Towards a Better Prescription: Graph AutoEncoder for Drug Recommendation

Mongi Kourchid[1], Olfa Adouni[2] Alaa Bessadok[3], and Nacim Yanes[4]*

[1] Higher Institute of Management of Gabes, University of Gabes, Tunisia
`mongi.kourchid@gmail.com`
[2] Higher Institute of Management of Gabes, University of GabesS, Tunisia
`olfaadouni77@gmail.com`
[3] Institute of Computational Biology, Helmholtz AI, Helmholtz Munich, Germany
`alaa.bessadok@gmail.com`
[4] RIADI Laboratory, La Manouba University, La Manouba, 2010, Tunisia
Higher Institute of Management, University of GABES, Tunisia
`nacim.yanes@univgb.tn`

Abstract. AI models have been widely used as recommender systems in domains such as nutrition, medicine, health status prediction and physical activity. However, their application to drug recommendation is limited due to the complex nature of the medical data stored in Electronic Health Records (EHRs). Existing AI-based drug recommender models either focus solely on current admission EHRs, neglecting historical records, or learn patient representations separately, ignoring the similarities in medical profiles that could enhance drug prediction accuracy. To address these limitations, we propose a novel Graph Autoencoder Drug Recommender (GADR) framework for predicting drugs for patients based on their similarity scores. First, we model the relationships between patients in an EHR system by creating a population graph, optimally capturing the medical similarities of patients. We then reduce the representation of the population by learning the graph embedding using a graph autoencoder. Finally, we train multiple classifiers using the learned population graph embedding to predict the appropriate drug for a particular patient. Our GADR framework demonstrates promising results on the publicly available MIMIC-III dataset achieving 98% accuracy. This framework is generic and can be applied to other EHR-based datasets.

Keywords: Drug recommender systems, Deep Learning, Graph representation learning, Graph neural networks

S. Krichen et al. (eds.), *Proceedings of International Conference on Decision Aid and Artificial Intelligence (ICODAI 2024)*, Atlantis Highlights in Intelligent Systems 12,
https://doi.org/10.2991/978-94-6463-654-3_2

1 Introduction

Artificial intelligence (AI) has made significant strides in revolutionizing various aspects of healthcare, including nutrition, personalized medicine, health status prediction, and physical activity monitoring[1] [2]. However, the application of AI in drug recommendation remains relatively underexplored, primarily due to the complex, heterogeneous, and often incomplete nature of EHRs [1] [2]. EHRs contain comprehensive patient histories, including diagnoses, medications, lab results, and treatment plans, which can provide valuable context and insights for drug recommendations. Existing AI-based drug recommendation systems often focus narrowly on data from current hospital admissions, overlooking the wealth of information contained in historical EHRs[1] [2]. Moreover, these systems typically learn patient representations in isolation, failing to account for the similarities in medical profiles across different patients that could significantly enhance the accuracy of drug recommendations. To address these challenges, we introduce a novel Graph Autoencoder Drug Recommender (GADR) framework designed to leverage the full breadth of similarities between patients at different clinical admission times. Our model not only predicts the drugs that should be prescribed to patients based on the similarities between medical profiles but also considers medical profiles at different timepoints in the EHR. This approach ensures that the model is well-trained on diverse historical records and capturing the evolving nature of patients' medical histories for more accurate and personalized drug recommendations. To the best of our knowledge, no existing AI-based drug recommender model attempted to predict drugs while considering the similarity of diverse historical records. We first propose to construct a graph population that models the relationships between patients within the EHR given their demographic and medical data. This graph effectively captures the evolving medical similarities among patients. Second, we propose to learn the population graph embeddings using a graph autoencoder consisting of an encoder E and a decoder D both defined as a Graph Convolutional Network (GCN). Our graph autoencoder is designed to learn compact and informative embeddings of the patient population, reducing the dimensionality of the input data while preserving its most important and representative features. Lastly, we train a set of classifiers using the previously learned graph embeddings to predict the most appropriate drugs for individual patients. In this way, drug prediction is enhanced by using the reduced and compact representation of the entire population, rather than relying on all the medical features existing in the EHR. The main contributions of our work are as follows:

- Graph population construction: We propose a novel approach to model the relationships between patients in an EHR system by creating a population graph. This method effectively captures the medical similarities of patients.
- Graph representation learning: We introduce a graph autoencoder to reduce the representation of the population, learning an optimal graph embedding that encapsulates the essential features.
- Drug prediction: To predict the appropriate drug for a particular patient, we leverage various classifiers, including Random Forest, Gradient Boosting,

KNN, Naive Bayes, XGBoost and MLP classifier, using the learned population graph embedding.

The remainder of this paper is structured as follows: Section 2 reviews related work that forms the foundation for our method. In Section 3, we detail the framework of our proposed GADR model, outlining its key components and design. Section 4 introduces the dataset utilized in our experiments, explains the experimental setup in detail, and presents the results, accompanied by an in-depth interpretation and discussion of the findings. Finally, Section 5 concludes the paper by summarizing our contributions and offering insights into potential directions for future research.

2 Related Work

Graph Embedding Methods Graph-based patient representation involves creating a compact graph for each patient, with nodes representing clinical events and edges indicating the relationships between these events. Several studies leverage this approach to improve their models by employing graph embedding techniques. For example, [3] introduced the SMR framework, which creates graph embeddings to recommend safe medicines for patients, especially those with multiple diseases. It builds a heterogeneous graph by integrating EMRs with medical knowledge from ICD-9 and DrugBank, and then embeds diseases, medicines, and patients into a shared lower-dimensional space to support medicine recommendations. [4] proposed MedGCN, a system that uses graph-based embeddings to recommend medications and estimate missing lab values from incomplete tests. By integrating various medical entities into a heterogeneous graph and using GCNs to learn their representations, the system captures complex relationships for improved recommendations. [5] presented WSNN, a novel signed directed graph neural network model designed to learn node embeddings for weighted signed networks. Using the embeddings generated by this model, it performs link sign prediction in signed networks. Another work [6] enhanced graph autoencoders by addressing their dependence on the quality of the adjacency matrix. The authors developed two unsupervised graph embedding methods, BAGE and VBAGE, incorporating an adaptive learning mechanism to self-learn the adjacency matrix. This innovation improves robustness, particularly when the original graph structure is incomplete or noisy. By embedding latent representations into the Laplacian graph structure, these methods better preserve the graph's topology in vector space, improving performance in tasks like node clustering, classification, and graph visualization. Moreover, [7] introduced Gemini, a neural network-based method for generating embeddings from the control flow graph of binary functions. Gemini encodes each function as a numeric vector, enabling efficient similarity detection by comparing these vectors. It significantly improves accuracy, accelerates embedding generation and training, and identifies more vulnerable firmware images than current methods. Recently, [8] presented INGRAM to generate embeddings for both new entities and new relations during inference. Unlike existing methods that focus solely on new entities, INGRAM constructs a relation graph and uses an attention mechanism to aggregate embeddings from neighboring nodes. This allows the model to dynamically generate embeddings for new entities and relations, significantly improving its ability to handle real-world knowledge graphs where new entities and relations often emerge.

Population Graphs for Patient Outcome Prediction Several authors have explored the use of population graphs in combination with GNNs as an effective approach for modeling patient relationships and predicting clinical outcomes by representing the intricate interactions and similarities among individuals within healthcare data. For instance, [9] evaluated a framework that integrates both imaging and non-imaging data for brain analysis in large populations using Graph Convolutional Networks (GCNs). The approach represents populations as a sparse graph, where nodes correspond to imaging-based features and edges incorporate phenotypic information as weights. Additionally, [10] introduced a novel spectral domain architecture for deep learning on graphs, specifically for disease prediction. The key innovation is the use of geometric 'inception modules' that can capture both intra- and inter-graph structural heterogeneity during convolutions. The architecture employs filters with different kernel sizes to enhance its ability to analyze graph structures. Similarly, [11] presented a method for predicting disease outcomes by leveraging multiple population graphs, each constructed from different combinations of patient attributes. The approach involves using a recurrent graph convolutional neural network architecture to analyze these graphs, with statistical tests determining the most relevant graphs for each medical feature. The method is applied to predict the progression of Alzheimer's disease in patients with mild cognitive impairment, demonstrating that incorporating multiple graphs enhances the predictive accuracy. In another example, [12] proposed a graph convolutional neural network model designed to improve disease prediction by effectively integrating multi-modal data, including both imaging and non-imaging information. The model leverages a novel self-attention layer to weigh demographic data elements according to their relevance to the disease, enabling a more nuanced analysis of patient conditions. This graph-based approach enhances both computational efficiency and predictive performance, outperforming state-of-the-art methods. Several studies have also aimed to enhance graph structure. Some, like [13], that introduced a generalizable framework for integrating imaging and non-imaging data within population graphs to enhance uncertainty-aware disease prediction. Central to this framework is a learnable adaptive population graph with variational edges, optimized through graph convolutional neural networks. The method introduces Monte-Carlo edge dropout to estimate predictive uncertainty related to graph topology. While others, focused on learning the optimal graph in an end-to-end manner, such as [14] that presented a novel approach to disease classification using Graph Convolutional Networks (GCNs) by automatically learning an optimal population graph. Unlike traditional methods that manually define patient similarities based on demographic or clinical data, this approach uses an end-to-end trainable architecture to dynamically and locally adjust the graph structure for improved accuracy in disease prediction. The learned population graph enhances classification performance, particularly in medical applications, by allowing the model to effectively generalize to new, unseen patients, and [15], which presented the Differentiable Graph Module (DGM), a novel approach for inferring graph structures directly from data, particularly in settings where the underlying graph is noisy, incomplete, or entirely unknown. DGM is a learnable function that predicts edge probabilities in a graph, optimizing the structure for the downstream task. It can be integrated with convolutional graph neural networks and trained end-to-end. Additionally, other research has addressed challenges such as missing data in [16] which introduced a method for disease classification and progression prediction in large population-based studies by addressing the challenge of missing data. The approach leverages graph convolutional neural networks (GCN) to model multi-modal clinical data as a matrix completion problem. By arranging subjects in a graph structure, the method simultaneously performs classification and non-linear imputation of missing features through geometric matrix completion, simulating a heat diffusion process, and imbalanced data in [17] which introduces the Re-weighted Adversarial Graph Convolutional Network (RA-GCN), a model designed to prevent bias in graph-based classifiers by avoiding the overemphasis on samples from any particular class. RA-GCN achieves this by associating a separate graph-based neural network with each class to reweight the importance of class samples dynamically. The classifier and the weighting networks are trained end-to-end using an adversarial approach, allowing the model to better determine class boundaries by focusing on the most relevant samples.In conclusion, modeling clinical data with population graphs has emerged as a promising method for predicting patient outcomes.

3 Methodology

Our method is designed to predict drugs for patients given their medical and demographic information. We extract such data from an EHR dataset that is composed of the clinical records of N patients. We describe in (**Fig. 1**) the three steps of our GADR drug recommendation model: 1) Graph Population Construction, 2) Graph Representation Learning, and 3) Drug Recommendation.

Graph population construction We start by constructing a graph population representing the patient data. In this graph, each patient is defined as a node, and the edges between nodes indicate the similarities in clinical features, shared diagnoses, and co-occurrences in medical procedures. As illustrated in (**Fig. 1**-1) we define our graph population as $G = (N, E, \mathbf{A}, \mathbf{F})$, where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a similarity matrix capturing the pairwise relationships between n subjects (patients), and $\mathbf{F} \in \mathbb{R}^{n \times f}$ is a feature matrix where f is the dimension of each node's feature vector. In our context, $\mathbf{F}$ represents the clinical features of patients, and $\mathbf{A}$ captures the similarity between patients based on their medical records (**Fig. 1**-2). To ensure consistency across various feature types, we utilize cosine similarity, a widely used metric that measures how similar two vectors are by effectively

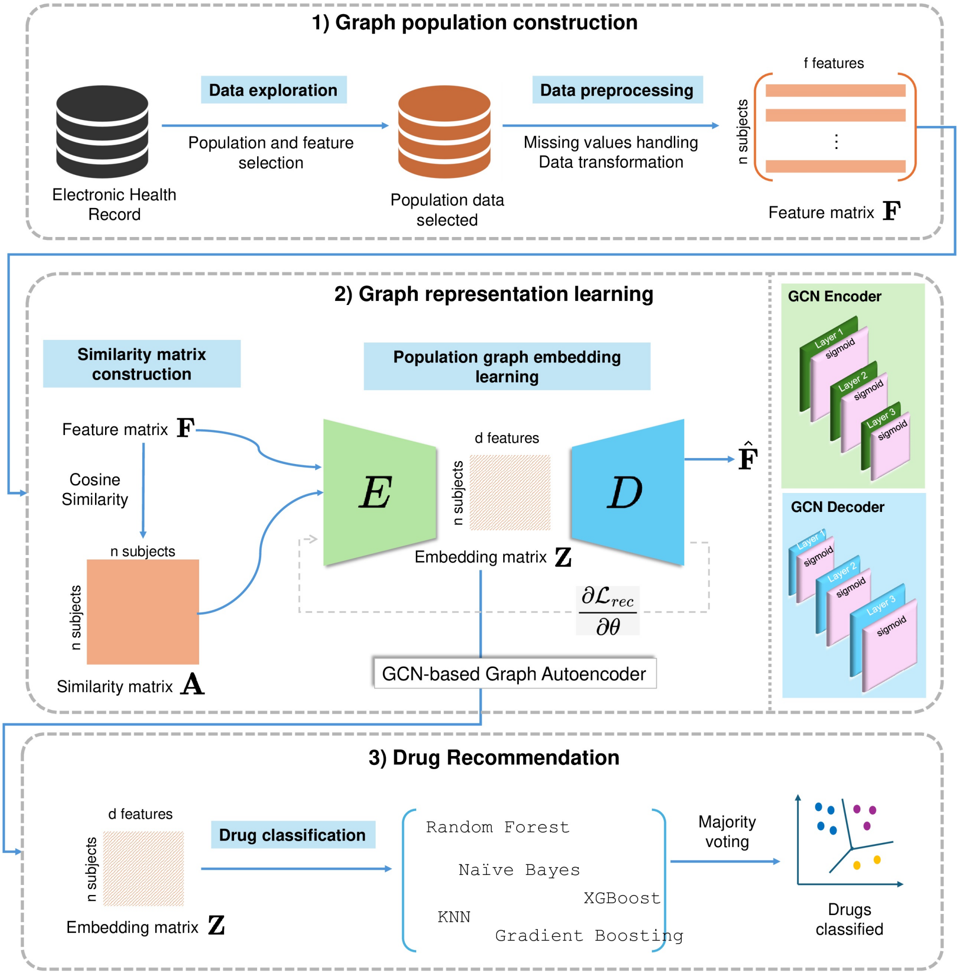

Fig. 1. Overview of our GADR architecture for drug recommendation

calculating the dot product of two normalized entities. For each node pair n_i and n_j representing the records r_i and r_j, we calculate a similarity score $S(r_i, r_j)$ between the features to determine the edge weight $e_{i,j}$. Hence, we calculate the similarity between subjects as follows:

$$S(r_i, r_j) = \frac{r_i \cdot r_j}{\|r_i\| \|r_j\|} \tag{1}$$

The choice of cosine similarity over other measures like Euclidean distance or Jaccard similarity is driven by its ability to handle high-dimensional and sparse data effectively. In the context of EHRs, where patient features may vary greatly in scale and sparsity, cosine similarity focuses on the directional alignment of feature vectors rather than their magnitude, making it less sensitive to large variations in feature values. This is particularly important when dealing with medical data, where some features may dominate due to their numerical values. Cosine similarity normalizes this effect, ensuring that the relational patterns between patients are more accurately captured. By contrast, Euclidean distance is highly sensitive to feature magnitudes and scales, which may distort the patient relationships in the graph, especially when certain features are disproportionately large. Jaccard similarity, while useful for binary data, may be less suitable for continuous and multi-dimensional clinical data, where nuanced feature similarities are better captured by cosine similarity. Graph construction is crucial as it allows us to capture different levels of patient similarity and interactions, providing a comprehensive view of the underlying clinical relationships. The use of cosine similarity ensures that these relationships are quantified in a consistent and meaningful way, making our graph-based approach unique and robust. This methodology not only highlights the interconnectedness of patients based on their clinical features but also enhances the accuracy of drug recommendations by leveraging these nuanced relationships.

Graph representation learning The second phase of our model involves transforming the graph population into a low-dimensional embedding matrix $Z \in \mathbb{R}^{n \times d}$, which captures the vital features and structures of the graph. This process, known as graph representation learning (**Fig. 1**-2), is essential to ensure that the key information needed for accurate drug prediction is retained while reducing the complexity of the data. To achieve this, we use a graph autoencoder, which consists of two key components:

- A GCN encoder: This encodes the graph data into a latent space (embedding) by applying a series of convolution operations.
- A GCN decoder: This decodes the embedding back to the original graph space, aiming to reconstruct the graph.

GCNs are neural networks specifically designed for graph-structured data. Unlike traditional neural networks, which work on grid-like structures (e.g., images), GCNs take into account the relationships between nodes in a graph. A GCN layer applies a graph convolution to update the feature representation of each node based on its neighbors' features. Thus we define the propagation rule of the GCN model as follows:

$$\mathbf{Z} = f_{\phi}(\mathbf{F}, \mathbf{A}|\mathbf{\Theta}) = \phi(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{F}\mathbf{\Theta}) \tag{2}$$

Here, $\mathbf{Z}$ is the learned graph representation resulting from a specific GCN layer. ϕ denotes the activation function, such as Rectified Linear Unit (ReLU), although different activation functions can be used for different layers. $\mathbf{\Theta}$ represents the graph convolutional weights. The graph convolution function $f(.)$ is defined where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ with $\mathbf{I}$ being an identity matrix, and $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$ is a diagonal matrix.

The graph autoencoder operates by encoding the graph G into a latent space through a series of graph convolutional layers. The encoding process transforms the graph data into a lower-dimensional embedding that retains the essential features and structures. The autoencoder then decodes the embedding back to the original graph space, aiming to reconstruct the graph. To ensure the embedding retains meaningful information, the reconstruction loss is minimized. We choose the mean squared error (MSE) between the original feature matrix $\mathbf{F}$ and the reconstructed feature matrix $\hat{\mathbf{F}}$ to be the loss function:

$$\mathcal{L}_{rec} = ||\mathbf{F} - \hat{\mathbf{F}}||^2 \tag{3}$$

By minimizing this loss, the graph autoencoder learns a compact and informative representation of the graph population. The learned embedding $\mathbf{Z}$ is used in the next step as input to a classifier, which predicts the most suitable drugs for each patient based on the low-dimensional representations. This step utilizes the rich, condensed information captured by the graph autoencoder to make accurate and reliable drug recommendations.

Drug Recommendation The drug recommendation phase is the final and crucial stage of our drug recommender system (**Fig. 1**-3). In this phase, we leverage the embedding matrix generated during the representation learning stage, incorporating it with the label column (drug) to accurately predict the most suitable medication for each patient profile. Our primary objective is to improve the precision and reliability of our drug recommendations by employing a set of machine learning and deep learning classifiers: Random Forest, Gradient Boosting, K-Nearest Neighbors, Naive Bayes, XGBoost, and MLP Classifier. We pick the best drug classification result by majority voting of results given by the classifiers.

4 Experiments

4.1 Dataset preparation

In our analysis of the MIMIC III database, we focused on a sample of 250 subjects, examining various categories of data. This includes Identification & Basic

Info, which provides fundamental details about each subject; Demographic Information, covering age, gender, and other personal characteristics; Clinical Data, which encompasses medical history, diagnoses, and treatment details; Medication Data, detailing the drugs prescribed and administered; and Other Information, which includes additional relevant data not categorized elsewhere. This comprehensive approach allows us to leverage a broad range of information for our analysis. Although there are 26 tables in the database, we only consider some of them. The tables relevant to the project include : `ICUSTAYS`, `PATIENTS`, `PRESCRIPTIONS`, `ICD_PROCEDURES`, `PROCEDURES_ICD`, `ADMISSIONS`, `D_ICD_DIAGNOSES`, `LABEVENTS`, `D_LABITEMS`, `MICROBIOLOGYEVENTS` and `DIAGNOSES_ICD`. We examined the selected tables in the MIMIC-III dataset to handle problems such as missing values, imbalanced labels and non normalised values. We start by a **feature engineering** step. We create new features that can potentially improve the model performance. For instance, we calculate patient age at admission from birth dates and admission dates, and we select patients within the age range of 15 to 90 years. Indeed, as illustrated by Fig 2, we found an age range between 15 and 146 years, which is not logical, and the number of patients above 90 years is very limited. Therefore, we decided to limit the age distribution to between 15 and 90 years by converting ages above 90 to 90. Fig.3 shows the new distribution of patients' ages. These engineered features are designed to capture important clinical insights and improve the predictive power of our models.

Table 1. Features used to create the graph population

Category	**Number of Features**
Identification & Basic Info	10
Demographic Information	8
Clinical Data	12
Medication Data	9
Other Information	4

Next we assessed the **quality control** of our data to ensure accuracy, consistency, and reliability throughout the data lifecycle. A common challenge is the presence of special characters, like slashes (//) and semicolons (;), which can disrupt data integrity and analysis. To address this problem, we replace these special characters with spaces, resulting in cleaner and more consistent data. Next, we address **missing values** problem using two main approaches. We first discard samples with missing values when their proportion is minimal to avoid impacting the analysis. Second we apply imputation techniques such as mode imputation for categorical data where missing values are replaced with the most frequent value, and mean imputation for numerical data where missing values are substituted with the mean of the variable. After handling missing, data we perform **data transformation** to ensure the data is clean, consistent,

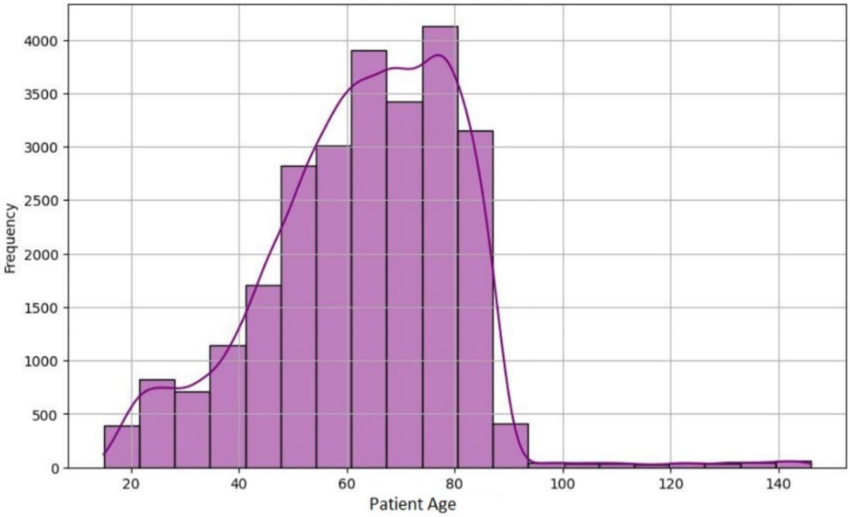

Fig. 2. Age Distribution of Patients

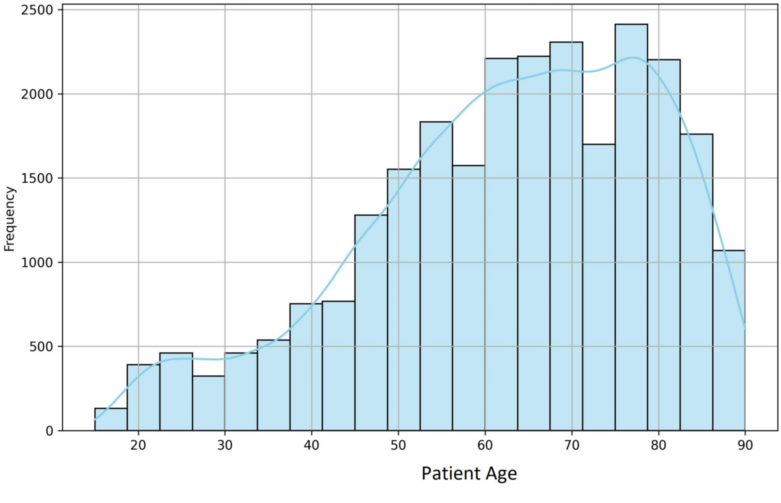

Fig. 3. Age Distribution of Patients after setting Age limit

and compatible with target systems. We use MinMaxScaler and StandardScalar to normalize numerical features to a range of [0, 1], and OneHotEncoder and LabelEncoder to convert categorical variables into binary vectors. In addition, we **handle imbalanced data**. In drug classification tasks, class imbalance can hinder model performance by under-representing minority classes. Therefore, we use the RandomOverSampler method, which increases the representation of minority classes by replicating their instances. This technique helps balance the class distribution compared to many other techniques including the undersampling [18] [19] and the oversampling method [20] - [21], enhancing the model's ability to classify drugs accurately across all classes. The clean dataset used to create the graph population contains 250 patients and 35 features obtained from 11 different tables within the MIMIC-III database.

4.2 Model training

Our Graph Autoencoder comprises an encoder and a decoder both modeled as GCNs. The encoder is composed of three layers: the first layer with 64 neurons, the second with 32 neurons, and the final layer with 16 neurons that produces the embeddings. The decoder follows the same structure in reverse, aimed at accurately reconstructing the input features. We incorporated a dropout rate of 0.5 to mitigate overfitting and enhance generalization. The model is optimized with a learning rate of 0.01 and weight decay of 5e-4, and we utilized a learning rate scheduler to adjust the rate according to the validation loss. Early stopping is applied with a patience of 20 epochs to cease training if no substantial improvement is detected. We use 3-fold cross-validation with grid search to improve our classifiers performance. Cross-validation divides the data into subsets, training on some and testing on others to evaluate generalization and prevent overfitting. Grid Search explores various hyperparameter settings exhaustively, finding the best combinations through cross-validation. This combination provides a more reliable evaluation and optimal settings, leading to models that perform well on both training and new data, ensuring efficient fine-tuning and robust results.

4.3 Model evaluation

In order to predict the drugs for each patient we proposed to leverage the Voting Classifier approach which is an ensemble technique in machine learning that enhances classification model performance by combining the predictions of multiple base models. Specifically, in our GADR framework we used the "soft" voting method which aggregate the results of several optimized models including Random Forest, Gradient Boosting, K-Nearest Neighbors, Naive Bayes, XGBoost, and MLP classifiers. To evaluate the performance of the classifiers in predicting drugs, we used five metrics: precision, recall, F1-score, accuracy, and Jaccard index. TABLE II demonstrates that Random Forest and Gradient Boosting are the top-performing models for predicting drugs given the learned graph embedding by our graph autoencoder. XGBoost also performs very well and could be a viable alternative. The classification results illustrated in Fig. 5 reveal that en-

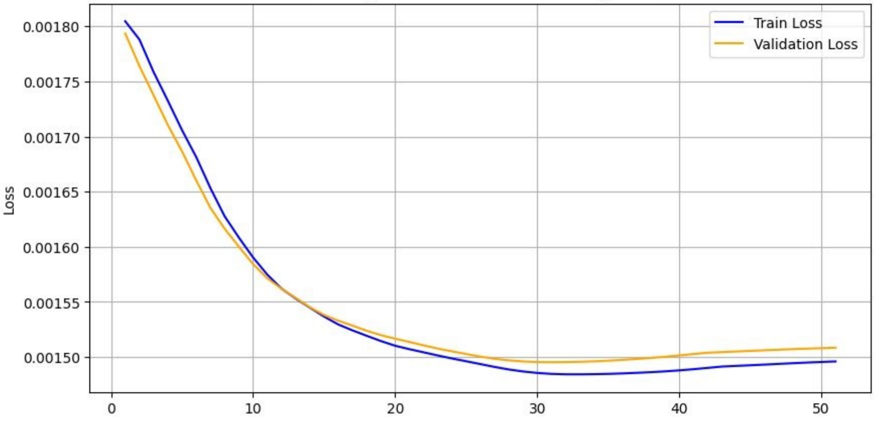

Fig. 4. Training and Validation Loss over Epochs

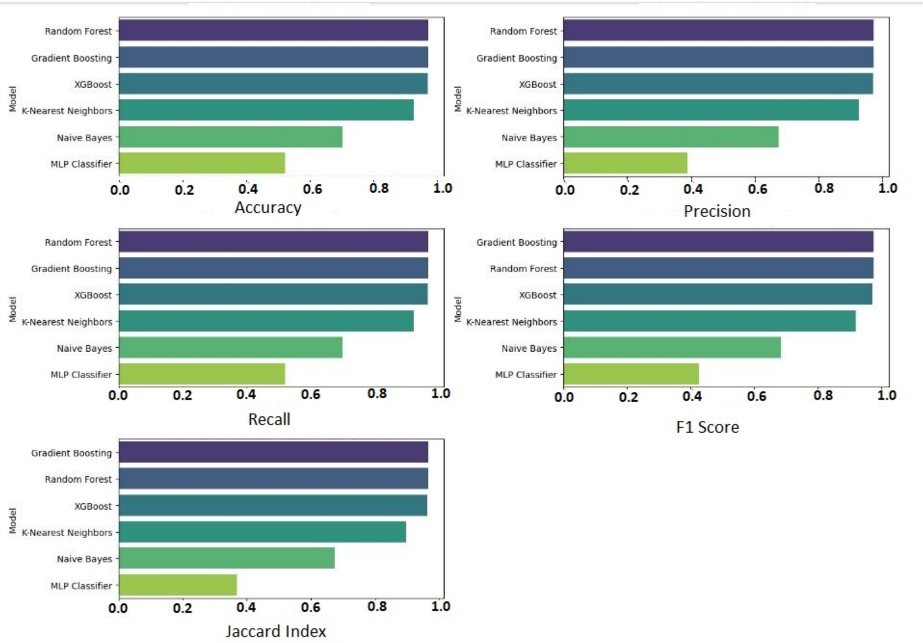

Fig.5. Comparative Analysis of Classifiers Performance in GADR

Table 2. Drug classification results

Model	Accuracy	Precision	Recall	F1 Score	Jaccard
RF	**0.98**	**0.97**	**0.98**	**0.98**	**0.97**
GB	**0.98**	**0.97**	**0.98**	**0.98**	**0.97**
KNN	0.93	0.93	0.93	0.92	0.90
NB	0.71	0.67	0.71	0.68	0.67
XGBoost	0.98	0.97	0.98	0.97	0.96
MLP	0.60	0.47	0.60	0.50	0.43

RF:Random Forest, GB: Gradient Boosting, KNN: K-Nearest Neighbors, NB: Naive Bayes, MLP: Multi-Layer Perceptron

semble models, particularly Random Forest and Gradient Boosting, outperform the others in terms of performance, achieving high scores in precision, recall, F1 score, and Jaccard index. These metrics were chosen to provide a well-rounded evaluation of the model's performance, as they each address critical aspects of drug recommendation accuracy in healthcare. We detail each metric as follows:

- **Precision** measures the proportion of correctly predicted drugs among those that were predicted. A high precision value indicates that the model is effective at minimizing false positives, which is crucial for avoiding the prescription of incorrect drugs that could lead to adverse effects [22]:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

- **Recall** assesses how well the model identifies the correct drugs from all possible true drug candidates. In the healthcare context, high recall is vital to ensure that essential medications are not overlooked:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

- **F1-score** provides a balance between precision and recall, particularly when both are critical to avoid both false positives and false negatives. This is especially important when predicting medications where both aspects are essential for patient safety [22]:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Jaccard index** measures the overlap between the predicted and actual drug sets, indicating how well the model's predictions match the true recommendations [22]:

$$\text{Jaccard Index} = \frac{|A \cap B|}{|A \cup B|}$$

A high Jaccard index reflects a strong agreement between the predicted and actual drug recommendations. These ensemble models excel at balancing precision and recall, making them particularly effective for this classification task.

In contrast, the MLP classifier shows significantly lower performance across all metrics, indicating that it is less suitable for this specific context. These findings highlight the efficiency and robustness of ensemble methods for achieving accurate and reliable classifications in drug recommendation.

5 Conclusion

In conclusion, our Graph Autoencoder Drug Recommender (GADR) framework effectively addresses key limitations of existing models by incorporating historical data and leveraging patient similarities. This approach not only achieves a 98% accuracy rate on the MIMIC-III dataset but also paves the way for more accurate and personalized drug recommendations. By improving patient outcomes and enhancing the integration of AI into clinical decision support systems, our framework represents a significant advancement in healthcare.

While GADR is designed to be versatile and applicable across various EHR datasets, there are potential challenges when dealing with highly heterogeneous data or datasets with sparse connections between patient records. In such cases, differences in clinical features may affect generalization, and sparse patient relationships could weaken the graph learning process. To address these challenges, techniques such as feature selection, graph augmentation, and incorporating external clinical knowledge can be used to enhance the model's robustness. Additionally, scalability and missing data issues may arise with larger datasets, but methods like graph sampling and data imputation can help mitigate these limitations. Despite these challenges, the GADR framework demonstrates strong potential for broader application, making it a valuable tool in advancing personalized healthcare and AI-driven clinical decision-making.

References

1. Junaid Bajwa, Usman Munir, Aditya Nori, and Bryan Williams. Artificial intelligence in healthcare: transforming the practice of medicine. *Future healthcare journal*, 8(2):e188–e194, 2021.
2. Talha Iqbal, Mehedi Masud, Bilal Amin, Conor Feely, Mary Faherty, Tim Jones, Michelle Tierney, Atif Shahzad, and Patricia Vazquez. Towards integration of artificial intelligence into medical devices as a real-time recommender system for personalised healthcare: State-of-the-art and future prospects. *Health Sciences Review*, page 100150, 2024.
3. Fan Gong, Meng Wang, Haofen Wang, Sen Wang, and Mengyue Liu. Smr: medical knowledge graph embedding for safe medicine recommendation. *Big Data Research*, 23:100174, 2021.
4. Chengsheng Mao, Liang Yao, and Yuan Luo. Medgcn: Medication recommendation and lab test imputation via graph convolutional networks. *Journal of Biomedical Informatics*, 127:104000, 2022.
5. Zekun Lu, Qiancheng Yu, Xia Li, Xiaoning Li, and Qinwen Yang. Learning weight signed network embedding with graph neural networks. *Data Science and Engineering*, 8(1):36–46, 2023.

6. Rui Zhang, Yunxing Zhang, Chengjun Lu, and Xuelong Li. Unsupervised graph embedding via adaptive graph learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5329–5336, 2022.
7. Xiaojun Xu, Chang Liu, Qian Feng, Heng Yin, Le Song, and Dawn Song. Neural network-based graph embedding for cross-platform binary code similarity detection. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 363–376, 2017.
8. Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. Ingram: Inductive knowledge graph embedding via relation graphs. In *International Conference on Machine Learning*, pages 18796–18809. PMLR, 2023.
9. Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrero, Ben Glocker, and Daniel Rueckert. Disease prediction using graph convolutional networks: application to autism spectrum disorder and alzheimer's disease. *Medical image analysis*, 48:117–130, 2018.
10. Anees Kazi, Shayan Shekarforoush, S Arvind Krishna, Hendrik Burwinkel, Gerome Vivar, Karsten Kortüm, Seyed-Ahmad Ahmadi, Shadi Albarqouni, and Nassir Navab. Inceptiongcn: receptive field aware graph convolutional network for disease prediction. In *Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2–7, 2019, Proceedings 26*, pages 73–85. Springer, 2019.
11. Juliette Valenchon and Mark Coates. Multiple-graph recurrent graph convolutional neural network architectures for predicting disease outcomes. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3157–3161. IEEE, 2019.
12. Anees Kazi, Shayan Shekarforoush, Karsten Kortuem, Shadi Albarqouni, Nassir Navab, et al. Self-attention equipped graph convolutions for disease prediction. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1896–1899. IEEE, 2019.
13. Yongxiang Huang and Albert CS Chung. Edge-variational graph convolutional networks for uncertainty-aware disease prediction. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VII 23*, pages 562–572. Springer, 2020.
14. Luca Cosmo, Anees Kazi, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein. Latent-graph learning for disease prediction. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23*, pages 643–653. Springer, 2020.
15. Anees Kazi, Luca Cosmo, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael M Bronstein. Differentiable graph module (dgm) for graph convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1606–1617, 2022.
16. Gerome Vivar, Andreas Zwergal, Nassir Navab, and Seyed-Ahmad Ahmadi. Multimodal disease classification in incomplete datasets using geometric matrix completion. In *Graphs in Biomedical Image Analysis and Integrating Medical Imaging and Non-Imaging Modalities: Second International Workshop, GRAIL 2018 and First International Workshop, Beyond MIC 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 2*, pages 24–31. Springer, 2018.

17. Mahsa Ghorbani, Anees Kazi, Mahdieh Soleymani Baghshah, Hamid R Rabiee, and Nassir Navab. Ra-gcn: Graph convolutional network for disease prediction problems with imbalanced data. *Medical image analysis*, 75:102272, 2022.
18. Max A Little. *Machine learning for signal processing: data science, algorithms, and computational statistics.* Oxford University Press, USA, 2019.
19. S Nagul and RK Kumar. An effective k-means approach for imbalance data clustering using precise reduction sampling. *Int. J. Comput. Sci. Eng.*, 6(3):65–70, 2018.
20. Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. Distributional random oversampling for imbalanced text classification. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 805–808, 2016.
21. Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. Ieee, 2008.
22. Satvik Garg. Drug recommendation system based on sentiment analysis of drug reviews using machine learning. In *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 175–181. IEEE, 2021.