Inductive link prediction boosts data availability and enables cross-community link prediction in ecological networks

Barry Biton [1], Rami Puzis [2], and Shai Pilosof [1,3,*]

[1] *Department of Life Sciences, Ben-Gurion University of the Negev, Beer-Sheva, Israel*
[2] *Department of Systems Information and Software Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel*
[3] *The Goldman Sonnenfeldt School of Sustainability and Climate Change, Ben-Gurion University of the Negev, Be'er Sheva, Israel*
[*] *Corresponding author: pilos@bgu.ac.il*

August 1, 2024

Abstract

Predicting species interactions within ecological networks is vital for understanding ecosystem functioning and the response of communities to changing environments. Traditional link prediction models often fall short due to sparse and incomplete data and are limited to single networks. Here, we present a novel approach using inductive link prediction (ILP), which leverages structural similarities across diverse ecological networks. Our model pools data across communities, and uses transfer learning to enable prediction within and between different ecological communities. We applied our model to 538 networks across four community types: plant-seed disperser, plant-pollinator, host-parasite, and plant-herbivore. ILP outperforms non-ILP models, particularly in host-parasite and plant-seed disperser networks. However, the efficacy of cross-community predictions varies, with plant-pollinator networks consistently under-performing as train and test sets. Moreover, we developed the first method to computationally estimate the limits of link prediction given a certain proportion of missing links, in which ILP performs better than a non-ILP model. This study underscores the potential of ILP to generalize link prediction across different ecological contexts.

Introduction

Ecosystem services, such as pollination, are fundamental for human societies and result from interactions between many organisms in ecological communities [1]. For example, crop pollination highly depends on the interactions between wild bees and non-crop flowers [2]. These dependencies can be described mathematically using networks in which nodes represent species and links represent their interactions. Complete sampling and representation of species interaction networks would enhance our understanding of the complex indirect dependencies between species. However, given the immense resources required to attain and validate each interaction [3–5], it is clear that ecological networks are consistently under-sampled, hampering our ability to analyze their true structure. One way to address the under-sampling problem is to predict the most probable, yet unconfirmed links [6–12]. Moreover, predicting interactions helps us understand how ecological networks will respond to anthropogenic changes and environmental shifts. For example, predicting disease hosts and interactions between local and invasive species [11]. However, link prediction in ecological networks is highly challenging because these networks are small (insufficient data) and sparse (many unobserved links and very few observed ones).

Ecological link prediction models can use information on species traits [9,13] and phylogeny [12], but these are often difficult to obtain or biased because some taxonomic groups are far more studied than others. Instead, it is possible to rely on the topology of the known part of the network to predict the unknown [7,8]. For instance, a well-connected species will likely have another link. Such approaches that predict missing links based on known properties within the same network are called *transductive* link prediction [14]. While this is the primary approach used in ecology so far [7,8], its performance is hindered in networks where only few links are known or where some parts are known much better than the others [15,16], two issues prevalent in ecological networks.

To overcome these issues, we take an approach called inductive link prediction (ILP), in which links in one network are predicted by learning the structure of others [16]. ILP harnesses the principle of universality [17–20], reflected in ecological networks by cross-system topological similarities [21,22]. The structure of ecological networks is shaped by multiple ecological and evolutionary processes, including spatiotemporal distributions, evolutionary history, and neutral processes [23–25]. Despite the very different nature and idiosyncrasy of ecological systems (e.g., mutualistic vs antagonistic networks), these processes generate non-random recurring patterns commonly observed in different types of networks [26–28]. For example, both antagonistic and mutualistic ecological networks typically exhibit a heavy-tail degree distribution whereby most nodes have few links (low degree), and a few nodes have many links (high degree) [29–31]. At the mesoscale level, structures such as nestedness and modularity have been detected in host-parasite, plant-pollinator, and plant-seed disperser networks [21,32–35].

Cross-system similarities in macroscopic patterns provide an opportunity to increase training data by using multiple networks within an ILP framework [36,37]. In addition, it enables transfer learning, where a model trained on data from one domain is applied to predict outcomes in another. The idea of using transfer learning in ecology has been recently proposed [38], but to the best of our knowledge, in ecology, only Caron et al. [13] used transfer learning to predict links in food webs in one area (e.g., Europe) based on knowledge in another (e.g., Serengetti). Using trait-based predictions, they found that pairwise interactions are better predicted using a model trained on the same food web than with models trained on other food webs. However, they did not pool data from different food webs for training.

Given the structural similarities between networks from different ecological communities, we hypothesize that ILP is more effective than transductive link prediction for predicting links in ecological networks. Further evidence for the plausibility of this hypothesis comes from two recent studies that showed that it is challenging to identify the type of network (e.g., plant-pollinator, host-parasite) based solely on its structure [22,39]. This is because variation in network structure is similar within and between different types of ecological networks. On the one hand, this observation means that it would be challenging to predict the ecological community of a network based on its structure [22]. On the other hand, the structural similarity of different community types can be utilized for link prediction by enlarging the training dataset. We tested this hypothesis using an ILP model we developed, which further enables cross-community prediction. We find that our model outperforms transductive link prediction models in predicting links, but that prediction accuracy varies by community type. Specifically, plant-pollinator networks weaken cross-community predictions while host-parasite networks enhance them.

Results

Data

We used the data set compiled by [22] (also later used by [39]). We used networks with ≥ 25 species and with connectance ≥ 0.1 (Table S1). This data set includes 205 plant-seed disperser networks (PSD), 217 plant-pollinator networks (PP), 84 host-parasite networks (HP), and 32 plant-herbivore networks (PH) (538 networks in total). A potential limitation of the data set could be the non-equal number of networks per community. To test for this, we included the type of community as a feature in the model, and it was not an important feature in the prediction. In addition, communities represented by more networks were not necessarily better train or test sets (see results below), indicating that the number of networks is not a limitation. As previously shown for this data set [22,39], variation in network structure was not higher between than within networks (Fig. S1).

Overview of the link prediction pipeline

We developed a pipeline to evaluate the performance of ILP models on multi-network data sets using nested cross-validation (Fig. 1). We split the data into training, validation and test sets, ensuring that instances (links) from the same network are always together in the set. We predicted links based on network properties, which we used as features in the model (feature extraction). In the inner loop, we used the training and validation sets to tune hyperparameters and select the best model to perform link prediction on the test set. We performed 5-fold cross-validation in the outer loop. We used popular machine learning models: logistic regression, random forest, and XGboost. The results did not qualitatively change between models (Supplementary note), and we present results for Random Forest. We considered the significant imbalance between the number of observed and unobserved interactions [40].

Inductive link prediction outperforms prediction with a single network

To test the hypothesis that collating data across communities would improve link prediction, we compared our model performance to three models that train and test on a single network at a time (transductive models). The first two were stochastic block model (SBM) and connectance, which were

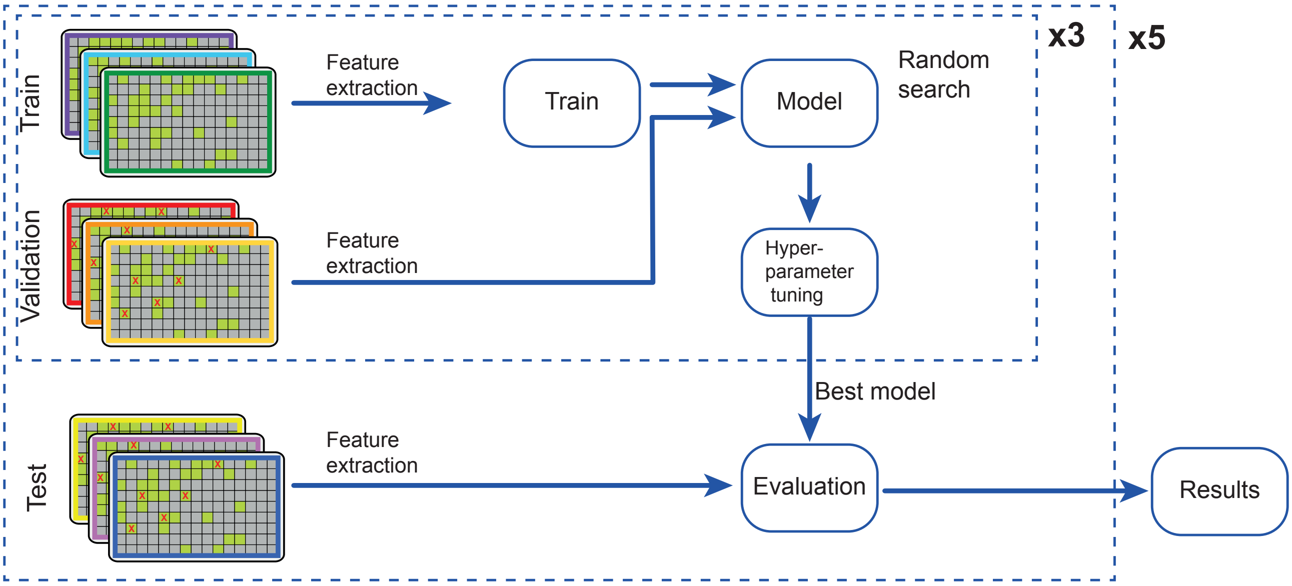

Fig. 1: Pipeline overview. (A) We used nested cross-validation, with 3 and 5 folds in the inner and outer loops, respectively. We split the data into train, validation, and test sets and calculated network properties (feature extraction). The validation and test sets contained subsampled links (red x's). We used the inner loop to tune the hyperparameters (using random search) and selected the best model to predict links in the test set in the outer loop. Model evaluation was based on a confusion matrix with TP, TN, FP, and FN values gathered from the outer loop's five folds. When splitting the data, we ensured that each network was included entirely in the set (i.e., we split the data by networks, not by links) and that each network appeared at least once in the test set. See details in Methods.

previously published and are not based on machine learning [8]). The third was a transductive machine learning model we developed transductive link prediction (TLP). Overall, the machine learning ILP and TLP models outperformed SBM and connectance (Fig. 2). ILP predicts 1's better than TLP as it has a higher recall but similar precision. However, this comes at the expense of its ability to recover 0's (lower specificity). Nevertheless, in the overall balance, the ILP model outperforms TLP (higher BA).

We further evaluated the ILP model using the ROC curve across thresholds. The prediction was highly accurate when using the full data set or per community (Fig. 3A). Because imbalance restricts the interpretation of ROC curves, we also used the precision-recall (PR) curve. The PR tradeoff indicates the ability of the model to retrieve links (recall) while minimizing false positives (precision). The PR AUC was higher than the random expectation for all communities (Fig. 3B). The AUC of both curves was similar to that of the TLP model. (Fig. 2). These results did not vary quantitatively when prediction was performed within each community type (Fig. S2).

The most influential network features were at the node level, including node degree, the PDI index of specificity [41], and two centrality indices. The only link-level feature was preferential attachment (the multiplication of the two interacting species' degrees) (Fig. S3).

Links are predicted better in particular community types

The ROC and PR curves show that predictive ability varies by community type, with the best predictions performed in host-parasite networks, followed by plant-seed dispersers (Fig. 3). Further analyzing the evaluation metrics separately for each ecological community reveals more nuanced differences (Fig. 4), whereby there are significant differences in prediction metrics between communities

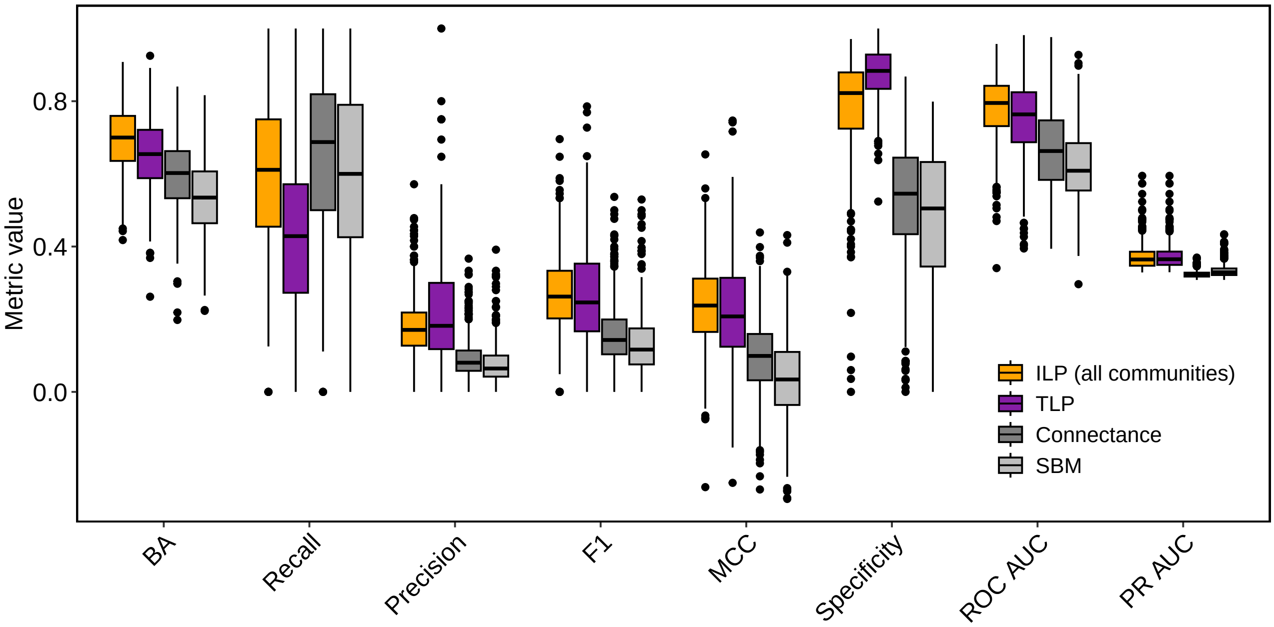

Fig. 2: Comparison of model performance. The inductive link prediction model was trained and tested on networks from all communities. We compared ILP to two transductive previously published models (stochastic block model and connectance; [8]) and one transductive machine learning model that we developed (TLP). Each boxplot is a distribution of an evaluation metric using a 0.5 classification threshold, bedsides the ROC AUC and PR AUC. Each data point is a network and boxplots contain networks from all the five outer folds. BA is balanced accuracy. See definitions of evaluation metrics in the Methods.

(Kruskal-Wallis test, Table S2, Fig. 4). The ability of our model to retrieve TP links (recall) was higher in host-parasite and plant-seed disperser communities compared to plant-pollinator and plant-herbivore communities. However, correctly predicting negative links (specificity) is lower in these two communities (Dunn post-hoc tests; Table S3).

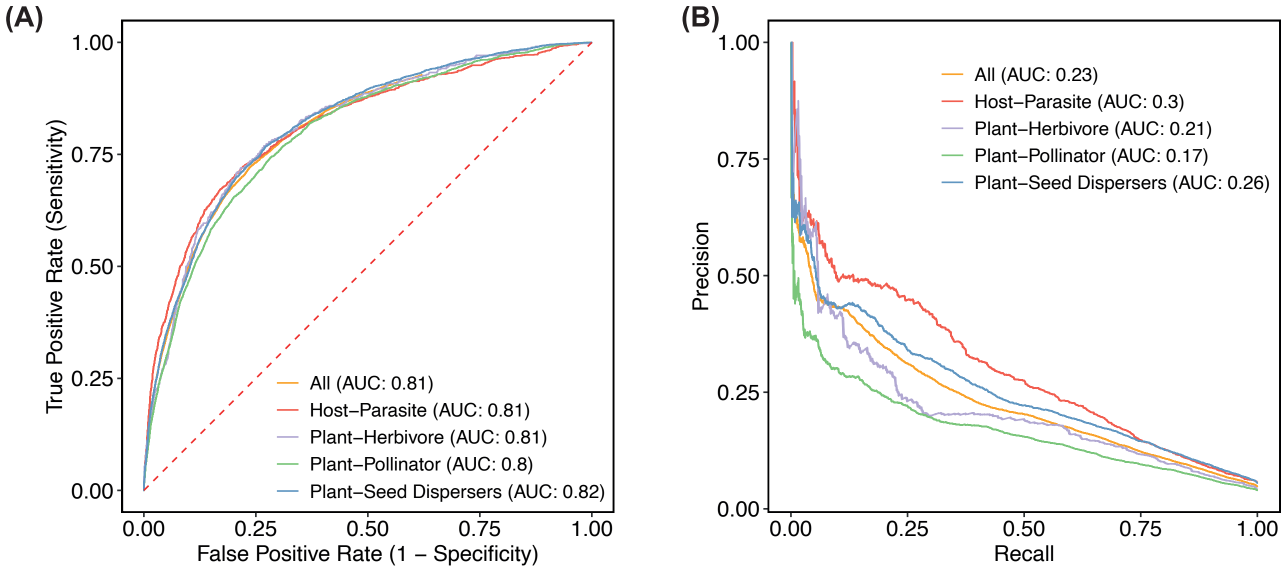

Fig. 3: Model evaluation across decision thresholds. The curves represent the performance across the 5 folds, either on all communities or on each community type separately using decision threshold levels ranging from 0 to 1. (A) The receiver operating characteristic (ROC) curve. The model outperforms random guessing as all the curves are above the diagonal dashed line, which depicts random guessing. (B) Model evaluation with the precision-recall tradeoff curve.

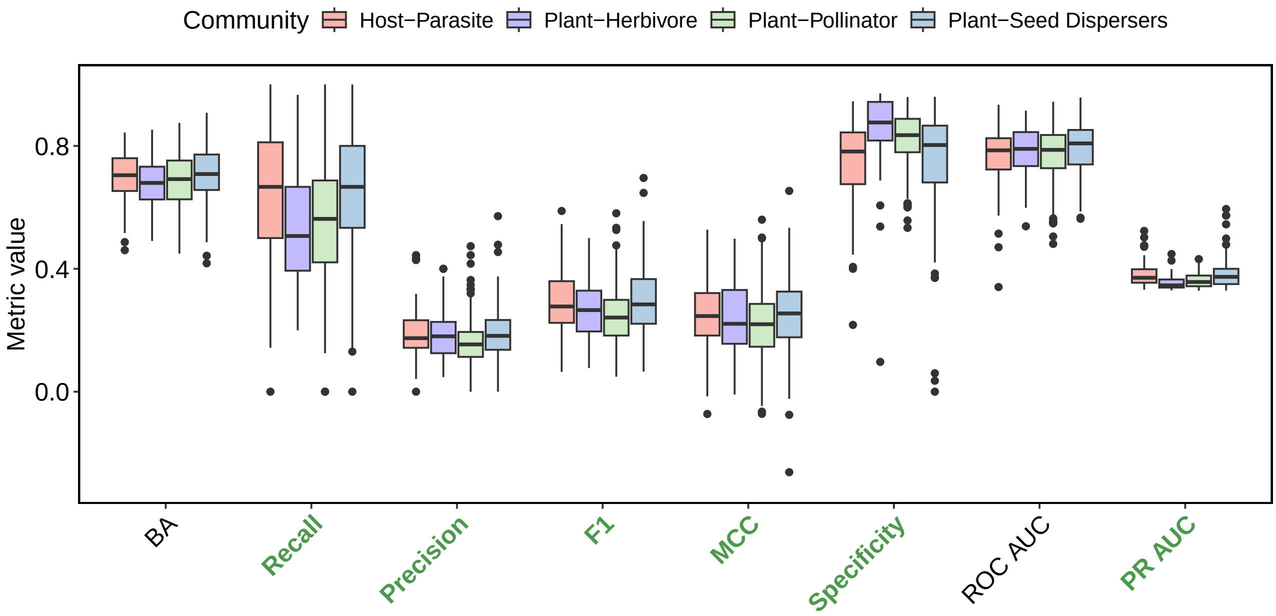

Fig. 4: Distributions of evaluation metrics per test community type. The model was trained on networks from all communities, and tested separately on each community type. Each data point is a network and box plots contain networks from all the five outer folds. Evaluation metrics whose labels are green showed a statistically significant difference between communities (Tables S2, S3). MCC is the only metric that can be negative (ranges -1 to 1).

Communities vary in their quality as train and test sets

ILP further allows us to perform cross-community link prediction via transfer learning. We trained the model using networks from a given community type and used them to predict links in the same community or in any other. To create unbiased comparisons, we ensured that the number and identity of the networks in the training set were the same across experiments. For instance, we used the same host-parasite networks for training when training on all networks or host-parasite networks alone. Because we are primarily interested in predicting missing links accurately, we focus on the F1 metric. We expected prediction to be more robust within community types (the diagonal in Fig. 5). This prediction was confirmed for host-parasite and plant-herbivore networks. In contrast, plant-seed disperser interactions are predicted similarly well when trained on host-parasite or plant-herbivore networks, and plant-pollinator interactions are better predicted when the model was trained on plant-seed disperser networks.

We further hypothesized that pooling data across all networks (All) for training will improve predictions (i.e., that the "All" column will have higher values than within-community predictions). In contrast to our hypothesis, pooling data worsened the predictions. This result can stem from the fact that plant-pollinator networks are highly sparse, and consistently under-perform (Fig. 3B), affecting overall predictions. Removing plant-pollinator networks from the training set (No PP) improved predictions. Yet, training with the No PP pool performs only as well as training with the worst single community.

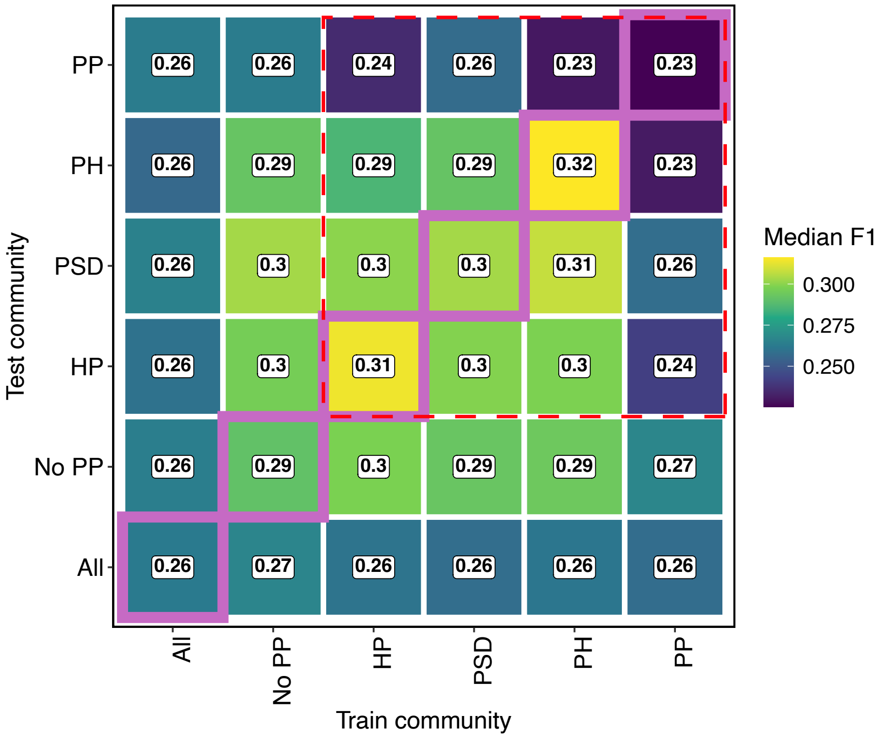

Fig. 5: Link prediction within and between community types. Each cell in the heatmap is the median of F1, calculated across all networks for a model that was trained on data from a certain community (columns) and tested on another (rows). The diagonal depicts model results for models trained and tested on the same community type(s). The red dashed border encloses models trained and tested using single community types. "All" is a model trained or tested using pooled data from all communities. "No PP" is the same as "All" but without plant-pollinator networks.

Estimating the bounds of link prediction performance

One issue common to all link prediction studies is the lack of knowledge of the ground truth. That is, which of the non-existing links are truly missing (if we had this knowledge, we would not need link prediction). Ideally, link prediction models would guide the field sampling efforts of ecologists who want to complete their networks. In turn, field data can evaluate model predictions. To date, no empirical study estimated model performance in light of ground truth. Undertaking such effort is necessary but may be extremely time consuming because if links were missing in the first place, an intensive amount of sampling would likely be needed to sample even a few of them. As a starting point, we take here an alternative computational approach. We calculate the bounds of the ILP and TLP model predictions across a theoretical range of proportion of missing links. We devised two scenarios. In the best-case scenario, the false positive links of the models were indeed missing links. In the worst-case scenario, the predicted negative links are actually positive in nature (see Methods).

We compared the models of the ILP and TLP. Generally speaking, the goal is to accurately predict missing links, evaluated with recall and F1. Recall is a good measure of our model's predictive ability because it is based solely on instances we manipulated (subsampled), which are observed links. The ILP model outperforms TLP in recovering missing links overall, as indicated by the consistently higher recall and F1 (which also considers precision) in the best-case and worst-case scenarios. Beyond a certain threshold of missing links, the models' recall will decline below the recall we observe in the data we use (horizontal line). This is because there are more missing links than the FPs predicted by the models. This threshold is higher in ILP than in TLP (0.33 vs. 0.27), providing further evidence that ILP is better suited for predicting missing links. In contrast to ILP, the TLP model will better predict zeros across the range of missing link proportions, as indicated by higher specificity.

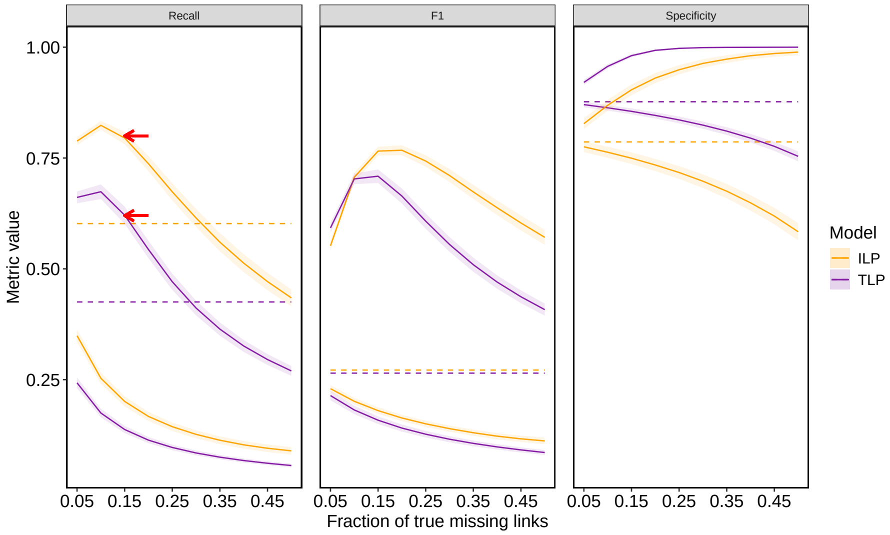

Fig. 6: Bounds of model predictions across proportions of ground-truth missing links. The horizontal dashed lines depict the current models' performance. The upper and lower curves (solid lines) represent the best and worst-case scenarios, respectively. An example for figure interpretation for the red arrows: If the true proportion of missing links in nature is 0.15, the ILP and TLP models best-scenario recall values would be ≈ 0.8 and ≈ 0.62, respectively. Bounds were calculated as means, and their confidence intervals (light-colored ribbons) were calculated across all networks. The ILP model was one in which the train and test contained networks from all communities.

Discussion

Species interactions are the backbone of ecosystem functioning. Link prediction helps us improve our knowledge of species interaction structure and gain insights into how interactions would change in response to perturbations. However, training data are often incomplete and biased. In this paper, we take a step forward using ILP, leveraging the structural similarities across different ecological communities [22]. The ILP approach also allowed us to address data sparsity and incompleteness in ecological networks by pooling information across multiple ecosystems. It also introduces a method to predict interactions in entirely new networks via transfer learning, which is particularly valuable for managing invasive species and predicting disease-host interactions in network contexts.

ILP offers three conceptual advantages over TLP. First, TLP is often limited by the sparse and incomplete nature of ecological network data. ILP models overcome these limitations by pooling information across different systems. Indeed, the model we developed generally outperformed transductive models per network in predicting 1's. However, TLP models better predicted 0's, pointing to a trade-off between predicting these two classes. Considering that ecologists are most interested in predicting 1's, ILP offers a better solution.

While increasing the data set size should theoretically improve predictions, we found that a model trained on all networks did not perform better than models trained on specific community types. Specifically, plant-pollinator networks weaken the prediction. This may happen because while numerous instances existed in the data set, plant-pollinator networks had the lowest connectance. [16]. Low connectance leads to a significant imbalance between existing and non-existing links, which can decrease the precision and recall scores.

Second, TLP models rely on the properties of a single network to predict missing links within that same network and are not generalized in their ability to predict interactions across various ecological contexts. In contrast, in ILP, the train and test sets do not need to have the same species. Therefore, it is possible to train a general model that can later be used to predict links in any new network. Such a general model can be applied to predict links of invasive species or new disease hosts in a network context. We trained such a model and predicted links in specific community types. While transfer learning is a promising approach [13,38], our results indicate that variation between communities in their quality as training data underscores the need to also consider the unique characteristics of each community type. Future studies can improve our model by incorporating features specific to community types, such as traits [9,12,13] or environmental features [42], increasing the train set performance. Nevertheless, adding traits to a model that trains on different kinds of networks is challenging due to substantial biases in data availability between species groups (e.g., parasites vs. birds).

Third, biases in sampling methods and efforts inevitably lead to biases in network data quality and completeness. Moreover, there is no ground truth to tell us what the actual missing and forbidden links are. This flaw permeates both the learning process and the evaluative framework. In ILP, missing links and subsampled links of the tested networks are not used in the learning phase. Therefore, despite the inherent biases, the evaluation of missing links is better in ILP models. We performed the first theoretical evaluation of model performance compared to true proportions of missing links and showed that ILP would retrieve missing links better than TLP. While ILP models provide a better guide to detecting missing links, validating their outcomes should be done via field sampling of well-known systems in which missing and forbidden links can be evaluated with high credibility or via literature reviews [43].

Prediction accuracy varied across different types of ecological networks. Host-parasite and plant-seed disperser networks exhibited higher recall and F1 rates than plant-pollinator and plant-herbivore, indicating better predictive performance for positive links. This may be because host-parasite and seed-disperser networks involve more specialized animal groups, making their topology more predictive. In contrast, plant-pollinator networks contain diverse, heterogeneous animal species with distinct behaviors and roles. Furthermore, host-parasite and plant-seed disperser networks have higher connectance (Table S1).

Using networks from multiple communities also allowed us to perform cross-community prediction—an idea that, to our knowledge, was tested once on only four food webs using trait data [13]. In contrast to that study, we found that predictions were not necessarily better within each community. Specifically, plant-pollinator interactions are predicted better when trained on plant-seed disperser networks. Apart from plant-pollinator networks, the overall between-community prediction was similar to within-community predictions. This finding is in line with the notion that different kinds of ecological networks can be subjected to similar assembly patterns or co-evolutionary dynamics [44], resulting in similar structures and constraints on interactions [22]. The ability to predict between communities using topological features reinforces that they share a non-random structure despite taxonomic differences in their species composition.

In conclusion, our study highlights the potential of ILP models to enhance our understanding of ecological networks by leveraging structural similarities across diverse communities. The ILP model we developed outperformed transductive models in retrieving links by pooling information from multiple networks, thus overcoming the limitations of data sparsity and incompleteness. However, our findings also reveal significant variability in prediction accuracy and the quality of train sets across different ecological communities, emphasizing the need for tailored approaches that consider the unique characteristics of each community. Future work should focus on integrating additional ecological and environmental variables such as phylogeny and species traits to improve model performance and on empirically validating predictions to refine further and enhance the applicability of ILP in ecological research. Developing a unified ILP model that considers the multifaceted nature of ecological networks could offer a practical solution to predicting links in our rapidly changing ecosystems.

Methods

Nested cross validation and hyperparameters

Nested cross-validation allows optimization of hyperparameters along with an unbiased estimation of the model's performance. This reduces the risk of over-fitting by ensuring that the model evaluation is conducted on data not seen during the hyperparameter tuning phase [45]. Hyperparameters control the learning process rather than being learned from the data, and are set before the training process begins [46]. The optimal values for the hyperparameters that maximize the machine learning model's performance are selected through hyperparameter tuning during that inner loop of the nested cross-validation (Table S4). We optimized hyperparameters using a random search, which is a computationally efficient tuning technique. The process involved randomly sampling values from a predefined search space, followed by training and evaluating model performance on those values using the training and validation sets. We used the F1-score as the performance metric to evaluate the hyperparameter combinations. The average F1-score across all folds was calculated for each random set of hyperparameters. We repeated this procedure for multiple random sets of hyperparameters, and the set that yields the highest F1 score was chosen as the optimal solution for the current iteration of the outer loop.

Because the data includes multiple networks, we used grouped cross-validation to prevent information leakage between training, validation, and test sets [47]. This method ensures each fold contains entire networks, increasing model robustness by preventing instances from the same network from appearing in the training, validation and test sets simultaneously.

We built and evaluated the machine learning models using the `Scikit-learn` package [48], with the addition of the `xgboost` package [49]. We executed hyperparameter optimization through randomized search using the scikit-learn's RandomizedSearchCV [48] and the `kerastuner` package [50]. We used randomized optimization due to its efficiency in exploring the hyperparameter space with fewer iterations compared to traditional grid search method. We optimized the model's hyperparameters using the F1-score.

Link subsampling and feature extraction

Because the goal is to predict missing links (possible yet unobserved interactions [51]), link prediction requires generating a ground truth. That is, a lack of link (0 in the matrix), of which we are certain the link actually exists. A common way to do so is by removing some existing links randomly (sub-sampling) [8,52]. Hence, to emulate real-world under-sampled networks, each network in the validation and test sets was sub-sampled by randomly removing 20% of the existing links (20% is the standard). Sub-sampling creates three types of links:

- *Existing links*: links that exist in the network and were not sub-sampled.
- *Non-existing links*: links that did not exist in the network. These may be missing or forbidden (interactions not possible due to some ecological, morphological or other constraints [51]).
- *Subsampled links*: links that existed in the network and are now missing because they were subsampled. Because we want to predict instances in which non-existing links should have existed, sub-sampled links were relabeled as existing after feature extraction.

Each instance in the data set, representing an interaction (or lack of an interaction), between each two species, constitutes a vector of topological features. Our features encompassed four levels as follows:

- Network-level features: Defined for a whole network (e.g., nestedness). Hence, all the instances that are related to a network will get the same values for those features. We also included the type of network (e.g., plant-pollinator).
- Meso-scale level features: defined for groups of nodes (e.g., motifs).
- Link-level features: Defined for a pair of nodes (e.g., preferential attachment: the multiplication of both node degrees).

- Node-level features: Defined for to each node (e.g., centrality, degree). Each instance will have two different versions of the feature, one for each node in the pair.

Feature extraction is done once on all the data, and is not related to the fold. We rescaled numerical features to a range of [0, 1] to ensure that no particular feature dominates others during the learning process. We calculated feature importance based on the average decrease in Gini impurity across all trees. Features that are more important are used more frequently and result in significant improvements in node purity.

For the analysis of ecological networks and extraction of topological features (network properties), we used the `networkx` package [53] in Python and the `igraph` [54] and `bipartite` [55] R packages. We handled data manipulation through `numpy` [56] and `Pandas` [57,58] in Python. A complete list of the features we used and their descriptions are provided on the GitHub repository accompanying this article (`https://github.com/Ecological-Complexity-Lab/eco_ILP/blob/main/results/final/features.csv`).

Dealing with class imbalance

In machine learning, class imbalance is a situation in which the distribution of classes in the training data is highly skewed, where one or more classes have considerably fewer samples compared to others. This imbalance can significantly impact the training and evaluation of prediction models [40]. Classifiers developed with such skewed data tend to favor the majority class, which can lead to subpar performance when identifying instances of the minority class. This issue is particularly prevalent in ecological networks, characterized by their sparsity (i.e., low connectance). In binary classification tasks, this sparsity creates a disparity between the small number of existing links (positive class) and the much larger set of non-existing links (negative class) [11,40]. To overcome this problem we incorporated cost-sensitive learning (Supplementary note) [48]. Specifically, to make the importance of both classes equal, we computed their weights inversely proportionally to the frequency of the respective classes [59,60].

Evaluation metrics

Performance of LP models can be evaluated using multiple indices, each providing a different perspective on the model's strengths and weaknesses [11,40]. The evaluation is based on a confusion matrix that contains the number of true positives, true negatives, false positives and false negatives (Supplementary Note). We used common evaluators adequate for imbalanced data: recall ($TP/(TP + FN)$), precision ($TP/(TP + FP)$), F1 (harmonic mean of precision and recall), specificity ($TN/(TN + FP)$), balanced accuracy (the arithmetic mean of recall and specificity), the area under the receiver operating characteristic curve (ROC AUC), and the area under the precision-recall curve (PR AUC). In addition, as recently recommended for ecological networks, we also used MCC [40]:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{1}$$

We evaluated links based on the common threshold of 0.5 (see Supplementary note for details).

Comparison to transductive models

We compared our model to three transductive link prediction models: The SBM and connectance, which were previously published [8] and are not based on machine learning, and to a transductive machine learning model we developed (TLP). All three models train and test on each network separately. The models do not separate links between the train and test set. In the train set subsampled links are classified as non-existing links, while in the test set there is a relabeling of the subsampled links to existing links.

In *SBM*, nodes are partitioned into blocks or groups, and the probability of a link between any two nodes depends on the blocks to which they belong [61]. The probability of a link between two nodes is higher if they belong to the same block, reflecting community structure. The stochastic block model is a degree corrected bipartite SBM algorithm, which accounts for the degree heterogeneity within the blocks. The *connectance model* is a model used to describe the pattern of interactions in ecological networks. It assigns a connectivity value to each species, reflecting its propensity to interact with others. The connectivity values are estimated by using maximum likelihood optimization, which adjusts the parameters to best fit the observed interaction data [8]. We used the R package cassandRa (https://github.com/jcdterry/cassandRa) to predict links using these two models. The TLP model we developed uses random forest with a standard 3-fold cross validation with hyper-parameters tuning (maximizing f1 score). The model balances the weights of the classes by computing weights inversely proportionally to the frequency of the respective classes.

Estimating the bounds of link prediction performance

We evaluate the models' ability to predict links under a range of true missing link values. For a theoretical proportion p (range 0.05-0.5) of ground-truth missing links, the **number** of missing links is calculated as $L_m = p \times L_{ne}$, where L_{ne} is number of zeros in the original matrix. In the *best case scenario*, we choose L_m FP links and change them to TP. This simulates a scenario in which the model indeed predicted links that were not observed. If $L_{FP} < L_m$ (L_{FP} is the number of FP links) we convert a remainder of $L_m - L_{FP}$ TN links to FN to simulate the scenario in which the links actually existed in nature. In the *worst-case scenario*, we choose L_m TN links and convert their class to FN. This simulates a scenario in which the links actually existed in nature but the model failed to retrieve them. If $L_{TN} < L_m$ (L_{TN} is the number of TN links) we convert a remainder of $L_m - L_{FN}$ FP links to TP. After the conversion of links, we recalculate each evaluation metric (e.g., F1, recall) for each of the two scenarios, to form the upper and lower bounds of model performance per network.

Code and data availability

The data are available in the repository set up in original publication https://osf.io/my9tv/. The full code and technical descriptions on how to run the pipeline are available on the GitHub repository https://github.com/Ecological-Complexity-Lab/eco_ILP.

Acknowledgments

We thank Prof. Peter Mucha for comments and suggestions.

Funding

This work was supported by grants to SP from the Israeli Data Science Initiative, Israel Science Foundation (grant 1281/20), the Human Frontiers Science Program Organization (grant number RGY0064/2022) and the Binational Science Foundation (NSF-BSF grant 2022721)

References

1. Windsor, F. M. *et al.* Network science: Applications for sustainable agroecosystems and food security. *Perspectives in Ecology and Conservation* **20,** 79–90. doi:10.1016/j.pecon.2022.03.001 (2022).

2. Magrach, A. *et al.* Plant-pollinator networks in semi-natural grasslands are resistant to the loss of pollinators during blooming of mass-flowering crops. *Ecography* **41,** 62–74. doi:10.1111/ecog.02847 (2018).

3. Falcão, J. C. F., Dáttilo, W. & Rico-Gray, V. Sampling effort differences can lead to biased conclusions on the architecture of ant–plant interaction networks. *Ecol. Complex.* **25,** 44–52. doi:10.1016/j.ecocom.2016.01.001 (2016).

4. Jordano, P. Sampling networks of ecological interactions. *Funct. Ecol.* **30,** 1883–1893. doi:10.1111/1365-2435.12763 (2016).

5. Dallas, T., Park, A. W. & Drake, J. M. Predicting cryptic links in host-parasite networks. *PLoS Comput. Biol.* **13,** e1005557. doi:10.1371/journal.pcbi.1005557 (2017).

6. Rohr, R. P., Naisbit, R. E., Mazza, C. & Bersier, L.-F. Matching-centrality decomposition and the forecasting of new links in networks. *Proc. Biol. Sci.* **283.** doi:10.1098/rspb.2015.2702 (2016).

7. Desjardins-Proulx, P., Laigle, I., Poisot, T. & Gravel, D. Ecological interactions and the Netflix problem. *PeerJ* **5,** e3644. doi:10.7717/peerj.3644 (2017).

8. Terry, J. C. D. & Lewis, O. T. Finding missing links in interaction networks. *Ecology* **101,** e03047. doi:10.1002/ecy.3047 (2020).

9. Pichler, M., Boreux, V., Klein, A., Schleuning, M. & Hartig, F. Machine learning algorithms to infer trait-matching and predict species interactions in ecological networks. *Methods Ecol. Evol.* **11,** 281–293. doi:10.1111/2041-210x.13329 (2020).

10. Stock, M. *et al.* Pairwise learning for predicting pollination interactions based on traits and phylogeny. *Ecol. Modell.* **451,** 109508. doi:10.1016/j.ecolmodel.2021.109508 (2021).

11. Strydom, T. *et al.* A roadmap towards predicting species interaction networks (across space and time). *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **376,** 20210063. doi:10.1098/rstb.2021.0063 (2021).

12. Benadi, G., Dormann, C. F., Fründ, J., Stephan, R. & Vázquez, D. P. Quantitative Prediction of Interactions in Bipartite Networks Based on Traits, Abundances, and Phylogeny. *Am. Nat.* **199,** 841–854. doi:10.1086/714420 (2022).

13. Caron, D. *et al.* Trait-matching models predict pairwise interactions across regions, not food web properties. *Glob. Ecol. Biogeogr.* **33.** doi:10.1111/geb.13807 (2024).

14. Ahmadi, Z. *et al.* Inductive and transductive link prediction for criminal network analysis. *J. Comput. Sci.* **72,** 102063. doi:10.2139/ssrn.4331130 (2023).

15. Fire, M., Puzis, R. & Elovici, Y. in *Handbook of Computational Approaches to Counterterrorism* (ed Subrahmanian, V. S.) 283–300 (Springer New York, New York, NY, 2013). doi:10.1007/978-1-4614-5311-6_14.

16. Galkin, M., Berrendorf, M. & Hoyt, C. T. An Open Challenge for Inductive Link Prediction on Knowledge Graphs. *arXiv [cs.LG]* (2022).

17. Watts, D. J. & Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature* **393,** 440–442 (1998).

18. Barabási, A.-L. *Network Science* (2021).

19. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286,** 509–512. doi:10.1126/science.286.5439.509 (1999).

20. Milo, R *et al.* Network motifs: simple building blocks of complex networks. *Science* **298,** 824–827. doi:10.1126/science.298.5594.824 (2002).

21. Fortuna, M. A. *et al.* Nestedness versus modularity in ecological networks: two sides of the same coin? *J. Anim. Ecol.* **79,** 811–817. doi:10.1111/j.1365-2656.2010.01688.x (2010).

22. Michalska-Smith, M. J. & Allesina, S. Telling ecological networks apart by their structure: A computational challenge. *PLoS Comput. Biol.* **15,** e1007076. doi:10.1371/journal.pcbi.1007076 (2019).

23. Vázquez, D. P., Blüthgen, N., Cagnolo, L. & Chacoff, N. P. Uniting pattern and process in plant–animal mutualistic networks: a review. *Ann. Bot.* **103,** 1445–1457. doi:10.1093/aob/mcp057 (2009).

24. Bascompte, J. & Jordano, P. *Mutualistic networks* (Princeton University Press, Princeton, 2014).

25. Segar, S. T. *et al.* The Role of Evolution in Shaping Ecological Networks. *Trends Ecol. Evol.* **35,** 454–466. doi:10.1016/j.tree.2020.01.004 (2020).

26. Bascompte, J. & Stouffer, D. B. The assembly and disassembly of ecological networks. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **364,** 1781–1787. doi:10.1098/rstb.2008.0226 (2009).

27. Delmas, E. *et al.* Analysing ecological networks of species interactions: Analyzing ecological networks. *Biol Rev* **94,** 16–36. doi:10.1111/brv.12433 (2019).

28. Guimarães, P. R. The Structure of Ecological Networks Across Levels of Organization. *Annu. Rev. Ecol. Evol. Syst.* doi:10.1146/annurev-ecolsys-012220-120819 (2020).

29. Jordano, P., Bascompte, J. & Olesen, J. M. Invariant properties in coevolutionary networks of plant–animal interactions. *Ecol. Lett.* **6,** 69–81. doi:10.1046/j.1461-0248.2003.00403.x (2003).

30. Vázquez, D. P., Poulin, R., Krasnov, B. R. & Shenbrot, G. I. Species abundance and the distribution of specialization in host–parasite interaction networks. *J. Anim. Ecol.* **74,** 946–955 (2005).

31. Williams, R. J. Biology, methodology or chance? The degree distributions of bipartite ecological networks. *PLoS One* **6,** e17645. doi:10.1371/journal.pone.0017645 (2011).

32. Bascompte, J., Jordano, P., Melián, C. J. & Olesen, J. M. The nested assembly of plant–animal mutualistic networks. *Proc. Natl. Acad. Sci. U. S. A.* **100,** 9383–9387. doi:10.1073/pnas.1633576100 (2003).

33. Olesen, J. M., Bascompte, J., Dupont, Y. L. & Jordano, P. The modularity of pollination networks. *Proc. Natl. Acad. Sci. U. S. A.* **104,** 19891–198916. doi:10.1073/pnas.0706375104 (2007).

34. Krasnov, B. R. *et al.* Phylogenetic signal in module composition and species connectivity in compartmentalized host-parasite networks. *Am. Nat.* **179,** 501–511. doi:10.1086/664612 (2012).

35. Dallas, T. & Cornelius, E. Co-extinction in a host-parasite network: identifying key hosts for network stability. *Sci. Rep.* **5,** 13185. doi:10.1038/srep13185 (2015).

36. Gao, M. *et al.* Inductive Link Prediction via Interactive Learning Across Relations in Multiplex Networks. *IEEE Transactions on Computational Social Systems* **PP,** 1–13. doi:10.1109/TCSS.2022.3176928 (2022).

37. Hao, Y., Cao, X., Fang, Y., Xie, X. & Wang, S. Inductive Link Prediction for Nodes Having Only Attribute Information. *arXiv [cs.LG]* (2020).

38. Strydom, T *et al.* Graph embedding and transfer learning can help predict potential species interaction networks despite data limitations. *Methods Ecol. Evol.* **14,** 2917–2930. doi:10.1111/2041-210X.14228 (2023).

39. Brimacombe, C., Bodner, K., Michalska-Smith, M., Poisot, T. & Fortin, M.-J. Shortcomings of reusing species interaction networks created by different sets of researchers. *PLoS Biol.* **21,** e3002068. doi:10.1371/journal.pbio.3002068 (2023).

40. Poisot, T. Guidelines for the prediction of species interactions through binary classification. *Methods Ecol. Evol.* **14,** 1333–1345. doi:10.1111/2041-210x.14071 (2023).

41. Poisot, T., Canard, E., Mouquet, N. & Hochberg, M. E. A comparative study of ecological specialization estimators. *Methods Ecol. Evol.* **3,** 537–544. doi:10.1111/j.2041-210X.2011.00174.x (2012).

42. Song, C. & Saavedra, S. Telling ecological networks apart by their structure: An environment-dependent approach. *PLoS Comput. Biol.* **16,** e1007787. doi:10.1371/journal.pcbi.1007787 (2020).

43. Farrell, M. J., Elmasri, M., Stephens, D. A. & Davies, T. J. Predicting missing links in global host-parasite networks. *J. Anim. Ecol.* **91,** 715–726. doi:10.1111/1365-2656.13666 (2022).

44. Ponisio, L. C. & M'Gonigle, L. K. Coevolution leaves a weak signal on ecological networks. *Ecosphere* **8,** e01798. doi:10.1002/ecs2.1798 (2017).

45. Yates, L. A., Aandahl, Z., Richards, S. A. & Brook, B. W. Cross validation for model selection: A review with examples from ecology. *Ecol. Monogr.* **93.** doi:10.1002/ecm.1557 (2023).

46. Bergstra, J. & Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **13** (2012).

47. Roberts, D. R. *et al.* Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography* **40,** 913–929. doi:10.1111/ecog.02881 (2017).

48. Pedregosa, F *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* (2011).

49. Chen, T. & Guestrin, C. *XGBoost: A Scalable Tree Boosting System* in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* San Francisco, California, USA (Association for Computing Machinery, New York, NY, USA, 2016), 785–794. doi:10.1145/2939672.2939785.

50. O'Malley, T. *et al. KerasTuner*

51. Olesen, J. M. *et al.* Missing and forbidden links in mutualistic networks. *Proc. Biol. Sci.* **278,** 725–732. doi:10.1098/rspb.2010.1371 (2011).

52. Lü, L. & Zhou, T. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications* **390,** 1150–1170. doi:10.1016/j.physa.2010.11.027 (2011).

53. Hagberg, A., Swart, P. J. & Schult, D. A. *Exploring network structure, dynamics, and function using NetworkX* research rep. LA-UR-08-05495; LA-UR-08-5495 (Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008).

54. Csárdi, G & Nepusz, T. The igraph software package for complex network research. *InterJournal Complex Systems* **1695,** 1695 (2006).

55. Dormann, C. F., Fründ, J., Blüthgen, N. & Gruber, B. Indices, graphs and null models: analyzing bipartite ecological networks. *The Open Ecology Journal* **2,** 7–24. doi:10.2174/1874213000902010007 (2009).

56. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585,** 357–362. doi:10.1038/s41586-020-2649-2 (2020).

57. Team, T. P. D. Pandas development Pandas-dev/pandas: Pandas. *Zenodo* **21,** 1–9 (2020).

58. McKinney, W. *Data Structures for Statistical Computing in Python* in *Proceedings of the 9th Python in Science Conference* Python in Science ConferenceAustin, Texas (SciPy, 2010). doi:10.25080/majora-92bf1922-00a.

59. Thai-Nghe, N., Gantner, Z. & Schmidt-Thieme, L. *Cost-sensitive learning methods for imbalanced data* in *The 2010 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2010), 1–8. doi:10.1109/IJCNN.2010.5596486.

60. Yang, Y., Lichtenwalter, R. N. & Chawla, N. V. Evaluating link prediction methods. *Knowl. Inf. Syst.* **45,** 751–782. doi:10.1007/s10115-014-0789-0 (2015).

61. Holland, P. W., Laskey, K. B. & Leinhardt, S. Stochastic blockmodels: First steps. *Soc. Networks* **5,** 109–137. doi:10.1016/0378-8733(83)90021-7 (1983).

Supplementary Information

Supplementary results

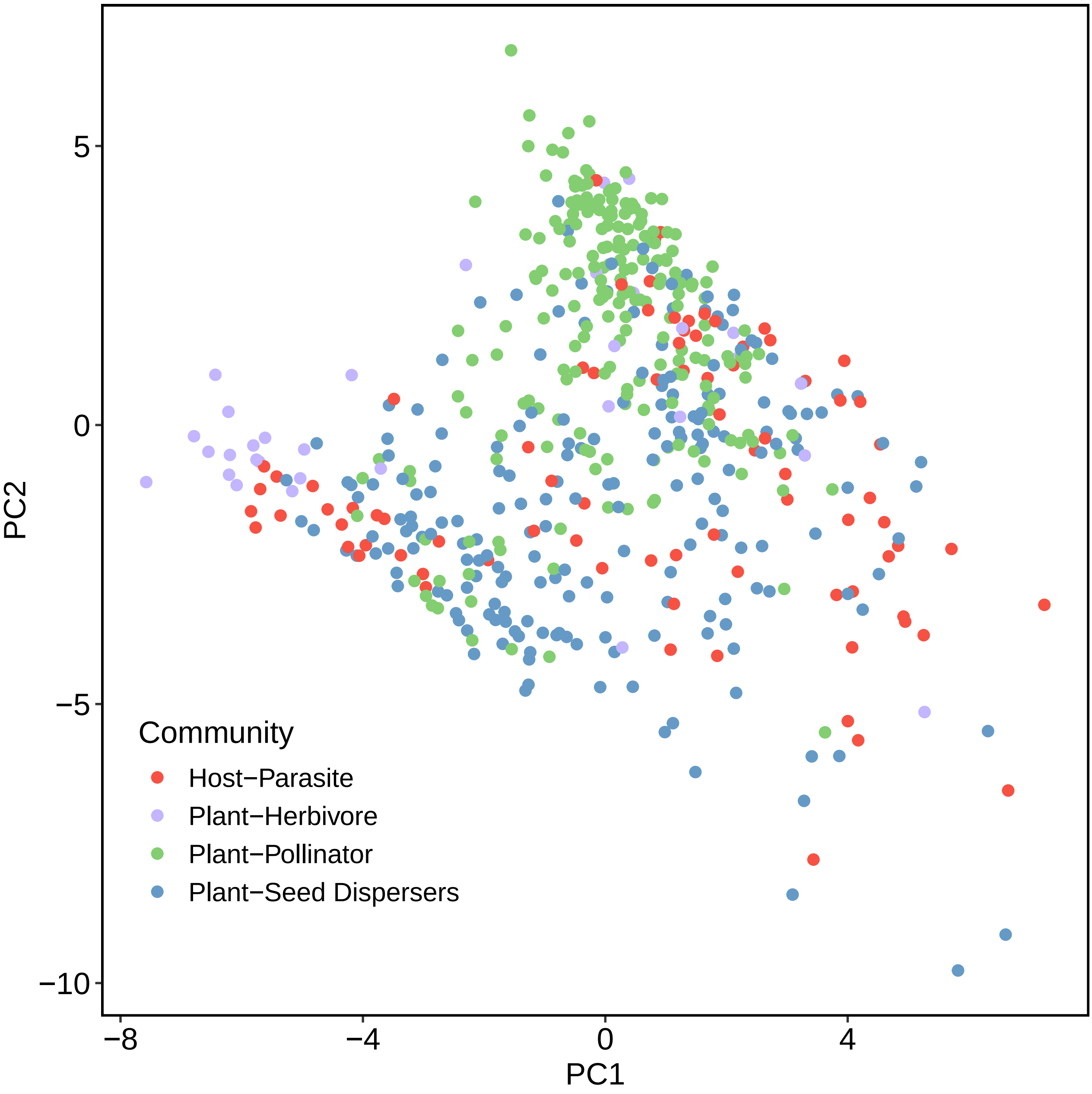

Fig. S1: Variation in network structure. The PCA shows no clear subgrouping of networks by their class, reflecting previous studies using the same data set [1,2]. The PCs were calculated using the topological features we used in the link prediction model.

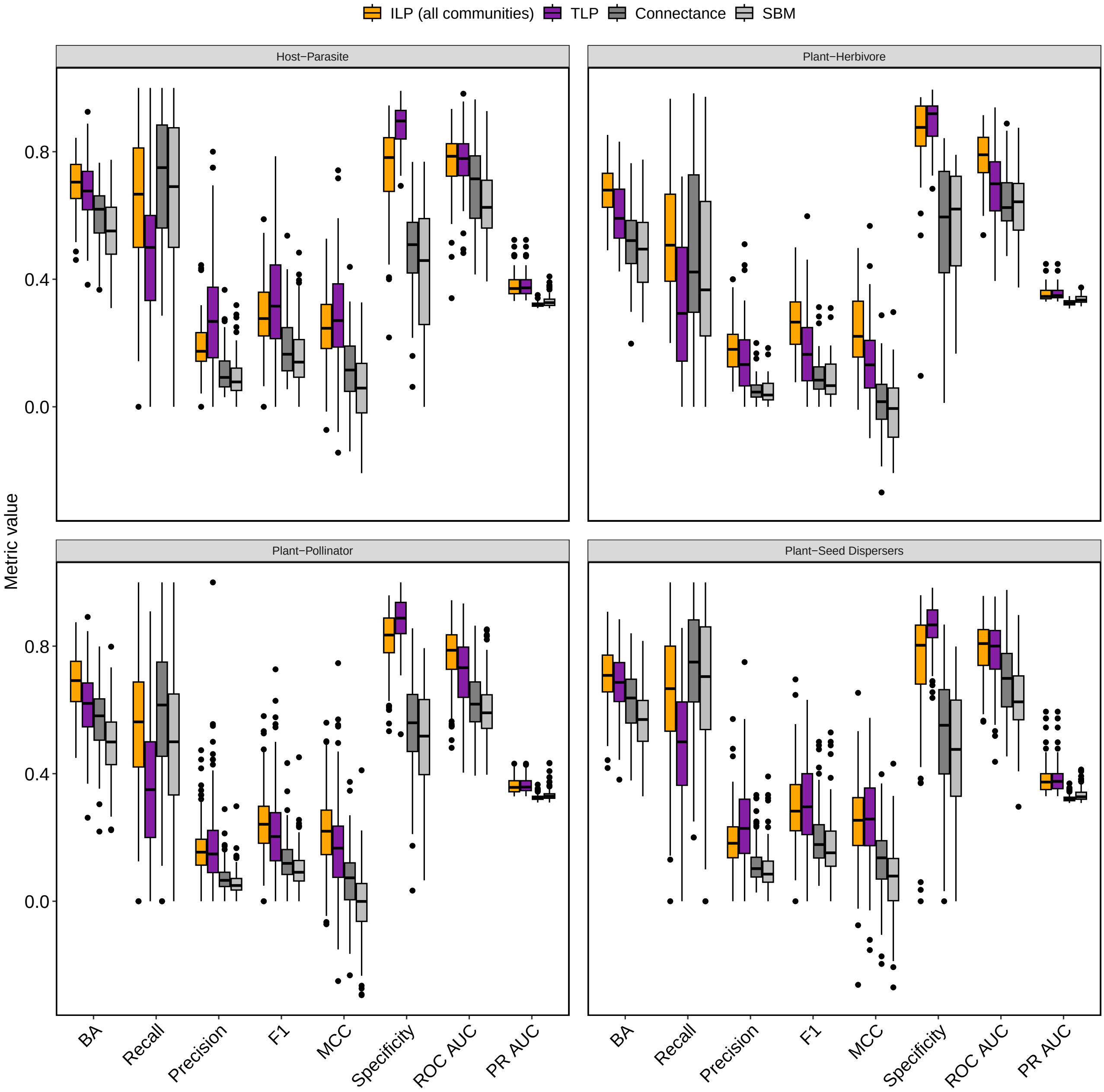

Fig. S2: Comparison of model performance per community. The inductive link prediction was trained on networks from all communities and tested per community type (the results were qualitatively the same when training and testing was done within the same community type). We compared ILP to two transductive previously published models (stochastic block model and connectance; [3]) and one transductive machine learning model that we developed (TLP). Each boxplot is a distribution of an evaluation metric using a 0.5 classification threshold, bedsides the ROC AUC and PR AUC. Each data point is a network and boxplots contain networks from all the five outer folds. BA is balanced accuracy. See definitions of evaluation metrics in the Methods.

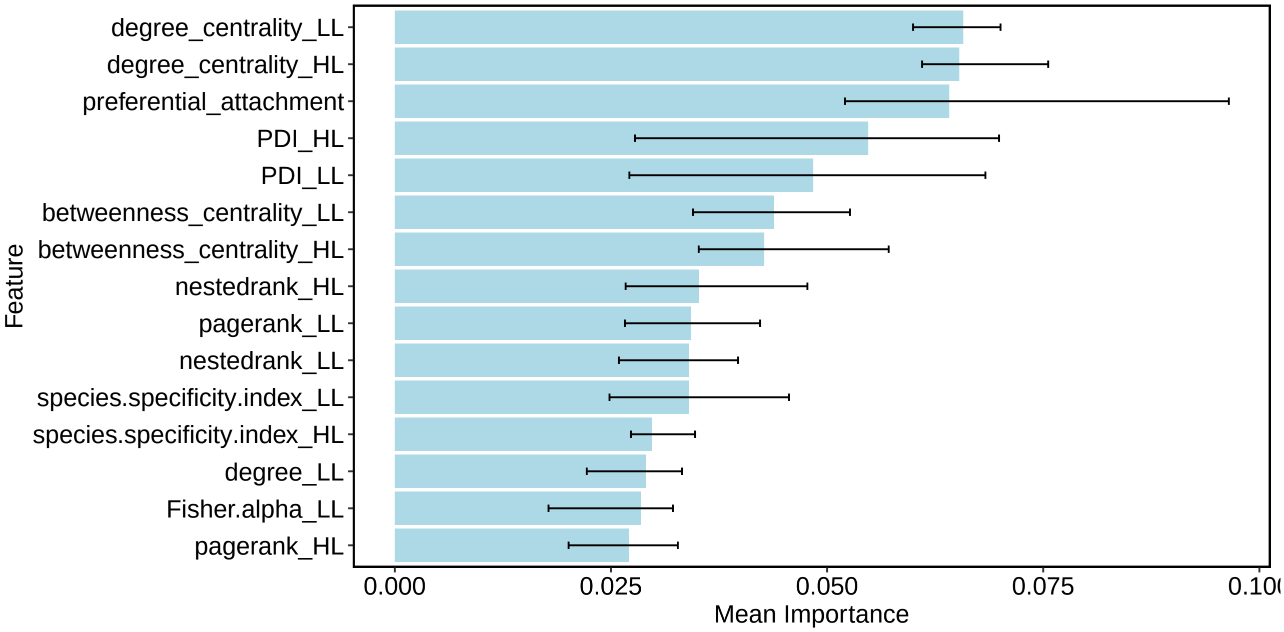

Fig. S3: Feature importance. Each bar presents the average value of feature importance across the five folds for the random forest model. Error bars depict minimum and maximum. HL are higher-level trophic species that consume resources (pollinators, parasites, seed dispersers, herbivores). LL are lower-level trophic species (resources) (plants, hosts, seeds).

Table S1: Summary of network properties. The table provides an overview of the network properties for different ecological communities, detailing each variable's mean, median, and range values. N: number of nodes; L_e: number of existing links; L_{ne}: number of nonexisting links; C: connectance, $C = L_e/(L_e+L_{ne})$

Variable	Community	Mean	Median	Range
N	Host-Parasite	37.58	33	20-137
	Plant-Herbivore	42.53	37	21-78
	Plant-Pollinator	43.36	36	20-205
	Plant-Seed Dispersers	43.96	36	20-213
L_e	Host-Parasite	82.86	63	19-490
	Plant-Herbivore	67.69	42.5	22-294
	Plant-Pollinator	72.94	52	21-631
	Plant-Seed Dispersers	98.59	72	21-720
L_{ne}	Host-Parasite	1622.08	991.5	349-18279
	Plant-Herbivore	1997.22	1332.5	409-5996
	Plant-Pollinator	2394.52	1260	363-41600
	Plant-Seed Dispersers	2585.2	1238	362-44687
C	Host-Parasite	0.28	0.25	0.11-0.61
	Plant-Herbivore	0.2	0.165	0.1-0.47
	Plant-Pollinator	0.22	0.2	0.1-0.42
	Plant-Seed Dispersers	0.28	0.27	0.1-0.71

Table S2: Results of Kruskal-Wallis test to detect differences between communities in multiple evaluation metrics. The KW test was performed for each metric separately, with communities as factor levels. Dunn post-hoc tests are in Table S3

Metric	χ^2	P Value
Balanced Accuracy	5.729	0.126
F1	29.460	1.79×10^{-6}
MCC	11.310	0.0102
PR AUC	47.862	2.28×10^{-10}
Precision	25.517	1.20×10^{-5}
Recall	37.908	2.96×10^{-8}
ROC AUC	7.053	0.0702
Specificity	37.557	3.51×10^{-8}

Table S3: Results of Dunn posthoc tests for the Kruskal-Wallis pairwise comparisons between communities in multiple evaluation metrics (Table S2). Only significant differences are presented here. Note that the median values are close for some comparisons despite being statistically significant (e.g., precision for Host-Parasite vs. Plant-Pollinator). A visualization of the values and their distributions is in the main text, in Fig. 4.

Metric	Community 1	Community 2	Median 1	Median 2	P Adjusted
Precision	Host-Parasite	Plant-Pollinator	0.17	0.15	4.10×10^{-11}
Precision	Plant-Herbivore	Plant-Pollinator	0.18	0.15	1.51×10^{-7}
Precision	Plant-Herbivore	Plant-Seed Dispersers	0.18	0.18	4.77×10^{-2}
Precision	Plant-Pollinator	Plant-Seed Dispersers	0.15	0.18	1.04×10^{-18}
Recall	Host-Parasite	Plant-Pollinator	0.67	0.56	4.10×10^{-11}
Recall	Plant-Herbivore	Plant-Pollinator	0.51	0.56	1.51×10^{-7}
Recall	Plant-Herbivore	Plant-Seed Dispersers	0.51	0.67	4.77×10^{-2}
Recall	Plant-Pollinator	Plant-Seed Dispersers	0.56	0.67	1.04×10^{-18}
PR AUC	Host-Parasite	Plant-Pollinator	0.37	0.36	4.10×10^{-11}
PR AUC	Plant-Herbivore	Plant-Pollinator	0.35	0.36	1.51×10^{-7}
PR AUC	Plant-Herbivore	Plant-Seed Dispersers	0.35	0.37	4.77×10^{-2}
PR AUC	Plant-Pollinator	Plant-Seed Dispersers	0.36	0.37	1.04×10^{-18}
Specificity	Host-Parasite	Plant-Pollinator	0.78	0.83	4.10×10^{-11}
Specificity	Plant-Herbivore	Plant-Pollinator	0.88	0.83	1.51×10^{-7}
Specificity	Plant-Herbivore	Plant-Seed Dispersers	0.88	0.80	4.77×10^{-2}
Specificity	Plant-Pollinator	Plant-Seed Dispersers	0.83	0.80	1.04×10^{-18}
F1	Host-Parasite	Plant-Pollinator	0.28	0.24	4.10×10^{-11}
F1	Plant-Herbivore	Plant-Pollinator	0.27	0.24	1.51×10^{-7}
F1	Plant-Herbivore	Plant-Seed Dispersers	0.27	0.28	4.77×10^{-2}
F1	Plant-Pollinator	Plant-Seed Dispersers	0.24	0.28	1.04×10^{-18}
MCC	Host-Parasite	Plant-Pollinator	0.25	0.22	4.10×10^{-11}
MCC	Plant-Herbivore	Plant-Pollinator	0.22	0.22	1.51×10^{-7}
MCC	Plant-Herbivore	Plant-Seed Dispersers	0.22	0.25	4.77×10^{-2}
MCC	Plant-Pollinator	Plant-Seed Dispersers	0.22	0.25	1.04×10^{-18}

Supplementary notes on methods

Models applied

Selecting a machine learning model a priori can be challenging [4]. There is no one-size-fits-all solution; different models have different strengths and weaknesses, and knowing which model will perform better is often impossible. Some models may perform well on particular data types, while others may perform better on different ones. Hence, model selection often requires some experimentation. In practice, trying multiple models and comparing their performance is often a good idea before deciding on a final model. Furthermore, using a model ensemble—a technique in which multiple models are combined—can often lead to better results than a single model [5]. In this study, we tried multiple models and their ensemble. For the ensemble, we averaged the probabilities for each prediction. The models performed similarly overall, and their ensemble did not improve the results (Fig. S4). Therefore, in the main text, we present results for random forest. In this section, we describe the models we used. We present model hyperparameters in Table S4.

To understand how these models work, it is necessary to first explain the terms bagging and boosting [6], two popular ensemble learning techniques. The key difference between bagging and boosting lies in how they combine multiple models to make predictions. Bagging, which stands for bootstrap aggregating, is a parallel ensemble technique. In bagging, multiple base models (multiple instances of the same algorithm) are trained independently on different random subsets of the training data using bootstrap sampling (sampling with replacement). The outputs of the individual models are then combined, often through a voting mechanism or by taking the average of their predictions, to produce a single output. Bagging is often used with decision trees as it has been repetitively shown to outperform other models. Boosting, on the other hand, is a sequential ensemble technique. Boosting works by iteratively training multiple weak models on modified versions of the training data, with each subsequent model trying to correct the errors of the previous models, focusing on the examples that the previous models misclassified. The training data is re-weighted at each iteration so that the misclassified examples receive higher weights and are given more importance in subsequent iterations. The final prediction of the boosted model is a weighted combination of the predictions of all the individual models, with the weights determined by the performance of each model on the training data. More weight is given to models that achieved higher performance.

Logistic regression. This generalized linear model uses a logistic function to model a binary dependent variable. The logistic function maps the linear combination of input variables to a discrete binary value between 0 and 1, which is interpreted as the probability of the input belonging to a particular class. During training, the logistic regression algorithm finds the input variables' best parameters (coefficients) by minimizing the error between the predicted probabilities and the actual class labels in the training data. Logistic regression is easy to interpret and implement without necessarily compromising performance [7].

Random Forest. [8,9] . This is a specific version of a bagging method with decision trees. A random forest model creates a collection of decision trees and combines their predictions to produce a final output. Decision tree models create a tree-like model of decisions and their possible consequences, with each internal node representing a feature, each branch representing a decision rule, and each leaf node representing the outcome class label. The algorithm starts at the root of the tree. It recursively splits the data into subsets using a feature that provides the most information gain at that stage until it reaches a leaf node, representing the predicted class label for the input data. Each tree in the random forest ensemble is trained on a different random subset of the data and features, and the final output is the combination of the predictions of all the decision trees, usually made by either averaging the results for regression tasks or by taking a majority vote for classification tasks.

XGBoost. This gradient-boosting tree-based algorithm is designed for speed and performance [10]. Gradient boosting [11] differs from other boosting algorithms in the way it calculates the weights of the models. Specifically, gradient boosting uses the gradient descent optimization algorithm to minimize the model's loss function. At each iteration, the algorithm calculates the (negative) gradient of the loss function with respect to the predictions of the previous ensemble. It uses this gradient to adjust the weights of the new model. The negative gradient represents the direction of the steepest descent for the loss function, which is the direction that will reduce the loss the most. By fitting a new tree to the negative gradient, gradient boosting focuses on the examples misclassified in the previous ensemble and attempts to correct those errors.

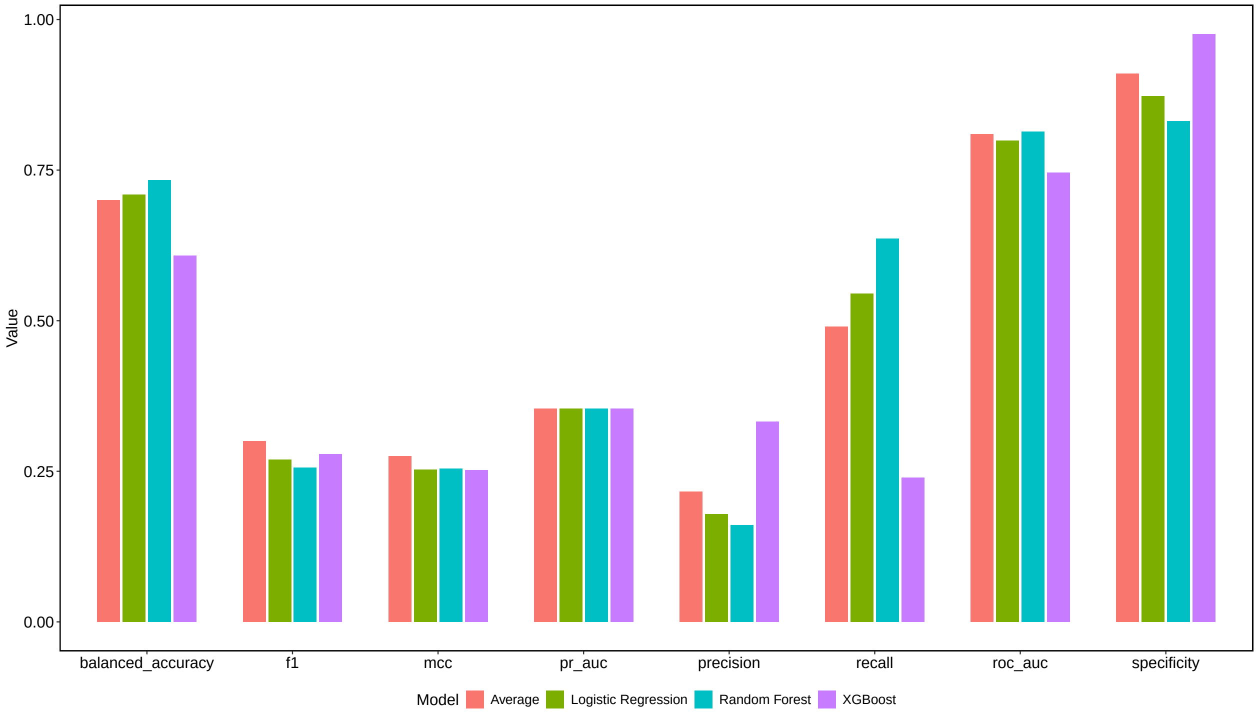

Fig. S4: Model comparison. We evaluated link prediction using various models. Average is a model ensemble. Overall, model choice did not qualitatively affect the results. Therefore, we present results of random forest throughout the main text.

Cost-sensitive learning

Cost-sensitive learning uses a parameter that the algorithm considers during the learning process. The penalty is the cost, which is minimized during the training process. This approach allows dealing with imbalance while ignoring less samples [12]. The scikit-learn package [13] has a built-in support for cost-sensitive learning. This implementation provides custom weights to a model's classes or samples during training. Depending on the learning algorithm, the loss function is modified to penalize mistakes according to the provided weights, such that higher weights lead to higher penalizing. Because in our study, the minority class is the existing links, these are assigned higher importance by using higher

Table S4: Summary of model hyperparameters. The table provides an overview of the best values for different parameters across five folds for each model. We optimized hyperparameters using a random search. We used the F1-score as the performance metric to evaluate the hyperparameter combinations. RFC: RandomForestClassifier; LR: LogisticRegression; XGB: XGBClassifier.

Model	Parameter	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Range
RFC	n_estimators	300	100	20	100	300	[10, 15, 20, 50, 100, 300]
RFC	min_samples_split	10	10	5	1	3	[1, 2, 3, 4, 5, 10]
RFC	min_samples_leaf	8	12	4	3	4	[3, 4, 5, 6, 7, 8, 9, 10, 12, 15, 20]
RFC	max_samples	0.6	0.9	0.9	0.3	0.6	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
RFC	max_leaf_nodes	8	64	128	64	16	[2, 4, 8, 16, 32, 64, 128]
RFC	max_features	log2	log2	log2	log2	log2	['sqrt', 'log2']
RFC	max_depth	5	5	50	7	5	[3, 5, 7, 10, 20, 30, 40, 50, 60, None]
RFC	criterion	gini	entropy	entropy	gini	gini	['gini', 'entropy']
RFC	bootstrap	True	True	True	True	True	[True]
LR	solver	saga	saga	saga	saga	saga	['newton-cg', 'lbfgs', 'sag']
LR	penalty	l2	l2	l2	l1	l1	['l2', 'none']
LR	max_iter	300	300	300	300	300	[300]
LR	C	0.1	0.01	0.001	0.001	0.001	[100, 10, 1.0, 0.7, 0.5, 0.3, 0.1, 0.01, 0
XGB	tree_method	hist	hist	hist	hist	hist	['hist']
XGB	subsample	0.5	0.1	1.0	0.30000000000000004	0.6	[0.1, 0.2, 0.30000000000000004, 0.4, 0
XGB	reg_lambda	1e-05	100	100	100	0.01	[0, 1e-05, 0.01, 0.1, 1, 100]
XGB	reg_alpha	1	0	0	0.01	1	[0, 1e-05, 0.01, 0.1, 1, 100]
XGB	objective	binary:logistic	binary:logistic	binary:logistic	binary:logistic	binary:logistic	['binary:logistic']
XGB	n_estimators	70	70	80	90	80	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100
XGB	max_depth	7	3	5	5	3	[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, None]
XGB	learning_rate	0.001	0.05	0.3	0.1	0.05	[0.001, 0.01, 0.05, 0.1, 0.2, 0.3]
XGB	gamma	0.3	0	0.1	1	1	[0, 0.1, 0.3, 0.5, 1]
XGB	colsample_bytree	0.2	0.1	0.1	0.1	0.5	[0.1, 0.2, 0.30000000000000004, 0.4, 0
XGB	booster	gbtree	gbtree	gbtree	gbtree	gbtree	['gbtree']

weights. Specifically, each class automatically gets a weight of 1 but can be set with a higher weight. For instance, a class with a weight 2 will have double importance. We aimed for ‘balanced’ weights, which are proportional to the classes’ frequency. For instance, if the training set contains 20 more times non-existing than existing links, then existing links will get 20 times more importance.

Details on model evaluation metrics

In binary classification, the terms “positive” and “negative” refer to the two possible outcomes, where the positive class is typically used to denote the class of interest. Here, the positive class is the presence of a link. There are four possible prediction outcomes defined for binary classification models, computed by comparing the predicted and actual values of the model’s outputs (Fig. S5):

- ***True Positives* (*TP*)**: instances correctly predicted as the positive class. That is, sub-sampled links that were correctly predicted as existing links.
- ***True Negatives* (*TN*)**: instances correctly predicted as the negative class. That is, non-existing links that were correctly predicted as non-existing links.
- ***False Positives* (*FP*)**: instances falsely predicted as the positive class. That is, non-existing links that were incorrectly predicted as existing links. This is also known as a Type I error.
- ***False Negatives* (*FN*)**: instances falsely predicted as the negative class. That is, sub-sampled links that were incorrectly predicted as non-existing links. This is also known as a Type II error.

The output of machine learning models is a probability of a link (Fig. S5). This continuous output is transformed into a categorical (positive / negative) prediction via a threshold. The threshold is typically 0.5, with values > 0.5 considered a link (and values below as a no-link). Therefore, classification into the four categories above depends on the threshold applied. The relationship between true positives, false positives, false negatives, and true negatives are summarized in a *confusion matrix* from which the following metrics are calculated.

ROC-AUC: The area under the receiver operating characteristic curve is a graphical representation of the actual positive rate (y axis) versus the false positive rate (x axis) of a model across different decision thresholds. The ROC-AUC score ranges from 0 to 1, where a score of 1 represents a perfect classification model, while a score of 0.5 represents a model with random guessing.

Although the ROC-AUC is a common measure, the number of true negatives in imbalanced data sets is very large, so even with a substantial number of false positives, the false positive rate might remain relatively small. This means that the ROC curve might not fully capture the cost of misclassifying a substantial number of the minority class instances. A better way to evaluate predictions in imbalanced data sets is by combining precision and recall metrics. Precision and recall provide a more granular understanding of a model's performance. They emphasize the importance of being confident about the most crucial prediction. These metrics act as safeguards against models that might achieve high overall accuracy/ROC-AUC by merely predicting the majority class, which, as discussed in section , is an easy task in imbalanced data.

Precision: The proportion of correctly predicted positive instances out of all positive predictions:

$$\frac{TP}{TP+FP} \tag{S1}$$

Precision is important when false positives are costly. For example, falsely predicting an interaction can lead ecologists to spend research efforts trying to validate a forbidden link in nature.

Recall or sensitivity: The proportion of correctly predicted positive instances out of all actual positive instances. Recall is important when false negatives are costly. For example, not predicting missing links can have consequences for conservation.

$$\frac{TP}{TP+FN} \tag{S2}$$

PR AUC: There is a tradeoff between precision and recall, which depends on the prediction threshold. To evaluate this tradeoff, the area under the PR curve provides a single number that summarizes the overall performance of a model across all possible classification thresholds. Like the ROC-AUC curve, the PR curve is calculated across all thresholds.

F1-score: Another way to evaluate the precision-recall tradeoff is via a balanced measure of performance that reflects the classifier's ability to identify true positive instances while avoiding false positives. The F1-score is defined as the harmonic mean between precision and recall:

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{S3}$$

Specificity: The proportion of correctly predicted negative instances out of all actual negative instances:

$$\frac{TN}{TN+FP} \tag{S4}$$

While recall and precision focus on true positives, specificity focuses on the retrieval of true negatives.

Balanced accuracy: Balanced accuracy aims to balance the prediction ability for links and no-links, particularly in situations where the classes are imbalanced. It is the average of the true positive rate (sensitivity or recall) and the true negative rate (specificity):

$$\frac{recall + specificity}{2} \tag{S5}$$

MCC: The Matthews Correlation Coefficient (MCC) takes into account the balance ratios of the four confusion matrix categories (TP, TN, FP, FN), and is, therefore, a balanced measure that can be used even if the classes are of very different sizes [14]. MCC values range from -1 to $+1$. A coefficient of $+1$ indicates a perfect prediction, 0 indicates no better than a random prediction, and -1 indicates total disagreement between prediction and observation.

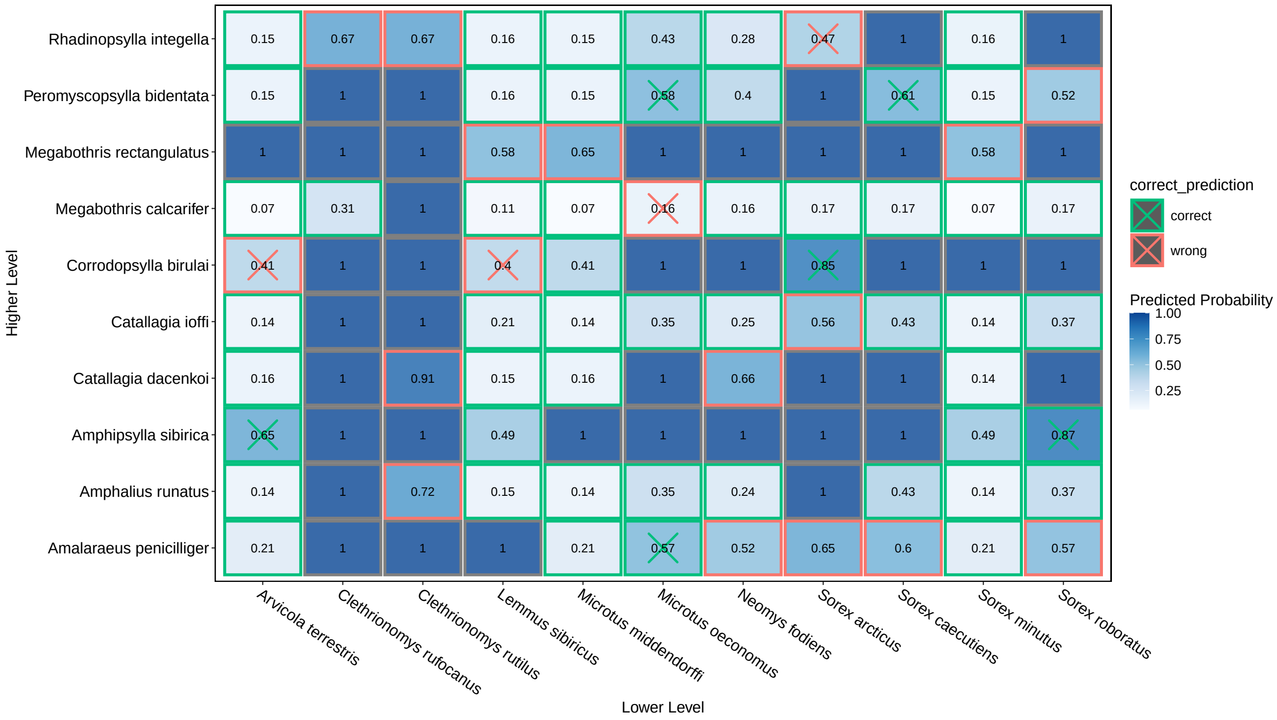

Fig. S5: Link prediction example for a host-parasite network. Values inside matrix cells are the predicted link probabilities. The sub-sampled links are marked with X. Cells above the 0.5 threshold are classified as links. Correct and wrong classifications are colored green and red, respectively. Therefore, true positives are green with X, true negatives are green, false positives are red (no X) and false negatives are red with X.

Exploration of the classification threshold

Given ecological networks' imbalanced and noisy nature, we explored how the classification threshold affects the PR tradeoff. As a first step, we plotted the link probabilities generated by the model. The non-existing links were generally correctly classified ($\approx 0.83\%$ out of 183K non-links; Fig. S6A). This is expected in imbalanced data sets. However, for the subsampled links, only $\approx 0.63\%$ out of 9K were classified correctly (Fig. S6B). This mismatch underlies the tradeoff between precision and recall. Further exploration of the precision-recall tradeoff (Fig. S7) indicated that there is no apparent threshold to choose from, and so we decided to use the common value of 0.5. In a more detailed examination of each community separately, as shown in Fig. S8, we observed a pronounced right-skewed distribution of the sub-sampled links. Specifically, there was a more apparent separation between the two link types in both host-parasite and plant-seed disperser networks.

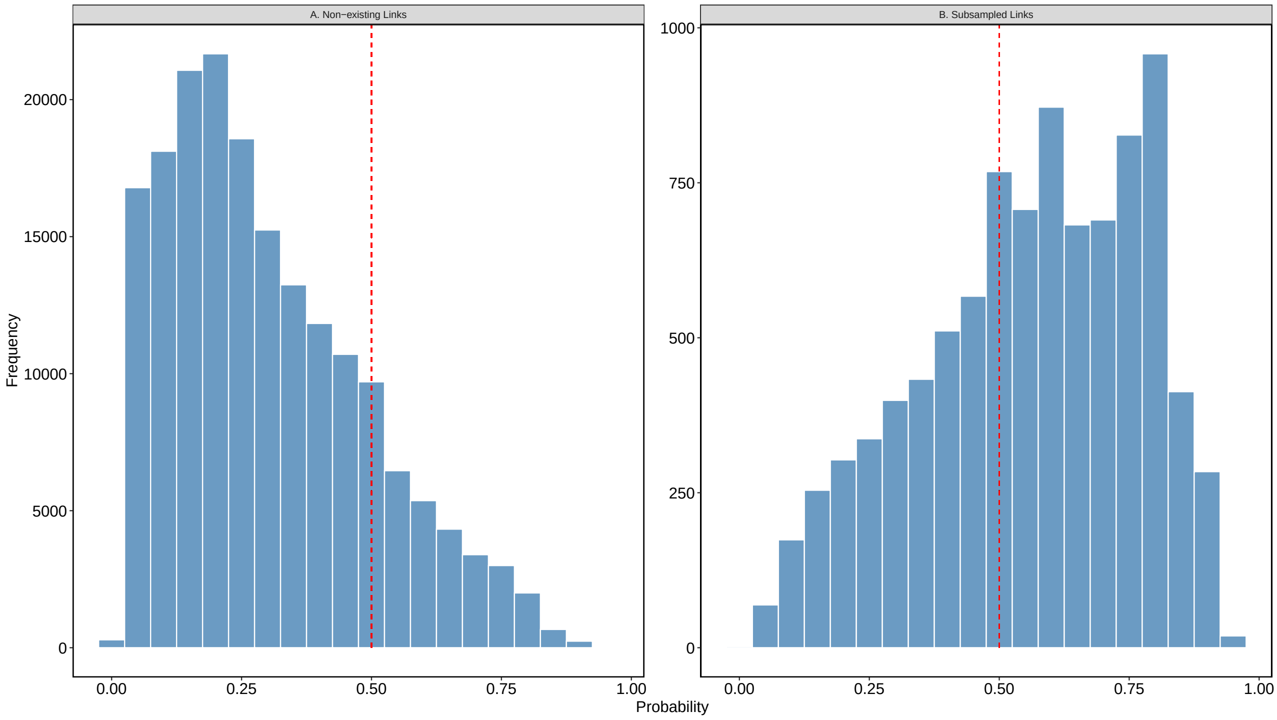

Fig. S6: Distribution of link probabilities obtained from the model. The histograms depict the distribution of the predicted probabilities for our binary classification task, representing the two classes in the test set: non-existing links (A) and sub-sampled links (B). The x-axis represents the predicted probabilities ranging from 0 to 1, while the y-axis represents the frequency of the observations. The dashed red line represents the decision threshold of 0.5.

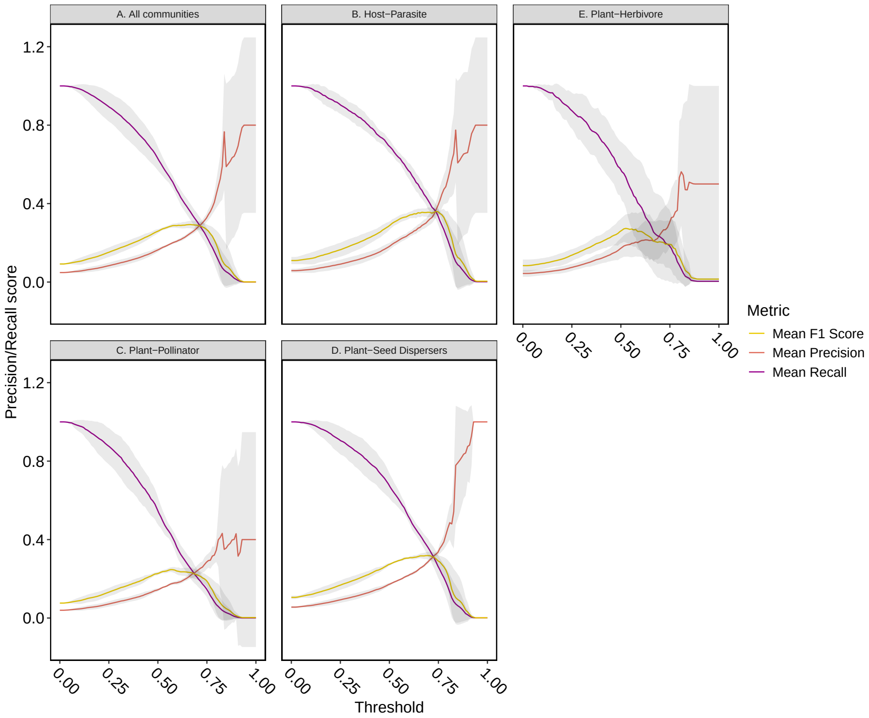

Fig. S7: The precision-recall tradeoff as a function of classification threshold. The tradeoff is presented when testing on all communities (A) or per community type (B-D). Each data point on the curve corresponds to a distinct cutoff threshold value (x-axis).

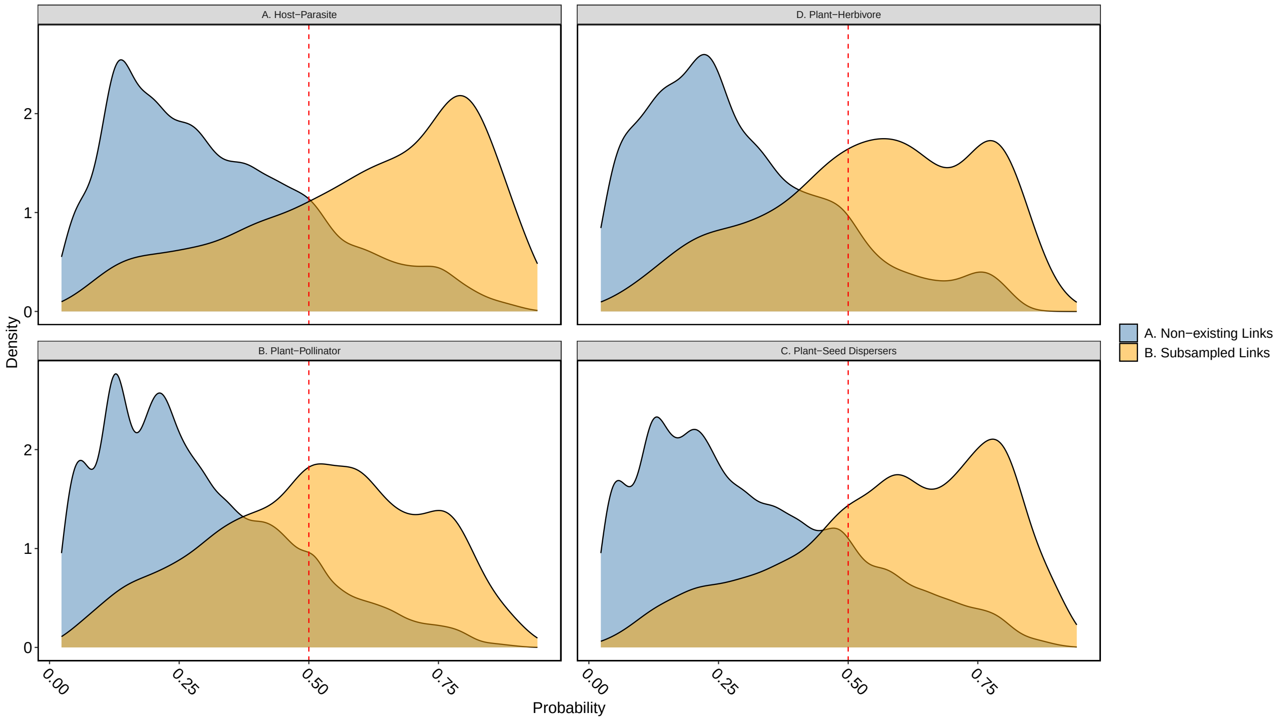

Fig. S8: Distribution of link probabilities across different ecological communities. The kernel density estimate (KDE) curves of the three communities: (A) Host-parasite networks, (B) Plant-pollinator networks, and (C) plant-speed disperser networks. The distributions for non-existing links and sub-sampled links are depicted in blue and orange, respectively. The x-axis represents the predicted probabilities ranging from 0 to 1, while the y-axis denotes the density estimation of the observations. A dashed red line marks the decision threshold of 0.5. The overlap between the blue and orange distributions represents areas of prediction ambiguity. In regions where non-existing links (blue) surpass the decision threshold, false positives emerge, negatively influencing precision. Conversely, in the region where sub-sampled links (orange) fall below the threshold, it results in false negatives, negatively influencing recall.

Appendix References

1. Michalska-Smith, M. J. & Allesina, S. Telling ecological networks apart by their structure: A computational challenge. *PLoS Comput. Biol.* **15,** e1007076. doi:10.1371/journal.pcbi.1007076 (2019).

2. Brimacombe, C., Bodner, K., Michalska-Smith, M., Poisot, T. & Fortin, M.-J. Shortcomings of reusing species interaction networks created by different sets of researchers. *PLoS Biol.* **21,** e3002068. doi:10.1371/journal.pbio.3002068 (2023).

3. Terry, J. C. D. & Lewis, O. T. Finding missing links in interaction networks. *Ecology* **101,** e03047. doi:10.1002/ecy.3047 (2020).

4. Raschka, S. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. *arXiv [cs.LG]* (2018).

5. Kotsiantis, S. B., Zaharakis, I. D. & Pintelas, P. E. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review* **26,** 159–190. doi:10.1007/s10462-007-9052-3 (2006).

6. Sagi, O. & Rokach, L. Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **8,** e1249. doi:10.1002/widm.1249 (2018).

7. Levy, J. J. & O'Malley, A. J. Don't dismiss logistic regression: the case for sensible extraction of interactions in the era of machine learning. *BMC Med. Res. Methodol.* **20,** 171. doi:10.1186/s12874-020-01046-3 (2020).

8. Breiman, L. Random Forests. *Mach. Learn.* **45,** 5–32. doi:10.1023/A:1010933404324 (2001).

9. Biau, G. & Scornet, E. A random forest guided tour. *Test* **25,** 197–227. doi:10.1007/s11749-016-0481-7 (2016).

10. Chen, T. & Guestrin, C. *XGBoost: A Scalable Tree Boosting System* in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* San Francisco, California, USA (Association for Computing Machinery, New York, NY, USA, 2016), 785–794. doi:10.1145/2939672.2939785.

11. Friedman, J. H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **29,** 1189–1232. doi:10.1214/aos/1013203451 (2001).

12. López, V., Fernández, A., Moreno-Torres, J. G. & Herrera, F. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. *Expert Syst. Appl.* **39,** 6585–6608. doi:10.1016/j.eswa.2011.12.043 (2012).

13. Pedregosa, F *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* (2011).

14. Poisot, T. Guidelines for the prediction of species interactions through binary classification. *Methods Ecol. Evol.* **14,** 1333–1345. doi:10.1111/2041-210x.14071 (2023).