Predicting Links from Education Network of Employees

Ceyda Kocaman and Günce Keziban Orman

***Abstract*—The advances in technology and data science affect many fields positively. One of these fields is education. Learning analytics have the potential to develop new ways of achieving excellence in teaching and learning. The companies try to use learning analytics techniques for their employees' education and aim to improve employee performance. The education data sets of Softtech employees are used in this study. Softtech is a software company in Turkey, and those data sets include different types of technical, non-technical, online, and offline education. All data sets are combined, and an employee network is created by connecting employees via education. In this study, complex network analysis, link prediction, and machine learning techniques are applied with the aim of creating an education recommendation system.**

***Index Terms*—complex network analysis, link prediction, machine learning, education recommendation, and learning analytics.**

I. Introduction

DATA science studies in the education field increase day by day, and simultaneously, studies in learning analytics that aim to improve more effective ways for learning and teaching increase too. Adult learning is a part of learning analytics. As a result of the fourth industrial revolution, adults are constantly trying to update their knowledge and skills about their jobs. Adapting to the changing nature of work, 21st-century skills, new technologies, etc. has critical importance for people. Because of the changing global conditions, businesses support their employees' education. Working adults continue their lifelong learning journey in a variety of ways, including through online education, graduate-level education, company education, and so on. Online education facilitates the lifelong learning process for adults, especially during the COVID pandemic era. In that process, the volume of data related to online education increases rapidly. That situation has supported the growth of the learning analytics and educational data mining communities. Many studies about learning analytics have been carried out to date, and it still provides an opportunity for new studies. Untapped potential exists, in particular, for understanding how adult learners navigate their learning experience across course options and over time. [1]

Indeed, the analysis of the collected data from online education resources is a challenging task because, first, it involves describing the formal problems with the effectiveness of the education and, second, it involves finding the appropriate methodology for the analysis. Among several different analysis techniques, such as supervised or unsupervised learning of tabular data sets, natural language processing, or video/sound analysis, network science is emerging with the richness of graph modeling for representing objects' interactions. Network science has benefited from the advanced computational capabilities and increased availability of digital data. It is not surprising that network analysis has contributed to the emergence of learning analytics. Because network analysis facilitates the analysis of teaching and learning with computational, analytical, and representational support, it offers a suite of methods to analyze learning and learners. Many different types of data about learning can be examined by using network approaches, for example, social interactions in forums, friendship ties in the classroom, co-enrollment in courses, etc. There are opportunities and challenges to strengthen future work on situating network-analytical research within learning analytics. [2]

In this study, we focus on the network modeling of the education data set. There is a need for better data modeling that is able to understand the hidden interests of the employees in education. For this issue, network modeling is employed. The education data sets of Softtech employees are used in this study. Softtech is a software engineering company in Turkey, and those data sets include different types of technical, non-technical, online, and offline education. Detail information about the data sets is explained in the data pre-processing step. Other information about Softtech is available on its website [3]. A good education recommendation system has an important role both in the employees' experiences and in economic gain. In this study, the aim is to build an education recommendation system. The education recommendation problem is formalized as predicting new possible connections in a complex education-employee interaction system. Even if employees do not know each other and there is no closeness between them, they can be influenced by each other because they are a part of the same system, and these hidden influences can affect their preferences. For that analysis, complex network modeling of employee-education interactions is useful.

In this study, link prediction, one of the sub-domains of network science, is used to find the missing links. It is possible to divide the hundreds of different link prediction approaches into three parts. In this study, machine-learning-based methods were preferred among the three categories. One of the other categories is traditional methods, and the last one is graph embedding techniques that have appeared in recent works. The details of the experiment that involves link prediction via machine learning can be seen in Fig. 1. The main data set is converted to a bipartite network, and then it is split into two different auxiliary projected networks

Manuscript received March 27, 2023. This work was supported by the Galatasaray University Research Fund (BAP) within the scope of project number FBA-2021-1063, and titled "Niteliklendirilmiş çift yönlü ağlarda bağlantı tahmini ile öneri Sistemleri geliştirilmesi".

C. Kocaman is with the Department of Smart Systems Engineering, Galatasaray University, Istanbul, 34349 TURKEY e-mail: ceyda.kocaman@ogr.gsu.edu.tr.

G. K. Orman is with the Department of Computer Engineering, Galatasaray University, Istanbul, 34349 TURKEY e-mail: korman@gsu.edu.tr.

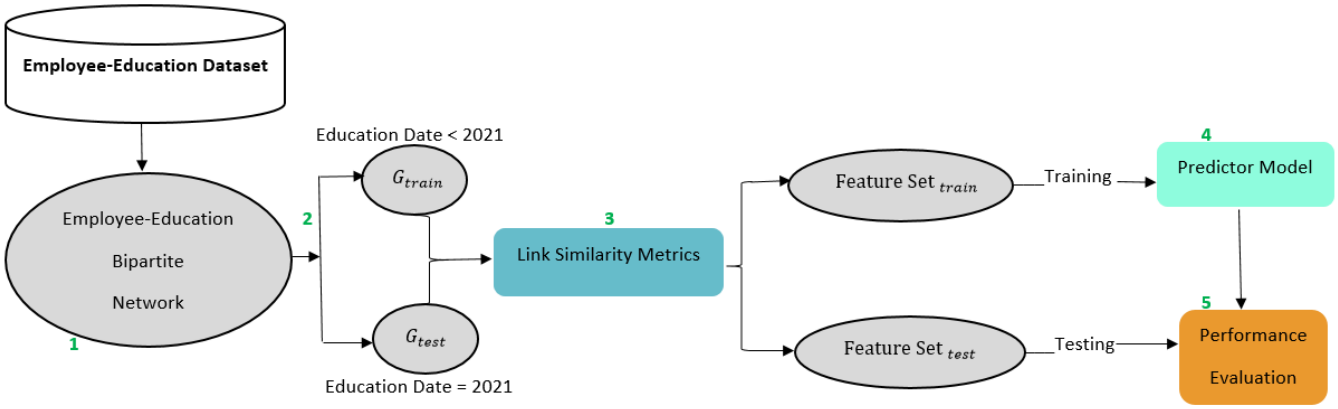

Fig. 1. Flowchart of the link prediction frameworks

by filtering the bipartite network by education date between employee-education pairs. If the education date is before 2021, it is training data, or in other words, the Gtrain projected network. After splitting the network into training and test networks, all link prediction features for all possible links are calculated. This operation includes both appearing and missing links for each network separately. FeatureSettrain and FeatureSettest are crated for Gtrain and Gtest projected networks, respectively. While creating these sets, a label column is created according to whether the link was seen or not. If the link is already seen in the relevant network, its label is 1, if it is not seen, its label is 0. FeatureSettrain and FeatureSettest are used as training and test data sets in machine learning algorithms. The results are evaluated by using different machine learning algorithms. Accuracy, precision, recall, and F1 scores are used for evaluating the performance of the machine learning models.

All the detail information about this education recommendation study is shared under the titles ”data pre-processing, link prediction,” and machine learning titles in order. The study is concluded with an explanation in the conclusion part.

II. Data Pre-Processing

In this part of the study, all information about data sets and data pre-processing steps is explained from beginning to end.

A. About Raw Data Sets

In this study, the education data sets of all employees who worked continuously at Softtech between 2017 and 2021 are used. There are primarily two raw data sets. One of them is the ”Online Education” data set, and the other is the ”Offline and Virtual Class Education” data set. All information about the identification of employees is masked; a user ID is used instead of an employee ID or employee name. There are 669 users in this study. The ”Online Education” data set includes four columns that are named ”User Id”, ”Education Name”, ”Sub Education Name” and ”End Date of the Education”. There are 24774 rows in that data set. The ”Offline and Virtual Class Education” data set includes five columns that are named ”User Id”, ”Education Name”, ”Sub Education Name”, ”Start Date of The Education” and ”End Date of The Education”. There are 16439 rows in that data set. ”User Id” columns are in string format, and they sequentially increase one by one like UserId1, UserId2, UserId3, etc. ”Education Name” and ”Sub Education Name” columns are in string format, and they include different types of technical or non-technical education names. Some of the technical educations are like Java, C#, .NET Core, SQL, Python, Data Science, Machine Learning, Big Data, Deep Learning, HTML5/CSS3, Vue.js, NodeJS, Javascript, IoT, etc. Some of the non-technical educations are like Proactive Behavior, Planning and Organization, Time Management, Stress Management, Emotional Intelligence, etc. These educations may be given as examples. All date columns include day, month, and year information.

B. Creating an Education Dictionary

First of all, there is a need to prepare an education dictionary. A new data set that is named ”Education Dictionary” is created. These data sets include two columns, ”Education Id” and ”Education Name”. All 720 different educations are added to this dictionary, and therefore this data set includes 720 rows. The ”Education Id” column is in string format, and it sequentially increases one by one, like Education1, Education2, Education3, etc. The ”Education Name” column is in string format and it includes different types of technical or non-technical education names previously described. ”Education Dictionary” would be used to convert education columns later.

C. Creating a Merged and Transformed Education Data Set

”Online Education” and ”Offline and Virtual Class Education” data sets are converted to the same format, and then they are merged to get only one ”Education” data set. ”Sub Education Name” columns are not used in the study, and so they are removed from the data sets. The ”Start Date of the Education” column in ”Offline and Virtual Class Education” is removed, and only the ”End Date of the Education” columns in two data sets are used in this study by converting it to a general ”Date” column. Only year information is extracted from ”Date” columns; day and month information in ”Date” columns are not taken into account in this study. ”Education Name” columns are converted to ”Education Id” by using the education dictionary. At the end, one merged ”Education” data set that has ”UserId”, ”EducationId”, and ”Year” columns is obtained from the ”Online Education” and ”Offline and Virtual Class Education” data sets.

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

D. Creating Networks

The "Education" data set is split into two parts: training and test data sets, according to the year information. If the year is smaller than 2021, the data is training data. If the year is 2021, the data is test data. After that operation, the year column is removed from training and test data sets. They include only "UserId" and "EducationId" columns; thus, training and test bipartite networks are got. The repetitive rows are deleted from training and test bipartite networks. The training bipartite network includes 11459 different UserId-EducationId links, and the test bipartite network includes 2241 different UserId-EducationId links.

The users in bipartite networks are associated with each other using education information; thus, bipartite networks are converted to training and test networks. The training network includes 119022 different UserId-UserId links, and the test network includes 64046 different UserId-UserId links. At the end, "User" string expressions are removed from the data sets, and thus numeric UserId-UserId links appear in the networks. All these training and test networks that include only numeric nodes are saved in edge files, and then the data pre-processing steps are ended. The users are nodes in the networks, and the educations are links in the networks.

Sample visualization of how employees are connected via education is shared in Fig. 2 to give an idea about network. The network is considered a dynamic network and includes only three months as a sample.

III. Link Prediction

In this part of the study, all link prediction features for all possible links are calculated, including appearing and missing ones, for each network separately by using the similarity/distance metrics. Link prediction features are classified in three parts according to their essential techniques: local, global, and embedding. 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. $N(u) = \{v \in V : (u, v) \in L\}$. The methods for link prediction with local information are explained below:

Common Neighbors (CN): It is the size of the set of common neighbors between any two nodes. If the number of degrees is higher, it is more possible to have higher Common Neighbors for the nodes. Because of that reason, Common Neighbors has a tendency to be high for any two hub nodes [4]. Its formula is given in Eq.1.

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

Adamic Adar (AA): It penalizes the scores for hub neighbors. In other words, it counts the total number of neighbors of all common neighbors, but depresses the score by a logarithmic function for demoting the scores of higher degree nodes [5]. Its formula is given in Eq.2.

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

Resource Allocation (RA): It is the same with Adamic Adar, but the difference between Resource Allocation and Adamic Adar that Resource Allocation considers the degrees, not their logarithms. In addition, it also counts the total number of neighbors of all common neighbors [6]. Its formula is given in Eq.3.

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

Jaccard Coefficient (JC): It is the ratio of the number of common neighbors to the number of all neighbors of two nodes and was developed for comparing two sets [7]. The formula is given in Eq.4.

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

Sørrenson/Dice Index (Dice): It measures the common parts of the neighborhoods, normalizes them with the sizes of the neighborhoods of the two studied nodes, and penalizes being a hub as well. Sørrenson/Dice becomes lower than Jaccard Coefficient when the neighborhoods have many nodes in common but also the common neighbors have many other links to the outside of the common neighborhood [8]. The formula is given in Eq.5.

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

Cannistraci-Alanis-Ravasi index (CAR): It is the sum of the numbers of common neighbors of two nodes, each having neighbors in common with the other [9]. Its formula is given in Eq.6.

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

CAR-based Adamic and Adar (CAA): It combines two strategies: favoring clique-like neighborhoods and penalizing being a hub. In other words, it is a combination of the Cannistraci-Alanis-Ravasi and Adamic Adar strategies [9]. The formula is given in Eq.7.

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

CAR-based Resource Allocation (CRA): It is another hybrid metric that combines the Cannistraci-Alanis-Ravasi index and Resource Allocation strategies [9]. Its formula is given in Eq.8.

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

Preferential Attachment (PA): It promotes the nodes that have higher degrees and assumes that the famous nodes should have a higher probability of connecting with each other. Shortly, it is the multiplication of degrees of two nodes [4]. The formula is given in Eq.9.

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

CAR-based Preferential Attachment (CPA): It is the combination of Cannistraci-Alanis-Ravasi and preferential attachment strategies [9]. Its formula is given in Eq.10.

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

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

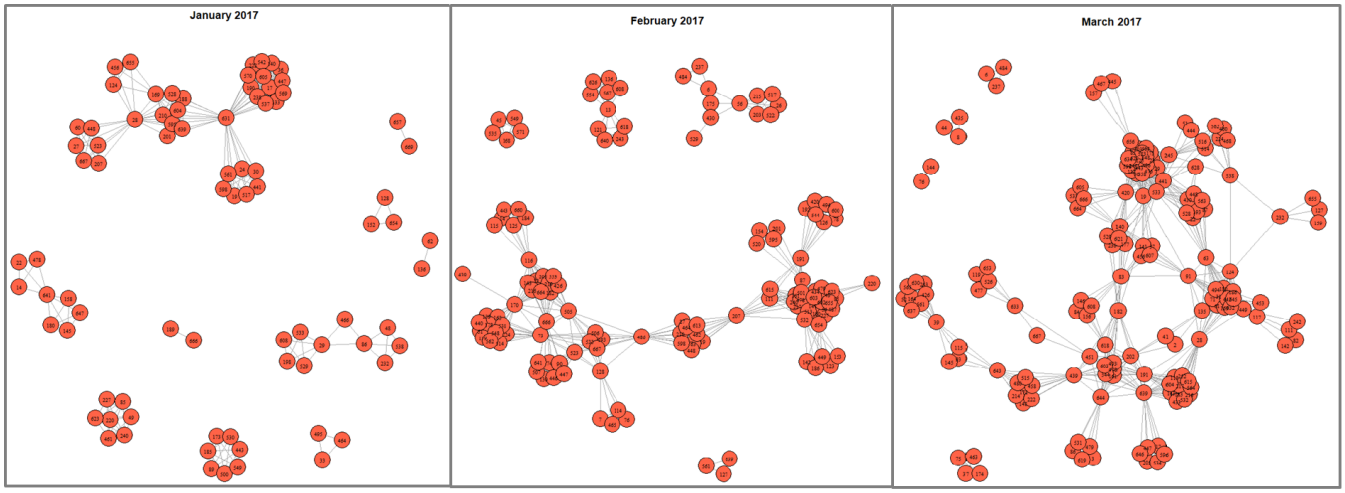

Fig. 2. Sample visualization of how employees are connected via education.

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.

The methods for link prediction with global information are explained below:

L3 link predictor (L3): It considers network paths of length three. The metric applies a degree normalization strategy because the third-level neighbors numbers are exponentially larger than the second-level ones. The biased high scores coming from the hub nodes, which are naturally building shortcuts and increasing the number of third-level neighbors for entire nodes, are also avoided [10]. L3 link predictor (L3), considers network paths of length three [10]. Its formula is given in Eq.11.

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

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.

Structural perturbation method (SPM): It is a technique that is similar to the first-order perturbation in quantum mechanics; it focuses on perturbing the adjacency matrix and observing the change in eigenvalues provided the fixed eigenvectors. The scores are produced for all links based on the perturbation of links removed from the adjacency matrix of the original network, basically. [11]

The methods for link prediction with embedding are explained below:

Isometric mapping (ISOMAP): Isometric mapping (ISOMAP), uses one of the traditional graph embedding techniques [12]. 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 $L \in R^l$ 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 $min_{x_1,x_2,...,x_n}(d_{uv} - ||x_u - x_v||)^2$. Once MDS generates new lower dimensional vectors for each node, then ISOMAP calculates basic euclidean distance between the nodes as their dissimilarities.

Laplacian Eigenmaps (LEIG): Firstly, the laplacian matrix of the original network is generated, and then the spectral decomposition of the corresponding laplacian matrix is computed because it uses a minimization function that can be solved by the generalized eigenvalue problem. Laplacian Eigenmaps find l eigenvalues and eigenvectors with l is the number of new dimensions. The link prediction is again done by considering the Euclidean distance of the node pairs after embedding[13].

Centered and non-centered Minimum Curvilinear Embedding (MCE & ncMCE): These are two network embedding techniques that use the distances in the minimum spanning trees of studied networks. Firstly, the minimum spanning tree is generated, and then the distances of every pair of nodes in the minimum-spanning tree are computed in both methods. The name of these distances, which are in the form of a distance matrix, is kernel. If the centering is not chosen in the algorithm, an economy-size singular value decomposition of the distance matrix is performed by non-centered Minimum Curvilinear Embedding. Otherwise, an algebraic operation is performed for kernel centering first, and then the decomposition is done. At the end, the new lower-dimensional space is produced by the transposition of the product of the computed singular values with the right singular vectors with the algebraic corrections [14].

In addition to all these methods, some other methods are applied. These methods are: Random Walk with Restart (RWR), Matrix Forest Index (MF), Local Paths Index (LP), Leicht-Holme-Newman Index (LHN_LOCAL), Leicht-Holme-Newman Global Index (LHN_GLOBAL), Pseudoinverse of the Laplacian (L), Katz Index (KATZ), Hub Promoted Index (HPI), Hub Depressed Index (HDI), Geodesic distance vertex similarity (DIST), Cosine Similarity based on the pseudoinverse of the Laplacian Matrix (COS_L),

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Cosine vertex similarity/ Salton index (COS), Average Commute Time (ACT) and Averag Commute Time, normalized (ACT_N).

The link prediction step begins with the preparation of training and test graphs. Firstly, the isolated nodes in the existing training and test networks are determined and saved. The training and test graphs are created from the existing training and test networks by separating the isolated nodes. The scores between two nodes are calculated for the new training and test graphs by using the link prediction methods that were previously mentioned. A file is created for each link prediction method, so 29 files are created for a graph because of the number of link prediction methods. In addition, a label file is created according to whether a link is visible or not for a graph. If the link is already seen in the relevant network, its label is 1, if it is not seen, its label is 0. In total, these 30 files are created for both training and test graphs separately. These files include node A, node B, and score columns. At the end, each training and test file is brought together among itself, and two main training and test data sets are created for use in the machine learning step. These files include 35 columns and 222778 rows. The target column is the label column, and the other columns are the scores of link prediction methods and information about nodes.

IV. Machine Learning

The output files created in the previous step are used as input files in this step, and the process begins with data pre-processing in those files to run the machine learning algorithms. The label columns in training and test data sets are converted from integer to factor format, so they would be used as target columns. All null values are replaced with zero values in training and test data sets. Only the scores of link prediction methods and label columns are included in machine learning models. If the data types are not the same in the training and test data sets, they are converted to be the same. After the standardization process on the data, the machine learning algorithms are run. The results are obtained by running XGBoost, gradient boosting, random forest, logistic regression, support vector machine, and multilayer perceptron machine learning algorithms.

Firstly, only training and test data sets are used for running machine learning algorithms. Accuracy, precision, recall, and F1 scores are used to evaluate the performance of the trained models. In table I, the values of accuracy, precision, recall, and F1 scores are shown for each model. The best scores are obtained by logistic regression and support vector machine. All values of accuracy, precision, recall, and F1 scores are 1.

TABLE I
First Link Prediction Results Via Machine Learning

Model	*Accuracy*	*Precision*	*Recall*	*F1 Score*
XGBoost	0.9989	1	0.9963	0.9981
Gradient Boosting	0.9989	1	0.9963	0.9981
Random Forest	0.9152	1	0.7723	0.8715
Logistic Regression	1	1	1	1
Support Vector Machine	1	1	1	1
Multilayer Perceptron	0.8181	1	0.6124	0.7596

In Fig. 3, all confusion matrices that belong to each machine learning model are shown. All machine learning algorithms are successful at predicting the true positive (TP) values. The differences appear when the models predict the true negative (TN) values and errors are observed. Logistic regression and support vector machine algorithms are successful in predicting both the true positive (TP) and the true negative (TN) values.

XGBOOST

	Reference	
Prediction	0	1
0	158494	238
1	0	64046

GRADIENT BOOSTING

	Reference	
Prediction	0	1
0	158494	238
1	0	64046

RANDOM FOREST

	Reference	
Prediction	0	1
0	139849	18883
1	0	64046

SUPPORT VECTOR MACHINE

	Reference	
Prediction	0	1
0	158732	0
1	0	64046

LOGISTIC REGRESSION

	Reference	
Prediction	0	1
0	158732	0
1	0	64046

MULTI LAYER PERCEPTRON

	Reference	
Prediction	0	1
0	118200	40532
1	0	64046

Fig. 3. Confusion matrices of the first experiment.

Then this experiment is repeated with a new validation data set. The training set is divided into two parts at a rate of 30% to 70% randomly and homogeneously. 30% of the training data set is used as validation data, and 70% of the training data set is used as new training data. The machine learning algorithms are run again with a new training data set. The experiment is repeated on both validation and test data. These two different analyses are done to notice a possible overfit of the models. The results are shown in table II. According to these comparative experiments, the algorithms are not overfitting. Again, the best scores are obtained by logistic regression and support vector machine. All values of accuracy, precision, recall, and F1 scores are 1. This shows us that the link structure in the network is regular. In other words, it can be said that making predictions using an education network is a logical method.

XGBOOST

	Reference	
Prediction	0	1
0	158472	260
1	0	64046

GRADIENT BOOSTING

	Reference	
Prediction	0	1
0	158472	260
1	0	64046

RANDOM FOREST

	Reference	
Prediction	0	1
0	140276	18456
1	0	64046

SUPPORT VECTOR MACHINE

	Reference	
Prediction	0	1
0	158732	0
1	0	64046

LOGISTIC REGRESSION

	Reference	
Prediction	0	1
0	158732	0
1	0	64046

MULTI LAYER PERCEPTRON

	Reference	
Prediction	0	1
0	138594	20138
1	0	64046

Fig. 4. Confusion matrices of the second experiment.

The confusion matrices that belong to the new experiment are shown in Fig. 4. It is possible to make similar comments with previous results. All machine learning algorithms are successful in predicting the true positive links. Only logistic regression and support vector machine algorithms are successful in predicting both true positive links and true

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE II
SECOND LINK PREDICTION RESULTS VIA MACHINE LEARNING

Model		*Accuracy*	*Precision*	*Recall*	*F1 Score*
XGBoost	**Validation**	1	1	1	1
	Test	0.9988	1	0.9960	0.9980
Gradient Boosting	**Validation**	1	1	1	1
	Test	0.9988	1	0.9960	0.9980
Random Forest	**Validation**	1	1	1	1
	Test	0.9172	1	0.7763	0.8741
Logistic Regression	**Validation**	1	1	1	1
	Test	1	1	1	1
Support Vector Machine	**Validation**	1	1	1	1
	Test	1	1	1	1
Multilayer Perceptron	**Validation**	1	1	1	1
	Test	0.9096	1	0.7608	0.8641

negative links. XGBoost and gradient boosting algorithms have the same accuracy value and show similar predicting performance in both experiments.

As a result of these experiments, we can accurately predict whether there is a link, according to the education that they received, between the two employees. For an employee, an education plan can be created from all education received by the employees that employee is related. All education that is received by non-related employees can be excluded from the priority of this plan. The final plan can be created after all education that has been received by the employee until that date is also removed from the plan. In this study, we do not keep education information on the link. In the next stage of the study, education information could be recorded on the link, and a direct education prediction would be made. According to the number of joint educations received, it is possible to take into account the strength of the bond between the employees, too.

V. CONCLUSION

In this study, a link prediction framework for education recommendations to employees is proposed by using the education data sets of Softtech employees. Softtech is a software engineering company in Turkey, as mentioned before, and these data sets were created specifically for this study. This study begins with the education records that belong to 2017, because the education data sets of the employees have been recorded regularly since 2017. These data sets are converted to a network so that employees are nodes and educations are links after data pre-processing steps. Then the link prediction processes are applied to the networks. At the end of the link prediction step, new data sets are created for use as input files in the machine learning step. XGBoost, gradient boosting, random forest, logistic regression, support vector machine, and multilayer perceptron machine learning algorithms are run. Accuracy, precision, recall, and F1 scores are used for evaluating the performance of the machine learning models. The logistic regression and support vector machine methods achieved the most accurate link prediction.

This modeling can be used for the education recommendation system. The preferences of other employees who have made the same education choices as themselves or who are similar to them in the system can be offered to the employees. This type of recommendation can include more interesting educational offers than attribute similarity-based collaborative filtering. Our study shows that the combination of different features in a machine learning model can result in accurate link predictions. Adding education attributes besides the network-based topological features can be complementary. In this study, we made a purely analytical estimation. However, we did not examine their corresponding results in real life. The current results show the feasibility of this study. But as the amount of data increases, the realistic nature of the results will increase proportionally. It is also possible to get more realistic results by detailing the education information in the links. The next steps can be guided by employees' feedback based on education recommendations to employees. Another alternative is to compare the results of recommended and preferred educations in real life.

REFERENCES

[1] C. E. Tatel, S. F. Lyndgaard, R. Kanfer, and J. E. Melkers, "Learning while working: : Course enrollment behaviour as a macro-level indicator of learning management among adult learners," *Journal of Learning Analytics*, vol. 9, no. 3, pp. 104–124, Dec. 2022.
[2] B. Chen and O. Poquet, "Networks in learning analytics: Where theory, methodology, and practice intersect," *Journal of Learning Analytics*, vol. 9, no. 1, pp. 1–12, Mar. 2022.
[3] softtech.com.tr, "Who are we?" https://softtech.com.tr/en/about-us/.
[4] M. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, no. 2, p. 025102, 2001.
[5] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.
[6] 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.
[7] P.Jaccard, "The distribution of the flora in the alpine zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, Feb. 1912.
[8] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, Jul. 1945. [Online]. Available: http://www.jstor.org/pss/1932409
[9] 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.
[10] I. Kovács, K. Luck, K. Spirohn, Y. Wang, C. Pollis, S. Schlabach, W. Bian, D. Kim, T. H. N. Kishore, M. Calderwood, and A. L. B. M. Vidal, "Network-based prediction of protein interactions," *Nature Communications*, vol. 10, no. 1, Dec. 2019.
[11] L. Lü, L. Pan, T. Zhou, Y. 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
[12] O. Kuchaiev, M. Rasajski, D. Higham, and N. Przulj, "Geometric denoising of protein-protein interaction networks," *PLoS Comput. Biol.*, vol. 5, no. 8, 2009.
[13] 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.
[14] 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, 2013.

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)