2007 KARABÜK ÜNİVERSİTESİ

COMPLEX NETWORK-BASED LINK PREDICTION IN COMPUTER SCIENCE, SOCIAL SCIENCE, AND MEDICAL SCIENCE PUBLICATIONS IN IRAQ

2023
MASTER THESIS
COMPUTER ENGINEERING

Albatol Abdulmahdi Saleh AL-DHAYAB

Thesis Advisor
Assist. Prof. Dr. Emrah ÖZKAYNAK

COMPLEX NETWORK-BASED LINK PREDICTION IN COMPUTER SCIENCE, SOCIAL SCIENCE, AND MEDICAL SCIENCE PUBLICATIONS IN IRAQ

Albatol Abdulmahdi Saleh AL-DHAYAB

Thesis Advisor
Assist. Prof. Dr. Emrah ÖZKAYNAK

T.C.
Karabuk University
Institute of Graduate Programs
Department of Computer Engineering
Prepared as
Master Thesis

KARABUK
September 2023

I certify that, in my opinion, the thesis submitted by Albatol Abdulmahdi Saleh AL-DHAYAB titled "COMPLEX NETWORK-BASED LINK PREDICTION IN COMPUTER SCIENCE, SOCIAL SCIENCE, AND MEDICAL SCIENCE PUBLICATIONS IN IRAQ" is fully adequate in scope and quality as a thesis for the degree of Master of Computer Engineering.

Assist. Prof. Dr. Emrah ÖZKAYNAK …………………...
Thesis Advisor, Department of Computer Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science thesis. September13, 2023

Examining Committee Members (Institutions)	Signature
Chairman : Assist. Prof. Dr. Muhammet ÇAKMAK (SU)	……………………
Member : Assist. Prof. Dr. Emrah ÖZKAYNAK (KBU)	……………….….....
Member : Assist. Prof. Dr. Mehmet Zahid YILDIRIM (KBU)	…...……….………

The degree of Master of Computer Engineering by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Assoc. Prof. Dr. Zeynep ÖZCAN ....
Director of the Institute of Graduate Programs

“I declare that all the information within this thesis has been gathered and presented by academic regulations and ethical principles, and I have, according to the requirements of these regulations and principles, cited all those which do not originate in this work.”

Albatol Abdulmahdi Saleh AL-DHAYAB

ABSTRACT

M. Sc. Thesis

COMPLEX NETWORK-BASED LINK PREDICTION IN COMPUTER SCIENCE, SOCIAL SCIENCE, AND MEDICAL SCIENCE PUBLICATIONS IN IRAQ

Albatol Abdulmahdi Saleh AL-DHAYAB

Karabuk University
Institute of Graduate Programs
The Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Dr. Emrah ÖZKAYNAK

September 2023, 85 pages

Scientific collaboration networks are used to display the relationships between researchers who work on joint research, projects or papers. The analysis of scientific cooperation networks plays an important role in the dissemination of knowledge, the creation of new associations and the emergence of new innovations. Especially in the establishment of national or international scientific cooperation, data obtained from scientific cooperation networks are widely used. In addition, scientific cooperation networks are used in the dissemination of joint studies in similar disciplines or interdisciplinary. Link prediction is widely used in the analysis of new associations based on scientific collaboration. Link prediction is the process of predicting new connections that may arise in the future by looking at the status of existing connections in the network. Link prediction in scientific collaboration networks is important for understanding and strengthening scientific collaborations and increasing efficiency in scientific collaboration. In addition, link prediction plays an important role in increasing interdisciplinary collaboration. Many methods based on data mining, machine learning, and complex network analysis have been proposed and used today for link prediction in networks. Neighborhood-based methods are the most common among the proposed methods. The most important reason why neighborhood-based methods are preferred is the high estimation success with little information in the network. These methods, which work based on the analysis of common neighbors between nodes, reveal similarities between nodes. In this thesis, scientific cooperation networks were created from the publications of Iraqi researchers in the fields of computer science, health sciences and social sciences by looking at the link prediction studies in scientific cooperation networks, and neighborhood-based link prediction processes were carried out in these networks. In the study, data belonging to the joint publications of Iraqi researchers were collected from many sources such as Web of Science, Google Scholar and Microsoft Academic. In link prediction processes, along with neighborhood-based link prediction methods, machine learning methods such as Support Vector Machine (SVM), Random Forest (RF) and Logistic Regression (LR) are also used. Results from experimental studies show that link prediction methods are successful in predicting new links in established scientific collaboration networks. Among the machine learning methods used, the RF classifier was the most successful with 96% accuracy. The study demonstrates the usability of neighborhood-based link prediction methods and machine learning methods in recommendation systems to be created for the dissemination of scientific collaborations

Keywords **:** Complex networks, link prediction, machine learning, data mining.
Science Code **:** 92429

ÖZET

Yüksek Lisans Tezi

IRAQ'TA BİLGİSAYAR BİLİMİ, SOSYAL BİLİMLER VE TIBBİ BİLİMLER YAYINLARINDA KARMAŞIK AĞ TABANLI BAĞLANTI TAHMİNİ

Albatol Abdulmahdi Saleh AL-DHAYAB

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi Emrah ÖZKAYNAK

September 2023, 85 sayfa

Bilimsel işbirliği ağları, ortak araştırma, proje veya makale çalışmaları yapan araştırmacılar arasındaki ilişkilerin gösteriminde kullanılmaktadır. Bilimsel işbirliği ağlarının analizi, bilginin yayılmasında, yeni birlikteliklerin oluşturulmasında ve yeni inovasyonların ortaya çıkmasında önemli rol oynamaktadır. Özellikle ulusal ya da uluslararası bilimsel işbirliklerinin oluşturulmasında bilimsel işbirliği ağlarından elde edilen veriler yaygın olarak kullanılmaktadır. Ayrıca benzer disiplinlerde ya da disiplinler arası ortak çalışmaların yaygınlaştırılmasında da bilimsel işbirliği ağları kullanılmaktadır. Bilimsel işbirliğine dayalı yeni birlikteliklerin analizinde bağlantı tahmini yaygın olarak kullanılmaktadır. Bağlantı tahmini, ağdaki mevcut bağlantıların durumuna bakarak gelecekte ortaya çıkabilecek yeni bağlantıları tahmin etme işlemidir. Bilimsel işbirliği ağlarında bağlantı tahmini, bilimsel işbirliklerin anlaşılması, güçlendirilmesi ve bilimsel işbirliğindeki verimliliğin artırılması için önemlidir. Ayrıca, disiplinler arası işbirliğinin arttırılmasında da bağlantı tahmini önemli bir rol oynamaktadır. Ağlarda bağlantı tahmini için veri madenciliği, makine öğrenmesi, karmaşık ağ analizi tabanlı pek çok yöntem önerilmiş ve günümüzde kullanılmaktadır. Önerilen yöntemler içerisinde en yaygın olanı ise komşuluk tabanlı yöntemlerdir. Komşuluk tabanlı yöntemlerin tercih edilmesinin en önemli sebebi ise ağdaki az bilgiyle yüksek tahmin başarısıdır. Düğümler arası ortak komşuların analizine dayalı çalışan bu yöntemler düğümler arasındaki benzerlikleri ortaya çıkarmaktadır. Bu tez çalışmasında bilimsel işbirliği ağlarında bağlantı tahmini çalışmalarına bakılarak bilgisayar bilimi, sağlık bilimleri ve sosyal bilimler alanlarında Iraklı araştırmacıların yapmış oldukları yayınlardan bilimsel işbirliği ağları oluşturulmuş ve oluşturulan bu ağlarda komşuluk tabanlı bağlantı tahmini işlemleri gerçekleştirilmiştir. Çalışmada, Web of Science, Google Scholar ve Microsoft Academic gibi bir çok kaynaktan Iraklı araştırmacıların ortak yayınlarına ait veriler toplanmıştır. Bağlantı tahmini işlemlerinde komşuluk tabanlı bağlantı tahmini yöntemleri ile birlikte Destek Vektör Makinesi (SVM), Rastgele Orman (RF) ve Lojistik Regresyon (LR) gibi makine öğrenmesi yöntemleri de kullanılmıştır. Deneysel çalışmalardan elde edilen sonuçlar bağlantı tahmini yöntemlerinin oluşturulan bilimsel iş birliği ağlarında yeni bağlantıları tahmin etmede başarılı olduğunu göstermektedir. Kullanılan makine öğrenimi yöntemleri içerisinden RF sınıflandırıcısı %96 doğruluk oranıyla en başarılı sınıflandırıcı olmuştur. Çalışma, bilimsel işbirliklerinin yaygınlaştırılması için oluşturulacak öneri sistemlerinde komşuluk tabanlı tahmini yöntemleri ve makine öğrenimi yöntemlerinin kullanılabilirliğini göstermektedir.

Anahtar Kelimeler: Karmaşık ağlar, bağlantı tahmini, makine öğrenmesi, veri madenciliği.

Bilim Kodu **:** 92429

ACKNOWLEDGMENT

I owe thanks and praise to God first and foremost for this success and facilitation as I bow to my beloved parents. My dear father gave me the most valuable things to make me a man of honor. My beloved mother is good at engineering my heart with her prayers. To my family that I grew up in and its extension gives me pride and honor. I owe a special thanks to my thesis supervisor, Assist. Prof. Dr. Emrah Özkaynak spared no effort in providing unlimited advice and guidance until the completion of this thesis to the fullest.

I dedicate this thesis to my beloved country, Iraq. Moreover, to the beautiful Turkey, which embraced this scientific experience and contributed to providing all possibilities for graduating in this distinguished way.

CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

Page

LIST OF TABLES

Page

INDEX OF ABBREVIATIONS

LP	: Link Prediction
ML	: Machine Learning
SVM	: Support Vector Machine
LR	: Logistic Regression
RF	: Random Forest
CS	: Computer Science
MS	: Medical Science
SS	: Social Science
SFCN	: Similarity-based Future Common Neighbors
CNDP	: Critical node Detection Problem Algorithm
GLHN	: Global Leicht Holme Newman
PTRN	: Public Transit Route Networks
CN	: Common Neighbors
AA	: Adamic/Adar
JC	: Jaccard Coefficient
LPP	: Link Prediction Problem
GANs	: Generative Adversarial Networks
TMFs	: Temporal Matrix Factorization
GCNs	: Graph Convolutional Networks
LSTMs	: Long Short-Term Memory Networks
LDM	: Latent Dirichlet Allocation
LPNR	: LP-based Network Representation
RFC	: Random Forest Classifier
SNA	: Social Network Analysis
DD	: Grade distribution
CC	: Clustering Coefficient
PAC	: Preferential Attachment Coefficient
RAI	: Resource Allocation Index

PART 1

INTRODUCTION

1.1. MOTIVATION

LP is an important study field in network analysis that estimates the chance of a connection being formed among two nodes. It has implications in various domains, including social networks, transportation networks, biological networks [1],[2], etc. In computer science publications, LP is also essential as it helps to predict the possibility of collaboration among researchers, co-authorship of papers, and citation networks. Complex network-based LP has emerged as a powerful technique to predict links in computer science, medical science, and social science publications.

The primary motivation behind this study is to address the critical issue of accurately predicting new links or relationships in large and complex networks. LP has several real-world applications, such as recommender systems, online shopping, social media, and collaboration in various fields. With the increasing size of networks and the number of users, it has become challenging to identify potential links that can form in the future. Therefore, developing effective LP algorithms is critical for the success of these systems.

The importance of LP can be observed in e-commerce websites, where it is necessary to suggest products that interest a user. By providing accurate recommendations, e-commerce sites can increase customer satisfaction and loyalty, increasing profits. Similarly, accurate LP is essential in social media applications like Instagram and Reddit to keep users engaged and interested. Instagram and Reddit's success relies on their ability to provide users personalized recommendations and content, making LP crucial in retaining user interest.

Moreover, LP has become increasingly important in research, where identifying potential collaborators can significantly impact the quality and scope of research projects. For instance, researchers can use LP algorithms to identify potential co-authors or collaborators with similar research interests. LP can also help identify potential funding sources, leading to new research opportunities.

To increase anticipated linkage reliability, this research presents a new method that integrates data from several temporal network sources with existing LP techniques. The study also proposes a novel evaluation metric that can compare the performance of different LP methods. The findings of this study have implications for the development of recommendation systems, online shopping, and social media applications, as well as academic research.

1.2. PROBLEM STATEMENT

Implementing complex network-based link prediction in the publications of computer science, social science, and medical science in Iraq poses several noteworthy challenges. The limited availability and quality of data pose significant challenges to the precision and effectiveness of link prediction algorithms in academic research. The issue is exacerbated by the absence of standardized data formats and interoperability among various research databases. The political and economic challenges in Iraq may potentially have a detrimental impact on the financial and resource allocation for research endeavors, consequently impeding the advancement and dissemination of state-of-the-art methodologies for forecasting connections. The political and economic challenges Iraq faces could be hindered by institutional barriers and a fragmented research landscape, as the intricate nature of interconnected systems requires collaboration among experts from various fields. Lastly, an issue about human resources in Iraq pertains to the scarcity of proficient data scientists and researchers with the requisite skills to undertake intricate network analysis. Collaboration among academia, government entities, and international organizations is imperative to facilitate data sharing, allocate resources toward research infrastructure, cultivate interdisciplinary cooperation, and cultivate a skilled workforce capable of effectively utilizing the potential of complex network-based link prediction to advance scientific knowledge.

The proliferation of communication tools has led to the development of complex network structures, which provide a framework for understanding relationships and dynamics among individuals, objects, and events in daily life. However, predicting new links in complex systems, including social and biological networks and transportation networks, is challenging. While there are several mathematical and computing processes for LP, there is a need to explore the use of complex network-based LP techniques to foresee potential connections among writers and their future computer science publishing output in Iraq.

1.3. OBJECTIVE

New connections among authors and their publications in (CS, MS, and SS) in Iraq will be predicted using complicated network-based LP algorithms, which will be investigated in this study. A bibliographic data collection culled from the Web of Science, Google Scholar, and Microsoft Academic will be used for the research. We used three different classifiers—the Support Vector Machine, the RF, and the LR—to predict how likely a new connection will be made among a certain author and their respective publication. The study aims to evaluate the classifiers' performance using various metrics, including Accuracy, precision, and recall. The project's findings will have applications in recommendation systems and academic network analysis.

1.4. SCOPE OF STUDY

The objective of this thesis is to undertake a comprehensive examination of the effectiveness of different methodologies employed for the evaluation and ranking of scholarly articles in the domains of Computer Science (CS), Medical Science (MS), and Social Sciences (SS). The research centers on comparing various architectures and evaluating training and validation results from different upscaling methods, activation functions, cost functions, post-processing techniques, and pre-processing methodologies.

The scope of this study is to assess the efficacy of these methodologies; distinct performance metrics are utilized, such as the Jaccard coefficient, sensitivity, and specificity. The metrics above are widely acknowledged in information retrieval and ranking tasks. We offer valuable insights into the models' capacity to accurately evaluate the relevance and quality of academic papers within the designated fields.

1.5. CONTRIBUTION

The contribution of this study lies in its application of complex network-based LP techniques to predict new links among authors and their publications in computer science, medical science, and social science. By utilizing a bibliographic dataset obtained from multiple sources and employing three different classifiers, the study demonstrates the effectiveness of these techniques in predicting new links in the three publications, with the RF classifier showing the highest performance. The study's findings have important applications in recommendation systems and academic network analysis, allowing for a better understanding of the relationships and dynamics within the publication networks. This study provides a practical example of how complex network-based LP techniques can be applied to real-world scenarios. It highlights their potential to make meaningful contributions to various fields.

The Random Forest (RF) model was trained on the utilized dataset, resulting in a notable accuracy rate of 96%. This achievement can be considered successful.

PART 2

LITERATURE REVIEW

A literature review on this topic would likely explore the concept of network-based complex correlation in computer science research. It can examine the different methodologies used to analyze complex networks and their correlations, such as network motifs and clustering algorithms. The review can also investigate the impact of complex network analysis on different fields within computer science, such as ML, social network analysis, and bioinformatics.

The literature review can also discuss the challenges and limitations of network-based complex correlation analysis, such as scalability issues, data sparsity, and network inference accuracy. It can also explore future research directions, including developing new methods and techniques for analyzing and modeling complex networks and integrating network analysis with other computational approaches.

2.1. SIMILARITY-BASED METHODS

Regarding LP, similarity-based algorithms are some of the simplest and most tried-and-true. These strategies figure out how similar two nodes are by comparing their neighbors. Eventually, connections will be made among nodes that share many neighbors. Common Neighbors, Jaccard's Coefficient, and Adamic-Adar are only a few similarity-based approaches available.

Li, Shibao, et al. [3] first offered LP as a concept in 2018. They developed the similarity-based Future Common Neighbors (SFCN) model. The SFCN model successfully predicts network links by locating nodes likely to have similar neighbors in the future. Three simulated experiments are run in MATLAB, with the first demonstrating that future shared neighbors are more influential in complex networks than present ones. The remaining two tests test the SFCN model's accuracy and performance resilience by comparing it against eight methods across five networks.

In 2019, Najari, Shaghayegh, et al. [4] presented a new method to predict missing or future links in multiplex networks, considering interlayer likeness and proximity-based features. The proposed framework considers the structural information of other layers when predicting links in one layer. Adamic-Adar and Jaccard Coefficient are proximity-based features that are easy to compute and don't require learning. When predicting connections in multiplex networks, the suggested method beats state-of-the-art methods.

Bastami et al.[5] introduced a novel unsupervised LP method that enhances local and global predictions by integrating node properties, community information, and graph characteristics. Local prediction accuracy is improved by using a gravitation-based method for community discovery, while global prediction error is decreased by scattering search results throughout the graph. The experimental results demonstrate the superiority of the proposed method over the existing similarity-based methods in terms of execution time and accuracy. The accuracy of the proposed technique improves when the network has robust communities, triangular links, and narrow diameters. Although there is a trade-off among accuracy and execution time, the method may be used for very large and complex networks.

The TSLP, introduced by Meybodi M. R. et al. [6] in 2019, is a novel similarity-based LP approach for temporal networks. In order to foretell missing connections in a network, the approach considers local and global temporal similarities. The authors utilized two real-world temporal networks to show that their method outperformed the current gold standard in LP. TSLP considers the network's history of foretelling how node connections will change over time. Their research shows that temporal likeness metrics might be useful for predicting links in temporal networks. In conclusion, TSLP is a promising method for enhancing the precision of LP in temporal networks.

In 2019, Gu, Weiwei, et al. [7] reviewed various methods used for LP in complex networks. They analyzed the performance of these methods on different types of networks and highlighted their strengths and weaknesses. The survey provides a valuable resource for researchers and practitioners interested in LP in complex networks. Overall, their work highlights the importance of understanding the strengths and limitations of different LP methods.

Complex network modeling and community identification were the basis for an LP model developed by Ai, Jun et al. [8] in 2019. Users' shared tastes in genres, rating distributions, and top-rated goods were considered as they built these intricate networks. Objects serve as nodes in a network, with likeness calculation results serving as link weights. Accuracy is improved, and multi-factor community identification streamlines the prediction process depending on node likeness, conducted after collected community information. Depending on the findings, user actions like rating and choosing products reveal a latent community structure that may be used for connection prediction and a deeper comprehension of complex systems.

Zareie, Ahmad, and Rizos Sakellariou [9] proposed using correlation and common neighbors to determine the grade of likeness among two nodes in 2020. Likeness-based methods determine the structural likeness of two nodes by tallying the number of their common neighbors. However, there are certain cases that this approach will not cover. The experimental results show that the innovative method is superior to state-of-the-art methods for LP.

In 2020, Liu, JiaHui, et al. [10] proposed a novel likeness-based method for LP in heterogeneous networks that considers the likeness of both nodes and links in the network. Presents their approach and its evaluation on several real-world networks. They demonstrated that their method outperforms existing likeness-based methods, indicating the potential to incorporate node and link likeness in LP. Their work highlights the importance of considering the heterogeneous nature of networks in LP and provides a promising direction for improving the accuracy of LP in such networks.

The CNDP algorithm, developed by Rafiee et al. [11] in 2020, is a likeness-based LP approach considering topological aspects and network architecture. The method introduces a new measure that depends on the clustering coefficient, a structural feature of the network. To further improve its efficiency, the CNDP algorithm considers the neighbors of shared neighbors. Synthetic and real-world network evaluations show that the proposed technique outperforms competing methods thanks to its high accuracy and low complexity.

The topological nearest-neighbor likeness was introduced by Guo et al. [12] in 2023 to anticipate linkages in directed networks. The research enhanced the Sorensen index and its variations in directed networks and created a matrix algebra representation for them. The topological nearest-neighbors likeness index was calculated by considering each index's GLHN likeness index and the nearest-neighbors topology. Several real-world directed network datasets were used to verify the approach, and three different evaluation criteria were used to compare the results to benchmark indices. The experimental findings demonstrated the suggested method's superiority in LP for directed networks compared to the benchmark indices.

2.2. LOCAL RANDOM WALK

Considering a node's close neighbors, the local random walk may mimic a random walk from a source node to a destination node. Local random walk is predicated on the assumption that nodes with comparable neighbors are more likely to be linked. LP in complex networks is common, typically employed inside likeness-based algorithms. By taking the network's local structure into account, local random walks may increase the accuracy of LP. Several research has employed this method with encouraging results for foreseeing missing connections in complicated networks.

Zhang, Lin, et al. [13] established an auxiliary optimization approach for public transit route networks (PTRN) utilizing LP in 2018. They used Space R to evaluate Jinan's PTRN's topological features and gathered LP summary indices and algorithm sets. Structural likeness-based LP would succeed since the network is a typical small-world network with a high average clustering coefficient. They chose Jinan's PTRN's three most accurate indices for auxiliary optimization depending on LP. The network topology was steady and organized except for a limited section that needs optimization and restoration.

Berahmand et al. [14] suggested a variant of DeepWalk that combines network topology with node characteristics in 2021 for LP in attributed networks. A novel random walk model is presented based on the hypothesis that two nodes on the network will be connected if they are geographically close or have other characteristics. The suggested approach is compared to other cutting-edge network embedding techniques and tested on six real-world attributed networks. According to the findings, a connection is more likely to form among two nodes with a comparable structure and set of attributes.

Kumar et al. [15] 2022, the LGQ model uses feature sets from different combinations of the L, G, Q, LG, LQ, GQ, and LGQ indices to improve ML-based LP. The local likeness was determined by CN, AA, JC, and PA (Common Neighbors, Adamic/Adar Index, and Jaccard Coefficient), whereas cos+, ACT, SP, and MFI measured global likeness. LP and L3 quasi-local indices were used. LGQ was tested using seven reference methods and six popular dynamic network datasets. The LGQ model and its modifications beat the baseline techniques in AUPR, F1 score, BAC, AUC, and other parameters. They also examined the accuracy of prediction models depending on Neural Networks and Xgboost for many variants of the suggested feature sets.

In 2023, Li, Wenjun, et al. [16] resolved the MSN LP Problem (LPP). The approach may determine reliable routes by assessing each Link in node communication channels. In network maps, the weighted network shows connection significance. MSN topological properties determine interlayer and intralayer connections and their respective significance. A weighted network and trustworthy paths were used to create the Local Random Walk measure of likeness. In order to capture the structure of networks, this metric employs a random walk to find commonalities and new connections. The approach was tested on seven different, authentic MSN datasets.

2.3. MACHINE LEARNING-BASED METHODS

NetworkGAN, developed by Yang, Min, et al. [17] in 2019, is a cutting-edge technique for accurate temporal LP. To simulate the geographical and temporal characteristics of dynamic networks, NetworkGAN employs deep learning methods. Graph convolutional networks (GCNs) and temporal matrix factorization (TMFs), This technique combines generative adversarial networks (GANs), long short-term memory networks (LSTMs), and others. First, a thorough GCN discovers the spatial properties of dynamic networks. We use a TMF-enhanced attentive LSTM to record temporal relationships and predict the network snapshot at the following timestamp. A GAN framework improves temporal LP after a discriminative model trains the deep generative model in an adversarial process.

A methodology for anticipating cross-industry trends of technology convergence was developed by Cho et al. [18] in 2021. utilizing ML approaches with different LP indices, they built a network of inter-process communication co-occurrences utilizing association rule mining to foresee where technologies would merge. Next, we utilized a topic modeling method called latent Dirichlet allocation (LDA) to find terms relevant to the expected merging technologies. In 2012 and 2014, the USPTO granted patents using this approach in chemical computing and ecological technology. Empirical results show that over a 4-year time horizon, the proposed framework's RF model yields the best reliable forecasts.

LP-based Network Representation (LPNR) was developed in 2021 by Gu, Weiwei, et al. [19], and it generalizes the most recent graph neural network to maximize a specially crafted objective function that maintains linkage structures. Superior accuracy in measuring node centrality and community connectivity and highly accurate performance in the LP challenge discovery are all possible because of the meaningful node representations that LPNR can learn. Three different real-world networks are used in experiments to demonstrate LPNR's efficacy. Using the mini-batch and fixed sampling technique, LPNR can learn to embed even very big graphs in only a few hours.

To automatically extract the best attributes for LP, Keikha et al.[20] presented a novel LP framework named "DeepLink" in 2021. DeepLink is an LP system that eliminates the need for human-created features. For optimal performance, the framework uses structural and content data. Several approaches of LP may be utilized to generate a wide variety of structural feature vectors. During structural feature learning, the framework can absorb all proximity orders shown on a network. DeepLink was tested using the Telegram and irBlogs real-world social network datasets.

In 2022, Anand et al. [21] suggested utilizing NSMLLP to forecast links. Each pair of network nodes has a set of characteristics generated by combining their centralities, similarities, and the results of ML classifiers. Each node's popularity is measured, and the likeness among any two pairs of nodes is assessed.

A novel graph embedding approach, informed by findings in network science, was suggested by Kerrache et al. [22] 2022. Using several real-world networks, They tested their link projection method, which depends on the spreading-likeness and local attraction theories. The experimental results demonstrated that their method is superior to state-of-the-art graph embedding techniques and stable in sparse data conditions and varying embedding dimensions.

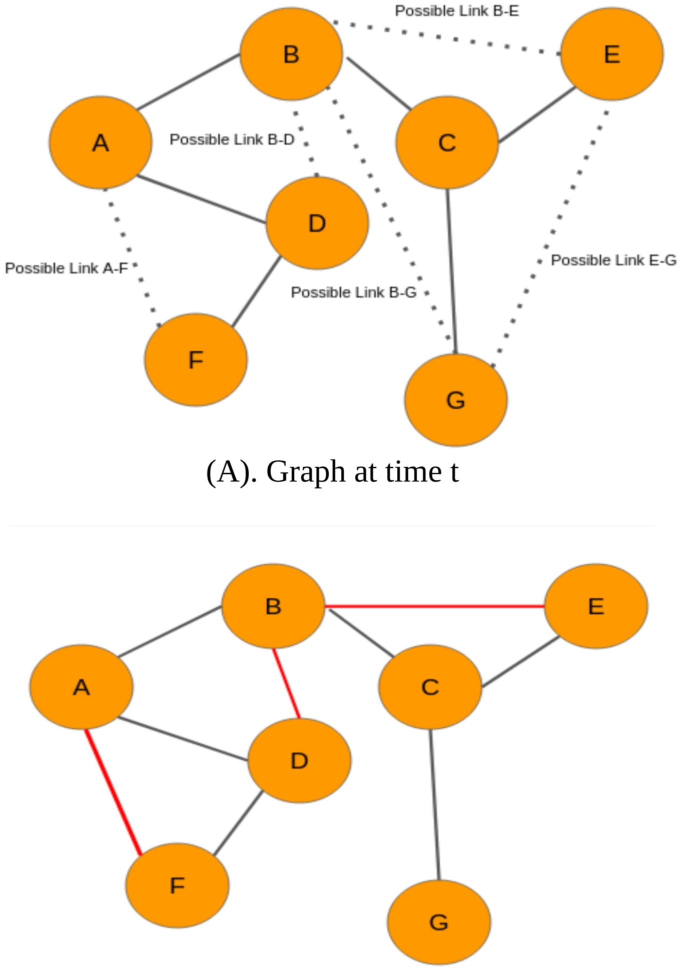
PART 3

LINK PREDICTION IN COMPLEX NETWORKS

3.1. OVERVIEW

In recent decades, computer science has witnessed substantial growth due to technological advancements, the evolution of the Internet, and the proliferation of digital communication networks [1],[2]. As a result, the number of computer science publications has increased considerably, making it challenging to navigate the vast research literature. LP has emerged as a powerful technique for analyzing and visualizing the relationships among different publications and researchers in this context.

LP is a technique that aims to predict the likelihood of a link forming among two nodes in a network [1],[2],[23]. In the case of computer science publications, the nodes represent the publications themselves, and the links represent the relationships among them, such as co-authorship, citation, or reference [24], [25]. By predicting these links, researchers can better understand the structure of the network of publications and identify important papers and authors.



(A). Graph at time t

Graph at time t+n. (B)

Figure 3.1. (A and B) represent the example of complex network analysis [26].

LP in computer science publications has been the subject of extensive research in recent years. Several techniques have been proposed, ranging from simple heuristics to complex ML models. However, most of these techniques have focused on analyzing the citation network, representing the connections among papers depending on their references. Although this provides valuable information, it does not capture the full complexity of the publication network in computer science.

Using complex network analysis, this thesis suggests a new method for predicting links in academic papers on computer science. We have produced a data collection detailing the interconnections among publications regarding authors and citations. SVM, RFC, and LR are the classifiers we use on this dataset. These classifiers have proven useful in many settings and see extensive usage in the field of ML.

Our dataset consists of information about the co-authorship and reference relationships among publications in computer science. We use complex network analysis to investigate the structural properties of the publication network, such as the grade distribution, clustering coefficient, and amazingness centrality. We then apply the SVM, RFC, and LR classifiers to this dataset and evaluate their performance.

Depending on our findings, the complex network analysis-based strategy we presented is superior to other approaches that depend exclusively on the citation network. We discovered that the SVM classifier provided the greatest Accuracy for predicting relationships among articles. In addition, our investigation into the network's underlying structure uncovered evidence of small-world and scale-free behaviors in the publishing sphere. This hints that our suggested method may be used in other research areas.

In conclusion, we have proposed a novel approach to LP in computer science publications that depend on complex network analysis. We have demonstrated that our approach outperforms existing methods that rely solely on the citation network. Our findings suggest that complex network analysis can provide a valuable tool for analyzing and visualizing the structure of publication networks in computer science and other scientific domains. Future research could focus on extending our approach to other fields and domains and exploring its potential applications in various areas.

3.2. GRAPH THEORY

It is the branch of mathematics concerned with studying graph and network characteristics. A graph is a mathematical structure with vertices (nodes) and edges (links) (sometimes called "lines") connecting the nodes (figure 3.2). Computer science, physics, the social sciences, and even computer science itself may all benefit from using graphs as a modeling tool [2],[23][27].

Edge type, orientation, weights, and the total amount of edges are only a few features that may be used to categorize graphs.

1) In a simple graph, the edges are not weighted and are undirected. Since there is no discernible pattern to the edges, they may be assumed to reflect purely binary connections among nodes.

2) Directed edges are present among nodes in directed graphs (digraphs). An edge's orientation specifies the path along which two nodes are connected.

3) Various linkages, including self-loops, may exist among nodes in a pseudograph. They may be represented as stacked graphs with the same nodes but only one kind of edge, and they can include both directed and undirected linkages. This notation may also depict multi-graphs where each edge has many labels or kinds .

4) Each edge has a certain weight in a weighted graph, often expressed as a real integer. The number of edges among any two nodes in a multi-graph may be used to create a weighted graph .

5) Hypergraphs have edges called hyperlinks or hyperedges, connecting more than two nodes. A folksonomy is a hypergraph of nodes representing people, resources, and tags. Users contribute content such as papers, photographs, audio files, connections to other websites, and other online resources to the network. Words or phrases that describe a resource are called tags. An 'edge' in the network represents a connection among three nodes (a user node, a resource node, and a tag node .

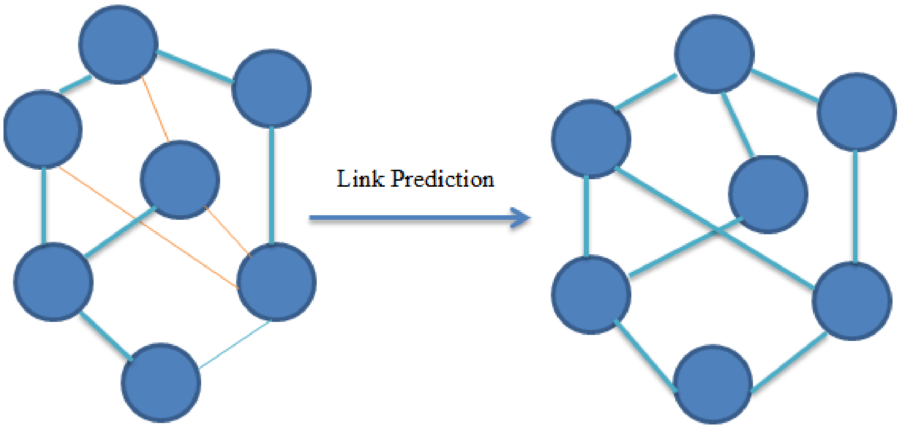


Figure 3.2. Graph Embeddings for Link Prediction.

3.2.1. Essential Concepts

The following are some important and basic concepts in graph theory. Vertex grade, path, Graph connectivity, Subgraph, and Maximum flow problem.

- Vertex grade

A vertex's grade in a graph without direction is denoted by the symbol How many edges have v as an occurrence vertex is denoted by G. The amount of edges that are directed toward v and away from v, respectively, makeup v's in-grade and out-grade, respectively, in a directed graph. Deg(v) is often used to represent the level of a vertex [27],[28].

$$\deg(i) = \sum_{j=1}^{n} A(i,j) \tag{3.1}$$

An adjacency matrix (sometimes termed a "graph matrix") in graph theory depicts how a graph is linked. A graph with n vertices may be characterized by an n-by-n matrix where each component A(i,j) is one if vertex i is related to vertex j by an edge and 0 otherwise. Vertices are near. An (i,j) may indicate the edge weight among points i and j in a weighted graph. Diagonal entries in the adjacency matrix indicate self-loops in directed graphs, generally 0. Diagonal entries are frequently 0 in graphs without direction.

- The path

It consists of a series of edges that connect a group of nodes. A route using the origin and destination vertices is considered simple. A path's length is equal to the sum of its edge counts. Among any two vertices in a graph, the route having the smallest length is known as the shortest path. Many techniques have been devised to determine the shortest route among two vertices in a graph since this is a basic topic in graph theory. Among the most well-known algorithms, Dijkstra's is among the [27],[29].

- Graph connectivity

If an edge can reach every pair of vertices in the graph, then we say the graph is linked. Divining an unconnected graph into two or more associated subgraphs is possible. Algorithms like depth-first and breadth-first search may be used to determine the total amount of vertices in a graph. [27],[30].

- Subgraph

It is a graph created by eliminating nodes and links from another graph. When describing a graph, "spanning subgraph" refers to a subgraph that includes all the vertices. A tree has no cycles, making it a linked acyclic graph. A tree with n nodes has n minus one edge. Since trees may be grouped to form a forest, the graph can contain more than one "tree [27]. "

- Maximum flow problem

A graph represents a network, and the edges represent pipes or channels that can transport some material, such as water or data. The maximum flow problem seeks to determine the maximum amount of material via the network from a starting point to an ending point. The Ford-Fulkerson and the Edmonds-Karp algorithms can solve the greatest flow issue [31].

Changing a graph's vertices so that no two neighbors share a color yields the chromatic amount of the graph. This fundamental idea in graph theory has several uses in computer science, particularly in graph coloring difficulties.

3.3. COMPLEX NETWORKS ANALYSIS

Early sociologists like George Simmel and Émile Durkheim [32],[33] recognized the theoretical merit of delving into patterns of interactions among social actors, laying the groundwork for what is now known as Social Network Analysis (SNA).

Numerous independent components engage in nonlinear interactions within a complex network [32],[33],[34].

Cells are networks of molecules linked by molecular interactions, and the nervous system consists of a network of nerve cells connected by axons [35],[36]. In addition, societies are complex interdependent webs of individuals linked by a wide range of interpersonal exchanges. At the ecological and food web sizes, predator-prey interactions may be represented as networks [37],[38]. Technological networks include the internet (a collection of related web pages), router networks, power grids, and transportation systems.

Graph theory has become an important method for studying intricate webs of connections. It has been widely used across biology, physics, telecommunications, computer science, and more for network research while originating in sociology and mathematics. Structured network analysis, temporal network evolution analysis, content-based network analysis, and more are all subfields of graph theory [39],[40]. Structural analysis of networks focuses on understanding the network's architecture, including the arrangement of nodes and edges and their connectivity patterns [41]. On the other hand, temporal analysis is concerned with the evolution of networks over time, including the study of how network properties change due to the addition or removal of nodes or edges [42]. Content-based analysis, as the name implies, involves analyzing the content of network components, such as the messages exchanged among nodes or the attributes of the nodes themselves [43],[44].

Complex networks are found in various biological, social, and technological systems, and their analysis is crucial to our understanding of these systems. Graph theory provides a powerful tool for exploring network properties, and its various branches offer different perspectives on network structure and function.

3.3.1. Complex Network Characteristics

A variety of complex networks exhibit similar topological characteristics that are shared across different domains. The following are among the most salient and significant features.

- **Connectedness**. Nodes in complicated networks tend to cluster into subnetworks. Subgraphs of a network with a route among every possible pair of nodes are called "connected components"; this implies that all vertices are connected to every other vertex. One or two of these components are often much bigger than the rest of the network, but the network as a whole also contains a great amount of smaller components[34],[45],[46].
- **Grade distribution (DD).** This means the likelihood that a node in a complicated network has k neighbors. As a functional connection among two variables, a power law describes the DD of these networks by describing the variation of one quantity as the power of another. Therefore, the distribution has a narrow peak followed by a lengthy tail [47],[48]. This kind of network was given the name "scale-free networks" by Barabasi [49],[50]. Numerous nodes with low grades and few with high grades characterize a network where the grades follow a power law dispersion. Increases in the power-law coefficient suggest a steeper decline in the grade distribution curve, whereas increases in the power-law coefficient indicate a more even distribution of node grades.
- **Clustering coefficient (CC).** Many real-world networks are transitive, meaning that pairs of nodes linked to the same node often form new connections [51],[52]. This is the social counterpart of the "friend of a friend is probably a friend of mine" principle. The local CC depends on this connectivity among nodes and their neighbors. Triangle counts in networks are also provided [53]. The formula [54],[55] for the local CC or local transitivity of a node vi V in a graph G = < V, E > reads as follows.

$$\mathrm{Cc}\ (\mathrm{vi}) = \frac{\mathrm{Ntriangles(vi)}}{\mathrm{Ntriples(vi)}} \tag{3.2}$$

The amount of triangles containing node vi is denoted by Ntriangles (vi), while the amount of triples produced at node vi is denoted by Ntriples (vi). The local CC is the ratio of linkages among a set of nodes to the maximum possible number of links. In an undirected graph, Ntriples (vi) = $\frac{ki(ki-1)}{2}$ and Ntriangles(vi) =| {(vj,vk) . vj,vk ∈ Γ(vi),(vj,vk) ∈ E} |.

The CC of a graph can be defined as the arithmetic mean of the local CC of all nodes, which is calculated by dividing the sum of the CC of individual nodes by the total amount of nodes in the graph [56],[57] Mathematically; it can be expressed as.

$$\text{Cc (G)} = \frac{1}{|\text{V}|} \sum_{vi\in\mathbf{V}} Cc(vi) \qquad (3.3)$$

V represents the set of all nodes in the graph, and Cc (vi) denotes the clustering coefficient of node vi.

Complex networks generally exhibit a significantly higher average CC than simple networks.

- **Average distance.** The average length of a route may be taken among any two nodes in a network. It is determined by adding up all the shortest paths among pairs of nodes and dividing that amount by the amount of pairs of nodes. This amount is rather low in many real-world complex networks, suggesting high connectivity among nodes [58],[59].

The average distance in an unweighted N-node graph G is calculated by dividing the sum of all shortest route lengths among any two nodes by the total amount of node pairs in the network.

$$Distance_{avg}\ \text{(G)} = \frac{2}{\text{N.(N}-1)} \sum_{vi\,,vj\,\in\mathbf{V}} dist\ (vi\,,vj) \qquad (3.4)$$

- **Diameter.** A graph's diameter refers to the shortest path's largest possible length among any two nodes within the graph [60],[61]. It can be defined formally as.

$$\text{Diameter (G)} = \max (\{\text{dist} (vi , vj)\ \forall\ vi , vj \in V\}) \tag{3.5}$$

The shortest route among two nodes, u and v, is denoted by d(u, v), where V is the collection of all nodes in the graph.

O(N^2) is the well-known computational difficulty of computing the diameter of a graph, where N is the total amount of nodes in the network. In complicated networks, however, the diameters are often much less than the total network size.

To compute the diameter of a network, the network must be connected. In the absence of connectivity, the maximum value of the shortest path among the connected nodes is considered. Alternatively, the average of the connected components' diameters can also be considered [62].

- **Density.** The number of sides in the graph is divided by the greatest possible number of edges [63],[64]. The density can be mathematically expressed for a graph G =< V, E >.

$$\text{Density (G)} = \frac{2|E|}{|V|\ \text{x}\ (|V|-1)} \tag{3.6}$$

Here, |V| represents the number of nodes in the graph, and |E| denotes the number of edges. The factor of 2 is included in the numerator to account for each edge connecting two nodes in the graph.

Complex networks are often depicted by a relatively low density, indicating that they are highly sparse [65].

- **Community structure**. Cluster nodes in components and communities often develop in complicated networks. When nodes in a network have commonalities, they form communities, which are sub-graphs of the network. Typically, there are more connections among nodes inside a community than those outside the community [66],[67]. Figure (1.3) depicts a sample community's basic organizational setup. In a network, there may be overlapping or non-overlapping communities.

In addition, real-world networks often demonstrate temporal variations, which may lead to the emergence and demise of vertices and edges for the network's lifetime. Because of this, the features of the graph, such as its average grade, density, average clustering coefficients, etc., might change over time as a result.

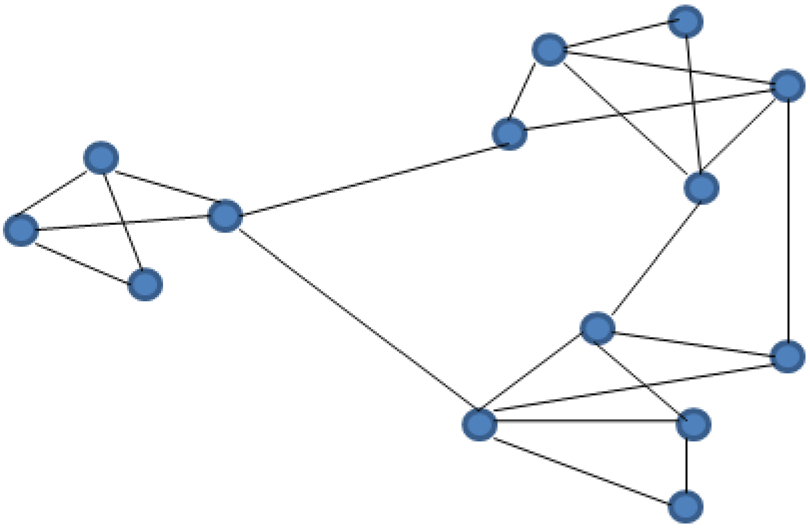


Figure 3.3. Networks with complicated community structures.

3.4. LINK PREDICTION

Predicting the probability of a connection among two nodes in a network is the focus of this subject of network analysis. Social media platforms, citation networks, and networks of biology are just a few examples of where this issue has surfaced. The purpose is to identify missing connections and extrapolate the system's structure. [68],[69].

One of the simplest and most widely used methods for LP is the common neighbor's method. This method assumes nodes with many common neighbors will likely be connected [68],[69]. The amount of common neighbors among two nodes, i and j, can be calculated using the following formula

$$\text{CN (i,j)}=|N(i) \cap N(j)| \quad (3.7)$$

Node i have neighbors who make up the set N (i), and node j also has neighbors who make up the set N (j). More often than not, nodes i and j will be linked in the future if has a high value.

The Jaccard coefficient approach is another common technique for LP. The number of shared neighbors among two nodes is used to calculate their likeness, with the total number of neighbors serving as a normalizer. [70]. The Jaccard coefficient among two nodes i and j is given by.

$$\text{JC (i,j)} = \frac{|\text{N(i)} \cap \text{N(j)}|}{|\text{N(i)} \cup \text{N(j)}|} \quad (3.8)$$

where |N(i) N(j)| is the sum of the neighboring i and j nodes, and |N(i) N(j)| is the number of neighbors that both i and j have in common. The value of JC (i,j) will increase if there is a high chance that nodes i and j will be connected.

The preferred attachment model is a more sophisticated strategy for predicting links. According to this theory, a higher grade at a node increases its chances of gaining linkages in the future [71],[72]. The number of edges passing through a given node, i, is its grade. As i increase in grade, so does the likelihood that a new link will be linked to it, as provided by.

$$\text{JC (i,j)} = \frac{\text{k(i)}}{\sum \text{j k(j)}} \quad (3.9)$$

Where k(i) is the grade of node i, the summation is over all nodes j in the network. The higher the value of PA (i), the more likely it is that node i will attract new links.

Another method for LP depends on the concept of node likeness. The assumption is that nodes similar in their properties or attributes will likely be connected. The likeness among nodes i and j can be computed using various likeness measures, such as cosine likeness, Pearson correlation, or Euclidean distance. The likeness score among nodes i and j can be used to predict the probability of a link using LR or a decision tree classifier.

PART 4

MACHINE LEARNING

4.1. OVERVIEW

Machine learning (ML) is a branch of artificial intelligence (AI) focused on creating algorithms and models that enable computers to learn from data and improve their performance without human intervention. The purpose of machine learning is to teach computers to draw conclusions and draw conclusions from data. Because they can be trained on massive datasets to improve Accuracy and efficiency, ML models are good tools for handling complex challenges in many domains. There are several types of ML, each with its own set of benefits and drawbacks; examples include supervised learning, unsupervised learning, and reinforcement learning.

4.2. SUPERVISED LEARNING

Supervised learning is an ML approach where machines are trained using labeled data, allowing them to predict outputs depending on that data [73],[74]. Labeled data refers to input data tagged with the correct output, which guides the machines to learn how to predict outputs accurately[73].

In the same way that a teacher guides and instructs their students, training data functions as a coach to educate computers on making accurate predictions of future results. This method has several applications, including computer vision, voice recognition, and NLP. [75],[76].

By using algorithms that can learn from labeled data and make predictions, supervised learning can generalize to new, unseen data. It has led to significant advances in image and speech recognition and has become an essential tool in developing various artificial intelligence applications. By leveraging labeled data to train ML models, supervised learning enables more accurate and efficient predictions, ultimately leading to improved decision-making and outcomes.

Modeling a system using input data x X and its matching output (response) y Y is supervised learning, an ML approach. X denotes the range of admissible input values. In contrast, the range of admissible output values is denoted by Y. Finding a decision function that reliably predicts the response of incoming samples given the observable input state x X [75],[76], [77] is the objective of supervised learning.

For binary classification issues, the training samples inform a decision function that will ultimately place unseen data into one of two classes. The labels Y = 1 and +1 make up the response set of the function, and a new sample is classified as either a member of the first or second group.

The function ψ in supervised learning represents the decision equation that converts x into y. The equation for the function ψ can vary depending on the specific problem and algorithm used.

$$\psi(x) = \text{sign}\ (w\text{^}T\ x + b) \qquad (4.1)$$

Where w is the weight vector, b is the bias term, and the sign function assigns the output label depending on whether the function's value is positive or negative.

In more complex problems, such as image recognition or natural language processing, the decision function can involve using neural networks or other deep learning techniques, which can have a more complex structure and involve many layers of interconnected neurons. In these cases, the function ψ can be represented as a series of connected nodes or layers, each performing a specific operation on the input data to produce the final output prediction [76],[78],[79].

4.3. MACHINE LEARNING TECHNIQUES

4.3.1 Logistic Regression

To predict binary or categorical outcomes in ML, statisticians turn to LR [80]. It is a kind of regression analysis in which the chances of something happening are calculated using data from several independent factors. The dependent variable in LR is binary, taking on only the values 0 or 1 (figure 4.1). There is flexibility in the form of the independent variables. Finding the optimal model for predicting the probability of the outcome variable from the input variables is the purpose of LR [81],[82],[83].

In the LR model, the likelihood of the result of the parameter is modeled by a logistic function. The logistic function is an S-shaped sigmoid function. A sigmoid function converts an input number to a likelihood between 0 and 1 between those extremes.

$$p = \frac{1}{(1 + e\hat{}(-z))} \qquad (4.2)$$

When z is the linear relationship among the independent parameters, p is the likelihood that the event will occur, e is the base of the logarithm of natural numbers, and the equation shows the Linear combining of the separate variables.

$$z = b0 + b1x1 + b2x2 + ... + bn*xn \qquad (4.3)$$

The intercept is denoted by b0, the independent variable coefficients by b1 through bn, and the values of the independent variables, x1 through xn, by the bracketed expression. For each independent variable, the LR model's coefficients express the proportion by which the log odds of the event happening shift for every one-unit shift in that variable. The log odds are the reciprocal of the odds ratio, which is the ratio between the chances of an event happening and the odds of it not happening [84].

The highest likelihood (LR) calculation is used throughout the training process. Finding the coefficient values that increase the probability of what was seen given the model is the purpose of maximum likelihood estimation [85]. The likelihood function is the equation that represents the probability that the data will be seen given the model.

$$L = \prod (p^{yi} * (1-p)^{(1-yi)}) \tag{4.4}$$

For each observation in the training data, we take the product yi(0,1), p(0,1), where yi is the observable result variable, and p(0,1) is the expected likelihood of the final result variable. When multiplying very tiny probabilities, underflow errors may occur if using the likelihood function directly; hence the model of the probability function is utilized instead. The equation for the log-likelihood operation, which is the logarithm of the likelihood function, looks like this.

$$l = \sum (yi*\log(p) + (1-yi)*\log(1-p)) \tag{4.5}$$

Finding the coefficient values that optimize the log-likelihood function is the key to fitting the LR model. Mathematical enhancement methods like descent gradient and Newton's approach are often used for this purpose [82,[86].

One-vs-all and softmax regression methods show how LR may be adapted to handle multi-class classification issues. To do one-versus-all LR, an LR model is fitted for each class independently, and the class with the greatest predicted probability is then forecasted. Fitting a single LR model that predicts the probability of each class and normalizes the probabilities using the softmax function is what softmax regression is all about[81],[87].

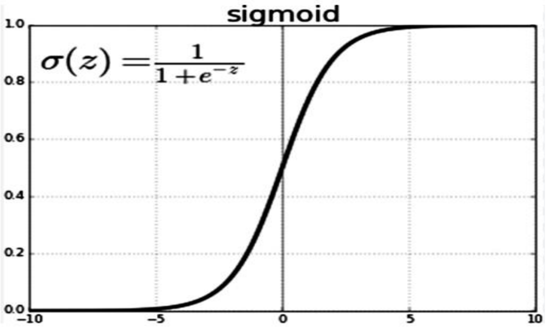


Figure 4.1. LR with sigmoid [88].

4.3.2. Support Vector Machines (SVM)

(SVMs) are an effective machine learning (ML) method for categorization and retraction. Using a cost function that imposes penalties for incorrect classifications, they locate the hyperplane that maximizes the gap between the two groups. To process non-linearly separable data, SVMs may be modified to use a kernel function to translate the input data into a higher-dimensional feature space. Figure 4.2 illustrates how the input data and the desired qualities of the decision boundary play a role in determining which kernel function is best suited for use with support vector machines [89],[90].

Support vector machines aim to find the excessive level that optimizes the margin between the two classes. Distance from the hyperplane to the nearest data points in each class constitutes the margin. In order to achieve the greatest margin while still accurately categorizing all of the training data points, the excessive level is selected in such a manner [89],[90],[91]. The equation stands for the excessive level that divides the two groups.

$$w^T x + b = 0 \tag{4.6}$$

Weights are denoted by w, input data by x, and bias by b. Because the weight component w is perpendicular to the excessive level, it sets the excessive level initial orientation in the input space. The excessive level is moved to the correct location in the input space by the bias term b. SVMs use cost functions that punish incorrect classifications for determining the best excessive utilization level. This is what we mean by "cost function."

$$C * \Sigma(\max(0, 1 - y_i(w^T x_i + b))) \tag{4.7}$$

For the i-th training data point, the variable of interest is y_i (which can be either 1 or -1), the input component is x_i, and the expected result is $wT x_i + b$, where C is the cost parameter that regulates the trade-off between greatest the margin and reducing the ranking mistakes.

To process non-linearly separable data, SVMs may be modified to use a kernel function to translate the input data into a higher-dimensional feature space. The kernel function maps the input data into a new space where a linear decision boundary may be located. The kernel function is selected according to the input data's features and the decision boundary's desired attributes. The linear kernel, polynomial kernel, and radial basis function (RBF) kernel are only a few examples of popular kernel functions [89],[90]. Since the RBF kernel is both computationally economical and capable of handling complicated decision boundaries, it is often used in support vector machines. The formula for the RBF kernel is.

$$K(x, x') = exp(-gamma * ||x - x'||^2) \tag{4.8}$$

Where x and x' are two input components, gamma is the kernel parameter that determines the size of the Gaussian function, and the square of the Euclidean distance among them is $||x - x'||^2$. The RBF kernel projects the input data into an infinite-dimensional feature space to locate a linear decision boundary.

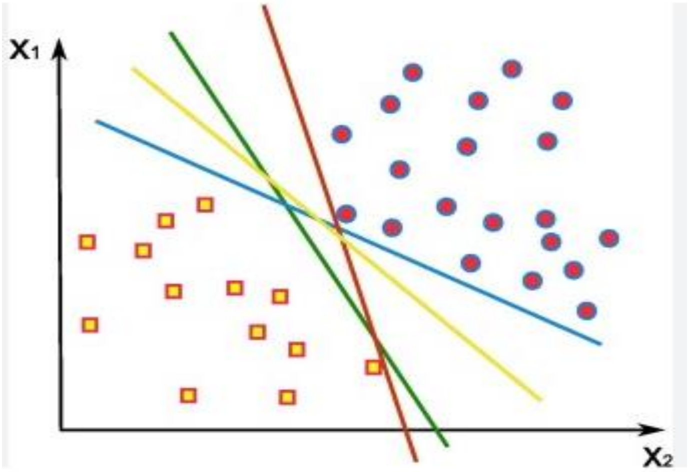


Figure 4.2. Support vector machine [92].

4.3.3. Random Forest (RF)

The ML method RF is widely used for both ranking and regression purposes. It is a kind of ensemble learning in which many different decision trees are used to create a single prediction. Accuracy, scalability, and high-dimensional data management are hallmarks of RFs [93],[94]. Different input data sets train many decision trees (figure 4.3). Each decision tree is taught to use a predetermined set of attributes to determine what category a given data item belongs to. In order to prevent overfitting and boost the classifier's precision, the trees are generated using a random subset of input characteristics [95]. The final Prediction depends on the votes of all the separate decision trees. For each input data point, the decision tree with the most votes for a certain class is chosen as the projected class.

The decision trees in an RF classifier are constructed using a recursive partitioning algorithm. The algorithm selects a feature that best separates the training data depending on a specified splitting criterion. The splitting criterion typically depends on impurity measures such as entropy, Gini index, or classification error. The splitting criterion determines the optimal feature and threshold for splitting the data into two subsets. When a stopping requirement is reached, such as achieving the deepest level or the smallest amount of data points in a node's leaf, each subset is repeated recursively. The output of a decision tree is a binary decision depending on the input features. The output of an RF classifier is the majority vote of the individual decision trees. The probability of the predicted class can also be estimated depending on the proportion of decision trees that predict each class [96],[97]. It can use this equation to explain the RF classifier mathematically..

f(x) = argmax(c) Σ(w_i * I(T_i(x) = c)) (4.9)

Where f(x) is the predicted class for input data point x, T_i(x) is the decision tree i, c is the class label, w_i is the weight assigned to the decision tree i, and I(T_i(x) = c) is an indicator function that equals one if the decision tree i predicts the class label c for input data point x and 0 otherwise.

Each decision tree may be given a different weight depending on how well it predicts or how critically important the information it uses is. The weights may be adjusted accordingly to ensure that each decision tree contributes an equal amount to the final Prediction. There are several ways in which RFs excel above other types of ML algorithms. They are adept at handling large, complex datasets and have a lower propensity for overfitting [95]. They can also withstand some grade of random variation or missing information. It has been shown that RFs are effective for both binary and multi-class issues in classification and regression analysis [95],[98],[99].

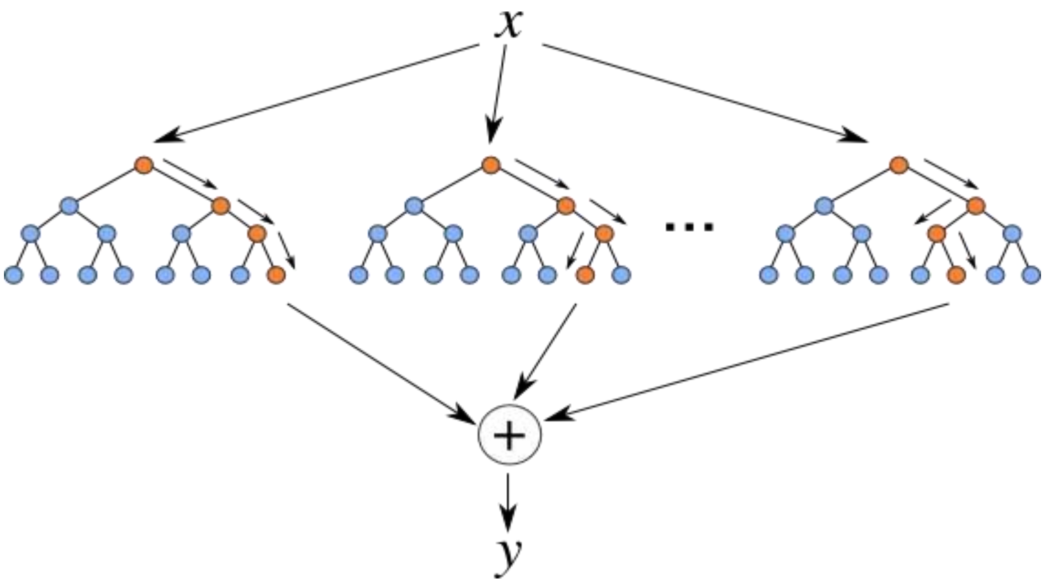


Figure 4.3. RF algorithm.

PART 5

METHODOLOGY

This chapter will present formulas and provide a detailed explanation of the complexity underlying LP algorithms. Subsequently, an exposition of the datasets employed in the study will be offered, followed by the necessary steps for data preparation. Finally, an in-depth discussion will be conducted on the particulars of the examination process. By delving into the intricacies of the LP algorithms, we aim to shed light on their mathematical foundations and elucidate the underlying principles driving their predictive power. The datasets used in this study will be described in detail, including their sources, characteristics, and preprocessing techniques applied.

5.1. EXPLORING GRAPH-BASED LIKENESS MEASURES

5.1.1. Common Neighbor (CN)

The common neighbor classifier is an ML technique for foretelling relationships in social networks and other graph-based data structures. The method determines the likelihood of a link between two nodes in a network by counting the number of common neighbors between them. The method operates on the premise that connected nodes are more likely to share neighbors than their unconnected counterparts [100],[101]. To describe this behavior, use the following equation.

$$P(i,j) = \Sigma\_k\ I(A\_ik = 1) * I(A\_jk = 1) \tag{5.1}$$

A_ik denotes the adjacency matrix entry between vertices i and k, and an indicator function, I(A_ik = 1), evaluates to 1 if a connection exists between vertices i and k and 0 otherwise.

One limitation is that it does not consider the network's grade distribution [102]. In networks with highly skewed grade distributions, nodes with many neighbors are likelier to share common neighbors than nodes with few neighbors, even if they are not directly connected. This can lead to false positives in LP. Despite its limitations, it is a simple and effective algorithm for LP in social networks and other graph-based data structures. It can be applied to various problems, including recommender systems, collaborative filtering, and social network analysis [102],[103].

5.1.2. Adamic Adar (AA)

In LP tasks in social networks and other graph-based data structures, it serves as a measure of how similar two nodes are [72]. The technique relies on the hypothesis that more connections will be made between nodes that share few neighbors. The equation for this is as follows.

$$AA\ (i,j) = \Sigma\_k\ I(A\_ik = 1) * I(A\_jk = 1) * (1/\log(d\_k)) \qquad (5.2)$$

Where AA(i,j) is the Adamic Adar coefficient among nodes i and j, A_ik is the adjacency matrix entry among nodes i and k, and I(A_ik = 1) is an indicator function that equals one if there is a link among nodes i and k and 0 otherwise. The term (1/log (d_k)) is a weighting factor that reflects the rarity of the neighbors shared by nodes i and j. The logarithmic function avoids giving too much weight to nodes with very high grades.

It can also be extended to include additional features, such as the preferential attachment score or the clustering coefficient, which capture additional aspects of the network structure [104]. These features can further improve the accuracy of the LP techniques. Despite its effectiveness, it has some limitations. It assumes that the network is static and does not take into account the temporal dynamics of the network. It also assumes that the rare neighbors shared by nodes i and j are equally important, which can not always be the case [105].

5.1.3. Jaccard Coefficient (JC)

The grade of likeness among two datasets may be calculated using the Jaccard coefficient (JC), also known as the Jaccard likeness coefficient. Common applications include clustering, classification, and recommendation systems [106], all of which fall within the purview of ML. The ratio of the crossroads of two sets to their union is known as the intersection ratio.

$$J\ (A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{5.3}$$

A and B are two collections, and A's cardinality (the number of items in A) and B's cardinality (the number of items in B) are, respectively. The value of JC may be either 0 (indicating that the two sets share no items) or 1 (indicating that they are identical).

The JC is often used to measure the likeness between two text documents. In this case, the sets are the words in each document. The JC can be used to compare the overlap among the words in the two documents, providing a measure of how similar the documents are regarding their content. It can also be used in clustering algorithms, where it is used to measure the likeness among clusters of data points. In this case, the sets are the data points that belong to each cluster. The JC can be used to compare the overlap among the data points in two clusters, providing a measure of how similar the clusters are [106],[107],[108].

One limitation of JC is that it does not consider the frequency of occurrence of the elements in the sets. Two sets with a high grade of overlap but with different frequencies of elements can have a low JC [107]. To overcome this limitation, the Jaccard index can be used, which considers the frequency of occurrence of the elements in the sets.

$$JI(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{5.4}$$

Where |A ∪ B| is the sum of the frequencies of the elements in the two sets.

5.1.4. Preferential Attachment (PA)

In network research, the PA describes the phenomena where higher-grade nodes draw more linkages than lower-grade nodes. To what extent a network displays preferential attachment may be measured using the PAC [109],[110]. The PAC is the constant-factor-normalized ratio of the product of any two nodes' grades to the square of the network's total amount of grades.

$$\text{PAC (i,j)} = \frac{k\_i * k\_j}{(2m)\text{\^{}}2} \tag{5.5}$$

In this case, 2m represents the total number of edges in the network, while k_i and k_j represent the quality of nodes i and j, respectively. If the PA is 1, all new edges are connected to the nodes with the highest grade; if it is 0, new edges are connected at random.

Machine learning programs like LP and community detection may benefit from the PAC. Using the two nodes' grades and the network's aggregate grade, the PAC can forecast the probability of a new connection forming between them in LP. A larger PAC indicates a higher probability of connectivity between nodes. Based on their ratings and the network's overall grade, the PAC can determine which nodes are more likely to belong to the same community. [109],[111].

One of the main features of the PAC is that it is a simple and efficient measure that can be calculated quickly for large-scale networks. It can also capture the heterogeneity of the network's grade distribution, a common feature of real-world networks. However, the PAC has some limitations. For example, it assumes that nodes with a higher grade are always more attractive to new links, which is not always true in practice. It also does not consider the network's topology, such as the presence of communities or the grade of correlation among nodes [112].

5.1.5. Resource Allocation Index (RAI)

It is an approach used in ML to evaluate the performance of multiple models when applied to a single task. RAI assesses how efficiently resources, such as computational time or memory, are allocated among different models to optimize their overall performance [113]. The formula of RAI can be expressed as follows.

$$\text{RAI} = \frac{\text{accuracy}}{\text{Computational cost\^{}k}} \tag{5.6}$$

Accuracy is the model's performance metric, such as precision or F1 score; computational cost is the resource consumed by the model, such as CPU time or memory usage; and k is a scaling exponent determining the relative importance of accuracy and computational cost.

RAI allows for comparing models with different performance and resource requirements and helps identify the most efficient models. For instance, if two models have similar accuracy, but one consumes fewer computational resources, the RAI will be higher for the more efficient model [113],[114].

RAI is good equipment for optimizing the performance of ML models, particularly in resource-constrained environments such as mobile devices or embedded systems. It can also help researchers better understand the trade-offs among accuracy and computational cost in ML and develop more efficient algorithms [113],[114].

5.2. DATASETS

The dataset utilized in this study was sourced from three distinct origins: (1) web of science; (2) microsoft academic; and (3) google scholar. This dataset had been collected in year 2022.

These websites contain significant metadata alongside citations to digital renditions of published materials. Within the given dataset, individual nodes correspond to distinct authors of scientific articles, while an edge connecting two authors signifies a collaborative publication involving both individuals. The publication date of the collaborative work can be determined by examining the timestamp associated with each edge. In instances where two authors have collaborated on multiple articles, only the initial publication they co-authored will be considered for the analysis.

The datasets were obtained through a meticulous process involving collecting and compiling articles from the journals above. Subsequently, a systematic categorization and tabulation procedure was employed to create three separate datasets. Computer science, Medical, and Social. In this context, each node within the datasets represents a cluster of authors who collaborated on scientific papers, while the edges symbolize the pooled connections among these authors. By structuring the datasets in this manner, the intricate network of author collaborations can be effectively captured and analyzed, providing valuable insights into the dynamics and patterns of scientific cooperation across different domains.

5.2.1. Dataset Properties

5.2.1.1. Computer Science Dataset

The data were coded, and the data sources were published papers acquired from various places, including scientific publications, conferences, and other places. The dataset had (5410) different nodes. When the code was executed, it returned the amount (838), the number of nodes in the utilized database. This database consists of three columns, each representing one of the following. The name of the piper, the name of the person participating, and the number of nodes that were supplied as a code when the code was executed.

5.2.1.2. Medical Dataset

The data were coded from written studies in various places, such as scientific journals, conferences, and other places. These papers were mostly about medicine and how medical image processing can be used to diagnose diseases. There were (3410) different nodes in the collection. When the code was run, it returned the amount (435) of used database nodes. This database has three columns. Each column shows one of the following. The name of the piper, the name of the person taking part in the piper, and the amount of nodes that were given as a code when the code was run.

5.2.1.3. Social Dataset

The information was coded and derived from textual research in several locations, including conferences, scientific publications, and other venues. Most of these publications dealt with geography articles and the application of social image processing. The collection had (883) unique nodes. The result showed that (237) database nodes were used. Three columns make up this database. The names of the piper, the participant in the piper, and the amount of nodes provided as a code when the code was executed are each shown in a separate column.

5.2.2. Dataset Splitting

Dividing the dataset into a set to be trained and a test collection is recommended, 30 % for the texting set and 70 % for the training set (1/3 ratio), with the training set taking up the biggest portion and the test set taking up the smallest. So, first, we use the training set to hone the model, and then we use the test set to see how well it performed.

There are several approaches to prevent overfitting, including splitting the data set into testing and training sets. The training and test sets must contain patterns like those seen in real-world data, as this improves the model's performance evaluation.
Set validation is essential when choosing among various models and evaluating which performs better, regardless of the model's performance (table 5.1).

Table 5.1. Allocation of data into training and testing set.

Dataset	No. of data	Training set	Testing set
Computer science dataset	5410	3787	1623
Medical dataset	3410	2387	1023
Social dataset	883	618	265

5.2.3. Data Processing

A graph's overall number of edges could be less important than the number of edges that should be used for training. On the other hand, there can be graphs that need a greater amount of edges in order to do this job. The precise percentages that are used to divide depend on a variety of different criteria. The amount of edges utilized for testing should be as few as is practically practicable, yet as many as is required. This indicates that the assessment shouldn't be swayed because only a limited amount of testing node pairs are available. However, access to the maximum amount of information feasible for training should also be available. The same is true about both the feature graph and the training graph. The percentage used to partition the training graph and the feature graph also lowers the features' quality and reduces the number of node pairs in the training set. If the percentage is too high, only a few positive cases in the training set will affect the execution of the little graphs. If the percentage is too low, on the other hand, the feature graph will become extremely fragmented, and the features that are created for the training set will not have any significant significance. Additionally, this will result in deficiencies during training, which will cause the performance of the supervised procedures to deteriorate as a direct consequence.

5.2.3.1. Processing Of Computer Science Dataset

Each node_A from 0 to 838 has been linked to another node_B from 0 to (838) in order to make a pair [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), and so on]. ----------- (619,618)]. After removing duplicate entries, the final database will contain 350,703 rows. Thanks to the previous phase, this ensures that every opportunity has been explored. To construct a pair by connecting two nodes [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7),] -- --------- (619,618)] inside the whole database that is going to be made .

The total amount of rows in the completed database is (350703), and the pairs [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), and (0, 7),] can be found there. ----------- (619,618)]. The value zero in the completed database created in the previous phases indicates that these 347998 nodes are not connected via a link or edge. The data carries a single amount; the rest indicates a connection (link) or edge among these nodes and their respective amount.

According to the previously reported data, the number of nodes that supplied an LP was (2705), whereas the number of nodes that provided a non-link forecast was (347998). The percentage allocation to nodes can be illustrated in Table (5.2) and Figure (5.1).

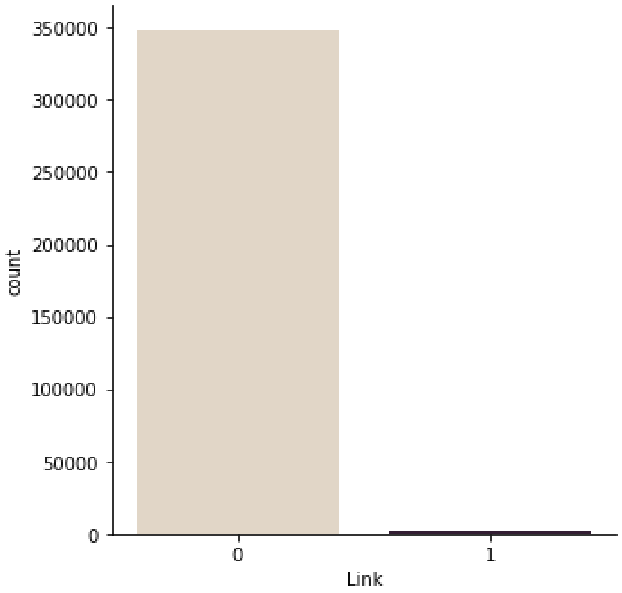


Figure 5.1. Computer science dataset LP diagram.

Table 5.2. Nodes with LP and non-LP in computer science dataset

No. of nodes	LP	%	Non-LP	%
27966	2705	0.77%	47998	99.23%

5.2.3.2. Processing of Medical Dataset

Each node_A among 0 and 838 has been connected to another node_B among 0 and 838 to form a pair [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7), and so on]. ----------- (619,618)]. The final database will have 350,703 rows once duplicate items are removed. Because of the last step, this assures that every possibility has been investigated. To build a pair, join two nodes [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7). - --------- (619,618)] throughout the whole database that will be created (figure 5.11).

There are 93961 rows in the finished database, including the pairings [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), and (0, 7). ----------- (619,618)] The amount 0 indicates that these 347998 nodes are not linked to one another through a link or edge in the finished database that was produced in the earlier steps.

Based on the information presented above, we can deduce that the number of nodes that provided an LP is (1704), whereas the number of nodes that provided a non-link forecast is (92257). The percentage allocation to nodes can be illustrated in Table (5.3) and Figure (5.2).

Table 5.3. Nodes with LP and non-LP in the medical dataset

No. of nodes	LP	%	Non-LP	%
27966	1704	1.81%	92257	98.19%

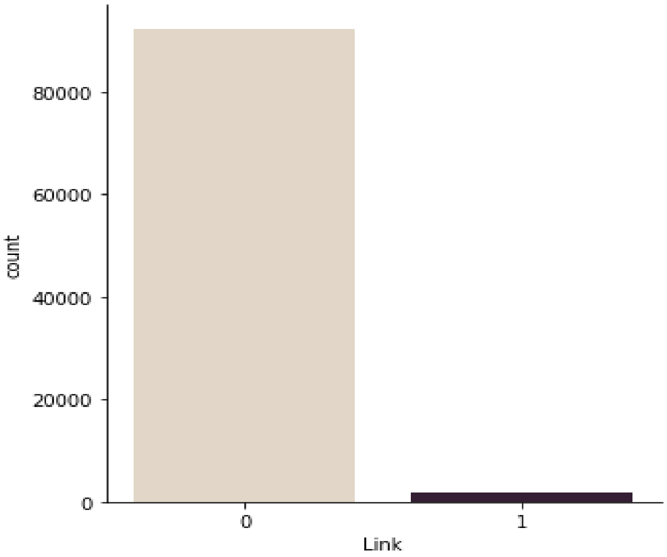


Figure 5.2. Medical dataset connection prediction diagram.

5.2.3.3. Processing For Social Dataset

The pairs [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7], etc., are formed by connecting one node_A among 0 and 838 to another node_B among 0 and 838. ----------- (619,618)]. (350,703) records make up the final database once duplicate entries are eliminated. This ensures that every option has been looked at because of the previous phase. Join two nodes [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7) to create a pair. The whole database that will be constructed will include - --------- (619,618)].

The final database has 93961 rows, which includes the pairs [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), and (0, 7). ----------- (619,618)]. These (27966) nodes are not connected by a link or edge in the final database created in the preceding phases, as indicated by the amount 0. Because all the data comprises a single amount.

Table (5.4) demonstrates that the amount of nodes that provided an LP is (441), and the amount of nodes that provided a non-link forecast is (27525) depending on the information presented previously. The percentage allocation to nodes can be illustrated in Figure (5.3).

Table 5.4. Nodes with LP and non-LP in the social dataset

No. of nodes	LP	%	Non-LP	%
27966	441	1.58%	27525	98.42%

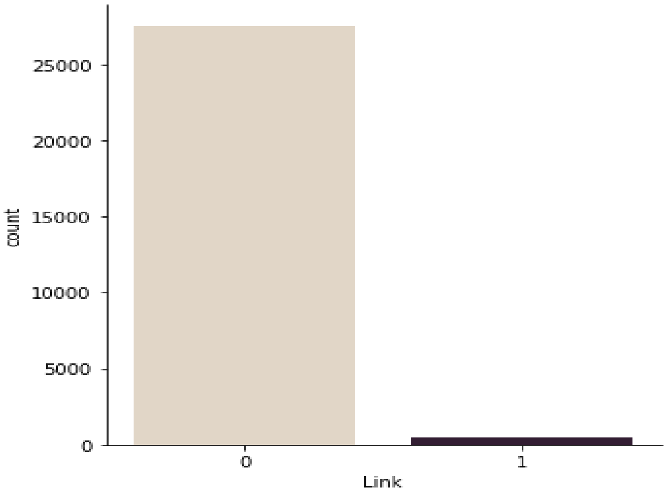


Figure 5.3. Social dataset connection prediction diagram.

5.3. IMPLEMENTATION

"The trials were carried out in the Python programming language with the assistance of PyCharm version 2019.3.3 Community edition. PyCharm is a Python Integrated Development Environment (IDE) created for experienced software programmers.

Because of the size of the graphs involved, it is necessary to carry out the calculations using techniques that have been optimized. In addition, one of the most important concerns is ensuring these procedures are successful and efficient. Creating such procedures from scratch would require significant time and effort. As a result, to reduce these expenditures, we decided to use libraries provided by other parties. We determined that the NetworkX[123] Pandas[124] Python Data Analysis Library[124] were the best options. While Pandas is a popular open-source toolkit for data analysis in Python, NetworkX[123]   is a Python package that offers complete network analysis capabilities.

The NetworkX[123] Python library, especially NetworkX[123] 2.4, is invaluable when managing complicated networks. It can read networks and create graphs for training and test sets.

The Pandas[124] Python Data Analysis Library, version 1.2.4, is useful for various data-related tasks, including analysis and the construction of data structures. It is used to store the outcomes of the procedures and write them in an Excel format.

The code file is divided up into three primary sections. The first part of the code is where the input datasets are read and where the graphs that correspond to those datasets are created. Figure (5.4) shows that this activity uses procedures found in the NetworkX library[123]. It also illustrates receiving the input and separating the graphs for the training and test sets. This is done so that the train and test graphs can be processed independently.

```
list_of_algorithms = []
list_of_algorithms.append(('Logistic                                        Regression',
LogisticRegression(max_iter=1000)))

list_of_algorithms.append(('Support Vector Classification', SVC()))

list_of_algorithms.append(('Random Forest Classifier', RandomForestClassifier()))

fig, ax = plt.subplots()

display_labels = ['No Link', 'Has a Link']

For name, algorithm in list_of_algorithms.
   newPipe = Pipeline(steps=[('preprocessor',preprocessor),
                (name,algorithm)])
   newPipe.fit(X_train, y_train)
   y_pred = newPipe.predict(X_test)
     roc_disp  =  RocCurveDisplay.from_estimator(newPipe,  X_test,  y_test,  ax=ax,
name='{}'.format(algorithm.__class__.__name__))
   print(f'Algorithm . {name}')
   print(classification_report(y_test,y_pred))
                    ConfusionMatrixDisplay.from_estimator(newPipe,            X_test,
y_test,display_labels=display_labels)
```

Figure 5.4. Analyzing inputs and displaying test-and-training data.

After the graphs have been built, any nodes not connected to other nodes are deleted to obtain the huge component. It is of the utmost importance that the train graph and the test graph have the same collection of nodes. This is because the major purpose of the experiments is to evaluate the efficacy of prediction algorithms in finding new connections among pre-existing nodes. Eliminating nodes that are exclusive to only one graph is required in order to fulfill this requirement—this phase, which brings the first section to a close.

After that, the supplied coding data visually represents building the network. In this stage, We will create a graphical depiction of the structural connections inside the network.

Establishing an experimental framework will be the main emphasis of the second stage of this procedure, which will allow the methods to be assessed. This section will discuss the implementation of the framework, focusing on evaluating the Common Neighborhood. On the other hand, it is crucial to emphasize that the other frameworks' implementation will adhere to a strategy similar to the one outlined above.

In order to ascertain the predictions that are produced by the common neighborhood approach, it is required to calculate the number of common neighbors for every prospective pair taken from the graph. Only then will it be possible to determine the predictions. In order to accomplish this goal, a framework consisting of nested loops has been put into place to create all possible pairings.

If a graph has n nodes, the total amount of potential node pairs can be determined by using a formula equivalent to selecting all subsets of the node set, provided that each subset has precisely two components. This method can compute the total amount of potential node pairs. As a result, the amount of possible pairings will be exceedingly high, given that it increases according to the order of a factorial function. As a direct consequence of this, it is not possible to generate and process all pairings concurrently. In order to overcome this obstacle, the generation of node pairs and the processing of those pairs are carried out in chunks.

For undirected networks, the number of potential combinations can be enumerated using n (n-1)/2, where n is the number of nodes. The order of the nodes will be irrelevant. Take, for instance, the links (1,3) and (3,1); they are identical.

When a chunk has been generated and filled with pairs, the next step is calculating the amount of common neighbors each pair shares inside the chunk. After that, the calculated values are contrasted with the best ones found up to this point. The candidates with the most potential are chosen to go on from the combined list of values, while the other values are thrown out. This strategy guarantees that only the candidates with the highest probability of success, i.e., the pairings It, which is going to be a part of the forecasting of the technique that is being assessed, are stored in memory. Consequently, memory use can be improved, and problems caused by insufficient memory can be alleviated. In addition, as each stage is completed, a larger proportion of pairings can be analyzed, and the final results indicate that the candidates with the highest likelihood of success depend on the total amount of graph that has been processed so far.

In order to calculate the metrics, namely Score, Shared Neighbors, Adamic Adar Index (AAI), Preferential Attachment (PA), Jaccard Coefficient (JC), and Link, for each edge, the following code will be executed. This code encompasses a set of computational instructions that systematically analyze the characteristics of the graph structure and the relevances among nodes. By traversing the graph's edges, the code applies mathematical formulas and algorithms to determine the corresponding metrics. These metrics provide insights into the graph's connectivity patterns, likeness measures, and growth tendencies. The code implementation enables the extraction of valuable information that can be utilized for various purposes, such as network analysis, recommendation systems, and community detection. Through the execution of this code, a comprehensive understanding of the graph's properties and the associations among its nodes can be achieved, facilitating further analysis and decision-making processes in scientific, social, and technological domains.

5.3.1. Model Testing

The suggested models are put through their paces by having test data input, preprocessed, and then fed into them during the testing phase. The test data are then analyzed to look for suitable features that could be used for LP. If the condition is met, the system generates an output that verifies the presence of LP using the data gathered while learning how to do something. If this is not the case, the system will give an LP assertion. There are two ways that the output model is checked. 1- Determine whether or not it corresponds to the labels in our dataset. 2- Make certain that a single data portability process minimizes lost data when applied to data collection. Figure (5.5) An explanation of the proposed model's training and testing procedure is included below.

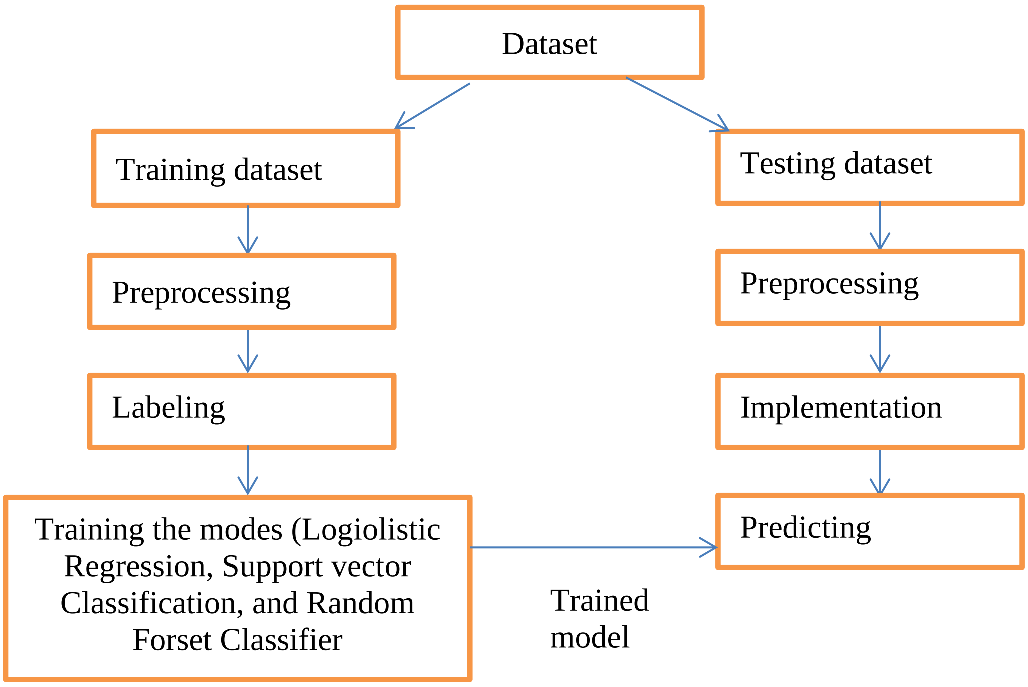


Figure 5.5. Testing-Training model Flowchart

5.4. PREDICTION

The data sets are converted into a graph representing the network during the stage responsible for preparing the input data. During the process stage known as "method execution," the prepared graph is fed into the method realization codes, which are then put through their paces. In conclusion, the evaluation metrics that were acquired before, during, and after the method execution are kept as part of the process of creating the final result report. The activities used to construct the result reports and the evaluation metrics defined in Section 3.4 are carried out during this stage. Section 3.4 Evaluation metrics are computed using the data recorded while the procedure is carried out.

Methods for forecasting, output formatting, the generation of visualizations, and output file storage are all included. Processing the graphs from real-world data sets is challenging due to the large quantities of information in these sets. The graphs can include thousands of nodes and edges connecting them. Attempting to keep track of all potential node pairings that may be linked in the next graph stage would be an enormous undertaking; we will quickly run out of memory for this endeavor. We have begun the process of producing candidate pairings in chunks of data as opposed to doing it all at once so that we can get around this problem. This enables us to free up the memory section used for a chunk after processing that chunk. Memory problems can be avoided, and the whole set of results can be acquired if the candidates are first broken up into manageable parts, and then the pieces are worked through in phases.

5.5. PLATFORM USED

The laptop utilized for this study was a Lenovo with an Intel(R) Core(TM) i5-7200CPU running at 2.5 or 2.7 GHz, a 64-bit, sixth-generation processor, 8 GB of RAM, and a Windows 10 operating system.

5.6. CLASSIFICATION METRICS

Several measures assess a model's performance in machine learning classes. Accuracy, Precision, Recall, F1-Score, and Area under the ROC Curve (AUC) are some of such measurements. Proper estimation of these parameters is essential for understanding the built model and its possible defects.

True Positive (TP) is a crucial metric that identifies phishing attacks correctly classified as positive by the model. On the other hand, True Negative (TN) metric identifies non-phishing measures that are accurately ranked as negative. False Positive (FP) is a metric that indicates non-phishing instances classified as phishing attacks, while False Negative (FN) identifies phishing attacks classified as non-phishing instances.

By utilizing these metrics, we can better comprehend the execution of an ML model and make informed decisions about potential improvements. These metrics allow researchers to make informed conclusions about the accuracy and precision of their models while identifying and addressing potential sources of errors. In conclusion, utilizing these metrics is crucial for obtaining comprehensive evaluations of ML models.

- **Accuracy** is the proportion of cases properly categorized relative to the total number of occurrences in the dataset [118],[119].

$$Accurac = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.7}$$

- **Precision** in phishing detection refers to the proportion of identified instances of phishing attacks that are accurate or have true positive results [118]. It is estimated as.

$$Precision = \frac{TP}{TP + FP} \tag{5.8}$$

- **Recall** refers to the proportion of true positive cases in a classification model that accurately identifies phishing attacks [120]; it is calculated as.

$$Recall = \frac{TP}{TP + FN} \tag{5.9}$$

- **The F1 score** is a common indicator of success in several domains, including ML and IR. This metric and recall measure accuracy in categorization or detection. Precision is the proportion of true positives among all accurate forecasts, whereas recall represents how many prognoses were right overall [121],[122]. The F1-score is a fair measure of the system's efficacy since it is the consistent mean of the accuracy and recall scores. The F1 score would be 1 in a perfect system, but in a random system, it would be 0.

$$FI - score = 2 * \frac{Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{5.10}$$

- **Receiver Operating Curve (ROC**) is a popular statistic for evaluating a model's performance when classifying data into positive and negative categories. The ROC curve illustrates the compromise among TPR and FPR at different cutoffs for classification. The entire performance of the ROC curve is summarized by the Area under the Receiver (AUC) statistic, which takes values between 0.5 and 1.0. A higher AUC suggests a superior classifier with a higher TPR and a lower FPR for each given decision criterion. In this sense, the AUC metric is a complete assessment of classifier performance since it considers every potential cutoff value.

PART 6

RESULTS AND DISCUSSION

The early results of the acquired metrics will be given in this part, both graphically and in tabular form. Methods and results from various studies will be compared to determine which ones provide the most promising performance measures and methods. In addition, a comprehensive investigation will be conducted to uncover the driving forces behind the observed accomplishment. We hope that by carefully analyzing these outcomes, we can better understand what makes the indicated measurements and methodologies so effective.

6.1. RESULTS

Because this investigation used three different datasets, the outcomes for the three suggested models, which were trained using those data sets, will be presented.

6.1.1. Computer Science Dataset Results

Visualizing the network once the computer science dataset has been applied to it is seen in Figure (6.1).

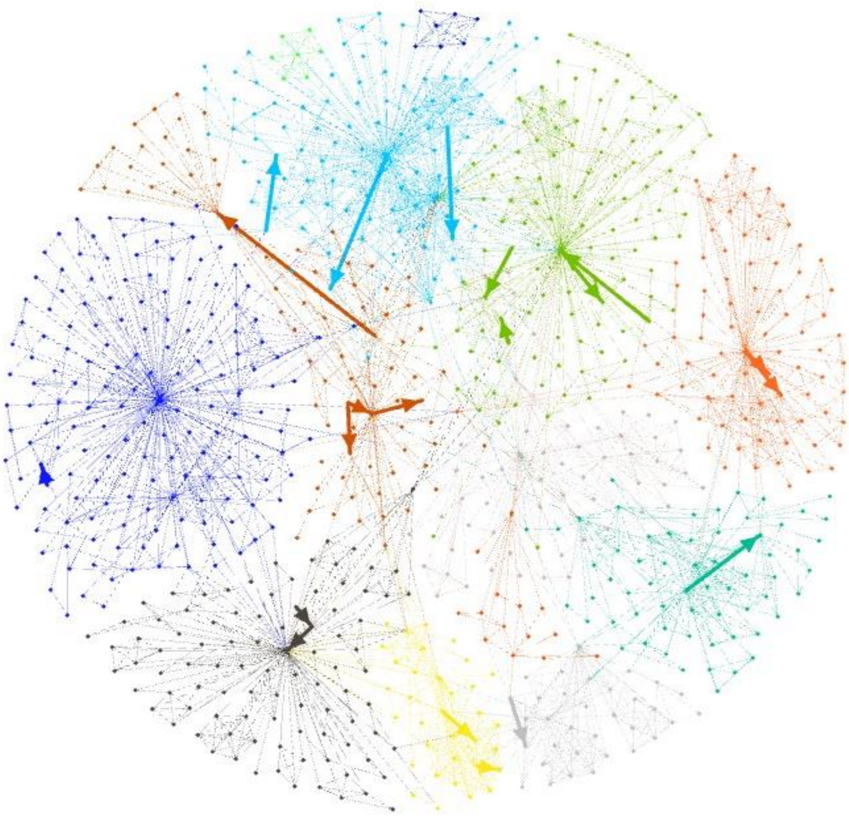
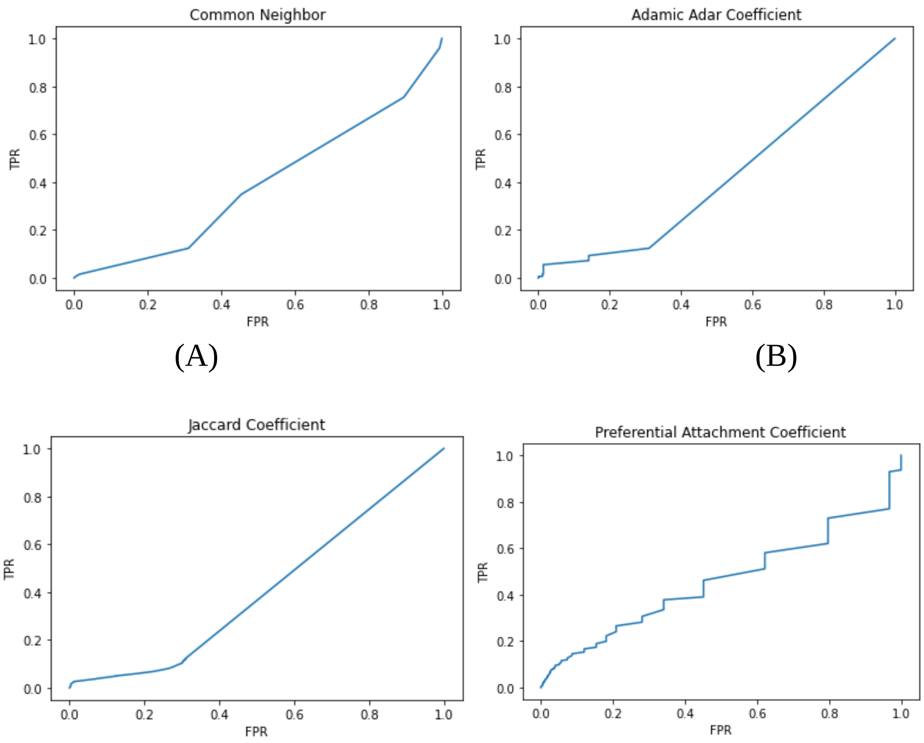


Figure 6.1. Visualizing the network for computer science dataset.

To measure and calculate the proximity of nodes on their shared neighbors. The following methods must be calculated when calculating the score for each node and edge with the computer science dataset: common neighbors, Adam's index, preferential attachment, and Jacquard's modulus. The result will be as in Figures (6.2), and Table (6.1) explains the AUC score for each method; Figure (6.3) reveals the histogram.

Table 6.1. AUC score performed using measurement methods to the computer science dataset.

Method	AUC score
Common Neighbor	0.38
Adamic Adar Coefficient	0.41
Jaccard Coefficient	0.40
Preferential Attachment Coefficient	0.46



(A) (B)

(B) (D)

Figure 6.2. Plot Representation of node affinity metrics applying to a computer science dataset. (A); Common Neighbor, (B). Adamic Adar Coefficient, (C); Jaccard Coefficient, and (D); Preferential Attachment Coefficient.

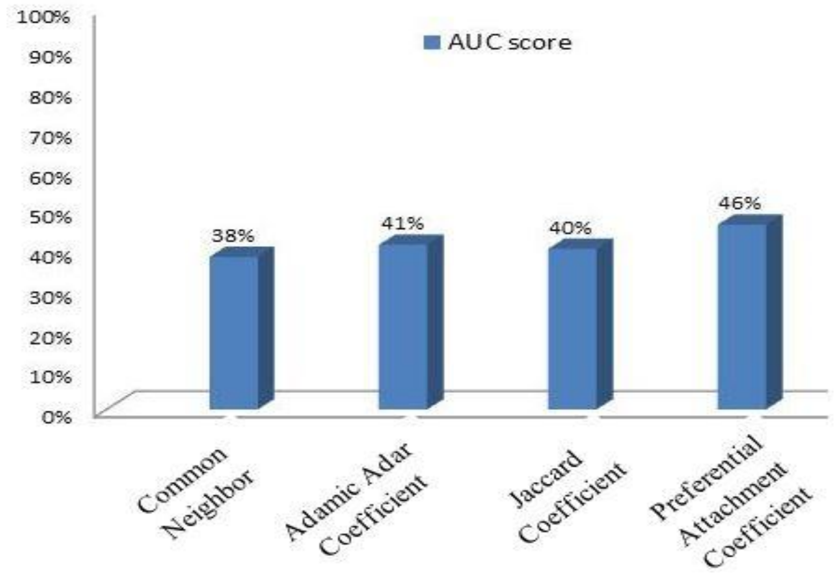


Figure 6.3. AUC score histogram for a computer science dataset.

The plot for AUC that results from applying the LR, Support Vector Classification, and RF Classifier on the computer science dataset is shown in Figure (6.4). Also, table (2) explains the results which had been got, and Figure (6.5) reveals the histogram.

Table 6.2. AUC score collected using the algorithms of the computer science dataset.

Model	**AUC score**
LR	0.99
Support Vector Classification	0.99
RF Classifier	0.99

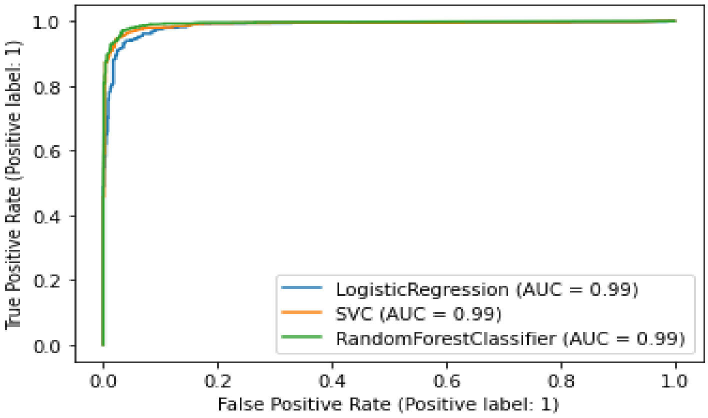


Figure 6.4. Plot for AUC score when applying the algorithms to computer sciencedataset to computer science dataset.

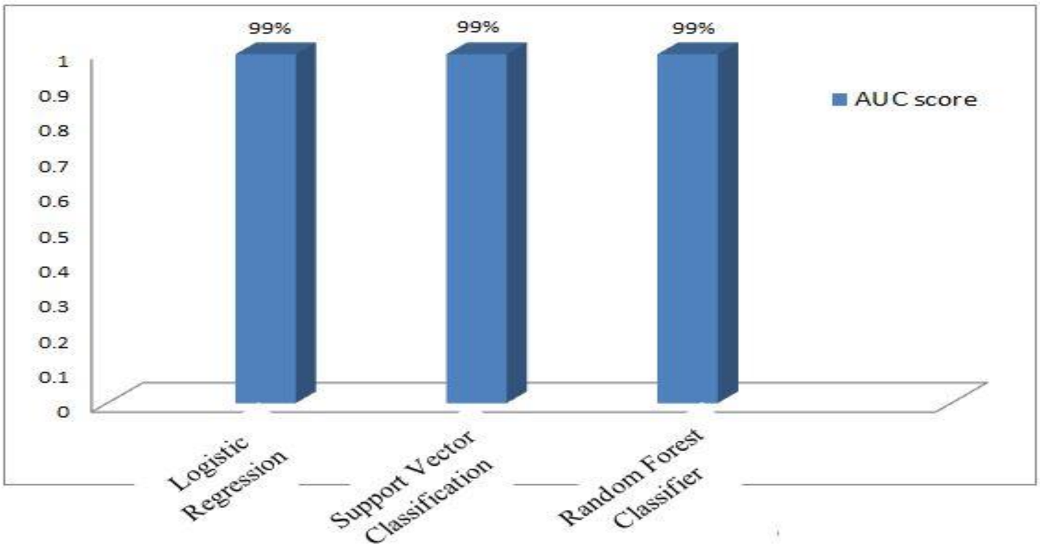


Figure 6.5. AUC score histogram for algorithms on computer science dataset.

LR was applied to the the computer science dataset; the results are shown in Table (6.3). Accuracy = 0.94, Precision = 0.91, Recall = 0.98, and F1-score = 0.94 for the non-LP. Accuracy = 0.94, Precision = 0.98, Recall = 0.91, and F1-score = 0.94 all indicate the LP shown in (1). Figure (6.6) displays confusion matrices for the proposed approaches applied to the medical dataset.

Table 6.3. Results performed using the LR model to the computer science dataset

Dataset	**Algorithm**	**Type of Link**	**Result**			
			Accuracy	Precision	Recall	F1-score
Computer science dataset	LR	Non-LP	0.94	0.91	0.98	0.94
		LP	0.94	0.98	0.91	0.94

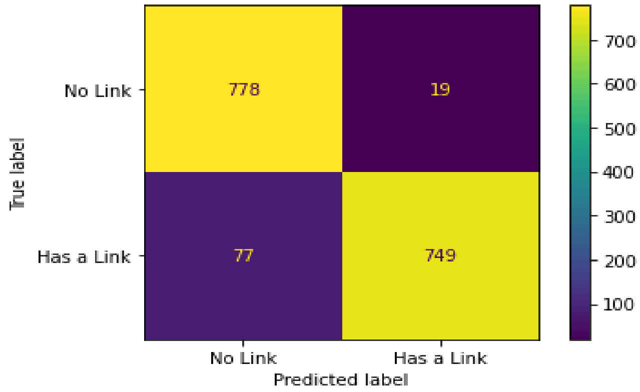


Figure 6.6. Confusion matrix using LR model to computer science dataset.

Table (6.4) shows the outcomes of applying Support Vector Classification to the computer science dataset. Accuracy = 0.94, Precision = 0.91, Recall = 0.98, and F1-score = 0.94 for the non-LP. Accuracy = 0.94, Precision = 0.98, Recall = 0.91, and F1-score = 0.94 all indicate the LP shown in (1). The application of the offered strategies is shown in Figure (6.7), along with corresponding confusion matrices.

Table 6.4. Results performed using the Support Vector Classification model to computer science dataset.

Dataset	**Algorithm**	**Type of Link**	**Result**			
			Accuracy	Precision	Recall	F1- score
Computer science dataset	Support Vector Classification	Non-LP	0.94	0.91	0.98	0.94
		LP	0.94	0.98	0.91	0.94

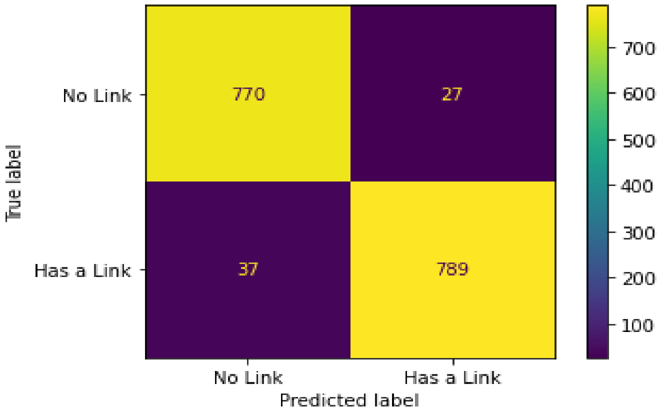


Figure 6.7. Confusion matrix using the Support Vector Classification model to computer science dataset.

Table (6.5) shows the outcomes of applying RF Classifier to the computer science dataset. Accuracy = 0.94, Precision = 0.91, Recall = 0.98, and F1-score = 0.94 for the non-LP. Accuracy = 0.94, Precision = 0.98, Recall = 0.91, and F1-score = 0.94 all indicate the LP shown in (1). The histogram is given in Figure (6.9), and the confusion matrices for the recommended procedures are shown in Figure (6.8).

Table 6.5. Results performed using the RF Classifier model to computer science dataset.

Dataset	**Algorithm**	**Type of Link**	**Result**			
			Accuracy	Precision	Recall	F1- score
Computer science dataset	RF Classifier	Non-LP	0.94	0.91	0.98	0.94
		LP	0.94	0.98	0.91	0.94

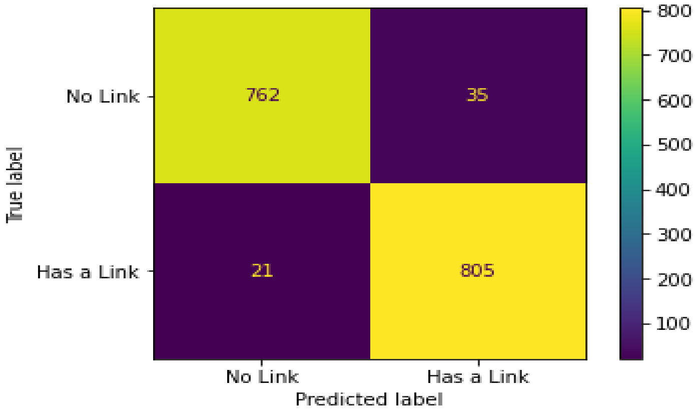


Figure 6.8. Confusion matrix using RF Classifier model to computer science dataset.

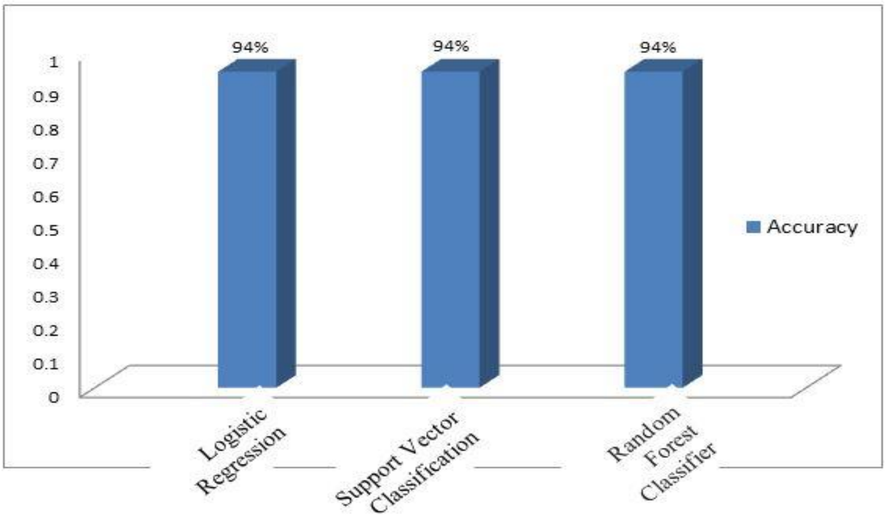


Figure 6.9. Histogram of accuracy for algorithms on computer science dataset.

6.1.2. Medical Dataset Results

Visualizing the network once the medical dataset has been applied to it is seen in Figure (6.10).

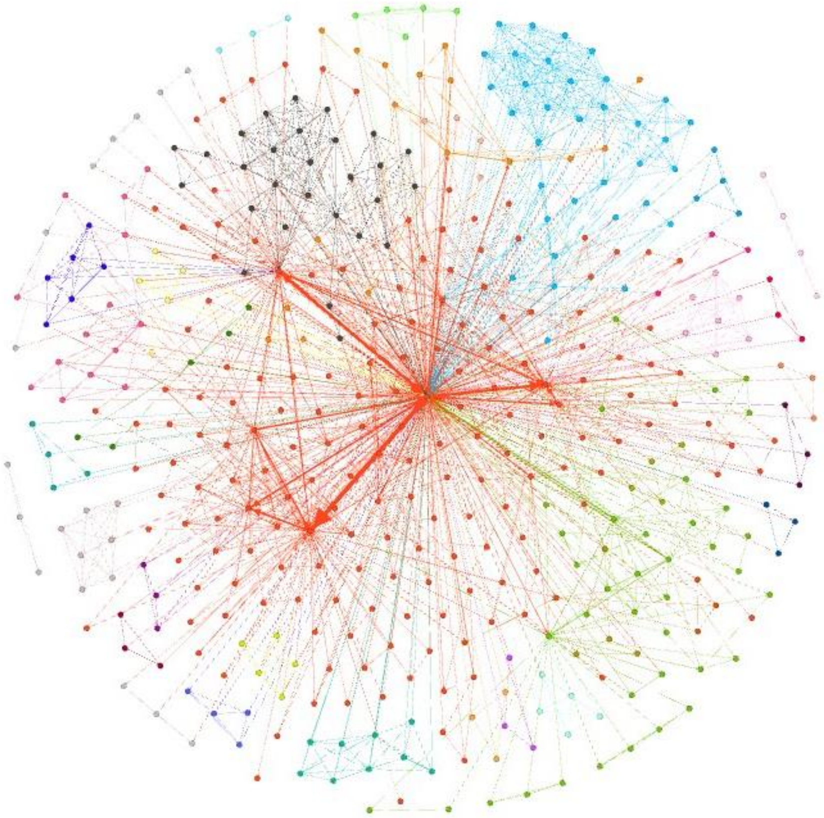
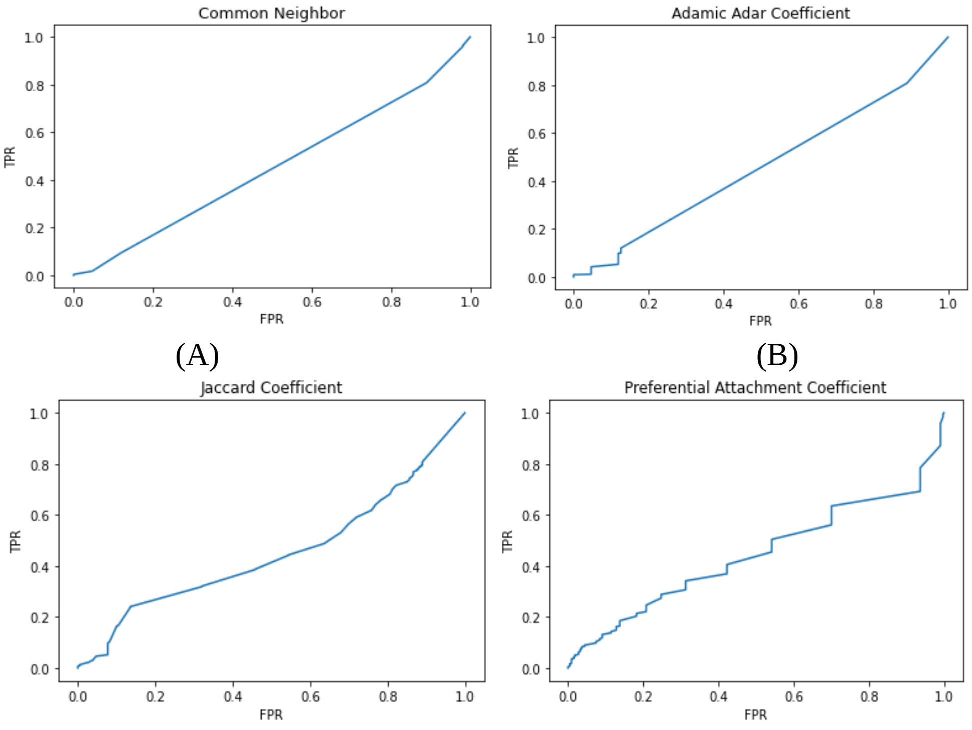


Figure 6.10. Visualizing the network for the medical dataset.

We compute the following methods when computing the score for each node and edge by medical dataset. Common neighbors, Adam's index, preferred attachment, and Jacquard's modulus. This allows us to quantify and compute the proximity of nodes depending on their shared neighbors. The outcome will be shown in Figures (6.11) and Table (6.6) to explain the AUC score for each methodology**;** **F**igure (6.12) represents the histogram.

Table 6.6. AUC score performed using measurement methods to the medical dataset.

Method	**AUC score**
Common Neighbor	0.45
Adamic Adar Coefficient	0.46
Jaccard Coefficient	0.46
Preferential Attachment Coefficient	0.44



(A)

(B)

(C)

(D)

Figure 6.11. Plot Representation of node affinity metrics applying to a medical dataset. (A); Common Neighbor, (B), Adamic Adar Coefficient, (C); Jaccard Coefficient, and (D); Preferential Attachment Coefficient.

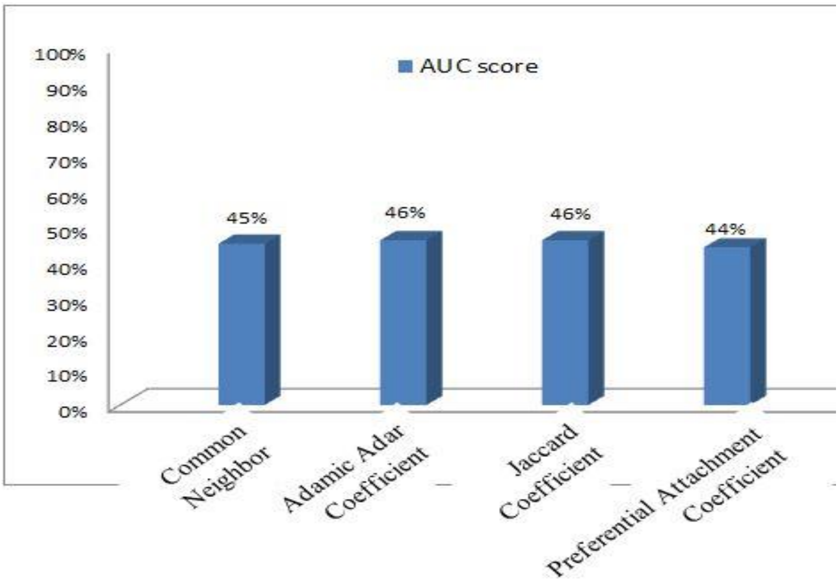


Figure 6.12. AUC score histogram for a medical dataset.

The plot for the area under the curve (AUC) that results from applying the LR, Support Vector Classification, and RF Classifier on the medical dataset is shown in Figure (6.13). Also, table (6.7) explains the results which had been got; Figure (6.14) represents the histogram.

Table 6.7. AUC score performed using the algorithms to a medical dataset

Model	**AUC score**
LR	0.99
Support Vector Classification	0.99
RF Classifier	1.00

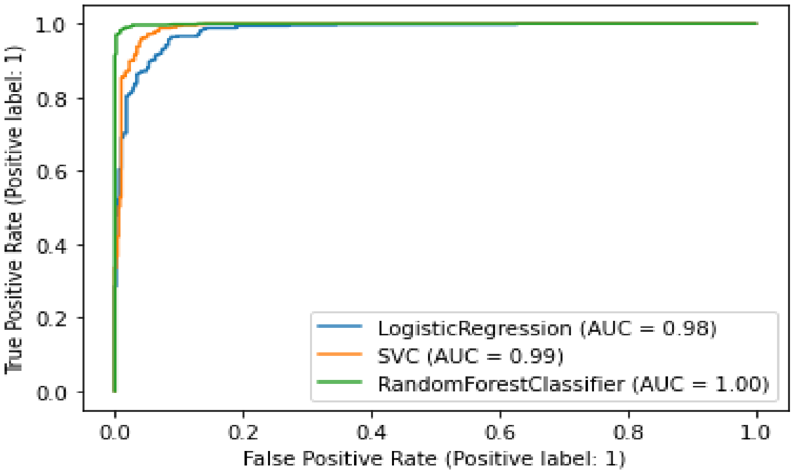


Figure 6.13. Plot for AUC score when applying the algorithms to a medical dataset.

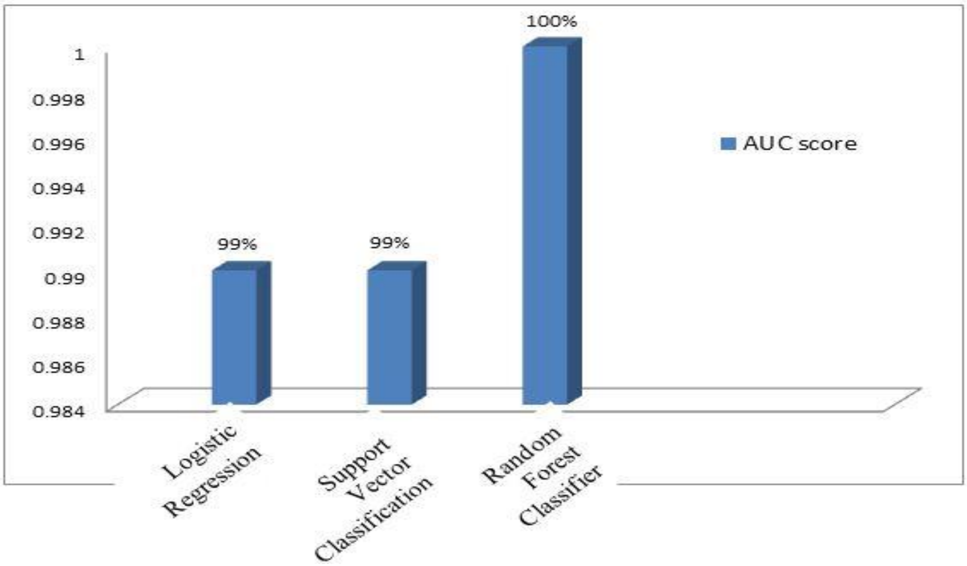


Figure 6.14. AUC score histogram for algorithms on medical dataset.

The outcomes of stratifying the LR method to the medical dataset are shown in Table (6.8). Accuracy = 0.91, Precision = 0.87, Recall = 0.95, and F1-score = 0.91 for the no-LP. The link (1) prediction has an F1-score of 0.91, a precision of 0.95, a recall of 0.87, and an accuracy of 0.91. Figure (6.15) displays confusion matrices for the proposed approaches applied to the medical dataset.

Table 6.8. Results performed by LR model with medical dataset

Dataset	**Algorithm**	**Type of Link**	**Result**			
			Accuracy	Precision	Recall	F1- score
Medical dataset	LR	Non-LP	0.91	0.87	0.95	0.91
		LP	0.91	0.95	0.87	0.91

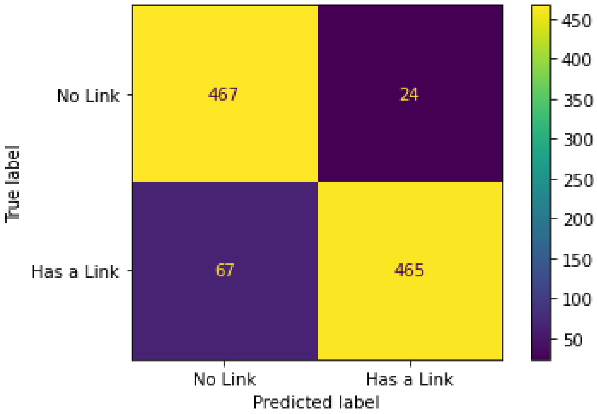


Figure 6.15. Confusion matrix using the LR to the medical dataset.

The outcomes of stratifying the Support Vector Classification method to the medical dataset are shown in Table (6.9). The non-LP with an accuracy of 0.96, precision of 0.96, recall of 0.95, and F1-score of 0.96 is represented by the amount zero. Accuracy, Precision, Recall, and F1-score were all 0.96 for the LP shown in (1). Figure (6.16) shows the confusion matrices generated by the proposed approaches applied to the medical dataset.

Table 6.9. Results performed using the Support Vector Classification model to medical dataset.

Dataset	**Algorithm**	**Type of Link**	**Result**			
			Accuracy	Precision	Recall	F1-score
Medical dataset	Support Vector Classification	Non-LP	0.96	0.96	0.95	0.96
		LP	0.96	0.95	0.97	0.96

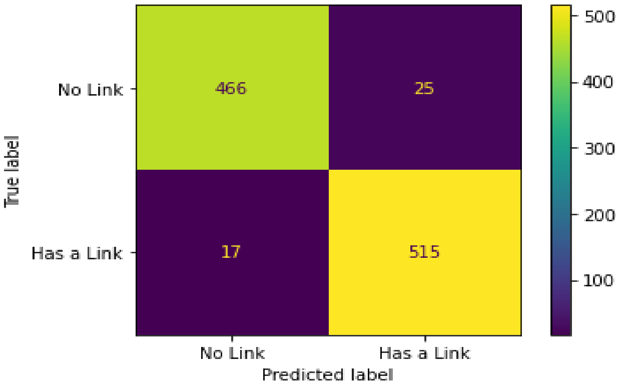


Figure 6.16. Confusion matrix using Support Vector Classification with medical dataset.

The outcomes of stratifying the RF Classifier method to the medical dataset are shown in Table (6.9). The non-LP with a score of 0 indicates perfect accuracy, precision, recall, and F1-score. The link (1) prediction has a perfect F1-score, F2-score, F1 accuracy, and F2 precision. Figure (6.17) depicts confusion matrices for the proposed approaches applied to the medical dataset, and Figure (6.18) is a histogram of the data.

Table 6.10. Results performed using the RF Classifier model to the medical dataset.

Dataset	**Algorithm**	**Type of Link**	**Result**			
			Accuracy	Precision	Recall	F1-score
Medical dataset	RF Classifier	Non-LP	0.99	0.99	0.98	0.99
		LP	0.99	0.99	0.99	0.99

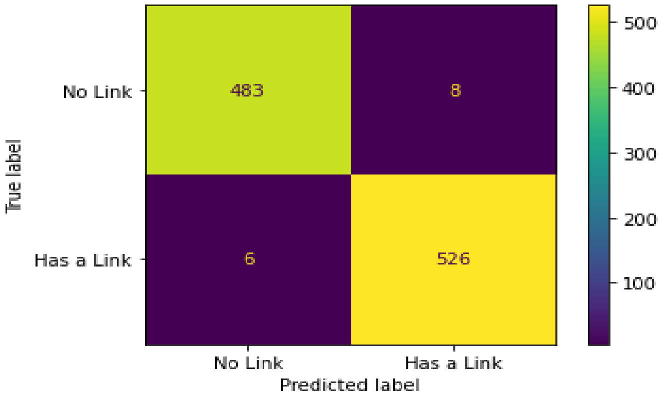


Figure 6.17. Confusion matrix using RF Classifier model to the medical dataset.

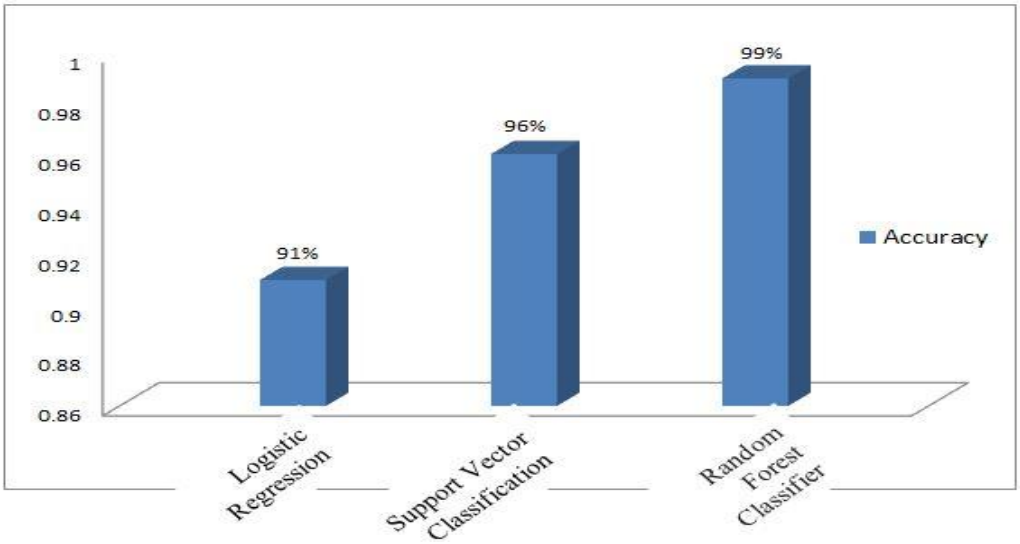


Figure 6.18. Histogram of accuracy for algorithms on medical dataset.

6.1.3. Social Dataset Results

Visualizing the network once the social dataset has been applied to it is seen in Figure (6.19).

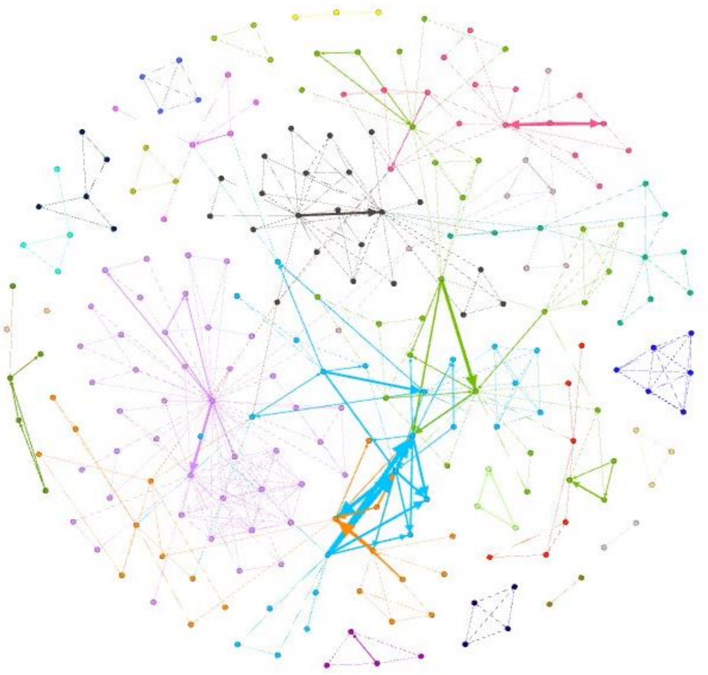
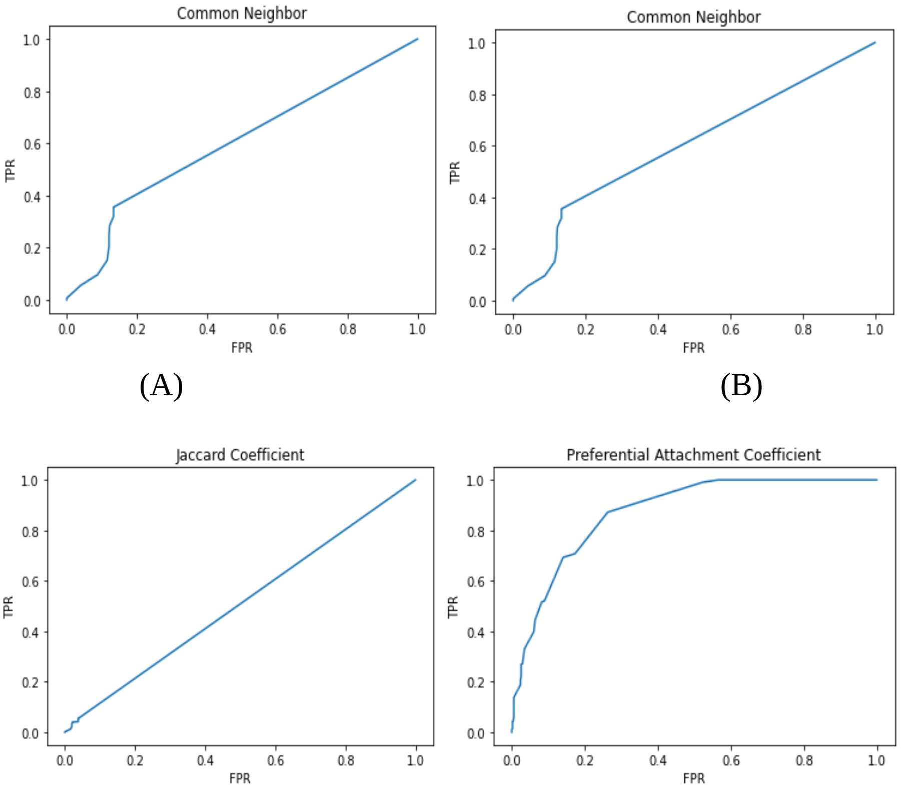


Figure 6.19. Visualizing the network for a social dataset.

We compute the following methods when computing the score for each node and edge by social dataset. Common neighbors, Adam's index, preferred attachment, and Jacquard's modulus. This allows us to quantify and compute the proximity of nodes depending on their shared neighbors. The outcome will be shown in Figure (6.20), and Table (6.11) will explain the AUC score for each methodology**; F**igure (6.21) represents the histogram.

Table 6.11. AUC score performed using measure methods to the social dataset.

Method	**AUC score**
Common Neighbor	0.60
Adamic Adar Coefficient	0.51
Jaccard Coefficient	0.51
Preferential Attachment Coefficient	0.87



(A) (B)

(C) (D)

Figure 6.20. Plot Representation of node affinity metrics applying to a social dataset . (A); Common Neighbor, (B), Adamic Adar Coefficient, (C); Jaccard Coefficient, and (D); Preferential Attachment Coefficient.

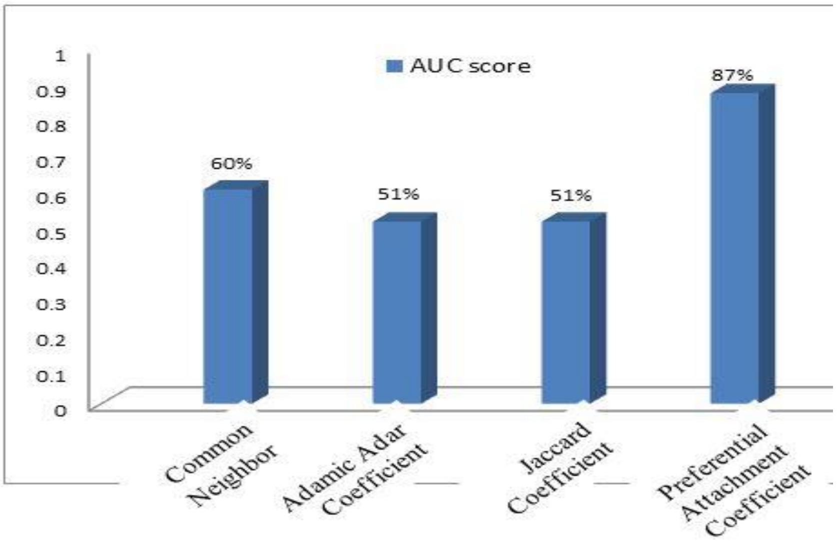


Figure 6.21. AUC score histogram for a social dataset.

The outcomes of using LR, Support Vector Classifier, and RF Classifier on the social dataset are shown in the area under the curve (AUC) plot in Figure (6.22). The results of the various categorization strategies are shown in this graphic. In addition, the obtained findings are broken down and explained in Table (6.12); Figure (6.23) represents the histogram.

Table 6.12. AUC score collected using the algorithms of the social dataset

Model	**AUC score**
LR	0.96
Support Vector Classification	0.97
RF Classifier	0.98

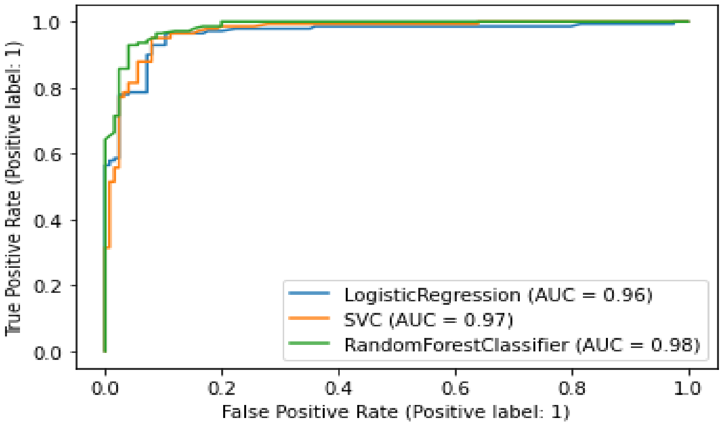


Figure 6.22. Plot for AUC score when applying the algorithms to a social dataset

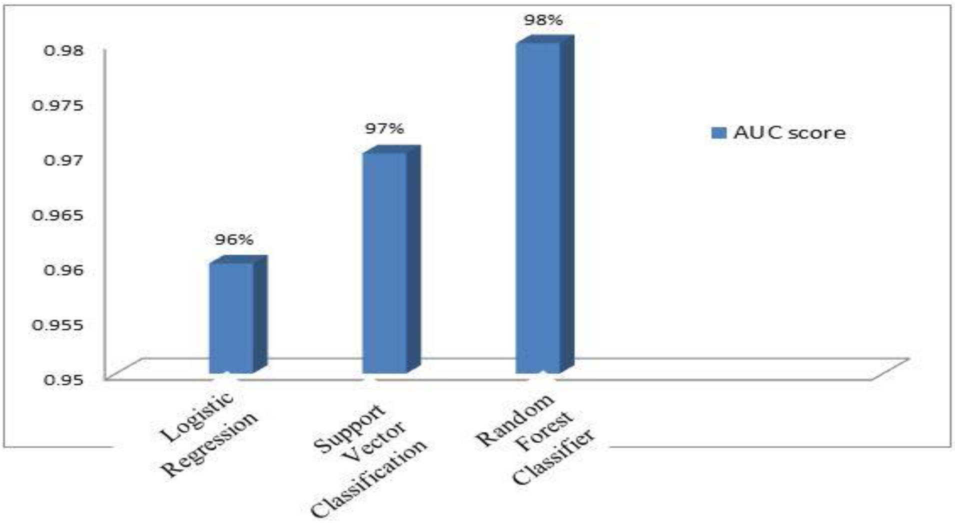


Figure 6.23. AUC score histogram for algorithms on the social dataset.

LR was stratified to the social dataset and the resulting table (6.13). Accuracy, Precision, Recall, and F1-score were all 0.87 for the case without a connection (represented by 0). Accuracy, Precision, Recall, and F1-score were all 0.87 for the LP shown in (1). Figure (6.24) displays confusion matrices for the proposed approaches applied to dataset.

Table 6.13. Results were applied to the social dataset using the LR model.

Dataset	**Algorithm**	**Type of Link**	**Result**			
			Accuracy	Precision	Recall	F1-score
Social dataset	LR	Non-LP	0.87	0.82	0.93	0.87
		LP	0.87	0.93	0.82	0.87

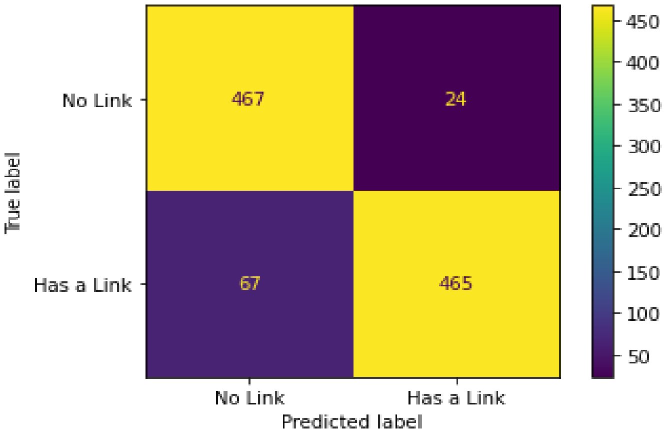


Figure 6.24. Confusion matrix using LR model to the social dataset.

The application of Support Vector Classification to the social dataset yielded the findings shown in Table (6.14). The non-LP with an accuracy of 0.93, precision of 0.93, recall of 0.92, and F1-score of 0.93 is represented by the amount zero. The link (1) prediction has a score of 0.93 for accuracy, 0.93 for precision, 0.94 for recall, and 0.94 for F1. Figure (6.25) displays confusion matrices for the proposed approaches applied to the dataset.

Table 6.14. Results performed using Support Vector Classification to social dataset.

Dataset	**Algorithm**	**Type of Link**	**Result**			
			Accuracy	Precision	Recall	F1- score
Social dataset	Support Vector Classification	Non-LP	0.93	0.93	0.92	0.93
		LP	0.93	0.93	0.94	0.94

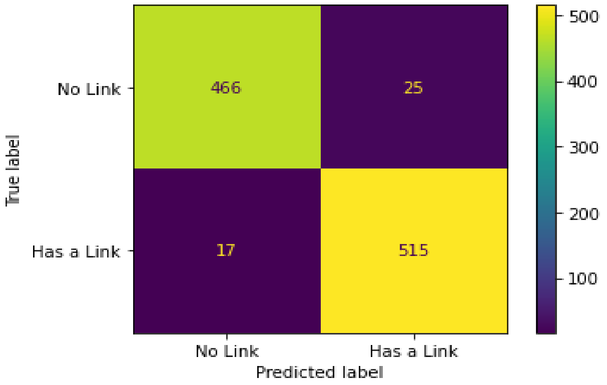


Figure 6.25. Confusion matrix for Support Vector Classification with a social dataset.

RF Classifier was applied to the social dataset and the resulting table (6.15). The non-LP with values of 0.94 for accuracy, precision, recall, and F1-score indicates a value of (0). The link (1) prediction has a precision of 0.94, recall of 0.94, accuracy of 0.94, and F1-score of 0.94. Figure (6.26) displays confusion matrices for the proposed approaches applied to the dataset, whereas Figure (6.27) depicts the histogram.

Table 6.15. Results performed using RF Classifier to a social dataset.

Dataset	**Algorithm**	**Type of Link**	**Result**			
			Accuracy	Precision	Recall	F1- score
Social dataset	RF Classifier	Non-LP	0.94	0.94	0.93	0.93
		LP	0.94	0.94	0.94	0.94

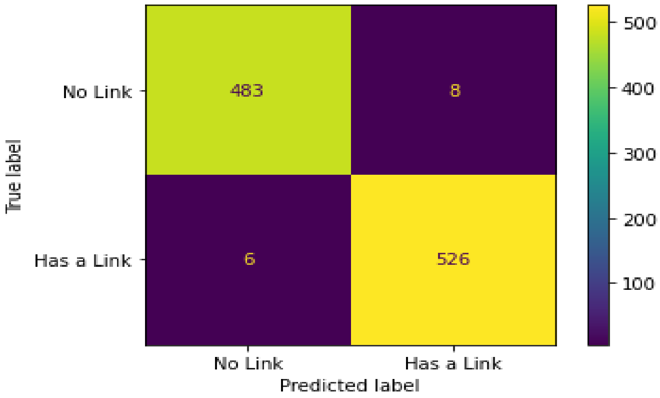


Figure 6.26. Confusion matrix using RF Classifier to a social dataset.

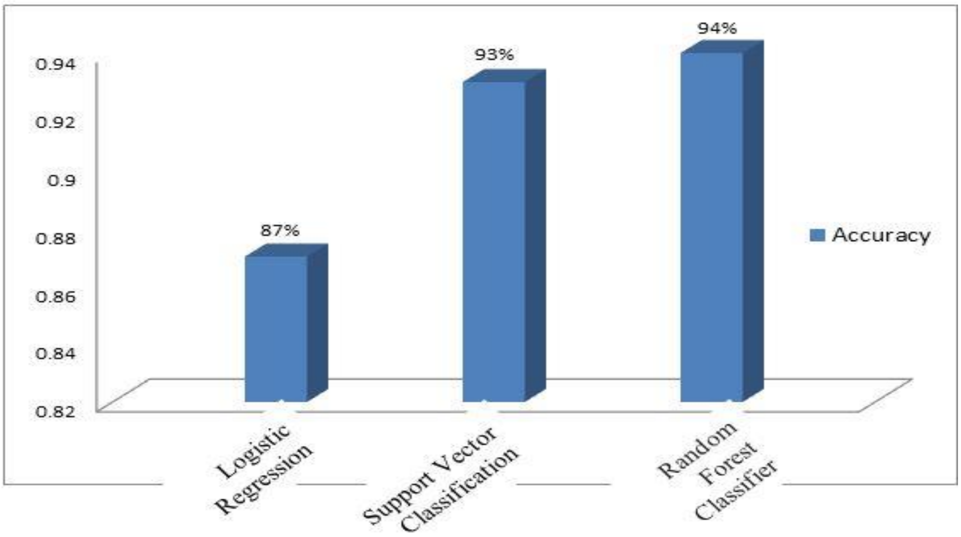


Figure 6.27. Histogram of accuracy for algorithms on social dataset.

6.2. DISCUSSION

Evaluating the AUC score offers insightful information on the effectiveness of various approaches used on the Computer science, Social, and Medical datasets. The Preferential Attachment Coefficient technique is on top compared to the other three methods investigated when applied to Computer science and Social datasets. On the other hand, when it comes to performance, the Jaccard Coefficient and Jaccard Coefficient approaches come out on top when compared to the other two ways. Both of these methods use the Jaccard Coefficient.

The social networks used for this research are non-targeting platforms; this attribute indicates that the material provided is open to large audience access instead of being tailored for particular persons.

Pairs of nodes with higher normalized values have a better likeness index when using the Preferential Attachment Coefficient approach, which indicates that they are more likely to be picked when the predictions are made. This gives rise to the hypothesis that the pairings above have a solid connection and are more likely to have features or interests in common. However, it is essential to keep in mind that a greater percentage of currently shared neighbors within an area's total population of neighbors may also be indicative of a feeling of complacency in the area. This indicates that a person has previously posted extensively about issues that they are interested in, making it less likely for them to continue posting frequently on the same themes they have already covered in depth.

On the other hand, the Adamic Adar Coefficient and the Jaccard Coefficient both award lower likeness indices to pairs of similar nodes, which indicates a decreased possibility of being picked. On the other hand, these approaches prefer pairings of nodes with lower ratios, which suggests that the common material among these pairs has not been thoroughly investigated. As a result, these strategies consider the possibility that users' interest levels in these subjects may grow throughout their stay with the platform. The Adamic Adar Coefficient and the Jaccard Coefficient techniques assume that users will continue to post within the areas where they already have expertise while also considering the likelihood that users' interests could branch into various diverse themes.

The study's findings show that user behavior matches more closely with the ideas covered by the Preferential Attachment Coefficient (PAC) approach compared to the concepts covered by other methods. This suggests that users tend to broaden the scope of their interests into other areas. On the other hand, the findings acquired from the European email core dataset make it clear that there is no substantial difference in the performance of these two approaches. This is apparent when one examines the data. This finding may be explained using the following components, which all contribute uniquely.

Compared to users of an online forum, the members of a big organization are often less dynamic in their interactions with one another than those of the forum. In addition, the carried out inside these kinds of organizations sometimes cover extended periods. As a consequence of this, it is more probable that nodes that are linked by similar edges will continue to be connected again. Consequently, we anticipate that the PAC approach will provide fruitful results in the circumstance above. However, it is essential to remember that certain projects may be completed, and employees may move on to other projects or departments after that. When this occurs, it is possible for people who previously had few to develop new relationships with one another. Consequently, it is predicted that the remaining approaches will likewise display adequate performance within the scope of this discussion.

After delving further into the study of the outcomes of the retrieval measure, one striking feature that immediately comes to mind is the much lower scores that were attained by using the Adamic Adar Coefficient (AAC) technique. In this aspect, it demonstrates a very bad performance. This finding lends credence to the hypothesis that the AAC technique does not perform at its peak efficiency level when incorrect negative predictions are included in the metric computations. It is important to point out that the Jaccard Coefficient (JC) approach has the propensity to produce a significant amount of erroneous negative predictions, one of the factors contributing to the bad performance of the AAC method.

Using the Figure (1.2) histogram, we can bring attention to the findings we acquired after applying the three suggested algorithms and training them on the three data sets. This will give us a clear perspective of the results we obtained (table 6.16)

Table 6.16. Results of execution of the algorithms to the datasets.

Algorithm	**Accuracy with specific dataset**		
	Computer science dataset	Medical dataset	Social dataset
LR	94%	91%	87%
Support Vector Classification	94%	93%	93%
RF Classifier	94%	99%	94%

The RF Classifier method outperformed previous algorithms, especially when trained on medical datasets. It displayed outstanding accuracy in both uncorrelated and correlated predictions. It stands out from other approaches because of its high rate of accuracy, which can be achieved using it. On the other hand, the accuracy produced via the training of the LR and Support Vector Classification algorithms was considerably lower.

PART 7

CONCLUSION

This research provides three training methods suggested for tackling the issue of correlation prediction on complex network. This research compares the accuracy of four popular methods for predicting relationships: The Common Neighbor, Adamic Adar, Jaccard, and Preferential Attachment Coefficient. Accuracy, recall, F1 score, precision, and AUC were used to evaluate the success of the procedures above in a battery of tests. The research used three different real-world data sets of varying sizes.

The Python-coded trials evaluated the effectiveness and efficiency of prediction algorithms. External tools like the Network X library expedite the development and guarantee quality.

Without taking into account the peculiarities of the dataset in issue, two approaches—SVC and RFC—come out on top in the analysis of performance indicators. Each of the assessed LP methods achieved similar results. Although the new LP method produced some promising findings, there is still room for improvement regarding the data's scalability. Over various dataset sizes, none of the approaches dissatisfied by acting strangely or outside the graph. The performance of the different techniques during runtime showed little to no variation.

All of the techniques evaluated in this study are categorized as topology-based approaches since they all provide predictions using node-based data. Among the many experimental approaches available for greater performance, We can use ML techniques to facilitate more powerful tactics, you can use path-based data, or you can use random traverse approach. This study may be expanded by including the methods chosen from the categories above in the trials to understand the LP strategies' performance statistics better. This addition could help practitioners comprehend things better.

For future work, we notice that to use the findings in research and development facilities, it is preferable to extend the study to incorporate bigger data sets and train additional classifiers and algorithms inside an ML method.

REFERENCES

1. Jia, M., Komeily, A., Wang, Y., and Srinivasan, R. S., "Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications", ***Automation In Construction***, 101 (July 2018): 111–126 (2019).

2. Khanna, Abhishek and Kaur, S., "Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture", ***Computers And Electronics In Agriculture***, 157: 218--231 (2019).

3. Li, S., Huang, J., Zhang, Z., Liu, J., Huang, T., and Chen, H., "Similarity-based future common neighbors model for link prediction in complex networks", ***Scientific Reports***, 8 (1): 1–11 (2018).

4. Najari, S., Salehi, M., Ranjbar, V., and Jalili, M., "Link prediction in multiplex networks based on interlayer similarity", ***Physica A: Statistical Mechanics And Its Applications***, 536: (2019).

5. Bastami, Esmaeil and Mahabadi, Aminollah and Taghizadeh, E., "A gravitation-based link prediction approach in social networks", ***Swarm And Evolutionary Computation***, 44: 176--186 (2019).

6. Khaksar Manshad, Mozhdeh and Meybodi, Mohammad Reza and Salajegheh, A., "A new irregular cellular learning automata-based evolutionary computation for time series link prediction in social networks", ***Applied Intelligence***, 51: 71--84 (2021).

7. Gu, W., Gao, F., Lou, X., and Zhang, J., "Link Prediction via Graph Attention Network", ***ArXiv Preprint ArXiv:1910.04807***, 1–12 (2019).

8. Ai, Jun and Liu, Yayun and Su, Zhan and Zhang, Hui and Zhao, F., "Link prediction in recommender systems based on multi-factor network modeling and community detection", ***Europhysics Letters***, 126: (2019).

9. Zareie, A. and Sakellariou, R., "Similarity-based link prediction in social networks using latent relationships between the users", ***Scientific Reports***, 10 (1): 1–11 (2020).

10. Liu, JiaHui and Jin, Xu and Hong, YuXiang and Liu, Fan and Chen, QiXiang and Huang, YaLou and Liu, MingMing and Xie, MaoQiang and Sun, F., "Collaborative linear manifold learning for link prediction in heterogeneous networks", ***Information Sciences***, 511: 297--308 (2020).

11. Rafiee, Samira and Salavati, Chiman and Abdollahpouri, A., "CNDP: Link prediction based on common neighbors degree penalization", ***Physica A: Statistical Mechanics And Its Applications***, 539: (2020).

12. Guo, Feipeng and Zhou, Wei and Wang, Zifan and Ju, Chunhua and Ji, Shaobo and Lu, Q., "A link prediction method based on topological nearest-neighbors similarity in directed networks", ***Journal Of Computational Science***, 102002 (2023).

13. Zhang, Lin and Lu, Jian and Yue, Xianfei and Zhou, Jialin and Li, Yunxuan and Wan, Q., "An auxiliary optimization method for complex public transit route network based on link prediction", ***Modern Physics Letters B***, 32: (2018).

14. Berahmand, Kamal and Nasiri, Elahe and Rostami, Mehrdad and Forouzandeh, S., "A modified DeepWalk method for link prediction in attributed social network", ***Computing***, 103: 2227--2249 (2021).

15. Kumar, Mukesh and Mishra, Shivansh and Biswas, B., "Features fusion based link prediction in dynamic neworks", ***Journal Of Computational Science***, 57: (2022).

16. Li, Wenjun and Li, Ting and Berahmand, K., "An effective link prediction method in multiplex social networks using local random walk towards dependable pathways", ***Journal Of Combinatorial Optimization***, 45: 31 (2023).

17. Yang, Min and Liu, Junhao and Chen, Lei and Zhao, Zhou and Chen, Xiaojun and Shen, Y., "An advanced deep generative framework for temporal link prediction in dynamic networks", ***IEEE Transactions On Cybernetics***, 50: 4946--4957 (2019).

18. Cho, Joon Hyung and Lee, Jungpyo and Sohn, S. Y., "Predicting future technological convergence patterns based on machine learning using link prediction", ***Scientometrics***, 126: 5413--5429 (2021).

19. Gu, Weiwei and Gao, Fei and Li, Ruiqi and Zhang, J., "Learning universal network representation via link prediction by graph convolutional neural network", ***Journal Of Social Computing***, 2: 43--51 (2021).

20. Keikha, Mohammad Mehdi and Rahgozar, Maseud and Asadpour, M., "No TitleDeepLink: A novel link prediction framework based on deep learning", ***Journal Of Information Science***, 47: 642--657 (2021).

21. Anand, Sameer and Mallik, Abhishek and Kumar, S., "Integrating node centralities, similarity measures, and machine learning classifiers for link prediction", ***Multimedia Tools And Applications***, 81: 38593--38621 (2022).

22. Kerrache, S. and Benhidour, H., "A Complex Network based Graph Embedding Method for Link Prediction", ***ArXiv Preprint ArXiv:2209.04884***, 1–20 (2022).

23. Khoshraftar, S. and An, A., "A Survey on Graph Representation Learning Methods", ***ACM Transactions On Intelligent Systems And Technology***, .

24. Ning, N., Li, Q., Zhao, K., and Wu, B., "Multiplex Network Embedding Model with High-Order Node Dependence", ***Complexity***, 2021: (2021).

25. Chu, S., Dao, T., Pan, J., and Nguyen, T., "Identifying correctness data scheme for aggregating data in cluster heads of wireless sensor network based on naive Bayes classification", (2020).

26. Afra, Salim and Aksacc, Alper and Ozyer, Tansel and Alhajj, R., "Link prediction by network analysis", ***Prediction And Inference From Social Networks And Social Media***, 97--114 (2017).

27. vZerovnik, Janez and Poklukar, D. R., "Advances in Discrete Applied Mathematics and Graph Theory", Advances in Discrete Applied Mathematics and Graph Theory, ***MDPI-Multidisciplinary Digital Publishing Institute***, (2022).

28. Fuchs, C., Spolaor, S., Nobile, M. S., and Kaymak, U., "A Graph Theory Approach to Fuzzy Rule Base Simplification", Communications in Computer and Information Science, ***Springer International Publishing***, 387–401 (2020).

29. Wang, H., "Network Graph Theory and Organization Model Analysis based on Mathematical Modeling with the Dynamic Systematic Data Perspective", ***2022 6th International Conference On Trends In Electronics And Informatics (ICOEI)***, 915--919 (2022).

30. Goutham, Akula and Reddy, Burri Gowtham and Kosuri, Rishitha and Praneeth, Parsharouthu and Lavanya, R., "A Novel Approach to Segment OCT layers using Graph method for Glaucoma Diagnosis", ***2022 6th International Conference On Trends In Electronics And Informatics (ICOEI)***, 1588--1590 (2022).

31. Zhou, Z., Shojafar, M., and Member, S., "EVCT : An efficient VM deployment algorithm for a software-defined data center in a connected and", ***IEEE Transactions On Green Communications And Networking***, 1–11 (2022).

32. Silver, D., "The Influence of Simmel on American Sociology Since 1975", ***Annual Review Of Sociology***, 87--108 (2021).

33. Mutzel, Sophie and Kressin, L., "From simmel to relational sociology", ***Handbook Of Classical Sociological Theory***, 217--238 (2021).

34. Rabinovich, M. I., Zaks, M. A., and Varona, P., "Sequential dynamics of complex networks in mind : consciousness and creativity", ***Physics Reports***, 1–39 (2020).

35. Leeuw, V. C. De, Oostrom, C. T. M. Van, Zwart, E. P., Heusinkveld, H. J., and Hessel, E. V. S., "Prolonged Differentiation of Neuron-Astrocyte Co-Cultures Results in Emergence of Dopaminergic Neurons", ***International Journal Of Molecular Sciences***, 3608 (2023).

36. Wu, Y., Ding, Q., Li, T., Yu, D., and Jia, Y., "Effect of temperature on synchronization of scale-free neuronal network Effect of temperature on synchronization of scale-free neuronal network", ***Nonlinear Dynamics***, (October): (2022).

37. Walle, R. Van De, Logghe, G., Haas, N., and Massol, F., "Arthropod food webs predicted from body size ratios are improved by incorporating prey defensive properties", ***Journal Of Animal Ecology***, (February): (2023).

38. Lu, Qi and Cheng, Chen and Xiao, Lingyun and Li, Juan and Li, Xueyang and Zhao, Xiang and Lu, Zhi and Zhao, Jindong and Yao, M., "Food webs reveal coexistence mechanisms and community organization in carnivores", ***Current Biology***, (2023).

39. Stergiopoulos, George and Dedousis, Panagiotis and Gritzalis, D., "Automatic analysis of attack graphs for risk mitigation and prioritization on large-scale and complex networks in Industry 4.0", ***International Journal Of Information Security***, 1--23 (2022).

40. Gosak, M. and Milojevi, M., "Networks behind the morphology and structural design of living systems", ***Physics Of Life Reviews***, 41: 1–21 (2022).

41. Ashok, Sachin and Godfrey, P Brighten and Mittal, R., "Leveraging service meshes as a new network layer", ***Proceedings Of The Twentieth ACM Workshop On Hot Topics In Networks***, 229--236 (2021).

42. Wang, Z. and Ward, T., "G ENERATIVE ADVERSARIAL NETWORKS IN TIME SERIES : A", ***ACM Computing Surveys***, (2021).

43. Singh, T., Roberts, K., Cohen, T., Cobb, N., Franklin, A., and Myneni, S., "Discerning conversational context in online health communities for personalized digital behavior change solutions using Pragmatics to Reveal Intent in Social Media (PRISM) framework", ***Journal Of Biomedical Informatics***, 140 (February): 104324 (2023).

44. Wernecke, Christian and Parzyjegla, Helge and Muhl, Gero and Danielis, Peter and Schweissguth, Eike and Timmermann, D., "Evaluating P4-based Virtual Delivery Trees for Content-based Publish/Subscribe", ***2022 IEEE Conference On Network Function Virtualization And Software Defined Networks (NFV-SDN)***, 78--84 (2022).

45. Sporns, O., "Structure and function of complex brain networks", (2013): (2022).

46. Nagler, J. and Arenas, A., "Explosive phenomena in complex networks", ***Advances In Physics***, 123--223 (2019).

47. Mata, S., "Complex Networks : a Mini-review", ***Brazilian Journal Of Physics***, 658–672 (2020).

48. Poor, H. V., "The effects of evolutionary adaptations on spreading processes in complex networks", ***Proceedings Of The National Academy Of Sciences***, 1–7 (2020).

49. Anwar, R., Yousuf, M. I., and Abid, M., "Analysis of a Model for Generating Weakly Scale-free Networks", ***Discrete Mathematics \& Theoretical Computer Science***, 22 (2018): (2019).

50. Osada, T., Coutinho, B., Omar, Y., Sanaka, K., Munro, W. J., and Nemoto, K., "Continuous-time quantum-walk spatial search on the Bollobás scale-free network", ***Physical Review A***, 101 (2): (2020).

51. Iacopini, I., Petri, G., Barrat, A., and Latora, V., "Simplicial models of social contagion", ***Nature Communications***, 10 (1): 1–9 (2019).

52. Robledo, O. F., Zhan, X. X., Hanjalic, A., and Wang, H., "Influence of clustering coefficient on network embedding in link prediction", ***Applied Network Science***, 7 (1): 1–20 (2022).

53. Trolliet, T., Cohen, N., Giroire, F., Hogie, L., and Pérennes, S., "Interest Clustering Coefficient: A New Metric for Directed Networks Like Twitter", ***Studies In Computational Intelligence***, 944: 597–609 (2021).

54. Urbani, J. and Jacobs, C., "Adaptive Low-level Storage of Very Large Knowledge Graphs", ***The Web Conference 2020 - Proceedings Of The World Wide Web Conference, WWW 2020***, 2: 1761–1772 (2020).

55. Petrolo, R., "Semantic-based discovery and integration of heterogeneous things in a Smart City environment To cite this version : HAL Id : tel-01403844 Semantic-based discovery and integration of heterogeneous things in a Smart City environment", ***University Of Lille 1***, (2016).

56. Krivonosov, M., Vershinina, O., Pirova, A., Gupta, S., Kanakov, O., and Kurths, J., "Comparative statistical study of two local clustering coefficient formulations as tropical cyclone markers for climate networks", ***ArXiv Preprint ArXiv:2212.13934***, 1–15 (2022).

57. Trinh, N. H., Ban, D. Van, Quang, V. V., and Tung, C. T., "A Fast Overlapping Community Detection Algorithm Based on Label Propagation and Social Network Graph Clustering Coefficient", ***Journal Of Computer Science And Cybernetics***, 38 (1): 31–46 (2022).

58. Abboud, A., Grandoni, F., and Williams, V. V., "Subcubic equivalences between graph centrality problems, APSP and diameter", ***Proceedings Of The Annual ACM-SIAM Symposium On Discrete Algorithms***, 2015-Janua (January): 1681–1697 (2015).

59. Dwivedi, V. P., Rampášek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., and Beaini, D., "Long Range Graph Benchmark", ***Advances In Neural Information Processing Systems***, (NeurIPS): 1–15 (2022).

60. Kogan, S. and Parter, M., "New Diameter-Reducing Shortcuts and Directed Hopsets: Breaking the O(P n) Barrier", ***Proceedings Of The Annual ACM-SIAM Symposium On Discrete Algorithms***, 2022-Janua: 1326–1341 (2022).

61. Bringmann, K., Kisfaludi-Bak, S., Künnemann, M., Nusser, A., and Parsaeian, Z., "Towards Sub-Quadratic Diameter Computation in Geometric Intersection Graphs", ***Leibniz International Proceedings In Informatics, LIPIcs***, 224 (850979): (2022).

62. Ferrari, G., Savic, D., and Becciu, G., "Graph-Theoretic Approach and Sound Engineering Principles for Design of District Metered Areas", ***Journal Of Water Resources Planning And Management***, 140 (12): (2014).

63. Li, H., Liu, X., Li, T., and Gan, R., "A novel density-based clustering algorithm using nearest neighbor graph", ***Pattern Recognition***, 102: 107206 (2020).

64. Malliaros, F. D., Giatsidis, C., Papadopoulos, A. N., and Vazirgiannis, M., "The core decomposition of networks: theory, algorithms and applications", ***VLDB Journal***, 29 (1): 61–92 (2020).

65. Dao, V. L., Bothorel, C., and Lenca, P., "Estimating the similarity of community detection methods based on cluster size distribution", ***Studies In Computational Intelligence***, 812: 183–194 (2019).

66. Riolo, M. A. and Newman, M. E. J., "Consistency of community structure in complex networks", ***Physical Review E***, 101 (5): 1–10 (2020).

67. Ghalmane, Zakariya and Cherifi, Chantal and Cherifi, Hocine and El Hassouni, M., "Exploring hubs and overlapping nodes interactions in modular complex networks", ***IEEE Access***, 8 (79650--79683): (2020).

68. Zhu, Z., Zhang, Z., Xhonneux, L., and Tang, J., "Neural Bellman-Ford Networks : A General Graph Neural Network Framework for Link Prediction", ***Advances In Neural Information Processing Systems***, (NeurIPS): 1–15 (2021).

69. Daud, Nur Nasuha and Ab Hamid, Siti Hafizah and Saadoon, Muntadher and Sahran, Firdaus and Anuar, N. B., "Applications of link prediction in social networks: A review", ***Journal Of Network And Computer Applications***, 166: (2020).

70. Mahapatra, Rupkumar and Samanta, Sovan and Pal, Madhumangal and Xin, Q., "RSM index: a new way of link prediction in social networks", ***Journal Of Intelligent \& Fuzzy Systems***, 37: (2019).

71. Zeng, S., "Link prediction based on local information considering preferential attachment", ***Physica A***, 443: 537–542 (2016).

72. Raut, Purva and Khandelwal, Harshita and Vyas, G., "A comparative study of classification algorithms for link prediction", ***2020 2nd International Conference On Innovative Mechanisms For Industry Applications (ICIMIA)***, 479--483 (2020).

73. Stern, M., Hexner, D., Rocks, J. W., and Liu, A. J., "Supervised Learning in Physical Networks: From Machine Learning to Learning Machines", ***Physical Review X***, 11 (2): 21045 (2021).

74. Bai, H., Cao, M., Huang, P., and Shan, J., "Self-supervised Semi-supervised Learning for Data Labeling and Quality Evaluation", ***ArXiv Preprint ArXiv:2111.10932***, (2021).

75. Hopkins, E. and others, "Machine learning tools, algorithms, and techniques", ***Journal Of Self-Governance And Management Economics***, 10: 43--55 (2022).

76. Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., and Xu, M., "A Survey on Machine Learning Techniques for Cyber Security in the Last Decade", ***IEEE Access***, 8: 222310–222354 (2020).

77. Convolutional, Y., Abbod, M. F., and Shieh, J., "Defect Detection in Printed Circuit Boards Using Neural Networks", ***Electronics***, 9 (1547): 1–16 (2020).

78. Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F., "Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What’s Next", ***Journal Of Scientific Computing***, 92 (3): 1–62 (2022).

79. Galaris, E., Fabiani, G., Gallos, I., Kevrekidis, I., and Siettos, C., "Numerical Bifurcation Analysis of PDEs From Lattice Boltzmann Model Simulations: a Parsimonious Machine Learning Approach", ***Journal Of Scientific Computing***, 92 (2): 1–30 (2022).

80. Liew, B. X. W., Kovacs, F. M., Rügamer, D., and Royuela, A., "Machine learning versus logistic regression for prognostic modelling in individuals with non-specific neck pain", ***European Spine Journal***, 31 (8): 2082–2091 (2022).

81. Nusinovici, Simon and Tham, Yih Chung and Yan, Marco Yu Chak and Ting, Daniel Shu Wei and Li, Jialiang and Sabanayagam, Charumathi and Wong, Tien Yin and Cheng, C.-Y., "Logistic regression was as good as machine learning for predicting major chronic diseases", ***Journal Of Clinical Epidemiology***, 122: 56--69 (2020).

82. Ye, Y. Z., Xiong, Y., Zhou, Q. J., Wu, J. N., Li, X. T., and Xiao, X. R., "Comparison of Machine Learning Methods and Conventional Logistic Regressions for Predicting Gestational Diabetes Using Routine Clinical Data: A Retrospective Cohort Study", ***Journal Of Diabetes Research***, 2020: (2020).

83. Daghistani, T. and Alshammari, R., "Comparison of statistical logistic regression and randomforest machine learning techniques in predicting diabetes", ***Journal Of Advances In Information Technology***, 11 (2): 78–83 (2020).

84. Pua, Yong-Hao and Kang, Hakmook and Thumboo, Julian and Clark, Ross Allan and Chew, Eleanor Shu-Xian and Poon, Cheryl Lian-Li and Chong, Hwei-Chi and Yeo, S.-J., "Machine learning methods are comparable to logistic regression techniques in predicting severe walking limitation following total

knee arthroplasty", ***Knee Surgery, Sports Traumatology, Arthroscopy***, 20: 3207--3216 (2020).

85. Kerrache, S., "LinkPred: a high performance library for link prediction in complex networks", ***PeerJ Computer Science***, 7: e521 (2021).

86. de Bruin, G. J., Veenman, C. J., van den Herik, H. J., and Takes, F. W., "Supervised temporal link prediction in large-scale real-world networks", ***Social Network Analysis And Mining***, 11 (1): 1–16 (2021).

87. King, I. J. and Huang, H. H., "Euler: Detecting Network Lateral Movement via Scalable Temporal Graph Link Prediction", ***ACM Transactions On Privacy And Security***, (2022).

88. Waoo, A. A. and Soni, B. K., "Performance analysis of sigmoid and relu activation functions in deep neural network", (2021).

89. Yuan, W., Han, Y., Guan, D., Han, G., Tian, Y., Al-Dhelaan, A., and Al-Dhelaan, M., "Weighted enclosing subgraph-based link prediction for complex network", ***Eurasip Journal On Wireless Communications And Networking***, 2022 (1): 1–14 (2022).

90. Chhajer, P., Shah, M., and Kshirsagar, A., "The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction", ***Decision Analytics Journal***, 2 (June 2021): 100015 (2022).

91. Wu, H., Song, C., Ge, Y., and Ge, T., "Link Prediction on Complex Networks : An Experimental Survey", ***Data Science And Engineering***, 7 (3): 253–278 (2022).

92. Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., and Lopez, A., "A comprehensive survey on support vector machine classification: Applications, challenges and trends", ***Neurocomputing***, 408: 189–215 (2020).

93. Liu, C., Gu, Z., and Wang, J., "A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning", ***IEEE Access***, 9: 75729–75740 (2021).

94. Sipper, M. and Moore, J. H., "Conservation machine learning: a case study of random forests", ***Scientific Reports***, 11 (1): 1–6 (2021).

95. Salam, M. A., Azar, A. T., Elgendy, M. S., and Fouad, K. M., "The Effect of Different Dimensionality Reduction Techniques on Machine Learning Overfitting Problem", ***International Journal Of Advanced Computer Science And Applications***, 12 (4): 641–655 (2021).

96. Abdulkareem, Nasiba Mahdi and Abdulazeez, A. M. and others, "Machine learning classification based on Radom Forest Algorithm: A review", ***International Journal Of Science And Business***, 5: 128--142 (2021).

97. Huynh-Cam, T. T., Chen, L. S., and Le, H., "Using decision trees and random forest algorithms to predict and determine factors contributing to first-year university students' learning performance", ***Algorithms***, 14 (11): (2021).

98. Jalal, N., Mehmood, A., Choi, G. S., and Ashraf, I., "A novel improved random forest for text classification using feature ranking and optimal number of trees", ***Journal Of King Saud University - Computer And Information Sciences***, 34 (6): 2733–2742 (2022).

99. Liu, K., Hu, X., Zhou, H., Tong, L., Widanage, W. D., and Marco, J., "Feature Analyses and Modeling of Lithium-Ion Battery Manufacturing Based on Random Forest Classification", ***IEEE/ASME Transactions On Mechatronics***, 26 (6): 2944–2955 (2021).

100. Cunningham, P. and Delany, S. J., "K-Nearest Neighbour Classifiers-A Tutorial", ***ACM Computing Surveys***, 54 (6): (2021).

101. Chen, R. C., Dewi, C., Huang, S. W., and Caraka, R. E., "Selecting critical features for data classification based on machine learning methods", ***Journal Of Big Data***, 7 (1): (2020).

102. Ali, N., Neagu, D., and Trundle, P., "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets", ***SN Applied Sciences***, 1 (12): 1–15 (2019).

103. Deng, Xuelian and Li, Yuqing and Weng, Jian and Zhang, J., "Feature selection for text classification: A review", ***Multimedia Tools And Applications***, 78: 3797--3816 (2019).

104. Aleta, A., Tuninetti, M., Paolotti, D., Moreno, Y., and Starnini, M., "Link prediction in multiplex networks via triadic closure", ***Physical Review Research***, 2 (4): 42029 (2020).

105. Aziz, Furqan and Gul, Haji and Muhammad, Ishtiaq and Uddin, I., "Link prediction using node information on local paths", ***Physica A: Statistical Mechanics And Its Applications***, 557: 124980 (2020).

106. Costa, L. da F., "Further Generalizations of the Jaccard Index", ***ArXiv Preprint ArXiv:2110.09619***, (2021).

107. Ayub, Mubbashir and Ghazanfar, Mustansar Ali and Khan, Tasawer and Saleem, A., "An effective model for Jaccard coefficient to increase the performance of collaborative filtering", ***Arabian Journal For Science And Engineering***, 45: 9997--10017 (2020).

108. Bag, S., Kumar, S. K., and Tiwari, M. K., "An efficient recommendation generation using relevant Jaccard similarity", ***Information Sciences***, 483 (January): 53–64 (2019).

109. Iskhakov, L., Kamiński, B., Mironov, M., Prałat, P., and Prokhorenkova, L., "Clustering properties of spatial preferential attachment model", ***Lecture Notes***

In Computer Science (Including Subseries Lecture Notes In Artificial Intelligence And Lecture Notes In Bioinformatics), 10836 LNCS: 30–43 (2018).

110. Ciaglia, F., Stella, M., and Kennington, C., "Investigating preferential acquisition and attachment in early word learning through cognitive, visual and latent multiplex lexical networks", ***Physica A: Statistical Mechanics And Its Applications***, 612: 1–18 (2023).

111. De Collibus, F. M., Partida, A., Piškorec, M., and Tessone, C. J., "Heterogeneous Preferential Attachment in Key Ethereum-Based Cryptoassets", ***Frontiers In Physics***, 9 (October): 1–18 (2021).

112. Siew, C. S. Q. and Vitevitch, M. S., "Investigating the influence of inverse preferential attachment on network development", ***Entropy***, 22 (9): (2020).

113. Ly, A. and Yao, Y. D., "A review of deep learning in 5G research: Channel coding, massive MIMO, multiple access, resource allocation, and network security", ***IEEE Open Journal Of The Communications Society***, 2 (February): 396–408 (2021).

114. Seid, Abegaz Mohammed and Boateng, Gordon Owusu and Anokye, Stephen and Kwantwi, Thomas and Sun, Guolin and Liu, G., "Collaborative computation offloading and resource allocation in multi-UAV-assisted IoT networks: A deep reinforcement learning approach", ***IEEE Internet Of Things Journal***, 8: 12203--12218 (2021).

115. Ley, M., "The DBLP computer science bibliography: Evolution, research issues, perspectives", (2002).

116. Viswanath, B., Mislove, A., Cha, M., and Gummadi, K. P., "On the evolution of user interaction in facebook", (2009).

117. Leskovec, J., Kleinberg, J., and Faloutsos, C., "Graph evolution: Densification and shrinking diameters", ***ACM Transactions On Knowledge Discovery From Data (TKDD)***, 1 (1): 2-es (2007).

118. Carvalho, D. V., Pereira, E. M., and Cardoso, J. S., "Machine learning interpretability: A survey on methods and metrics", ***Electronics (Switzerland)***, 8 (8): 1–34 (2019).

119. Lui, Thomas KL and Tsui, Vivien WM and Leung, W. K., "Accuracy of artificial intelligence--assisted detection of upper GI lesions: a systematic review and meta-analysis", ***Gastrointestinal Endoscopy***, 92: 821--830 (2020).

120. Miao, Jiaju and Zhu, W., "Precision--recall curve (PRC) classification trees", ***Evolutionary Intelligence***, 15: 1545--1569 (2022).

121. Loola Bokonda, Patrick and Sidibe, Moussa and Souissi, Nissrine and Ouazzani-Touhami, K., "Machine Learning Model for Predicting Epidemics", ***Computers***, 12: 54 (2023).

122. Veluchamy, A., Nguyen, H., Diop, M. L., and Iqbal, R., "Comparative Study of Sentiment Analysis with Product Reviews Using Machine Learning and Lexicon-Based Approaches", ***SMU Data Science Review***, 1 (4): 1–22 (2018).

123. Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, “Exploring network structure, dynamics, and function using NetworkX”, in Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008

124. The pandas development team. (2023). pandas-dev/pandas: Pandas (v2.1.1). Zenodo. https://doi.org/10.5281/zenodo.8364959

RESUME

The researcher was graduated from the Computer Engineering Department, AlHussain Engineering collage in 26/6/2016 for the academic year 2015/2016 . She joined the master study in Karabuk University 22/2/2021.