Machine Learning-Based Link Prediction for Hotel Network

Yiğit Sevim, Günce Keziban Orman, and Meltem Turhan Yöndem

Abstract—**There is a need for better data modeling that is able to understand the latent interests of the customers in the tourism sector. In this work, network modeling is employed to address this issue. We propose to use link prediction in hotel networks for later development of a recommendation system. SeturTech and Otelpuan, two new data sets from Turkey, are used in the experiments. The baseline is set according to the traditional ranking of fifteen features that categorized as local, global, and embedding, representing the likelihood of the links. Two global features, structural perturbation method and L3, got the best AUPR and AUROC results in the baseline experiments, but still their average precision was low. Also, a machine learning model from all fifteen features together was built. According to F1 scores, extreme gradient boosting outperforms in predicting both newly appearing and already existing links. The deep neural network overfits, especially when forced to find new links. In both networks, it can be said that machine learning-based modeling seems to give more successful results than ranking-based prediction.**

Index Terms—**Link Prediction, Hotel Recommendation, Machine Learning, Feature Ranking**

I. Introduction

It is reported that the tourism sector is one of the fastest growing sectors as well as many nations' economic income depends on it [1]. In this context, a good recommendation system has an important role both in the travelers' experiences and in economic gain. One of the technical challenges that have not been considered by the recommendation systems community in the tourism domain is listed in [1], [2] as the issue of data modeling. It is underlined that there is a need for better data modeling that is able to understand latent interests. In this work, we are interested in building a hotel recommendation system. The hotel recommendation problem can be seen as predicting new possible connections in a complex hotel-user interaction system. A complex network modeling of user-hotel interactions might overcome previously underlined data modeling issues. It allows us to consider both the system-dependent hidden effects and local effects all at once in the analysis.

In the literature, data for hotels is mostly collected from Expedia, Tripadvisor, or similar systems. In most of this data, it is decided whether the users like the hotels or not, according to the user comments or the ratings they give to the hotels. Recommendation systems are then built through a supervised learning process on raw data set. For example, the hotel features and user IDs that prefer hotels from Expedia data is used in [3]. The authors applied Random Forest, Stochastic Gradient Descent, Naive Bayes, Extreme Gradient Boosting, and Ensemble methods. In [4], a comparison of collaborative filtering and matrix factorization methods for hotel recommendation systems is made. Hotel reviews from Tripadvisor were used as data. Although the features of the hotels are included, the hotel preferences of the users can only be understood from the comments they write. For this, an natural language processing (NLP) was also operated. It was stated that matrix factorization found more accurate results but required a lot of time for execution. The authors made it clear in the study that they can only predict the reviews or ratings that users will give to hotels, but they cannot predict whether this will have an impact on bookings or sales. Similarly, in [5], a recommendation system was developed with Tripadvisor data, which are the comments given by users for hotels, and hotel features. As in the previous study, the main challenge is text interpretation and there is no evidence whether the suggestions made in this one turn into sales or reservations.

In this study, we work with the data provided by Setur Servis Turistik A.S., a tourism agency operating in Turkey. Our main goal is to build an accurate hotel recommendation system by overcoming the previously mentioned issues. We extract a network to represent the hotel preferences of the customers and develop a recommendation system on this network through link prediction. We propose a supervised learning-based link prediction model on real-world data. Thus, this study differs from previous hotel recommendation system studies in terms of both data modeling and the nature of the data used. In order to ensure that our model is not data-dependent, we have also used another new dataset showing the hotel choices of users from Otelpuan, a website operating in Turkey. These two data sets are previously used in our work [6].

Previous studies suffer from the problem of not using the hidden effects mentioned earlier, as they do not take into account hotel-customer, hotel-hotel, or customer-customer interactions. However, recently, the work consulting network modeling has also been proposed by Kaya [7]. Kaya used the customer-hotel preferences obtained from Tripadvisor. First, a bipartite network was extracted between the people and the hotels according to the likes/dislikes left by the people for the hotels on Tripadvisor. Second, a projected network was extracted. Third, link prediction was made with the most widely known method in the literature; ranking of the link scoring features such as adamic adar, jaccard, or resource allocation. This study is similar to our proposal in terms of modeling the data under the form of a network. However, basic ranking of the link scoring features can be misleading in many cases because a link can still take a high score but cannot have a place in the ranking order.

Manuscript received October 12, 2022; revised October 21, 2022. This work was supported in part by the SeturTech R&D, Turkey.

Y. Sevim is a researcher in the SeturTech R&D Department, İstanbul, 34770, Turkey (yigit.sevim@setur.com.tr).

M. Turhan Yöndem is a senior research consultant in the SeturTech R&D Department, İstanbul, 34770, Turkey (meltem.yondem@setur.com.tr).

G.K. Orman is an Assistant Professor of Computer Engineering Department, Galatasaray University, İstanbul, 34349, Turkey (korman@gsu.edu.tr).

Link prediction is one of the most studied sub-domains in network science. The main purpose is to find the missing links with the highest likelihood of appearing by using the topology and algebra of the network structure [8]. There are hundreds of different related approaches. We can categorize the methods into three parts. First, *traditional methods* calculate a score of a feature for each possible link that are not seen in the network, based on a strategy [9]–[11]. Afterwards, these scores are ranked from largest to smallest, and the desired number of links is selected in order to be predicted. The scores quantify the likelihood of links with different methods using network topology. Second, *machine-learning-based methods* calculate the scores of all links by using several different methods as in the *traditional* ones. However, this time, instead of making a simple ranking, each of these scores is used as a feature of the link set. If a link already exists in the network, its label is 1, otherwise it is 0. The link prediction task is executed as a supervised learning experiment [12], [13] from a feature set of likelihood scores. Those methods differ from each other in terms of both the supervised learning experiment design and the machine learning algorithms they use. Finally, *graph embedding techniques* are held [14], [15] in recent works. Those methods first project network information onto low-dimensional euclidean space, then find possible links by using distances in the euclidean space or features from the embedding strategy. The embedding can be done by either graph algebra or deep learning. Deep-learning based embedding is successful, especially when the network size is large.

We predict links with both traditional and machine learning-based methods. We also deploy methods using algebra-based embedding techniques for scoring. However, we did not use deep learning-based embedding methods because the networks we used were not large.

Our main contributions are as follows:

- We are making link predictions to develop a hotel recommendation system based on network modeling on two new real-data sets: SeturTech and Otelpuan.
- We measure the likelihood of a missing link between two nodes by using fifteen different features with different scoring strategies that we separate into three categories: local, global, and embedding.
- We measure the success of the features through traditional ranking-based prediction to create a baseline for the data we use.
- We are building a link prediction model that uses these fifteen features with supervised learning by various machine learning algorithms.

The rest of the article includes the details of data sets, link prediction features, and machine learning algorithms in section II. Then, we explain the two experiments, ranking-based and machine-learning, and all related results in section III. Finally, we summarize the work with future perspectives in section IV.

II. Method

A. Data Sets

We have used two distinct data sets in this study. The first data set contains historical hotel sales in Setur between 2013 and 2021. Setur Servis Turistik AS provides travel bookings for air, land, and sea travel for both individuals and businesses. Setur also provides services for duty-free goods, and it is one of the leading tourism agencies in Turkey. The data was provided by SeturTech R&D department and has features for customer and hotel id's, dates of purchase and entry into the hotel, hotel features such as location, services, and customer features such as age, gender, etc. There are 45332 unique customers and 1552 unique hotels, with a total of 57262 interactions between customers and hotels.

The second data set was collected by the SeturTech R&D department from the Otelpuan.com website by using web scraping methods. Otelpuan.com was founded in 2008 to inform customers about tourism services by collecting ratings and comments on their website. The data set contains the customer ratings in the 1–10 range for the hotels on Otelpuan.com. This data has 170517 unique customers and 959 unique hotels. There are 179996 total interactions between customers and hotels. The average interactions between randomly selected customers and hotels are 2.44 and 2.10 for SeturTech and Otelpuan networks respectively.

B. Link Prediction Features

In the next part, we explain the similarity/distance metrics that assess the likelihood of having a link between any pair of nodes (u, v). We categorized those metrics into three categories: local, global, and embedding according to their essential techniques, which are used for link prediction tasks.

1) Link Prediction with Local Information: According to Kovacs et al. state-of-the-art network based link prediction algorithms rely on the triadic closure principle (TCP) [16]. This principle explains the tendency of having a link between two nodes that share common neighbors. This concept relies on neighborhood, a well-known topological definition on complex networks. Let $G = (V, L)$ be a network with V is its node set and L is its link set. Neighborhood, $N(u)$, or (N_u), of a node $u \in V$ is the set of nodes directly connected to u. $N(u) = \{v \in V : (u, v) \in L\}$.

Definition II.1. Common Neighbors (CN) is the size of the set of common neighbors between any two nodes [17]. Its formula is given in Eq.1.

$$s(u, v) = |N_u \cap N_v| \tag{1}$$

More generally, the higher the number of degrees, the more possible to have higher CN for the nodes. Thus CN has a tendency of being high for any two hub nodes.

Definition II.2. Adamic Adar (AA) counts the total number of neighbors of all common neighbors [18]. But it depresses the score by logarithmic function for demoting the scores of higher degree nodes. Shortly, it penalizes the scores for hub neighbors. Its formula is given in Eq.2.

$$s(u, v) = \sum_{i \in N_u \cap N_v} \frac{1}{\log_2(|N_i|)} \tag{2}$$

Definition II.3. Resource Allocation (RA) is almost the same with AA [19]. It also counts the total number of neighbors of all common neighbors. But differently from AA, it considers the degrees not their logarithms. Its formula is given in Eq.3.

$$s(u,v) = \sum_{i \in N_u \cap N_v} \frac{1}{|N_i|} \tag{3}$$

Definition II.4. Jaccard Coefficient (JC), originally developed for comparing two sets [20]. It is the ratio of the number of common neighbors to the number of all neighbors of two nodes. The formula is given in Eq.4.

$$s(u,v) = \frac{|N_u \cap N_v|}{|N_u \cup N_v|} \tag{4}$$

Definition II.5. Sørrenson/Dice Index (Dice) measures the common parts of the neighborhoods and normalizes it with the size of the neighborhoods of two studied nodes [21]. If the neighborhoods have many nodes in common but also the common neighbors have many other links to the outside of the common neighborhood, Dice becomes lower than JC. It penalizes being a hub as well. The formula is given in Eq.5.

$$s(u,v) = \frac{2 \cdot |N_u \cap N_v|}{|N_u| + |N_v|} \tag{5}$$

Definition II.6. Cannistraci-Alanis-Ravasi index (CAR) is the sum of the number of common neighbors of two nodes each having neighbors in common with those nodes [22]. Its formula is given in Eq.6.

$$s(u,v) = \sum_{i \in N_u \cap N_v} 1 + \frac{|N_u \cap N_v \cap N_i|}{2} \tag{6}$$

Definition II.7. CAR-based Adamic and Adar (CAA), is a hybrid metric of the AA with CAR strategy [22]. It merges two strategies of favoring clique-like neighborhoods with the penalization of being hub. The formula is given in Eq.7.

$$s(u,v) = \sum_{i \in N_u \cap N_v} \frac{|N_u \cap N_v \cap N_i|}{\log_2 (N_i)} \tag{7}$$

Definition II.8. Another hybrid metric is CAR-based Resource Allocation (CRA) [22]. It merges the two strategies of CAR with RA which are explained previously. Its formula is given in Eq.8.

$$s(u,v) = \sum_{i \in N_u \cap N_v} \frac{|N_u \cap N_v \cap N_i|}{|N_i|} \tag{8}$$

Definition II.9. Preferential Attachment (PA), is the multiplication of degrees of two nodes [17]. PA promotes the nodes having higher degree. It assumes that the famous nodes should have more probability of connecting with each other. The formula is given in Eq.9.

$$s(u,v) = |N_u| \cdot |N_v| \tag{9}$$

Definition II.10. CAR-based Preferential Attachment (CPA), merges the strategies of CAR and preferential attachment [22]. Its formula is given in Eq.10.

$$s(u,v) = e_u.e_v + e_u.CAR(u,v) + |e_v.CAR(u,v) + CAR(u,v)^2 \tag{10}$$

Here, $e_u = |N_u \backslash (N_u \cap N_v)|$ and $e_v = |N_v \backslash (N_u \cap N_v)|$ is the number of the neighbors that are not common neighbors of u and v, and $CAR(u,v)$ is the CAR score between nodes u and v.

2) Link Prediction with Global Information: In local methods, the metrics completely focus on the common neighborhood which was based on the TCP idea. Here, we explain the metrics using other strategies related to network topology.

Definition II.11. L3 link predictor (L3), considers network paths of length three [16]. Its formula is given in Eq.11.

$$s(u,v) = \sum_{ij} \frac{a_{\text{ui}}.a_{\text{ij}}.a_{\text{jv}}}{\sqrt{k_i.k_j}} \tag{11}$$

Here, a_{ui} is 1 if there is a link between the nodes u and i. And k_i is the degree of node i. Since the third level neighbors numbers are exponentially larger than the second level ones, the metric applies a degree normalization strategy. It also avoids the biased high scores coming from the hub nodes which are naturally building shortcuts and increases the number of third level neighbors for entire nodes.

Definition II.12. Structural perturbation method (SPM), focuses on perturbing the adjacency matrix and observing the change of eigenvalues provided the fixed eigenvectors [23]. This technique is similar to the first-order perturbation in quantum mechanics. Basically, it produces the scores, which are similar to previously explained similarities, for all links based on the perturbation of removal links from the adjacency matrix of the original network.

3) Link Prediction with Embedding: Beyond the usage of TCP principle or network structural information, there are other techniques of link prediction which transform the network into the lower dimensional euclidean space. Such a transformation is called graph embedding. There are several different techniques of graph embedding. Here we focus on the ones which are using graph algebra.

Definition II.13. Isometric mapping (ISOMAP), uses one of the traditional graph embedding techniques [24]. The studied network, $G = (V, L)$, is first transformed to a distance matrix D of its nodes in which each member d_{uv} of D is the shortest distance between the nodes u and v from V. Then D is transformed to a lower dimensional matrix $L \in \mathbb{R}^l$ with Multidimensional scaling based on non-linear embedding method, MDS. Here l is the new dimension that G is transformed to. MDS tries to keep original distance d_{uv} between the node pairs and generates new vectors $x_1, x_2, ..., x_n$ for each node whose lengths are l. $x_1, x_2, ..., x_n$ is found as a minimizer of some cost function $\text{mi}n_{x_1,x_2,...,x_n}(d_{\text{uv}} - ||x_u - x_v||)^2$. Once MDS generates new lower dimensional vectors for each node, then ISOMAP calculates basic euclidean distance between the nodes as their dissimilarities.

Definition II.14. Laplacian Eigenmaps (LEIG), uses a minimization function that can be solved by the generalized eigenvalue problem [25]. Hence, it first generates the laplacian matrix of the original network, then spectral decomposition of the corresponding laplacian matrix is computed. LEIG finds l eigenvalues and eigenvector with l is the number of new dimensions. After embedding, the link prediction is again done by regarding euclidean distance of the node pairs.

Definition II.15. Centered and non-centered Minimum Curvilinear Embedding (MCE) and (ncMCE) respectively, are two network embedding techniques using the distances in the minimum spanning tree of studied networks [26]. Both methods first generate the minimum spanning tree, MST of corresponding G, then computes the distances of every pair of nodes in the MST. These distances under the form of distance matrix are called the kernel. In the algorithm if centering is not chosen, the ncMCE performs an economy size singular value decomposition of the distance matrix. Otherwise an algebraic operation is performed for kernel centering at first and then the decomposition is done. Finally the new lower dimensional space of G is produced by the transpose of the product of computed singular values with right singular vectors with the algebraic corrections.

C. Machine Learning Algorithms

In our binary classification task, we represent the training and test sets with the scores of the fifteen link prediction features explained in the previous section. We compared the results of various machine learning methods. We used tree-based techniques such as decision tree (DT), gradient boosting (GB), and extreme gradient boosting, a.k.a. XGBoost (XGB). In our experiment, we also utilized a deep neural network (DNN) to predict the positive links. During the training process, by using features of the data, the tree algorithms try to separate classes in the most homogeneous way possible into the most compact tree-structure possible. Among them, the decision tree is the most basic one. It uses a single tree to form the predictor model. GB makes use of several trees. Each tree is built sequentially, and the outcome of the previous tree influences the next tree. Thus, by improving the previous tree, it overcomes the drawbacks of using one single tree. The XGB classifier is built in the same way as gradient boosting. The primary distinction between both methods is that XGB can use regularization metrics to improve performance.

Besides the tree-based methods, we used two-layered simple multilayer perceptrons as a DNN. A DNN in general is made up of layers of neurons, each of which receives input from the previous layer, performs a simple computation in an activation function, and then sends the result to the next layer. The hyper-parameters of all algorithms are optimized by a tuning process. When the tree-based algorithms are run with default parameter values, they may overfit. We regularize by limiting the maximum-depth and by sub-sampling for boosting-based GB and XGB. Finally, we applied the most performing versions of the mentioned algorithms.

III. Experiments and Results

We propose a framework that models the hotel-customer data set in the form of an appropriate complex network and finds the proper hotel suggestions. The flowchart of this framework is shown in Fig. 1. The different steps of this framework are numbered in the figure. Accordingly, step 1 is dedicated to modeling the raw data set in the form of a network.

First, hotel-customer interactions are transformed to a bipartite network as $G_0 = (V_H, V_U, L)$ (see step1 of Fig. 1). V_H is the hotel node set whose members are the node representations of the hotels. V_U is the customer node set whose members are the node representations of the customers. L is the link set, whose members are the node pairs between V_H and V_U. If a customer visits or prefers a hotel, there is a link between their represented nodes. Hotel recommendation can be completed directly from the bipartite network model by identifying appropriate missing links between a hotel type node in V_H and a customer type node in V_U. However, link prediction techniques for bipartite networks are case-specific and limited [27], whereas link prediction in uni-partite networks has a number of works dedicated to it [9]–[11]. Thus, we transform bipartite network into uni-partite ones in our experiments as it was done in [7].

After the bipartite network extraction, the framework splits into two major branches: Experiment 1 (Exp.1) and Experiment 2 (Exp.2). These two parts are dedicated to different link prediction processes. In the next sections, we will explain them in detail.

A. Experiment 1: Link Prediction via Feature Ranking

In the Exp.1, first an auxiliary projected network (G) with hotel nodes from V_H is extracted (see Step2 of Exp.1 in Fig. 1). In the G, links are formed between hotel pairs if each hotel in a pair are linked to at least one common customer in the bipartite network G_0. Then the link prediction process is handled as the *traditional methods*. This is a semi-supervised learning technique [10]. Its steps are shown in the Fig. 1. First, a training network G_{train} is assigned by removing N randomly selected links from the projected hotel network (see Step3 of Exp.1 in Fig. 1). In our experiments, N is chosen as the %20 of the existing links in both studied networks; SeturTech and Otelpuan. On the G_{train} network, we calculate all of the link prediction features described in the II-B section. The scores are calculated for all possible missing links of G_{train} (Step4 of Exp.1 in Fig. 1). Then the scores of each feature is ranked. The first N links with the highest scores have been predicted. For each feature, the predicted links were evaluated as true or false predictions by determining whether or not the projected hotel network contained the predicted links. Then the performance of the link prediction methods is calculated for each feature (Step5 of Exp.1 in Fig. 1).

The performance evaluation of ranking-based link prediction features is a challenging issue itself. The precision or recall are the most commonly used metrics [28]. However, they consider confusion matrix which is built according to the exact matching of the links that are ranked by their prediction scores with the links in N. They can be misleading since many true links might stay behind in the rankings although they have high scores. Hence, ranking-free metrics such as AUROC, AUPR, and average precision can give a more robust evaluation state [29]. We measure the performance of link prediction methods with these three metrics. The area under the ROC curve, a.k.a the AUROC of a link prediction method, can be interpreted as the probability that the method assigns a higher score to a randomly selected link from N, the removed link set for testing, than to a randomly selected link from the unobserved link set in the G_{train}. The better the link prediction, the higher the values of AUROC. The area under the precision-recall curve, a.k.a. the AUPR, is a good metric for the cases where there is an imbalance in the predicted class, as in the case of a link prediction problem.

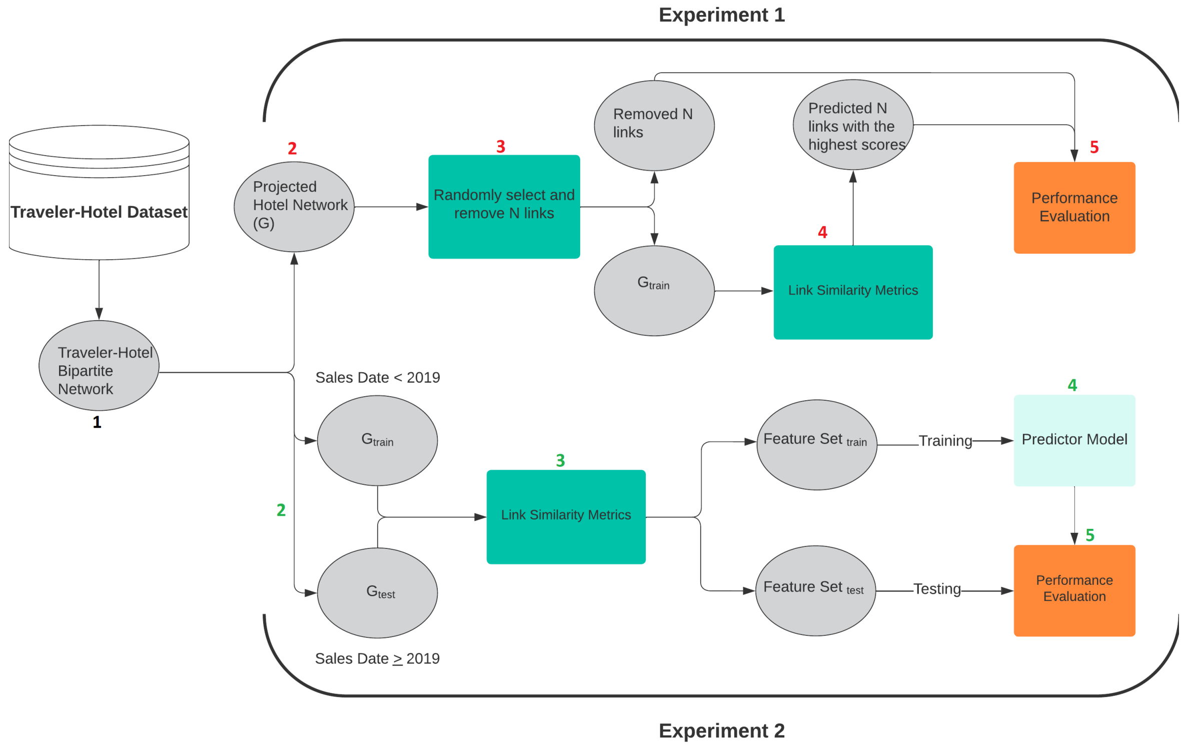

Figure 1. Flowchart of the link prediction frameworks

The last evaluation metric we use is the average precision at the point where recall reaches its maximum value of 1. The details of these metrics can be found in [29].

The results of AUPR, AUROC, and averge precision showing the performance of the ranking-based experiments on the SeturTech and Otelpuan networks are given in tables I and II, respectively. In both networks, SPM and L3 methods using global information gave the best results according to all three metrics. ISOMAP, LEIG and MCE, the methods using graph embedding, seem to obtain the lowest performance scores for both networks. Among other prediction methods, RA also performed well in both networks.

Table I
SETURTECH LINK PREDICTION RESULTS VIA FEATURE RANKING

Category	Method	AUPR	AUROC	Avg.Prec.
Local	CN	0.41	0.95	0.05
	AA	0.45	0.95	0.05
	RA	0.54	0.96	0.05
	JC	0.25	0.93	0.04
	DICE	0.25	0.93	0.04
	CAR	0.41	0.93	0.05
	CAA	0.41	0.93	0.05
	CRA	0.48	0.94	0.05
	PA	0.34	0.92	0.05
	CPA	0.39	0.93	0.05
Global	**L3**	**0.80**	**0.99**	**0.06**
	SPM	**0.92**	**0.99**	**0.06**
Embedding	ISOMAP	0.03	0.75	0.02
	LEIG	0.01	0.50	0.01
	MCE	0.02	0.63	0.02

There is a high variance between the AUPR obtained by different featues in the SeturTech network. Some features like LEIG and MCE can get values like 0.01, while JC or DICE can get 0.25. The highest score was 0.92. In the Otelpuan network, the variances are high for AUPR but not as high as for SeturTech. In particular, all local methods gave similar results. Briefly, success will vary dramatically depending on the feature to be chosen.

It seems different link prediction features scored differently. This gave different success rates in prediction. AUPR and AUROC show that good scores with global metrics for both networks can be obtained. Nonetheless, the average precision shows that there are too many false positives when the recall is 1, that is, when the feature threshold is set to find all missing N links. In this case, we assume that many of the missing links in the network have similar and high scores, but there are too many false positives when a ranking-based prediction is made. In other words, ranking-based link prediction based on a single feature will not be sufficient. It can be complementary to use all of these features that use different types of information in the network. Here, each feature use different types of information. For stronger link prediction, a machine learning model that uses all features together rather than a simple ranking of a single feature may be helpful. Therefore, we designed Exp. 2.

B. Experiment 2: Link Prediction via Machine Learning

In Exp. 2, we use the fifteen features we described earlier in a machine learning model for link prediction. We split training and test sets as it is done in [12]. The details of this experiment can be seen in Fig. 1. Accordingly, we defined two auxiliary projected network with hotel nodes from V_H. The first projected hotel network, G_{train}, was formed by filtering the bipartite network by sales date between customer-hotel pairs (see step2 of Fig. 1). In this scenario, the filter date was 2019. Every customer-hotel link that occurred before 2019 was used to form the G_{train} projected network. Every customer-hotel link that occurred in 2019 and after 2019 was used to form the G_{test} projected network.

Table II
OTELPUAN LINK PREDICTION RESULTS VIA FEATURE RANKING

Category	Method	AUPR	AUROC	Avg.Prec.
Local	CN	0.78	0.96	0.12
	AA	0.78	0.96	0.12
	RA	0.79	0.96	0.12
	JC	0.72	0.95	0.12
	DICE	0.72	0.95	0.12
	CAR	0.77	0.96	0.12
	CAA	0.77	0.96	0.12
	CRA	0.77	0.96	0.12
	PA	0.78	0.96	0.12
	CPA	0.78	0.96	0.12
Global	**L3**	**0.92**	**0.99**	**0.13**
	SPM	**0.97**	**0.99**	**0.13**
Embedding	ISOMAP	0.12	0.83	0.07
	LEIG	0.32	0.89	0.09
	MCE	0.14	0.79	0.07

After training and test networks split, we calculate all link prediction features for all possible links, including appearing and missing ones for each network separately (see step3 of Fig. 1). For each projected network, link prediction features were used to create the $FeatureSet_{train}$ and $FeatureSet_{test}$. In these sets, if the link is already seen in the relevant network, its label is 1, if it is not seen, its label is 0. We used the $FeatureSet_{train}$ to train all of the machine learning algorithms mentioned in the II-C , and the $FeatureSet_{test}$ to validate the models' performance. In the validation part, we used 2 different test sets. The first test set contains all current and non-existent possible links that came from G_{test}. This test set was named as *all* test set. The second test set only contains newly occurred links (not observed on the G_{train}) and newly non-existing links (observed on the G_{train} but not on the G_{test}). This test set was named as *sampled* test set. We did these two different analyses to notice a possible overfit of the models. A good model should be successful when predicting the changing parts of the network. To evaluate the performance of the trained models, we used accuracy, precision, recall, and F1 scores. In the table III performance of this experiment is shown for each data set, for each model, and for each test type.

Table III
LINK PREDICTION RESULTS VIA MACHINE LEARNING

Data Set	Model	Test Type	Acc.	Prec.	Recall	F1 Score
SeturTech	DT	Sampled	0.98	0.97	0.95	0.95
		All Test	0.99	0.97	0.53	0.69
	GB	Sampled	0.96	0.89	0.95	0.92
		All Test	0.99	0.89	0.49	0.63
	XGB	Sampled	0.98	0.99	0.95	**0.97**
		All Test	0.99	0.99	0.58	0.73
	DNN	Sampled	0.73	0.02	1.00	0.05
		All Test	0.99	0.98	0.69	**0.81**
Otelpuan	DT	Sampled	0.95	0.98	0.67	0.79
		All Test	0.99	0.98	0.39	0.55
	GB	Sampled	0.95	0.99	0.65	0.79
		All Test	0.99	0.99	0.37	0.54
	XGB	Sampled	0.96	0.97	0.98	**0.90**
		All Test	0.99	0.93	0.44	0.59
	DNN	Sampled	0.92	0.16	1.00	0.27
		All Test	0.99	0.80	0.58	**0.63**

All of the *sampled* test F1 scores, except the one with DNN, exceeded 0.90. The best score is obtained by XGB.

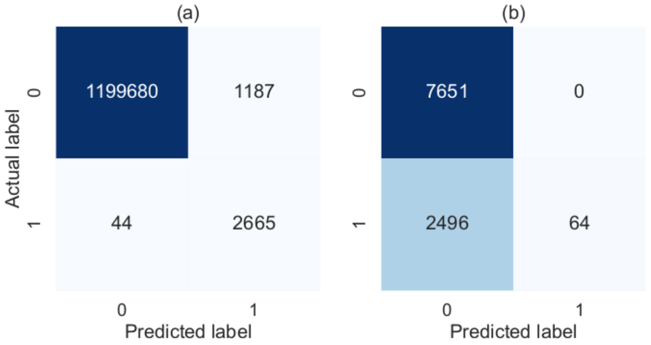

Figure 2. Confusion matrices obtained with DNN in Seturtech *all* (a) and *sampled* (b) test experiments.

On the contrary, in the DNN *sampled* test, the precision and F1 scores were the lowest. However, this test reached the perfect recall score of 1. For SeturTech, among the tests done with *all* test data, the best performing model is DNN, with an F1 score of 0.81. It is quite larger than other algorithms' results. The most striking difference between *all* and *sampled* is obtained by DNN. While we achieve successful results in the *all* test, we obtain failed results on the *sampled* test set which has only new 1s and new 0s. This shows us that this algorithm is overfitting. That is, DNN got used to the link status in the training data and could not learn whether this status changed in the test data.

This fact can more clearly be observed in related confusion matrices given in Fig. 2. In the *sampled* test, the number of true positive links is too low (see Fig. 2-b) while the number of false positives is too high. One reason for this result could be that the number of positive links in the training set is not enough for DNN to learn the linking mechanism. DNN needs lots of data to build an accurate model. Unlike the DNN results, other methods are much more successful with the *sampled* set than the ones with the *all* set. This means that the features that we used are compatible with such link prediction. We assume that the low performance on *all* set of those algorithms is due to the high number of false negatives, which creates the biggest challenge in link prediction for all methods. Since links within nodes in large complex networks are rare, detecting the true links among the vast number of possible links is significantly more valuable than detecting non-existing links correctly. Thus, reducing the number of false negative predictions were our main objective while training the models.

Model performances for the tests done with Otelpuan data are similar to Seturtech results as it can be seen in table III. The *sampled* test on DNN is the worst performing one in terms of accuracy and precision while DNN is the best in terms of recall and F1 score among the experiments done with *all* test set. The best performance is obtained by XBG for *sampled* and by DNN for *all*. Among all results, XGB on *sampled* takes the most significant one with a F1 score of 0.90. The related confusion matrices of XGB is shown in Fig. 3. True positive values are also high when XGB tries to predict the new 1 and new 0 labels as well (see Fig. 3-b). These confusion matrices demonstrate that the algorithm does not memorize but learns enough to predict the link change in the system.

We can see that in Otelpuan, all methods have lower

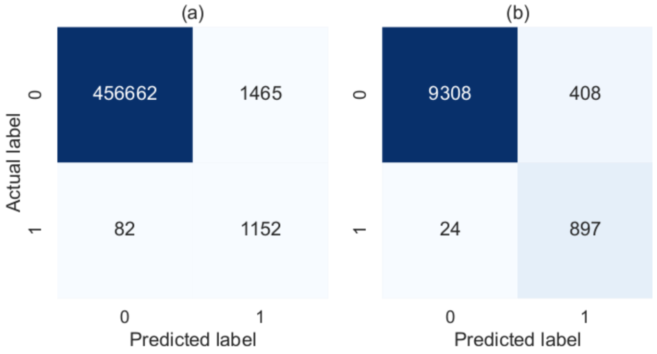

Figure 3. Confusion matrices obtained with XGboost in Otelpuan *all* (a) and *sampled* (b) test experiments.

success than those in SeturTech. Compared to SeturTech, Otelpuan is a network with higher link density. When we examine the topologies of the two networks, the transitivity of SeturTech (~ 0.22) is lower than Otelpuan (~ 0.67), and its eigenvector centralization (~ 0.91) is higher than Otelpuan (~ 0.81). In brief, these two networks have topologically different properties. Indeed, in Exp.1, we noticed that there were also score differences between the success of same features on two networks (see table I and II). From here, the lower performance of Otelpuan compared to SeturTech in Exp.2 can be due to the difficulty of link predictability level between these two networks. It seems that SeturTech is a more prone network for link prediction with the features we use.

In this work, we did not apply any feature engineering steps before machine learning experiments. We used fifteen link prediction features as a raw data set. However, some of those features use the similar information. For instance, all local features are based on the TCP principle. This can cause redundancy in the learning phase. Intelligent feature reduction can increase the success of some algorithms. Some of the leading results from evaluating these two experiments together are as follows:

- For link prediction in both SeturTech and Otelpuan networks, a machine learning-based model may be more appropriate than a ranking-based approach.
- When deciding whether there will be a link between two nodes, global features using general information are better than local methods. Graph algebra-based embedding methods are the most unsuccessful ones.
- This modeling can be used for the hotel recommendation system. The preferences of other people who have made the same choices as themselves or who are similar to them in the system can be offered to the customers. This type of recommendation can include more interesting hotel offers than attribute similarity-based collaborative filtering.

IV. Conclusion

In this work, we propose a link prediction framework for hotel recommendations to tourism customers. We work with two new data sets; SeturTech, which is one of the foremost travel agencies in Turkey, and Otelpuan, which is a website for travel organizations. The framework is based on the modelling of hotel-to-hotel networks. The links are represented by fifteen different features, each of which takes a score showing the likelihood of the studied link. Those features are calculated by different methods which use different network properties. We categorized them as local, global, and embedding. We performed two different experiments. As a baseline, the first experiment is dedicated to evaluating the performance of traditional ranking-based link prediction. The results demonstrate that global features, L3 and SPM are two foremost features with the highest AUROC and AUPR, but they are still not good enough when predicting links with simple ranking. We have concluded that both local, global and embedding types of information can be complementary. Hence, in experiment 2, we designed a supervised learning task using all fifteen features. The deep neural network and XGBoost methods achieved the most accurate link prediction according to their F1 scores but deep neural network seemed to overfit. It was not successful when predicting newly appearing links but successful to predict already existing ones.

Our experiments show that the usage of different features together in a machine learning model can result in accurate link predictions. Adding hotel attributes besides the network-based topological features can be complementary and can result in even higher accuracy. Some future perspectives of this work can be listed as first, using other hotel data sets from Tripadvisor and Expedia, which have been worked on in the literature before by comparing our approach with previous approaches. Second, graph neural network-based embedding techniques can be used. They can be used both to produce new link prediction features and directly to develop a new link prediction method as well. Third, there are other features that we did not use here, such as the Katz index or hub promoted index, etc. They can be added to the feature set. In this study, we made a purely analytical estimation. However, we did not examine their corresponding results in business. We wanted to use all the data we had for modelling. In the following steps of this study, it will be complementary to make predictions with the model we have established for the coming months and observe how these predictions address the needs of customers in the real world in a live system. In addition, when these analytical results are converted into a live system, it will be necessary to deal with the updating and renewal of the models we have established.

References

[1] M. R. Dareddy, “Challenges in recommender systems for tourism,” in *Proceedings of the Workshop on Recommenders in Tourism co-located with 10th ACM Conference on Recommender Systems (RecSys 2016), Boston, MA, USA, September 15, 2016*, ser. CEUR Workshop Proceedings, D. R. Fesenmaier, T. Kuflik, and J. Neidhardt, Eds., vol. 1685. CEUR-WS.org, 2016, pp. 59–61. [Online]. Available: http://ceur-ws.org/Vol-1685/paper11.pdf

[2] J. Adamczak, G. P. Leyson, P. Knees, Y. Deldjoo, F. B. Moghaddam, J. Neidhardt, W. Wörndl, and P. Monreal, “Session-based hotel recommendations: Challenges and future directions,” *CoRR*, vol. abs/1908.00071, 2019. [Online]. Available: http://arxiv.org/abs/1908.00071

[3] A. A. Mavalankar, Ajitesh Gupta, Chetan Gandotra, and Rishabh Misra, “Hotel recommendation system,” 2019. [Online]. Available: http://rgdoi.net/10.13140/RG.2.2.27394.22728/1

[4] K. Khaleghi, Ryan; Cannon and R. Srinivas, “A comparative evaluation of recommender systems for hotel reviews,” *SMU Data Science Review*, vol. 1, no. 4, 2018.

[5] B. Ray, A. Garain, and R. Sarkar, “An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews,” *Applied Soft Computing*, vol. 98, p. 106935, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620308735

[6] Y. Sevim, G. K. Orman, and O. Kılıçlıoğlu, “A link prediction framework for hotel recommendations,” in *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2022*, London, U.K., pp. 64–69.

[7] B. Kaya, “Hotel recommendation system by bipartite networks and link prediction,” *Journal of Information Science*, vol. 46, no. 1, pp. 53–63, 2020. [Online]. Available: https://doi.org/10.1177/0165551518824577

[8] L.-N. D. and K. J, “The link-prediction problem for social networks,” *Journal of The American Society For Information Science and Technology*, vol. 58, no. 7, p. 1019–1031.

[9] V. Martínez, F. Berzal, and J.-C. Cubero, “A survey of link prediction in complex networks,” *ACM Comput. Surv.*, vol. 49, no. 4, dec 2016. [Online]. Available: https://doi.org/10.1145/3012704

[10] L. Lü and T. Zhou, “Link prediction in complex networks: A survey,” *Physica A: Statistical Mechanics and Its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037843711000991X

[11] E. C. Mutlu, T. Oghaz, A. Rajabi, and I. Garibay, “Review on learning and extracting graph features for link prediction,” *Machine Learning and Knowledge Extraction*, vol. 2, no. 4, pp. 672–704, 2020.

[12] H. R. de Sá and R. B. C. Prudêncio, “Supervised link prediction in weighted networks,” in *The 2011 International Joint Conference on Neural Networks*, 2011, pp. 2281–2288.

[13] D. Malhotra and R. Goyal, “Supervised-learning link prediction in single layer and multiplex networks,” *Machine Learning with Applications*, vol. 6, p. 100086, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666827021000438

[14] M. Wang, L. Qiu, and X. Wang, “A survey on knowledge graph embeddings for link prediction,” *Symmetry*, vol. 13, no. 3, 2021. [Online]. Available: https://www.mdpi.com/2073-8994/13/3/485

[15] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo, “Knowledge graph embedding for link prediction: A comparative analysis,” *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 2, jan 2021. [Online]. Available: https://doi.org/10.1145/3424672

[16] I. Kovács, K. Luck, K. Spirohn, Y. Wang, C. Pollis, S. Schlabach, W. Bian, D. Kim, N. Kishore, T. Hao, M. Calderwood, M. Vidal, and A. Barabási, “Network-based prediction of protein interactions,” *Nature Communications*, vol. 10, no. 1, Dec. 2019.

[17] M. Newman, “Clustering and preferential attachment in growing networks,” *Physical Review E*, vol. 64, no. 2, p. 025102, 2001.

[18] L. A. Adamic and E. Adar, “Friends and neighbors on the web,” *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[19] T. Zhou, L. Lü, and Y. Zhang, “Predicting missing links via local information,” *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 71, no. 4, pp. 623–630, 2009.

[20] P. Jaccard, “The distribution of the flora in the alpine zone,” *New Phytologist*, vol. 11, no. 2, pp. 37–50, Feb. 1912.

[21] L. R. Dice, “Measures of the amount of ecologic association between species,” *Ecology*, vol. 26, no. 3, pp. 297–302, July 1945. [Online]. Available: http://www.jstor.org/pss/1932409

[22] C. V. Cannistraci, G. Alanis-Lobato, and T. Ravasi, “From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks,” *Scientific Reports*, vol. 3, 2013.

[23] L. Lü, L. Pan, T. Zhou, Y.-C. Zhang, and H. E. Stanley, “Toward link predictability of complex networks,” *Proceedings of the National Academy of Sciences*, vol. 112, no. 8, pp. 2325–2330, 2015. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1424644112

[24] O. Kuchaiev, M. Rasajski, D. J. Higham, and N. Przulj, “Geometric denoising of protein-protein interaction networks.” *PLoS Comput. Biol.*, vol. 5, no. 8, 2009.

[25] M. Belkin and P. Niyogi, “Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering,” in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, 2002, pp. 585–591.

[26] C. V. Cannistraci, G. Alanis-Lobato, and T. Ravasi, “Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding.” *Bioinform.*, vol. 29, no. 13, pp. 199–209, 2013.

[27] J. Kunegis, E. W. D. Luca, and S. Albayrak, “The link prediction problem in bipartite networks,” in *International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*. Springer, 2010, pp. 380–389.

[28] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, p. 5–53, jan 2004. [Online]. Available: https://doi.org/10.1145/963770.963772

[29] T. Zhou, “Discriminating abilities of threshold-free evaluation metrics in link prediction,” 2022. [Online]. Available: https://arxiv.org/abs/2205.04615