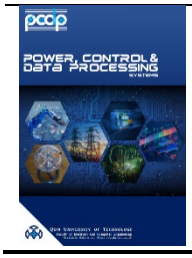
DOI: 10.30511/pcdp.2025.2047611.1012 - ISSN: 3060-8376 - Vol 2(3), 2025



POWER, CONTROL AND DATA PROCESSING SYSTEMS

Available Online at: https://pcdp.qut.ac.ir/

Graph Topology and Machine Learning for Enhanced Link Prediction

ARTICLE INFO

Article Type
Original Research

Authors
Lale Madahali[1, *]

[1] Department of Computer Science, Iowa State University, Ames, Iowa.
lalemadahali@gmail.com

*** *Correspondence***
Address: Department of Computer Science, Iowa State University, Ames, Iowa.
lalemadahali@gmail.com

Article History
Received: December 09, 2024
Accepted: June 07, 2025
ePublished: September 01, 2025

ABSTRACT

Social networking platforms have emerged as a focal point for academic and practical research, driven by their growing influence in modern society. Among various analytical tasks, link prediction has gained prominence as a critical challenge in social network analysis. This study examines three primary link prediction strategies: feature-based methods, Bayesian statistical models, and probabilistic relational models. Acknowledging the significant challenge of class imbalance in link prediction, we explore a combination of algorithmic techniques, advanced data preprocessing, and effective feature selection methods to improve predictive outcomes. Our research specifically focuses on coauthorship networks, leveraging topological attributes and enhanced data mining practices to extract meaningful patterns. Through extensive experimentation, we evaluate the performance of different approaches, emphasizing decision trees and Naive Bayes classifiers. These models consistently outperform alternatives in terms of predictive accuracy, particularly when assessed using F-measure and AUC metrics. Notably, our findings underscore the critical role of robust data preprocessing in achieving superior results, highlighting its potential to mitigate the impact of class imbalance. This study contributes valuable insights to the field of link prediction, offering practical guidance for developing more effective algorithms and addressing challenges in real-world social network applications.

Keywords: Graph Topology, Machine Learning, Link Prediction.

1 Introduction

The link prediction problem is defined as *By examining a particular moment in a social network's timeline, one can predict which pairs of currently disconnected nodes are poised to form a link. In this research, we consider three distinct snapshots of our networks taken at distinct intervals* [1]. The link prediction problem can be tackled through three primary strategies: feature-based link prediction, Bayesian probabilistic models, and probabilistic relational models [1]. In the feature-based approach, link prediction is treated as a supervised learning task, where each instance in the dataset represents a possible link (or edge) between two nodes (or vertices) [1]. Like other supervised learning challenges, the model is trained using data from a test period to make inferences about potential future connections. Bayesian probabilistic models aim to ascertain a probability value that reflects the likelihood of a link forming between two currently unlinked nodes [1]. This probability can serve as a feature within a classification model. Probabilistic relational models (PRMs), on the other hand, leverage the attributes of both nodes and edges to forge a comprehensive probability distribution for a network's nodes and edges [1]. PRMs are bifurcated into two sub-approaches: one based on Bayesian networks, which is appropriate for predicting directed links, and another based on relational Markov networks, suitable for undirected links [1]. In this paper, we employ the feature-based approach for link prediction. We utilized two datasets, Hep-ph and Gr-qc, which represent coauthorship networks spanning from 1992 to 2000. The Hep-ph dataset comprises 16,402 nodes and 156,742 edges, while the Gr-qc dataset contains 6,640 nodes and 27,443 edges. In both datasets, all connections are treated as undirected and unweighted. In this study, the graph G(V,E) represents a network where V denotes the set of vertices, corresponding to authors, and E represents the set of edges, indicating coauthorship links between them. The graph does not contain multiple edges between any two vertices. We divide the timeline into three distinct intervals: t-1, t, and t+1. The graph G[t-1] covers the period 1992 to 1994, G[t] covers the years 1995 to 1997, and G[t+1] spans from 1998 to 2000. The first interval, G[t-1,t], serves as the training period, while the subsequent interval, G[t,t+1], is used for testing. The process involves preparing the datasets, extracting relevant features, and then employing these features to train various classifiers. The goal is to enhance the predictive accuracy of these classifiers. For the training process, Weka is utilized, which is a free and open-source machine learning software [2]. We employ supervised learning to forecast which nodes within a social network, currently unconnected, might form a link. To optimize the use of training and testing data, we adopt three distinct methodologies: cross-validation, a supplied test set, and percentage split. In the supplied test set method, we use the network graph from 1992 to 1994 as the training set, while the test set comprises a prepared graph spanning from 1995 to 1997 and then from 1998 to 2000. The percentage split approach allocates a portion of the training set as the test set; in this experiment, the division was set at a 66% threshold—meaning 66% of the data was used for training and the remaining 34% for testing. Furthermore, we categorize features into three groups: neighborhood features, path-based features, and degree-based features, each of which is used independently for prediction purposes. In the analysis of the Hep-ph dataset, it was observed that the node neighborhood features outperformed the other categories. Conversely, for the Gr-qc dataset, the path-based features yielded the best results. To address the issue of data imbalance, we implemented both preprocessing and algorithmic approaches. The preprocessing techniques, specifically undersampling and oversampling, demonstrated superior effectiveness in our experiments.

2 Related Work

In addressing the link prediction problem, a prevalent approach involves feature-based link prediction. In their work, Liben-Nowell and Kleinberg [3] developed predictors, or features, derived from the graph and compared their performance against a random predictor. They focused on a core set of authors, defined as those who have co-authored at least three papers within the training and testing intervals. The predictor's output consists of a list of missing link probabilities, ranked in descending order. Graph-based features represent the most commonly employed features in link prediction. Cukierski [4] extracted a set of 94 distinct graph features as inputs for classification, utilizing Random Forests. They also highlighted the challenge of processing graphs with over a million nodes. Their findings suggested that the most robust classification performance is achieved by combining various categories of features that illuminate different aspects of the graph structure. Furthermore, Aouay et al. [5] approached the link prediction problem as a supervised learning task and amalgamated multiple features as input data for classification. To enhance prediction accuracy, they employed a select attributes algorithm. In their work [6], Fire et al. introduced two novel features: "friends-measure" and "same-community." The "friends-measure" quantifies the number of connections between the neighbors of two nodes [6], while "same-community" determines whether two nodes belong to a common community [6]. Their experiments, conducted on ten datasets, demonstrated improved predictive performance. Besides topological features, researchers have explored alternative features, such as node and edge attributes, for predicting new links. Al Hassan [7] utilized features like keyword match counts and the total number of papers in coauthorship networks. In a separate study, Scellato et al. [8] delved into the link prediction problem within online location-based social networks, employing the Gowalla social network dataset. They introduced a novel feature known as "place-friends," representing users who fre-

quent the same locations. This led to the creation of new prediction features based on the characteristics of places visited by users. They further elucidated a supervised learning framework that harnessed these prediction features to forecast new links among friends-of-friends and place-friends. The dynamic nature of social networks, with millions of nodes and edges constantly being added and removed, presents a significant challenge. Song et al. [9] addressed this challenge by defining proximity measures and an algorithm for estimating real-time proximity in highly dynamic social networks. Link prediction can be conceptualized as a random walk from a source node to a target. Backstrom et al. [10] formulated a supervised random walk method to learn a scoring function for each edge, enhancing the likelihood of a random walk encountering nodes with a higher likelihood of being connected. Their experiments encompassed networks like Facebook and other large-scale networks.

3 Research Methodology

In this section, we discuss the process of feature extraction for prediction and the preparation of data for classification. As is common in many classification problems, our dataset exhibited class imbalance, prompting the application of various techniques aimed at enhancing performance. In the final part of this section, we provide an explanation of the evaluation measures employed for assessing the algorithms. We also delve into the suitability of specific measures for addressing class imbalance issues

3.1 Data and Feature Extraction

In this research, two coauthorship network datasets were employed to assess the effectiveness of various techniques. These are publicly available datasets.

3.2 Preprocessing

To facilitate the feature extraction process using the LPmade software, certain data transformations were necessary. Firstly, as the software doesn't accept text inputs, author names were replaced with numerical representations. Secondly, the software is designed for processing directional graphs, whereas our graphs are unidirectional. To address this, each edge was duplicated, resulting in the creation of two edges for each original edge. For instance, edge "ab" was replaced by both "ab" and "ba". These transformations were carried out using Python.

3.3 Feature Extraction

In this step, for potential links, features from the topological structure of the graph are extracted. Common Neighbours. Let $\Gamma(a)$ be the set of neighbors of node a in the graph. $|\Gamma(a) \cap \Gamma(b)|$ is the number of neighbors nodes a and b have in common [11]. Idegree/Jdegree: The degree of the source or target node.

The Jaccard Coefficient, alternatively termed the Jaccard index or Jaccard similarity coefficient, serves as a statistical metric for gauging the similarity and diversity among sample sets. It quantifies the extent of similarity between finite sample collections by calculating the ratio of the magnitude of their intersection to the magnitude of their union [12].

$$J(a,b) = \frac{|a \cap b|}{|a \cup b|} \tag{1}$$

The Adamic/Adar measure, situated within the realm of web mining, is employed to ascertain the similarity between two web pages, particularly in understanding the degree of relatedness between them. This metric calculates the similarity score by evaluating the shared features between two given pages [6].

$$\text{Adamic/Adar}(a,b) = \sum_{z \in \Gamma(a) \cap \Gamma(b)} \frac{1}{\log(\Gamma(z))} \tag{2}$$

ShortestPathCount is a feature that quantifies the number of shortest paths linking a source node to a target node within a network. It operates by executing a breadth-first search that concludes upon reaching the level where the target node resides, tallying the frequency with which the target node appears at that specific level.

Preferential attachment is essentially a model for the growth of networks. According to this model, the likelihood of forming a new connection to a particular node is directly proportional to the degree of that node—meaning the number of neighbors it already has. Applying this to the context of co-authorship, we can infer that the probability of a co-authorship emerging between two individuals, nodes a and b, is proportional to the product of the number of neighbors each node has [11].

$$\text{Preferential Attachment}(a,b) = |\Gamma(a)| \cdot |\Gamma(b)| \tag{3}$$

Katz is a measure that takes into account a set of paths, with shorter paths given greater importance through an exponential damping factor. It is computed using the following formula:

$$\sum_{l=1}^{\infty} \beta^l \cdot |\text{paths}_{a,b}^{\langle l \rangle}| \tag{4}$$

Where $\text{paths}_{a,b}^{\langle l \rangle}$ is the set of all l-length paths from a to b, and $\beta > 0$ is a parameter of the predictor. Two variants of the Katz measure are considered: (1) unweighted, in which $\text{paths}_{a,b}^{\langle l \rangle} = 1$ if a and b have collaborated and 0 otherwise; (2) weighted, in which $\text{paths}_{a,b}^{\langle l \rangle}$ is the number of times that a and b have collaborated. [13]. Pagerank is a measure that involves a random walk on a graph (denoted as G), which begins at a specific node "a" and then proceeds iteratively to a neighbor of "a" chosen uniformly at random from the set $\Gamma(a)$ [12]. One challenge with this measure is its sensitivity to parts of the graph that are far removed from nodes "a" and "b", even when "a" and "b" are directly

connected by very short paths. To mitigate this sensitivity, a method involves allowing the random walk from "a" to "b" to occasionally reset, with the walker returning to "a" with a fixed probability denoted as α at each step. In this manner, remote parts of the graph are seldom explored. The Rooted Pagerank measure can be defined with a parameter $\alpha \in [0, 1]$, representing the stationary probability of reaching node "b" in a random walk. This walk returns to "a" with probability α at each step, while with probability (1-α), it moves to a random neighbor [12]. SimRank is a metric derived from a recursive definition that characterizes the similarity between two nodes in a graph [14]. It is based on the concept that two nodes are considered similar to the degree that they share common neighbors [14]. This similarity is quantified numerically, and the definition is initialized by setting similarity(a, a) to be equal to 1.

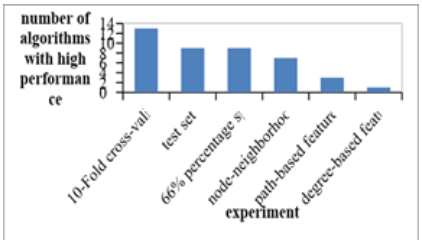
$$\text{similarity}(a,b) := \gamma \cdot \frac{\sum_{x \in \Gamma(a)} \sum_{y \in \Gamma(b)} \text{similarity}(x,y)}{|\Gamma(a)| \cdot |\Gamma(b)|} \quad \text{for } \gamma \in [0,1] \quad (5)$$

3.4 Experiments

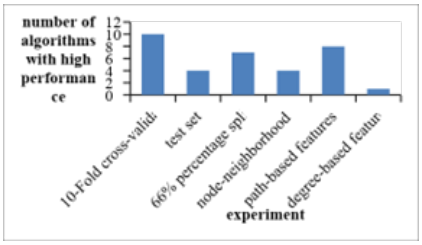
In this experiment, two coauthorship networks, namely hep-ph, and gr-qc, were utilized. The hep-ph network consists of 16,402 nodes and 156,742 undirected and unweighted edges, while the gr-qc network comprises 6,640 nodes and 27,443 undirected and unweighted edges. The training set was defined as the transition from the graph at time (t-1) to time (t), while the test set encompassed the transition from the graph at time (t) to time (t+1). Following the extraction of features, classifiers were trained, and their results were subsequently compared. To facilitate the use of classification algorithms, the free and open-source machine learning software Weka was employed [13]. In our application of supervised learning to predict unconnected nodes, we employed three distinct methods for utilizing the training and test data: cross-validation, a supplied test set, and percentage split. In the "supplied test set" approach, we utilized a test set that had been previously prepared from the graph transitions between the years 1995-1997 and 1998-2000. For the "percentage split" method, we allocated a specific percentage of the training set to serve as the test set. In this experiment, the chosen percentage was 66%, which implies that 66% of the data was designated for the training set, with the remaining 34% utilized as the test set. The chart below illustrates the number of classification algorithms that exhibited precision, recall, and F-measure values greater than zero, indicating their acceptable performance. Further discussion on the F-measure will follow.

3.5 Dealing with Imbalanced Data

Many link prediction scenarios face a common challenge known as the class imbalance problem. This issue typically arises when there is a significant disparity in the number of negative links (non-existent connections) compared to positive links (actual connections). Specifically, class imbalance occurs in classification problems where one class, often referred to as the majority class, has a much larger number of examples than the other class, known as the minority class. Imbalanced datasets can present challenges for machine learning algorithms, as standard techniques may be biased toward the majority class and neglect the minority class.



(a)



(b)

Figure 1: (a) Number of algorithms with acceptable performance for Hep-ph dataset. (b) Number of algorithms with acceptable performance for Gr-qc dataset.

Class imbalance problems often lead to the misclassification of minority class samples, as classification algorithms tend to prioritize correctly classifying the majority class samples. However, in many real-world applications such as fraud detection and the identification of cancerous cells [15], misclassifying minority class samples can have more severe consequences than errors in the majority class. To address this problem, three main approaches are commonly employed to deal with the classification of imbalanced data.

Algorithmic Approach. This approach involves modifying or designing classification algorithms to better handle imbalanced datasets. These adaptations can include introducing cost-sensitive learning, using ensemble methods, or exploring various sampling techniques.

Preprocessing. In this approach, the dataset itself is modified before applying standard machine learning algorithms. Techniques such as oversampling the minority class, undersampling the majority class, or generating synthetic data are used to rebalance the dataset.

Feature-Selection Approach. This approach focuses on selecting or engineering relevant features that can help improve classification performance on imbalanced data. Feature selection may involve prioritizing

attributes that have a more significant impact on the minority class, thus enhancing the classifier's ability to identify them. By adopting one or a combination of these approaches, machine learning models can better handle imbalanced data, leading to more accurate and balanced predictions. In this paper, both preprocessing and algorithmic approaches were employed to enhance the performance of classification algorithms in dealing with class imbalance. The class imbalance ratio in our datasets is substantial, with a ratio of 1/318 in the training set and 1/587 in the test set for the Hep-ph dataset. In the Gr-qc dataset, the ratio is similarly imbalanced, with a ratio of 1/231 in the training set and 1/353 in the test set. These ratios underscore the significance of addressing class imbalance to ensure the accurate classification of minority class samples in these domains.

3.5.1 Preprocessing Approach

As previously mentioned, one effective strategy for enhancing algorithm performance is to address class imbalance by manipulating the dataset. Two common techniques for this purpose are undersampling and oversampling. In under-sampling, the dataset is balanced by randomly removing some samples from the majority class. This process involves randomly deleting samples from the majority class until its size matches that of the minority class. However, a drawback of under-sampling is the potential loss of valuable data. In contrast, over-sampling focuses on the minority class. This technique involves randomly duplicating or copying minority class samples to increase their representation in the dataset. Over-sampling helps balance the class distribution without discarding data. Both under-sampling and over-sampling aim to mitigate class imbalance and improve the performance of classification algorithms, ensuring that the minority class is adequately represented in the training data. The choice between these techniques often depends on the specific dataset and the nature of the problem being addressed.

3.5.2 SMOTE

SMOTE, or Synthetic Minority Oversampling Technique, is an over-sampling method designed to combat class imbalance by generating synthetic instances for the minority class [16]. Instead of merely duplicating existing samples, SMOTE creates new synthetic examples by selecting a minority class sample and then connecting it to a subset of its k nearest neighbors. These nearest neighbors are chosen randomly for the creation of synthetic data, effectively increasing the representation of the minority class and balancing the dataset, thereby enhancing the classifier's performance in accurately predicting minority class instances while retaining the integrity of the original data [16].

3.6 Algorithmic Approach

In this technique, a combination of two or more algorithms is employed, where Bagging (Bootstrap Aggregation) and Boosting are two algorithmic approaches that are utilized in conjunction with other algorithms [17]. For instance, Boosting is commonly paired with Support Vector Machines (SVM) to enhance performance. In this particular experiment, Bagging and AdaboostM1 are employed in combination with other algorithms to assess their impact [17]. The Bagging method is employed to generate multiple iterations of a predictor, leading to the creation of an aggregated predictor [18]. When predicting numerical outcomes, the aggregated version calculates the average over these iterations, while in the case of class prediction, it conducts a majority vote. By creating bootstrap replicates of the learning set and utilizing these as new learning sets, multiple versions of the predictor are generated. This technique has shown significant improvements in accuracy when tested on real and simulated datasets, particularly using classification and regression trees. Bagging is especially effective when perturbing the learning set can result in noticeable changes in the generated predictor, thereby enhancing accuracy [18].

Algorithm 1 Bagging

Input: S: Training set, T: Number of iterations, N: Bootstrap size, I: weak learner
Output: Bagged classifier: $H(x) = \text{sign}\left(\sum_{t=1}^{T} h_t(x)\right)$ where $h_t \in [-1, 1]$ are the induced classifiers
    Assumptions:
        h_t are weak classifiers.

for $t = 1$ **to** T **do**
    $S_t \leftarrow \text{RandomSampleReplacement}(N, S)$
    $h_t \leftarrow I(S_t)$
end for

Boosting, also known as Arcing, Adaptive Resampling, and Combining, was introduced by Schapire [19], demonstrating that a weak learner can be transformed into a strong learner within the framework of Probably Approximately Correct (PAC) learning. AdaBoost [20], a prominent algorithm in the Boosting family, was the pioneering approach in Boosting and is recognized as one of the top ten data mining algorithms [21]. AdaBoost significantly reduces bias (by 85%) and shares similarities with support vector machines (SVMs by boosting margins [22]). In the training process, AdaBoost serially trains each classifier on the entire dataset. After each iteration, it places a strong focus on challenging instances, aiming to accurately classify examples in the next iteration that were previously misclassified. Consequently, it places a greater emphasis on samples that are inherently more challenging to classify. Initially, all instances have equal weights, but with each iteration, the weights of misclassified in-

stances are increased, while the weights of correctly classified instances are reduced [17]. This approach underscores AdaBoost's ability to adapt and continually refine its performance to better handle complex datasets.

3.7 Performance Evaluation Metrics

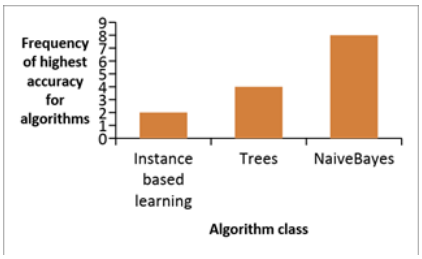
In the evaluation of classification algorithms, several measures are used to gauge the performance of a classifier with respect to a given dataset. One of these crucial measures is the confusion matrix, which provides a detailed breakdown of the classification results. This matrix illustrates the number of samples from each class that are correctly or incorrectly classified. Key components of the confusion matrix include: True Positives (TP): The number of samples that are genuinely positive and correctly predicted as positive. False Positives (FP): The number of samples that are actually negative but erroneously predicted as positive. True Negatives (TN): The number of samples that are genuinely negative and correctly predicted as negative. False Negatives (FN): The number of samples that are truly positive but mistakenly predicted as negative. ROC Curve, which provides a visual representation of a classifier's performance [23]. This curve is constructed by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR). The ROC curve offers a valuable means of assessing how well a binary classifier distinguishes between positive and negative classes. Ideally, the curve should be closer to the point (0,1), indicating a high TPR while keeping the FPR low. In addition to the ROC curve, the ROC Area, often referred to as AUC (Area Under the Curve), is calculated. The ROC area quantifies the area under the ROC curve, with a larger area indicating superior classifier performance. A higher AUC signifies that the classifier is better at distinguishing between positive and negative instances [23]. In cases where binary classification involves imbalanced datasets, where the minority class holds greater significance than the majority class, traditional accuracy may not adequately reflect performance [18]. For instance, an algorithm that consistently assigns new samples to the majority class can achieve high accuracy, but this may not be ideal when the minority class is more critical. In such scenarios, evaluation metrics like precision, ROC area, and F-measure are more suitable for demonstrating classifier performance, as they provide a more nuanced view of how well the model handles the minority class [15].

4 Results

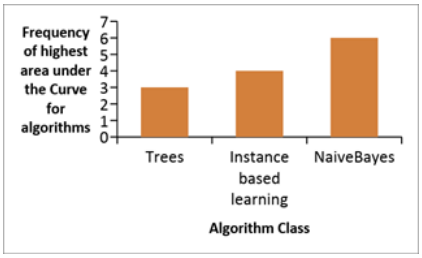
The experiments conducted in this study aimed to mitigate the effects of imbalanced data in a classification problem, and two distinct approaches were explored: algorithmic level and data level, often referred to as preprocessing. Notably, the preprocessing approach yielded superior results on our datasets. In Weka, classification algorithms are categorized into several categories, including Bayes (comprising BayesNet, NaiveBayes, ComplementNaiveBayes, and more), Functions (encompassing SVMs and Neural Nets), Lazy or Instance-based learning, meta (for ensemble algorithms), and trees (such as J48, random tree, NBTree, etc). For the Hep-ph dataset, the most significant contributions to accuracy were made by algorithms falling into different categories, including a contribution of 2 for Instance-based learning, 4 for trees, and 8 for the NaiveBayes category. The distribution of algorithm categories in terms of their highest performance measures is displayed below. Remarkably, it is evident that the under-sampling method proved to be the most effective technique for our datasets, as illustrated in Figures 3 and 4. Furthermore, the algorithmic approaches employed resulted in significant changes in the performance outcomes of nearly 50% of classification algorithms, while the remaining 50% remained unaffected [13].

5 conclusion

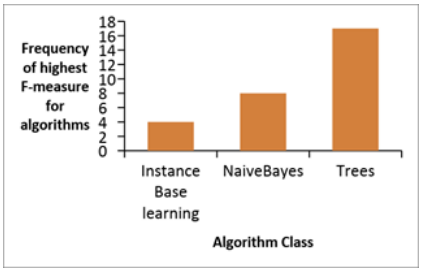
Addressing the issue of imbalanced data is undeniably crucial, and this paper has introduced a range of strategies to tackle this challenge. The proposed approaches have proven successful in enhancing the results.



(a)

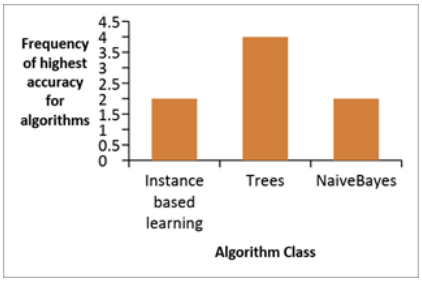


(b)

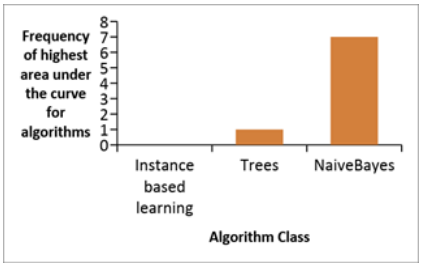


(c)

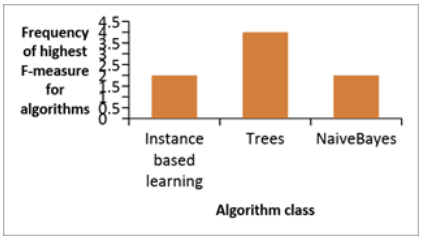
Figure 2: (a) Frequency of highest accuracy for Hep-ph. (b) Frequency of highest AUC for Hep-ph. (c) Frequency of highest F-measure for Hep-ph.



(a)



(b)



(c)

Figure 3: (a) Frequency of highest accuracy for Gr-qc. (b) Frequency of highest AUC for Gr-qc. (c) Frequency of highest F-measure for Gr-qc.

However, it's worth noting that our approach had a limitation in terms of computational resources, as running the algorithms on these extensive datasets was time and memory-intensive. Future research efforts should focus on addressing these time and memory complexities to facilitate further studies in this area. There are several promising avenues for future work in this field. One potential direction is the inclusion of new graphical features to further enhance algorithm performance. Additionally, exploring alternative learning methods, such as active learning, cost-sensitive learning, and kernel-based learning, could offer valuable insights and improvements to the classification of imbalanced data.

Table 1: Percentage of improvement for preprocessing approaches for Gr-qc dataset

Improvement technique	Percentage Change
10-fold cross validation	37%
Undersampling	91%
Oversampling	63%

Table 2: Percentage of improvement for preprocessing approaches for Hep-ph dataset

Improvement technique	Percentage Change
10-fold cross validation	33%
Undersampling	87%
Oversampling	63%

Disclosure of Potential Conflicts of Interest

The Authors declare that there is no conflict of interest.

References

[1] M. Al Hasan and M. J. Zaki, "A survey of link prediction in social networks," in *Social Network Data Analytics*. Springer, 2011, pp. 243–275. doi: 10.1007/978-1-4419-8462-3-9 .

[2] M. Hall *et al.*, "The WEKA data mining software: An update," *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009. doi: 10.1145/1656274.1656278 .

[3] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Am. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007. doi: 10.1145/956863.956972 .

[4] W. Cukierski, B. Hamner, and B. Yang, "Graph-based features for supervised link prediction," in *Neural Networks (IJCNN), The 2011 International Joint Conference on*, 2011, pp. 1237–1244. doi: 10.1109/IJCNN.2011.6033365 .

[5] S. Aouay, S. Jamoussi, and F. Gargouri, "Feature based link prediction," in *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, 2014, pp. 523–527. doi: 10.1109/AICCSA.2014.7073243 .

[6] M. Fire, L. Tenenboim-Chekina, R. Puzis, O. Lesser, L. Rokach, and Y. Elovici, "Computationally efficient link prediction in a variety of social networks," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 1, p. 10, 2013. doi: 10.1145/2542182.2542192 .

[7] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," Tech. Rep., 2006.

[8] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting place features in link prediction on location-based social networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, p. 1046. doi: 10.1145/2020408.2020575 .

[9] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu, "Scalable proximity estimation and link prediction in online social networks," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, 2009, pp. 322–335. doi: 10.1145/1644893.1644932.

[10] U. L. Backstrom and U. J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *WSDM '11 Proceedings of the fourth ACM international conference on Web search and data mining*, 2011, pp. 635–644. doi: 10.1145/1935826.1935914.

[11] M. E. Newman, "Clustering and preferential attachment in growing networks," *Phys. Rev. E. Stat. Nonlin. Soft Matter Phys.*, vol. 64, no. 2 Pt 2, p. 025102, 2001. doi: 10.1103/PhysRevE.64.025102.

[12] F. Chung and W. Zhao, "Pagerank and random walks on graphs," in *Fete of Combinatorics and Computer Science*. Springer, 2010, pp. 43–62. doi: 10.1007/978-3-642-13580-4-3.

[13] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 2011. doi: 10.1016/C2009-0-19715-5.

[14] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 538–543. doi: 10.1145/775047.77512.

[15] R. Longadge and S. Dongre, "Class imbalance problem in data mining review," Tech. Rep., 2013.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002. doi: 10.1613/jair.953.

[17] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *Syst. Man, Cybern. Part C Appl. Rev. IEEE Trans.*, vol. 42, no. 4, pp. 463–484, 2012. doi: 10.1109/TSMCC.2011.2161285.

[18] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, pp. 123–140, 1996. doi: 10.1007/BF00058655.

[19] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990. doi: 10.1007/BF00116037.

[20] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Springer, Berlin, Heidelberg*, 1995, pp. 23–37. doi: 10.1006/jcss.1997.1504.

[21] X. Wu *et al.*, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, Jan. 2008. doi: 10.1007/s10115-007-0114-2.

[22] C. Rudin, I. Daubechies, and R. E. Schapire, "The dynamics of adaboost: Cyclic behavior and convergence of margins," *J. Mach. Learn. Res.*, vol. 5, pp. 1557–1595, 2004.

[23] K. P. Murphy, "Performance evaluation of binary classifiers," Tech. Rep., 2007.