EasyChair Preprint

№ 13986

Towards Automated Algorithm Selection for Link Prediction

Lienke Brown and Stephan Nel

July 15, 2024

Towards automated algorithm selection for link prediction

L.M. Brown[1][0000−0001−5361−4688] and G.S. Nel[2][0000−0002−0293−1234]

[1] Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, South Africa
22941096@sun.ac.za

[2] Stellenbosch Unit for Operations Research in Engineering, Department of Industrial Engineering, Stellenbosch University, South Africa
gsnel@sun.ac.za

Abstract. Link prediction represents an important field within network science which involves the systematic (*i.e.* algorithmic) prediction of missing edges within a graph-based representation. In this study, the seminal algorithm selection problem is formally proposed and formulated for the domain of link prediction so as to facilitate the automated selection of algorithms. More specifically, a regression-based meta-learning approach is proffered, the aim of which is to approximate the relationship between algorithmic performance and graph-based features, thereby facilitating data-driven algorithm selection. The study's contributions include an appropriate formalisation of the algorithm selection problem for link prediction, together with the extraction of informative graph-based features as well as the generation of algorithmic performance in respect of various prominent link prediction approaches. A suitable meta-learner is trained with respect to the aforementioned meta-data in order to induce automated algorithm selection. Feature importance is also carried out so as to identify pertinent graph-based features in respect of the predictive task at hand. It may be inferred from the results that the meta-learner showcases admirable predictive capabilities in respect of diverse network data sets. Decision support in respect of link prediction algorithm selection may be induced — a novel and significant contribution to the domain of link prediction.

Keywords: Link prediction · Algorithm selection · Meta-learning.

1 Introduction

A *network* represents an effective approach towards abstracting the conceptual interconnectedness of objects that constitute a system. Networks are therefore foundational when attempting to model problems within various domains such as biology, sociology, and information technology [7, 23, 62]. Modelling (and subsequently analysing) networks by means of appropriate representational and computational techniques can result in important insight at different levels of

abstraction, ranging from the low-level dynamics of the system's individual elements to the high-level disposition of the system (as a whole) [62].

A *graph* represents the mathematical approach towards abstracting (*i.e.* modelling) networks which subsequently enables the computational representation and analysis thereof. A graph-based representation typically comprises a set of so-called *vertices*, together with a set of distinct *edges* [30]. These vertices and edges correspond to the objects and connections (*i.e.* relationships) within a network, respectively.

One of the most prevalent tasks within the multi-disciplinary field of network science is the systematic (*i.e.* algorithmic) prediction of *links*[3] within a network. This established task (or problem) is formally referred to as *link prediction* [39]. Links that are to be predicted may be regarded as *missing* links (due to erroneous representation) or *latent* links (which are yet to manifest temporally). Link prediction has been successfully applied to various domains, for example social networks, drug interaction networks, transportation systems, and recommender systems, to name but a few [3, 4, 37].

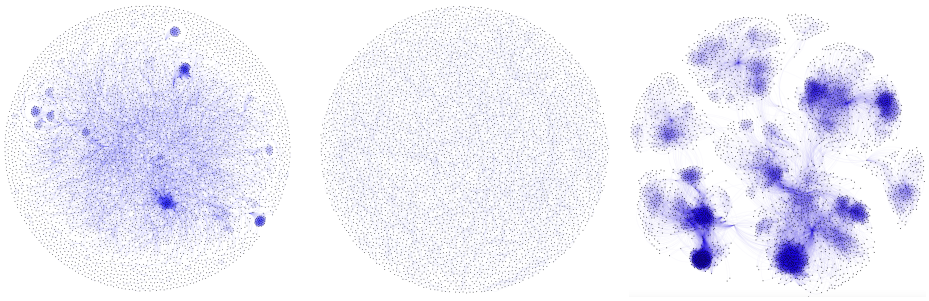
A key challenge associated with the task of predicting links accurately and in an automated (data-driven) manner relates to the importance of constructing a predictive model that incorporates various contextual features or properties. These network features ought to be informative in respect of different levels of abstraction, *i.e.* properties pertaining to individual nodes and their local neighbours, or properties describing the arrangement of multiple nodes in a more global context.

The diversity and complexity of networks have necessitated a broad range of link prediction algorithms which can be generally classified into similarity-based, classifier-based, and network embedding approaches [65]. The characteristic structure of three real-world networks, *i.e.* citation networks, infrastructure networks, and social networks, is visualised in Figure 1. The evident diversity of network structural characteristics may be observed.

No universal algorithm excels across all networks, as posited by the *No Free Lunch* (NFL) theorem [42, 64]. Consequently, link prediction algorithms ought to be carefully selected based on the structural characteristics of a network as it can significantly influence their algorithmic performance [15, 20, 65].

The prevailing diversity of link prediction algorithms and the distinct structural properties of different networks collectively induces the well-established *algorithm selection problem* (ASP) which was first proposed by John Rice in 1976 [49]. Rather than adopting some haphazard approach towards algorithm selection (which typically relies on rudimentary heuristics or some arbitrary intuition), a more structured, standardised, and systematic approach towards ASP is warranted so as to facilitate high-quality link prediction analyses in respect of different algorithmic approaches and diverse network problems [57].

[3] The term "link" refers to a network connection (or graph edge) and is ubiquitous within the domain of link prediction — this terminology is henceforth adopted in this paper.



(a) Citation network (b) Infrastructure network (c) Social network

Fig. 1: Network visualisations showcasing differences in network structural characteristics of (a) a citation network, (b) an infrastructure network, and (c) a social network [19].

Problem characterisation is a manifestly important facet of the ASP, as it involves the identification and extraction of correlative problem features from which inferential relationships are to be approximated in respect of algorithmic performance prediction. Successful ASP approaches should comprehensively address these aspects by incorporating diverse problem instances, computable features, algorithm suites, and performance metrics [57].

It is proffered that the domain of link prediction — accompanied by the diversity of algorithms and distinct network structural properties — is well-suited for the application of a generic and systematic algorithm selection approach. Inherent network characteristics (such as community structure, degree distribution, and assortativity, to name a few) represent informative features in respect of problem characterisation. Analysing and synthesising these network features together with, more importantly, their impact on algorithmic performance can be instrumental towards designing an automated (data-driven) algorithm selection approach.

Numerous studies in the literature have explored automated feature extraction techniques towards improving the effectiveness of predictive learning algorithms by means of algorithm selection — a general categorisation of this domain is *meta-learning* [8, 22, 32]. In the context of meta-learning, knowledge is gleaned from a collection of *meta-examples* (problem instances), each of which comprises their respective *meta-features* (problem characteristics) and the corresponding performance achieved by one or more algorithms, called *base-learners*. Conceptually, a so-called *meta-learner* is tasked with approximating the functional mapping from meta-features to algorithmic performance in a *supervised learning* manner — a predictive model may therefore be constructed that can aid in the selection of an appropriate algorithm for a new problem instance.

One prevalent meta-learning strategy involves the application of a *regression-based* learning model which aims to predict some numerical performance metric

(*e.g.* prediction error) of a base-learner in respect of a problem instance based on its meta-features. Various regression techniques, including *linear regression* [58] and *decision trees* [54], can be employed towards this end.

In this paper, a regression-based meta-learning approach is proposed for automated link prediction algorithm selection. The meta-learner is trained in respect of a diverse suite of network data sets, characterised by various network features, to which a broad range of base-learners are applied. The aim of the proposed methodology is to enhance link prediction quality by systematically selecting algorithms based on network characteristics and historical algorithmic performance data. The primary contributions can be summarised as follows:

- The ASP is formalised within the domain of link prediction which represents a structured basis on which various numerical experimentation is carried out.
- A comprehensive suite of benchmark network data sets is curated which represents a robust evaluatory foundation in respect of the considered link prediction algorithms.
- A diverse set of network structure features is proposed which effectively characterise various network data sets so as to facilitate informed algorithm selection.
- The construction and computerised implementation of a regression-based meta-learner is detailed from which key insight is inferred.
- An analysis of feature importance in respect of the meta-learning predictive task.

The remainder of this paper is organised as follows. A brief review of literature related to link prediction and algorithm selection is first presented in Section 2. Section 3 contains the formalisation of the link prediction algorithm selection problem. The computerised implementation — encompassing the meta-learning model, link prediction base-learners, meta-features, and benchmark data sets — is then outlined in Section 4. The experimental results are discussed in Section 5. The paper concludes in Section 6 with a summary of its contents and recommendations for future work.

2 Literature review

This section contains literature related to link prediction algorithm selection. In particular, the following topics are discussed: A formal description of the link prediction problem, a taxonomy of link prediction algorithms, and an elucidation of algorithm selection.

2.1 Link prediction

In network science, link prediction refers to the task of predicting missing (or latent) connections between nodes within a network [39]. Mathematically, let $G = (\mathcal{V}, \mathcal{E})$ denote a graph comprising n vertices and m edges, with vertex set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and edge set $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$. The set of non-existing

edges (*i.e.* non-adjacent vertex pairs) is denoted by $\hat{\mathcal{E}} = \{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_{\hat{m}}\}$, where $\hat{m} = \frac{n(n-1)}{2} - m$. Let $\mathcal{U}$ denote the set of all possible edges (often referred to as the *universal set*) which may be expressed as $\mathcal{U} = \mathcal{E} \cup \hat{\mathcal{E}} = \{e_1, \ldots, e_m, \hat{e}_1, \ldots, \hat{e}_{\hat{m}}\}$. For simplicity, let u_i denote a possible edge in $\mathcal{U}$, where u_i corresponds to e_i for $i \in \{1, \ldots, m\}$, while u_{m+j} corresponds to $\hat{e}_j$ for $j \in \{1, \ldots, \hat{m}\}$. Therefore $\mathcal{U} = \{u_1, u_2, \ldots, u_{m+\hat{m}}\}$.

A labelled data set $\mathcal{D}$ is defined according to which instances are assigned binary labels y_i for each edge u_i, where $y_i = 1$ if $u_i \in \mathcal{E}$, and $y_i = 0$ if $u_i \in \hat{\mathcal{E}}$. Feature vectors $\boldsymbol{x}_i$, capturing both local and/or global structural information of the graph, are associated with the end-vertices of u_i. Each feature vector comprises h network features. The data set $\mathcal{D}$ therefore comprises data instance pairs $(\boldsymbol{x}_i, y_i)$, from which a binary classification problem may be induced. The task of a link prediction algorithm therefore involves approximating the functional mapping $f : \mathbb{R}^h \to \{0, 1\}$ according to which a feature vector $\boldsymbol{x}_i$ is mapped to a label y_i, *i.e.* $y_i = f(\boldsymbol{x}_i)$ for all $(\boldsymbol{x}_i, y_i) \in \mathcal{D}$. The manner according to which the function f is approximated depends on the algorithmic approach adopted (discussed later). Conventionally, a fraction p of the data set $\mathcal{D}$ is randomly sampled in order to construct the training set $\mathcal{D}_{\text{train}}$, comprising $p(m + \hat{m})$ instances, while the remaining $1 - p$ fraction constitutes the test set $\mathcal{D}_{\text{test}}$.

A large number of link prediction algorithms (*i.e.* the base-learners) have been reported in the literature, each of which may be characterised by distinct strengths and weaknesses in respect of diverse network characterisitcs and application domains [21]. These algorithms can be broadly classified according to three paradigms: Similarity-based, classifier-based, and embedding-based methods [65]. Discussions regarding each of these categories are presented hereafter.

Similarity-based The working of similarity-based link prediction algorithms is based on the calculation and assignment of a so-called *similarity score* between a pair of vertices, denoted by $S_{(v_i, v_j)}$ for vertices v_i and v_j [39]. This score is employed towards approximating the likelihood of a potential edge between the vertex-pair — a large score is indicative of link existence (and *vice versa*). These scores utilise network topology in order to estimate the similarity between vertices, such as the number of *common neighbours* shared by a pair of vertices [39]. After scores are computed in respect of each vertex-pair under consideration, they are ranked, after which some classification threshold is typically applied in order to establish a suitable heuristic for link prediction. Similarity-based algorithms are typically categorised as follows: Neighbour-, path-, and random walk-based measures [65].

Classifier-based Link prediction may be contextualised by means of a machine (or statistical) learning formulation, according to which different features of the network may be utilised towards predicting link formation. This approach was first pioneered by Al-Hasan *et al.* [28] who formulated the link prediction problem as a supervised classification task, according to which the considered

data set comprises independent input variables (*i.e.* descriptive network features) together with the target dependent variable (*i.e.* binary-valued class instances) [35]. This binary classification task is amenable to various supervised learning approaches [28]. Typical classifiers implemented for link prediction include: Decision trees [54], *random forests* [11], *logistic regression* [31], *gradient boost* [18], and *multi-layered perceptrons* [50], to name a few.

Embedding-based The construction of a so-called embedding involves the abstraction of the essential structural and/or feature information of a vertex, edge, or even an entire subgraph into a fixed-size, real-valued vector [27]. More formally, given a graph $G = (\mathcal{V}, \mathcal{E})$, the task of generating network embeddings is akin to learning the functional mapping $f : v_i \mapsto \boldsymbol{r}_i \in \mathbb{R}^w$, where $\boldsymbol{r}_i$ denotes the real-valued vector representation of vertex v_i for $i \in \{1, \ldots, n\}$, and w denotes the dimensionality of the embedding space [66]. Two popular embedding-based categories include random walks and *graph neural networks* (GNNs) [65], each of which necessitates distinct modelling approaches.

Random walks generate embeddings based on the notion that vertices that tend to appear together in random walks[4] tend to share similar properties and should therefore have similar[5] embeddings [24, 44]. The effectiveness of these algorithms are predicated on the notion that node neighbourhoods are sufficiently informative in respect of link formation [62]. GNNs, on the other hand, involve the construction of embeddings by learning layered abstractions from feature information of a vertex's local and global neighbourhood, thereby leveraging the graph's structure in order to generate informative representations [67].

Embedding-based approaches vary in respect of their algorithmic working. Certain approaches involve the calculation of link probabilities directly [66], whilst other approaches generate vertex embeddings (during training) from which probabilities may be derived. Common techniques include the application of similarity measures between vertex embeddings, *e.g.* dot products, so as to estimate link probabilities. Another popular approach involves concatenating embeddings of vertex pairs, *i.e.* $(\boldsymbol{r}_j, \boldsymbol{r}_k)$, so as to form an edge feature vector [24, 59].

2.2 Algorithm selection

The ASP, first introduced by Rice in 1976 [49], is based on the selection of the most suitable algorithm (from a set of available algorithms) in respect of some problem class [10, 57]. The so-called *per-instance* ASP represents a special case of the ASP, according to which the most suitable algorithm is selected in respect of a specific problem instance (rather than selecting an algorithm based

[4] A random walk may be defined as a type of walk in which successive vertices (or edges) are selected randomly according to some probability distribution — it is important to note that vertices (or edges) can be revisited (or retraced) unless some condition is imposed [44].

[5] Similarity corresponds to closeness based on some distance metric, *e.g.* Euclidean distance.

on expected performance aggregated across all instances of a problem class). A fundamental assumption is that features (or characteristics) can be extracted from each problem instance and subsequently employed towards governing the selection process. A formal elucidation of the per-instance ASP follows.

Let $\mathcal{P}$ denote the set of problem instances, *i.e.* the problem space, where an individual problem instance is denoted by $p \in \mathcal{P}$. Let $\mathcal{A}$ denote the set of available algorithms, *i.e.* the algorithm space, where an individual algorithm that can be applied to solve instances within $\mathcal{P}$ is denoted by $a \in \mathcal{A}$. Let $\phi(p)$ denote the feature extraction function, *i.e.* $\phi : \mathcal{P} \mapsto \mathbb{R}^h$, according to which each problem instance $p \in \mathcal{P}$ is mapped to a feature vector $\phi(p)$ in h-dimensional real space. Furthermore, let $f_a(p)$ denote an algorithmic performance measure (*e.g.* accuracy or compute time) of algorithm a with respect to problem instance p. The performance measure represents the objective function to be maximised or minimised, depending on the task at hand. The per-instance ASP may therefore be formulated as the task of approximating the mapping function $s : \mathbb{R}^h \mapsto \mathcal{A}$. An optimal selector function, denoted by s^*, should therefore select an algorithm $a = s^*(\phi(p))$ that maximises the performance measure for any problem instance p, denoted by $f_{s(\phi(p))}$, which may be described mathematically as

$$\arg\max_s f_{s(\phi(p))}(p), \tag{1}$$

or, in a minimisation context,

$$\arg\min_s f_{s(\phi(p))}(p). \tag{2}$$

The task of approximating a high-quality mapping function, denoted by $\hat{s}$, may be contextualised as a supervised learning problem, *i.e.* the functional relationship between problem features and algorithmic performance is to be learnt algorithmically based on historical (algorithmic performance) data [1, 60]. When presented with a new (unseen) problem instance $p' \notin \mathcal{P}$, the learnt mapping function $\hat{s}$ "predicts" the best performing algorithm, denoted by $\hat{a}$, based on the extracted features, *i.e.* $\hat{a} = \hat{s}(\phi(p'))$, which is expected to maximise (or minimise) $f_{\hat{a}}(p')$.

The following terminology is adopted in this paper: An algorithm $a \in \mathcal{A}$ is called a base-learner; problem features $\phi(p)$ are called meta-features; the task of approximating an appropriate mapping function s is called meta-learning; and, finally, the approximated mapping function $\hat{s}$ is called a meta-learner. Furthermore, it is assumed that the set of problem instances $\mathcal{P}$ (*i.e.* problem space) represents the entire training set, while p' denotes a new (unseen problem instance, therefore $p' \notin \mathcal{P}$). The algorithms to be selected (*i.e.* $a \in \mathcal{A}$) may *also* include some machine (or statistical) learning models, therefore the aforementioned learning task is referred to as meta-learning, *i.e.* learning to learn [8]. The evidential utility of machine learning algorithms in respect of many use cases certainly warrant their application to meta-learning [33, 41].

3 Problem formulation

Algorithm selection for link prediction may be contextualised within the confines of the ASP. In the context of link prediction, the problem instances $p \in \mathcal{P}$ correspond to link prediction problems (*i.e.* network data sets). The available algorithms $a \in \mathcal{A}$ (*i.e.* base-learners) correspond to the various link prediction algorithms (as discussed in Section 2.1). The network meta-features obtained from $\phi(p)$ relate to network structure features, such as the number of vertices and edges, density, average clustering coefficient, modularity, average path length, average degree, and measures related to node centrality. These features ought to be informative in respect of the inherent structure of the network in order to facilitate the task of learning inferential relationships between these features and algorithmic performance. Metrics for performance evaluation $f_a(p)$ correspond to popular classification metrics, such as *area under the receiver operating curve* (AUROC), *area under the precision-recall curve* (AUPRC), and other pertinent evaluatory metrics, such as computational expenditure (*i.e.* compute time).

The different steps that constitute a per-instance link prediction ASP are proffered as follows:

1. **Meta-feature extraction**: Extract relevant network structure meta-features from each problem instance $p \in \mathcal{P}$ by means of $\phi(p)$.
2. **Base-learner performance evaluation**: For each network $p \in \mathcal{P}$, implement the selection of link prediction algorithms $a \in \mathcal{A}$, and measure the algorithmic performance achieved, *i.e.* $f_a(p)$.
3. **Meta-learning data set construction**: Aggregate the computed features $\phi(p)$ and the link prediction algorithm performance metrics $f_a(p)$ into a tabular meta-data set.
4. **Formulate meta-learning problem**: Specify the nature of the predictive task to be performed. Various formulations may be considered, such as univariate and multivariate regression as well as binary and multiclass classification.
5. **Meta-learner training**: Train the regression (or classification, depending on the adopted meta-learning formulation) meta-learning algorithm(s) on the meta-data set, during which the aim is to construct a predictive meta-learner $\hat{s}$ that can accurately predict the dependent variable(s), *i.e.* $f_a(p)$, based on the independent variables, *i.e.* $\phi(p)$, for all $p \in \mathcal{P}$.
6. **Base-learner algorithm selection**: For an unseen problem instance p', derive its meta-features by means of $\phi(p')$ and employ $\hat{a} = \hat{s}(\phi(p'))$ so as to determine the most suitable link prediction algorithm a (based on expected performance).
7. **Meta-learner performance evaluation**: Apply the selected link prediction algorithm $\hat{a}$ to p' and evaluate its performance $f_{\hat{a}}(p')$.

4 Methodology

In this section, the methodology employed towards demonstrating link prediction algorithm selection, which is predicated on the problem formulation proposed in

Section 3, is elucidated. First, the meta-learning modelling approach is presented which is followed by a discussion on the selection criteria adopted for link prediction base-learners. The meta-features included in this study are then presented, followed by a discussion on the considered network problem instances.

4.1 Meta-learning model

In this paper, the meta-learning implementation for link prediction algorithm selection, as formulated in Section 3, employs regression-based meta-learners. The meta-learners are therefore tasked with predicting some numerical algorithmic performance measure achieved by the link prediction base-learners. The predicted performance metrics can then be subsequently ranked and the base-learner corresponding to the top-ranked predicted performance is deemed the most suitable algorithm. The focus in this paper, however, involves showcasing the computational feasibility of predicting link prediction algorithmic performance. Three popular regression models are employed towards this end, namely: Linear regression [17], random forest [11], and a gradient boosting algorithm [18].

Linear regression involves abstracting the relationship between a dependent variable and one or more independent variables by means of a linear function [17]. Random forests represent an ensemble learning method that involves constructing a 'forest' of decision trees from which the mean prediction of the individual trees is employed as the final prediction [11]. Random forests have demonstrated admirable performance in respect of abstracting non-linear relationships embedded within data. Finally, gradient boosting represents an advanced ensemble technique for constructing a predictive model in a systematic (*i.e.* iterative) manner, from which a gradient-based approach may be employed towards minimising an appropriate loss function [18].

Hyperparameter tuning is performed in respect of the gradient boosting and random forest models which is performed by means of a grid search approach. The hyperparameters considered for random forest are the *number of estimators* (NoE) and the *maximum tree depth* (MTD), while the hyperparameters considered for gradient boosting also include the NoE and the *learning rate* (LR). The corresponding hyperparameter ranges for each model is presented in Table 1. A train-test split of 80/20 is employed. Towards evaluating the performance of regression-based meta-learners, two key metrics are employed, namely: *Mean squared error* (MSE) and the R^2 error.

Table 1: Hyperparameters values for the gradient boosting and random forest regression meta-learners.

Model	**Hyperparameter**	**Values**
Random forest	NoE	[100, 200, 300]
	MTD	[None, 10, 20, 30]
Gradient boosting	NoE	[100, 200, 300]
	LR	[0.01, 0.1, 0.2]

In addition to the primary analysis, this study also incorporates the application of *SHapley Additive exPlanations* (SHAP) [56] so as to determine the

(meta-)feature importance with respect to the meta-learners. SHAP is a game theory-based approach towards effectively interpreting predictive models according to which feature contributions (in respect of final predictions) are determined. The application of SHAP results in additional insight with respect to the meta-features (*i.e.* network structural characteristics) that are deemed most informative towards predicting the performance of link prediction algorithms — such insight may form the basis for improved feature engineering and subsequent algorithm development.

4.2 Base-learners

The selection of link prediction algorithms is guided by means of consensus within the literature and empirical observations in respect of the most widely recognised categories of algorithms, as discussed in Section 2.1, thereby ensuring a comprehensive representation of the prevailing approaches in the field. The following similarity-based algorithms are considered: The *common neighbours index* (CNI) [39] and the *Adamic-Adar index* (AAI) [2]. In the case of classifier-based methods, the following supervised learning algorithms are considered: Random forests [12], decision trees [54], and logistic regression [12]. Embedding-based methods include DeepWalk [46], Node2Vec [24], GCN [34], GraphSAGE [26], and GAT [61].

In Table 2, the implemented hyperparameter values and associated Python packages for each of the selected base-learners are detailed. The NetworKit [6] package was utilised for the heuristic-based link prediction methods. The numerical experiments were conducted on a MacBook Pro equipped with an Apple M1 chip and 8GB of random access memory.

Table 2: Hyperparameters and corresponding values for the classifier- and embedding-based methods. The relevant packages employed are also stated.

Method	**Hyperparameters**	**Package**
Logistic Regression	Maximum iterations = 1000	Scikit-learn [45]
Decision Tree	Criterion = 'gini', Splitter = 'best', Minimum samples split = 2	Scikit-learn [45]
Random Forest	Criterion = 'gini', NoE = 25, Maximum features = 0.2	Scikit-learn [45]
DeepWalk	Number of walks = 10, Walk length = 80, Window size = 10, Workers = 1	Gensim [48]
Node2Vec	Number of walks = 10, Walk length = 80, Window size = 10, Workers = 1, p = 1, q = 1	node2vec [24]
GCN	Input channels = Output channels = 128, Number of layers = 2, LR = 0.001	PyTorch Geometric [16]
GraphSAGE	Input channels = Output channels = 128, Number of layers = 2, LR = 0.001	PyTorch Geometric [16]
GAT	Input channels = Output channels = 128, Number of layers = 2, LR = 0.001	PyTorch Geometric [16]

The aforementioned selection of algorithms is guided by both diversity (in respect of their fundamental working) and their differing empirical performance dynamics, as reported in various studies [21, 29, 39, 65]. Furthermore, the (link) prediction task carried out by the base-learners is contextualised as a binary classification problem — a node-pair is classified based on link presence (*i.e.* positive class) or link absence (*i.e.* negative class). A train-test ratio of 90/10 is employed.

Classification threshold curves are deemed suitably robust for evaluating algorithmic performance [40]. The *receiver operating characteristic* curve (ROC) is a popular threshold curve which plots the *true positive rate* (*i.e.* recall) against the *false positive rate* at different threshold values — the AUROC quantifies a predictive model's performance in a unary manner and is employed in this paper to measure the algorithmic performance of the base-learners.

4.3 Meta-features

Different computational measures have been reported in the literature towards quantifying topological features of a network (*i.e.* the meta-features) which provide insight into its structural properties, upon which further analyses can be based. An abundance of network structural measures have been reported, each of which can abstract various facets of a network's structure. The network structure measures that are considered in this paper can be categorised according to *general statistics, centrality, community, clustering, connectivity* and *degree*-based measures[6].

General statistics General statistics of a network include its *size* and *order* which are informative indicators of its scale and complexity. The network size is determined by the total number of vertices, while the network order corresponds to the total number of existing edges. Additionally, the network's *density* which is the ratio between the existing edges and the maximum possible edges, offers insight into the extent to which a network is interconnected [30].

Centrality Centrality measures involve the identification of 'important' vertices within a network. *Betweenness* centrality quantifies a vertex's importance based on the number of shortest paths[7] passing through it, and *closeness* centrality is based on the proximity of a vertex to all other vertices within the network, which employs the shortest path as a measure [44, 53].

[6] It is important to note that some of these measures provide (based on their original formulation) contextual insight into individual nodes, whilst other measures synthesise information in respect of the overall network. Node-specific measures can be aggregated in respect of the entire network in order to characterise the network as a whole.

[7] A path is a sequence of vertices connected by edges, where each edge is included once and no vertex is repeated [44].

Community *Modularity* quantifies the degree to which a network can be partitioned into communities by analysing the proportion of edges within and between these communities [43]. A community refers to a subset of vertices within a graph that are more densely connected to each other when compared with other subset of vertices. These communities often represent groups of vertices sharing similar properties within the overall structure of the graph [44]. The number of communities in a network refers to the count of denser, more connected subgraphs within the larger graph [47].

Clustering The global *clustering coefficient* is a key metric in network analysis quantifying the presence of *triangles*[8] in a graph [44]. Additionally, *transitivity* measures the extent of clustering by calculating the ratio of possible triangles to the number of *triads*[9] in the network [25, 55].

Connectivity Connectivity metrics include *average path length* which indicates the average shortest distance between vertex pairs, and a network's *diameter*, representing the maximum shortest path between any two vertices. These metrics are essential towards understanding the extent to which nodes may be traversed within the network. *Global efficiency* is another connectivity-based measure which indicates the network's overall efficiency in information exchange and is based on the average inverse distance between vertex pairs [30, 52].

Degree Important metrics include the average degree[10] of vertices within the network and the variance of the degree distribution which indicates the diversity of vertex connectivity. Degree *assortativity* measures the tendency of vertices to connect with other vertices that have similar degrees [9, 44].

4.4 Problem instances

Publicly available network data sets are considered in this study from which undirected and unweighted graphs are induced. This delimitation may be ascribed to the ubiquity of such network problems and due to the accessibility of such data sets. The reader is referred to [36, 38, 51] for a comprehensive collection of network data sets. A total of 220 problem instances are considered. These network data sets are expressed by means of so-called edge lists. Additionally, the data sets stem from a broad range of domains (such as social, biological, infrastructure, and citation networks) of which there are eighteen in total — this facilitates a comprehensive analysis. Towards further enhancing the diversity of the data sets, synthetic data sets are generated by means of various established

[8] A triangle is a set of three vertices that are mutually connected by edges, *i.e.* each vertex is connected to the remaining two vertices [44].
[9] A triad refers to two edges with a shared vertex [25].
[10] The degree of a vertex is the number of edges that are incident to the vertex [30].

approaches, namely: Erdős-Rényi [14], Barabási-Albert [5], Watts-Strogatz [63], and the forest-fire model [13].

Aggregation statistics relating to the network meta-features of the problem instances are presented in Table 3.

Table 3: Aggregation statistics of network meta-features in respect of the suite of network problem instances.

Meta-feature	**Mean**	**Variance**	**Standard deviation**	**Minimum**	**Maximum**
Network size	18 804.059	9.138×10^8	30 229.456	39	180 000
Network order	2 959.100	1.767×10^7	4 203.062	21	26 588
Density	0.059	0.015	0.121	0.000141	0.718
Clustering coefficient	0.254	0.052	0.229	0	0.808
Average path length	4.942	18.336	4.282	1.282	35.349
Average degree	19.439	1 040.995	32.264	1.734	178.880
Variance of degree distribution	934.399	4.921×10^6	2 218.489	0.350	12 722.375
Degree assortativity	−0.086	0.044	0.209	−0.761	0.650
Global efficiency	0.323	0.027	0.163	0.040	0.859
Diameter	12.886	148.753	12.196	2	99
Transitivity	0.213	0.044	0.210	0	0.792
Modularity	0.577	0.059	0.243	0.0641	0.961
Number of communities	27.941	1 724.029	41.521	2	291
Average betweenness centrality	0.011	0.001	0.026	0.000109	0.178
Average closeness centrality	0.289	0.023	0.153	0.0138	0.789

5 Results

In this section, the findings stemming from the computational analyses carried out are presented. First, a discussion is presented on the algorithmic performance achieved by the base-learners, which is followed by a discussion on the results achieved by the meta-learners. Finally, a discussion is presented on the feature importance results from which insight is gleaned into the relationships between network characteristics and link prediction algorithms.

5.1 Base-learners

A comparative analysis of the considered base-learners, as depicted in Figure 2, reveals distinct performance variations across multiple data sets. The logistic regression and random forest classifiers showcase superior performance achieving

the largest mean AUROC scores, while GraphSAGE achieves the smallest mean AUROC scores. Notably, similarity-based methods such as the CNI and AAI demonstrate moderate performance indicative of their conditional effectiveness. Furthermore, the variance showcased by the different algorithmic approaches differs rather markedly indicative of underlying complexities.

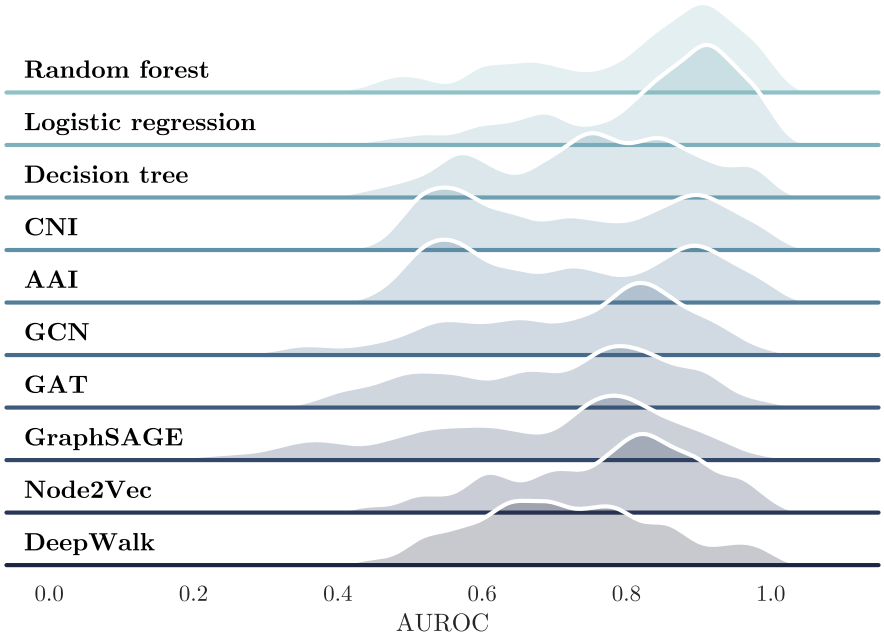


Fig. 2: Distribution of AUROC scores in respect of the different link prediction base-learners. The ridge line plot illustrates the variability in AUROC scores aggregated with respect to the 220 data sets.

These results empirically exemplify the NFL theorem in the context of link prediction, underscoring the necessity for a systematic, data-driven approach towards informed algorithm selection. The varying performance achieved by the base-learners across network data sets emphasises the importance of assimilating network-specific characteristics and their relationship with algorithmic performance. This aligns with the proposition of a regression-based meta-learning approach towards informed link prediction algorithm selection.

5.2 Meta-learning

Hyperparameter tuning is conducted in respect of the meta-learning models by means of a grid search approach. The resulting hyperparameter values identified

are presented in Table 4. It should be noted that meta-learner performance demonstrated general insensitivity to hyperparameter values.

Table 4: High-quality hyperparameters identified by means of grid search in respect of the meta-learning models.

Link prediction algorithm	**Gradient boosting**		**Random forest**	
	LR	**NoE**	**MTD**	**NoE**
CNI	0.1	300	30	100
AAI	0.1	300	20	200
Logistic regression	0.1	100	20	200
Decision tree	0.1	300	20	300
Random forest	0.1	300	20	300
DeepWalk	0.2	300	30	200
Node2Vec	0.1	200	10	200
GCN	0.01	300	None	100
GraphSAGE	0.01	300	30	100
GAT	0.01	300	None	200

After adopting the high-quality hyperparameter values, the three meta-learners were evaluated in respect of the various base-learners and problem instances. The MSE achieved by each meta-learner is presented in Figure 3. It may be observed that the gradient boosting model consistently achieves superior performance, showcasing robustness in respect of algorithm selection across various base-learners. Linear regression, on the other hand, performs markedly inferior suggesting computational inadequacy when attempting to approximate the possibly non-linear relationship between network characteristics and the performance of link prediction algorithms. The random forest algorithm performs admirably when compared with its gradient boosting counterpart. In some cases, it performs best overall. The R^2 scores, presented in Figure 4, showcase a similar pattern with respect to the efficacy of different meta-learning models.

Variability in respect of both MSE and R^2 scores across the different base-learners is indicative of the disparate nature of the relationships between network features and algorithmic performance. The underlying relationship is evidently less complex to abstract, as demonstrated in the case of the two similarity-based heuristics (*i.e.* CNI and AAI) and the more simplistic embedding-based methods (*i.e.* DeepWalk and Node2Vec). Conversely, more complex approaches, such as the machine learning based classifiers and the graph neural networks, present a slightly more pronounced challenge when attempting to predict algorithmic performance.

In summary, the meta-learner performance analysis, encompassing MSE and R^2 metrics, reveals a stratified landscape in respect of predictive accuracy. Gradient boosting, achieving consistently smaller error scores and larger R^2 scores, emerge as the preferred choice due to its overall robustness. Linear regression, which achieved larger error scores, is deemed markedly less effective in capturing

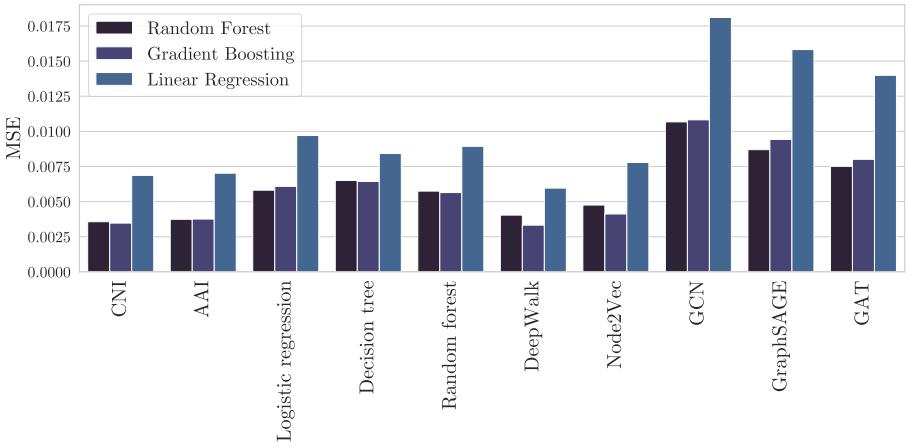


Fig. 3: MSE achieved by the regression-based meta-learning models in respect of the different base-learners.

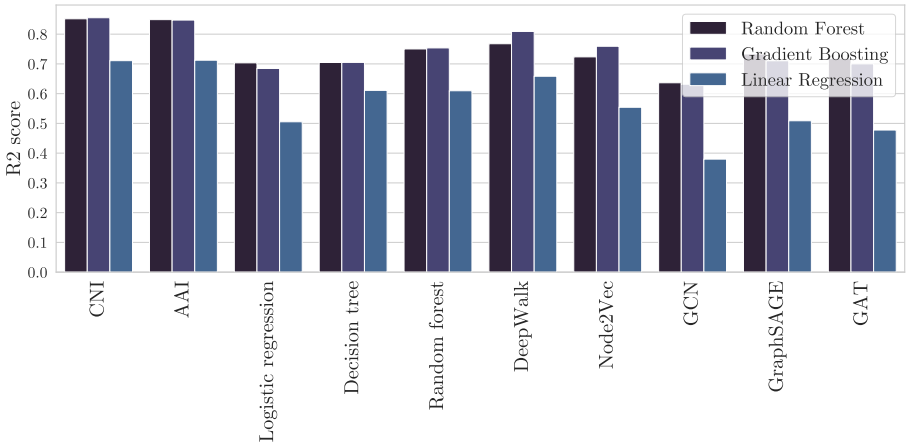


Fig. 4: R^2 achieved by the regression-based meta-learning models in respect of the different base-learners.

the relationship between link prediction algorithmic performance and network characteristics, attributable to potentially non-linear relationships. Furthermore, the random forest model may be deemed a high-quality counterpart to gradient boosting, as it achieves superior performance in respect of some instances. These results indicate the computational utility of employing ensembling-based predictive models. Finally, it may be concluded that algorithmic performance may be systematically and reliably predicted by means of appropriate meta-learners. Such findings are, to the best of our knowledge, novel.

5.3 Feature importance

Findings stemming from a SHAP analysis carried out are now discussed. The results are presented in Table 5. Notable variability may be observed in respect of feature importance across different link prediction algorithms which underscores the nuanced impact of network characteristics on algorithmic performance. Notably, features such as the variance of the degree, clustering coefficient, and transivity consistently rank as top features with respect to a majority of the algorithms indicative of their informative nature towards abstracting their impact on algorithmic performance. This aligns with the intuitive understanding that closely knit communities often play a critical role in link formation [15].

Table 5: Feature importance rankings derived from a SHAP analysis in respect of the various base-learners and meta-features. A ranking of 1 indicates highest importance, whereas 15 indicates the lowest importance.

Meta-feature	**CNI**	**AAI**	**Logistic regression**	**Decision tree**	**Random forest**	**DeepWalk**	**Node2Vec**	**GCN**	**GraphSAGE**	**GAT**	**Average rank**
Variance of degree distribution	2	2	1	1	1	1	1	3	3	5	**2.0**
Clustering coefficient	1	1	3	3	2	7	3	1	2	1	**2.4**
Transitivity	4	5	5	6	7	3	4	4	4	3	**4.5**
Average degree	3	3	11	12	13	4	10	2	1	2	**6.1**
Degree assortativity	10	10	6	5	5	2	2	9	7	8	**6.4**
Modularity	8	8	2	4	3	13	7	5	8	7	**6.5**
Diameter	13	13	4	8	4	10	6	7	6	4	**7.5**
Average betweenness centrality	9	9	7	2	6	9	9	11	5	13	**8.0**
Network order	5	4	9	7	8	5	5	13	13	14	**8.3**
Network size	7	7	13	13	12	6	8	8	9	12	**9.5**
Number of communities	12	12	10	9	9	8	13	6	11	15	**10.5**
Density	6	6	14	10	14	15	15	10	12	9	**11.1**
Average path length	14	14	12	14	11	12	11	12	10	6	**11.6**
Global efficiency	11	11	8	11	10	14	12	14	14	11	**11.6**
Average closeness centrality	15	15	15	15	15	11	14	15	15	10	**14.0**

Conversely, features such as average closeness centrality, global efficiency, and average path length exhibit lower importance rankings thereby indicating their limited influence on algorithmic efficacy. This may be attributed to their global nature which might not be as informative in respect of certain algorithms relying on more localised inferential patterns in order to predict missing links (*e.g.* CNI and AAI).

The variance of degree distribution emerges as a dominant feature. This highlights the significance of node degree variability in predicting link formation and potentially reflecting the heterogeneity of connections in networks. Interestingly, network order showcases notably varied importance in respect of the different base-learners. Although this measure is deemed important in some contexts (*e.g.* AAI, DeepWalk, and Node2Vec), it is markedly less important in the case of other contexts (*e.g.* GCN, GraphSAGE and GAT). The average degree of vertices demonstrates the most significant variability in respect of importance — it is deemed markedly important in respect of certain approaches, while being inconsequential in others.

These findings further substantiate the assertion that different network manifestations are suited to different algorithmic approaches — the NFL theorem may certainly be invoked in this context. Furthermore, the utility of a systematic, data-driven approach towards algorithm selection is demonstrated.

6 Conclusion

In this paper, a novel application of meta-learning algorithm selection was investigated in the context of link prediction. The utility of a regression-based meta-learning approach was demonstrated, according to which link prediction algorithms may be systematically selected based on network characteristics and historical performance data. An in-depth analysis was carried out in respect of multiple regression-based meta-learners, namely linear regression, random forest, and gradient boosting. It was reported that gradient boosting and random forest generally display predictive performance superiority, as indicated by consistently smaller MSE scores and larger R^2 scores. The main finding from this analysis relates to the computational feasibility of predicting base-learner performance (in respect of AUROC). Consequently, appropriately trained meta-learners may therefore be employed towards predicting algorithmic performance in respect of newly presented problem instances, thereby streamlining typically cumbersome or rudimentary approaches towards algorithm selection.

A feature importance analysis based on SHAP values was also carried out, from which the impact (and extent thereof) of certain network characteristics on the performance of link prediction algorithms was inferred. Transitivity, clustering coefficient, and variance of degree emerged as informative features. The varying importance of features such as closeness centrality and network efficiency, however, suggests that some global network properties may not consistently enhance predictive performance across different algorithms. This variation highlights the importance of thoughtful feature selection and algorithm application based on the specific characteristics and structural properties of the network being analysed.

Future work could expand the scope of this research by exploring other meta-learning strategies (such as a classification-based approach), incorporating a broader range of network features, and applying this approach to other network problem instances. Furthermore, the development of an automated (com-

puterised) tool for link prediction algorithm selection could significantly streamline additional analyses and enable other researchers to contribute towards this promising direction by expanding upon the numerical database constructed thus far.

In conclusion, various contributions are proffered by work carried out in this project, the first of which relates to the furtherance of the domain of link prediction by formalising (mathematically) the algorithm selection problem in link prediction. Another contribution relates to the proposal of a comprehensive suite of network data sets for evaluation, from which important insights into the relationships between network features and algorithmic performance may be gleaned. The proposed meta-learning approach facilitates a structured, standardised, and systematic approach towards enhancing link prediction. Another important contribution relates to insight gained into the importance of certain meta-features with respect to different base-learns — a preliminary basis may therefore be inferred upon which algorithmic design and improvement may be carried out.

References

1. Abdulrahman, S.M., Brazdil, P., Zainon, W.M.N.W., Adamu, A.: Simplifying the algorithm selection using reduction of rankings of classification algorithms. In: 8th International Conference on Software and Computer Applications. pp. 140–148. Penang (2019). https://doi.org/10.1145/3316615.3316674
2. Adamic, L.A., Adar, E.: Friends and neighbours on the web. Social Networks **25**(3), 211–230 (2003). https://doi.org/10.1016/S0378-8733(03)00009-1
3. Aiello, L.M., Barrat, A., Schifanella, R., Cattuto, C., Markines, B., Menczer, F.: Friendship prediction and homophily in social media. Transactions on the Web **6**(2), 1–33 (2012). https://doi.org/10.1145/2180861.2180866
4. Akcora, C.G., Carminati, B., Ferrari, E.: Network and profile based measures for user similarities on social networks. In: 12th IEEE International Conference on Information Reuse & Integration. pp. 292–298. Las Vegas (NV) (2011)
5. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. Reviews of Modern Physics **74**(47), 47–97 (2002). https://doi.org/10.1103/RevModPhys.74.47
6. Angriman, E., van der Grinten, A., Hamann, M., Meyerhenke, H., Penschuck, M.: Algorithms for large-scale network analysis and the networkit toolkit. In: Bast, H., Korzen, C., Meyer, U., Penschuck, M. (eds.) Algorithms for Big Data, pp. 3–20. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-21534-6_1
7. Barabási, A.L.: Network science. Cambridge University Press, Glasgow (2016)
8. Bensusan, H., Kalousis, A.: Estimating the predictive accuracy of a classifier. In: 12th European Conference on Machine Learning. pp. 25–36. Springer, Freiburg (2001)
9. Borgatti, S.P., Everett, M.G., Johnson, J.C.: Analyzing social networks. Sage, London (2018)
10. Brazdil, P., Carrier, C.G., Soares, C., Vilalta, R.: Metalearning: Applications to data mining. Springer Science & Business Media, Berlin (2008)
11. Breiman, L.: Random forests. Machine Learning **45**, 5–32 (2001). https://doi.org/10.1023/A:1010933404324

12. Breiman, L.: Classification and regression trees. Routledge, New York (NY) (2017). https://doi.org/10.1201/9781315139470
13. Drossel, B., Schwabl, F.: Self-organized critical forest-fire model. Physical Review Letters **69**(11), 1629–1632 (1992). https://doi.org/10.1103/physrevlett.69.1629
14. Erdős, P., Rényi, A.: On random graphs. Publicationes Mathematicae **6**(3-4), 290–297 (1959). https://doi.org/10.5486/PMD.1959.6.3-4.12
15. Feng, X., Zhao, J., Xu, K.: Link prediction in complex networks: A clustering perspective. The European Physical Journal B **85**(3), 1–9 (2012). https://doi.org/10.1140/epjb/e2011-20207-x
16. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: 7th International Conference on Learning Representations. pp. 1–8. New Orleans (LA) (2019). https://doi.org/10.48550/arXiv.1903.02428
17. Freedman, D.A.: Statistical models: Theory and practice. Cambridge University Press, New York (NY) (2009)
18. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. Annals of Statistics **29**(5), 1189–1232 (2001). https://doi.org/10.1002/spe.4380211102
19. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. Software: Practice and Experience **21**(11), 1129–1164 (1991)
20. Gao, F., Musial, K., Cooper, C., Tsoka, S.: Link prediction methods and their accuracy for different social networks and network metrics. Scientific Programming **2015**, 1–13 (2015). https://doi.org/10.1155/2015/172879
21. Ghasemian, A., Hosseinmardi, H., Galstyan, A., Airoldi, E.M., Clauset, A.: Stacking models for nearly optimal link prediction in complex networks. Proceedings of the National Academy of Sciences **117**(38), 23393–23400 (2020). https://doi.org/10.1073/pnas.1914950117
22. Giraud-Carrier, C., Vilalta, R., Brazdil, P.: Introduction to the special issue on meta-learning. Machine Learning **54**, 187–193 (2004)
23. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences **99**(12), 7821–7826 (2002). https://doi.org/10.1073/pnas.122653799
24. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: 22nd International Conference on Knowledge Discovery and Data Mining. pp. 855–864. San Francisco (CA) (2016). https://doi.org/10.1145/2939672.2939754
25. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using networkx. In: Varoquaux, G., Vaught, T., Millman, J. (eds.) 7th Python in Science Conference. pp. 11–15. Pasadena (CA) (2008)
26. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: 31st Conference on Neural Information Processing Systems. pp. 1025–1035. Long Beach (CA) (2017)
27. Hamilton, W.L.: Graph representation learning. Morgan & Claypool Publishers, Cham (2020)
28. Hasan, M.A., Chaoji, V., Salem, S., Zaki, M.: Link prediction using supervised learning. In: 30th Workshop on Link Analysis, Counter-terrorism and Security. pp. 798–805. Newport Beach (CA) (2006)
29. Hasan, M.A., Zaki, M.J.: A survey of link prediction in social networks. In: Aggarwal, C.C. (ed.) Social Network Data Analytics, pp. 243–275. Springer, Boston (MA) (2011). https://doi.org/10.1007/978-1-4419-8462-3_9
30. Henning, M.A., van Vuuren, J.H.: Graph and network theory: An applied approach using Mathematica. Springer, Cham (2022)
31. Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X.: Applied logistic regression. John Wiley & Sons, Hoboken (NJ), 3 edn. (2013)

32. Kalousis, A., Hilario, M.: Representational issues in meta-learning. In: 20th International Conference on Machine Learning. pp. 313–320. Washington (DC) (2003)
33. Khan, I., Zhang, X., Rehman, M., Ali, R.: A literature survey and empirical study of meta-learning for classifier selection. IEEE Access **8**, 10262–10281 (2020). https://doi.org/10.1109/ACCESS.2020.2964726
34. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations. pp. 1–14. Toulon (2017). https://doi.org/10.48550/arXiv.1609.02907
35. Kumar, A., Singh, S.S., Singh, K., Biswas, B.: Link prediction techniques, applications, and performance: A survey (2020), physica A: Statistical Mechanics and its Applications, **553**, Manuscript 124289
36. Kunegis, J.: Konect: The koblenz network collection. In: 22nd International Conference on World Wide Web. pp. 1343–1350. Rio de Janeiro (2013)
37. Lei, C., Ruan, J.: A novel link prediction algorithm for reconstructing protein-protein interaction networks by topological similarity. Bioinformatics **29**(3), 355–364 (2013). https://doi.org/10.1093/bioinformatics/bts688
38. Leskovec, J., Krevl, A.: Snap datasets: Large network dataset collection (12 2023), http://snap.stanford.edu/data
39. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: 12th International Conference Information and Knowledge Management. pp. 556–559. New Orleans (LA) (2003). https://doi.org/10.1093/bioinformatics/bts688
40. Lichtenwalter, R., Lussier, J., Chawla, N.: New perspectives and methods in link prediction. In: 16th International Conference on Knowledge Discovery and Data Mining. pp. 243–252. Washington (DC) (2010). https://doi.org/10.1145/1835804.1835837
41. Lindauer, M., Hoos, H.H., Hutter, F., Schaub, T.: Autofolio: An automatically configured algorithm selector. Journal of Artificial Intelligence Research **53**, 745–778 (2015). https://doi.org/10.1613/jair.4726
42. Lü, L., Pan, L., Zhou, T., Zhang, Y.C., Stanley, H.E.: Toward link predictability of complex networks. Proceedings of the National Academy of Sciences **112**(8), 2325–2330 (2015). https://doi.org/10.1073/pnas.1424644112
43. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences **103**(23), 8577–8582 (2006). https://doi.org/10.1073/pnas.0601602103
44. Newman, M.E.J.: Networks. Oxford University Press, New York (NY), 2 edn. (2018)
45. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
46. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: 20th International Conference on Knowledge Discovery and Data Mining. pp. 701–710. New York (NY) (2014). https://doi.org/10.1145/2623330.2623732
47. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proceedings of the National Academy of Sciences **101**(9), 2658–2663 (2004). https://doi.org/10.1073/pnas.0400054101
48. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Workshop on New Challenges for NLP Frameworks. pp. 45–50. Valletta (2010)
49. Rice, J.R.: The algorithm selection problem. In: Rubinoff, M., Yovits, M.C. (eds.) Advances in Computers, vol. 15, pp. 65–118. Academic Press, New York (NY) (1976)

50. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review **65**(6), 386 (1958)
51. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization (12 2023), https://networkrepository.com
52. Rubinov, M., Sporns, O.: Complex network measures of brain connectivity: Uses and interpretations. Neuroimage **52**(3), 1059–1069 (2010). https://doi.org/10.1016/j.neuroimage.2009.10.003
53. Sabidussi, G.: The centrality index of a graph. Psychometrika **31**(4), 581–603 (1966)
54. Safavian, S.R., Landgrebe, D.: A survey of decision tree classifier methodology. Transactions on Systems, Man, and Cybernetics **21**(3), 660–674 (1991)
55. Schank, T., Wagner, D.: Approximating clustering coefficient and transitivity. Journal of Graph Algorithms and Applications **9**(2), 265–275 (2005). https://doi.org/10.5445/IR/1000001239
56. Shapley, L.S.: A value for n-person games, vol. 2, pp. 307–317. Princeton University Press, Princeton (NJ) (1953)
57. Smith-Miles, K.A.: Cross-disciplinary perspectives on meta-learning for algorithm selection. Computing Surveys **41**(1), 1–25 (2009). https://doi.org/10.1145/1456650.1456656
58. Su, X., Yan, X., Tsai, C.L.: Linear regression. Wiley Interdisciplinary Reviews: Computational Statistics **4**(3), 275–294 (2012). https://doi.org/10.1002/wics.1198
59. Tsitsulin, A., Mottin, D., Karras, P., Müller, E.: Verse: Versatile graph embeddings from similarity measures. In: 27th International World Wide Web Conference. pp. 539–548. Lyon (2018). https://doi.org/10.1145/3178876.3186120
60. Vanschoren, J.: Meta-learning. In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds.) Automated machine learning: Methods, systems, challenges, pp. 35–61. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-05318-5_2
61. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: 6th International Conference on Learning Representations. pp. 1–12. Vancouver (2018). https://doi.org/10.48550/arXiv.1710.10903
62. Wang, P., Xu, B., Wu, Y., Zhou, X.: Link prediction in social networks: The state-of-the-art. Science China: Information Science **58**(1), 1–38 (2015). https://doi.org/10.48550/arXiv.1411.5118
63. Watts, D.J., Strogatz, S.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (1998)
64. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. Transactions on Evolutionary Computation **1**(1), 67–82 (1997). https://doi.org/10.1109/4235.585893
65. Wu, H., Song, C., Ge, Y., Ge, T.: Link prediction on complex networks: An experimental survey. Data Science and Engineering **7**, 253–278 (2022). https://doi.org/10.1007/s41019-022-00188-2
66. Zhang, D., Yin, J., Zhu, X., Zhang, C.: Network representation learning: A survey. Transactions on Big Data **6**(1), 3–28 (2018). https://doi.org/10.1109/TBDATA.2018.2850013
67. Zhang, M., Chen, Y.: Link prediction based on graph neural networks. In: 32nd International Conference on Neural Information Processing Systems. pp. 5171–5181. Montréal (2018)