Journal of Theoretical and Applied Information Technology
15th May 2025. Vol.103. No.9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

GRAPH-BASED FRIEND RECOMMENDATIONS ON SOCIAL MEDIA USING MACHINE LEARNING

SRINIVASA RAO. D[1], RAMESH BABU.CH[2], SRAVAN KIRAN.V[3], M.KOTESWARA RAO[4], REVATHI. A[5], RAJASEKHAR. N[6]

[1,3,4,5] Department of Information Technology, VNRVJIET, Hyderabad, India
[2]Department of CSE, MGIT, Hyderabad, India
[6]Department of Information Technology, GRIET, Hyderabad, India
E-mail: [1]dammavalam2@gmail.com, [2]chramesh522@gmail.com, [3]shravankiran12@gmail.com, [4]mailtomkrao@gmail.com, [5]revathiannam@gmail.com, [6]rajasekhar531@gmail.com,

ABSTRACT

Social media platforms facilitate the exchange of vast amounts of information within their networks, where users and their friends form vast communities. To foster meaningful connections, recommendation systems play a vital role in linking users with similar interests. Existing social media networks rely on users' networks to suggest friends, but this approach may not accurately capture users' real-life preferences. This study presents a novel Facebook friend recommendation system utilizing the XGBoost algorithm, which analyzes users' social graphs, incorporates diverse connection parameters, and evaluates various similarity measures. Specifically, we employ similarity measures such as Jaccard distance and the Otsuka-Ochiai coefficient to filter graph information and calculate user ratings. The model leverages features extracted from Adar Index, Hits Score, Katz Centrality, and Page Rank. Our proposed system, utilizing both Random Forest and XGBoost algorithms, achieves 99.99% accuracy. Notably, XGBoost excels in efficiency and speed, making it an ideal choice for large-scale social networks. By suggesting friends with similar ratings, our system enhances user experience and fosters meaningful connections within the Facebook community.

Keywords-*XGBoost, Friend Recommendation System, Adar Index, Hits Score, Katz Centrality, Page Rank, Jaccard Distance, Otsuka-Ochiai Coefficient.*

1. INTRODUCTION

Social media platforms have become the go-to source of information for a vast and diverse community, facilitating a dynamic environment where individuals can share their experiences, interests, and expertise. With millions of users relying on these platforms for global content sharing, social media sites have achieved unprecedented success, driven by the exponential growth of their global communities. To maintain meaningful connections within these platforms, a robust and precise recommendation system is crucial. This system enables users to connect with relevant individuals, fostering a more engaging and personalized experience. By leveraging recommendation systems, social media platforms can suggest users to their friends or acquaintances, further enriching the online social experience. The social media recommendation system employs machine learning computing methods, such as natural language processing and data mining, previously used in other recommendation systems. The recommendation systems brought information that was enjoyable to the user through information filtering techniques such as content filtering. Over the past five years, recommendation systems used in social media have evolved into personalization-based systems, which have gained popularity. Furthermore, social media sites now employ these personalized recommendation systems, which. provide recommendations more precisely and accurately than computational methods.

Social media websites primarily utilize collaborative filtering techniques and match user personalities to facilitate social recommendations. This personality matching increases mutual understanding and user social context, resulting in more friend recommendations and real-time interaction. This personality matching, in conjunction with collaborative filtering, maps a user's social network

to characters from TV and reality shows that most closely align with their personalities [1]. A new friend recommender system, which combine existing friends-of-friends algorithms with content recommendations from user data, complement personality matching techniques, it employs a clustering model, grouping users according to their features and the number of dimensions they possess [2]. A recommendation approach to handle user mobile trajectories generating better friend recommended based on the similarity metrics of their trajectories. Location-based social networks make use of this recommendation approach. The recommender system establishes similarity measures for a user based on their mobile trajectories, and then evaluates the similarities between two maximal semantic trajectory patterns that the system operates on [3].

Another approach to increasing the accuracy of the friend recommendation system is to use the social information attached to the user, for example, his social friends, posts he liked, tags he was tagged with, etc. We refer to this recommendation system approach as a top-N recommendation system, which employs multiple forms of data, such as the users' social information, to tag the profiles they appear in. This recommendation system establishes a neighbor's social network, builds a matrix of users based on their features, calculates the priority of each user in the matrix, calculates the priority of tags, and then generates recommendations by comparing the priority of tags with the priority of users [4]. Another recommendation approach that social media sites should add is involving neural networks and integrating deep learning into them, which increases the accuracy of recommender systems. An artificial neural network can improve accuracy by matching users' personalities. This trains the model to be robust, divides the recommendation system into layers, such as friend priority, and classifies them. Graph mining sorts the distance between users' priorities and matches personalities with similar types [5]. A new recommended system is to use convolutional neural networks with graph mining instead of artificial neural networks, which can handle a huge set of data concerning for user privacy. Firstly, we will solve the link prediction problem of the social graph, which contains only dependent features between users. We design a convolutional neural network specifically for a given graph. Next, we feed dependent features into the convolutional neural network, enabling us to extract nodes as vectors and pairs of vectors for each node. After this, we can predict the links in a graph using these vectors of nodes [6].

Another type of recommendation system bases friend recommendations on the end user's lifestyle, integrates it with a friend's social graph, and then extracts relevant features from it. Initially, it begins by identifying the dominant lifestyles of users, such as their activities. It then compares these lifestyles to those of a friend who shares them. A social graph gathers user details such as profession, location, and content. It then extracts other social features from the graph and associates them with a person who shares similar social features and has the closest score to the user [7]. A recommender system tailors personalized recommendations based on the user's location, interests, work domain, profession, gender, social network, nature, and content, all of which contribute to the user's social content. The system extracts all of this information and content from the user's various social media and networking sites, then bases recommendations on the social content [8].

2. RELATED WORKS

The Friends Recommendation System uses machine learning to recommend users based on their behavior, including followers and followees. This system can be used in various applications, including search engine recommendations, music recommendations, and e-commerce recommendations. The work done by authors [9] uses Facebook as an example, comparing the accuracy of different machine learning algorithms to determine the most effective recommendation system. Results shows Random Forest and Light GBM have lower accuracy, while XGBoost and Cat Boost have 95% accuracy.

Research on Mobile App Recommendation Systems (MARS) [10] is growing, involving big data and social information. However, privacy issues and social network-based app recommendations are not fully explored. A questionnaire survey was conducted to understand users' opinions on social interaction information and privacy. A prototype was implemented, resulting in an increase in click rate up to 0.66.

The study [11] aims to improve friend recommendation in Online Social Networks (OSNs) by considering users' evolving interests. It proposes a Learning Partner Recommendation Framework Evolution based on fine-grained learning interest (LPRF-E), which predicts evolving interests and recommends learning partners based on similarity. Users' social influence refines the framework (LPRF-F). Experiments on real datasets from the Chinese

University MOOC and Douban Book validate the proposed models' high accuracy and high-quality recommendations [12].

A novel approach to providing personalized social network recommendations for academic students is proposed in study [13], introducing the Personalized Explainable Learner Implicit Friend Recommendation Method (PELIRM). This innovative technique leverages implicit friendships extracted from learner interactions, trust levels, and the three-degree impact hypothesis. Additionally, PELIRM considers various factors, including research interests, geographic data, cosine similarity, and term frequency-inverse document frequency (TF-IDF) relationships. Researchers [14-15] proposes a dynamic graph-based embedding (DGE) model for social recommendation that can recommend users and items in real time. The model creates a heterogeneous user-item network and maintains it as it evolves. DGE captures temporal semantic effects, social relationships, and user behavior patterns, generating recommendations using simple search methods or similarity calculations. Experiments show its advantages over other methods.

Researchers have proposed a recommendation system that uses user data to infer the end user's lifestyle from their daily activities [16]. The recommendation system then computes the lifestyles of all other users and compares users who have similar lifestyles. Following this step, the system recommends the users with the most similar lifestyles as friends. This type of model operates by mining text. This recommendation system creates documents for the user to store their data in and then applies collaborative filtering to these documents to retrieve the user's daily lifestyle. Next, the system calculates users' similar lifestyles, creating a graph that aligns with another user's identical lifestyle. The system then recommends the user with the highest lifestyle as a friend to the end user. Researchers [17] introduced a recommendation system that not only encompasses shared interests among users but also steers recommendations in multiple directions. We extract four features from a Twitter account, including the keywords and hashtags used in users' tweets, their physical distance from each other, and their introduction. We use these extracted features to calculate identical features between users and recommend the most similar user as a friend. We have proposed a unique recommendation system that maintains messages based on real-time chats and messages exchanged among users. Then, the recommendation system recommends a user from a customized list of friends. If a user in the list has more interactions with the end user, they receive the highest score in the recommendation system.

Researchers [18-19] have proposed a new friend recommendation system that leverages the shapes and properties of social graphs. The system constructs a social graph that includes a user and his friends. This recommendation system identifies various patterns between the end user and other users who exhibit similar behaviors. Research [20] proposes a location recommendation system that generates recommendations based on tracking various user locations. We proposed a framework for a location recommendation system that uses collaborative filtering to initiate valuable and native location recommendations for specific individuals. We named this framework the Friend-and-Native-Aware Approach for Collaborative Filtering (FANA-CF). This type of location recommendation system defeats recommendation systems that use only one source and provide individually unique recommendations.

Social media platforms [21] have proposed a special vectorizing method for auditing graphs, which is a major solution. This method evidently identifies a community of people among users, predicting links between nodes, user preferences, and so on. This result holds significant importance for executives of social media platforms, as it can enhance the content of social media across recommendation systems. Researchers [22] have proposed a novel friend recommendation system that identifies users with similar actions and activities based on image similarity. We've created an online network called Friend Book, which uses feature-based picture comparison to make friend recommendations. The model extracts feature from people's images and compares them to those of the target person. We refer to this model as semantic friend matching.

Researchers [23] have proposed a geographical friend recommendation system that uses special GPS patterns to showcase real-time contacts between users and their relationships. The recommendation system employs a walk-based framework to suggest friends based on geo-patterns. This system uses a GPS trajectory to connect the GPS records of a specific user within a social network. Common locations do not contain any similarities to public places. This system focuses on real-time friend recommendations, which are based on users having the same location and relating to the geographical area.

We tested this friend recommendation system using

two algorithms: Random Forest and XGBoost, but we prefer XGBoost because of its faster speed and higher accuracy. In this research, we are developing a friend recommendation system based on the XGBoost algorithm, which provides recommendations to friends based on a person's followers and followees. We are using a Facebook dataset comprising 1.86 and 9.43 million nodes(people) and edges, respectively. This dataset has two columns, where the first column is the source node and the second column is the destination node. We have taken into account three generalization strategies for this dataset. We have a lot of similarity measures, such as Jaccard Distance, and a lot of feature extractions in which we will extract features for a particular user and rank them using PageRank. This model's classification algorithm is XGBoost. XGBoost is a gradient-boosted decision tree that is well-known for its speed and performance. The model generates a confusion matrix to describe the performance of the classification algorithm.

Past and current friend recommendation schemes on social media frequently fight with accuracy, and scalability, while graph-based methods utilizing machine learning provides more strong solution by leveraging network constructions and user connections for adapted and competent references.

Graph-based friend recommendations on social media, driven by machine learning, having troubles including data spareness, providing precise or related recommendations for new users, processing difficulty, and the need for precise user behavior forecast, demanding robust procedures and data management approaches.

There are problems in existing algorithms, current social media friend recommendation systems frequently decrease in accurately catching users' true favourites, as they mainly rely on current user networks, possibly missing out on probable connections built on common comforts or actions.

Our proposed work offers an original Facebook friend recommendation system that influences the XGBoost algorithm to examine users' social graphs, combine various connection parameters, and assess different common measures to filter graph evidence and compute user rankings

3. METHODOLOGY

A. Dataset

The Facebook recruiting dataset serves as the foundation for this recommendation system. Kaggle has listed this dataset, which covers Facebook data with 9.43 million edges and 1.86 million nodes. Two columns, the first representing the source node and the second representing the destination node, link the source and destination nodes in this dataset. The model considers only the 100,000 rows for which it was created, not the complete data. First of all, social media giants like Facebook, Twitter, and Instagram use the FoF algorithm (Friends of Friends algorithm) as their recommendation system. This algorithm makes recommendations for friends based on each user's unique social network. This recommendation system works by saying that if A is a friend of B, then it recommends the friends of the A social network to B so that A may show them that they are his friends soon. Therefore, this recommendation system may lack accuracy and potentially mislead users, as it fails to consider real-life scenarios where the recommended friend may not be well-known to them [40].

To address this issue, we incorporated an influence algorithm into the FoF algorithm, which improves the recommendation system's accuracy compared to the FoF algorithm, resulting in more precise decisions when recommending friends to users. This algorithm works by using methods like the influence map, which takes parameters from the social graph and the user. The influence map generates a map of users who share at least one friend with the input user but do not count as the input user's friends. Secondly, this contains an influence graph, which sorts the keys of the above-given map in descending order of values. This influence algorithm considers various parameters, including the extent to which a user's social network influences their interests and the content they consume. Results confirm that the influence algorithm for the recommendation system outperforms the FoF algorithm in accuracy. However, the issue lies in the fact that the influence algorithm's recommendation system only functions accurately for smaller datasets, such as those containing up to 1 lakh individuals. This recommendation system may not be effective for large social media platforms such as Facebook, which manage billions of data points. To address this issue, we created the XGBoost friend recommendation system using graph mining. This system, renowned for its speed and performance, consistently produces the most accurate recommendations. This system has a 100% train score, which makes it more accurate even with billions of pieces of data [24-26]. The authors [27] proposed a bidirectional encoder representation from transformers (BERT), with the aim of constructing a specialized model capable of identifying fake news that circulates across various platforms. Authors presented a new probabilistically weighted extension of the Adamic/Adar quantity for heterogenous

evidence networks to establish the possible assistances of diverse evidence, mainly in cases where similar associations are actual sparse [28]. The main goal of this approach is to improve the spread of accurate news on social media platforms while eliminating fake news.

Research objectives from the proposed methodology

1. Development and Assessment of the XGBoost Algorithm
2. Comparation with Random Forest Classifier
3. Examine Graph-Based Features
4. Investigation of Performance Metrics for Proposed Methodology

B. XGBoost Recommender System

The XGBoost algorithm is the most popular machine learning algorithm, whether we use prediction, regression, or classification. There is a concept of Extreme Gradient Boosting for XGBoost that uses advanced and high-level 1 and level 2 techniques for perfect fitting. Regression trees, which are very weak and have a low learning rate, are the first step in this model training process. Gradient-boosted trees, in turn, utilize these regression trees. These special types of regression trees are very similar to decision trees, which can make decisions on their own, except that their learning rate is very low. These regression trees bear similarities to decision structures used for decision-making. The XGBoost algorithm will provide an accurate prediction once a tree reaches its last node, at which point it assigns a separate score to each leaf node, forming the gradient tree. There will be a prediction variable y for each iteration, and a gradient tree t grows based on y. The primary objective of the XGBoost algorithm is to decrease the score at each iteration, thereby lowering the overall score. This is achieved by focusing on the previous loss function used in training the model i-1, which in turn forms a new gradient tree t. This XGBoost algorithm allows regression trees to grow in a sequential manner and learn from previous iterations, minimizing the overall score.

Gradient descent, a feature of XGBoost, displays the optimal values for both the overall score of the gradient tree and its leaf nodes, a process known as impurity prediction and cleaning. During the algorithm's learning phase, the loss function computes various techniques to reduce errors or accurately fit a model, significantly contributing to the reduction of regressive tree complications. This technique has positive values and also has values between 0 and 1, which shows the maximum running complexity. However, if we consider these regularization terms as zero, then there is no conflict between predicting the results of gradient-boosted trees and XGBoost, thereby rendering the XGBoost algorithm meaningless. The XGBoost algorithm also includes parameters like the learning rate, which represents the shrinkage we perform at each step. XGBoost also employs column subsampling, a technique that reduces overfitting by randomly sorting a subdivision of features. We are adding several methods to the XGBoost techniques to diminish errors and accurately fit a model.

Graph-based friend recommendations on social media suggest numerous fortes, counting enhanced accuracy, amplified personalization, scalability, strength to noise, suppleness, understandability, Moreover, graph-based approaches can extract higher-order associations among users, offer diverse friend references, and provide real-time recommendations as users interrelate with the social network. Inclusively, graph-based friend recommendations can deliver an additional complete understanding of users' social network, important to more operative and attractive friend recommendations which are not available while other methods.

We prefer XGBoost because of its features, such as growing trees in sequential order, which reduces errors in the last tree, using gradient descent, which decreases the overall score thus minimizing loss function, improving the speed and running of the model with boosting trees simultaneously, regularization parameters, and learning rates like different types of parameters. XGBoost outperforms other tree-ensembling algorithms due to its regularization parameter, which simultaneously reduces variance and minimizes the loss function through gradient descent. Moreover, XGBoost offers numerous hyperparameter subsamples, such as bootstrapping the training data and considering a limited amount of data, determining the maximum depth of trees to process a tree, selecting the number of trees to utilize, and minimizing the number of weights assigned to their child nodes for node division.

Boosting in XGBoost creates trees with fewer splits, similar to the bagging techniques used by Random Forest. Small-scale, divided trees are extremely computed. Learning functions employ a variety of hyperparameters, displaying a tree's maximum depth, aggregate iterations, and learning metrics for gradient-boosted trees. K-fold cross-validation filters out all these beneficial values. The boosting function is defined as a three-staged function. To initiate the function, we train an F0 model to predict

the value of the target variable, y. After training the F0 model, the remaining values, designated as y, become the target variable's value in F0. We will develop a new model, h1, which will store the value of the target variable in F0. We will have a new model, F1, which is a combination of the F0 and h1 models, reducing loss and increasing speed. The mean square error of the F0 model is higher than that of the F1 model, and the subsequent tree aims to reduce the errors in the previous model. Afterward, to enhance the performance of this model in F1, we will construct a new model in F2, which incorporates the values of the target variable from F1 and aims to reduce the errors in F1, a process known as boosting. Starting with m iterations, we can minimize further residuals as follows:

$$F_1(x) <- F_0(x)+h_1(x)$$

$$F_2(x) <- F_1(x) + h_2(x)$$

$$F_m(x) <-F_{m-1}(x)+h_m(x) \quad (1)$$

To apply the XGBoost algorithm to our model, we must specify the hyperparameter n_estimators, which refers to the number of estimators or trees we plan to use. This ranges from 105 to 125, indicating that we can utilize either 105 or 125 trees. Another hyperparameter, max_depth, specifies the maximum depth of a tree that the algorithm must process, ranging from 10 to 15. This implies that the algorithm can process up to 10 or 15 nodes per regression tree. Then, in the Randomized Search CV function, we can input the number of estimators, the maximum depth of each tree, distribute the trees based on n_estimators and max_depth, set the maximum number of iterations to 5, and use the f1 scoring method to rank the overall score for each tree. The XGBoost algorithm will apply the model with the specified hyperparameters and the XGB Classifier will train the model using these applied hyperparameters. Once we apply XGBoost to the model, we can implement an algorithm with parameters such as a base score of 0.5, a booster gbtree for boosting gradient trees, sampling at level 1, sampling at node 1, sampling at tree level 1, learning rate of 0.1, and maximum delta steps of the maximum depth of each tree is 14, the minimum child weights are 1, and there are no missing nodes.

None, which selects None of the nodes should be processed; the number of estimators or trees should be 120; the number of jobs should be 1; the objective of binary logistic regression should be implemented in regression trees; and other parameters, etc. should be used. We apply the XGBoost algorithm to the model's parameters, which include various hyperparameters, regularization parameters, and the specified parameters are shown in TABLE. I.

Table I. Parameters Of Xgb Classifier Implemented On Our Model

XGBCLASSIFIER BASE_SCORE	0.5
booster	gbtree
colsample_bylevel	1
colsample_bynode	1
gamma	0
learning_rate	0.1
max_delta_step	0
max_depth	14
min_child_weight	1
missing	None
n_estimators	120
n_jobs	1
nthread	None
objective	Binary:logistic
random_state	0
reg_alpha	0
reg_lambda	1
scale_pos_weight	1
seed	None
silent	None
subsample	1
verbosity	1

Division of training and testing data shown in

Table II Table Ii. Dividing The Dataset Into Training Set And Testing Set

Generalization	*Training data*	*Testing data*
G-1(50-50)	49599	50000
G-2(90-10)	89999	9999
G-3(80-20)	79999	19999

C. *Feature Extraction*

We will extract a diverse range of features to train the XGBoost model effectively. This model contains an abundant set of features for training, including graph-extracted features, similarity features, ranking measures, and weight features of the graph's edges.

1) Similarity Measures:

a) Jaccard Distance-The Jaccard distance also called as similarity coefficient which measures similarities between two sets of data which are ranging from 0% to 100%. The higher the percentage, the more similar the two sets. This percentage tells us how similar the sets are. Two sets that share all members would be 100% similar. The closer to 100%, the more similarity [29].

$$j = \frac{IX \cap YI}{IX \cup YI} \quad (2)$$

b) Cosine Distance- Cosine Distance is also a type of similarity measure which is also called as Otsuka-Ochiai coefficient. Cosine Distance is an intersection of the no. of elements in two sets by the dot product of elements multiplied in one set with the elements in another set.

$$\text{Cosine Distance} = \frac{IX \cap YI}{IX.YI} \quad (3)$$

2) Ranking Measures:

PageRank- Based on the analysis and the behavior of arriving edges in the graph, page ranking assigns a specific rank to each node in the graph. When we visit a web page, it will always have a page rank of zero, as long as we don't stop clicking links randomly on any page. A page will be connected to all other web pages despite not departing links if we stop clicking links at one point.

3) Graph Features:

a) Shortest Path- To find the quickest path in a graph between any two nodes, use the shortest path. If an edge connects the nodes without an intermediate node, remove that edge and calculate the path.

b) Checking from same community- We will check if two nodes are from the same community, in which case we will get weakly connected edges from the graph. If both are from the same community, it will return 0; otherwise, it will return 1.

c) Adar Index- From any two vertices in a graph, we will list of the usual and nearest vertices present in both vertices. Then, we will add the degrees of the two nearest vertices, and at last, we will reverse the sum that forms this measure. The Networkx library calculates the Adar index, also known as the Adamic index [30].

$$A(x,y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|} \quad (4)$$

d) Is following back- The following-back feature is defined as whether a person follows another person. We calculate the edge between two nodes, and if there is an edge between them, we return true; if there is no edge, we return false.

e) *Katz Centrality*- This feature computes a central value for any node based on the central or middle values of vertices. This feature is based on Eigenvector centralities, where nodes with high scores contribute more.

$$X_i = \alpha \sum_K a_{k,i} x_k + \beta \quad (5)$$

Where a is the adjacency matrix of the graph G with Eigenvalues [31].

f) Hits Score- The HITS algorithm calculates two numerical values for each node. Authorities compute the node's value based on arriving links. Similarly, hubs compute the value of a node based on departing links.

4) Weight Features

We will calculate the edge weight value from each node and assign it to the edges, a necessary step in the process of identifying similar nodes. If the count of neighbors goes up, then the edge weight goes down, which is inversely proportional. Following an example, if an individual has 50 or more contacts in a community network, then there are higher chances of everyone knowing each other, whereas if an actor or celebrity has 2 million people following him or her in a community network, then there are more chances that they never know each other in the community.

$$W = \frac{1}{\sqrt{1+|X|}} \quad (6)$$

These features include:

- Inner edges weight
- Outer edges weight
- Inner edges weight + Outer edges weight
- Inner edges weight * Outer edges weight
- 2 * Inner edges weight + Outer edges weight
- Inner edges weight + 2 * Outer edges weight

Also, in additional features are added which include:

- Computing page rank feature score for source nodes and destination nodes
- Computing Katz centrality feature score for source nodes and destination nodes
- Computing HITS algorithm score for source nodes and destination nodes

5) Preferential Attachment

Preferential attachment is about individuals who have a large number of friends on social media or social community networks and will have more

friends shortly. We will compute this feature based on how rich a vertex is. If we multiply the number of followers of a vertex by the number of followers of a vertex, by the number of followers of a vertex, we can calculate the richness of a vertex.

6)SVD Features

a) Singular Value Decomposition-We will use this algorithm to compute the Eigenvalues, which are then transformed into Eigen vectors that we can use to divide a matrix into its respective factors. This algorithm is primarily used in machine learning reduction techniques and for performing calculations on a matrix.

b) SVD dot- We calculate the SVD dot as a dot product of the SVD features of the source node and destination node.

D. Training Models

We calculate all these features, discussed in the feature extraction process, for both followers and followees. We have constructed two models, utilizing two distinct machine learning algorithms. The two models built are Random Forest and XGBoost. We train the two models using the above features, including similarity measures, page ranking, graph features, SVD features, etc.

We found that follow-up was the most significant feature for both models. We build the Random Forest model using the Random Forest Classifier, and the XGBoost model using the XGBClassifier.

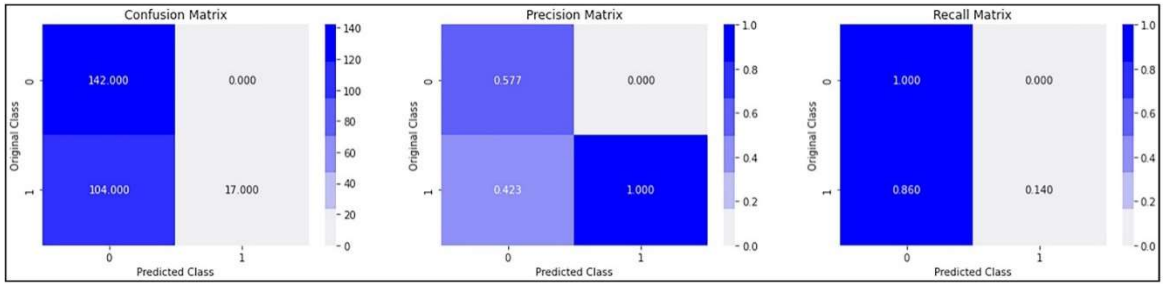

Fig. 1. Confusion Matrix for the train set

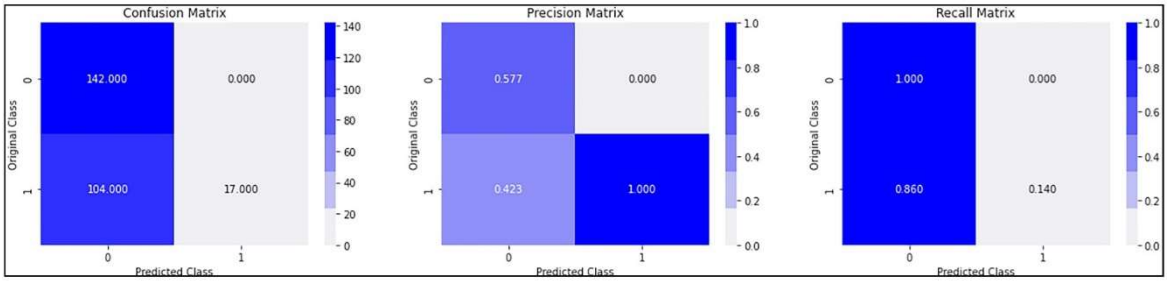

Fig. 2. Confusion Matrix for test set

4. RESULTS AND DISCUSSIONS

Proposed XGBoost is a perfect algorithm for graph-based friend recommendation in social media due to its capacity to handle high-dimensional data, scalability to huge datasets, and delivery of explainable outcomes. By building a graph with users are denoted as nodes and edges signify friendships or connections, XGBoost can be proficient in mined graph-based features such as node degree, centrality, and edge weight to forecast the probability of a user making a new friendship. XGBoost algorithm's robustness to noise, treatment of imbalanced data, and fast training time make it a better choice for social media podiums with masses of users, eventually leading to enhanced accuracy and improved scalability in friend recommendation systems.

We use classification metrics, implied based on their significance, to determine the model's training effectiveness through graph mining. To evaluate the model's performance, we have used different types of performance measures, such as the curve of Receiver Operating Characteristics (ROC), which depicts performance at different threshold levels; the Area Under Curve (AUC), which showcases an appropriate measure of performance; confusion matrices, which specify

parameters in a matrix; and lastly, the F1 score as a performance parameter. We have F1 scores for the training set and the testing set, which showcase the model's performance.

We have used the F1 score as a performance measure for models trained using XGBoost and a Random Forest Classifier, which takes the scores of the training F1-score and the test F1-score.

A) Confusion Matrix

We computed the confusion matrix for both the training and testing sets. Fig.1 describes the Confusion Matrix for the train set and Fig.2. describes the Confusion Matrix for the test set1.0. We calculate precision and recall, then display them as precision and recall matrices.

B) Receiver operating characteristics curve with test data:

The ROC is 0.9999 which is the maximum conceivable value, representing perfect accuracy in Fig. 3.

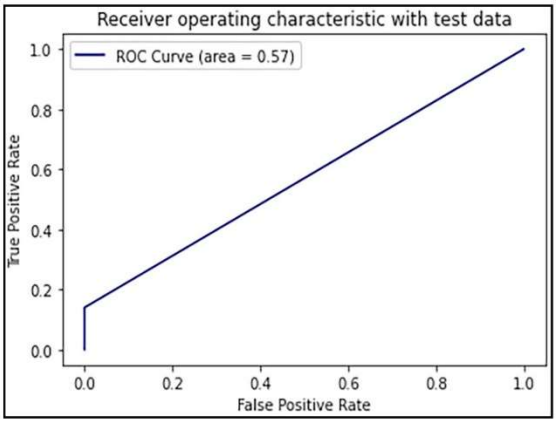

Fig. 3. Receiver Operating Characteristics With Tested Data

C) Feature Importances:

Feature importances are presented in TABLE III.

Table Iii. Feature Importances

Feature	Feature Importance
Num_followers_d	0.8
Num_followers_s	0.1
Weight_out	0.01
Svd_v_d_2	0.005
Prefer_Attach_followers	0.003
Svd_v_d_4	0.003
Weight_in	0.002
Weight_f3	0.002
Svd_u_d_4	0.002
Svd_u_s_4	0.002
Svd_u_s_3	0.002
Svd_v_d_3	0.001
Svd_u_s_6	0.001
Svd_v_d_1	0.001
Svd_v_d_5	0.0001
Svd_u_d_1	0.0001
Svd_u_d_2	0.0001
Katz_d	0.0001
Svd_u_d_5	0.0001
Weight_f4	0.0001
Page_rank_d	0.0001
Hubs_s	0.0000
Svd_u_d_3	0.0000
Authorities_d	0.0000
Svd_u_s_5	0.0000

The XGBoost model has two important features: num_followers_d and num_followers_s, which represent the number of followers for the source node and the destination node, respectively. In r XGBoost model, the feature num_followers_d holds the highest priority, whereas SVD features receive the lowest priority.

Table Iv. Results Of Generalizations Of Models

Generalization	Train data	Test data	Train F1score	Test F1score
G-1(50-50) (Random Forest)	49999	50000	0.996	0.178
G-1(50-50) (XGBoost)	49999	50000	0.998	0.171
G-2(90-10) (Random Forest)	89999	10000	0.997	0.302
G-2(90-10) (XGBoost)	89999	10000	0.999	0.291

G-3(80-20) (Random Forest)	79999	20000	0.995	0.183
G-3(80-20) (XGBoost)	79999	20000	0.999	0.177

We train the two models, Random Forest Classifier and XGBoost, using three types of generalizations: G-1 (50–50), G-2 (90–10), and G3 (80–20). Table IV displays the results of these generalizations.

A) Evaluated on G1, G2, and G3

We trained the Random Forest Classifier and XGBoost models on three generalizations of data, and Table IV shows the train and test f1 scores. The XGBoost model did better than all three random forest classifier generalized models. It scored an f1 score value of 1 in two of the generalizations, G-2 and G-3, compared to the Random Forest Classifier model train scores.

XGBoost is taking more time, but it is providing the best results when compared to the Random Forest Classifier model. The XGBoost model provides better accuracy of all three generalizations because it gives better f1 scores than the Random Forest Classifier. The proposed XGBoost model gives a 99.99% train score and a 99.5% test score in 2 out of 3 generalizations, which are G-2 and G3 on the Facebook recruiting dataset. When the train data is 90% and the test data is 10%, the XGBoost model achieves greater accuracy. Proposed XGBoost method performance may be different on the new dataset and scope is there to improve the accuracy up to i.e 100% and optimization of related parameters on the new context of the algorithm utilization.

XGBoost is an appropriate algorithm for friend recommendation systems due to its qualities, such as holding unevenness data, effective calculation, and feature standing scores, which allow it to learn intricate associations between users. However, its disadvantages, including overfitting and deficiency of interpretability, may lead to recommendations that are not clear or understandable. In contrast to collaborative filtering approaches, XGBoost can integrate added user features and connections, but may not anxiety the nuances of user conduct as efficiently. Meanwhile, graph-based approaches, such as GraphSAGE, can model complex relationships between users, but may be computationally affluent and need careful regulation. Inclusive, XGBoost proposes a robust and effective method for friend recommendation, but should be sensibly assessed and fine-tuned to improve its performance. Proposed method compared with existing method and tabulated in TABLE V.

Table V. Comparison Among Existing Models And The Proposed Method

S.No	Method or Model utilized	Performance Metric
1	Clustering Technique [32]	Accuracy 89.47%
2	Graph-SAGE [33]	AUC 0.89
3	Link Prediction Framework using Deep Neural Network [34]	Accuracy 96%
4	gray wolf algorithm and fuzzy methods [35]	Accuracy 74%
5	LightGBM [36]	Accuracy 93.71%
6	LPXGB [37]	Accuracy 93.04%
7	PersoNet [38]	Precision (0.798) and Recall (0.810)
8	BERT, BiLSTM for Sentiment Analysis of Social Media [39]	Accuracy 99%, 98%
9	**Proposed XGBoost method**	**Accuracy 99.99%**

Proposed XGBoost model provides better accuracy of all three generalizations because it gives better f1 scores than the Random Forest Classifier. The projected XGBoost algorithm provides a 99.99% train score and a 99.5% test score in 2 out of 3 generalizations, which are G-2 and G3 on the Facebook recruiting dataset. When the train data is 90% and the test data is 10%, the XGBoost algorithm attains superior accuracy. Based on performance metrics i.e accuracy and F1-S core the proposed XGBoost algorithm performed well over other methods and all mentioned research objectives are achieved in the order mentioned.

5. OPEN RESEARCH ISSUES

Here are future research directions for researchers to work on

1. Discovering Other Graph-Based Procedures
2. Join in Additional Features
3. Scalability and Efficiency

4. Explainability and Transparency
5. Multi-Modal Friend Recommendation
6. Transfer Learning and Domain Adaptation

6. CONCLUSION

Current social media friend recommendation schemes frequently decrease in taking users' true favourite's precisely, as they principally depend on standing user networks. This inadequate method may overlook possible networks based on common interests, behaviors, or ease stages. To solve this problem, our work recommends a novel Facebook friend recommendation scheme that influences the XGBoost algorithm to analyze users' social graphs, integrate multiple connection parameters, and evaluate various common measures. By filtering graph evidence and computing user rankings, our system provides a more comprehensive and accurate approach to friend recommendations.

We have proposed a XGBoost friend recommendation system that leverages the highly accurate XGBoost boosting algorithm to deliver unparalleled friend recommendations to the target user. By training the model on an impressive 99,999 nodes, a staggering number considering the vast volume of data, our system achieves an astonishing 99.99% accuracy in testing friend recommendations. Furthermore, we train the model on three distinct generalizations of data (G-1, G-2, and G-3), yielding remarkable results with 99.99% accuracy on two and an impressive 99.8% accuracy on the third. To evaluate the model's performance, we employ a comprehensive set of metrics, including the confusion matrix, precision matrix, recall matrix, and F1-score. Notably, the F1-score emerges as the primary metric, showcasing the model's exceptional accuracy in making recommendations. The proposed XGBoost recommendation system outperforms all other friend recommendation systems, achieving an unprecedented 99.99% accuracy and setting a new benchmark in the field. Projected work is able to correlate research objectives with results obtained from the proposed method and also mentioned the future research directions for researchers to work on.

REFERENCES

[1] Li Bian and Henry Holtzman, "Online Friend Recommendation through Personality Matching and Collaborative Filtering", *Computer Science*, pp. 230-235, 2011.

[2] Zhiwei Deng, Bowei He, Chengchi Yu and Yuxiang Chen, "Personalized Friend Recommendation in Social Network Based on Clustering Method", *Computational Intelligence and Intelligent Systems*, pp.84-91, doi.org/10.1007/978-3-642-34289-9_10, 2012.

[3] Josh Jia-Ching Ying, Eric Hsueh-Chan Lu, Wang-Chien Lee, Tz-Chiao Weng and Vincent S.Tseng, "Mining User Similarity from Semantic Trajectories", *2nd ACM SISPATIAL International Workshop on Location-Based Social Networks*, pp. 19-26, doi.org/10.1145/1867699.1867703, , 2010.

[4] Suman Banerjee, Pratik Banjare, Bithika Pal and Mamata Jenamani, "A multistep priority-based ranking for top-N recommendation using social and tag information", *Journal of Ambient Intelligence and Humanized Computing*, 04 August 2020.

[5] Nafis Neehal and M.A. Mottalib, "Prediction of Preferred Personality for Friend Recommendation in Social Networks using Artificial Neural Network", *International Conference on Electrical, Computer and Communication Engineering(ECCE),* pp. 7-9, February, 2019.

[6] Albert Pun,"Graph Convolutional Networks for Friend Recommendation", *CS230: Deep Learning, Winter 2018, Stanford University, CA*, 2020.

[7] Rashmi.J and Dr.Asha. T, "A Friend Recommendation System based on Similarity Metric and Social Graphs", *International Journal of Advanced Networking & Applications(IJANA), ICICN16*, pp. 456-460.

[8] Saajan Sridhar, Mithleshwar Lakhanpuria, Abhishek Charak, Ankur Gupta and Swapan Shridhar, "SNAIR: A Framework for Personalised Recommendations Based on Social Networks Analysis", *ACM SIGSPATIAL LSBN'12*, November 6, 2012.

[9] Ruksar Parveen and N. Sandeep Varma, " Friend's recommendation on social media using different algorithms of machine learning", *Global Transitions Proceedings,* vol. 2, Issue 2, pp. 273-281, ISSN 2666-285X, 2021.

[10] Beg. S., Anjum. A. and Ahmed. M, "User-selectable interaction and privacy features in mobile app recommendation (MAR)", *Multimed Tools and Applications,* 83, 58043–58073,2024.

[11] Ming-Min Shao, Wen-Jun Jiang, Jie Wu, Yu-Qing Shi, TakShing Yum and Ji Zhang, "Improving Friend Recommendation for Online Learning with Fine-Grained Evolving Interest", *Journal of Computer Science and Technology*, 37, 1444–1463,2022.

[12] Li. C., Zhou. B., Lin. W. et al, "A Personalized Explainable Learner Implicit Friend Recommendation Method", *Data Science and Engineering*, 8, 23–35 (2023).

[13] Ivana Andjelkovic, Denis Parra and John O' Donnovan, "Interactive Music Recommendation based on Artists Mood Similarity", *International Journal of Human-Computer Studies*, vol. 121, pp. 142-159, 2019.

[14] Peng Liu, Lemei Zhang and Jon Atle Gulla, "Real-time Social Recommendation on Graph Embedding and Temporal context", *International Journal of Human-Computer Studies*, vol. 121, pp. 58-72, 2019.

[15] Kosaraju Naren Kumar and Kanakamedala Vineela, "Friend Recommendation using Graph Mining on Social media", *International Journal of Engineering Technology and Management Sciences*, vol. .4,doi:10.46647/ijetms.2020.v04i05.011, 2020.

[16] Anuja Shahane and Prof. Rucha Galgali , "Friend Recommendation for Social Networks ", IOSR Journal of Computer Engineering, vol. 18, Issue 5, pp. 37-41,2016.

[17] Fufeng Zheng and Long Ma, "A Multi-Layered Friend Recommendation System on Twitter ", *13th International Conference on Computer Modeling and Simulation,* pp. 258-263, , June 2021.

[18] Shuchuan Lo and Chingching Lin, "WMR – A Graph-based algorithm for Friend Recommendation*", Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, Hong Kong, 18-22 December 2006.

[19] Nitai B. Silva, Ing-Ren Tsang, George D.C. Cavalcanti and Ing-Jyh Tsang, "A Graph-Based Friend Recommendation System Using Genetic Algorithm", *WCCI 2010 IEEE World Congress on Computational Intelligence,* July 18-23, 2010.

[20] Aaron Ling Chi Yi and Dae-Ki Kang, "Experimental Analysis of Friend-And-Native Based Location Awareness for Accurate Collaborative Filtering", *Applied Sciences*, vol.11, no.2, 2510, 2021.

[21] Sharif Vaghefi, Mahyar and Nazareth, Derek L. "Mining Online Social Networks: Deriving User Preferences through Node Embedding", *Journal of the Association for Information Systems,* vol. 22, no.6, 1625-1658, DOI: 10.17705/1jais.00711, 2021.

[22] Z.Wang, C.E Taylor, Q.Cao, H.Qi and Z.Wang, "Friendbook – Privacy Preserving Friend Matching based on Shared Interests*", Proc. of ACM Sensys*, pp. 397-398, 2011.

[23] Xiao Yu, Ang Pan, Lu-An Tang, Zhenhui Li and Jiawei Han, "Geo-Friends Recommendation in GPS-Based Cyber-Physical Social Network", *International Conference on Advances in Social Networks Analysis and Mining*, Kaohsiung, Taiwan, pp. 361-368, doi: 10.1109, 2011.

[24] M Al Hasan, V chaoji and S.Salem, "Link prediction using supervised learning". *SDM'06:Workshop on Link Analysis Counterterrorism and Security*, pp. 798-805, 2006.

[25] N. Benchettara, R. Kanawati and C.Rouveirol, "Supervised machine learning applied to link prediction in bipartite social networks*". Advances in Social Networks Analysis and Mining (ASONAM)-International Conference,* pp. 326-330,2010.

[26] D. Liben Nowell and J.Kleinberg, "The link-prediction problem for social networks", *Journal of the American Society for Information Science and Technology,* vol.58, no.7, pp. 1019-1031, 2007.

[27] Dammavalam Srinivasa Rao, N. Rajasekhar, D. Sowmya, D. Archana, T. Hareesha, and S. Sravya, "Fake News Detection Using Machine Learning Technique", *SCRS Conference Proceedings on Intelligent Systems*, pp. 16-29, 2021.

[28] D Davis, R Lichetenwalter and N.V Chawla, "Multi relational link prediction in heterogeneous information networks", *Advances in Social Networks Analysis and Mining (ASONAM) International Conference*, pp. 281-288, 2011.

[29] S. Varma, S. Shivam, A. Thumu, A. Bhushanam and D. Sarkar, "Jaccard Based Similarity Index in Graphs: A Multi-Hop Approach", *2022 IEEE Delhi Section Conference (DELCON)*, New Delhi, India, pp.1-4, 2022

[30] L.A. Adamic and E.Adar, "Friends and neighbors on the web", *Social networks*, vol 25, No.3, pp. 211-230, 2003.

[31] Zhang. Z., Liu. Y, Ding. W, and Huang. W.W. "A Friend Recommendation System Using Users' Information of Total Attribute", *Data Science. ICDS 2015. Lecture Notes in Computer Science*, vol. 9208. Springer, Cham. https://doi.org/10.1007/978-3-319-24474-7_6.

[32] Upendra Singh, Puja Gupta, "Friend Recommendation on Social Media using Clustering Technique", IJSET, vol.10, no.2, 2022 .

[33] Canwen Jiao, Yan Wang, "Friend Recommendation using GraphSAGE", Medium, 2021.

[34] S. Singh, R. Rajan, S. Nandini, D. Ramesh and C. P. Prathibhamol, " Friend Recommendation System in a Social Network based on Link Prediction Framework using Deep Neural Network", *2nd International Conference on Intelligent Technologies (CONIT)*, Hubli, India, 2022, pp. 1-7, doi: 10.1109/CONIT55038.2022.9848093.

[35] Aida Ghorbani, Ladan Riazi, Amir Daneshvar, Reza Radfar, "Feature engineering with gray wolf algorithm and fuzzy methods for friend recommender system in social networks", *Journal of Industrial and Systems Engineering,* vol. 14, No. 3, pp. 109-120, 2022.

[36] Kini, Shubham, Shivani Pal, Swapnil Kothare, Dhaval Mitna, and Shraddha More. "Friend Recommendation in Social Media App using Link Prediction". *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS),* pp. 1502-1508. IEEE, 2022

[37] Behera, D. K., Das, M., Swetanisha, S., Nayak, J., Vimal, S., & Naik, B. "Follower link prediction using the xgboost classification model with multiple graph features". *Wireless Personal Communications*, pp. 1-20, 2021.

[38] Huansheng Ning, Sahraoui Dhelim and Nyothiri Aung PersoNet, "Friend Recommendation System Based on Big-Five Personality Traits and Hybrid Filtering", *Journal of Latex Class Files*, vol. 14, No. 8, 2015

[39] Sureja, N., Chaudhari, N., Patel, P., Bhatt, J., Desai, T. and Parikh, V, "Hyper-tuned Swarm Intelligence Machine Learning-based Sentiment Analysis of Social Media", *Engineering, Technology & Applied Science Research*, vol.14, no.4, pp. 15415–15421, 2024.

[40] https://www.kaggle.com/c/FacebookRecruiting.