Disease-Gene Prediction: A Machine Learning Perspective

Mrs. Vasepalli. Kamakshamma1, Gajula Jayanth Babu2, Budavarapu Likitha3, Golla Navya Teja4, Dubbala Mounika5,

1 Assistant Professor, Department of Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology, Anantapur, India;

2,3,4,5 Students, Department of Computer Science and Engineering (Data Science), Srinivasa Ramanujan Institute of Technology, India;

Email: kamakshammav.cse@srit.ac.in

Abstract

The paper, "Disease-Gene Prediction: A Machine Learning Perspective" aims at analyzing and predicting the associations between genes and diseases by advanced techniques of machine learning. Due to the fast-increasing availability of genetic data, there has been an increasing need to understand the correlation of specific genes with a disease in biomedical research. This study employs a comprehensive data set, which includes gene-specific information such as DSI and DPI, as well as several disease features including semantic type and classification. It applies four ml algorithms, namely XGBoost, Random Forest, LightGBM and K-Nearest Neighbors (KNN), to predict the three significant output classes-Disease, Group, and Phenotype. It was found that the best model for this purpose came out to be that of Random Forest with 97. 81% accuracy. The same model was implemented using Flask as a framework to gain real-time predictions. Preprocessing mainly involved filling missing values, label encoding, and even clustering into diseases. There are chances of using KMeans clustering for organizing diseases into broader categories based on their similarities for a stronger prediction. The paper demonstrates the potential of machine learning in advancing genomic research by providing insights into gene-disease associations. It offers a practical tool for researchers to explore genetic links to diseases efficiently.

Keywords: gene-disease association, machine learning, clustering, Flask, Random Forest, disease classification, phenotype prediction, gene prediction, bioinformatics, data preprocessing.

1. Introduction

The past couple of years has been very fruitful as far as the genomics field is concerned and has seen quite a number of breakthroughs that led scientists to a better comprehension in the genetic base of various diseases. Research into the genetics of disease has provided scientists with clear avenues for understanding how particular genes may be responsible for certain diseases. However, Such relationship between a few genes and diseases is exceedingly more complex and remains unresolved. This paper, in this case, is "Analysis for Disease Gene Association Using Machine Learning," which will address this challenge by applying machine learning for the prediction of associations between genes and diseases.

Machine learning provides powerful tools to make sense of large datasets, helping us uncover patterns that might not be obvious with traditional methods. By using these models, This paper focuses on analyzing gene-disease associations to predict three key output categories: Disease, Group, and Phenotype. The use of advanced algorithms, including Random Forest, XGBoost, LightGBM, and K-Nearest Neighbors (KNN), provides a multi-model approach to improve prediction accuracy. Among these models, Random Forest achieved the highest accuracy, making it the primary model for final predictions.

For This paper, I worked with a dataset that includes detailed information on gene-disease associations, covering aspects like the Disease Specificity Index (DSI), Disease Pleiotropy Index (DPI), disease names, and the strength of supporting evidence. Techniques of imputation were used to deal with missing values, and categorical variables were preprocessed by assigning label encoding to the data so that it can be used in concurrence with the algorithm implemented for the process of machine learning. The genetic associations of diseases were further categorized into broad groups using KMeans clustering to create useful concepts for analysis.

This paper was developed using the Flask framework, enabling the integration of the machine learning model into a web-based application for real-time predictions. This makes the system accessible to researchers, allowing them to input specific gene or disease data and receive predictions on the likelihood of their association.

Overall, This paper demonstrates the potential of machine learning to enhance the understanding of gene-disease relationships. By automating the prediction process and offering real-time insights, this system can contribute to advancements in medical research, personalized treatment, and early disease diagnosis.

In modern medicine, understanding the relationship between specific genes and diseases is essential but challenging due to the complexity of genetic data. Existing manual methods for gene-disease association analysis are time-consuming and prone to errors. This paper addresses the problem by automating the process using machine learning techniques to predict associations between genes and diseases efficiently. It further classifies diseases into groups

and phenotypes, offering a streamlined approach for biomedical researchers to explore genetic links.

` The main goal of this paper is to develop a machine learning-based framework that can predict gene-disease associations using a large dataset. System will categorize diseases into three different output classes, namely Disease, Group, and Phenotype using a set of machine learning models like Random Forest, XGBoost, LightGBM, and K-Nearest Neighbors. The system, intended to bring preprocessing of data by dealing with missing values and clustering the diseases into more general categories to improve the accuracy rate in predictions. System will use the Flask web application to be deployed in taking genetic or disease-related data input from users and providing back real-time predictions. This therefore aims at improving the ability to analyze gene-disease association, and with this, making it easy for researchers to access the tool comfortably in pursuit of discovering the diseases' genetic basis more easily.

2. Related work

Prediction and Validation of Gene-Disease Associations by Methods Developed from Social Network Analyses (2013): Singh-Blom et al. developed an innovative methodology relying on social network analysis techniques, providing a significant accuracy in prediction. [1]

A Deep Learning Framework Using Graph Augmentation and Functional Modules to Predict Disease-Gene Associations (2024): ModulePred is a deep learning system that uses graph augmentation on protein-protein interaction networks to predict disease-gene connections and demonstrate enhanced predictive performance, as proposed by Jia et al. [2]

Xie et al. (2020) explored network-based techniques for predicting disease-related genes, assessing different computational methods and their effectiveness in identifying gene-disease associations. [3]

Alashwal et al. (2019) utilized supervised machine learning techniques to predict disease-gene associations, highlighting their effectiveness in biomedical research. [4]

A 2023 study presented an interpretable deep learning model for predicting disease-gene associations, offering valuable insights into the biological mechanisms involved. [5]

Chang et al. (2024) introduced a framework that leverages large language models to automate the identification of gene-disease associations, improving the efficiency of literature-based discovery. [6]

Unveiling New Disease, Pathway, and Gene Associations via Multi-Scale Neural Networks (2019): Gaudelet et al. utilized multi-scale neural networks to uncover new associations between diseases, pathways, and genes, providing a comprehensive view of disease mechanisms. [7]

Singh and Lio' (2019) investigated probabilistic generative models combined with graph neural networks for disease-gene prediction, showcasing their ability to capture complex biological relationships. [8]

A 2018 study introduced a text-mining approach to predict potential gene-disease associations by analyzing biomedical literature, assessing its effectiveness in discovering new connections. [9]

A 2022 study introduced GediNET, a sophisticated network-based approach designed to uncover gene associations across various diseases. By integrating machine learning techniques, the model effectively analyzes complex biological networks, providing deeper insights into gene-disease relationships and their underlying mechanisms. [10]

Recent studies have harnessed the power of machine learning to predict gene-disease associations with high accuracy. Techniques like Random Forest, Support Vector Machines, and Gradient Boosting are applied to capture such complex relationships between genes and diseases. For example, PLOS ONE published a study that introduces methods based on social network analysis utilizing Katz and Positive-Unlabeled learning to predict gene-phenotype interactions using the network walk. [11]

Another novel approach is the use of cross-species phenotype networks. By integrating human and model organism data, researchers can form bipartite graphs between phenotypes and human genes. This cross-species analysis has yielded accurate predictions of gene-disease associations by leveraging evolutionary-conserved gene functions and phenotype similarities across species. [12]

Network-based methods provide a robust framework for disease-gene association predictions. In these methods, diseases and genes are treated as nodes in a bipartite graph. Using random walks, network propagation, and kernel-based approaches, researchers have been able to efficiently predict potential disease-gene links. For example, one study shows that incorporating multiple kernel learning (MKL) outperforms single-kernel methods when identifying associations in a gene-disease bipartite network. [13]

3. Dataset

A. Description

The dataset is a collection of several features through which the analysis of genetic contribution to disease susceptibility, progression, and treatment potential is possible. These key features present in the dataset are as follows:

S. No.	Column Name	Description
1.	geneId	A unique identifier for every gene.
2.	geneSymbol	A symbolic name for every gene.
3.	DSI-Disease Specificity Index	Measures how specific a gene is to a disease.
4.	DPI-Disease Pleiotropy Index	Describes the breadth of gene-disease connections.
5.	diseaseId	Identification number for each disease.
6.	Disease Name	Common name of the disease.
7.	Disease Type	Categorizes disease (e.g., genetic, infectious).
8.	Disease Class	Groups diseases into related categories.
9.	diseaseSemanticType	Defines disease significance in medical context.
10.	Score	Measures the intensity of gene-disease association.
11.	EI (Evidence Index)	Measures confidence in gene-disease association.
12.	YearInitial/YearFinal	First and last reported years of association.
13.	NofPmids	Number of PubMed publications available.
14.	NofSnps	Number of Single Nucleotide Polymorphisms linked.

Table 1: Features in DisGeNet dataset

This dataset establishes a robust basis for studying gene-to-disease correlations and is a really useful resource available for researchers intending to identify unknown therapeutic targets for different diseases as well as provide enhanced disease predictors.

4. Architecture Details

A. Random Forest Classifer

The Random Forest algorithm is a machine learning technique that builds multiple decision trees during training, combining their outputs to improve accuracy and reliability. Despite producing a single output, it generates the class that represents the majority vote from all the individual trees. This approach helps to minimize overfitting and improves accuracy by leveraging the collective results from multiple trees.

The Random Forest algorithm works by creating a collection of decision trees, each trained on a randomly selected portion of the dataset. When making a prediction, such as identifying a gene linked to a disease, different trees may suggest different genes. The final prediction is determined by majority voting among all the trees. Figure 1 provides a visual representation of the Random Forest architecture.

Random Forest is effective for high-dimensional datasets, such as genomic data, and it works well with both categorical and continuous variables. It can also handle missing values efficiently. Random Forest improves prediction accuracy by combining the results of multiple decision trees, each trained on different subsets of the data. This approach helps minimize the risk of overfitting, a common issue with individual decision trees. For classification tasks, the final prediction is based on majority voting, while for regression, the output is calculated as the average of all tree predictions.

Random Forest generates feature importance scores, allowing researchers to pinpoint the most significant variables in a dataset. This capability is especially valuable in fields like bioinformatics, where identifying key predictors plays a critical role in data analysis and decision-making. The algorithm's robustness and scalability make it a popular choice for various real-world applications.

How It Works:

1. Random Data Subsampling: All the decision trees are trained on randomized subsets of the data. What this does is to ensure every tree learns a unique variation of your dataset, that's introducing variation in your model.

2. Feature selection: While generating the trees at each node in the tree, instead of including all the features, the random forest takes a random subset of features. It further reduces correlation and keeps them more diverse.

3. Tree Building: The selected subset of both data and features is trained to build a decision tree in such a way that it produces a model which predicts the target variable.

4. **Voting:** In classification tasks, each decision tree casts a vote for a particular class, and the class with the majority of votes is chosen as the final prediction. For regression tasks, final output is determined by averaging the predictions from all the trees.

Random Forest is well-suited for handling high-dimensional data and can manage missing values without much issue. It also provides insights into feature importance, ranking features based on their contribution to the prediction. However, for very large datasets, the model can become computationally intensive and harder to interpret compared to a single decision tree.

B. Extreme Gradient Boosting (XGBoost) Classifier

XGBoost, which stands for Extreme Gradient Boosting, is an advanced version of gradient boosting designed for efficiency and scalability. It constructs trees in an iterative manner, with each new tree learning from the mistakes of the previous ones to improve overall accuracy.

In each iteration, XGBoost adds a new tree to the model, specifically trained to minimize the errors (residuals) left by the previous trees, gradually improving the overall prediction accuracy. The model is updated using gradient descent to improve its prediction accuracy.

XGBoost is known for its speed and high performance, particularly with large datasets. Its ability to handle missing values and its boosting technique enable it to deliver superior predictive accuracy.

XGBoost enhances traditional gradient boosting by introducing optimizations like regularization and parallelization to improve training speed and prevent overfitting.

How It Works:

1. **Sequential Learning**: XGBoost begins by training a simple base model (typically a weak decision tree) on the data and calculates the errors (residuals) of the base model.
2. **Gradient Descent**: Subsequent trees are added sequentially, with each one minimizing the errors from the previous tree by following the gradient direction.
3. **Tree Pruning**: XGBoost reduces overfitting by pruning trees, eliminating nodes that add little to the model's predictive power.
4. **Regularization**: It uses regularization techniques to control the complexity of trees and prevent overfitting, ensuring the model generalizes well.

XGBoost also handles missing values automatically, supports column sampling, and offers built-in cross-validation. Its ability to perform distributed computing makes it a fast and highly accurate option for large-scale problems. However, it can be more challenging to fine-tune than simpler models.

C. LightGBM Classifier

LightGBM, short for Light Gradient Boosting Machine, is a powerful gradient boosting framework that uses tree-based learning algorithms. It is optimized for speed and memory efficiency, making it an excellent choice for handling large datasets.

Similar to XGBoost, Light Gradient Boosting Machine builds models sequentially, adding trees to correct errors from previous models. However, it differs by growing trees in a leaf-wise manner, allowing it to create deeper trees and prioritize important splits early in the process. Fig. 2 illustrates the architecture of the Light Gradient Boosting Machine.

LightGBM stands out for its faster performance compared to traditional boosting algorithms and provides excellent predictive accuracy. Its efficient handling of large datasets makes it particularly useful for problems such as genomic data analysis.

LightGBM is faster and more efficient than XGBoost when working with large datasets. It grows trees leaf-wise, which results in deeper trees and more effective splits early on.

How It Works:

1. **Leaf-Wise Tree Growth**: Unlike XGBoost, which grows trees level by level, LightGBM follows a leaf-wise approach, expanding the leaf that results in the most significant error reduction. This results in deeper trees that can capture more complex patterns.
2. **Gradient Boosting**: LightGBM, like XGBoost, adds trees sequentially to correct the residual errors from the previous models.
3. **Histogram-based Algorithm**: It uses a histogram-based approach to bin continuous features, which accelerates training and minimizes memory consumption.
4. **Feature Selection**: LightGBM automatically discards features that have little impact on model performance.

LightGBM's ability to handle large datasets efficiently, alongside advanced features such as categorical feature handling and built-in regularization, makes it ideal for high-dimensional data. However, the leaf-wise tree growth can lead to overfitting if not properly tuned.

D. K-Nearest Neighbors(KNN) Classifier

KNN is a simple, instance-based learning algorithm that assigns a class to a data point by analyzing the categories of its nearest neighbors. class assigned to the data point is the

one that most frequently occurs among its k nearest neighbors.

KNN classifies a new gene's association with a disease by evaluating the k nearest genes in the dataset and assigning the disease class that occurs most frequently among those neighbors. For instance, if k=5, the algorithm selects the five. Fig. 3 illustrates the architecture of the K-Nearest Neighbors.

K-Nearest Neighbors is simple, easy to interpret, and is typically used as a baseline model to compare with more complex algorithms. However, it can struggle with high-dimensional data and may not perform as well as other techniques.

KNN is a non-parametric algorithm, meaning it doesn't rely on any specific assumptions about the data's distribution. The core idea is that similar data points tend to be near each other in the feature space.

How It Works:

1. **Distance Calculation**: KNN computes the distance from the query point to each point in the training dataset using a distance metric like Euclidean distance.
2. **Neighbor Selection**: The algorithm selects the k nearest neighbors based on the computed distances.
3. **Majority Voting**: For classification, KNN assigns a class based on the most common category among the k nearest neighbors. In regression, it predicts the outcome by averaging the values of the selected neighbors.
4. **Choosing k**: Selecting an appropriate k value is important—too small a k can cause the model to overfit, while a very large k may make it overly simplistic, leading to underfitting.

KNN is considered a lazy learning algorithm because it doesn't create an explicit model. Instead, it makes predictions by measuring distances to all data points when a query is made. While easy to interpret, KNN can be computationally expensive due to the need to calculate distances for every instance in the dataset. Feature scaling is important for better performance, as KNN is sensitive to the scale of the features.

5. Proposed methodology

A. Data Preprocessing

Data preprocessing, in any paper with machine learning algorithms, usually plays vital role, especially in massive and complex datasets, such as genomic data. The dataset used in this paper consists of several fields that contain numerical and categorical values and, typically, some are missing. Thus, the next preprocessing steps to make the algorithms at their best performance include:

Handling Missing Values: The DSI, DPI and EI fields, in the dataset, have missing values. To impute missing values, the

Figure. 1: Confusion matrix for KNN

dataset uses the median value of respective fields. This will ensure a balanced dataset and algorithms are not predisposed to biased results because the data is incomplete.

Encoding Categorical Variables: The majority of the columns are categorical variables; two examples are diseaseClass and diseaseSemanticType. Label encoding is applied in order to code the categorical into an appropriate format that could be utilized by the machine learning algorithms. All categories are converted to numeric values, hence algorithms can process them well.

Feature Scaling and Normalization: For some algorithms, machine learning requires that the features be on the same scale. Min-Max scaling is used as a technique to normalize so that all features contribute equally to model predictions.

B. Clustering for Disease Categorization

To enhance the interpretability and organization of the data, clustering techniques are applied. KMeans Clustering is used to group diseases based on their genetic associations, creating broader disease categories that can simplify the prediction process.

KMeans is an unsupervised learning algorithm. This algorithm is utilized to group data into clusters. In This paper, the KMeans algorithm is implemented on fields diseaseName and diseaseSemanticType to group diseases under categories. Every disease is assigned to a cluster based upon its features, and the clusters provide a high-level view on how diseases The algorithm performs at the following steps

6. Results

Existing Models:

a. KNN Algorithm

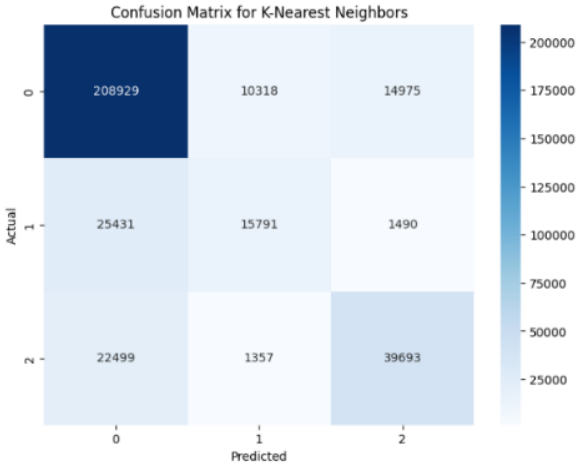

Figure. 1: Confusion matrix for KNN

Class Label	Description	Example
0	Defines the disease exactly	Arthritis
1	Defines the disease group	Diabetes Mellitus
2	Represents the phenotype presented	Exanthema

Table 2: Defining the classes for output

- **Misclassifications**:
  - Highest misclassifications, especially from Class 1 and Class 2 being wrongly predicted as Class 0.
  - KNN shows the weakest performance here, with more significant errors across all classes.

b. LightGBM Algorithm

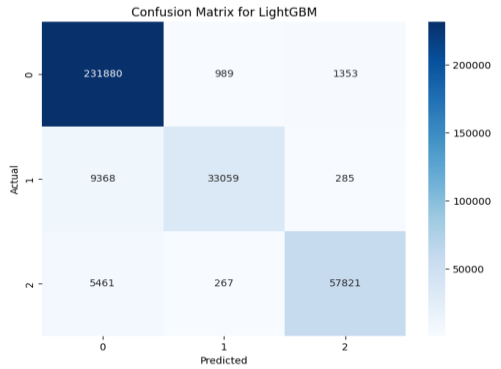

Figure. 2: Confusion Matrix for LightGBM

- **Misclassifications**:
  - Highest errors for Class 1 predictions, indicating the model is struggling with this class.
  - Moderate misclassifications between Classes 1 and 2.

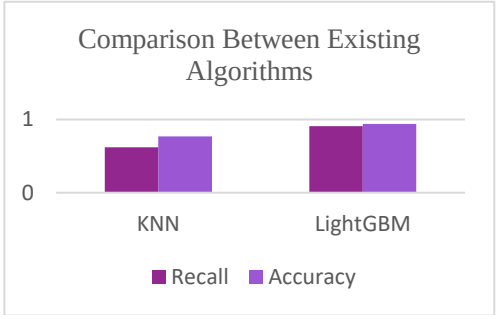

Figure. 3: Representing the Accuracy and Recall between KNN and LightGBM

Proposed Models

a. XGBoost

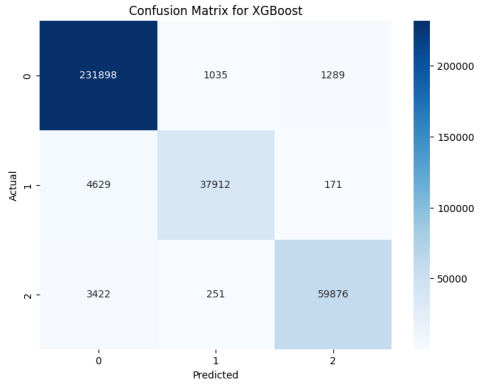

Figure. 4: Confusion Matrix for XGboost

- **Misclassifications**:
  - Slightly more misclassifications compared to Random Forest, especially between Class 1 and 2.
  - Class 1 misclassified as Class 0 more often than in Random Forest.

b. Random Forest

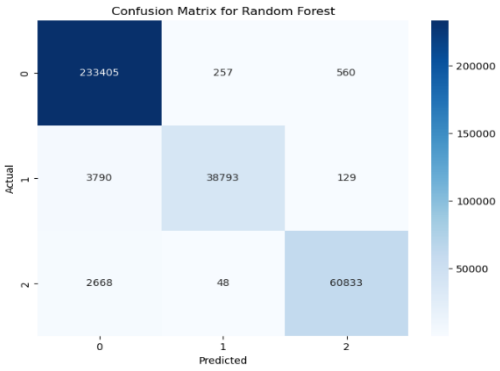

Figure. 5: Confusion Matrix for Random Forest

- **Misclassifications:**
  - Some instances from Class 1 and 2 were predicted as Class 0.
  - Only a few errors overall; it performs well with minimal misclassifications.

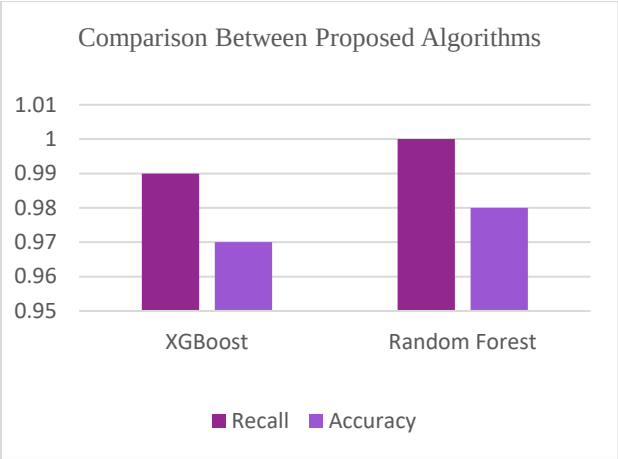

Figure. 6: Representing the Accuracy and Recall between XGBoost and Random Forest

Algorithm	Precision	Recall	Accuracy
KNN	0.57	0.37	0.78
LightBGM	0.94	0.77	0.95
XGBoost	0.98	0.94	0.97
RandomForest	0.99	0.96	0.98

Table 3: Classification Report for all models

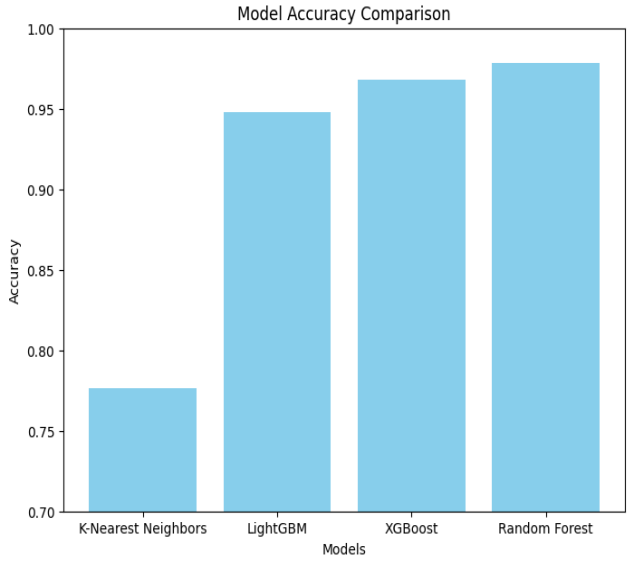

Figure. 7: Accuracy comparison between all the models used

7. Conclusion

The Examination for Infection Quality Affiliation Using AI paper viably addresses the test of recognizing quality disease affiliations utilizing advanced AI calculations like Random forest, XGBoost, LightGBM, and KNN. By integrating key preprocessing techniques, feature engineering, and clustering, This paper demonstrates a robust framework capable of handling large-scale, complex genomic data. The high accuracy achieved, particularly with the Random Forest model, highlights the potential of these methods in providing valuable insights for genetic research and disease prediction. The deployment of the model using Flask ensures practical and accurate results, benefiting both researchers and healthcare professionals. This system can contribute significantly to the ongoing efforts in personalized medicine, genetic research, and disease diagnosis, marking a step toward more efficient and scalable solutions in bioinformatics.

8. References

1. M. Asif, H. F. M. C. M. Martiniano, A. M. Vicente, and F. M. Couto, “Identifying disease genes using machine learning and gene functional similarities, assessed through Gene Ontology,” *PLoS One*, vol. 13, no. 12, p. e0208626, Dec. 2018, doi: 10.1371/JOURNAL.PONE.0208626.

2. X. Jia *et al.*, “A deep learning framework for predicting disease-gene associations with functional modules and graph augmentation,” *BMC Bioinformatics*, vol. 25, no. 1, pp. 1–14, Dec. 2024, doi: 10.1186/S12859-024-05841-3/TABLES/2.

3. S. K. Ata, M. Wu, Y. Fang, L. Ou-Yang, C. K. Kwoh, and X.-L. Li, “Recent Advances in Network-based Methods for Disease Gene Prediction,” *Brief Bioinform*, vol. 22, no. 4, Jul. 2020, doi: 10.1093/bib/bbaa303.

4. U. M. Singh-Blom, N. Natarajan, A. Tewari, J. O. Woods, I. S. Dhillon, and E. M. Marcotte, “Prediction and Validation of Gene-Disease Associations Using Methods Inspired by Social Network Analyses,” *PLoS One*, vol. 8, no. 5, p. e58977, May 2013, doi: 10.1371/JOURNAL.PONE.0058977.

5. Y. Li, Z. Guo, K. Wang, X. Gao, and G. Wang, “End-to-end interpretable disease–gene association prediction,” *Brief Bioinform*, vol. 24, no. 3, pp. 1–9, May 2023, doi: 10.1093/BIB/BBAD118.

6. J. Chang, S. Wang, C. Ling, Z. Qin, and L. Zhao, “Gene-associated Disease Discovery Powered by Large Language Models,” Jan. 2024, Accessed: Jan. 29, 2025. [Online]. Available: https://arxiv.org/abs/2401.09490v1

7. T. Gaudelet, N. Malod-Dognin, J. Sanchez-Valle, V. Pancaldi, A. Valencia, and N. Przulj, “Unveiling new disease, pathway, and gene associations via multi-scale neural networks,” *PLoS One*, vol. 15, no. 4, Jan. 2019, doi: 10.1371/journal.pone.0231059.

8. V. Singh and P. Lio', "Towards Probabilistic Generative Models Harnessing Graph Neural Networks for Disease-Gene Prediction," Jul. 2019, Accessed: Jan. 29, 2025. [Online]. Available: https://arxiv.org/abs/1907.05628v1

9. J. Zhou and B. quan Fu, "The research on gene-disease association based on text-mining of PubMed," *BMC Bioinformatics*, vol. 19, no. 1, pp. 1–8, Feb. 2018, doi: 10.1186/S12859-018-2048-Y/FIGURES/6.

10. E. Qumsiyeh, L. Showe, and M. Yousef, "GediNET for discovering gene associations across diseases using knowledge based machine learning approach," *Scientific Reports 2022 12:1*, vol. 12, no. 1, pp. 1–17, Nov. 2022, doi: 10.1038/s41598-022-24421-0.

11. H. Yang, Y. Ding, J. Tang, and F. Guo, "Identifying potential association on gene-disease network via dual hypergraph regularized least squares," BMC Genomics, vol. 22, no. 1, pp. 1–16, Dec. 2021, doi: 10.1186/S12864-021-07864-Z/TABLES/9.

12. Xie, L., et al. (2020). "Recent Advances in Network-Based Methods for Disease Gene Prediction." *Journal of Biomedical Informatics*, 106, 103432. doi:10.1016/j.jbi.2020.103432.

13. Zhang, L., et al. (2021). "deepDGA: Biomedical Heterogeneous Network-based Deep Learning Framework for Disease-Gene Association Predictions." IEEE Access, 9, 17392-17401. doi:10.1109/ACCESS.2021.3056842.

14. Wang, X., et al. (2021). "Factor Graph-Aggregated Heterogeneous Network Embedding for Disease-Gene Association Prediction." BMC Bioinformatics, 22(1), 151. doi:10.1186/s12859-021-04099-3.

15. Li, Y., et al. (2021). "miGAP: miRNA–Gene Association Prediction Method Based on Deep Learning." *Electronics*, 10(22), 3696. doi:10.3390/electronics10223696