	0.17625	22
$DC1 * DC2 * LGI0.7^2$	0.0094	53	0.33098	34	0.11305	25	0.15801	10	0.18625	66	0.00982	29	0.17523	24
$BC1 * BC2 * LGI0.5^2$	0.01244	31	0.13035	61	0.07136	42	0.12059	37	0.24265	26	0.03963	16	0.22435	6
$BC1 * BC2 * LGI0.7^2$	0.01333	27	0.13741	60	0.07271	41	0.11554	41	0.23936	29	0.02988	17	0.22917	4
$CC1 * CC2 * LGI0.5^2$	0.0043	69	0.06027	69	0.04242	53	0.12161	34	0.15047	69	0.02967	18	0.19328	19
$CC1 * CC2 * LGI0.7^2$	0.00398	71	0.06144	68	0.04486	51	0.12031	38	0.1441	71	0.02439	20	0.19758	16

Table 11: AUC for Leicht-Holme-Newman Global Index (LGI) with different centralities. Highlights in dark grey represent that Inequality 18 holds, and light grey represents AUC values lower than the AUC of a random predictor

In Table 11 we can see that Leicht-Holme-Newman Global Index (LGI) when combined with degree centrality always exhibits improved performance for *hep-th* and *hep-ph* datasets. These improvements might indicate that, this combination performs well for collaboration networks. Because *hep-th* and *hep-ph* both are the only collaboration networks we have. These improvements could suggest that for collaboration networks, combining LGI with degree centrality using Equation 2 could be a good approach for predicting future collaborations. However, this claim for collaboration network needs to be corroborated by evaluating this combination for more network datasets of collaboration networks. Performance for combination of LGI with betweenness centrality for the *hep-th* and *MITContact* datasets, and closeness centrality for *hep-ph* dataset, are also improved. Here, we have weak evidence of degree and betweenness centrality to perform better when combined with LGI, thus a better predictor than LGI itself.

5.1.11. *Combinations with Matrix Forest Index (MFI)*

	college Msg	rnk	contact	rnk	hep-th	rnk	hep-ph	rnk	hyper text	rnk	infectious Contact	rnk	MIT Contact	rnk
MFI	0.00978	48	0.27332	46	0.03846	58	0.12256	33	0.21244	57	0.21582	3	0.17394	25
$DC1 * DC2 * MFI^2$	0.01397	19	0.38795	3	0.09617	29	0.12733	30	0.24704	20	0.00846	35	0.13802	62
$BC1 * BC2 * MFI^2$	0.01535	6	0.20197	49	0.03556	60	0.06253	59	0.27084	2	0.07619	9	0.15345	55
$CC1 * CC2 * MFI^2$	0.01114	41	0.36762	10	0.03858	57	0.12314	32	0.26806	7	0.15773	5	0.22291	8

Table 12: AUC for Matrix Forest Index (MFI) with different centralities. Highlights in dark grey represent that Inequality 18 holds, and light grey represents AUC values lower than the AUC of a random predictor

Table 12 shows that Matrix Forest Index (MFI) when combined with degree centrality using Equation 2 outperforms the predictability of 1) MFI when used on its own and 2) product of degree centrality divided by shortest path. This can be observed for four out of seven datasets: *collegeMsg*, *hep-th*, *hep-ph*, and *hypertext*. Also, in two datasets, similar improvement is seen when combined with closeness (*hep-ph* and *hypertext*) and betweenness (*collegeMsg* and *hypertext*) centrality. We hence argue that MFI combined with degree centrality is a strong method for link prediction.

5.1.12. *Combinations with Shortest Path*

	college Msg	rnk	contact	rnk	hep-th	rnk	hep-ph	rnk	hyper text	rnk	infectious Contact	rnk	MIT Contact	rnk
$DC1$	0.00778	63	0.10156	64	0.03643	59	0.04261	65	0.20073	64	0.00543	56	0.15655	52
$DC2$	0.00893	56	0.17552	52	0.02298	66	0.03956	68	0.20564	60	0.00472	70	0.12636	69
$BC1$	0.00886	57	0.09526	65	0.01959	69	0.0358	69	0.22064	51	0.00947	31	0.15653	53
$BC2$	0.00907	55	0.18862	50	0.01516	70	0.03071	71	0.21233	58	0.01139	24	0.13938	59
$CC1$	0.00847	60	0.0902	66	0.03301	62	0.04356	64	0.23007	39	0.00507	65	0.20749	14
$CC2$	0.00865	58	0.16996	56	0.02245	67	0.04154	67	0.20218	63	0.00449	74	0.17528	23
$DC1 * DC2$	0.01377	20	0.38969	2	0.08237	36	0.05854	60	0.24605	21	0.00466	72	0.13807	61
$DC1 * DC2 * 1/sp^2$	0.0136	23	0.38976	1	0.08983	33	0.07696	51	0.24601	22	0.00483	67	0.13812	60
$BC1 * BC2 * 1/sp^2$	0.01492	13	0.17511	53	0.0288	63	0.04596	63	0.26983	3	0.0085	34	0.15626	54
$CC1 * CC2 * 1/sp^2$	0.01314	28	0.38662	4	0.07964	39	0.09746	47	0.25756	13	0.00511	63	0.22749	5

Table 13: AUC for Shortest path with different centralities. Highlights in dark grey represent that a combination method performs better than PA, and light grey represents AUC values lower than the AUC of a random predictor

From Table 13 we could see that for the case where we use the shortest path in combination with degree centrality, even with a slight variation of shortest path length (due to the small-world phenomena the range of shortest path tend to be small) gives better performance than only using the product of degree centrality i.e. the Preferential Attachment (PA) similarity measurement. These improvements are seen in five out of seven datasets. This finding is consistent with findings by Wahid-Ul-Ashraf et al. [52]. Here we have compared degree centrality combined with the shortest path against PA because PA is the product of degree centrality. The baseline method here is PA instead of Inequality 18 as the combination of centrality and the shortest path itself served as baselines for other results of combination methods discussed so far. PA is a well-established link prediction method that we have discussed further in Section 4.2 [26]. Other than the DC with the shortest path, BC and CC combined with the shortest path also perform better than PA. BC with the shortest path performs better in four datasets and CC with the shortest path performs better in three datasets (although it performs better than PA for the *infectiousContact* dataset the predictability is not better than a random predictor).

5.2. Best Methods

Methods which satisfy Inequality 18 are the only ones which we analyse here. The reason for this selection is discussed in Section 5.1. From the selected combination methods, we use three different evaluation techniques to calculate scores in Table 14. The 'Dataset variability score' is the number of datasets for which a combination approach satisfies Inequality 18. We also calculate a score based on ranks. In our analysis the lowest rank of a method is 74, as we have 74 methods in total including the standalone methods from Tables 2-13. We subtract 73 (so that the worst method with rank 74 will have a score 1) from the rank of a method in a dataset to get a score instead of rank. Afterwards, we sum the scores up to get the final score which is represented as 'Score (73-Rank)' in the table. This score not only tells us for how many datasets a method performs well but also that method's relative performance among all the other methods. Finally, we normalise 'Score (73-Rank)' by the number of datasets for which a method satisfies Inequality 18. This normalised version of the rank-based score is represented as 'Normalised Score (73-Rank)' and considers a combination method's rank based on the average performance on all datasets.

Method	college Msg	contact	hep-th	hep-ph	hypertext	infectious Contact	MIT Contact	Dataset Variablity Score	Score (73-Rank)	Normalised Score (73-Rank)
RPR0.25+DC	Y(16)		Y(11)	Y(1)	Y(14)			4***	254***	63.5
MFI+BC	Y(6)				Y(2)			2*	214**	107***
MFI+DC	Y(19)		Y(29)	Y(30)	Y(20)			4***	198*	49.5
RPR0.15+DC	Y(17)			Y(11)	Y(16)			3**	178	59.3
DC+SP		Y(1)	Y(33)	Y(51)			Y(60)	3**	151	50.3
LGI0.5+DC			Y(24)	Y(2)				2*	122	61
LGI0.7+DC			Y(25)	Y(10)				2*	113	56.5
LPIeps0.02+BC	Y(1)						Y(36)	2*	111	55.5
LPIeps0.01+BC	Y(2)						Y(37)	2*	109	54.5
MFI+CC				Y(32)	Y(7)			2*	109	54.5
LGI0.7+BC			Y(41)				Y(4)	2*	103	51.5
JC+BC		Y(48)		Y(72)	Y(1)			3**	101	33.67
LGI0.5+BC			Y(42)				Y(6)	2*	100	50
ACTN+DC	Y(18)		Y(32)					2*	98	49
RPR0.25+BC	Y(10)		Y(44)					2*	94	47
ACT+CC	Y(24)		Y(37)					2*	87	43.5
RPR0.15+BC	Y(11)	Y(51)						2*	86	43
PsInLap+CC							Y(1)	1	73	73**
PsInLap+BC							Y(3)	1	71	71*
LPIeps0.02+DC	Y(3)							1	71	71*
CN+CC			Y(4)					1	70	70
LPIeps0.02+CC	Y(4)							1	70	70
ACTN+BC	Y(7)							1	67	67
ACT+BC	Y(8)							1	66	66
LPIeps0.01+DC	Y(9)							1	65	65
RPR0.25+CC				Y(14)				1	60	60
RPR0.15+CC				Y(16)				1	58	58
JC+DC					Y(19)			1	55	55
ACT+DC	Y(22)							1	52	52
LGI0.5+CC				Y(34)				1	40	40
LGI0.7+CC				Y(38)				1	36	36

Table 14: Methods which satisfy Inequality 18. The dataset(s), in which a method satisfied Inequality 18 is marked as Y, and the rank of that method mentioned in the parenthesis, i.e. Y(rank). First best score is marked with ***, second best with** and third best with *. For all the scores, higher is better. The '+' operator entails a combination based on the Equation 2.

5.3. Results Analysis for each Dataset

Based on the methods we use and the combination of them we conclude that some datasets can be be more predictable than others. By comparing AUC of the PR curves, it seems that *hep-th* and *collegeMsg* datasets are the most predictable, as only two methods perform worse than a random predictor. Overall the collaboration networks *hep-th* and *hep-ph* have good predictability. Only two methods for the *hep-th* and three methods for the *hep-ph* collaboration network perform worse than a random predictor. On the other hand, *infectiousContact* dataset has the lowest predictability – there are 37 out of 74 (including combinations) methods whose performance is worse than a random predictor. The second worst dataset in terms of predictability is *MITContact* where 11 methods perform lower than a random predictor. For *hypertext* we have six methods performing worse than a random predictor. Overall, except *contact* dataset, where we have only three methods with AUC lower than a random predictor, all the networks representing human contact seem to have low predictability. We discuss below the results from the perspective of individual datasets and interpret those outcomes in the context of characteristics of each social network tested:

1. **collegeMsg:** Overall, performance of methods on *collegeMsg* does not appear to be very good when compared to the remaining datasets. However, when we compare with a random predictor, many of the predictors seem to perform better. The best performing methods for *collegeMsg* are those based on LPI in combination with BC. As LPI in its nature is similar to CN it is surprising that the highest rank for CN–based method for *collegeMsg* dataset is ranked 34. It means that consideration of friend-of-friend-of-friend (path of length three) in LPI rather than friend-of-a-friend (CN) makes a (positive) difference for prediction.
2. **contact:** For contact network the best performing methods are the ones based on DC and the top ranked is DC coupled with the shortest path. Also, DC on its own (rank two), DC+MFI (rank three), CC+shortest path (rank four), DC+ACTN (rank five) and DC+ACT (rank six) perform well. All these methods are path-based but they must be combined with information about node degree to achieve good performance, e.g. DC+ACTN has rank five and ACTN on its own is last in the ranking (rank 74). However, this improved performance when combined with DC might be due to the fact that Preferential Attachment (product of degrees) is the second best predictor. Thus, although dividing DC by ACTN still makes it a good predictor, its performance is worse than when only degree product is used.
3. **hep-th:** Although the best method for *hep-th* is AA, the best performing set of methods are those based on Katz and combined with CC. Katz2 and Katz3 also performed very well with ranks five and two respectively. Also, methods combing CC with Katz3, CN, and Katz2 were performing very well (rank three, four, and six respectively). However, standalone Katz performs better than in a combination. On the other hand, note that again, we need to have a proper combination of metrics because CC combined with JC gives the poorest performance. It shows that taking into account the greater network (Katz enables that) not only the immediate neighbourhood of a node (JC) may result in better performance. It is surprising that although AA is very similar to JC, their performance differs so much with AA being ranked one and JC - 43 (0.06 accuracy for JC and 0.13 for AA). The interpretation may be that AA gives importance to the degree of common neighbour and if common neighbour degree is lower then there is a bigger chance that he/she will introduce two of his/her neighbours to each other. JC on the other hand focuses only on overall number of common friends. This indicates that when developing new prediction methods, we should also focus on other factors and capacity of other nodes rather than just the nodes in question.
4. **hep-ph:** Overall, for *hep-ph* dataset methods based on Katz and Katz combined with CC and DC perform best. However, the top two results are those that combine DC with RPR and LGI. Methods based on JC combined with different centralities give the worst results. It seems that merging local information (DC) with knowledge about paths throughout the network and appropriately weighting them (Katz, RPR, LGI) gives the best results. Similarity RPR and LGI combined with degree centrality outperform DC, RPR or LGI used as a standalone predictor. Similarly, for this dataset, LGI performs better (compared with using it independently) when combined with betweenness and closeness centrality.

5. **hypertext:** For the *hypertext* dataset the best set of methods are those that use BC as the centrality measure which is the most overreaching centrality out of those we analysed. BC is present in 11 out of 13 top ranked methods for this dataset. This improvement could be explained by looking at Table 13. We can see that BC combined with shortest path is the third best predictor for this dataset. In addition, Table 5 shows that JC works well for a measurement of distance for *hypertext* dataset when JC is combined with BC, it has the best predictability.
6. **infectiousContact:** Most of the predictors perform poorly for the *infectiouseContact* dataset. This low predictability may be indicative of the dataset containing many random interactions between people. Each of the edges represents interaction between two people at the INFECTIOUS: STAY AWAY exhibition at the Science Gallery in Dublin, Ireland, from April 17th to July 17th, 2009 [74]. This dataset captured interactions between members of general public at the exhibition [74]. Other contact networks however, such as the *hypertext* network, capture interaction between the attendees [75]. It would be more likely that in the conference people would have interacted less randomly than the exhibition. This is because in the conference, people would speak to other people who might have similar research interests. Also, in a conference one person who might have a very interesting research contribution might get more interaction with other people. Methods based on PsInLap work best for *infectiousContact* network. It is very interesting as PsInLap can be interpreted using the concept of conductance and it can be very much connected with the fact that the network is a set of face-to-face interactions that took place in one location.
7. **MITContact:** This dataset is interesting as methods that include Katz are the ones whose performance is the poorest and this is very uncommon that Katz performance capability is so low. 11 out of 12 worst performing methods include Katz element. However, Katz seems to perform better for collaboration networks as it has been seen in the study by Liben-Nowell and Kleinberg [18]. We also see similar result in Table 2 that for both of the collaboration networks *hep-th* and *hep-ph*, performance of Katz is good. It is interesting to see that when PsInLap is combined with closeness centrality and betweenness centrality, it outperforms PsInLap used as a standalone predictor. Also, using inverse of PsInLap instead of geodesic path as a measurement of distance gives better performance for this dataset only. In addition, LPI combined with BC satisfies Inequality 18.

5.4. Computational Complexity

In terms of computational complexity, we have discussed in Section 5 that we need to make predictions for $\frac{|V|(|V|-1)}{2} - |E|$ links in total. Thus the time complexity is $O(|V|^2)$, if we wish to predict all possible non-existing links based on Equation 2. However, based on different algorithms, each of the methods (i.e. CN, Katz, rooted PageRank etc.) we have used in our combination approach may have different time and space complexities. For example, for CN, JC and AA, where traversal of node neighbourhood is required, the computational complexity is at least $O(|V|b^2)$, where b is the average degree of the graph [82, 64]. Among all the methods, PA has the lowest computational complexity of $O(2|V|)$, as we only need to multiply the predicted pair of nodes' degree. RPR could be calculated using different algorithms and the complexities vary from $O(|V|)$ to $O(|V|^2)$ (in case of a sparse network) [83, 84]. The computational complexity of calculating an inverse or pseudoinverse of a matrix is usually $O(|V|^3)$ [85] which is required for MFI, PsInLap, ACT, ACTN, Katz, and LGI. However, there is a faster alternative algorithm proposed especially for Katz, reducing the computational complexity from $O(|V|^3)$ to $O(|V|+|E|)$ [86]. LPI has a computational complexity of $O(|V|b^3)$ [64].

As for centralities, DC has a time complexity of $O(|V|^2)$. BC has $O(|V||E|)$ [56] and CC also has the same time complexity of $O(|V||E|)$ [56, 87, 57]. However, the complexity may vary depending on the algorithm used as pointed out in [57].

For shortest path calculation, there is a range of algorithms available and time complexity depends on the used algorithm. Algorithm selection for shortest path calculation of a graph is based on several factors, such as available computational power and memory, graph type (weighted, directed etc.), graph size, and graph density. Additionally, calculating a selective set of pairs' shortest path or calculating an all pair shortest path could require different algorithms, resulting in different computational and space complexities. For example, all pair shortest path calculation using the Floyd–Warshall algorithm has a time complexity of $O(|V|^3)$ [88] and the Seidel's algorithm has complexity of $O(H(|V|)log|V|)$ (where $H(|V|)$ is the time complexity of multiplying two $|V| \times |V|$ matrices of small integers) [89]. The time complexity of the Johnson's all pair shortest path is $O(min(|V|^{2+\frac{1}{k}} + |V||E|, |V|^2 log|V| + |V||E|log|V|))$ [90].

The space complexity of CN, AA, JC is $O(|V|b^2)$ [64] and for a matrix inversion it is $O(|V|^2)$ [64]. Floyd–Warshall algorithm has a space complexity of $O(|V|^2)$.

All the time complexities discussed here are based on a serial processor. However, with the advancement of GPU and distributed computing, parallel and distributed graph algorithms are emerging and can be found in the literature very often. For example, You et al. [91] proposed an algorithms to calculate degree, closeness, and betweenness centrality measures in directed graphs. In terms of GPU computation, Gunrock is an excellent library which can calculate different centrality measures and shortest path [92]. In his paper Wang et al. [92] used very large graphs with millions of vertices and edges and shown the performance of their GPU computation from their graph analysis library Gunrock, which is much better than the performance of a serial processor. There is also another graph processing library with GPU computation available, which comes free with CUDA (NVIDIA's parallel computing framework) named nvGraphs, which shows a very fast PageRank calculation on a very large 1.5 billion edge dataset [93]. The library currently supports PageRank, single-source shortest path, and single-source widest path calculation [93]. The recent revolution of the GPU computation is not only benefiting deep learning but also graph computation [94, 95, 96, 97, 98].

6. Conclusions and Future Work

In this paper, we proposed a new approach to link prediction in social networks, inspired by Newton's law of universal gravitation, which states that the force exerted between two masses is proportional to the product of those masses, and inversely proportional to the squared distance between their centres [50]. We have performed extensive empirical analysis to investigate the potential of our link prediction method.

Our experiments indicate that in many cases a combination method, using Equation 2 improves performance with respect to either standalone similarity measure used in that combination or the product of centralities divided by distance squared (Inequality 18). In cases where we see these improvements (i.e. for all the datasets except *infectiousContact*), we have also seen that AUC values are higher than that of a random predictor. The significant improvements of RPR, LGI,and MFI in terms of the AUC on average, demonstrate that our combination approach has great potential as a link prediction method. Combinations of LGI, shortest path, and MFI with DC work well for both of the collaboration networks, *hep-th* and *hep-ph*. ACT, ACTN with DC, LPI with DC, BC, and CC, MFI and RPR with DC and BC, work best for *collegeMsg* dataset. JC with BC and shortest path with DC work best for *contact* dataset. As for *hypertext* dataset JC with BC and DC, RPR with DC, MFI with DC, BC, and CC, work best. In *MITContact* dataset, PSInLap with BC and CC, LPI with BC, LGI with BC perform best. As for *infectiousContact* none of the combinations works well. In fact, most of the standalone similarity measures perform worse than a random predictor. The exception is PsInLap which works best for *infectiousContact* dataset.

From our empirical analysis, we have concluded that there are a number of combinations which perform better than others. The combination of RPR with degree centrality in Table 5.1.7 can be used as a better predictor than using RPR on its own. In addition to RPR, LGI with DC for collaboration networks, MFI with DC, and DC with shortest paths are the best overall combinations that we found in our study.

One powerful property of our approach also allows us to combine local and global measures (e.g. DC with RPR, which considers the larger structure of the surrounding vertex or vertices such) for link prediction. For a pair of vertices, it might happen that the global structure may not indicate link formation probability strongly enough, but the local structure indicates otherwise or vice versa. Due to the combination of local and global measures, in such cases, the final score of link formation would still be higher compared with considering only a local or global measure. Thus, a combination of global and local may improve link formation predictability for pairs of nodes which are likely to be ignored (i.e. false negatives) by a predictor which considers only single local or global measure.

We have discussed similarities between physical networks and social networks in Sections 1 and 2.2. Our Newtonian gravity inspired link prediction method shows that even at a local level the dynamics of a social network can be interpreted through physical law. The similarity between physical and social world are often encountered. Perhaps one of the most well-known examples is the similarity between complex weather models and social dynamics [99], which supports the idea of benefiting from this kind of similarities between social and physical world. The benefits would come from cross-applying modelling and analytical tools from these domains. However, most of these similarities are emergent phenomena due to the characteristics of a complex system, at a global level. For example, we have discussed how physical and social networks exhibit similar global properties like high clustering coefficient, degree centrality etc. However, our study shows that we may also benefit from applying laws from physical world to a social network even at the local level.

The inverse square relation between physical quantity (or intensity) and distance is widely found in nature and is known as the Inverse-Square Law. Some examples include sound transmission [100], force between two electrostatic charges [101], intensity of radiation [102] and more. The quadratic form of inverse squared distance that we observe for several cases of intensity or quantity in nature is due to three spatial dimensions, which characterise our physical world [103]. In our case of social networks, we are directly using the same Inverse-Square Law found in nature. For example, in the combination method of RPR with DC, the inverse of RPR is the path length analogous to the distance in Newton's gravitational law in Equation 1. The squared distance in Newton's law is a result of three spatial dimensions. But for our approach in Equation 2, other than the quadratic order, it might be possible to obtain better performance by using an order of one, three, four etc of the RPR. Optimal order of the dissimilarity measure could be learnt from the ground truth of the data such that the dimension for which using Equation 2 gives the best prediction result. This is something we aim to do in future and goes beyond the scope of one study.

In terms of computational and space complexity, we have discussed in Section 5.4 that we need to make a prediction of $\frac{|V|(|V|-1)}{2} - |E|$ links in total. Thus the worst case time complexity is at least $O(|V|^2)$, if we wish to predict all possible non-existing links. However, each of the methods (i.e. Katz, rooted PageRank etc.) we have used in our combination approach may have different time and space complexity. For example computational complexity of different algorithms to calculate Katz could range from $O(|V|^3)$ to $O(|V|+|E|)$ [86]. A detailed and in-depth analysis of the complexity goes beyond the scope of one paper and we hope to discuss this in our future work.

References

[1] J. E. Cohen, F. Briand, C. M. Newman, Community food webs: data and theory, volume 20, Springer Science & Business Media, 2012.

[2] H. Jeong, S. P. Mason, A.-L. Barabási, Z. N. Oltvai, Lethality and centrality in protein networks, Nature 411 (2001) 41–42.

[3] D. S. Bassett, E. T. Bullmore, Small-world brain networks revisited, The Neuroscientist (2016) 1073858416667720.

[4] D. Krioukov, M. Kitsak, R. S. Sinkovits, D. Rideout, D. Meyer, M. Boguñá, Network cosmology, Scientific reports 2 (2012).

[5] W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of anthropological research 33 (1977) 452–473.

[6] J. Scott, Social network analysis, Sage, 2017.

[7] A.-L. Barabasi, Z. N. Oltvai, Network biology: understanding the cell's functional organization, Nature reviews genetics 5 (2004) 101–113.

[8] O. Sporns, J. D. Zwi, The small world of the cerebral cortex, Neuroinformatics 2 (2004) 145–162.

[9] S. P. Borgatti, A. Mehra, D. J. Brass, G. Labianca, Network analysis in the social sciences, science 323 (2009) 892–895.

[10] M. Gong, B. Fu, L. Jiao, H. Du, Memetic algorithm for community detection in networks, Physical Review E 84 (2011) 056101.

[11] G. Bianconi, A.-L. Barabási, Bose-einstein condensation in complex networks, Physical review letters 86 (2001) 5632.

[12] K. Juszczyszyn, A. Musial, K. Musial, P. Bródka, Molecular dynamics modelling of the temporal changes in complex networks, in: Evolutionary Computation, 2009. CEC'09. IEEE Congress on, IEEE, 2009, pp. 553–559.

[13] J. Urry, Small worlds and the new social physics, Global networks 4 (2004) 109–130.

[14] M. Budka, K. Juszczyszyn, K. Musial, A. Musial, Molecular model of dynamic social network based on e-mail communication, Social Network Analysis and Mining 3 (2013) 543–563.

[15] C. A. Bliss, M. R. Frank, C. M. Danforth, P. S. Dodds, An evolutionary algorithm approach to link prediction in dynamic social networks, Journal of Computational Science 5 (2014) 750–764.

[16] D. Hristova, A. Noulas, C. Brown, M. Musolesi, C. Mascolo, A multilayer approach to multiplexity and link prediction in online geo-social networks, EPJ Data Science 5 (2016) 24.

[17] L. Getoor, C. P. Diehl, Link mining: a survey, Acm Sigkdd Explorations Newsletter 7 (2005) 3–12.

[18] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, journal of the Association for Information Science and Technology 58 (2007) 1019–1031.

[19] L. Lü, T. Zhou, Link prediction in complex networks: A survey, Physica A: statistical mechanics and its applications 390 (2011) 1150–1170.

[20] M. Al Hasan, M. J. Zaki, A survey of link prediction in social networks, in: Social network data analytics, Springer, 2011, pp. 243–275.

[21] P. Wang, B. Xu, Y. Wu, X. Zhou, Link prediction in social networks: the state-of-the-art, Science China Information Sciences 58 (2015) 1–38.

[22] V. Martínez, F. Berzal, J.-C. Cubero, A survey of link prediction in complex networks, ACM Computing Surveys (CSUR) 49 (2016) 69.

[23] M. E. Newman, Clustering and preferential attachment in growing networks, Physical review E 64 (2001) 025102.

[24] S. Gerard, J. M. Michael, Introduction to modern information retrieval, McGraw-Hill, New York, 1983.

[25] L. A. Adamic, E. Adar, Friends and neighbors on the web, Social networks 25 (2003) 211–230.

[26] A.-L. Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, T. Vicsek, Evolution of the social network of scientific collaborations, Physica A: Statistical mechanics and its applications 311 (2002) 590–614.

[27] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1953) 39–43.

[28] S. Brin, L. Page, Reprint of: The anatomy of a large-scale hypertextual web search engine, Computer networks 56 (2012) 3825–3833.

[29] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American society for information science 41 (1990) 391.

[30] U. Essen, V. Steinbiss, Cooccurrence smoothing for stochastic language modeling, in: Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on, volume 1, IEEE, 1992, pp. 161–164.

[31] L. Lee, Measures of distributional similarity, in: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, Association for Computational Linguistics, 1999, pp. 25–32.

[32] K. Juszczyszyn, K. Musial, M. Budka, Link prediction based on subgraph evolution in dynamic social networks, in: Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), IEEE, 2011, pp. 27–34.

[33] L. Backstrom, J. Leskovec, Supervised random walks: predicting and recommending links in social networks, in: Proceedings of the fourth ACM international conference on Web search and data mining, ACM, 2011, pp. 635–644.

[34] R. N. Lichtenwalter, J. T. Lussier, N. V. Chawla, New perspectives and methods in link prediction, in: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2010, pp. 243–252.

[35] J. Chen, W. Geyer, C. Dugan, M. Muller, I. Guy, Make new friends, but keep the old: recommending people on social networking sites, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2009, pp. 201–210.

[36] R. N. Lichtenwalter, N. V. Chawla, Lpmade: Link prediction made easy, Journal of Machine Learning Research 12 (2011) 2489–2492.

[37] M. Fire, L. Tenenboim, O. Lesser, R. Puzis, L. Rokach, Y. Elovici, Link prediction in social networks using computationally efficient topological features, in: Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational

Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on, IEEE, 2011, pp. 73–80.

[38] R. Tan, J. Gu, P. Chen, Z. Zhong, Link prediction using protected location history, in: Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on, IEEE, 2013, pp. 795–798.

[39] M. Al Hasan, V. Chaoji, S. Salem, M. Zaki, Link prediction using supervised learning, in: SDM06: workshop on link analysis, counter-terrorism and security, 2006.

[40] F. Papadopoulos, M. Kitsak, M. Á. Serrano, M. Boguná, D. Krioukov, Popularity versus similarity in growing networks, Nature 489 (2012) 537–540.

[41] P. Thwe, Proposed approach for web page access prediction using popularity and similarity based page rank algorithm, International Journal of Scientific & Technology Research 2 (2013) 240–246.

[42] L. C. Freeman, A set of measures of centrality based on betweenness, Sociometry (1977) 35–41.

[43] M. J. Bannister, D. Eppstein, M. T. Goodrich, L. Trott, Force-directed graph drawing using social gravity and scaling, in: International Symposium on Graph Drawing, Springer, 2012, pp. 414–425.

[44] F. Simini, M. C. González, A. Maritan, A.-L. Barabási, A universal model for mobility and migration patterns, Nature 484 (2012) 96–100.

[45] H. C. Carey, Principles of social science, volume 3, JB Lippincott & Company, 1867.

[46] D. W. Griesinger, Reconsidering the theory of social gravity, Journal of Regional Science 19 (1979) 291–302.

[47] M. Levy, J. Goldenberg, The gravitational law of social interaction, Physica A: Statistical Mechanics and its Applications 393 (2014) 418–426.

[48] G. K. Zipf, Human behaviour and the principle of least effort: an introduction to human ecology, 1949.

[49] J. Q. Stewart, Demographic gravitation: evidence and applications, Sociometry 11 (1948) 31–58.

[50] I. Newton, Philosophiæ naturalis principia mathematica (mathematical principles of natural philosophy), London (1687) (1987).

[51] A. Crombie, Newton's conception of scientific method, Physics Bulletin 8 (1957) 350.

[52] A. Wahid-Ul-Ashraf, M. Budka, K. Musial-Gabrys, Newtons Gravitational Law for Link Prediction in Social Networks, in: C. Cherifi, M. K. Hocine Cherifi, M. Musolesi (Eds.), Complex Networks & Their Applications VI. COMPLEX NETWORKS 2017, Springer, Cham, 2018, pp. 93–104. doi:10.1007/978-3-319-72150-7_8.

[53] M. Newman, Networks: an introduction, Oxford university press, 2010.

[54] G. Csardi, T. Nepusz, The igraph software package for complex network research, InterJournal, Complex Systems 1695 (2006) 1–9.

[55] J. M. Anthonisse, The rush in a directed graph, Stichting Mathematisch Centrum. Mathematische Besliskunde (1971) 1–10.

[56] U. Brandes, A faster algorithm for betweenness centrality, Journal of mathematical sociology 25 (2001) 163–177.

[57] A. Landherr, B. Friedl, J. Heidemann, A critical review of centrality measures in social networks, Business & Information Systems Engineering 2 (2010) 371–385.

[58] J. Kunegis, A. Lommatzsch, Learning spectral graph transformations for link prediction, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 561–568.

[59] F. Fouss, A. Pirotte, J.-M. Renders, M. Saerens, Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation, IEEE Transactions on knowledge and data engineering 19 (2007) 355–369.

[60] D. J. Klein, M. Randić, Resistance distance, Journal of mathematical chemistry 12 (1993) 81–95.

[61] L. Lovász, Random walks on graphs, Combinatorics, Paul erdos is eighty 2 (1993) 1–46.

[62] D. Zhou, B. Schölkopf, Learning from labeled and unlabeled data using random walks, Lecture notes in computer science (2004) 237–244.

[63] T. Zhou, L. Lü, Y.-C. Zhang, Predicting missing links via local information, The European Physical Journal B-Condensed Matter and Complex Systems 71 (2009) 623–630.

[64] L. Lü, C.-H. Jin, T. Zhou, Similarity index based on local paths for link prediction of complex networks, Physical Review E 80 (2009) 046122.

[65] E. A. Leicht, P. Holme, M. E. Newman, Vertex similarity in networks, Physical Review E 73 (2006) 026120.

[66] P. Chebotarev, E. Shamis, The matrix-forest theorem and measuring relations in small social groups, arXiv preprint math/0602070 (2006).

[67] J. Kunegis, arxiv hep-th network dataset konect, http://konect.uni-koblenz.de/networks/ca-cit-HepTh, 2017. Accessed: November 2017.

[68] J. Kunegis, Konect: the koblenz network collection, in: Proceedings of the 22nd International Conference on World Wide Web, ACM, 2013, pp. 1343–1350.

[69] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Transactions on Knowledge Discovery from Data (TKDD) 1 (2007) 2.

[70] J. Kunegis, arxiv hep-ph network dataset konect, http://konect.uni-koblenz.de/networks/ca-cit-HepTh, 2017. Accessed: November 2017.

[71] J. Leskovec, J. Kleinberg, C. Faloutsos, Graphs over time: densification laws, shrinking diameters and possible explanations, in: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, ACM, 2005, pp. 177–187.

[72] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, J. Scott, Impact of human mobility on opportunistic forwarding algorithms, IEEE Transactions on Mobile Computing 6 (2007).

[73] J. Kunegis, Haggle network dataset konect, http://konect.uni-koblenz.de/networks/contact, 2013. Accessed: April 2017.

[74] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, W. Van den Broeck, What's in a crowd? analysis of face-to-face behavioral networks, Journal of theoretical biology 271 (2011) 166–180.

[75] J. Kunegis, Hypertext 2009 network dataset konect, http://konect.uni-koblenz.de/networks/contact, 2017. Accessed: April 2017.

[76] P. Panzarasa, T. Opsahl, K. M. Carley, Patterns and dynamics of users' behavior and interaction: Network analysis of an online community, Journal of the Association for Information Science and Technology 60 (2009) 911–932.

[77] J. Kunegis, Infectious network dataset konect, http://konect.uni-koblenz.de/networks/sociopatterns-infectious, 2017. Accessed: November 2017.

[78] J. Kunegis, Reality mining network dataset konect, http://konect.uni-koblenz.de/networks/contact, 2017. Accessed: April 2017.

[79] N. Eagle, A. S. Pentland, Reality mining: sensing complex social systems, Personal and ubiquitous computing 10 (2006) 255–268.

[80] J. Keilwagen, I. Grosse, J. Grau, Area under precision-recall curves for weighted and unweighted data, PLOS ONE 9 (2014).

[81] J. Grau, I. Grosse, J. Keilwagen, Prroc: computing and visualizing precision-recall and receiver operating characteristic curves in r, Bioinformatics 31 (2015) 2595–2597.

[82] A. Papadimitriou, P. Symeonidis, Y. Manolopoulos, Fast and accurate link prediction in social networking systems, Journal of Systems and Software 85 (2012) 2119–2132.

[83] T. Haveliwala, S. Kamvar, D. Klein, C. Manning, G. Golub, Computing PageRank using power extrapolation, Technical Report, Stanford, 2003.

[84] P. Berkhin, A survey on pagerank computing, Internet Mathematics 2 (2005) 73–120.

[85] P. Courrieu, Fast computation of moore-penrose inverse matrices, arXiv preprint arXiv:0804.4809 (2008).

[86] K. C. Foster, S. Q. Muth, J. J. Potterat, R. B. Rothenberg, A faster katz status score algorithm, Computational & Mathematical Organization Theory 7 (2001) 275–285.

[87] K. Okamoto, W. Chen, X.-Y. Li, Ranking of closeness centrality for large-scale social networks, in: International Workshop on Frontiers in Algorithmics, Springer, 2008, pp. 186–195.

[88] R. W. Floyd, Algorithm 97: shortest path, Communications of the ACM 5 (1962) 345.

[89] R. Seidel, On the all-pairs-shortest-path problem, in: Proceedings of the twenty-fourth annual ACM symposium on Theory of computing, ACM, 1992, pp. 745–749.

[90] D. B. Johnson, Efficient algorithms for shortest paths in sparse networks, Journal of the ACM (JACM) 24 (1977) 1–13.

[91] K. You, R. Tempo, L. Qiu, Distributed algorithms for computation of centrality measures in complex networks, IEEE Transactions on Automatic Control 62 (2017) 2080–2094.

[92] Y. Wang, A. Davidson, Y. Pan, Y. Wu, A. Riffel, J. D. Owens, Gunrock: A high-performance graph processing library on the gpu, in: ACM SIGPLAN Notices, volume 51, ACM, 2016, p. 11.

[93] Nvidia, nvgraph, 2019. URL: `https://docs.nvidia.com/cuda/nvgraph/index.html#nvgraph-api-reference`.

[94] S. N. Aher, S. M. Walunj, Accelerate the execution of graph processing using gpu, in: Information and Communication Technology for Intelligent Systems, Springer, 2019, pp. 125–132.

[95] D. Merrill, M. Garland, A. Grimshaw, Scalable gpu graph traversal, in: ACM SIGPLAN Notices, volume 47, ACM, 2012, pp. 117–128.

[96] P. Harish, P. Narayanan, Accelerating large graph algorithms on the gpu using cuda, in: International conference on high-performance computing, Springer, 2007, pp. 197–208.

[97] J. Zhong, B. He, Medusa: Simplified graph processing on gpus, IEEE Transactions on Parallel and Distributed Systems 25 (2014) 1543–1552.

[98] X. Shi, Z. Zheng, Y. Zhou, H. Jin, L. He, B. Liu, Q.-S. Hua, Graph processing on gpus: A survey, ACM Computing Surveys (CSUR) 50 (2018) 81.

[99] D. Helbing, Systemic risks in society and economics, in: Social Self-Organization, Springer, 2012, pp. 261–284.

[100] K. Marten, P. Marler, Sound transmission and its significance for animal vocalization, Behavioral ecology and sociobiology 2 (1977) 271–290.

[101] C. de Coulomb, Premiere memoire sur lelectricite et le magnetism. second memoire sur lelectricite et le magnetism. troisieme memoire sur lelectricite et le magnetism, Histoire de lAcadémie Royal des Sciences (1785) 569–638.

[102] C. E. Gutiérrez, A. Sabra, The reflector problem and the inverse square law, Nonlinear Analysis: Theory, Methods & Applications 96 (2014) 109–133.

[103] E. G. Adelberger, B. R. Heckel, A. E. Nelson, Tests of the gravitational inverse-square law, Annual Review of Nuclear and Particle Science 53 (2003) 77–121.